UNIVERSIDAD DE LA REPÚBLICA URUGUAY

FACULTAD DE INGENIERÍA

Herramienta de apoyo para la aplicación de la metodología CaDQM, centrada en la evaluación de la calidad de datos

Informe de Proyecto de Grado presentado por

Mauricio Xavier, Francisco Cosco

en cumplimiento parcial de los requerimientos para la graduación de la carrera de Ingeniería en Computación de Facultad de Ingeniería de la Universidad de la República

Supervisores

Flavia Serra
Adriana Marotta

Montevideo, 26 de noviembre de 2025

Agradecimientos

Esta tesis está dedicada y es fruto del apoyo incondicional y el aliento constante de nuestras familias a lo largo de todos estos años de carrera, a quienes agradecemos profundamente.

Agradecemos también a los amigos que nos han acompañado y motivado en este camino.

Nuestro más sincero agradecimiento se dirige también a nuestras tutoras, Flavia Serra y Adriana Marotta, por su guía, dedicación y paciencia a lo largo de este proyecto.

Resumen

La evaluación de la calidad de datos (CD) es una actividad esencial dentro de cualquier proceso de gestión de CD. Dicho proceso no es aislado, sino que se enmarca en una secuencia de actividades que incluye la definición de un modelo de CD, la ejecución de mediciones de CD, y la evaluación final a través de los valores de CD obtenidos. De acuerdo con la bibliografía, estas actividades están influenciadas por el contexto en el que se utilizan los datos, de modo tal que la calidad varía según distintos escenarios. Sin embargo, en la práctica, pocas metodologías abordan la gestión de la CD considerando explícitamente dicho contexto a lo largo de todas sus etapas.

Para abordar dicha problemática, este proyecto se basa en la metodología *Context-aware Data Quality Management* (CaDQM), propuesta en la tesis de doctorado de Flavia Serra, para formalizar la integración del contexto en la gestión de la CD. Mas precisamente, el presente proyecto se enfoca en la Fase 2 - *DQ Assessment* de CaDQM, centrada en la evaluación de CD y su aplicación. Con tal propósito, se propone el desarrollo de una herramienta para dar soporte a expertos en CD en la ejecución de todas las actividades necesarias para la evaluación, partiendo de la definición de un modelo de CD basado en un modelo de contexto. Además, aborda la implementación de métodos de CD para la medición y obtención de metadatos de calidad. Dicha solución consiste en una aplicación web implementada con *Angular* para el cliente, y *Python* para la lógica de negocio y procesamiento de datos, y *PostgreSQL* como sistema de gestión de bases de datos. Adicionalmente, la herramienta incorpora funcionalidades de inteligencia artificial (IA), integradas para apoyar al experto en CD en el proceso de definición del modelo de CD.

Finalmente, la solución propuesta se valida a través de dos casos de estudio donde se realiza la ejecución completa de la herramienta desarrollada. El primer caso utiliza un *dataset* sobre libros de *Amazon* y un contexto definido manualmente. El segundo caso emplea datos reales médicos, basándose en un modelo de contexto generado por la herramienta que implementa la Fase 1 de CaDQM, la cual fue desarrollada por otro proyecto de grado. Este segundo caso verifica la interoperabilidad entre las herramientas propuestas por dos proyectos de grado diferentes, como punto de partida para una herramienta unificada de gestión de CD sensible al contexto.

Palabras clave: Calidad de Datos, Contexto, Inteligencia Artificial, Metodología de Calidad de Datos.

Índice general

Capítulo 1

Introducción

En este capítulo se introducen las características generales del proyecto, describiendo la motivación, los objetivos generales y específicos que guiaron este trabajo. Finalmente, se presenta la estructura del documento.

1.1. Motivación

En la actualidad, los datos se han convertido en uno de los activos más valiosos de las organizaciones, las cuales, en su gran mayoría, basan sus decisiones en la información derivada de ellos. Dado ello, la calidad de los datos (CD) resulta esencial para garantizar la confiabilidad de los procesos de toma de decisiones, ya que una mala calidad puede conducir a conclusiones erróneas o decisiones inadecuadas, impactando en la eficiencia y eficacia de las organizaciones.

Lograr una buena CD no es una tarea sencilla, sino que requiere un proceso sistemático de gestión de la CD que involucra diversas actividades, entre ellas, su evaluación. Un aspecto clave a tener en cuenta es que la CD no corresponde a una propiedad intrínseca del dato, sino que depende del contexto en el que este es utilizado. A pesar de la relevancia del contexto, el relevamiento de la bibliografía muestra que pocas metodologías de gestión de CD lo consideran explícitamente en sus actividades. Esta falta de contextualización puede conducir a diagnósticos incompletos o incluso erróneos sobre el estado real de la calidad de los datos, al no reflejar adecuadamente las condiciones o requerimientos específicos de cada entorno. Por ejemplo, un mismo valor puede ser considerado correcto o incorrecto según el contexto: un rango permitido de valores, un nivel de granularidad requerido o un porcentaje aceptable de datos faltantes puede variar según el propósito del proceso que utiliza esos datos. En algunos escenarios, un 20 % de valores nulos puede ser tolerable, mientras que en otros es completamente inaceptable. Para realizar estas evaluaciones se emplean métricas de CD que cuantifican aspectos como validez, completitud o actualización mediante porcentajes u otros indicadores. Los resultados de estas métricas sirven para determinar, según los umbrales definidos por el contexto, si los datos cumplen o no con el nivel de calidad requerido.

Frente a esta limitación, la metodología *Context-aware Data Quality Management* (CaDQM), definida en la tesis de Doctorado de Flavia Serra [29], propone un marco de trabajo que integra explícitamente el contexto en todas las actividades del proceso de gestión de CD, incluyendo la evaluación. Dado que la aplicación de esta metodología requería un extenso trabajo de ejecución manual, se identificó la necesidad de contar con una herramienta que facilite dicho proceso y brinde soporte a los expertos en CD durante la aplicación de la metodología.

1.2. Descripción del proyecto

Dada la motivación de contar con una herramienta que brinde soporte a los expertos en CD en la aplicación de la metodología CaDQM a lo largo de sus diferentes fases, este proyecto de grado se centra en el desarrollo de una herramienta que ejecute la Fase 2 - *DQ Assessment*. Esta herramienta utiliza como entrada, entre otros elementos, un modelo de contexto generado por una herramienta enfocada en la Fase 1 - *DQ Planning*, desarrollada en paralelo por otro proyecto de grado. Más precisamente, la Fase 2 se centra en la evaluación de la CD de un mismo conjunto de datos empleado en la Fase 1, basándose en el modelo de contexto y en la lista de problemas de CD definidos en la Fase 1, almacenados en una base de datos compartida entre ambas herramientas para asegurar su interoperabilidad.

La herramienta propuesta permite ejecutar las tres etapas correspondientes de la Fase 2 de manera iterativa, que consisten en la definición de un modelo de CD (*DQ Model Definition*), la ejecución de métricas de CD mediante métodos CD para su medición (*DQ Measurement*), y finalmente la evaluación de la CD (*DQ Assessment*). Además, en el proceso de definición del modelo de CD, la solución incorpora funcionalidades basadas en inteligencia artificial (IA) que asisten al usuario en actividades específicas. Estas funcionalidades incluyen la recomendación de dimensiones y factores de CD a partir de problemas de CD priorizados y componentes de contexto, así como la generación automática métodos de CD a partir de la definición de las métricas de CD que implementan.

1.3. Objetivos

El objetivo general de este proyecto de grado consiste en diseñar y desarrollar una herramienta que dé soporte a los expertos en CD en la aplicación de la Fase 2 - *DQ Assessment* de la metodología de gestión de CD dependiente del contexto, *Context-aware Data Quality Management* (CaDQM).

Para lograr dicho objetivo general, se plantean los siguientes objetivos específicos:

- Realizar un análisis de herramientas existentes para la aplicación de metodologías de calidad de datos.
- Diseñar una herramienta en base al análisis realizado y al relevamiento de requerimientos, centrado en la Fase 2 de la metodología CaDQM, asegurando la interoperabilidad con otra herramienta implementada en paralelo por otro proyecto de grado, encargado del desarrollo de la Fase 1 (*DQ Planning*).
- Implementar un prototipo de la herramienta propuesta.
- Validar la herramienta implementada mediante la aplicación de dos casos de estudio, con el fin de verificar:
  - La correcta ejecución de las tres etapas de la Fase 2.
  - La interoperabilidad entre las herramientas desarrolladas por ambos proyectos.
  - La correctitud del modelo de CD obtenido con la herramienta.

1.4. Resultados esperados

Como resultado de este proyecto, se espera obtener un prototipo de una herramienta para la ejecución de la Fase 2 - *DQ Assessment* de la metodología CaDQM, centrada en la evaluación de la calidad de datos (CD). La herramienta debe permitir al experto en CD avanzar de manera progresiva e iterativa por cada una de las etapas de la fase abordada, con el fin de definir un modelo de CD sensible al contexto, ejecutar mediciones de CD y, finalmente, evaluar la CD a partir de los resultados obtenidos. Asimismo, la herramienta debe ser interoperable con el prototipo desarrollado en el otro proyecto de grado correspondiente a la Fase 1, mediante un diseño de base de datos común. Esto permitirá consumir y utilizar, como entrada en la Fase 2, los datos generados en la fase anterior. Se entregará una documentación completa del proyecto, que incluirá el diseño, la implementación y la validación de la herramienta, así

como un manual de usuario detallado en el que se dispondrá el acceso a la herramienta implementada y las instrucciones necesarias para su utilización.

1.5. Estructura del documento

El presente documento se organiza de la siguiente manera:

- Capítulo 2: presenta el marco teórico y principales conceptos abordados en el proyecto, incluyendo trabajos relevantes previos.
- Capítulo 3: describe el análisis y relevamiento de requerimientos realizados para el desarrollo de la herramienta.
- Capítulo 4: expone el diseño y la arquitectura de la herramienta, destacando los elementos esenciales de la solución propuesta.
- Capítulo 5: detalla la implementación de la herramienta, explicando su funcionamiento y detalles relevantes de su construcción.
- Capítulo 6: aborda la integración de inteligencia artificial en la herramienta y la experimentación realizada de las mismas.
- Capítulo 7: presenta los casos de estudio utilizados para validar la herramienta y su interoperabilidad con la Fase 1.
- Capítulo 8: expone las conclusiones y trabajo futuro.

Además, se incluyen los siguientes anexos:

- Anexo A: se presenta el manual de usuario con instrucciones de instalación, configuración y uso de la herramienta.
- Anexo B: se presenta una documentación técnica para el *frontend* y *backend* de la herramienta desarrollada.
- Anexo C: se presentan resultados de experimentación utilizados la prueba y validación de las funcionalidades basadas en inteligencia artificial.
- Anexo D: se presentan el modelo de contexto y los problemas de calidad de datos utilizados en el Caso de Estudio 1.
- Anexo E: se presentan el conjunto de datos, el modelo de contexto y los problemas de calidad de datos utilizados, así como el modelo de calidad de datos construido con la herramienta como resultado del Caso de Estudio 2.

Capítulo 2

Marco Teórico

En este capítulo se presentan los principales conceptos de calidad de datos (CD) y su relación con el contexto. Tras una revisión de trabajos relevantes en el área, se introduce la metodología *Context-aware Data Quality Management* (CaDQM) propuesta por Flavia Serra en su tesis doctoral [29], en la que se basa el presente trabajo. Esta metodología propone y define un metamodelo de CD sensible al contexto que integra los conceptos de CD y de contexto. Además, se detallan las fases y etapas de CaDQM, estableciendo las bases teóricas para el desarrollo de este proyecto de grado.

2.1. Calidad de Datos

La calidad de los datos (CD) es un concepto fundamental en la gestión de la información. Si bien no existe una definición universalmente consensuada, uno de los más utilizados es que los datos deben ser *fit for use*, es decir, que sean capaces de cumplir con los requisitos del entorno en el que se utilizan. Esta perspectiva sostiene que la CD se relaciona directamente con la confiabilidad y utilidad de los datos para el análisis, la toma de decisiones y la generación de conocimiento. Sin datos de buena calidad, las organizaciones y las personas pueden enfrentarse a errores en sus conclusiones, desperdicio de recursos y dificultades en la optimización de procesos [19].

En particular, este trabajo se basa fuertemente en los conceptos de CD presentados en el curso "Calidad de Datos e Información" de la Facultad de Ingeniería (UdelaR) [30], así como en un trabajo presentado en la conferencia "ER" [31]. A continuación, se detallan los principales conceptos de CD utilizados:

Modelo de Calidad de Datos (*DQ Model*): Se construye como el conjunto de dimensiones, factores, métricas y métodos de calidad de datos, conceptos de CD organizados jerárquicamente como se ilustra en la Figura 2.1.

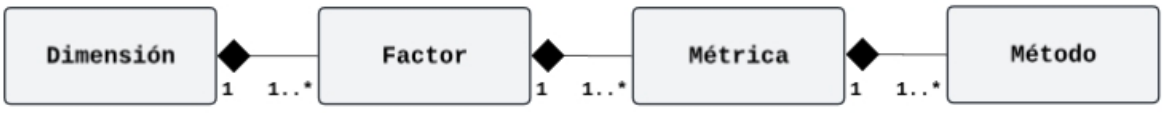

Figura 2.1: Jerarquía de conceptos de CD.

Dimensión de Calidad de Datos (*DQ Dimension*): Representa facetas de alto nivel que caracterizan la calidad de los datos. Si bien existen diversas propuestas de dimensiones según diferentes autores, las comúnmente utilizadas y presentadas en el curso "Calidad de Datos e Información"[30], son las siguientes:

- **Exactitud *(Accuracy)***: Grado en que los datos representan fielmente las entidades del mundo real o los valores verdaderos, considerando aspectos sintácticos (formato), semánticos (significado) y de precisión.
- **Completitud *(Completeness)***: Medida en que todos los datos requeridos están presentes y disponibles, considerando la densidad de valores no nulos y la cobertura del dominio requerido.
- **Consistencia *(Consistency)***: Uniformidad y ausencia de contradicciones en los datos, tanto dentro de un mismo conjunto (intra-relacional) como entre diferentes conjuntos (inter-relacional).
- **Frescura *(Freshness)***: Perspectiva temporal que indica qué tan actualizados y vigentes son los datos para su uso.
- **Unicidad *(Uniqueness)***: Ausencia de representaciones redundantes de una misma entidad del mundo real.

Factor de Calidad de Datos (*DQ Factor*): Un factor especifica aspectos particulares de una dimensión de CD. Cada dimensión puede refinarse en un conjunto de factores que describen sus diferentes facetas. A continuación, se presentan algunos de los factores más representativos para las dimensiones previamente definidas:

- **Exactitud** se evalúa mediante:
  - **Exactitud Semántica (*Semantic Accuracy*)**: Correspondencia entre el dato y su significado real.
  - **Exactitud Sintáctica (*Syntactic Accuracy*)**: Conformidad con reglas de formato y estructura.
  - **Precisión (*Precision*)**: Nivel de detalle adecuado para el propósito.
- **Completitud** se compone de:
  - **Densidad (*Density*)**: Proporción de valores no nulos en el conjunto.
  - **Cobertura (*Coverage*)**: Alcance de los datos respecto al dominio requerido.
- **Consistencia** incluye:
  - **Integridad de Dominio (*Domain Integrity*)**: Cumplimiento de valores válidos según reglas de dominio.
  - **Integridad Intra-relacional (*Intra-relational Integrity*)**: Coherencia interna dentro de un mismo conjunto.
  - **Integridad Inter-relacional (*Inter-relational Integrity*)**: Coherencia entre conjuntos relacionados.
- **Frescura** considera:
  - **Vigencia (*Currency*)**: Grado de actualización temporal de los datos.
  - **Oportunidad (*Timeliness*)**: Disponibilidad cuando son requeridos.
  - **Volatilidad (*Volatility*)**: Frecuencia de cambios esperados.
- **Unicidad** se verifica con:
  - **No-duplicación (*Non-redundancy*)**: Ausencia de registros redundantes.
  - **No-contradicción (*Non-contradiction*)**: Consistencia entre representaciones alternativas.

Métrica de Calidad de Datos (*DQ Metric*): Instrumento que define la forma de medir un factor de CD, obteniendo un valor de CD correspondiente. Una métrica de CD se define mediante la especificación de: (i) la propiedad concreta que se mide, (ii) el dominio del resultado (valores *booleanos*, porcentajes o rangos), y (iii) la granularidad (celda, tupla, columna, tabla).

A continuación se presentan dos ejemplos de métricas de CD definidas en el curso "Calidad de Datos e Información" [30], para medir los factores de CD: Exactitud Semántica y Densidad, respectivamente.

- **ExacSemantica-Bool**:
  - Descripción: Verifica si un dato existe en la realidad.
  - Dominio del resultado: $\{0, 1\}$ (*boolean*)
  - Granularidad: Celda individual
- **Densidad-Grado**:
  - Descripción: Calcula la proporción de valores no nulos en una columna.
  - Dominio del resultado: [0..1] (donde 1 = 100 % completo)
  - Granularidad: Columna completa

Método de Calidad de Datos (*DQ Method*): Es un procedimiento o algoritmo que permite implementar una métrica de CD.

2.1.1. Importancia de la Calidad de los Datos

La motivación para buscar calidad en los datos se encuentra en su impacto directo en la eficacia y eficiencia de los procesos organizacionales. Disponer de datos de buena calidad ofrece beneficios clave:

- **Reducción de costos**: La mala CD puede generar grandes pérdidas económicas para las organizaciones. Problemas de CD como información incorrecta puede llevar a procesos ineficientes, retrabajo. y potenciales errores en la toma de decisiones [4].
- **Mejora en los procesos operativos**: La falta de datos precisos y disponibles a tiempo puede dificultar la planificación y ejecución de servicios. Contar con datos de buena CD permite una mejor planificación, resultando en servicios más eficientes, como se ejemplifica en un caso de estudio relacionado con la medicina [19].
- **Cumplimiento normativo**: Datos incompletos o inconsistentes pueden impedir el cumplimiento de regulaciones. Una deficiente CD puede llevar a incumplir normativas crediticias o gubernamentales [21].
- **Impacto en el entrenamiento de modelos de IA**: Diversos estudios demuestran que un conjunto de datos pequeño pero de alta CD puede ser más efectivo para el ajuste fino de un modelo de IA que un conjunto de datos más grande pero de menor CD [18].

2.2. Relación entre Calidad de Datos y Contexto

Este trabajo surge como una implementación de la tesis de doctorado de Flavia Serra, *Context-aware Data Quality Management* [29]. En esta, se plantea la necesidad del uso del contexto de los datos a la hora de definir un modelo de CD.

De acuerdo con una revisión sistemática de la literatura sobre el concepto de contexto, realizada en la tesis de Serra [29], se destaca que:

- El contexto es una fuente de información que ha sido poco utilizada en entornos de computación.
- No existe una definición única, ya que se aborda desde diferentes perspectivas, tales como la presentación, la ubicación, el usuario o la comunidad, dependiendo del foco de interés.

Por su parte, para superar esta ambigüedad y permitir su uso práctico en la gestión de la CD, Serra propone una definición concreta de contexto que puede instanciarse en diferentes organizaciones según

sus realidades. El contexto es entendido como el conjunto de elementos que caracterizan la situación de uso de los datos.

A partir de este marco, se establece que la CD es dependiente del contexto. Los modelos de CD se definen a través de múltiples dimensiones, y éstas deben seleccionarse en función del escenario y del *dataset* específico. El contexto permite definir métricas personalizadas teniendo en consideración aspectos como el dominio de aplicación, los usuarios finales o los objetivos del análisis. Por ejemplo, en un contexto médico, la exactitud de los datos puede ser prioritaria, mientras que en un contexto de marketing, la frescura y la relevancia pueden tener mayor peso. Este marco conceptual lleva a la necesidad de formalizar el contexto, lo que se aborda mediante el metamodelo presentado en la Sección 2.4.

2.2.1. Trabajos Relevantes

Como se mencionó previamente, este trabajo se desarrolló en el marco de la tesis de doctorado de [29]. Si bien la bibliografía principal consultada proviene de esta tesis, a continuación se presenta una revisión de otros trabajos relevantes que han abordado la calidad de datos y su relación con el contexto, destacando sus enfoques, limitaciones y aportes.

Los autores [2] han planteado un enfoque formal para modelar la calidad de los datos en función de su contexto, introduciendo la idea de generar versiones limpias alternativas de una base de datos y medir la calidad a partir de su distancia respecto a estas versiones contextuales. Este modelo permite mejorar la respuesta a consultas asegurando que solo se utilicen datos que cumplan con criterios previamente definidos. Sin embargo, el contexto que se menciona no está definido formalmente, lo que limita su aplicabilidad general.

Según [6], la evaluación de la calidad de datos en entornos de *Big Data* enfrenta múltiples desafíos debido al volumen, variedad y velocidad con que se generan los datos. En este sentido, un enfoque adaptativo resulta clave para ajustar las métricas y técnicas de evaluación según el contexto específico en que se utilicen los datos. Dada la dificultad de evaluar el conjunto de datos completo, el trabajo plantea una técnica de optimización que busca el mejor equilibrio entre la minimización del tiempo y del presupuesto, y la maximización de la confianza en los resultados. Un caso de estudio a partir de datos reales de una ciudad inteligente ha demostrado cómo esta aproximación puede optimizar la gestión de datos urbanos, considerando factores como la estructura y el origen de los datos.

En el ámbito del Gobierno Digital en Uruguay [5], han desarrollado un *framework* para gestionar la calidad de los datos, con un modelo de referencia que define dimensiones como exactitud, consistencia y completitud. Este *framework* establece un proceso estructurado que incluye la caracterización técnica y de negocio, la medición de la CD y la implementación de planes de mejora, demostrando su utilidad en la optimización de la toma de decisiones y la prestación de servicios públicos.

Finalmente, desde una perspectiva más teórica, [9] busca identificar los desafíos y requisitos para construir un marco de evaluación de calidad de datos basado en el contexto en entornos de *Big Data*. Para ello, se realiza una revisión de alcance de las soluciones existentes, evidenciando que ninguna aborda de manera integral el contexto de los datos y el manejo de grandes volúmenes de información. Como resultado, se proponen recomendaciones y un marco metodológico para diseñar servicios de evaluación de CD basados en el contexto. Entre los principales desafíos, se destacan la estandarización de dimensiones de CD y la seguridad de los datos, entre otros.

En resumen, los trabajos revisados abordan la calidad de datos desde perspectivas diversas, desde métodos adaptativos para entornos de *big data* hasta marcos institucionales para la gestión de CD, pero presentan una limitación común: la ausencia de una definición formal y sistemática del contexto que pueda integrarse de manera directa en el proceso de evaluación de la CD.

2.3. Metamodelo de CD

En esta sección se presenta el metamodelo de calidad de datos *(DQ metamodel)* propuesto por [29] en su tesis de doctorado.

Este metamodelo, ilustrado en la Figura 2.2, organiza los conceptos de CD ya definidos en la sección anterior que componen el modelo de CD, incluyendo dimensiones, factores, métricas y métodos de CD. Además introduce el concepto de Método Aplicado de CD *(Applied DQ Method)*. Este último se define como la aplicación de un método de CD a un conjunto de datos específico. Los métodos aplicados pueden ser de dos tipos: de medición *(Measurement DQ Method)*, para computar un valor de CD, o de agregación *(Aggregation DQ Method)*, para consolidar múltiples resultados.

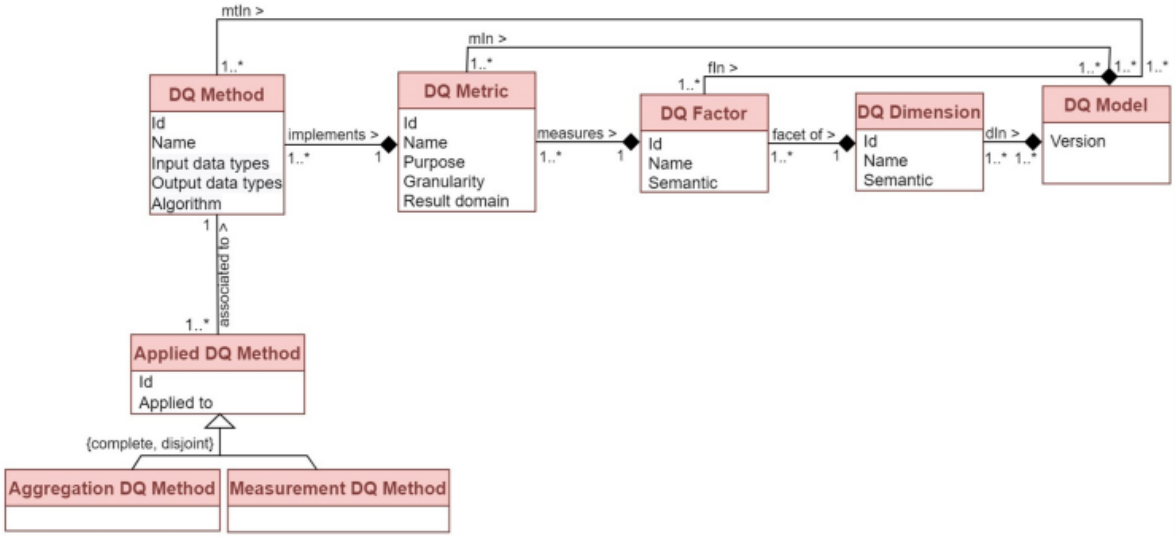

Figura 2.2: Metamodelo de CD extraído de [29].

2.4. Metamodelo de Contexto

El metamodelo de contexto define el conjunto de componentes que determinan el contexto de los datos cuya calidad se evalúa. El metamodelo se presenta en la Figura 2.3 y sus elementos se describen a continuación:

- **Modelo de Contexto (*Context Model*)**: Conjunto de componentes contextuales.
- **Dominio de Aplicación (*Application Domain*)**: Área donde se evalúan los datos (Ejemplo: Salud, Farmacología).
- **Reglas de Negocio (*Business Rules*)**: Restricciones en los datos, por ejemplo: 'El antibiótico debe ser "vanco" si la concentración en líquido cefalorraquídeo no es nula.'
- **Tipos de Usuarios (*User Types*)**: Clasifica usuarios según su rol (Ejemplo: "Investigador", "Médico").
- **Tarea en Mano (*Task at Hand*)**: Propósito y uso de los datos.
- **Requerimientos de Calidad de Datos (*DQ Requirements*)**: Especificaciones que los datos deben cumplir.
- **Filtrado de Datos (*Data Filtering*)**: Selección de datos relevantes (puede expresarse en SQL).
- **Requerimientos del Sistema (*System Requirements*)**: Condiciones del sistema donde se manejan los datos.

- **Metadatos de Calidad de Datos (*DQ Metadata*)**: Resultados de mediciones de CD previas.
- **Otros Metadatos y Datos (*Other Metadata & Other Data*)**: Información adicional sobre los datos evaluados.

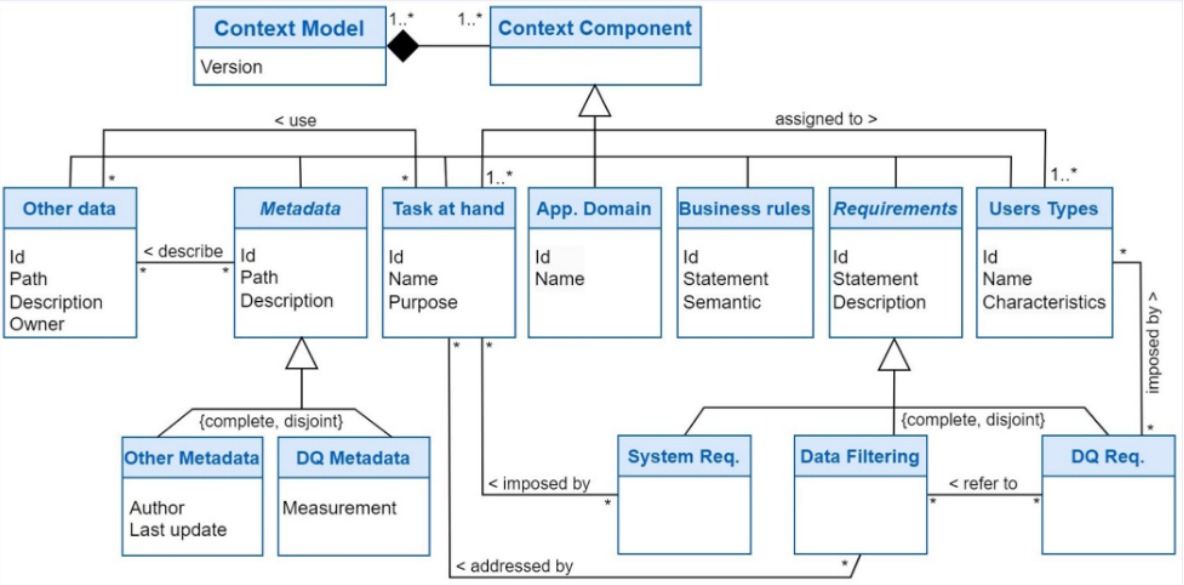

Figura 2.3: Metamodelo de Contexto extraído de [29].

2.5. Metamodelo de CD sensible al Contexto

Una de las principales contribuciones de la tesis de doctorado de [29] es la definición de un metamodelo de CD sensible al contexto *(Context-aware DQ Metamodel)*. Este metamodelo integra los conceptos del metamodelo de CD y el de contexto, estableciendo las relaciones entre ellos, lo que permite la definición de un modelo de CD adaptado a un escenario de uso específico.

Este metamodelo, ilustrado en la Figura 2.4, no solo muestra las relaciones entre los elementos de los metamodelos principales, sino que también incluye clases auxiliares que son cruciales para su interacción. Estas clases adicionales son:

- ***Data at hand***: Es el conjunto de datos específico cuya calidad se evalúa. Ambos modelos (de CD y del contexto) se relacionan a partir de ella.
- ***Data***: Esta clase representa el *Data at hand* y cualquier otro conjunto de datos que se utilice para contextualizarlo.
- ***Data schema***: Representa la estructura de los datos. Se relaciona con la clase *Data*, y es crucial para la definición de las *DQ metrics*, ya que los *DQ methods* que implementan las métricas, se aplican sobre los atributos del esquema de datos.
- ***Users***: Son los individuos o grupos que interactúan con los datos. Se agrupan según sus tipos y roles, imponiendo requisitos que influyen en el modelo de CD.
- ***Execution results***: Contienen el valor de calidad de los datos, así como la fecha y la hora en que se realizó la medición. Estos resultados pueden usarse posteriormente como metadatos de CD o para analizar la evolución de la misma.

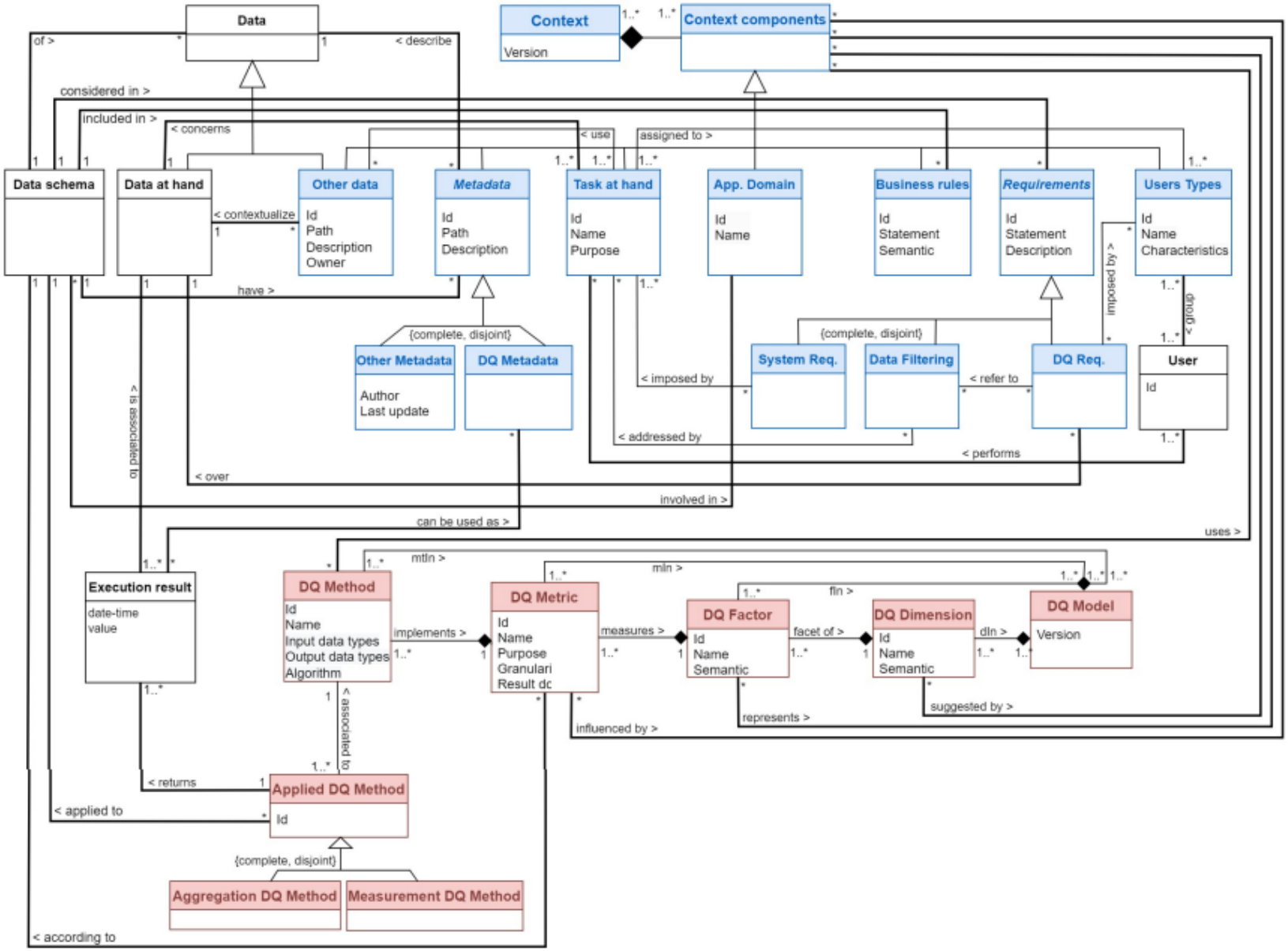

Figura 2.4: Metamodelo de CD sensible al Contexto *(Context-aware DQ Metamodel)* extraído de [29].

2.6. Metodología CaDQM

La metodología *Context-aware Data Quality Management* (CaDQM), propuesta por [29] en su tesis doctoral, constituye un marco estructurado y sistemático para la gestión de la CD. CaDQM identifica y considera el contexto como elemento clave en todas sus fases. Este enfoque reconoce que la CD no debe evaluarse de manera aislada, sino que debe adaptarse a las particularidades del entorno, usuarios y procesos en los que dichos datos intervienen. CaDQM define el contexto de los datos en base al metamodelo presentado previamente en la Sección 2.4.

Según su definición, CaDQM se compone de una visión estática y una visión dinámica. La visión estática define los elementos de la metodología, como las fases, actividades, roles y artefactos. Por su parte, la visión dinámica describe un ciclo de vida que guía el proceso de gestión de CD, estableciendo la secuencia en la que se deben ejecutar las fases. Estas fases son:

- **Fase 1: Planificación de Calidad de Datos *(DQ Planning)***
- **Fase 2: Evaluación de Calidad de Datos *(DQ Assessment)***
- **Fase 3: Mejora de Calidad de Datos *(DQ Improvement)***

A continuación se describen cada una de estas fases, indicando el alcance del presente trabajo.

2.6.1. Fase 1: Planificación de Calidad de Datos *(DQ Planning)*

La Fase 1 se encarga de describir y modelar el contexto en el que los datos serán utilizados. Su principal objetivo es definir un modelo de contexto que sirva como base para las fases siguientes. La planificación se organiza en las siguientes etapas:

- **ST1 - Elicitación *(Elicitation)***: En esta etapa, se selecciona el *data at hand* (conjunto de datos a evaluar) y se recopila información de la organización relacionada con él. La actividad principal es la definición de los componentes del contexto, en base al metamodelo de contexto (Figura 2.3). Esto incluye la identificación de dominios de aplicación, reglas de negocio, perfiles de usuarios y la documentación de problemas de CD ya reportados.
- **ST2 - Análisis de Datos *(Data Analysis)***: La actividad principal de esta etapa es el perfilado de datos *(data profiling)*. Se analizan las características y relaciones del *data at hand* para descubrir patrones y anomalías. Los resultados de este análisis pueden revelar nuevos requerimientos de CD, restricciones o reglas de negocio. Además, a partir de los resultados del perfilado se pueden identificar diversos problemas de CD potenciales, tales como datos incompletos, duplicados o inconsistentes. En este etapa también es posible obtener una estimación de CD preliminar, que puede ser un indicador del volumen de trabajo para evaluar y mejorar la CD. Al finalizar, el modelo de contexto se puede actualizar con los nuevos elementos descubiertos.
- **ST3 - Análisis de Requerimientos de Usuario *(User Requirements Analysis)***: Esta etapa se enfoca en la interacción con los usuarios y otras partes interesadas a través de entrevistas, encuestas o reuniones. El objetivo es identificar problemas de CD que solo son conocidos por los usuarios y entender sus expectativas. Al igual que en la etapa anterior, la información recopilada permite actualizar y refinar el modelo de contexto.

La salida de esta fase es un modelo de contexto formalizado y una lista de problemas de CD identificados, vinculados al *data at hand*, que servirán como insumos para la siguiente fase.

2.6.2. Fase 2: Evaluación de Calidad de Datos *(DQ Assessment)*

Esta fase tiene como objetivo medir y evaluar la calidad de los datos seleccionados *(data at hand)*, a partir de la construcción de un modelo de CD *(DQ Model)*. Para lograrlo, utiliza el contexto previamente definido en la Fase 1. El proceso se organiza a través de las siguientes tres etapas que se ejecutan de manera secuencial:

- **ST4 - Definición del Modelo de CD *(DQ Model Definition)*:** En esta etapa, se define un modelo de CD sensible al contexto, basándose en el metamodelo de CaDQM (Figura 2.4), que integra los metamodelos de CD y de contexto. Antes de comenzar a definir los conceptos de CD, el proceso inicia con la priorización de los problemas de CD identificados en la Fase 1, que en conjunto a los componentes de contexto, guian la selección de las dimensiones y factores de CD. A continuación, se definen las métricas, teniendo en cuenta que un mismo factor de CD puede medirse de diversas maneras. Finalmente, para cada métrica, se especifica su implementación a través de un método de CD, el cual se materializa en algoritmos que a su vez pueden ser instanciados como métodos aplicados sobre los datos a evaluar *(data at hand)*.
- **ST5 - Medición de la CD *(DQ Measurement)*:** Una vez definido el modelo de CD, se procede a medir la CD mediante la ejecución de los métodos aplicados, implementados para cada una de las métricas definidas. Los resultados obtenidos se almacenan como metadatos de CD, manteniendo la trazabilidad de los conceptos de CD (dimensión, factor y métrica) asociados al método ejecutado. Para ello, esta etapa incluye el diseño de una base de datos específica y separada para almacenar estas mediciones, conocida como *DQ Metadata*, considerando en su diseño que también se guardarán en ella los resultados cualitativos de la evaluación posterior.
- **ST6 - Evaluación de la CD *(DQ Assessment)*:** En esta etapa, se evalúan los valores de CD obtenidos durante la medición. La evaluación consiste en comparar los valores cuantitativos medidos (*DQ values*) con valores de referencia para emitir un diagnóstico de CD. Para ello, se definen enfoques de evaluación que especifican umbrales (*thresholds*) para clasificar los valores en categorías cualitativas como "baja", "aceptable" o "excelente", por ejemplo. Generalmente estos umbrales se derivan de componentes de contexto, como los requerimientos de CD (por ejemplo, exigir que al menos el 90 % de los correos electrónicos no sean nulos). Otro componente clave es el tipo de usuario, que puede generar diferentes perfiles y, por ende, distintos enfoques de evaluación, lo que refleja el carácter subjetivo de esta actividad. Finalmente, al comparar los valores de CD medidos con los umbrales especificados, se obtienen los resultados cualitativos que se almacenan en la base de datos de metadatos de CD definida en la etapa previa.

2.6.3. Fase 3: Mejora de Calidad de Datos (*DQ Improvement*)

La Fase 3 se enfoca en analizar las causas de los problemas de CD detectados, para luego definir y ejecutar un plan estratégico de mejora. Las etapas que la componen son:

- **ST7 - Análisis de Causas de los Problemas de CD (*DQ Problems Causes Analysis*)**: En esta etapa se analizan los resultados de la evaluación para identificar las causas raíz de los problemas de CD, las cuales luego se priorizan. Este análisis es fundamental, ya que a veces es más importante resolver la causa (p. ej., falta de capacitación de un usuario) que el problema superficial (p. ej., datos desactualizados).
- **ST8 - Definición del Plan de Mejora (*DQ Improvement Plan Definition*)**: En esta etapa se diseña un plan para resolver los problemas de CD priorizados. Se analizan diversas técnicas y estrategias de mejora, considerando los costos asociados y el contexto de los datos (p. ej., el tipo de usuario o las reglas de negocio). El objetivo es seleccionar las soluciones más adecuadas y equilibradas para el plan.
- **ST9 - Ejecución del Plan de Mejora (*DQ Improvement Plan Execution*)**: Esta es la etapa final, donde se ejecutan todas las técnicas y estrategias definidas en el plan de mejora. Una vez finalizada la ejecución, el proceso de gestión de la CD puede finalizar, regresar a la fase de evaluación para verificar los resultados, o entrar en un monitoreo periódico que reinicia el ciclo continuamente.

2.6.4. Flujo de las Fases de CaDQM

Tras presentar las diferentes fases de CaDQM con sus correspondientes etapas y actividades (vista estática), en este apartado, se describe el ciclo de vida del proceso de gestión (vista dinámica). Esta vista define el orden en que las etapas de cada fase deben ser ejecutadas y las transiciones posibles entre ellas. El flujo se ilustra en la Figura 2.5, y a continuación se profundiza en sus principales detalles.

Flujo de las Fases

- **Fase 1 - *DQ Planning***: CaDQM comienza siempre con la etapa de Elicitación (ST1) de la fase de Planificación. Luego, las etapas de Análisis de Datos (ST2) y Análisis de Requerimientos de Usuario (ST3) pueden ejecutarse en cualquier orden, incluso en paralelo. Una vez finalizada la Fase 1 - *DQ Planning*, se puede pasar a la Fase 2 - *DQ Assessment* tanto desde ST2 como desde ST3.
- **Fase 2 - *DQ Assessment***: Esta fase tiene un flujo secuencial: ST4 (Definición del Modelo de CD), seguido de ST5 (Medición de CD) y ST6 (Evaluación de CD). Al finalizar la evaluación, el proceso puede tomar diferentes caminos. Podría continuar el flujo secuencial, pasando a la Fase 3, a menos que el objetivo fuese solo evaluar la CD, es posible finalizar acá el proceso de gestión, sin realizar mejoras.
- **Fase 3 - *DQ Improvement***: La fase de mejora comienza con el Análisis de Causas de los Problemas de CD (ST7). El proceso de gestión puede finalizar si se decide no trabajar en la eliminación de las causas de los problemas. Si se decide continuar, se sigue con la Definición del Plan de Mejora (ST8) y su posterior Ejecución (ST9), completando el ciclo.

Ciclo Continuo de CaDQM

Si bien la Fase 3 - *DQ Improvement* aparece como la última de CaDQM, no implica que la gestión de la CD necesariamente deba finalizar con la misma. Una vez que se ejecuta el plan de mejora y se completa la Fase 3, se contemplan diferentes posibilidades:

- Finalización: El proceso puede terminar si se considera que los problemas de CD han sido resueltos.
- Verificación: Es posible regresar a la Fase 2 - *DQ Planning* para medir nuevamente la CD y verificar la efectividad de las mejoras realizadas.
- Monitoreo periódico: La opción más completa y recomendada es un monitoreo continuo, donde el ciclo completo a traves de todas las fases de CaDQM se ejecuta de forma periódica, permitiendo que la organización se adapte a los cambios en el contexto de los datos.

Fases abordadas

El presente trabajo se centra en la implementación de la Fase 2 - *DQ Assessment*. Como entrada, se requieren los insumos que provienen de la Fase 1 - *DQ Planning*: un modelo de contexto y una lista de problemas de CD identificados en dicha fase, considerados en la definición del modelo de CD. La Fase 3 - *DQ Improvement* se encuentra fuera del alcance de esta tesis, por lo que su implementación no ha sido abordada.

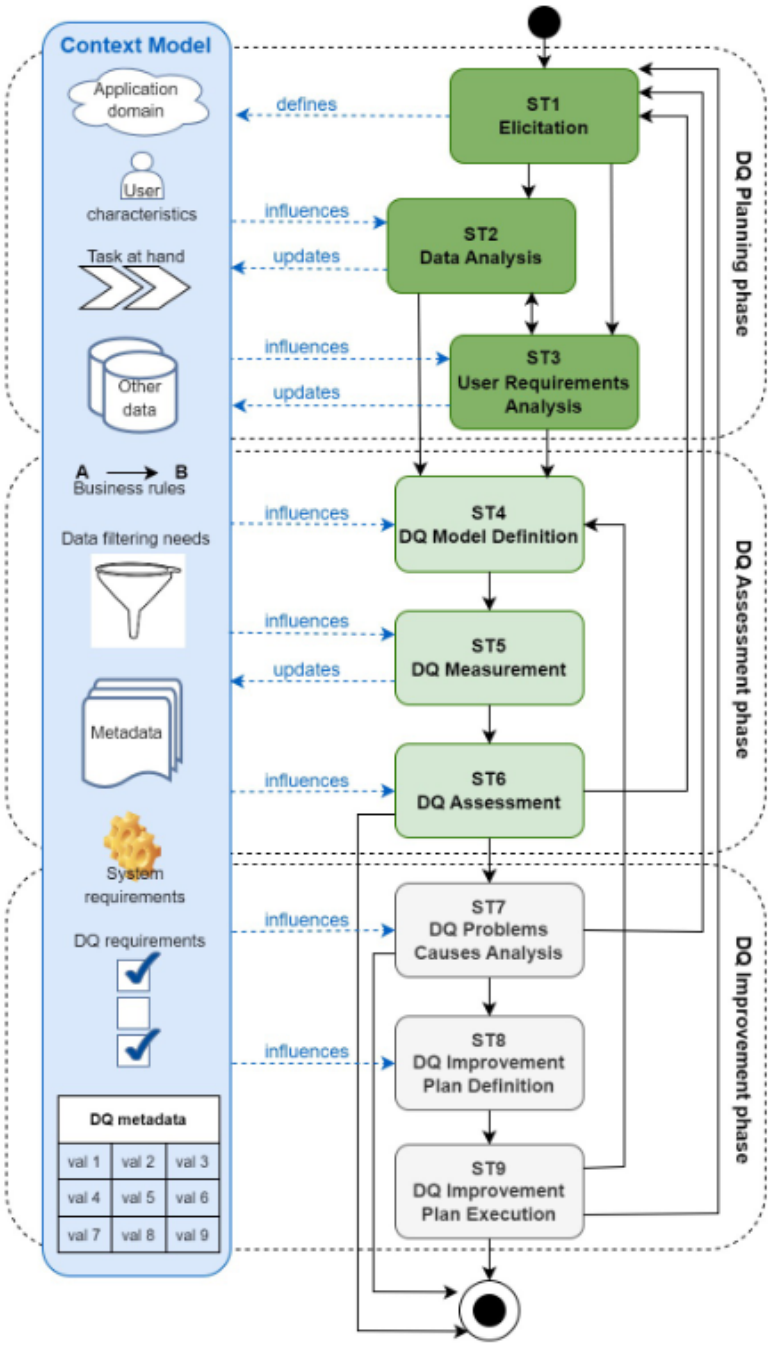

Figura 2.5: Proceso de gestión de CD aplicando las etapas de CaDQM, extraído de [29]
Las flechas azules punteadas representan relaciones entre las etapas y el modelo de contexto, y sus etiquetas distinguen si el contexto es definido, actualizado o si simplemente influye en las actividades de la etapa.

Capítulo 3

Análisis y Relevamiento de Requerimientos

El presente trabajo se centra en la implementación de una herramienta capaz de ejecutar la Fase 2 - *DQ Assessment* de CaDQM, presentada en la Sección 2.6.2 del marco teórico. En este capítulo, se especifican los requerimientos funcionales y no funcionales de la herramienta. Estos requisitos se definieron a partir de los objetivos del proyecto y de la definición de CaDQM, sentando las bases para su diseño y desarrollo.

3.1. Relevamiento y Análisis de Requerimientos

El objetivo general planteado por Serra y Marotta, consistía en diseñar y desarrollar una herramienta que dé soporte a los expertos en CD, en la aplicación de la metodología de gestión de CD, sensible al contexto (CaDQM), presentada previamente. Para esto, se plantearon dos proyectos de grado diferentes trabajando en paralelo: uno enfocado en la implementación de la Fase 1 - *DQ Planning* y otro, correspondiente a esta tesis, en la implementación de la Fase 2 - *DQ Assessment*.

Para asegurar la coherencia entre ambas fases y facilitar su futura integración en una herramienta integral, los equipos trabajaron de manera coordinada, acordando principalmente dos requerimientos clave:

- **BD común:** Utilización de un Modelo Entidad-Relación (MER) unificado, basado en [29] para definir una BD común en la implementación de cada proyecto de grado. Este requerimiento es fundamental para asegurar la interoperabilidad mediante uso de datos compartidos entre las herramientas de la Fase 1 - *DQ Planning* y de la Fase 2 - *DQ Assessment*.
- **Estilo de Interfaz de Usuario consistente:** Establecimiento de un estilo visual uniforme entre los dos proyectos de grado para lograr un *"Look & Feel"* similar en ambas herramientas. Esto busca garantizar que las herramientas, a pesar de ser desarrolladas por equipos distintos, se perciban como parte de un sistema unificado, lo que facilitará la curva de aprendizaje del usuario y mejorará la experiencia de uso general.

3.1.1. Requerimientos Generales

A partir de los objetivos planteados y la definición de la Fase 2 de CaDQM propuesta por Serra, presentados en la Sección 2.6 del marco teórico, se llevó a cabo un relevamiento de requerimientos. De este análisis, se desprende como objetivo principal el desarrollo de una aplicación que permitiera y facilitara a los usuarios la definición de modelos de CD en base a un contexto, para la posterior medición y evaluación de la CD para un conjunto de datos relacionales asociados. Para lograrlo, la herramienta debe integrar los artefactos generados en la Fase 1, como los componentes de contexto y un conjunto de problemas de CD.

Dada a la naturaleza secuencial y ordenada de las actividades y tareas necesarias para la ejecución de las etapas de dicha fase, especialmente, la construcción del modelo de CD, se identificó la necesidad de un flujo de trabajo que guíe al usuario paso a paso, asegurando que no se omitan pasos clave. A lo largo de este proceso, el sistema debe ofrecer distintas funcionalidades de apoyo, como la posibilidad de consultar los problemas de CD y componentes del contexto como referencia, promoviendo decisiones informadas y alineadas con las necesidades reales relacionadas con el uso de los datos. Además, se espera que el sistema integre mecanismos de sugerencia automática que apoyen la construcción de métodos de CD mediante inteligencia artificial.

3.1.2. Requerimientos Funcionales Principales

Con el fin de implementar la herramienta que ejecute de manera completa la Fase 2 - *DQ Assessment* de CaDQM, utilizando como entrada los datos provenientes de la Fase 1 - *DQ Planning*, se identificaron los siguientes requerimientos funcionales del sistema:

- **Priorización de Problemas de CD:**
  - Permitir al usuario identificar y priorizar problemas de CD relevantes en función del dominio de aplicación y los objetivos del proyecto.
- **Definición del Modelo de CD:**
  - Permitir la construcción jerárquica del modelo de CD, estableciendo la relación entre sus elementos: dimensiones, factores, métricas y métodos de CD, asegurando la trazabilidad entre ellos (un factor se asocia a una dimensión, una métrica a un factor y un método a una métrica de CD).
  - Permitir la asociación de cada concepto de CD del modelo de CD (dimensiones, factores, métricas y métodos de CD) con componentes de contexto relevantes que justifiquen su inclusión. A su vez, en conjunto, los problemas de CD pueden sugerir dimensiones y factores, registrando también esta asociación.
  - Permitir la edición y eliminación de artefactos del modelo de CD definidos por el usuario.
  - Definición de Métodos de CD:
    - Permitir la creación de métodos de CD asociados a métricas de CD, implementados a través de algoritmos *SQL*.
    - Permitir la instanciación de los métodos de CD (de medición y agregación, (Figura 2.2) aplicados a elementos específicos de bases de datos relacionales (tablas, columnas, filas, etc.) del *data at hand* (conjunto de datos cuya calidad es evaluada), reutilizando los algoritmos genéricos definidos y adaptándolos al esquema relacional concreto.
- **Ejecución de Mediciones de CD y Almacenamiento de los Resultados:**
  - Ejecutar los métodos de CD aplicados definidos en el modelo de CD sobre el *data at hand*, para obtener las mediciones de CD correspondientes.
  - Almacenar los resultados de CD obtenidos en una BD relacional dedicada a los metadatos de CD, manteniendo la trazabilidad con los conceptos de CD del modelo de CD.
- **Evaluación de la CD:**
  - Evaluar los resultados de las mediciones frente a umbrales de CD definidos, permitiendo una interpretación cualitativa de la CD (ej. “baja”, “aceptable”, “excelente”).

- **Visualización de Resultados de CD:**
  - Proporcionar interfaces para la visualización clara y comprensible tanto de los resultados numéricos de las mediciones como de los resultados cualitativos de las evaluaciones de CD.
  - Permitir al usuario explorar los datos de CD a distintos niveles de granularidad (a nivel de tabla, columna o tupla/fila) y filtrar por métodos aplicados, manteniendo la trazabilidad con los conceptos de la jerarquía del modelo de CD asociado.
- **Asistencia mediante Inteligencia Artificial (IA):**
  - Sugerir conceptos del modelo de CD en función de los problemas de CD priorizados y los componentes del contexto.
  - Asistir al usuario en la definición de métodos de CD, completando automáticamente todos sus campos a partir de la información de la métrica de CD asociada, con especial énfasis en la generación de su algoritmo.

3.1.3. Expectativas del Sistema

Se espera que la aplicación proporcione una experiencia de usuario clara e intuitiva, facilitando la definición del modelo de CD y los posteriores procesos de ejecución de la medición y evaluación de la CD. El sistema deberá ofrecer soporte a usuarios con distintos niveles conocimiento en CD.

La asistencia mediante IA se integrará en el flujo de trabajo como un recurso de apoyo explícito y valioso. Esta funcionalidad podrá ajustarse para personalizar recomendaciones y sugerencias basadas en la preferencia del usuario. Esto incluye, específicamente, la recomendación de dimensiones y factores del modelo de CD, así como la generación de algoritmos para métodos. En todo momento, esta asistencia no comprometerá la posibilidad de edición o intervención manual por parte del usuario experto.

El diseño del sistema priorizará la trazabilidad completa de los elementos de CD y contexto utilizados a lo largo del proceso. Esto significa que la herramienta debe ser capaz de mostrar, de manera clara y respetando la jerarquía, la relación entre los componentes del contexto que justifican la definición del modelo de CD (también los problemas de CD cuando corresponda). Dado esto, al finalizar la construcción del modelo de CD, se debe permitir la generación de un reporte descargable que dicha jerarquía de los conceptos de CD y los distintos artefactos relacionados en su definición. Además, la aplicación debe ofrecer la capacidad de visualizar los resultados desde diferentes perspectivas, permitiendo al usuario filtrar por tablas, atributos o elementos del modelo de CD.

Como resultado final, se espera que la herramienta de la Fase 2 - *DQ Assessment* pueda integrarse con la de la Fase 1 - *DQ Planning* a través de una BD relacional compartida. En esta BD, la herramienta de la Fase 1 almacenará los resultados obtenidos de su aplicación. Desde la herramienta de la Fase 2, se podrá acceder a esta BD, considerando esta información de los datos provenientes de la fase previa, para la construcción del modelo CD y la evaluación de la CD sobre un *data at hand*, que también será común para ambas herramientas.

Capítulo 4

Diseño

En este capítulo se detalla el diseño de la solución, abordando los diversos aspectos fundamentales que definen su implementación. Se presenta la arquitectura general de la aplicación, el modelado de la base de datos para gestionar los elementos del modelo de CD y su interacción con el modelo de contexto, la capa de integración mediante *APIs REST* y el diseño de la interfaz de usuario, que guía al usuario a través del proceso de ejecución de la Fase 2 - *DQ Planning* de CaDQM.

4.1. Arquitectura de la Aplicación

Con el objetivo de desarrollar una herramienta accesible, modular y escalable, se optó por implementar una aplicación web con una arquitectura de tipo cliente-servidor. Esta decisión permite desacoplar el *frontend* del *backend*, facilitar el mantenimiento y habilitar una futura evolución independiente de cada componente. La comunicación entre ambas capas se plantea mediante una *API REST*, favoreciendo la interoperabilidad y la integración con otros servicios o herramientas en el futuro. La Figura 4.1 presenta un diagrama de como se organiza dicha arquitectura.

Tecnologías Analizadas

En función de esta arquitectura, se analizaron diversas tecnologías para cada uno de los componentes principales del sistema. Para el desarrollo del *frontend*, se consideraron los *frameworks*: *Angular* [32] y *React* [22], tomando en cuenta criterios como robustez, modelo de desarrollo, comunidad y experiencia previa del equipo. Para el *backend*, se consideraron los *frameworks* de desarrollo web: *Django* [10] y *Flask* [25], de *Python* [28], y se consideró también de *Node.js* [23] usando *Express.js* [8], priorizando la rapidez de desarrollo, el soporte para creación de *APIs* y la disponibilidad de librerías. En cuanto a la base de datos, se consideraron *PostgreSQL* [11] y *MySQL* [24], atendiendo a la necesidad de utilizar un modelo de datos relacional.

Tecnologías Seleccionadas

Tras analizar las tecnologías mencionadas, se seleccionaron las siguientes:

- ***Frontend***: Se optó por *Angular*. Su amplio ecosistema, que incluye una robusta interfaz de línea de comandos (*CLI*) y un sistema integrado para el manejo de rutas y formularios, permite un desarrollo eficiente y organizado, ideal para aplicaciones web.
- ***Backend***: Se seleccionó el *framework Django* de *Python*, junto con *Django REST Framework* [7]. La decisión se basó en la madurez y rapidez de desarrollo que ofrece, la familiaridad del equipo con el lenguaje y su amplio ecosistema de bibliotecas para análisis de datos e inteligencia artificial, lo que facilita la integración de funcionalidades como el módulo de generación de recomendaciones.
- **Base de datos**: Se eligió *PostgreSQL* por ser una opción popular y confiable entre las bases de datos relacionales, con amplio soporte y comunidad activa. Además, su robusta compatibilidad

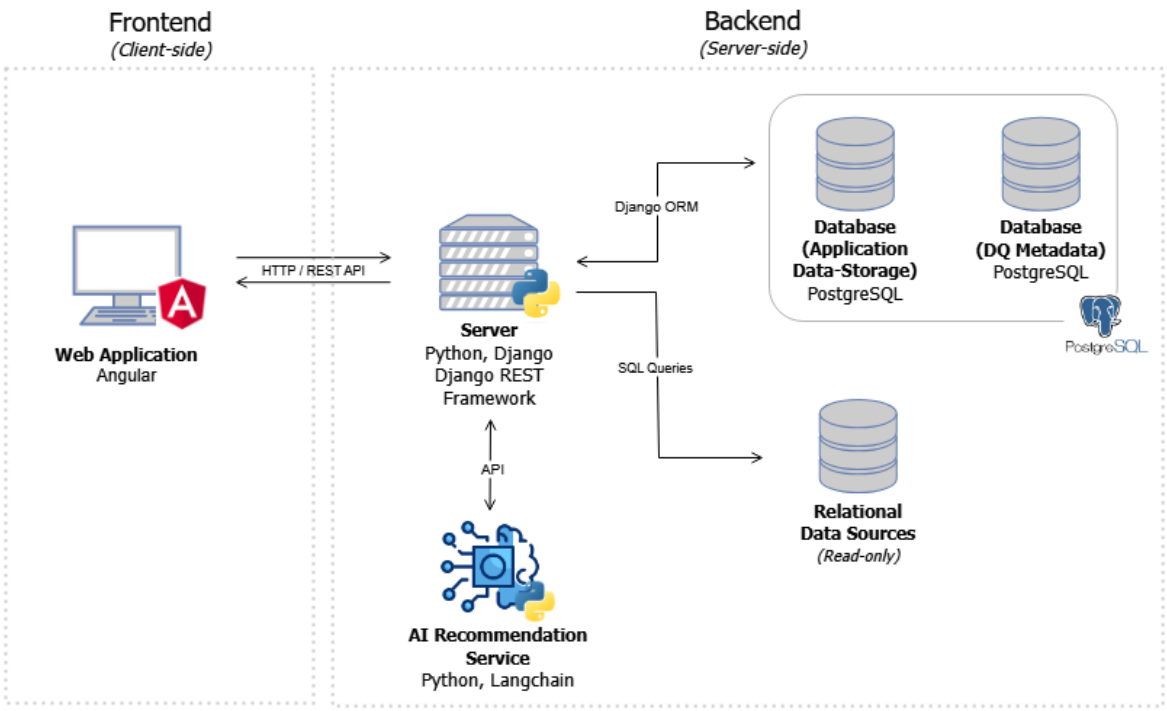

Figura 4.1: Arquitectura de la herramienta de la Fase 2 - *DQ Assessment* de CaDQM.

nativa con el *framework Django* en el *backend* consolidó su elección. Esta decisión también se tomó en consenso con el equipo del otro proyecto de grado, encargado de trabajar en la Fase 1 - *DQ Planning*, previendo facilitar la interoperabilidad entre ambas herramientas.

Esta arquitectura modular favorece la escalabilidad, facilita el mantenimiento y permite integrar nuevas funcionalidades sin afectar la estructura general del sistema. Esto incluye, por ejemplo, la incorporación de módulos de IA mediante la creación de nuevos *endpoints* para la *API*.

4.2. Diseño del Modelo de Datos

El modelo de datos para la BD común se diseñó tomando como referencia los metamodelos de CD y de contexto definidos por CaDQM. Esto permitió representar los artefactos involucrados en las Fases 1 y 2 de CaDQM de forma modular, reutilizable y adaptable a distintos proyectos.

Este diseño surgió de un esfuerzo colaborativo con el otro proyecto de grado, responsable de la Fase 1 - *DQ Planning*. Como parte de este proceso, se dividieron las responsabilidades de la siguiente manera:

- La definición del modelo de contexto estuvo a cargo del equipo del proyecto de grado responsable de la Fase 1.
- El diseño del modelo de CD, junto con los resultados de medición y evaluación, correspondió exclusivamente a este trabajo, como parte de la Fase 2 de CaDQM.

Un concepto clave representado mediante la entidad `Project`, fue concebido en este esfuerzo colaborativo como un nexo clave para establecer la integración entre ambos modelos (CD y Contexto). Su inclusión permitió a cada equipo trabajar de manera independiente en la definición de sus respectivos modelos, asegurando al mismo tiempo que los elementos de una fase pudieran relacionarse con los de la otra, cumpliendo así con los requisitos de la metodología CaDQM.

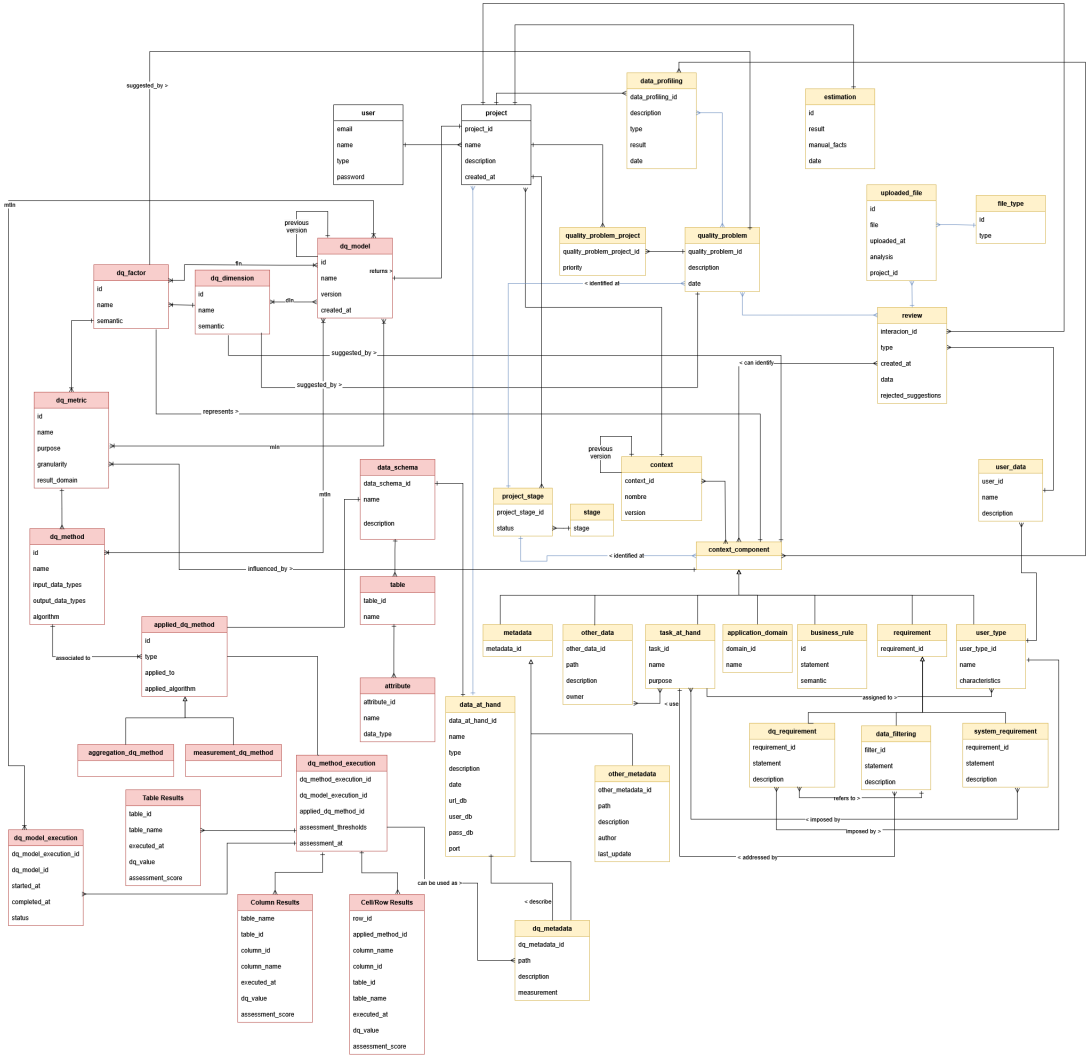

Figura 4.2: Diagrama del Modelo Entidad-Relación (MER) del sistema. Las entidades en amarillo corresponden al diseño de la Fase 1 (modelo de contexto), las entidades en rojo al diseño de la Fase 2 (modelo de CD), y las entidades en blanco a los elementos compartidos que garantizan su interoperabilidad.

Como parte exclusiva del diseño de la Fase 2 - *DQ Assessment*, para el almacenamiento de los resultados de CD, se diseñó otra BD (`DQMetadata`), encargada de gestionar las ejecuciones de de las mediciones y evaluaciones de CD en la Fase 2, mediante `DQModelExecution`, almacenando los resultados de forma organizada según diferentes granularidades.

La Figura 4.2 presenta el modelo entidad-relación (MER) final, del sistema, en el cual las entidades clave se diferencian por color: rojo para el modelo de CD, amarillo para el modelo de contexto, y blanco para los elementos que garantizan la interoperabilidad. En las siguientes subsecciones se examinará con mayor detalle el diseño de estas entidades.

4.2.1. Gestión y Versionado del Modelo de CD

La entidad `DQModel` representa el concepto central del diseño para la Fase 2 de CaDQM, encapsulando una versión concreta de un modelo de CD para un proyecto particular.

Cada modelo de CD cuenta con un identificador, nombre, versión, estado (`draft` o `finished`), fecha de creación y de finalización, y versiones previas, lo cual permite mantener una trazabilidad entre versiones sucesivas. Además, un `DQModel` debe comenzar en estado `draft` y, una vez completa su definición, pasa a estado `finished`, impidiendo futuras modificaciones sobre el mismo.

Si bien, por si sola, la definición de la entidad es simple, su rol es fundamental ya que, además de gestiona su ciclo de vida y habilitar el versionado, cada instancia de `DQModel` sirve como un punto de anclaje al cual se asocian todos los conceptos de CD (dimensiones, factores, métricas, métodos y métodos aplicados) que definen el modelo.

4.2.2. Conceptos de CD Reutilizables

Una decisión clave para el diseño del modelo de CD consistió en representar los conceptos de CD (dimensiones, factores, métricas y métodos) como artefactos reutilizables. Esto se logró mediante un conjunto de entidades base (`DQDimensionBase`, `DQFactorBase`, `DQMetricBase` y `DQMethodBase`), las cuales almacenan las definiciones generales de dichos conceptos y mantienen las siguientes relaciones jerárquicas:

- Un `DQFactorBase` está asociado a una `DQDimensionBase` mediante la relación `facetOf`.
- Una `DQMetricBase` está relacionada a un `DQFactorBase` mediante `measures`.
- Un `DQMethodBase` puede implementar una `DQMetricBase` mediante `implements`.

Esta separación permite mantener un repositorio común de conceptos de CD predefinidos, que puede ser enriquecido y reutilizado por múltiples modelos de CD. Esto no solo evita la replicación de información, sino que también facilita y agiliza la construcción de nuevos modelos de CD. Además, esta estructura respeta la jerarquía entre los conceptos de CD presentada en el marco teórico (Figura 2.1).

4.2.3. Modelos de CD: Asociación de Conceptos de CD y Contexto

La estructura de un modelo de CD se define a través de un conjunto de entidades jerárquicamente relacionadas. Estas entidades de asociación se encargan de vincular las definiciones de los conceptos de CD, almacenados como artefactos reutilizables, a una versión específica de un modelo de CD (instancia de un `DQModel`). Estas entidades son:

- `DQModelDimension`: Representa la inclusión de una dimensión de CD al modelo de CD. Vincula una `DQDimensionBase` a una versión específica de un modelo de CD (`DQModel`), sin necesidad de crear una nueva definición.
- `DQModelFactor`: Incorpora un factor de CD al modelo de CD, asociando un `DQFactorBase` que por definición, ya se encuentra asociada a una dimensión base. Esta instancia se vincula explícitamente a la `DQModelDimension` correspondiente, preservando así la estructura jerárquica.

- `DQModelMetric`: Agrega una métrica al modelo de CD, estableciendo un vínculo con una `DQMetricBase` previamente asociada en su definición a un factor base. Esta métrica se asocia de forma explícita con el `DQModelFactor` correspondiente, manteniendo la coherencia jerárquica.
- `DQModelMethod`: Define un método para el modelo de CD, través de su asociación con un `DQMethodBase`, que se encuentra ligado en su definición a una métrica base. Esta instancia se conecta directamente con la `DQModelMetric` determinada, respetando la estructura jerárquica.

Además, cada una de estas entidades de asociación mantiene un vínculo directo con la instancia específica del `DQModel`, como se especificó para la dimensión de CD.

Otra característica clave del enfoque de este diseño es el uso explícito de los componentes del contexto, que permiten justificar la inclusión de los conceptos de CD en función de las condiciones particulares del contexto de los datos, así como también si como de los problemas de CD. Estos se representan mediante el campo:

- `context_components`: Registra los componentes del contexto (por ejemplo, reglas de negocio, requerimientos de CD, tipos de usuario, entre otros) que sugieren la inclusión de cada concepto de CD. Este campo está presente en todos los conceptos de CD del modelo.
- `dq_problems`: Permite asociar problemas de CD particulares que se busca abordar con la inclusión del concepto de CD. Este campo se incluye solamente para las dimensiones y factores de CD.

Este enfoque garantiza una clara separación entre las definiciones generales de los conceptos de CD y su utilización en modelos de CD para contextos de datos específicos. La Figura 4.3 ilustra esta estructura jerárquica entre las entidades correspondientes.

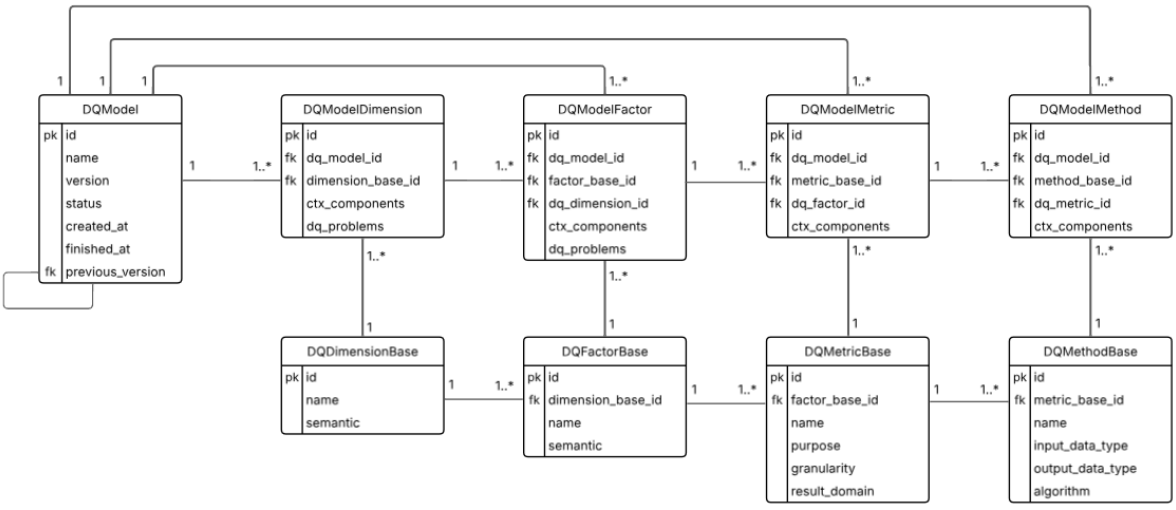

Figura 4.3: Diseño del esquema de base de datos para la definición del modelo de CD, mediante la asociación jerarquía de los conceptos de CD.

Métodos de CD Aplicados

Para completar la definición de un modelo de CD aplicados, se requiere también la inclusión de métodos aplicados. Estos, a diferencia de las entidades anteriores, no se modelan como conceptos de CD reutilizables, ya que representan la instanciación de un método de CD para un conjunto de datos fuente específico. Por ello, su propia definición implica la asociación directa a un método de CD (`DQModelMethod`) incluido en el modelo de CD, sin necesidad de vincularse explícitamente a un `DQModel` ni a sus componentes de contexto.

4.2.4. Gestión de *Project*

La entidad `Project` fue concebida como un elemento central para posibilitar la integración de los distintos elementos propuestos en el metamodelo de la metodología CaDQM, unificando los artefactos involucrados en la Fase 1 - *DQ Planning* y la Fase 2 - *DQ Assessment* en el diseño de la BD común para ambos proyectos de grado. Esta entidad resuelve el desafío de unificar el modelo de contexto y el modelo de CD sin introducir una dependencia directa entre ellos. De esta forma, actúa como un facilitador clave para la interoperabilidad de ambas herramientas, incluso cuando se desarrollan de forma independiente.

A continuación, se detallan los elementos que una instancia de `Project` gestiona de forma independiente:

- Una versión del modelo de CD, representada por una instancia de `DQModel`.
- Una versión del modelo de contexto, representada por una instancia de `ContextModel`.
- Un conjunto de datos de fuente, identificado como `data_at_hand`, sobre el cual se aplicará la evaluación de CD.

Este diseño implica que, si bien una misma versión de un contexto (instancia de `ContextModel`) puede ser utilizado para la definición de diferentes modelos de CD, la creación de un nuevo modelo de CD (instancia de `DQModel`) siempre requerirá la creación de un nuevo proyecto (instancia de `Project`) para encapsular dicha asocación, no permitiendo combinaciones repetidas del par (`ContextModel` version, `DQModel` version).

Para complementar la gestión de cada `Project`, la solución incorpora una entidad `ProjectStage`. Esta entidad mapea el ciclo de vida de cada instancia de `Project` a lo largo de las etapas de la metodología CaDQM y permite controlar el estado de avance de cada una de ellas mediante los estados `TO_DO`, `IN_PROGRESS` y `DONE`. Este mecanismo permite una visión integral del flujo de ejecución de las Fases 1 y 2 desde ambas herramientas. Cabe mencionar que, a diferencia de la Fase 1, donde es posible trabajar en múltiples etapas de manera simultanea, la Fase 2 impone un restricción de unicidad, permitiendo que solo una etapa se encuentre activa en un momento dado y siguiendo una secuencia de avance incremental entre las etapas.

De esta manera, la incorporación de `Project` cubre los siguientes aspectos clave:

- Vincular versiones específicas del modelo de contexto y del modelo de CD sin que exista una dependencia directa entre ellos.
- Actuar como nexo entre las responsabilidades de los distintos equipos, facilitando el trabajo colaborativo en un entorno desacoplado y coherente.
- Posibilitar un manejo iterativo y versionado de cada modelo, manteniendo un historial de evolución y adaptación a lo largo del tiempo.
- Ajustar los modelos a partir de nuevas condiciones del contexto de datos o problemas de CD identificados en pasos anteriores.

4.2.5. Ejecución del Modelo de CD y Metadatos de CD

Una vez definido el modelo de CD, las etapas posteriores de la Fase 2 de CaDQM se centran en medir y evaluar la CD de los *data at hand*. Este proceso se basa en la ejecución de los métodos de CD definidos en el modelo de CD, generando valores de CD cuantitativos (metadatos de CD).

Para representar cada instancia completa de medición, se introduce la entidad `DQModelExecution`, que encapsula una ejecución del modelo de CD. Cada ejecución incluye múltiples ejecuciones de métodos individuales, registradas mediante la entidad `DQMethodExecutionResult`, que almacena información como el método aplicado, la fecha de ejecución y los umbrales de CD definidos para la evaluación de la CD. Aunque la interfaz para esta funcionalidad no fue definida en el alcance final de este proyecto, con este diseño es posible de realizar múltiples ejecuciones sucesivas de un mismo modelo de CD, contemplando así la comparación de resultados o el análisis de variaciones bajo diferentes contextos de datos.

La aplicación de los métodos de CD produce como resultado distintos valores de CD que se almacenan de forma granular —a nivel de tabla, columna, fila o celda— a través de las siguientes entidades:

- `ExecutionTableResult`, para resultados a nivel de tabla.
- `ExecutionColumnResult`, para resultados a nivel de columna.
- `ExecutionRowResult`, para resultados a nivel de fila o celda.

Además, otra decisión clave del diseño, relacionada a la evaluación de la CD, es que cada método aplicado puede definir explícitamente sus propios umbrales de evaluación. Estos umbrales (*thresholds*) son almacenados en la misma entidad que contiene los resultados (`DQMethodExecutionResult`) y se utilizan para llevar a cabo el proceso de evaluación de CD en la etapa final de la Fase 2. En dicho proceso, el valor de CD obtenido se compara contra los rangos definidos, permitiendo asignar una `assessment_score`. Este valor cualitativo refleja el grado de cumplimiento esperado (por ejemplo, "Aceptable", "Moderado", "Crítico"), y se almacena en la misma entidad donde se almacena el resultado de la medición, de acuerdo con la granularidad correspondiente.

La decisión de almacenar tanto los resultados de CD (valores cuantitativos) como sus evaluaciones (valores cualitativos) únicamente en los niveles granulares y no en la entidad de ejecución del método de CD, responde a un criterio de organización orientado a la estructura propia de los datos. Esto permite acceder, consultar y analizar los resultados de CD de forma alineada con la arquitectura de los datos evaluados, evitando redundancias y asegurando una trazabilidad precisa.

Finalmente, dado que estos resultados de CD representan un insumo clave para el diagnóstico y evolución del sistema, se define una BD separada a la de la aplicación, denominada `DQ Metadata`, dedicada exclusivamente al almacenamiento de resultados de CD. Esta separación no solo centraliza el acceso a la información, sino que también habilita el uso de estos metadatos de CD, para una eventual actualización de la Fase1 para el componente de contexto `DQMetadata` del modelo de contexto.

4.3. Diseño de la Capa de Integración

Como parte de la arquitectura cliente-servidor adoptada, se diseñó una capa de integración basada en una *API REST*, encargada de exponer los datos y funcionalidades del sistema mediante una interfaz estandarizada. Esta capa tiene como objetivo desacoplar el *frontend* del *backend*, permitiendo una comunicación clara, extensible y escalable entre ambos módulos.

Para ello, el *backend* fue diseñado con una estructura modular, organizada en distintos artefactos funcionales que corresponden a áreas específicas del dominio de la aplicación. Esta organización se alinea con la arquitectura del *framework Django* [10], que permite agrupar funcionalidades en grandes módulos denominados `apps`, respondiendo a principios de separación de responsabilidades y favoreciendo la manutención del sistema.

Más específicamente, se definieron los siguientes dos módulos funcionales clave:

- `projects`, encargado de gestionar la organización general de los proyectos y sus elementos asociados.
- `dqmodels`, que contiene la lógica relacionada al modelo de CD y sus ejecuciones.

Cada uno de estos módulos define un conjunto de recursos que se exponen de manera uniforme a través de una interfaz *RESTful*, siguiendo principios de coherencia, reutilización y bajo acoplamiento. Esta interfaz fue diseñada para facilitar el consumo desde el *frontend*, y contempla tanto operaciones de consulta como de modificación. Esta decisión de diseño permite una evolución flexible del sistema, habilitando la incorporación de nuevas funcionalidades sin afectar la estructura general ni la comunicación entre módulos.

4.4. Diseño de la Interfaz de Usuario de la Aplicación

Esta sección describe el diseño general de la interfaz de usuario y decisiones tomadas que aplican para todas las pantallas de esta interfaz.

4.4.1. Enfoque de Navegación

La interfaz se diseñó empleando un enfoque secuencial tipo *Wizard*, el cual surge de una alineación identificada con el proceso de trabajo de CaDQM, dado el flujo de ejecución de sus etapas. Esta estructura guía al usuario paso a paso a lo largo de las actividades que componen y definen cada una de las etapas de la Fase 2 - *DQ Assessment*. Cada actividad se mapea con su propia pantalla, la cual incluye elementos como:

- Indicadores de progreso.
- Botones de navegación, habilitados únicamente tras completar la actividad actual.
- Accesos directos al contexto de los datos y a los problemas de CD identificados.

Con esto, se busca un uso disciplinado de la herramienta, de manera alineada con conceptos clave de CaDQM.

4.4.2. Aspectos Transversales del Diseño de la Interfaz

El diseño de la interfaz gráfica se fundamenta en principios de claridad, navegación intuitiva y consistencia visual. Independientemente de la sección en la que se encuentre el usuario, la aplicación mantiene ciertos patrones visuales y de interacción que buscan facilitar el uso progresivo de la herramienta, acompañando el flujo de CaDQM.

- **Mensajes informativos e instrucciones de uso**: Cada bloque de trabajo incluye instrucciones al usuario, las cuales son presentadas al inicio, de forma clara y destacada. Estos mensajes orientan sobre qué acciones debe realizar el usuario, en qué orden y con qué propósito. La presencia de mensajes informativos, advertencias y confirmaciones forma parte del diseño global para asegurar la comprensión del proceso.
- **Consistencia en la interacción**: Se utilizan elementos de interfaz reutilizables que mantienen el mismo comportamiento visual y funcional en toda la aplicación, como botones de acción, modales para confirmación o visualización y tarjetas informativas.

Capítulo 5

Implementación

En este capítulo se explica la implementación de la herramienta. A través de un recorrido por el flujo funcional de la aplicación web, se presentan las interfaces diseñadas para las diferentes actividades necesarias para la ejecución de cada etapa de la Fase 2 - *DQ Assessment* de CaDQM, destacando sus funcionalidades clave y los criterios adoptados en su desarrollo.

5.1. Tecnologías Utilizadas

La herramienta se implementó como una aplicación web *full-stack*. La interfaz de usuario se desarrolló con *Angular*, mientras que el *backend* se construyó con *Python* y el *framework Django* (utilizando su extensión *Django REST Framework* para la *API* de la aplicación). Para la persistencia de datos, el sistema utiliza dos BD relacionales gestionadas por *PostgreSQL*: una para los datos de la aplicación y otra para los metadatos de CD, siendo estos los resultados de las mediciones y evaluaciones de la CD.

El código fuente se encuentra alojado en dos repositorios de *GitHub*: uno para el cliente *frontend* [27] y otro para el *backend* [26]. En el Anexo B se presenta una documentación técnica detallada para ambas partes.

5.2. Aplicación Web

La aplicación web implementada permite ejecutar todas las etapas de la Fase 2 de manera secuencial y progresiva, usando un contexto generado durante la ejecución completa de la Fase 1, a través de otra herramienta desarrollada por otro proyecto de grado. Interoperabilidad entre herramientas posible a través de una base de datos común. Para gestionar este flujo de trabajo y la relación entre ambas fases, se definió un concepto central llamado `Project`. Esta entidad actúa como un espacio de trabajo único que une y organiza los elementos clave para la ejecución de las fases y respectivas etapas de CaDQM: una versión del modelo de contexto, una versión del modelo de CD y el *data at hand*.

Para la gestión y el seguimiento de los proyectos durante el proceso de gestión de CD, se utilizaron dos conceptos clave: Etapa (`Stage`) y Estado (`Status`). La ejecución de cada etapa cuenta con un estado propio, que puede ser `TO_DO`, `IN_PROGRESS`, o `DONE`. Estos estados representan una secuencia progresiva en el flujo de trabajo en la aplicación de CaDQM. Cada proyecto lleva el registro de todas sus etapas, desde `ST1` a `ST6`, mapeando cada una a su estado correspondiente. Adicionalmente, se considera una variable `current_stage` que indica la etapa (única) en la que el proyecto se encuentra activo.

A continuación, se detalla la implementación de cada una de las interfaces y sus funcionalidades clave, siguiendo el flujo funcional de la aplicación tal como se presenta al usuario.

5.2.1. Punto de Partida (Nexo entre Fase 1 y Fase 2)

Esta interfaz (Figura 5.1) constituye el punto de entrada funcional de la herramienta, sirviendo como un punto de transición entre la Fase 1 - *DQ Planning* y la Fase 2 - *DQ Assessment* de la metodología CaDQM. En esta vista se presentan todos los proyectos existentes en el sistema, permitiendo iniciar la ejecución de la Fase 2 sobre el proyecto seleccionado. Cada proyecto disponible posee las etapas `ST1`, `ST2` y `ST3` en estado `DONE`, lo que indica que el modelo de contexto y los problemas de CD ya fueron definidos durante la Fase 1, encontrándose disponibles para la ejecución de la fase siguiente de CaDQM.

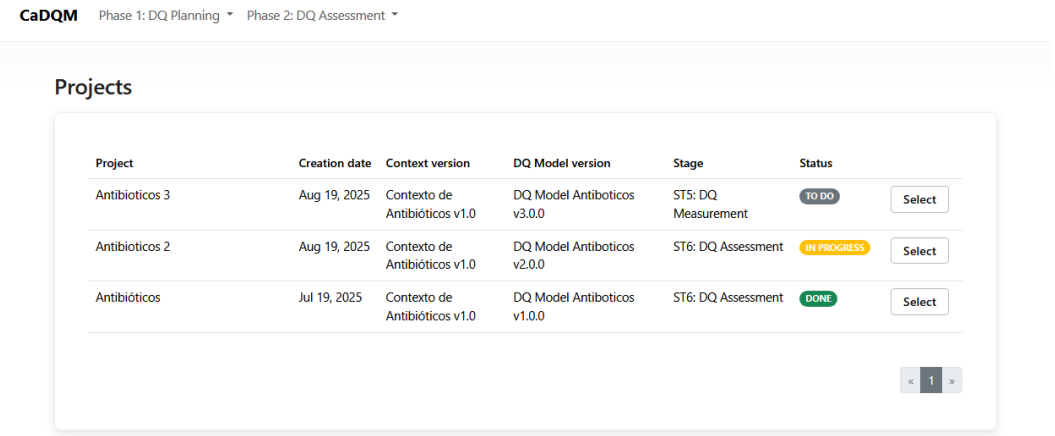

Figura 5.1: Interfaz de la pantalla de inicio de la herramienta.

5.2.2. Dashboard del Proyecto

El *Dashboard* constituye la vista central de la aplicación para la gestión del flujo de trabajo dentro de un proyecto. Su propósito principal es centralizar información general del proyecto en ejecución, incluyendo su etapa actual (`current_stage`) y su estado de avance, como se observa en la Figura A.5. Además, desde esta vista, el usuario puede retomar la ejecución de la Fase 2, accediendo a la última actividad pendiente de la etapa actual.

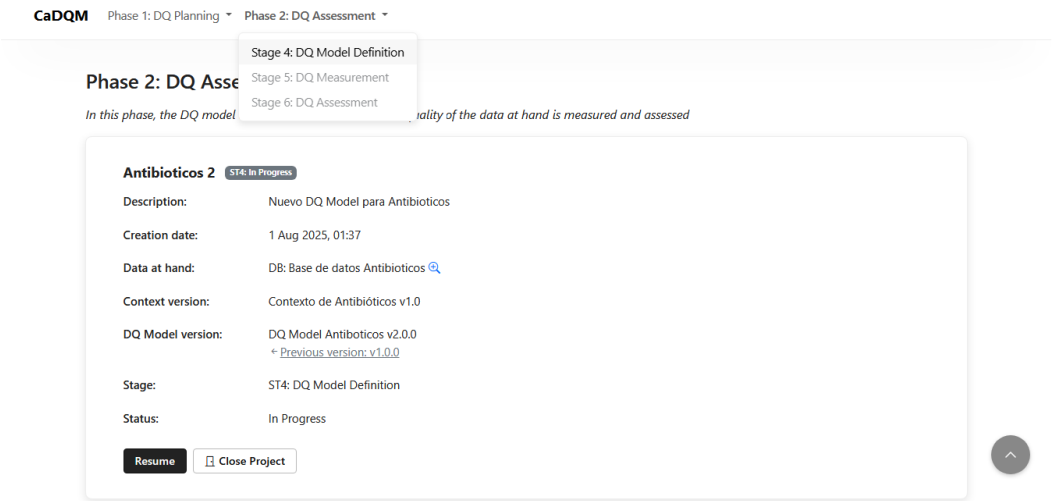

Figura 5.2: Interfaz del *Dashboard* de un proyecto que se encuentra ejecutando la Etapa 4 de CaDQM, definiendo un modelo de CD, que cuenta con una versión previa.

El inicio de la ejecución de la Fase 2 ocurre cuando el proyecto aún no cuenta con un modelo de CD definido, encontrándose en la etapa `ST4` con estado `TO_DO`. A partir de esta instancia, la herramienta permite la creación de un nuevo modelo de CD, registrado con una versión inicial (`1.0.0`) y asociado automáticamente con la versión del modelo de contexto y el *data at hand* correspondientes al proyecto.

Si bien más adelante en el actual capitulo se presentan diagramas de flujo detallados para cada etapa, a continuación se resume cómo se controla el avance de las etapas ejecutadas (`ST4`, `ST5` y `ST6`) mediante los estados `TO_DO`, `IN_PROGRESS` y `DONE`, según la siguiente lógica de transición:

- Al crear el modelo de CD, el estado de `ST4` cambia de `TO_DO` a `IN_PROGRESS`.
- Al finalizar la definición del modelo de CD, el estado de `ST4` cambia de `IN_PROGRESS` a `DONE`.
- Cuando `ST4` alcanza el estado `DONE`, la etapa `ST5`, que se encuentra en `TO_DO`, se convierte automáticamente en la `current_stage`. Esta misma lógica se aplica para la etapa `ST6`.

Esta lógica de avance está directamente vinculada con el menú principal (Figura A.5), que habilita progresivamente el acceso a las etapas de la Fase 2 según el estado registrado, permitiendo así revisar las etapas ya completadas.

Adicionalmente, en este *Dashboard*, una vez que el proyecto cuenta con un modelo de CD finalizado (`ST4` en estado `DONE`), la herramienta habilita la creación de una nueva versión del modelo de CD (por ejemplo, de `1.0.0` a `2.0.0`). Esta acción implica la generación automática de un nuevo proyecto, el cual mantiene la asociación con la misma versión de contexto y el mismo *data at hand*, considerados en la versión previa (`1.0.0`) del modelo de CD. La aplicación ofrece dos modalidades para la creación de la nueva versión del modelo de CD: crear un modelo de CD vacío o generar una copia completa del modelo de CD previo, reutilizando su estructura, priorización y selección de problemas de CD, así como la definición de todos los conceptos de CD, desde las dimensiones hasta los métodos de CD aplicados.

En caso de que existan múltiples versiones de un modelo de CD, el *Dashboard* permite acceder tanto al modelo de CD inmediatamente anterior como el siguiente, lo que implica cambiar el proyecto en el que se está trabajando, ya que cada versión corresponde a un proyecto distinto.

5.2.3. Etapa 4: Definición del Modelo de CD *(ST4: DQ Model Definition)*

La Etapa 4 de CaDQM sigue un flujo de navegación tipo asistente *(wizard)*, que guía al usuario a través de las distintas actividades de la etapa. La navegación entre estas actividades se habilita progresivamente según el criterio de realización de cada tarea, permitiendo la revisión de pasos previos. Para apoyar este proceso de definición de un modelo de CD considerando un modelo de contexto, se incluye un botón fijo en la esquina inferior derecha de cada interfaz (Figura 5.3), que permite consultar en cualquier momento todos los componentes de contexto y los problemas de CD del proyecto.

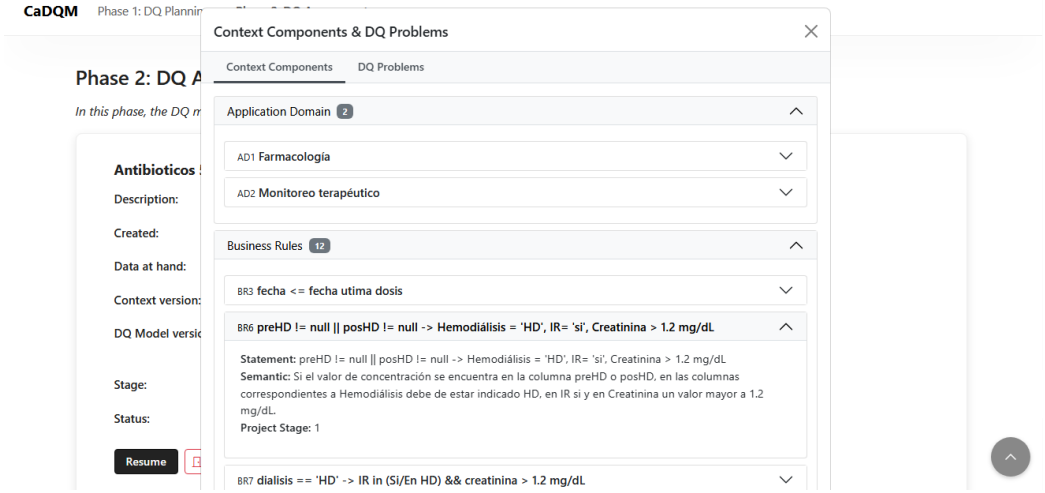

Figura 5.3: Vista emergente para componentes de contexto y problemas de CD disponibles para el proyecto en ejecución.

En la Figura 5.4 se presenta un diagrama donde se muestra el flujo de ejecución para la Etapa 4, mapeando los diferentes estados de dicha etapa con las actividades o tareas correspondientes.

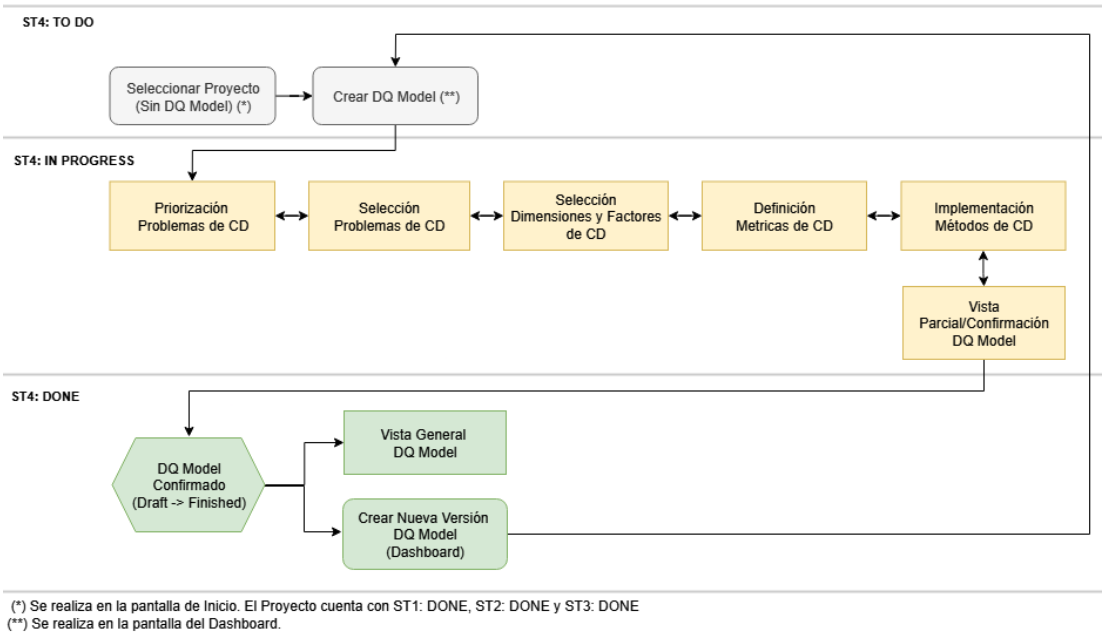

Figura 5.4: Diagrama de flujo de la ejecución de la Etapa 4 *(ST4: DQ Model Definition)*.

Como se aprecia en el diagrama de la Figura 5.4, una característica importante para la ejecución de la Etapa 4, es la incorporación al modelo de CD de un estado propio: `draft` o `finished`, independiente del estado de la etapa del proyecto. Esto se debe a que, a diferencia de las otras etapas, la finalización del modelo de CD no se infiere automáticamente, sino que requiere una acción manual por parte del usuario. Esto permite confirmar que todos los conceptos de CD requeridos fueron definidos y agregados al modelo de CD. De este modo, mientras el modelo de CD se encuentra en construcción (estado `draft`), la etapa `ST4` tendrá el estado `IN_PROGRESS`.

A continuación, se describe la implementación realizada para cada una de las actividades definidas por CaDQM, explicando su funcionamiento y las distintas decisiones de implementación adoptadas.

Priorización y Selección de Problemas de CD

La primera actividad de la Etapa 4 de CaDQM tiene como objetivo priorizar los problemas de CD identificados en la Fase 1, y seleccionar los problemas de CD más relevantes para la definición del modelo de CD. Esta priorización guía la selección de las dimensiones y los factores de CD que compondrán el modelo de CD. Para implementar esta actividad, se elaboró un flujo compuesto por dos interfaces, una para la priorización y otra para la selección de problemas de CD, lo que permitió separar responsabilidades y reducir la carga de trabajo del usuario.

- **1. Priorización Problemas de CD**

  En esta interfaz se muestran todos los problemas de CD del proyecto, provenientes de la Fase 1, con una prioridad inicial media por defecto. La implementación de esta interfaz sigue un mecanismo de reorganización visual de los problemas de CD, permitiendo actualizar la prioridad de los mismos mediante la asignación a columnas que representan las prioridades: `High`, `Medium` y `Low`. En la Figura 5.5 se presenta un ejemplo de esta interfaz.

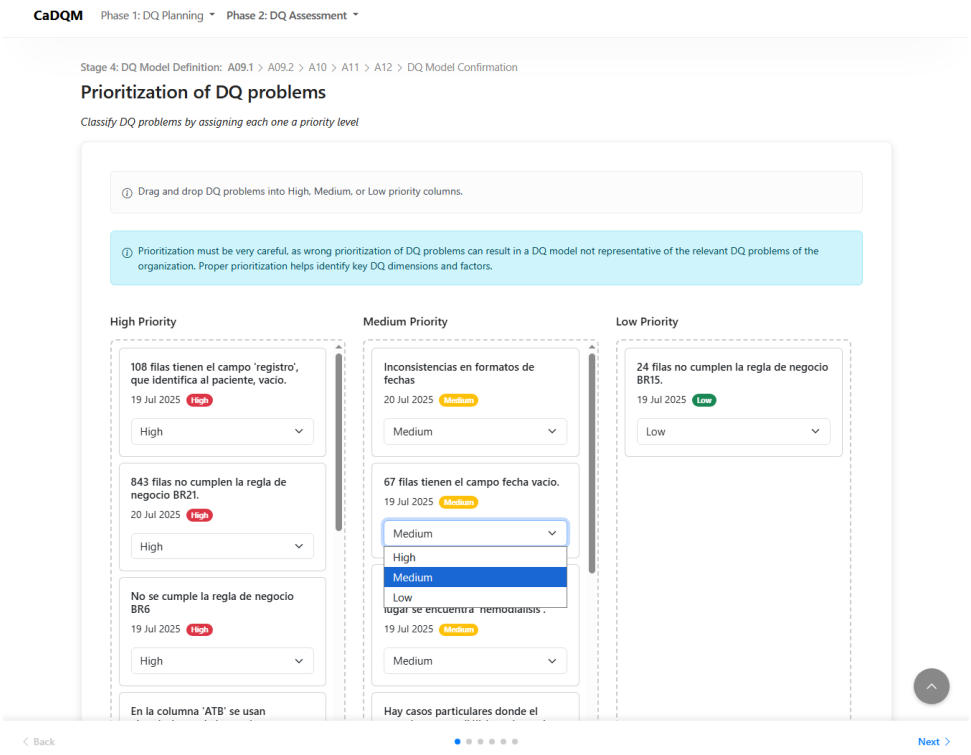

Figura 5.5: Vista de la interfaz de la primera actividad de la Etapa 4: priorización de problemas de CD mediante la reorganización de columnas de prioridad.

- **2. Selección de Problemas de CD**

  En una segunda interfaz se presentan todos los problemas de CD previamente priorizados, permitiendo la selección de aquellos a tener en cuenta para la definición del modelo de CD. Además, si un problema de CD ya ha sido considerado y asociado a la definición de alguna dimensión o factor de CD, no puede ser removido de la selección de los problemas de CD mientras existan dependencias.

Selección de Dimensiones y Factores de CD

El proceso de definición de un modelo de CD basa en el uso de conceptos de CD reutilizables. Por ejemplo, una dimensión de CD es un concepto que contiene los atributos que la definen, y se incorpora al modelo de CD mediante una asociación. Para apoyar al usuario en este proceso, la herramienta sugiere inicialmente un conjunto de dimensiones y factores de CD predefinidos, que surgen en el marco teórico. El usuario puede seleccionar estos conceptos de CD existentes para agregarlos al modelo de CD o, alternativamente, crear nuevos conceptos de CD. En la Figura 5.6 se presenta este flujo de trabajo, el cual además, sienta las bases de los principales pasos a seguir para la selección e inclusión en el modelo de CD de los diferentes conceptos de CD.

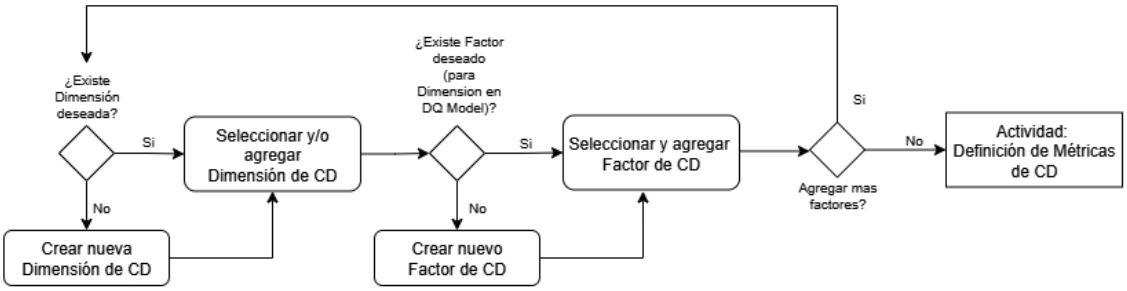

Figura 5.6: Diagrama de flujo para la definición de dimensiones y factores de CD en la Etapa 4.

La interfaz de esta actividad se organiza en tres bloques funcionales que ofrecen distintas maneras de definir dimensiones y factores para el modelo de CD:

- **Definición de Dimensiones y Factores desde cero:** Este enfoque sigue exactamente el flujo presentado en la Figura 5.6, donde se debe seleccionar un concepto de CD existente para agregarlo al modelo de CD, mediante su asociación al mismo

  - Las dimensiones de CD pueden seleccionarse de una lista de definiciones existentes, o crear nuevas en caso que no hallar entre las sugeridas la dimensión que se desea agregar. Al momento de agregar una dimensión al modelo de CD, este paso va acompañado, por la posibilidad asociar (o no) componentes de contexto y/o problemas de CD que sugieran su inclusión.

  - Para los factores de CD la lógica es similar, pero sigue una jerarquía. Primero se debe seleccionar una dimensión de CD ya agregada al modelo de CD, lo que habilita la lista de factores de CD posibles para la dimensión de CD seleccionada. A partir de este punto, el flujo es análogo al de la selección de dimensiones, pudiendo seleccionar un factor de CD existente o crear uno nuevo, el cual puede surgir (o no), de un componente de contexto y/o problemas de CD. En la Figura 5.7 se ilustra un ejemplo de la interfaz de la aplicación para esta tarea.

  En este proceso de definición de dimensiones y factores de CD, la herramienta se va retroalimentando de nuevos conceptos de CD a ser sugeridos por la misma. Para una gestión adecuada de estos conceptos de CD, se permite la eliminación de las dimensiones y factores sugeridos, a excepción de aquellos ya predefinidos, cargados inicialmente en la herramienta (conceptos de CD que surgen del marco teórico).

Figura 5.7: Vista parcial de la interfaz para la selección de dimensiones y factores de CD. En este caso se selecciona un factor de CD basada en la asociación con los componentes de contexto y los problemas de CD que justifican su inclusión al modelo de CD.

- **Selección de Dimensiones y Factores a partir de Problemas de CD:** En este bloque se listan los problemas de CD previamente priorizados y seleccionados. Al elegir un problema de CD, se presentan todas las dimensiones y sus respectivos factores de CD asociados por defecto. Este enfoque más dinámico permite agregar múltiples dimensiones y factores de CD simultáneamente. En este caso, al seleccionar un factor de CD, la dimensión de CD correspondiente también se agregará al modelo de CD si no existe, o se asociará al factor de CD si la dimensión ya está incluida en el modelo de CD.
- **Recomendación de Dimensiones y Factores generada por IA:** El tercer bloque incorpora un motor de sugerencias impulsado por IA, que combina las dimensiones y factores de CD predefinidos en la herramienta, con los componentes de contexto y los problemas de CD seleccionados en la actividad anterior. A partir de esta información, el sistema genera un par dimensión–factor sugerido, indicando los componentes de contexto y problemas de CD de los que surgen cada recomendación, y ofreciendo una texto explicativo del razonamiento detrás de ella. Esta funcionalidad se describe en detalle en el Capítulo 6, "Integración con IA".

Para el seguimiento del proceso de construcción del modelo de CD y del avance de la actividad, la interfaz incorpora un bloque clave que, además de presentar información general del modelo de CD y del proyecto al que pertenece, permite visualizar de forma anidada las dimensiones y sus factores de CD definidos en la actividad. Adicionalmente, en caso que existan, se muestran aquellos componentes de contexto y/o problemas de CD que haya sugerido cada concepto de CD. Este enfoque se incorpora en todas las interfaces para la selección o confirmación de los conceptos de CD agregados al modelo de CD, pudiendo observar un ejemplo en la Figura 5.12.

Mientras el modelo de CD se encuentra en fase de construcción (estado `draft`), el usuario puede seguir agregando dimensiones y factores. En esta vista parcial, también se habilita la edición de los elementos del modelo de CD, permitiendo ajustar la vinculación de los componentes de contexto y los problemas de CD (seleccionar nuevos o deseleccionar existentes), asociados a las dimensiones y factores agregados. Además, se permite la eliminación de dimensiones y factores del modelo. Este proceso elimina la asociación entre el concepto de CD y el modelo de CD, lo que conlleva la eliminación de todos los conceptos de CD posteriores en la jerarquía asociados a la dimensión o factor de CD eliminado.

Definición de Métricas de CD

Una métrica de CD representa una forma de medir un factor de CD, y su definición implica la especificación de su propósito, la granularidad de la medición y el dominio de los valores de CD obtenidos como resultado. El objetivo de esta actividad es definir al menos una métrica de CD para cada factor de CD incluido en el modelo de CD.

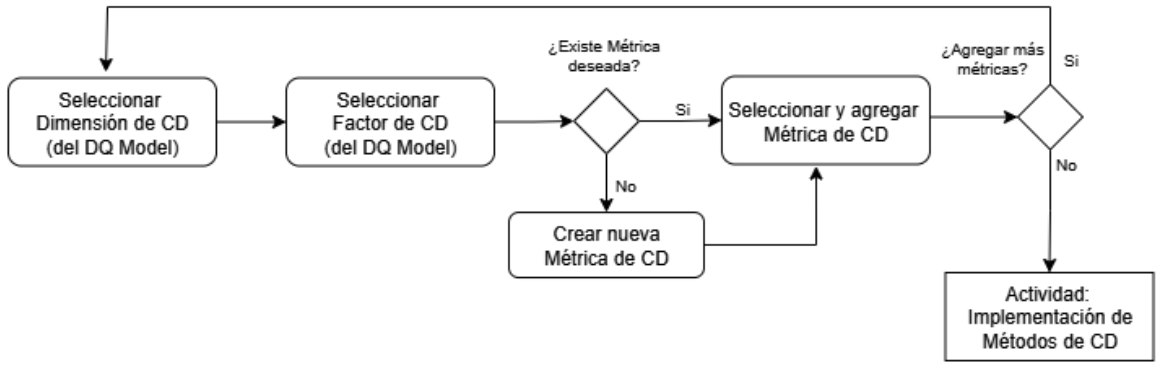

Figura 5.8: Diagrama de flujo para la definición de métricas de CD en la Etapa 4.

Para esta actividad, la herramienta ofrece una interfaz que sigue un flujo (Figura 5.8) prácticamente análogo al de la selección de factores de CD, impuesto por la jerarquía de los conceptos de CD ya agregados al modelo de CD. Primero se presentan las dimensiones del modelo de CD, luego sus factores asociados. A partir de un factor de CD, es posible seleccionar una métrica de CD ya existente o crear una nueva para su inclusión al modelo de CD. Como apoyo, en este paso se presentan los componentes de contexto (si hay) que sugirieron anteriormente al factor de CD) que la métrica de CD desea medir. Se asegura que el experto en CD tenga control total sobre esta selección, pudiendo deseleccionar los componentes de contexto listados o añadir cualquier otro componente de contexto que sugiera a la métrica de CD.

Para la creación de una nueva métrica de CD, la herramienta ofrece como opciones las posibles granularidades de una BD relacional: base de datos, tabla, columna, tupla o celda. En cuanto al dominio de resultados, se limita la elección a valores de tipo *Boolean* $\{0, 1\}$ o *Float* en el rango $[0, 1]$, utilizados para representar porcentajes.

Implementación de Métodos de CD

Una vez definidas las métricas de CD, la última actividad para la completar la construcción del modelo de CD, consiste en la definición de los métodos de CD que implementen dichas métricas de CD. Un método de CD se define principalmente mediante la definición de un algoritmo según el propósito de la métrica de CD, indicando además el tipo de datos de entrada y de salida. Este algoritmo consiste en un código *SQL* parametrizable, adaptable a diferentes esquemas y conjuntos de datos, permitiendo así además, la reutilización de métodos de CD en la herramienta.

El flujo de trabajo (Figura 5.9) mantiene la misma lógica jerárquica de las actividades previas: se navega desde la dimensión, hacia el factor y finalmente la métrica de CD para la cual se definirá el método de CD. En este punto, a diferencia de lo que ocurre en la selección de los conceptos de CD anteriores, donde se permite asociar libremente componentes de contexto, al agregar un método de CD al modelo de CD, solo pueden asociarse aquellos componentes de contexto que ya hayan sido vinculados previamente a la métrica de CD correspondiente. Esta restricción se debe al hecho de que el método de CD es una implementación de la métrica de CD, y por lo tanto, solo puede considerar los componentes de contexto que hayan sugerido la métrica de CD.

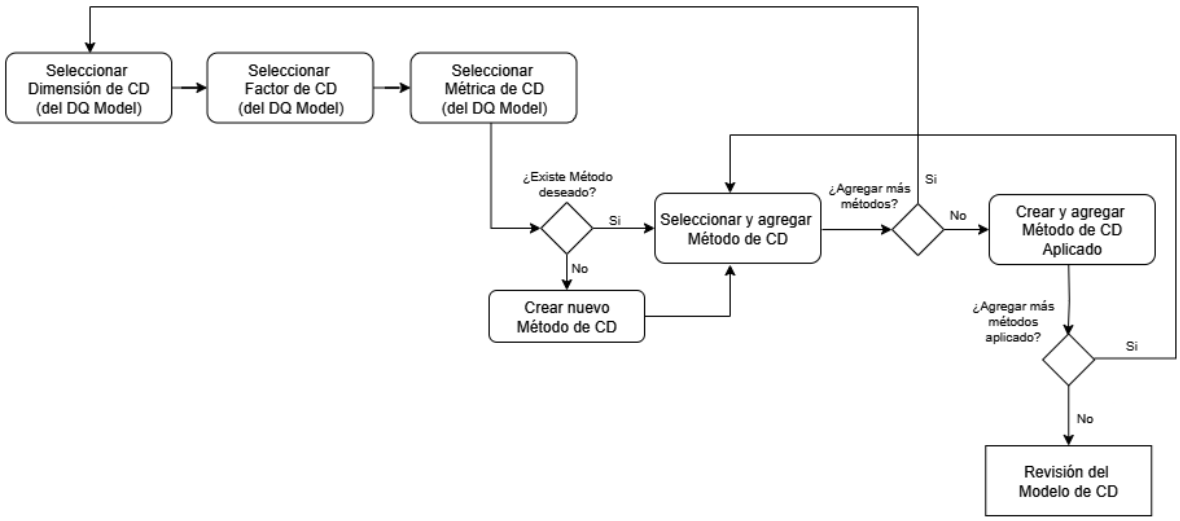

Figura 5.9: Diagrama de flujo para la definición de métodos de CD e implementación de sus métodos de CD aplicados en la Etapa 4.

Para apoyar al usuario en la definición de un método de CD, la herramienta incorpora una funcionalidad de asistencia basada en IA. Esta funcionalidad permite autocompletar los campos del formulario de creación del método de CD (Figura 5.10), generando una sugerencia inicial a partir de la información definida en la métrica de CD correspondiente. La propuesta incluye un algoritmo expresado en código *SQL* como posible implementación, que el usuario puede revisar, modificar o reemplazar por nuevas sugerencias. Esta funcionalidad se describe con mayor detalle en el Capítulo 6.

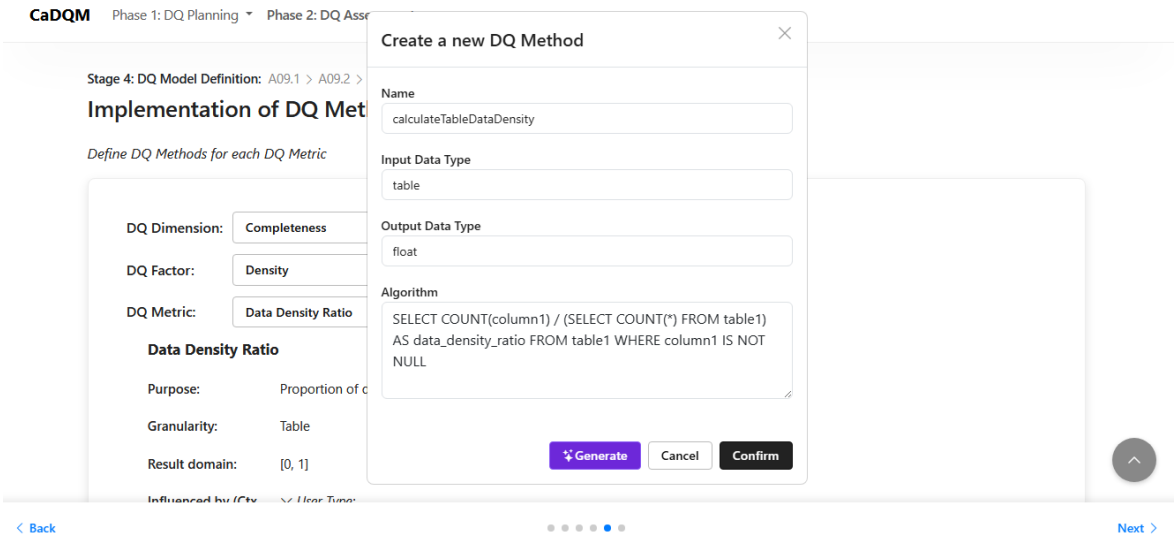

Figura 5.10: Vista de interfaz para la creación de métodos de CD con autocompletado mediante IA.

Finalmente, para completar la definición del modelo de CD, la herramienta permite definir los métodos de CD aplicados, es decir, la instanciación de los métodos de CD sobre atributos específicos del *data at hand*, adaptando el algoritmo del método base según corresponda. Durante este proceso, también se indica si el método aplicado corresponde a una operación de medición o de agregación. La definición de estos métodos aplicados se apoya en un flujo simplificado que evita la repetición de pasos de navegación en la jerarquía del modelo de CD, donde la interfaz (Figura 5.11) presenta un filtro por dimensión, a partir del cual se listan los métodos de CD asociados, manteniendo la trazabilidad con el factor y la métrica de CD correspondientes.

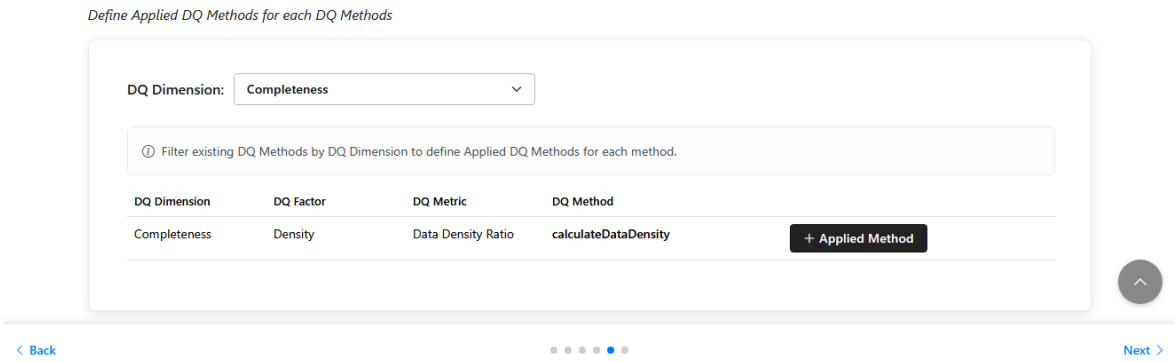

Figura 5.11: Vista del bloque para la implementación de métodos de CD en interfaz de la herramienta para la actividad para definición de métodos de CD.

Vista General del Modelo CD

El último paso de la Etapa 4 consiste en la revisión global del modelo de CD una vez definidos e incorporados todos los conceptos de CD, desde las dimensiones hasta los métodos aplicados. Para ello, la herramienta incluye una interfaz (Figura 5.12) que ofrece una vista del modelo de CD organizada de manera jerárquica, permitiendo recorrer de forma anidada todos los conceptos de CD que lo componen, mediante un mecanismo de ítems colapsables. Adicionalmente, esta interfaz incorpora la opción de generar y descargar un reporte en formato *PDF* con el modelo de CD definido, ya sea en su versión parcial (`draft`) o definitiva (`finished`).

Esta vista funciona como un punto de control, cuyo propósito varía según el estado del modelo de CD:

- Estado `draft`: la vista actúa como pantalla de confirmación, permitiendo al usuario finalizar el modelo de CD. Al hacerlo, su estado pasa a `finished` y la etapa `ST4` se marca como `DONE`, habilitando la transición hacia la siguiente etapa (`ST5`).
- Estado `finished`: el modelo de CD se presenta de forma inmutable, reflejando su versión final. En este estado se habilita la posibilidad de generar una nueva versión del modelo de CD desde el *Dashboard* del proyecto.

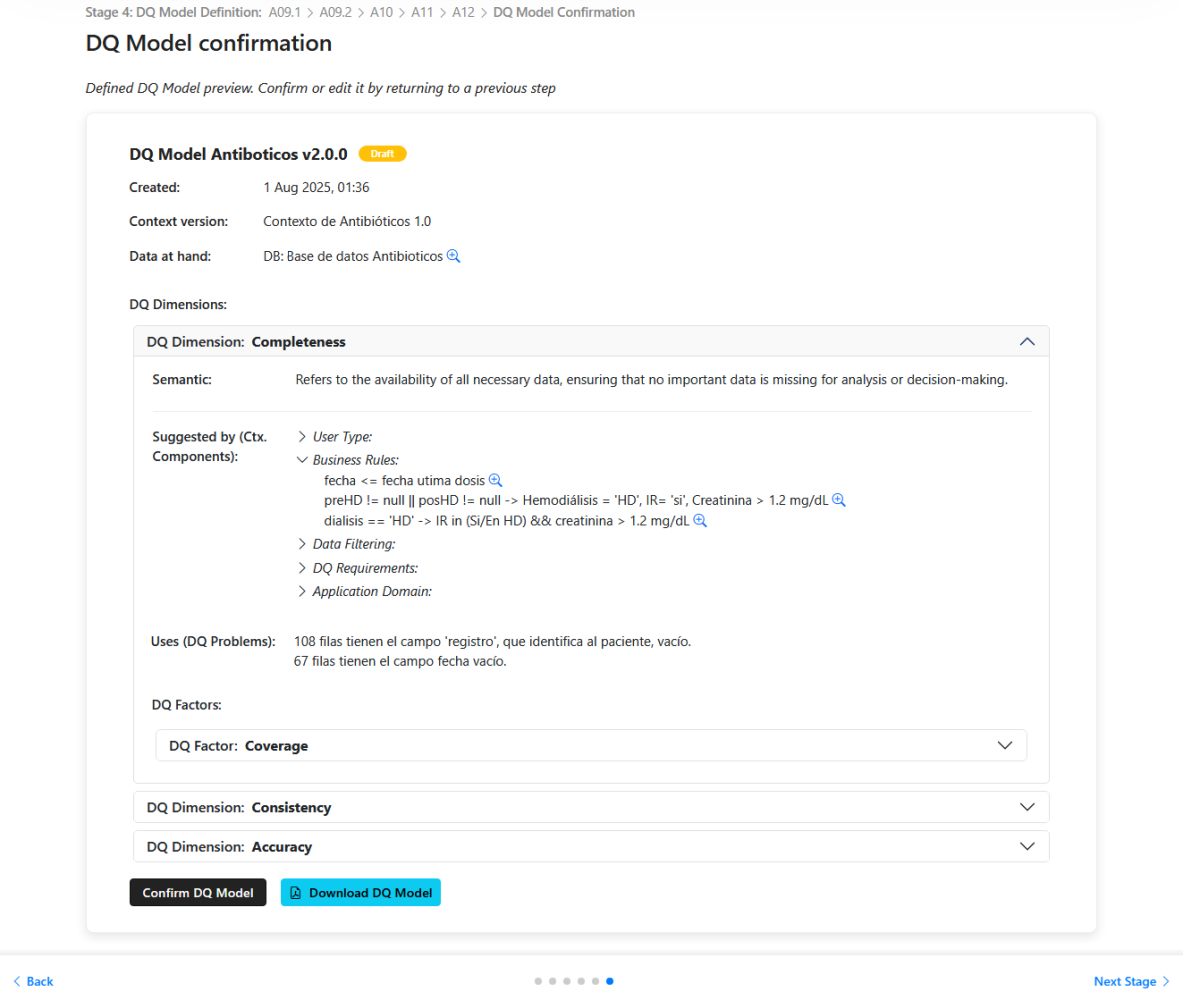

Figura 5.12: Vista de revisión del modelo de CD completo, previo a confirmarlo (en estado `draft`). Se muestran las dimensiones en el primer nivel de la jerarquía del modelo de CD, incluyendo una dimensión de CD desplegada, con su definición, los componentes de contexto y los problemas de CD que lo sugieren, y sus factores de CD asociados.

5.2.4. Etapa 5: Medición de la CD *(ST5: DQ Measurement)*

Una vez finalizada la definición del modelo de CD, se habilita automáticamente la ejecución de la Etapa 5 (estado `TO_DO`), enfocada en la medición de la CD. Las actividades de esta etapa se organizan en dos interfaces de la herramienta, una dedicada ejecución de los métodos de CD aplicados y otra a la visualización de los resultados de las mediciones realizadas. Los valores de CD obtenidos como resultado de la medición de la CD se almacenan en una BD dedicada, denominada `DQ Metadata`

A continuación, se describen las actividades de esta etapa, cuyo principal flujo de trabajo a través de los diferentes estados, se resume en la Figura 5.13.

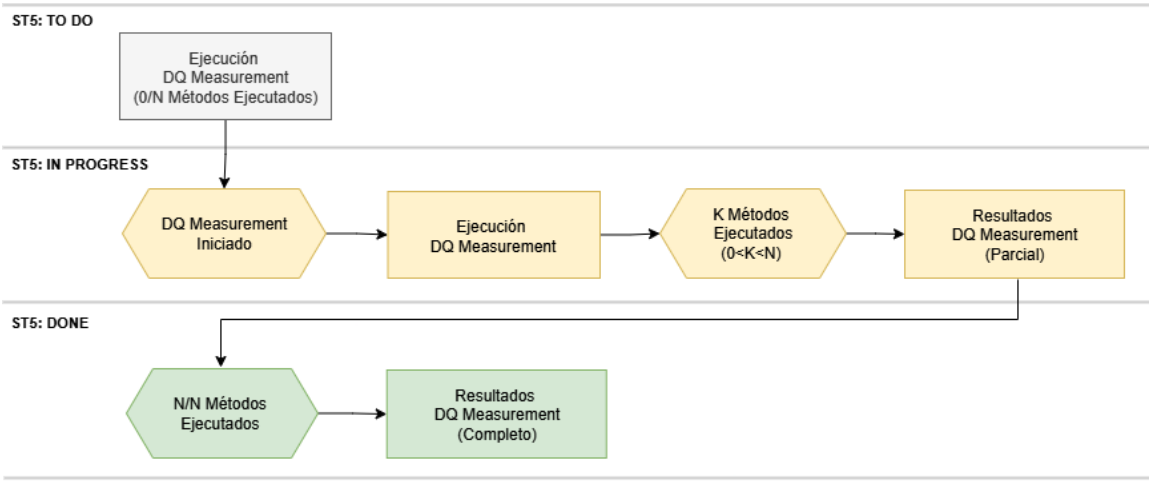

Figura 5.13: Diagrama de flujo de la Etapa 5 *(ST5: DQ Measurement)*.

Ejecución de la Medición de la CD

El proceso de medición de CD se inicia mediante una acción explícita del usuario, que crea una nueva instancia de ejecución para el modelo de CD, cambia el estado del proyecto de `TO_DO` a `IN_PROGRESS`, y habilita la interfaz que lista todos los métodos de CD aplicados definidos para el modelo de CD.

Para un mejor seguimiento del proceso y avance de la actividad, la herramienta permite el filtrado de los métodos de CD aplicados según su estado: `pending` (aún no ejecutados) o `completed` (ya ejecutados). En ambos casos, los métodos se presentan de manera compacta a través de una tabla que incluye para cada uno, la siguiente información:

- Nombre del método y de su método de CD aplicado.
- Tablas y columnas de la BD (*data at hand*) sobre las que se aplica el método de CD.
- Trazabilidad de la jerarquía del modelo de CD completo (métrica, factor y dimensión de CD).

Desde esta vista (Figura 5.14), el usuario puede ejecutar uno o varios métodos de CD aplicados en paralelo, optimizando el proceso de medición de la CD. Además, mientras un método no haya sido ejecutado correctamente, la herramienta permite su edición, cubriendo un posible caso en que el usuario haya definido una consulta *SQL* incorrecta en la definición del algoritmo del método de CD aplicado, permitiéndole realizar los ajustes necesarios antes de su ejecución.

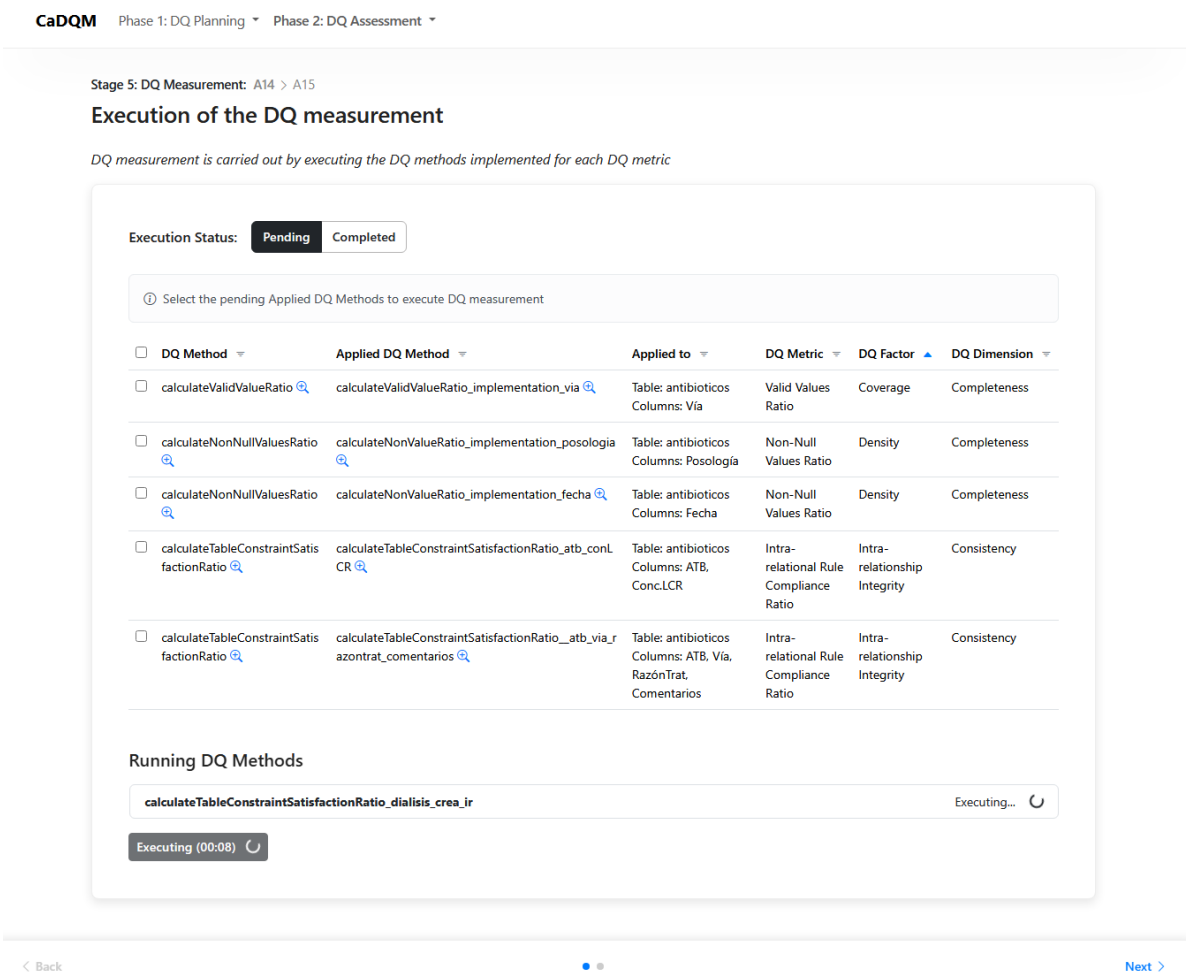

Figura 5.14: Interfaz de la Medición de la CD mediante la ejecución de los métodos de CD aplicados.

La medición de la CD y por lo tanto la Etapa 5, se da por completada (`IN_PROGRESS` a `DONE`) cuando todos los métodos de CD aplicados del modelo de CD fueron ejecutados, como se observa en el diagrama de la Figura 5.13. Esta acción, habilita el pasaje a la Etapa 6 (Evaluación de la CD) dentro de la herramienta.

Resultados de la Medición de la CD

En esta interfaz se presentan los valores de CD (*DQ Values*), obtenidos mediante la ejecución de los métodos de CD aplicados. Esta pantalla (Figura 5.15) es accesible en cualquier momento, incluso si la ejecución completa no ha finalizado, lo que permite al usuario verificar resultados parciales.

La visualización de los resultados ofrece dos enfoques principales:

- Orientado a Métodos de CD: muestra los resultados para todos los métodos de CD aplicados ejecutados.
- Orientado a Datos: se enfoca en los resultados para elementos específicos de las tablas del *data at hand*.

En particular, el enfoque orientado a datos, requiere seleccionar una tabla específica para ver los resultados de las mediciones en ella. Además, el usuario puede aplicar un filtro por la granularidad del método de CD (`Table`, `Column`, `Cell` o `Tuple`) o seleccionar la opción `All` para ver todos los resultados sin distinción de granularidad. En la Figura 5.15 se presenta un ejemplo de esta interfaz con los resultados de la medición de CD ejecutada para un método de CD aplicado, filtrando con granularidad columna y una tabla especifica del *data at hand*.

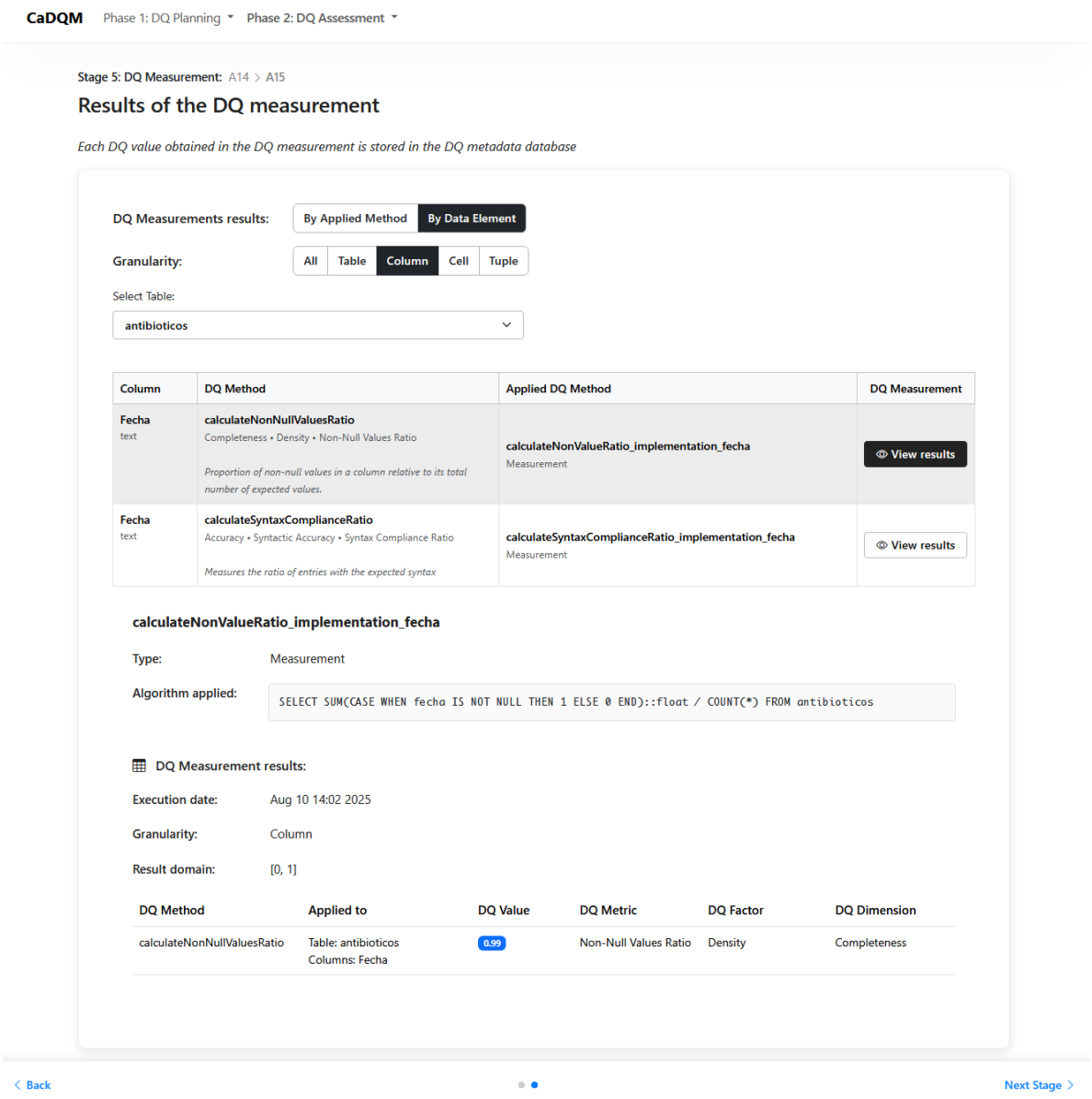

Figura 5.15: Interfaz de los resultados de la Medición de CD para un método de CD aplicado, filtrado por granularidad columna para una tabla especifica de la BD del *data at hand*.

5.2.5. Etapa 6: Evaluación de la CD *(ST6: DQ Assessment)*

La Etapa 6 es la última de la Fase 2 de CaDQM. Con un modelo de CD ya definido y las mediciones de CD realizadas, esta etapa se centra en la evaluación de la CD a partir de los valores de CD obtenidos. Su implementación se organizó a través de dos interfaces con los siguientes propósitos: primero, la definición de umbrales (*thresholds*) de evaluación, y finalmente, la ejecución de la evaluación de CD.

A continuación, se describen las actividades de esta etapa a través de cada interfaz, cuyo flujo de trabajo, junto a las transiciones de estado, se ilustra de manera resumida en la Figura 5.16.

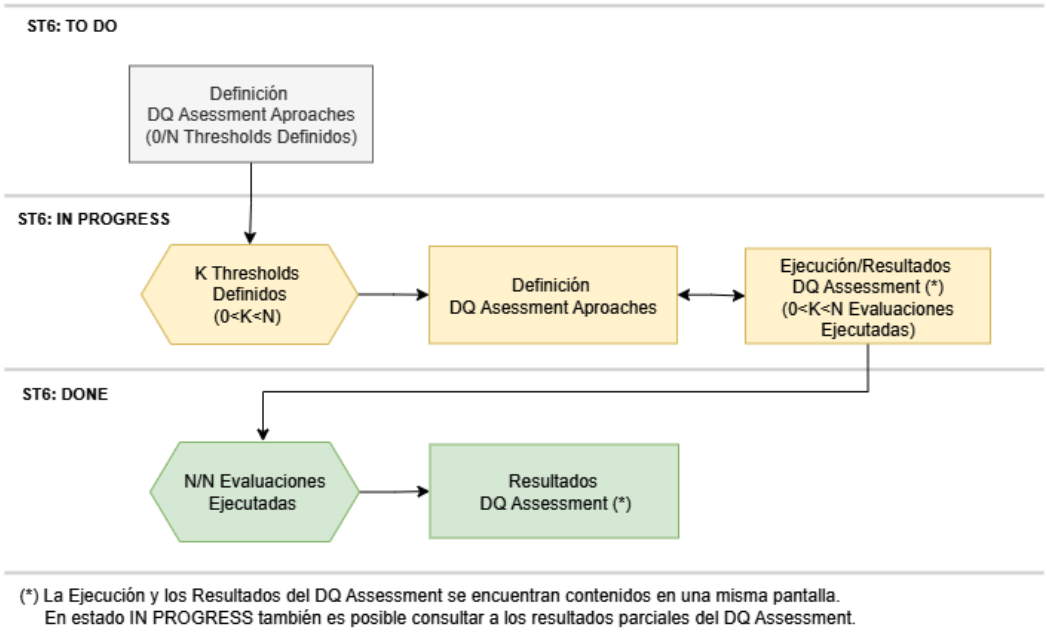

Figura 5.16: Diagrama de flujo de la Etapa 6 *(ST6: DQ Assessment)*.

Definición de Umbrales para la Evaluación de CD

Esta interfaz (Figura 5.17) tiene como propósito la definición de umbrales de evaluación de CD (*thresholds*). Estos umbrales representan criterios de evaluación cualitativos que permiten clasificar los valores de CD cuantitativos obtenidos en la Etapa 5. La herramienta permite que el usuario personalice estos criterios definiéndolos mediante rangos numéricos, tales como: 0–80 % (calidad baja), 81 %–90 % (calidad media) y >90 % (calidad excelente), por ejemplo.

Los umbrales de evaluación de CD se definen individualmente para cada método de CD aplicado. Para facilitar el seguimiento del proceso de definición de los umbrales, se les asigna un estado: inicialmente todos los métodos de CD se encuentran en `pending`, pasando a `defined` una vez que se han establecido sus umbrales.

Para facilitar la definición de los umbrales, se integró como apoyo mediante la herramienta la siguiente información:

- La definición del método de CD aplicado.
- Un resumen de los resultados de su medición obtenidos en la Etapa 5.
- Un mensaje informativo que explica cómo definir los umbrales.
- Los componentes de contexto asociados al método de CD, que podrían proponer los rangos numéricos de los valores de CD para los umbrales.

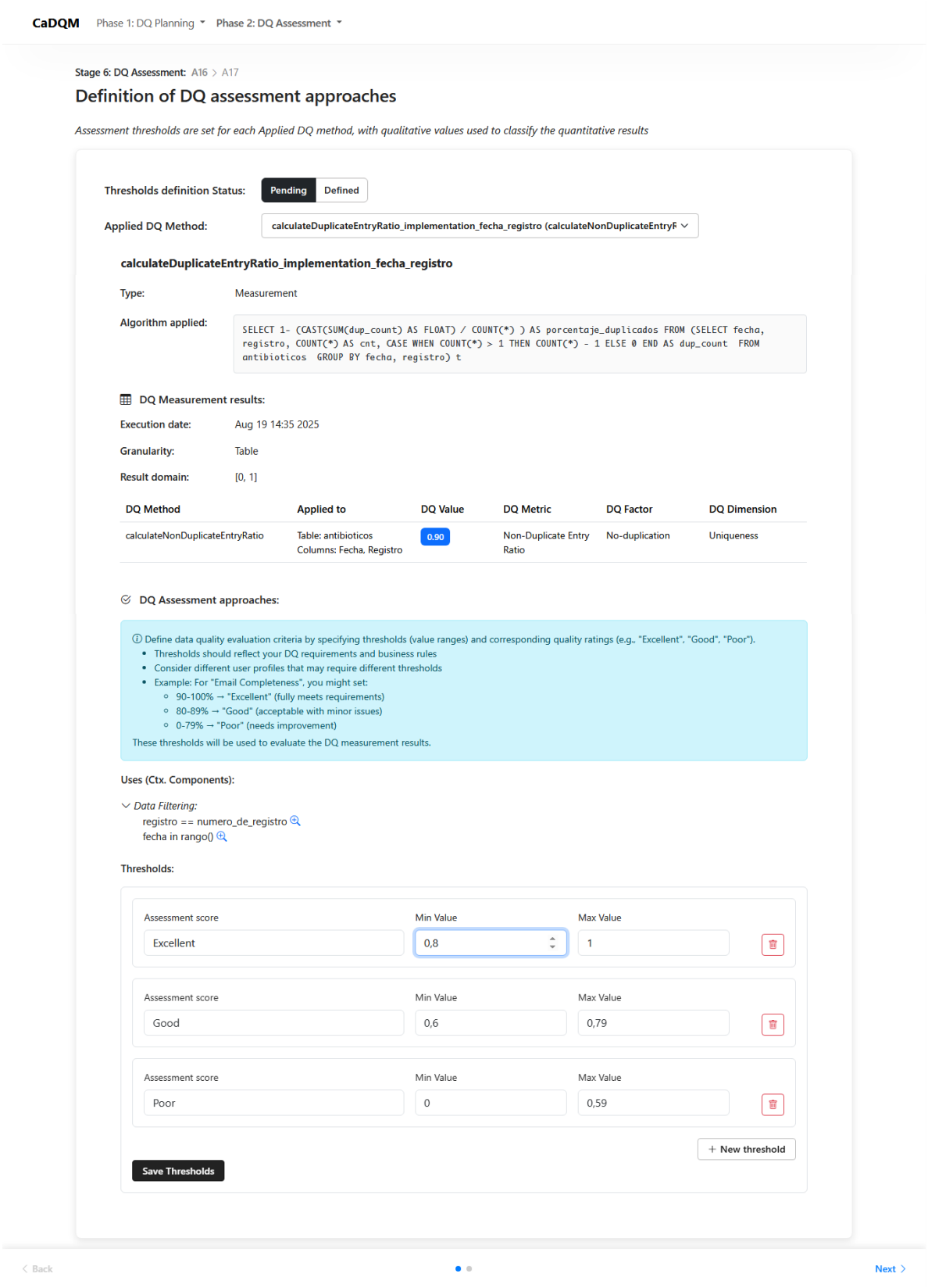

Figura 5.17: Interfaz para la definición de los umbrales de evaluación para un método de CD aplicado especifico, con dominio de resultado de tipo *Float*.

La interfaz (Figura 5.17) para definir los umbrales varía según la granularidad de la medición de la CD:

- **Mediciones con dominio de resultado de tipo *Float* ($[0, 1]$):** Para mediciones de CD con granularidad tabla o columna que devuelven un único valor de CD, la herramienta sugiere inicialmente tres umbrales por defecto: *Excellent* (0.8–1), *Good* (0.6–0.79) y *Poor* (0–0.59). Sin embargo, esta tarea es completamente personalizable: el usuario puede modificar los valores mínimos y máximos de cada umbral, así como la calificación cualitativa asociada a cada rango (la etiqueta, como *Poor*, *Good*, *Excellent*), o bien eliminar los sugeridos y crear nuevos. Como restricción, se necesitan al menos dos umbrales definidos, y los rangos no pueden superponerse ni dejar valores de CD sin clasificar.
- **Mediciones con dominio de tipo *Boolean* ($\{0, 1\}$):** Para mediciones de CD con resultados a nivel de fila (granularidad celda o tupla), lo que implica devolver un valor binario (*true* o 1, y *false* o 0) para cada registro evaluado, solo se pueden definir dos umbrales. El usuario puede nombrar estas clasificaciones cualitativas, con una sugerencia por defecto de *Passed* para 1 y *Failed* para 0.

Ejecución de la Evaluación de CD y Visualización de Resultados

En esta última actividad de la ejecución de la Fase 2 con la herramienta, se implementa una interfaz que gestiona la ejecución de la evaluación de la CD utilizando los umbrales previamente definidos, e integra la visualización de los resultados cualitativos obtenidos. En particular, esta interfaz es accesible en todo momento dentro de la Etapa 6. Como se muestra en el diagrama de la Figura 5.16, esto permite al usuario ir definiendo umbrales, ejecutando evaluaciones, y accediendo a la visualización de resultados parciales, aun cuando no se hayan definido todos los umbrales necesarios para la evaluación de la CD.

Este doble propósito se logra a partir de un filtro que permite organizar los métodos de CD aplicados con umbrales definidos que aún no han sido ejecutados (`pending`), de aquellos que sí ya han sido ejecutados (`completed`), pudiendo visualizar dichos resultados. La Figura 5.18 presenta un ejemplo de esta interfaz, mostrando los resultados obtenidos de la evaluación de CD para un método de CD aplicado.

Una vez ejecutada la evaluación de CD para la totalidad de los métodos de CD aplicados definidos para el modelo de CD, la Etapa 6 se da por finalizada. Su estado cambia de `IN_PROGRESS` a `DONE`, concluyendo así la ejecución completa de la Fase 2 de CaDQM a través de la herramienta implementada.

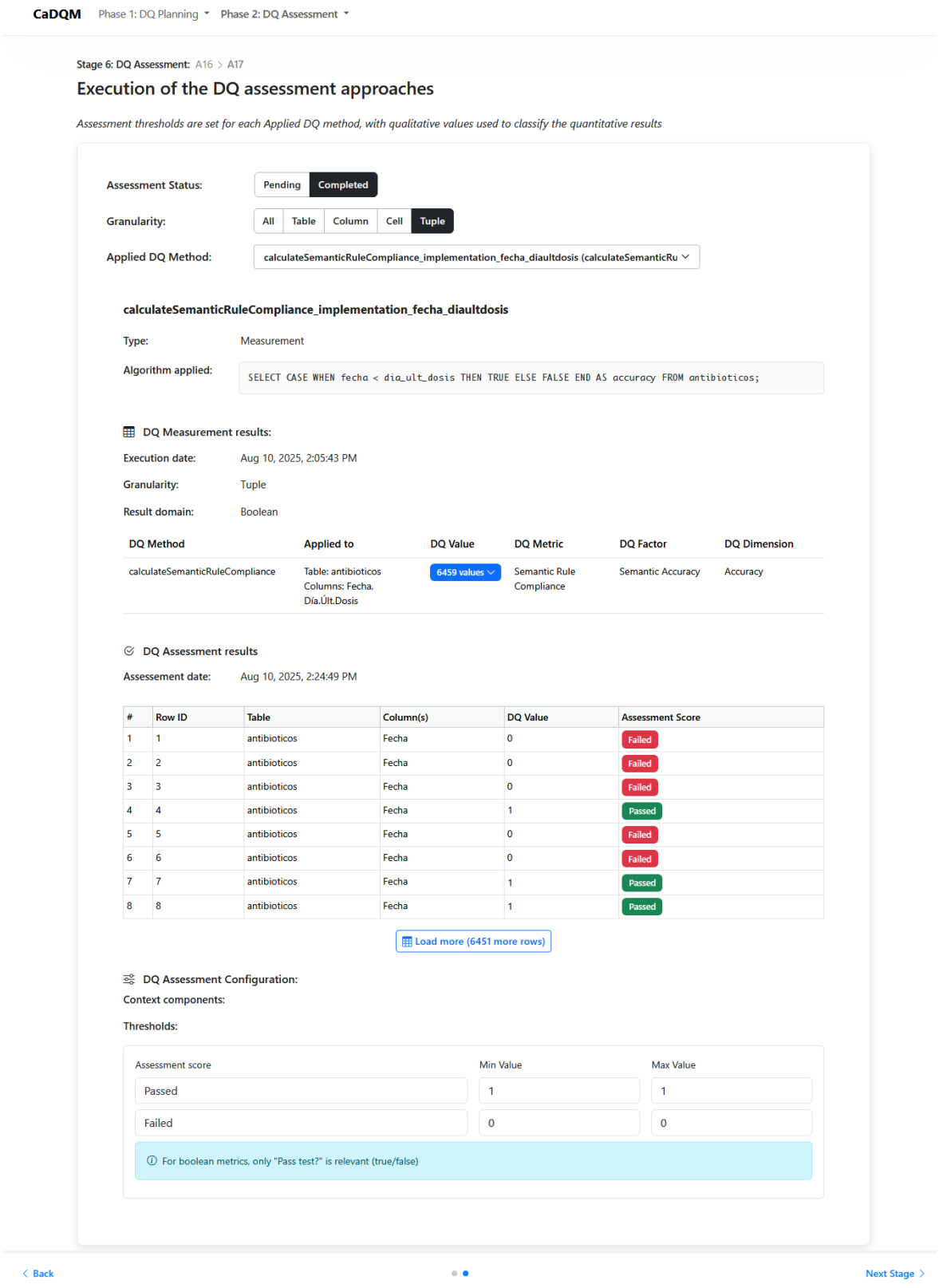

Figura 5.18: Interfaz de resultados de la evaluación de CD para un método de CD aplicado especifico. Se muestran los valores cualitativos obtenidos a partir del umbral definido sobre una medición de CD con dominio de resultado *Boolean* y granularidad tupla.

Capítulo 6

Integración de IA

En este capítulo se presenta un análisis detallado de la integración de funcionalidades de Inteligencia Artificial (IA) en la solución. Se describe el diseño e implementación de dichas funcionalidades y la experimentación realizada para analizar su desempeño y verificar su utilidad práctica en el sistema.

6.1. Tecnologías Utilizadas

Para el desarrollo e integración de funcionalidades de IA en el sistema, se seleccionaron las siguientes tecnologías con el objetivo de combinar capacidad técnica con viabilidad práctica:

- ***LangChain***: *Framework* en *Python* para desarrollar aplicaciones con modelos de lenguaje (*LLMs*) [20]. Se utilizó para construir la lógica que conecta el sistema con el modelo de IA, facilitando el procesamiento de datos de entrada y la generación de respuestas a partir de dichos datos.
- ***Groq***: Plataforma que acelera la ejecución de *LLMs* [12]. Entre los proveedores considerados, *Groq* destacó por su plan gratuito y su fácil integración con *LangChain*. A diferencia de servicios como *OpenAI* o *Gemini*, su plan *Free Tier* permitió iterar rápidamente sin costos en la fase experimental, manteniendo un rendimiento adecuado y tiempos de respuesta bajos.

Modelos de Lenguaje

Se trabajó con la familia de modelos *Llama 3* de *Meta*, analizando las siguientes variantes de *LLMs* disponibles en *GroqCloud* como *Production Models* [13]: *Llama-3-70B-8192* [14], *Llama-3-8B-8192* [15], *Llama-3.1-8b-instant* [16] y *Llama-3-70B-Versatile* [17]. Dichos modelos se consideraron como opciones viables debido a su compatibilidad con *LangChain*, la posibilidad de implementarlos mediante *API* sin necesidad de instalaciones locales, y su adecuación para entornos de producción. En el Anexo C.1 se presenta una tabla comparativa técnica que muestra distintas características de las cuatro variantes de los modelos *Llama 3* consideradas para la experimentación.

6.2. Funcionalidades implementadas

Con el objetivo de apoyar a los usuarios y agilizar ciertas actividades en la definición de modelos de CD, se incorporaron las siguientes dos funcionalidades que integraron el uso de IA en la herramienta:

- Recomendación automática de dimensiones y factores de CD.
- Generación automática de métodos de CD.

Es fundamental destacar que el uso de estas funcionalidades es completamente opcional para el experto en CD. La IA se ofrece como una asistencia durante el proceso de definición del modelo de CD, pero la

ejecución de la metodología no depende de ella. Por lo tanto, la herramienta desarrollada no requiere un gasto económico obligatorio asociado al consumo de *LLMs*.

6.2.1. Recomendación de Dimensiones y Factores de Calidad de Datos

Esta funcionalidad utiliza modelos de lenguaje para analizar un conjunto de componentes de contexto y problemas de CD considerados para la definición del modelo de CD, y a partir de ellos sugerir automáticamente dimensiones y factores de CD.

Diseño de la Solución

La construcción de la solución se basa fuertemente en definición de un *prompt* que actúa como la interfaz principal de comunicación con el *LLM*. Este *prompt* se diseñó para que el modelo de lenguaje, al analizar los datos de entrada (dimensiones y factores de CD, problemas de CD y componentes de contexto de los datos), genere recomendaciones justificadas en una salida en un formato específico. El *prompt* definido se presenta en el Bloque 6.1.

```
DQ_RECOMMENDATION_PROMPT = """As a data quality expert, analyze these
    components and return ONLY a JSON object with:

1. The most relevant DQ dimension and factor
2. Choose ONLY THE STRONGEST supporting evidence and
3. A concise rationale

Available Dimensions and Factors: {dimensions_and_factors}

Data Quality Problems: {dq_problems}

System Context: {context_components}

Return EXACTLY this JSON structure WITHOUT ANY OTHER TEXT OR COMMENTS:
{{
    "recommended_dimension": "dimension_name",
    "recommended_factor": "factor_name",
    "supported_by_problems": [problem_id1, problem_id2],
    "supported_by_context": {{
        "appDomain": [ids], "bizRule": [ids], "dataFilter": [ids],
        "dqMeta": [ids], "dqReq": [ids], "otherData": [ids],
        "otherMeta": [ids], "sysReq": [ids], "task": [ids],
        "userType": [ids]
    }},
    "rationale": "Provide a clear, concise yet detailed explanation that
    links the selected problem descriptions and context component values to
    the recommended dimension and factor. Avoid using IDs or references like
    '(problem 1)'."
}}

IMPORTANT:
- Return ONLY the JSON object, no explanations, headers, or any other text.
- "Try to avoid repeating previously recommended factors, and consider
    alternative factors if similarly supported."
- "Your goal is to recommend a DQ factor that is both well-supported AND not
    previously selected, to encourage coverage across dimensions."
"""
```

Bloque 6.1: Prompt del algoritmo de recomendaciones de dimensiones y factores

El *prompt* definido establece un diálogo claro con el modelo, comenzando por asignarle un rol experto *("As a data quality expert...")* para contextualizar la tarea. Luego, se especifican tres requisitos clave: (1) identificar la dimensión y factor más relevantes, (2) seleccionar únicamente la evidencia más sólida (evitando listas exhaustivas), y (3) proporcionar una justificación concisa que asocie los *inputs* con la recomendación generada, usando lenguaje natural en lugar de referencias crudas a IDs. Finalmente, el

prompt incluye instrucciones para estructurar la respuesta en formato *JSON*, con el fin de facilitar la integración con el *backend* de la aplicación.

Finalmente, este *prompt* (Bloque 6.1) se alimenta de tres estructuras de datos de entrada diseñadas para optimizar la eficiencia y el consumo de *tokens*, que dependen directamente de la cantidad de caracteres procesados. A continuación, se describen en detalle estas estructuras y el origen de los datos que las completan.

1. Dimensiones y Factores de CD

El sistema utiliza un vocabulario controlado fijo de 13 factores de CD asociados a 5 dimensiones de CD (conceptos presentados en el teórico), que actúa como principal base de conocimiento para generar las recomendaciones (par dimensión-factor de dicho subconjunto de datos). Esta estructura (Bloque 6.2) se definió siguiendo la jerarquía de los conceptos de CD (dimensión → factores), con descripciones concisas para optimizar el consumo de *tokens*.

```
dimensions_and_factors = {
    "Accuracy": {
        "semantic": "Indicates that the data is correct...",
        "factors": {
            "Semantic Accuracy": "...",
            "Syntactic Accuracy": "..."
        }
    },
    ...
    # Consistency, Completeness, Uniqueness, Freshness
}
```

Bloque 6.2: Estructura de las dimensiones y factores de CD usados en el prompt.

2. Problemas de CD

Esta estructura (Bloque 6.3) representa los problemas de CD codificados como pares id–descripción . Los datos de entrada provienen de los problemas de CD priorizados y seleccionados por el usuario para la definición del modelo de CD, en la ejecución de la actividad correspondiente en la aplicación web.

```
dq_problems = {
    1: "Null values in required fields",
    2: "Inconsistent data formats",
    ...
}
```

Bloque 6.3: Estructura de los Problemas de CD usados en el prompt.

3. Componentes de Contexto

Esta estructura (Bloque 6.4) contiene los componentes de contexto agrupados según los tipos de componentes de contexto definidos en la aplicación. Los componentes se presentan como una lista de objetos codificados con identificadores y definiciones concisas. La codificación emplea abreviaciones (como por ejemplo, *"appDomain"* en lugar de *"Application Domain")*, lo que permite optimizar el consumo de *tokens* sin comprometer la información de entrada.

```
context_components = {
    "appDomain": [{"n": "E-commerce", "id": 1}],
    "bizRule": [{"s": "Field must be unique", "id": 2}],
    ...
}
```

Bloque 6.4: Estructura de los Componentes de Contexto usados en el prompt.

Aplicación de la funcionalidad en la herramienta

Esta funcionalidad se integra directamente en la interfaz de la aplicación dedicada a la definición de dimensiones y factores de CD, la cual forma parte de la Etapa 4 enfocada a la definición del modelo de CD. La funcionalidad se ofrece como una alternativa (asistida por IA) al proceso manual, permitiendo al usuario obtener, mediante una única ejecución, un par dimensión–factor de CD recomendado y agregarlos al modelo de CD. Dicha recomendación, devuelve también los componentes de contexto y/o problemas de CD, junto a un texto explicativo con el razonamiento detrás de la recomendación, vinculando los diferentes conceptos involucrados. Un aspecto importante a aclarar es que el usuario conserva el control sobre esta recomendación, ya que puede modificarla ajustando los problemas de CD y componentes de contexto seleccionados automáticamente en la interfaz, así como agregar o eliminar factores antes de incorporarlos formalmente al modelo de CD.

Limitación identificada: Recomendaciones repetidas

Como se explicó anteriormente, el sistema opera con un conjunto de datos de entrada fijo, incluyendo un conjunto de dimensiones y factores de CD predefinidos en la herramienta, problemas de CD y componentes de contexto. Esta invariabilidad en los *inputs* puede provocar la obtención de recomendaciones repetitivas, devolviendo constantemente factores de CD más fuertemente relacionados a los componentes de contexto y problemas de CD disponibles.

Solución implementada: Filtrado del conjunto de de datos de entrada

Para abordar este problema y permitir la generación de recomendaciones más variadas, se implementó un mecanismo de filtrado previo que procesa las dimensiones y factores de CD antes generar la recomendación. El sistema combina dos enfoques complementarios:

- **Filtrado automático:** se excluye automáticamente del conjunto de entrada los factores de CD ya incluidos en el modelo de CD.
- **Filtrado manual configurable:** permite al usuario excluir explícitamente dimensiones de CD especificas y sus correspondientes factores disponibles, evitando que sean recomendados. Esta configuración se aplica previo a la confirmación de cada ejecución en la propia interfaz.

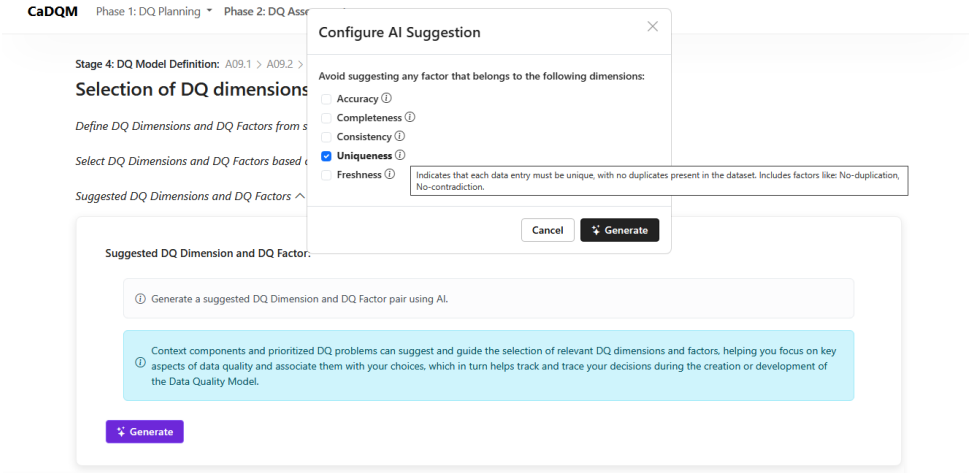

Figura 6.1: Vista de la interfaz para la configuración de filtrado personalizado previo a generar una recomendación por IA de un par dimensión-factor de CD.

Finalmente, esta solución ofrece un balance entre automatización y control manual. Por diseño, puede generar hasta N recomendaciones distintas (según factores disponibles), aunque también permite al usuario acotar estratégicamente este número, de manera personalizada.

6.2.2. Generación automática de Métodos de CD

Esta funcionalidad utiliza un modelo de lenguaje para generar automáticamente un método de CD basándose en la definición de una métrica de CD. La misma se integra a la interfaz de la aplicación, específicamente en la actividad de implementación de métodos de CD de la Etapa 4, actuando como una función de autocompletado que sugiere valores para cada campo requerido. El sistema genera un nombre descriptivo para el método, los tipos de datos de entrada y salida, y el algoritmo *SQL* que implemente la métrica de CD.

Diseño de la Solución

Para implementar esta funcionalidad, se definió un *prompt* (Bloque 6.5) que instruye al *LLM* a generar la definición del método de CD. Este *prompt* le proporciona la especificación de la métrica de CD de entrada, junto con instrucciones claras, definiciones conceptuales y ejemplos para asegurar la generación correcta del método de CD, incluyendo el algoritmo *SQL*.

```
DQ_METHOD_GENERATION_PROMPT = """Generate a data quality method based on the
    following metric. A data quality method is a process that implements a
    certain data quality metric.
Return ONLY a valid JSON object without any additional text or explanation.

Input Metric:
- Name: {name}
- Purpose: {purpose}
- Granularity: {granularity}
- Result Domain: {resultDomain}

The name of the method should reflect the nature of the metric, based on its
    purpose or granularity. For example, if the metric's purpose is to detect
     duplicates, the name could be something like detectDuplicateEntries.

The algorithm field must be an appropriate SQL query that matches the purpose
     of the metric. For example, if the metric is related to detecting
    duplicates, the SQL query should find duplicate entries.

The response must be a single JSON object with this exact structure:
{{
    "name": "generated method name",
    "inputDataType": "appropriate input type",
    "outputDataType": "appropriate output type",
    "algorithm": "SQL query for the metric",
    "implements": {metric_id}
}}

For the table and column generic names, please use table1, column1.
"""
```

Bloque 6.5: Prompt del algoritmo de generación de métodos de calidad

El *prompt* (Figura 6.5) definido presenta las siguientes características:

- Establece la diferencia entre una métrica de CD y un método CD que la implementa, lo que orienta al modelo sobre el tipo de salida esperada.
- Proporciona ejemplos de nombres de métodos de CD basados en el propósito de la métrica de CD.
- Ofrece indicaciones sobre cómo debe ser la consulta *SQL* según el propósito de la métrica de CD.
- Especifica el uso de nombres genéricos como `table1` y `column1`, buscando la generalización y reutilización del método de CD generado.
- Exige que la salida sea exclusivamente un objeto *JSON* con una estructura fija, sin incluir texto adicional, en pos de facilitar el procesamiento directo de la salida generada.

6.3. Experimentación para Recomendaciones de Dimensiones y Factores de CD

Para analizar el funcionamiento de esta funcionalidad, se implementó un *script* en *Python* que permitiera la ejecutar el algoritmo de recomendaciones utilizando distintos *LLMs*. El *script* comienza procesando un conjunto completo de entradas predefinidas (dimensiones y factores de CD, problemas de CD y componentes de contexto). En cada iteración, el modelo devuelve un único factor de CD (junto a su dimensión, los problemas de CD y el contexto que justifican la recomendación). El algoritmo, a su vez, remueve dicho factor del conjunto de conceptos de CD disponibles y solicita una nueva recomendación, repitiendo este ciclo hasta que no queden factores por recomendar.

Con el objetivo de determinar el modelo de lenguaje que ofreciera el mejor rendimiento general para su utilización en el sistema, los resultados obtenidos se analizaron en diferentes pasos según los siguientes enfoques y objetivos:

1. **Rendimiento general de los modelos de lenguaje**: Se registraron y compararon métricas de eficiencia (ejecuciones totales y uso de *tokens*) para determinar la viabilidad operativa de los *LLMs* para su inclusión en la herramienta.
2. **Análisis orden de las recomendaciones**: Se realizaron múltiples ejecuciones del *script* para cada *LLM*, analizando el orden ocurrencia de los conceptos de CD en las recomendaciones obtenidas, con el fin de detectar patrones de estabilidad y comportamientos particulares.
3. **Validación de las justificaciones de las recomendaciones**: Se estudió la coherencia de las justificaciones proporcionadas, verificando si los factores de CD recomendados estaban correctamente alineados con los problemas de CD y los componentes de contexto de entrada.

6.3.1. Rendimiento General de los Modelos de Lenguaje

Para un análisis inicial, se ejecutó el *script* de pruebas completo para cada *LLM*, el cual constó de 13 iteraciones correspondientes al número máximo de factores de CD a recomendar. Dado que el algoritmo de recomendaciones requiere una salida en un formato estructurado específico (*JSON*), se implementó un mecanismo de reintento de modo que, cuando la respuesta no cumplía con el formato requerido, el *script* generaba nuevos intentos hasta obtener una respuesta con una dimensión y factor recomendado en el formato esperado (recomendación válida).

Con el fin de observar el rendimiento general y comportamiento de cada modelo de lenguaje, en la Tabla 6.1 se presentan un conjunto de métricas operativas obtenidas a partir de la ejecución de las pruebas para los cuatro *LLMs*, enfocadas en aspectos de eficiencia (ejecuciones requeridas para obtener una recomendación válida), consumo de recursos y costo económico asociado. Estos resultados evidencian dos comportamientos claros:

- El modelo *Llama-3-8B-8192* se considera inadecuado para su uso práctico debido a su alta tasa de fallos (4.77 ejecuciones/recomendación vs. 1.0–1.07 de otros modelos), con alto costo operativo, considerando incluso mecanismos automáticos de reintento en la implementación de la solución.
- Los tres modelos de lenguaje restantes son equivalentes en eficiencia (1.0–1.07 ejecuciones), y con costos marginales por lote ($0.001-$0.013), siendo todos opciones viables hasta el momento.

	Llama-3-70B	Llama-3-8B	Llama-3.1-8B	Llama-3.3-70B
Recomendaciones válidas	13	13	13	13
Ejecuciones totales	14	62	13	13
Ejecuciones/Recomendación	1.07	4.77	1.00	1.00
Tokens totales (USD)	19,416 ($0.011 USD)	100,888 ($0.005 USD)	21,083 ($0.001 USD)	21,270 ($0.013 USD)
Tokens/Ejecución (USD)	1,386 ($0.0008 USD)	1,602 ($0.00008 USD)	1,622 ($0.00009 USD)	1,634 ($0.001 USD)
Tokens/Recomendación (USD)	1,493 ($0.0008 USD)	7,761 ($0.0004 USD)	1,622 ($0.00009 USD)	1,634 ($0.001 USD)

Tabla 6.1: Comparación de rendimiento entre modelos de lenguaje utilizados para ejecución completa de pruebas.

6.3.2. Análisis del Orden de los Factores de CD Recomendados

Tras analizar el rendimiento general de los *LLMs* en el paso anterior, el análisis siguiente se centró en estudiar el comportamiento de las recomendaciones desde una visión más amplia, comparando el orden de los factores de CD recomendados para cada modelo considerado viable (se excluyó *Llama-3-8B-8192*). Para este proceso experimental se realizaron los siguientes pasos:

- Se realizó un ciclo de cinco ejecuciones completas del *script* por cada *LLM*. Esto significó obtener las 13 recomendaciones válidas en cada una de las cinco repeticiones para analizar su orden.
- Se registró la posición de cada factor en las recomendaciones obtenidas durante las cinco ejecuciones.
- Se calcularon las siguientes métricas:
  - Orden promedio (AVG): Para identificar la posición promedio de aparición de cada factor recomendado en el ciclo de ejecuciones.
  - *Ranking* general consolidado: Orden final de relevancia que se obtiene al rankear todos los factores basándose en su Orden Promedio (AVG), proporcionando una visión general de la importancia relativa de cada factor.
  - Desviación estándar: Medida de la variabilidad del orden en que se obtiene cada factor. Junto con el AVG, permite analizar la estabilidad de las recomendaciones generadas entre las ejecuciones.

En la sección C.2.1 del Anexo, se presenta el análisis detallado de las pruebas realizadas para cada uno de los modelos viables. A partir de estos resultados, se realizó un análisis comparativo general de los resultados consolidados de estas ejecuciones presentado a continuación.

Comparativa Resultados Consolidados por Modelos de Lenguaje:

La Tabla 6.3.2 resume los resultados consolidados obtenidos al promediar las cinco ejecuciones individuales realizadas para cada uno de los tres *LLMs* viables, presentados y analizados en el Anexo (C.2.1). El Orden Promedio (AVG) y la Desviación Estándar (Desv. Est.) se utilizan para establecer el ranking general consolidado y analizar la estabilidad de las recomendaciones generadas entre los distintos modelos de lenguaje, mientras que aquellas con mejor ranking general se consideran como los factores de CD más relevantes para el contexto de los datos y los problemas de CD dado.

Al comparar dichos resultados, pudo observarse cierto consenso y estabilidad en algunos aspectos. En particular, se observó que el factor *Density* obtuvo el mejor ranking promedio y presentó la menor variabilidad entre todos los modelos, siendo consistentemente uno de los primeros factores en ser recomendados sin importar el *LLM*, lo que muestra una fuerte asociación al contexto y a los problemas de CD dados. Por el contrario, se detectó una baja relevancia constante para factores como *Syntactic*

Dimensión / Factor	3.1-8b-instant	3.3-70b-versatile	3-70b-8192	AVG	Ranking gral.	Desv. Est.
Accuracy						
Semantic Accuracy	6°	5°	2°	4.67	4°	2.52
Syntactic Accuracy	11°	10°	9°	10.00	10°	1.00
Precision	10°	6°	7°	7.67	8°	2.08
Completeness						
Density	3°	2°	3°	2.67	1°	0.47
Coverage	5°	4°	6°	5.00	6°	1.00
Freshness						
Currency	8°	13°	11°	10.67	11°	2.52
Timeliness	12°	11°	12°	11.67	12°	0.47
Volatility	13°	12°	13°	12.67	13°	0.47
Consistency						
Domain Integrity	2°	7°	10°	6.33	7°	4.04
Intra-rel. Integrity	7°	3°	4°	4.33	2°	1.53
Inter-rel. Integrity	1°	1°	12°	4.67	5°	6.23
Uniqueness						
No-duplication	4°	8°	1°	4.33	3°	3.51
No-contradiction	9°	9°	5°	7.67	9°	2.31

Tabla 6.2: Ranking general del orden de los factores de CD recomendados en las cinco ejecuciones completas del *script* de prueba para cada uno de los tres *LLMs* analizados.

Accuracy, *Timeliness* y *Volatility*, siendo recomendados en los últimos puestos para todos los modelos de lenguaje, lo que sugiere una débil vinculación con los problemas de CD o el contexto de los datos.

Otro comportamiento consistente de factores con menor relevancia fue detectado para los factores correspondientes a la dimensión *Freshness* (*Currency*, *Timeliness* y *Volatility*). Su consistente ubicación en los puestos finales sugiere que su inclusión en la recomendación final se debe a un proceso de descarte de factores, y no a una preferencia explícita. Esto indica que la dimensión *Freshness* es menos comúnmente considerada en las recomendaciones automatizadas para el contexto de datos de entrada con el que se trabajó.

Por el contrario, factores como *Inter-relationship Integrity* y *Domain Integrity* muestran comportamientos divergentes entre los modelos. El caso de *Inter-relationship Integrity* resulta particularmente llamativo: mientras fue el más importante para dos modelos (1°, 1°), pero con el modelo *Llama3-70b-8192* quedó surgió en los últimos lugares en el orden de recomendaciones (12°). También se destaca la variabilidad en *No-duplication*: el modelo *Llama3-70b-8192* lo generó como primera recomendación, a diferencia de *Llama-3.3-70b-versatile*, quedando en segundo plano (8°), mientras que con el modelo restante, promedió el cuarto puesto.

En resumen, pese a la utilización de diferentes modelos de lenguaje, fue posible obtener salidas y comportamientos generales similares en las recomendaciones generadas. Se destacó la recomendación consistente del factor *Density* (de *Completeness*) como una de las primeras posiciones, mostrando una fuerte vinculación con el contexto y los problemas de CD dados. Por el contrario, los factores de *Freshness* fueron los de más débil asociación a los datos de entrada, siendo sistemáticamente sugeridos en los últimos lugares. Sin embargo, también se identificaron casos de alta inestabilidad según el orden que fueron recomendados, como los factores *No-duplication* (4°, 8°, 1°) e *Inter-relationship Integrity* (1°, 1°, 12°), indicando tambien que las recomendaciones generadas puede variar significativamente según el modelo utilizado, incluso con datos de entrada idénticos. .

6.3.3. Validación de las Justificaciones de las Recomendaciones

Con el objetivo de analizar la pertinencia de las recomendaciones generadas por cada modelo de lenguaje, se examinó el razonamiento lógico y la fundamentación que vincula los factores de CD recomendados con los problemas de CD y los componentes de contexto utilizados, de los cuales surgieron. Para ello, se analizaron tres aspectos fundamentales:

- Uso de Problemas de CD: se examinó qué tan bien los problemas de CD utilizados respaldan los factores de CD recomendados, analizando la coherencia entre los conceptos involucrados.
- Uso de Componentes de Contexto: se analizó la selección de los componentes de contexto con el fin de sugerir el factor de CD, revisando cómo estos contribuyen a la justificación dada.
- Justificación: se examinó el razonamiento lógico incluido en la salida generada por el modelo, mediante un texto explicativo, analizando cómo se integraron los problemas de CD y los componentes de contexto en dicha fundamentación para recomendar el factor de CD.

Para este análisis, se seleccionaron dos factores que representan casos opuestos en el orden de relevancia de las recomendaciones. Por un lado, se analizó un factor que apareció con alta frecuencia en las primeras posiciones del ranking *(Density)*. El objetivo consistía en analizar la adecuación de las justificaciones y observar si los modelos se basaron en conceptos similares, o no, para generar dicha recomendación. Por otro lado, se examinó un factor que se ubicó sistemáticamente en los últimos puestos *(Currency)*, con el fin de analizar la calidad argumentativa de estas recomendaciones y determinar si son adecuadas o si, por el contrario, son producto de una recomendación forzada por descarte. Para la ejecución de este análisis se utilizó un modelo de contexto y un conjunto de problemas de CD definidos manualmente, compartidos como parte del Caso de Estudio 1, presentado en el Anexo D.

Métrica de CD Recomendada: *Density (Completeness)*

El factor *Density* se ubicó consistentemente entre los primeros puestos en las recomendaciones de los tres modelos, aunque cada uno justificó su elección de manera distinta. Para analizar estas diferencias, se presentan en la Tabla C.5 los problemas de CD y componentes de contexto que cada modelo de lenguaje utilizó para sugerir dicho factor de CD.

Density (Completeness)	**3-70b-8192**	**3.3-70b-versatile**	**3.1-8b-instant**
Problemas de CD			
PC1: Valores nulos en `reviewText` o `title`	✓	✓	
PC7: Reseñas duplicadas			✓
PC13: Entradas redundantes			✓
PC14: Relaciones faltantes			✓
Componentes de Contexto			
AD1: Comercio electrónico de libros y reseñas en Amazon		✓	✓
BR2: `(title, userId)` debe ser único		✓	✓
DF1: Excluir reseñas con menos de 20 caracteres		✓	✓
DF2: Incluir solo reseñas en inglés desde 2018		✓	✓
DQR2: Completitud $\geq$95 % en campos obligatorios	✓	✓	✓
T1: Calcular *sentiment scores*	✓	✓	✓
T2: Detectar reseñas sospechosas		✓	✓
T3: Generar ranking de calidad textual		✓	
UT1: Analistas de marketing		✓	✓
UT2: Científicos de datos		✓	✓
UT3: Investigadores académicos		✓	

Tabla 6.3: Problemas de CD y componentes de contexto utilizados por modelo para generar la recomendación del factor *Density*.

Análisis Detallado: Modelo *3.3-70b-versatile*

- **Uso de Problemas de CD**: El modelo utiliza adecuadamente el factor *Density* con el problema de valores nulos en `reviewText` y `title` (PC1), reconociendo que estos nulos afectan la cantidad de datos completos disponibles para el análisis.
- **Uso de Componentes de Contexto**: La recomendación surge de una amplia cobertura de los componentes disponibles. Entre ellos, se apoyó adecuadamente en el requerimiento de CD (DQR2), que establece un umbral de completitud mínima del $\geq$95 % en los campos obligatorios, lo que permite asociar directamente este componente con el factor *Density*. Si bien componentes como el dominio de aplicación (AD1) y las tareas de cálculo de *sentiment scores* o generación de *rankings* (T1, T2, T3) no explican la elección del factor *Density* por sí solos, la recomendación se sustenta en que estas tareas dependen de datos completos para asegurar el análisis y la toma de decisiones. Por otro lado, la recomendación también incluye componentes de filtrado de datos (DF1, DF2) que no muestran una relación explícita con el factor o la dimensión recomendada.
- **Justificación**:

  > "La dimensión de Completitud, específicamente el factor de Densidad, es crucial en este contexto, ya que impacta directamente la disponibilidad de los datos necesarios para el análisis y la toma de decisiones. Los valores nulos en campos requeridos y en campos clave indican brechas en los datos, afectando su densidad. El dominio de aplicación de *e-commerce* y reseñas de clientes depende en gran medida de datos completos para asegurar análisis y recomendaciones precisas. Las reglas de negocio, como evitar múltiples reseñas por usuario y por libro, y los filtros de datos, como incluir solo reseñas en inglés publicadas desde 2018, enfatizan aún más la necesidad de datos densos y completos. El requisito de una métrica de completitud de al menos el 95 % para los campos obligatorios también respalda la importancia de la densidad. Por lo tanto, asegurar una alta densidad de datos es esencial para la integridad y fiabilidad de los procesos basados en datos en este contexto de *e-commerce* y análisis de reseñas."

  El texto explicativo devuelto integra todos los problemas de CD y componentes de contexto utilizados, siendo claro al relacionar el factor *Density* con el problema de CD considerado (PC1) y fundamenta adecuadamente la métrica de completitud mínima (DQR2) asociada al factor. La explicación sobre componentes de filtrado de datos (DF1, DF2) y algunas tareas presenta argumentos menos directos y no necesariamente relevantes para el factor.

Los análisis detallados de las recomendaciones generadas con los modelos *3-70b-8192* y *3.1-8b-instant* para el factor *Density* se encuentran en el Anexo (C.2.2). En base a los resultados observados para los tres *LLMs*, se presenta la siguiente síntesis comparativa:

Síntesis Comparativa

- ***3.3-70b-versatile***: realiza un uso adecuado de los problemas de CD y una amplia cobertura de componentes de contexto para generar la recomendación del factor. Si bien incluye el componente más relevante y directamente relacionado, también considera otros cuya vinculación es más indirecta. El texto explicativo busca establecer conexiones entre los conceptos de forma más elaborada, integrando los diferentes componentes utilizados en la justificación, y logra en conjunto una argumentación válida y coherente, aunque menos concisa y puntual.
- ***3-70b-8192***: muestra un uso preciso y adecuado tanto de los problemas de CD como de los componentes de contexto, limitándose a los elementos directamente relacionados con el factor. Su justificación se considera clara y conceptualmente correcta, argumentando de manera valida la recomendación del factor.
- ***3.1-8b-instant***: utiliza una mayor cantidad de problemas de CD, incluyendo el de valores nulos, pero también otros cuya vinculación con el concepto de densidad es débil o superficial. Además, realiza una cobertura amplia de componentes de contexto, aunque sin explicar adecuadamente su inclusión ni su relación con el razonamiento del modelo. En conjunto, la explicación muestra cierta confusión conceptual y una argumentación débil en cuanto a la justificación del uso de los componentes y problemas seleccionados para sustentar la sugerencia de dicho factor.

En resumen, tanto el modelo *3.3-70b-versatile* como el *3-70b-8192* presentan recomendaciones respaldadas por una argumentación correcta, identificando de manera adecuada los problemas de CD y componentes de contexto más relacionados con el factor. El modelo *versatile* ofrece una explicación más amplia y detallada, mientras que el *3-70b-8192* plantea un razonamiento más puntual y directo.

Métrica de CD Recomendada: *Currency (Freshness)*

El factor *Currency*, correspondiente a la dimensión *Freshness*, fue consistentemente sugerido en las últimas posiciones del ranking general de recomendaciones generadas por los distintos modelos. Este comportamiento motivó el análisis de sus respectivas salidas para evaluar si la inclusión del factor respondió a una justificación sólida o si, por el contrario, se trató de una recomendación residual, posiblemente derivada de un proceso de descarte. La Tabla C.6 resume los problemas de CD y los componentes de contexto que cada modelo utilizó para fundamentar esta recomendación.

Currency (Freshness)	**3-70b-8192**	**3.3-70b-versatile**	**3.1-8b-instant**
Problemas de CD			
PC15: Timestamps faltantes	✓	✓	✓
PC16: Valores fuera de límites establecidos			✓
PC17: Valores inconsistentes entre fuentes			✓
Componentes de Contexto			
AD1: Comercio electrónico de libros impresos y digitales en Amazon		✓	✓
BR1: Validación de rango 1-5 para score		✓	✓
DF2: Incluir solo reseñas en inglés desde 2018		✓	✓
DQR2: Completitud mínima 99 % en campos obligatorios	✓	✓	✓
T1: Cálculo de *sentiment scores*		✓	✓
T2: Detección y filtrado de reseñas sospechosas		✓	✓
UT1: Analistas de marketing		✓	✓
UT2: Científicos de datos		✓	✓
UT3: Investigadores académicos		✓	✓

Tabla 6.4: Problemas de CD y componentes de contexto utilizados por modelo para generar la recomendación del factor *Currency*.

Análisis Detallado: Modelo *3.3-70b-versatile*

- **Uso de Problemas de CD**: El modelo asocia el factor *Currency* con el problema de marcas de tiempo faltantes (PC15), vinculando la ausencia de `timestamps` con la capacidad de determinar la actualidad de los datos.
- **Uso de Componentes de Contexto**: La recomendación utiliza múltiples componentes de contexto. Entre ellos, el componente de filtrado de datos que restringe las reseñas a partir de 2018 (DF2), aporta una conexión explícita con la necesidad de datos actualizados, en línea con el factor *Currency*. También se incluyen el dominio de aplicación (AD1) y los diferentes tipos de usuario (UT1, UT2, UT2), que podrían implicar una necesidad indirecta de vigencia temporal. Sin embargo, se considera que las tareas T1 y T2, así como el requerimiento DQR2, no presentan una relación clara con el concepto de CD, y su inclusión no contribuye directamente a justificar la recomendación.
- **Justificación**:

  > "La dimensión y el factor recomendados están bien respaldados por la presencia de marcas de tiempo faltantes, lo que se relaciona directamente con la actualidad de los datos. El dominio de aplicación de *e-commerce* y reseñas de clientes en Amazon, donde la frescura de los datos es crucial para el análisis en tiempo real y la toma de decisiones, apoya aún más esta recomendación. Además, el filtro de datos que incluye solo reseñas en inglés publicadas desde 2018 y el requisito de baja latencia y altas métricas de completitud también enfatizan la importancia de tener datos actualizados."

  La justificación devuelta establece una conexión válida entre el factor *Currency* y la falta de marcas de tiempo (PC15), destacando su impacto en la vigencia de los datos. Luego, si bien

intenta incorporar los demás componentes utilizados al texto explicativo, la argumentación de los mismos introduce ruido, intentando justificar la inclusión de estos en escenarios de necesidad de actualidad de los datos, en lugar de suponer una relación directa.

Los análisis detallados de las recomendaciones generadas con los modelos *3-70b-8192* y *3.1-8b-instant* para el factor *Currency* se encuentran en el Anexo (C.2.2). En base a los resultados observados para los tres *LLMs*, se presenta la siguiente síntesis comparativa:

Síntesis Comparativa

- ***3.3-70b-versatile***: ofrece una argumentación coherente, al asociar la falta de marcas de tiempo con la actualidad de los datos y al utilizar el componente de contexto de filtrado de datos, que incluye solo reseñas en inglés desde 2018, directamente relacionado con la actualidad de los datos. Por otro lado, la inclusión de otros componentes de contexto sin relación directa con el factor introduce cierto ruido en la justificación.
- ***3-70b-8192***: identifica correctamente el problema de CD asociado con la actualidad de los datos, pero no utiliza el componente de contexto clave para la sugerencia del factor recomendado, relacionado con las fechas de las reseñas. Además incluye problemas de CD y componentes de contexto adicionales sin conexión clara con el factor ni con la dimensión *Freshness*, mezclando incluso nociones de integridad y completitud.
- ***3.1-8b-instant***: se basa en los mismos componentes de contexto que el modelo *versatile* y combina problemas de CD con fundamentos dispares, lo que, sumado a una explicación confusa que mezcla conceptos de integridad y respuesta en tiempo real, muestra una comprensión parcial del factor *Currency*.

En resumen, el modelo *versatile* surge como la mejor opción al presentar la justificación más adecuada, logrando una asociación razonable entre el factor y la actualidad de los datos, aunque con algunos desvíos conceptuales menores. En cambio, las versiones *70b-8192* e *instant* muestran un desempeño menos preciso, con argumentaciones más débiles y relaciones poco claras entre los problemas y los componentes de contexto utilizados.

Conclusiones Generales

El análisis comparativo de las justificaciones generadas por los tres modelos de lenguaje para las recomendaciones de los factores de CD *Density* y *Currency* permitió observar comportamientos diferentes en la forma en que cada modelo construyó y fundamentó dichas recomendaciones, en función de la información disponible del modelo de contexto y los problemas de CD considerados.

En el caso del factor *Density*, representativo entre aquellos que surgieron con mayor frecuencia en las primeras posiciones del orden de recomendaciones, los modelos mostraron una buena capacidad para identificar los problemas de CD y los componentes de contexto más pertinentes para sugerir el factor. Si bien el nivel de detalle y extensión del razonamiento varió entre modelos, el *3.3-70b-versatile* y el *3-70b-8192* lograron establecer vínculos conceptuales sólidos que respaldaron adecuadamente la recomendación, mientras que el *3.1-8b-instant* presentó una justificación confusa.

Para el factor *Currency*, representativo entre aquellos que aparecieron de forma consistente en los últimos puestos del ranking de recomendaciones, los modelos mostraron un desempeño menos preciso en la justificación de sus sugerencias. Si bien, el modelo *3.3-70b-versatile* logró proporcionar una recomendación con una justificación razonable, en general, las argumentaciones tendieron a incluir asociaciones débiles o poco pertinentes entre los componentes de contexto y los problemas de CD utilizados, y el factor mismo, evidenciando dificultades para captar adecuadamente la noción de actualidad de los datos a partir de la información disponible.

En conjunto, considerando el comportamiento observado y los resultados analizados para dichos factores de CD, el modelo *3.3-70b-versatile* se destacó como la opción más confiable. En términos generales,

mostró una mejor capacidad para vincular los factores de CD con la información disponible, generando justificaciones más adecuadas que las presentadas por los modelos *3-70b-8192* y *3.1-8b-instant*

En síntesis, los resultados permitieron identificar que:

- El orden en que los factores surgieron en las recomendaciones se relacionó con la calidad de las justificaciones, ya que los modelos tendieron a elaborar respuestas más completas cuando el factor ocupó posiciones iniciales.
- El rendimiento mostró una cierta disminución al abordar factores ubicados en posiciones finales, aunque modelos como el *3.3-70b-versatile* mantuvieron un nivel razonable de coherencia incluso en esos casos.
- La elección del modelo resultó relevante, dado que se observaron variaciones significativas en profundidad técnica y manejo contextual entre los modelos analizados.

Finalmente, a partir de este análisis, se decidió utilizar el modelo ***3.3-70b-versatile*** para la implementación de esta funcionalidad en la herramienta.

6.4. Experimentación para Generación Automática de Métodos de CD

Para validar la utilidad y coherencia de las salidas generadas por la funcionalidad de generación automática de métodos de CD, se llevó a cabo un análisis basado en los resultados obtenidos a partir de un conjunto de métricas de CD definidas. Este análisis buscó examinar el comportamiento del algoritmo generador centrándose principalmente en los siguientes dos aspectos:

- Verificar que el método de CD generado aborda correctamente el propósito de la métrica de CD.
- Analizar si el código *SQL* generado es fácilmente adaptable a un *dataset* específico.

Cabe destacar que los métodos de CD generados no pretenden ser implementaciones finales, sino plantillas sugeridas que proporcionan un algoritmo parametrizable en forma de consulta *SQL*. Estas plantillas buscan ofrecer un punto de partida técnico para el usuario, utilizando parámetros genéricos (`table1`, `column1`) para permitir una mayor generalidad y reutilización. Por lo tanto, requieren ajustes posteriores para convertirse en métodos aplicados, asociándose a las tablas y columnas específicas del esquema de datos real para los cuales se ejecutarán las mediciones de CD.

Proceso de Análisis y Validación

Se implementó un *script* para ejecutar múltiples pruebas del algoritmo generador de métodos de CD, utilizando como entrada un conjunto de métricas de CD típicas. Para cada métrica de CD:

- Se realizaron 10 ejecuciones por métrica de CD.
- Se analizó la estabilidad de los resultados (variación entre ejecuciones).
- Se examinó el código *SQL* generado según su aplicabilidad real sustituyendo los parámetros genéricos por elementos de un esquema concreto (*dataset Amazon Reviews* [1]).

La validación práctica de las consultas *SQL* obtenidas se realizó utilizando *PostgreSQL* como motor de base de datos. Las pruebas se ejecutaron sobre la BD correspondiente al *dataset Amazon Reviews*, sustituyendo los parámetros genéricos de las consultas *SQL* generadas, por tablas y columnas específicas.

LLM seleccionado: *Llama-3.3-70B-Versatile*

Para esta funcionalidad de generación automática de métodos, se seleccionó el modelo *3.3-70b-versatile*, y todas las pruebas de validación se realizaron con esta versión.

A diferencia del proceso exhaustivo realizado para seleccionar el *LLM* más adecuado en las recomendaciones de factores de CD, en este caso la elección del modelo se basó en su idoneidad para tareas técnicas y estructuradas frente a los otros modelos, respaldada además por su desempeño superior mostrado en la experimentación de las recomendaciones. Sus características técnicas oficiales, resumidas en la Tabla 6.5, muestran que se destacó particularmente en los siguientes aspectos:

- Presenta la mayor precisión en pruebas de generación de código (88.4 % en *HumanEval*) y en tareas de lógica general (86 % en *MMLU*), lo que resulta especialmente adecuado para el caso de uso de generación de plantillas *SQL*.
- Ofrece un balance costo-beneficio favorable. Si bien su costo por 1K *tokens* es el más alto entre los modelos comparados, el costo por generar un método de CD es marginal (aproximadamente $0.00021 por ejecución), y su mayor precisión podría contribuir a reducir errores y retrabajos, haciendo viable un uso intensivo de la funcionalidad.

En conjunto, estas características respaldan la elección del modelo *3.3-70b-versatile* para la generación automática de métodos de CD.

Criterio	Llama-3.3-70B-Versatile	Llama3-70B-8192	Llama-3.1-8b-instant
Precisión (MMLU)	86.0 %	79.5 %	69.4 %
Generación de código (HumanEval)	88.4 %	81.7 %	72.6 %
Costo por 1K *tokens*	$0.59	$0.55	$0.15
Capacidad de contexto (*tokens*)	8192	8192	2048

Tabla 6.5: Comparativa aspectos técnicos entre *LLMs* considerados [14, 16, 17].

6.4.1. Métodos de CD Generados por Métrica de CD

Para analizar el comportamiento del algoritmo en diferentes escenarios, se aplicó la prueba de 10 ejecuciones a tres métricas de CD con distintas granularidades y propósitos. Este análisis no pretende ser concluyente sobre el rendimiento general de la funcionalidad implementada, sino explorar la variabilidad de la salida del algoritmo a lo largo de diferentes ejecuciones, y determinar si los algoritmos devueltos son correctos y funcionales. Dicha funcionalidad fue validada mediante pruebas prácticas, instanciando los parámetros genéricos del código *SQL* para el *dataset* especifico correspondiente (*Amazon Reviews*).

A continuación, se presenta el análisis detallado de los métodos de CD generados a partir de la definición de la métrica de CD *Null Value Ratio* como ejemplo representativo. Los resultados y el análisis para las otras métricas de CD estudiadas se encuentran en el Anexo (C.3.1).

Métrica de CD: *Null Value Ratio*

El algoritmo recibe como entrada la definición estructurada de la métrica de CD, compuesta por los siguientes atributos:

Propósito: Medir porcentaje de valores nulos en columna.
Granularidad: Columna.
Dominio de resultado: Porcentaje entre [0,1].

Salidas Generadas

De las 10 ejecuciones realizadas, se obtuvo 3 variantes de métodos de CD que se agruparon según el código *SQL* generado:

- **Variante Principal (7/10):** En el Bloque 6.6 se presenta el método de CD generado para esta variante principal.

```
{
    "name": "measureColumnNullValueRatio",
    "inputDataType": "table",
    "outputDataType": "float",
    "algorithm": "SELECT (COUNT(CASE WHEN column1 IS NULL THEN 1 END) * 1.0) /
    COUNT(*) AS null_ratio FROM table1"
}
```

Bloque 6.6: Método de CD generado – Variante principal (70 %)

- **Variante Secundaria (2/10):** En el Bloque 6.7 se muestra el método de CD generado para esta última variante.

```
{
    "name": "measureNullValueRatioPerColumn",
    "inputDataType": "table",
    "outputDataType": "float",
    "algorithm": "SELECT (COUNT(CASE WHEN column1 IS NULL THEN 1 END) * 1.0) /
                 COUNT(column1) AS null_value_ratio FROM table1"
}
```

Bloque 6.7: Método de CD generado – Variante secundaria (20 %)

- **Variante Alternativa (1/10):** En el Bloque 6.8 se muestra el método de CD generado para esta última variante.

```
    {
        "name": "measureColumnNullValueRatio",
        "inputDataType": "table",
        "outputDataType": "float",
        "algorithm": "SELECT CAST(SUM(CASE WHEN column1 IS NULL THEN 1 ELSE 0 END)
                     AS FLOAT) / COUNT(column1) AS null_value_ratio FROM table1"
    }

```

Bloque 6.8: Método de CD generado – Variante alternativa (10 %)

Validación Práctica
Adaptación para esquema real (*dataset Amazon Reviews*):

- `table1` → `books_rating`
- `column1` → `profilename`

A partir de esta sustitución de parámetros por elementos específicos del *dataset* a medir, se ejecutaron las variantes de consultas generadas para la métrica de CD, resumiendo los resultados obtenidos en la Tabla 6.4.1.

Consulta SQL (Método Aplicado)	Resultado	Tiempo	Correctitud

Variante Principal:
`SELECT`
`  (COUNT(CASE WHEN profilename`
`  IS NULL THEN 1 END)*1.0)`
`  /COUNT(*)`
`FROM books_rating`
	18.73 % (Ratio NULLs)	00:01:36.656	✓

Variante Secundaria:
`SELECT`
`  (COUNT(CASE WHEN profilename`
`  IS NULL THEN 1 END)*1.0)`
`  /COUNT(profilename)`
`FROM books_rating`
	23.04 % (Ratio sesgado)	00:01:29.956	×

Variante Alternativa:
`SELECT`
`  CAST(SUM(CASE WHEN profilename`
`  IS NULL THEN 1 ELSE 0 END)`
`  AS FLOAT)/COUNT(profilename)`
`FROM books_rating`
	23.04 % (Ratio sesgado)	00:01:12.877	×

Tabla 6.6: Resultados de ejecución de los métodos de CD generados para la métrica *Null Value Ratio*

Análisis de las Variantes *SQL* Generadas:

- La Variante Principal (`COUNT(*)`) calcula el porcentaje respecto al total de filas.
- Las Variantes Secundaria y Alternativa (`COUNT(column)`) excluyen NULLs del denominador, subestimando la proporción real.

Esta diferencia en el diseño de la consulta impacta directamente en los resultados obtenidos. Por ejemplo, considerando una tabla con 3,000,000 filas y 561,905 valores NULL, los cálculos serían:

- Variante Principal: 561905/3000000 = 0.1873 (correcto)
- Variantes Secundaria/Alternativa: 561905/(3000000-561905) = 0.2304 (sesgado)

Como se observa también en la Tabla 6.4.1, esta diferencia en la formulación de la consulta produce una desviación significativa en el ratio obtenido, confirmando que el uso de `COUNT(*)` ofrece una medición más fiel al propósito de la métrica.

Conclusiones:

- La Variante Principal (`COUNT(*)`) resulta la más adecuada para medir valores nulos, ya que calcula correctamente el porcentaje sobre el total de filas y, además, fue la más generada (70
- Las Variantes Secundaria y Alternativa (`COUNT(column)`) comparten una estructura algorítmica similar, pero su diferencia clave (excluir los valores `NULL` del denominador) las vuelve inadecuadas para evaluar la métrica de CD, ya que distorsionan el resultado real al subestimar la proporción de nulos.

Conclusiones Generales

A partir del análisis de los resultados obtenidos al aplicar la funcionalidad de generación automática de métodos de CD sobre tres métricas de CD específicas (*Null Value Ratio*, *Valid Score Range* y *Secure URLs Ratio*, estas últimas dos documentadas en el Anexo C.3.1), se observó que las sugerencias generadas lograron, en general, resultados alineados con los propósitos planteados para cada métrica de CD.

En todos los casos, los métodos de CD generados resultaron adecuados y funcionales, reproduciendo correctamente la lógica de medición esperada, siendo necesario ajustes menores para su adaptación al esquema real de los datos y su ejecución. De esta manera, los resultados obtenidos en la experimentación realizada mostraron que las plantillas parametrizables de métodos de CD generadas constituyeron una base sólida y reutilizable sobre la cual se pudieron construir y refinar las implementaciones concretas correspondientes al conjunto de métricas de CD analizadas.

Capítulo 7

Casos de Estudio

En este capítulo se presentan dos casos de estudio diseñados para validar la herramienta implementada. El primero se enfoca en verificar la correctitud de las funcionalidades, y fue ejecutado durante el desarrollo de la herramienta. El segundo caso de estudio, se centra en verificar la interoperabilidad entre las herramientas que implementan la Fase 1 y la Fase 2 de CaDQM, al mismo tiempo que analiza la influencia del contexto en la construcción de un modelo de CD a partir de un análisis comparativo con un modelo de CD de referencia, obtenido por expertos en CD mediante la ejecución manual de CaDQM.

7.1. Caso de Estudio 1: Funcionalidad

El propósito de este caso de estudio es demostrar la capacidad de la herramienta implementada para ofrecer un soporte completo para llevar a cabo la ejecución de la Fase 2 -*DQ Assessment* de CaDQM. Para lograrlo, se establecieron los siguientes objetivos específicos:

Objetivos

- Demostrar la capacidad de la herramienta para guiar al usuario en la construcción de un modelo de CD fundamentado en el contexto y los problemas de CD.
- Evaluar la utilidad de las funcionalidades de IA, verificando que actúen como apoyo para la definición del modelo de CD.
- Verificar el correcto funcionamiento de la herramienta en el proceso de medición y evaluación de CD, demostrando su capacidad para ejecutar los métodos de CD definidos, manejar umbrales personalizados y almacenar los resultados en la base de datos de *DQ Metadata*.

Alcance

- Configuración del *data at hand* mediante la carga del *dataset* seleccionado a una BD relacional *(PostgreSQL)* para establecer la conexión.
- Definición manual de un modelo de contexto y de diferentes problemas de CD, a partir de un análisis del *dataset* a evaluar.
- Ejecución de la Etapa 4 de CaDQM para construir el modelo de CD completo, en base al modelo de contexto y los problemas de CD definidos. La Etapa 4, también incluye la definición de métricas con el fin de cubrir diferentes granularidades y tipos de resultados, así como la implementación de métodos de CD como plantillas *SQL* parametrizables, a ser instanciadas en los métodos de CD aplicados.
- Ejecución de la Etapa 5 CaDQM para la medición de la CD a través de la ejecución de todos los métodos de CD aplicados.

- Ejecución de la Etapa 6 de CaDQM para la evaluación de la CD mediante la definición de umbrales (rangos de aceptación para valores de CD) y ejecución de dicha evaluación para todos los métodos de CD aplicados.

7.1.1. Conjunto de Datos

- Se emplea un *dataset* público sobre libros de Amazon y sus reseñas, disponible en la plataforma *Kaggle*[1].
- El conjunto contiene dos tablas: una con información bibliográfica y otra con las reseñas correspondientes.
- El esquema detallado de las tablas se presenta en la Figura 7.1.

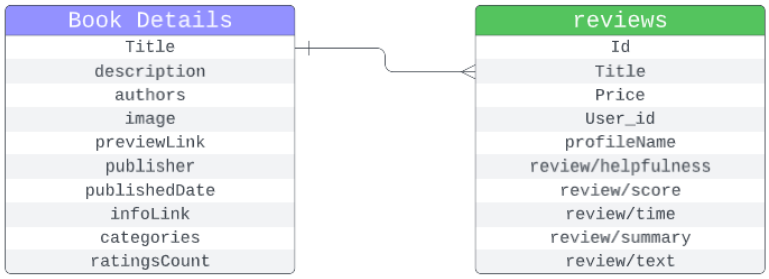

Figura 7.1: Esquema del *dataset* sobre libros y reseñas.

7.1.2. Modelo de Contexto

Para la ejecución de este caso de estudio, fue necesario definir un modelo de contexto de forma manual, por ser una de las entradas de la Etapa 4, y necesario para la definición del modelo de CD utilizando la herramienta.

Para ello, se realizó un análisis del *dataset* para identificar problemas de CD y componentes de contexto que sirvieran como insumos para la construcción del modelo de CD. El modelo de contexto(conjunto de componentes de contexto) y los problemas de CD definidos se encuentran en el Anexo D.

7.1.3. Construcción del Modelo de CD

Inicialmente, dado que la aplicación web implementada no incluye una interfaz para esta tarea, tanto los componentes de contexto como los problemas de CD, se cargaron manualmente en la base de datos a través de la vista de la *API* proporcionada por el *framework* del *backend* utilizado. El proyecto creado se inició automáticamente con el estado `TO_DO` para la Etapa 4, y se asoció al modelo de contexto y al *dataset* definidos.

Una vez dadas esas condiciones, se pudo dar comienzo formal a la ejecución de la Etapa 4 utilizando íntegramente la interfaz de la herramienta. El primer paso consistió en la creación de un modelo de CD vacío, el cual se asoció automáticamente al proyecto, provocando un cambio de estado de `TO_DO` a `IN_PROGRESS` para la Etapa 4. Una vez replicados todos los atributos necesarios, fue posible definir el modelo de CD en función del contexto asociado.

[1] https://www.kaggle.com/datasets/mohamedbakhet/amazon-books-reviews

Priorización y Selección de Problemas de CD

La primera actividad realizada en la Etapa 4 consistió en priorizar y seleccionar los problemas de CD que sirvieron de guía para el proceso de definición del modelo de CD. Esta priorización se llevó a cabo a partir de una valoración sobre la criticidad de cada problema de CD, considerando su potencial uso en la construcción del modelo de CD. Para este caso, una vez priorizados todos los problemas de CD, se seleccionaron aquellos de prioridad alta *"High"*, como se muestra en la Fig. 7.2, con el fin de asegurar que el modelo de CD se enfocara en los aspectos más significativos de la CD.

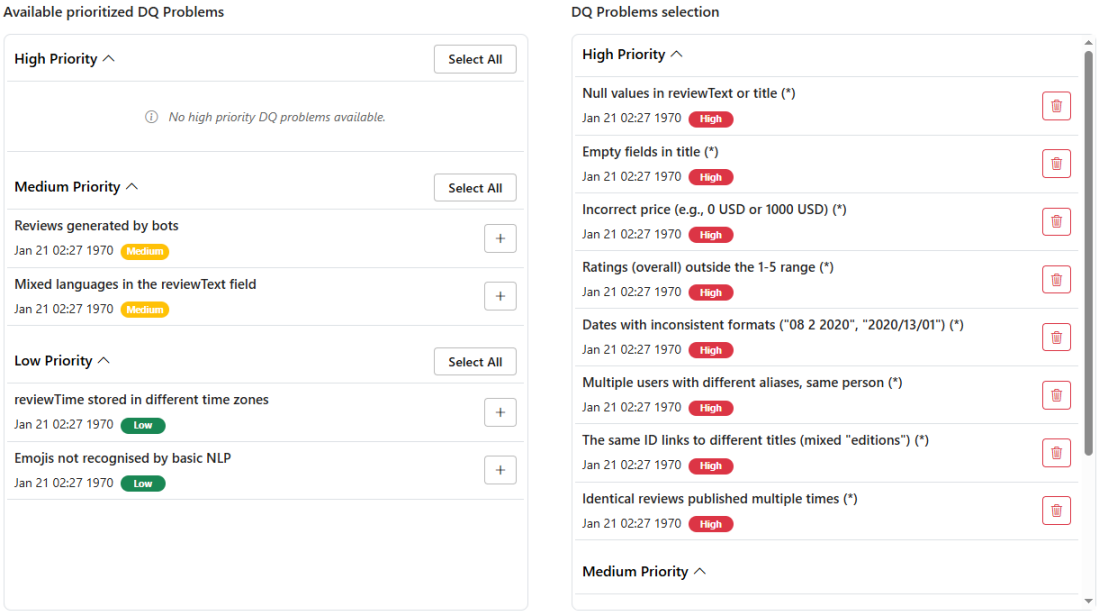

Figura 7.2: Problemas de CD priorizados seleccionados en el caso de estudio 1, a través de la interfaz de la herramienta.

Selección de Dimensiones y Factores de CD

La definición de las dimensiones y factores de CD representa una de las tareas centrales de la Etapa 4 de CaDQM. Para este caso de estudio, se utilizaron todas las funcionalidades de la herramienta diseñadas para esta actividad, con el objetivo de validar su correcto funcionamiento.

- Definición a partir de Problemas de CD: Por un lado, se utilizó la interfaz de selección manual para inspeccionar las dimensiones y factores precargados en la aplicación (presentados en el marco teórico en la sección 2.1), para seleccionar aquellos que se ajusten al contexto del caso de estudio. De manera análoga, se seleccionaron factores precargados para las dimensiones agregadas. Además, se probó la funcionalidad de creación de nuevos conceptos de CD mediante la interfaz, creando la dimensión *Credibility* y los factores asociados, que no existían previamente en la aplicación.

  Por otro lado, se utilizó una funcionalidad alternativa para la definición de dimensiones y factores. Esta permite iniciar el flujo de trabajo desde un problema de CD y a partir de este, identificar los factores y dimensiones que se asocian. Por ejemplo, a partir del problema PC6: "Identificadores que enlazan a títulos distintos (ediciones mezcladas)", se identificó la relevancia de los factores *Intra-relationship Integrity* (para la dimensión *Consistency*) y *No-duplication* (para la dimensión *Uniqueness*).

- Motor de recomendaciones (IA): Como parte de la validación de la herramienta, se utilizó la funcionalidad de IA para asistir en la definición de dimensiones y factores. Aunque esta funcionalidad permite al usuario ajustar las sugerencias generadas (por ejemplo, removiendo o añadiendo

componentes de contexto y problemas de CD asociados), en este caso no fue necesario realizar modificaciones. La propuesta de la IA resultó ser precisa y coherente con las definiciones previamente establecidas. Mediante esta funcionalidad, se crearon los siguientes conceptos de CD:

- *Density* como factor de la dimensión *Completeness*. Los componentes de contexto y problemas de CD asociados son PC1, PC2 y DQR1, todos relacionados con la necesidad de atributos sin valores nulos.
- *Semantic Accuracy* como factor dentro de la dimensión *Accuracy*. Este incluyó los problemas de CD: PC3 y PC5, que hacen referencia a valores de atributos que están dentro de lo esperado.

A continuación, se presenta una tabla que resume las dimensiones y los factores del modelo de CD, y los vincula a los componentes de contexto y problemas de CD que los surgieron.

Dimensión	**Factor**	**Problemas de CD / Componentes de Contexto asociados**
Accuracy	*Semantic Accuracy*	PC3, PC5
	Syntactic Accuracy	UT1, T3
Completeness	*Density*	PC1, PC2, DQR2
	Coverage	DQR1
Consistency	*Intra-relationship Integrity*	PC4, PC3, PC6
Uniqueness	*No-duplication*	PC6, PC8
Freshness	*Currency*	
Credibility	*Check Users Name*	DFR1, PC8
	Valid Review Text	DFR1, PC9, PC12, PC11
	Check Valid Description	T1, PC1, PC2

Tabla 7.1: Dimensiones y factores del modelo de CD, vinculados a componentes de contexto y problemas de CD. Mas detalles pueden ser consultados en el Anexo D.

Definición de Métricas de CD

La primera actividad consistió en definir las métricas que se utilizarían para cuantificar la CD. Estas métricas fueron concebidas para cada factor del modelo de CD, asociando a cada una los componentes de contexto que justificaban su inclusión. Se definieron métricas de distintas granularidades. Aquellas a nivel de columna o tabla se diseñaron para obtener proporciones o porcentajes (con un dominio entre 0 y 1), mientras que las de granularidad a nivel de fila o celda se definieron para obtener un resultado de dominio booleano que indicara el cumplimiento de una determina condición.

Implementación Métodos de CD

Una vez definidas las métricas, para cada una se crearon los métodos de CD que implementan la lógica de los algoritmos de medición. Para esta tarea, se utilizó la funcionalidad de generación automática de métodos con IA, la cual se encarga de autocompletar los campos del método a partir de la definición de la métrica.

Esta funcionalidad fue utilizada particularmente para las métricas asociadas a los factores de la dimensión *Credibility*, resultando la implementación sugerida adecuada para su inclusión en el modelo de CD. Por ejemplo, el algoritmo generado para el método relacionado a la métrica *CheckReviewByValidUser* fue el siguiente: `SELECT column1 FROM table1 WHERE LENGTH(column1) != 0`. Esta lógica es coherente con el objetivo de la métrica de validar que los usuarios que hicieron reseñas son válidos.

Cabe mencionar que en este paso, antes de agregar un método al modelo de CD, también se realizó la asociación de los componentes de contexto relacionados, seleccionando exclusivamente aquellos componentes de contexto vinculados a la métrica.

Métodos de CD Aplicados: Como parte de la misma actividad, se definieron los métodos aplicados. Dado que los métodos definidos sirvieron como una plantilla *SQL* parametrizable, los métodos aplicados son una instanciación de dichos métodos, donde se especifica el nombre de los atributos y tablas. Esta información surge de los requerimientos de CD o las reglas de negocio incluidas en los componentes de contexto del método, lo que define la consulta *SQL* final para su posterior ejecución.

En la Tabla 7.2 se presenta un resumen de las métricas, los métodos y los métodos de CD aplicados.

Confirmación del Modelo de CD

Una vez definidos e incluidos todos los conceptos de CD, asegurando una jerarquía completa desde cada dimensión hasta el método aplicado, se confirmó manualmente la finalización del modelo de CD construido, concluyendo así la ejecución de la Etapa 4 - *DQ Assessment* de la herramienta. Además, se generó un archivo descargable en formato *PDF* que contiene el modelo de CD completo, siguiendo la jerarquía de los conceptos de CD incluidos y los componentes de contexto y problemas de CD que justifican cada inclusión.

7.1.4. Medición de CD

Una vez finalizada la definición del modelo de CD, se procedió a la ejecución de la Etapa 5 - *DQ Measurement* de CaDQM, correspondiente a la medición de la calidad de los datos del *dataset* utilizado.

Este proceso se inició a través de la inicialización explicita de la medición para el modelo de CD, en la interfaz de la herramienta. Esto habilitó la ejecución de todos los métodos aplicados, definidos en la etapa previa.

Para completar esta etapa se ejecutaron todos los métodos aplicados a través de herramienta. A efectos prácticos, con el propósito de acelerar la ejecución sin perder representatividad, las consultas se limitaron a una muestra de 1000 registros del *data at hand*. Esta restricción permitió reducir significativamente el tiempo de procesamiento.

El resultado de la ejecución de estos métodos fue la obtención y el almacenamiento de los valores de CD en la base de datos *DQ Metadata*, diseñada específicamente para este fin. Finalmente, se verificó el correcto almacenamiento de estos resultados, tanto en las tablas de dicha base de datos organizadas por granularidad, como en las vistas proporcionadas por la interfaz de la herramienta.

7.1.5. Evaluación de CD

Una vez que se obtuvieron los valores de la medición de la CD, se procedió a la ejecución de la Etapa 6 - *DQ Assessment*, la cual se dividió en dos actividades principales.

Definición de Umbrales

La primera actividad consistió en la definición de umbrales para cada método de CD. Esto fue necesario para obtener valores de CD cualitativos a partir de la comparación de los valores cuantitativos de la medición de CD. Dichos umbrales se definieron considerando:

- La criticidad del atributo evaluado (atributos clave vs. no clave).
- El dominio de aplicación y la cantidad de intervalos requeridos para cada métrica de CD.

Para algunos casos, se aplicaron los umbrales presentados de manera predeterminada en la interfaz de la herramienta, principalmente a los métodos con resultados de tipo *boolean*, representando el caso positivo como *true*. Sin embargo, para los métodos aplicados a atributos clave, se ajustaron valores más

Métrica	Método	Método Aplicado
Check Valid Rating *factor: Semantic Accuracy; granularidad: columna; resultado: float*	***calculateCheckValidRating***: calcula la proporción de reseñas cuyo valor se encuentra dentro del rango.	aplicado a: `books_rating.review_score`.
Check Valid Price *factor: Semantic Accuracy; granularidad: celda; resultado: boolean*	***calculateCheckValidPrice***: verifica si el precio se encuentra dentro de un rango esperado.	aplicado a: `books_rating.price`.
Syntax Error Rate *factor: Syntactic Accuracy; granularidad: columna; resultado: float*	***detectSyntaxErrors***: devuelve el porcentaje de filas donde cierto atributo cumple las reglas sintácticas.	aplicado a: `books_data.image`.
Check Title Completion *factor: Density; granularidad: tupla; resultado: boolean*	***calculateCheckTitleCompletion***: evalúa por fila si el campo `title` no está vacío.	aplicado a: `books_data.title`.
Check Review Count *factor: Intra-relationship Integrity; granularidad: tabla; resultado: float*	***calculateCheckReviewCount***: devuelve el porcentaje de entradas cuyo dato de cantidad de *reviews* es igual a la cantidad actual de *reviews*.	aplicado a: `books_data.ratingsCount`, `books_rating.title`.
Unique Entry Ratio *factor: No-duplication; granularidad: tabla; resultado: float*	***calculateUniqueEntryRatio***: devuelve el porcentaje de libros no repetidos.	aplicado a: `books_data.title`.
Check Data Age *factor: Currency; granularidad: tupla; resultado: boolean*	***calculateCheckDataAge***: mide si la tupla de reseña es reciente.	aplicado a: `books_rating.review_time`.
CheckReviewByValidUser *factor: CheckUsersName; granularidad: celda; resultado: boolean*	***validateUsername***: verifica que el nombre de usuario sea válido.	aplicado a: `books_rating.profilename`.
isValidReviewText *factor: validReviewText; granularidad: columna; resultado: float*	***shortReviewPercentage***: calcula la proporción de reseñas "suficientemente largas".	aplicado a: `books_rating.review_text`.
Validate Book Description *factor: Check Valid Description; granularidad: celda; resultado: boolean*	***validateBookDescription***: verifica que la descripción no sea vacía.	aplicado a: `books_data.description`.

Tabla 7.2: Métricas, métodos y métodos aplicados del modelo de CD.

estrictos en los umbrales. Por ejemplo, en el método para medir el porcentaje de entradas únicas, el valor considerado "Excelente" se ajustó para superar el 95 %.

Ejecución de la Evaluación

Con los umbrales definidos, se procedió a la ejecución de la evaluación a través de la herramienta. Esta ejecución, comparó los valores de CD obtenidos en la etapa previa con los umbrales definidos. Luego se registran los resultados de la evaluación en la base de datos *DQ Metadata*.

En general, se obtuvo un nivel de CD aceptable, no obstante, la evaluación de los resultados de ciertos métodos arrojó resultados que no fueron óptimos. Esto se observó, por ejemplo, a partir de la discrepancia entre el valor asociado al atributo `reviews_count` y el número real de reseñas. Ademas, se encontraron ciertos nombres de usuario y precios que excedían los rangos esperados.

Finalmente, al ejecutarse las evaluaciones para los resultados de todos los métodos aplicados, la Etapa 6 se dio por concluida. Esto marcó la finalización completa de la Fase 2 de CaDQM y permitió verificar la ejecución de todas las etapas correspondientes a dicha fase utilizando la herramienta.

7.1.6. Conclusiones Generales

La ejecución del presente caso de estudio permitió cumplir con los objetivos definidos, incluyendo principalmente la verificación de un correcto uso de la herramienta implementada para la ejecución completa de la Fase 2 de CaDQM. A continuación se destacan algunas de las principales conclusiones:

- Soporte para la definición de un Modelo de CD basado en contexto: Se verificó la capacidad de la herramienta para guiar al usuario en la construcción de un modelo de CD fundamentado en el contexto de los datos y los problemas de CD disponibles. Además, se demostró la flexibilidad de la herramienta en este proceso, al permitir la creación de conceptos de CD desde cero o la reutilización de conceptos de CD predeterminados, cubriendo desde la definición de las dimensiones hasta la creación de los métodos y sus métodos aplicados.
- Utilidad de Funcionalidades IA: Se verificó que fue posible obtener dimensiones y factores recomendados adecuadamente fundamentados a partir del contexto y los problemas de CD. Por lo tanto, se constató que las funcionalidades de IA proporcionadas como asistencia para la construcción del modelo de CD representan un agregado que efectivamente ayuda en el proceso de ejecución de esta etapa.
- Capacidad de Medición y Evaluación de la CD: Se demostró la capacidad de la herramienta para llevar a cabo el proceso de medición de la CD para el *dataset* en cuestión, mediante la ejecución de todos los métodos aplicados. Además, fue posible definir umbrales personalizados para la evaluación de la CD, según las necesidades del contexto o criterios de expertos en CD. Esto, permitió obtener valores cualitativos de CD al comparar los resultados de las mediciones con los umbrales definidos. Finalmente se verifico que todos los resultados se almacenen en la base de datos de *DQ Metadata*. Adicionalmente, para algunos resultados fue posible corroborar la correctitud de los valores de CD obtenidos según estadísticas proporcionadas en la fuente original del *dataset*.

En resumen, a partir de los objetivos logrados, se concluye que la herramienta implementada funciona como un soporte completo para la ejecución de todas las etapas de la Fase 2 de CaDQM.

7.2. Caso de Estudio 2: Interoperabilidad

Este caso de estudio presenta dos objetivos principales: verificar la interoperabilidad entre las herramientas que implementan la Fase 1 - *DQ Planning* y la Fase 2 - *DQ Assessment* de CaDQM, y analizar cómo el contexto influye en la construcción de un modelo de CD.

Para el primer objetivo, se verificó que la herramienta de la Fase 2 fuera capaz de usar e integrar directamente los artefactos de la Fase 1 (componentes de contexto y problemas de CD). Estos datos

fueron almacenados en una base de datos común tras la ejecución de la Fase 1 (datos de salida), y posteriormente consumidos por la herramienta de la Fase 2 (datos de entrada).

Por último, con el fin de validar y analizar la pertinencia del modelo de CD resultante, se realizó una comparación con un modelo de CD de referencia, definido manualmente por expertos de dominio y en CD en el marco de la tesis de doctorado de [29]. Este análisis también se apoyó en los resultados de la comparación entre los modelos de contexto de referencia y el generado por la herramienta de la Fase 1, facilitados por los estudiantes del otro proyecto de grado.

7.2.1. Conjunto de Datos

Para este caso de estudio se utilizó un *dataset* con datos reales proporcionados por el Centro de Evaluación de Biodisponibilidad y Bioequivalencia de Medicamentos de la Universidad de la República (CEBIOBE) [3]. Este conjunto de datos contiene información sobre la administración y seguimiento de antibióticos en pacientes anonimizados.

Estos datos se cargaron en una base de datos relacional para ser usados como *data at hand* en la ejecución de la Fase 2. Se organizaron en un esquema simple de una única tabla, denominada "Antibioticos". La descripción del dataset se encuentra en el Anexo E.1.

7.2.2. Modelo de Contexto

Para este caso de estudio, se utilizó un modelo de contexto generado a través de la ejecución de la Fase 1 de CaDQM, realizada por otro grupo de proyecto de grado sobre el mismo conjunto de datos.

Interoperabilidad

En el proceso de acceder al modelo de contexto, se verificó la interoperabilidad entre las herramientas que implementan la Fase 1 - *DQ Planning* y la Fase 2 -*DQ Assessment*, desarrolladas por los dos proyectos de grado. El objetivo se logró mediante el uso de una base de datos común para ambos sistemas. En esta BD, la herramienta de la Fase 1 almacenó los artefactos generados como salida de su ejecución, incluyendo los componentes de contexto y los problemas de CD identificados, asociados a una entidad denominada `Proyecto`.

Cabe mencionar que, a pesar de trabajar sobre un Modelo Conceptual compartido, las decisiones específicas de implementación de cada equipo llevaron a algunas inconsistencias como nombres de tablas y atributos distintos, o tipos de datos diferentes. Estas diferencias se resolvieron a través de un proceso de comunicación constante, con intercambios de respaldos *dumps* de bases de datos entre los equipos. Este esfuerzo colaborativo aseguró que la base de datos común cumpliera con los requisitos de ambos proyectos.

Finalmente, una vez resueltos los inconvenientes, se recibió esta base actualizada por el grupo del otro proyecto de grado, con los resultados de la ejecución para su caso de estudio. Al configurar dicha BD como predeterminada en la implementación del *backend*, fue posible consumir e integrar directamente todos los artefactos. Este "Proyecto", definido en la Fase 1, sirvió como punto de partida para la ejecución de la Fase 2, lo que representó una continuación semántica según el flujo de CaDQM y demostró una interoperabilidad exitosa entre ambas fases.

Componentes de Contexto

Un análisis comparativo del modelo de contexto, realizado por el grupo de la Fase 1 - *DQ Planning*, reveló que, con la ayuda de la herramienta, se logró identificar correctamente el 52 % de los componentes de contexto presentes en el modelo de referencia. El 46 % de los componentes no coincidieron y el 2 % restante coincidió con algunas diferencias. Cabe destacar que el 46 % de los componentes de contexto faltantes corresponden a reglas de negocio y requerimientos de CD. Estos componentes de contexto se identifican generalmente en la interacción con los expertos del dominio y los usuarios de datos, tarea que se realiza en la Etapa 3 de la Fase 1. De acuerdo con lo registrado en la Fase 1, la Etapa 3 no fue

ejecutada, por no tener acceso a los expertos del dominio. Esta disparidad entre los modelos de contexto es una condición clave a tener en cuenta para la tarea de análisis comparativo del modelo de CD de referencia, ya que la construcción de un modelo de CD se basa directamente en el modelo de contexto dado.

En el Anexo E.2 se presenta el modelo de contexto utilizado para este caso de estudio, incluyendo los componentes de contexto generados por la herramienta de la Fase 1 y los componentes de CD no identificados en el modelo de contexto de referencia.

7.2.3. Modelo de CD

Como parte central del caso de estudio, se construyó un modelo de CD utilizando la herramienta implementada. Este modelo se definió en el marco de la ejecución de la Etapa 4 de la Fase 2 de CaDQM, basándose en el modelo de contexto generado durante la ejecución de la Fase 1 - *DQ Planning*. En esta sección, se presenta este modelo de CD, organizado por dimensiones y factores, fundamentando la inclusión de cada uno de ellos, según los componentes de contexto y los problemas de CD identificados en la Fase 1. En el Anexo E.3 se presenta este modelo de CD de forma completa.

El modelo de CD obtenido se comparo con un modelo de CD de referencia definido manualmente por expertos de dominio y en CD que usaron un modelo de contexto definido manualmente. En la comparación se detallan también las métricas, los métodos y los atributos sobre los que se aplico el modelo de CD. Estos detalles se presentan en tablas para cada factor donde el color de cada celda indica el nivel de similitud con el modelo de referencia:

- Verde: Indica coincidencia en ambos modelos.
- Amarillo: Indica que el concepto de CD se encuentra en ambos modelos de CD, pero con alguna diferencia.
- Rojo: Indica que no hay coincidencia entre los modelos de CD.

Dimensión	**Factor**
Accuracy	***Syntactic Accuracy***
	Semantic Accuracy
	Precision
Completeness	***Density***
	Coverage
Consistency	***Domain Integrity***
	Intra-relationship Integrity
Freshness	***Currency***
Uniqueness	***No Duplication***

Tabla 7.3: Resumen de dimensiones y factores del modelo de CD definido con la herramienta de la Fase 2. Las dimensiones y factores en verde fueron definidos también en el modelo de CD de referencia definido manualmente. En rojo, las dimensiones y factores que solo están en el modelo de CD construido con la herramienta.

DQ Dimension: Consistency

Como se observa en la Tabla 7.3, la dimensión *Consistency* fue definida en ambos modelos de CD, con el objetivo de capturar la satisfacción de reglas semánticas definidas sobre los datos. Para abordar aspectos específicos de esta dimensión, se definieron los factores de CD *Intra-Relationship Integrity* y *Domain Integrity*, los cuales también fueron definidos en el modelo de CD de referencia.

DQ Factor: Domain Integrity

Este factor se refiere a la satisfacción de reglas sobre el contenido de un atributo. Su inclusión surgió a partir de las reglas de negocio del contexto utilizado, referidas de rangos de valores válidos. Por su parte, en el modelo de referencia, este factor estuvo respaldado por una mayor cantidad de reglas de negocio y requerimientos de CD, definidos exclusivamente en el modelo de contexto asociado.

En la tabla que se presenta en la tabla 7.4 se muestra la métrica que se definió para medir el factor *Domain Integrity* y los métodos de CD que lo implementan. En particular, se definió una métrica con el propósito de calcular el porcentaje de conformidad para un rango de valores validos *(Value Range Compliance)*, con una granularidad por columna. Por su parte, en el modelo de referencia, se definió una métrica a nivel de tupla, que verifica las mismas condiciones pero devuelve un resultado booleano. Por lo tanto, si bien ambas buscan medir lo mismo, lo hacen con enfoques y niveles de granularidad distintos.

Métrica	**Método**	**Método Aplicado**
Value Range Compliance Ratio Granularity: Column Result domain: [0,1]	*calculateValueRangeComplianceRatio* Input: Lista Output: Float Logic: devuelve el porcentaje de rangos válidos para cierto atributo	Type: Measurement Applied to: Crea, Conc.LCR

Tabla 7.4: Métrica, método y métodos aplicados propuestos por la herramienta para el factor de CD *Domain Integrity*

Más allá de las diferencias en granularidad y tipo de medición, también se observan diferencias en los métodos aplicados y la cantidad de atributos que estos cubren. El modelo de CD de referencia aborda un mayor número de atributos, lo cual es resultado directo de la mayor cantidad de reglas de negocio y requerimientos de CD definidos en su modelo de contexto, que no fueron incluidos en el modelo de contexto que guia nuestro modelo de CD. Estos resultados se presentan en la tabla 7.4

DQ Factor: Intra-Relationship Integrity

Para este factor, se observa un comportamiento similar al de *Domain Integrity*. En el modelo construido, el factor surgió a partir de reglas de negocio que involucran relaciones entre diferentes atributos de la misma tabla. Por su parte, la justificación para la inclusión de este factor en el modelo de CD de referencia es más exhaustiva, dado que su modelo de contexto es más amplio, con un mayor número de reglas de negocio y requerimientos de CD relacionados. Adicionalmente, las métricas y los métodos utilizados en ambos modelos de CD, si bien buscan medir lo mismo, lo realizan de forma distinta, manteniendo el enfoque de porcentaje a nivel de columna, en contraste al de tuplas con resultados *booleanos* en el modelo de CD de referencia. Estos resultados se presentan en la Tabla 7.5.

Métrica	**Método**	**Método Aplicado**
Constraint Satisfaction Ratio Granularity: Table Result domain: [0,1]	*calculateTableConstraintSatisfactionRatio* Input: List Output: Float Logic: devuelve el porcentaje de cumplimiento de reglas entre atributos	Type: Measurement Applied to: Posología, Conc.Cont IR, Crea, Diálisis, Conc.PreHD, Conc.PostHD, IR, Crea, Diálisis, ATB, Conc.LCR, ATB, Vía, RazónTrat, Comentarios, ATB, Posología

Tabla 7.5: Métrica, método y métodos aplicados propuestos por la herramienta para el factor de CD *Intra-Relationship Integrity*

DQ Dimension: Completeness

La dimensión *Completeness*, que aborda la disponibilidad de los datos necesarios y asegura que no falte información importante para el análisis o la toma de decisiones, fue considerada en ambos modelos de CD. En el modelo construido, se definieron los factores *Density* y *Coverage*. De estos, el modelo de CD de referencia también incluye el factor *Density*, mientras que *Coverage* no fue abordado, siendo un factor exclusivo del modelo de CD construido con la herramienta.

Dq Factor: Density

Este factor surgió de varios requerimientos de CD relacionados con la cantidad de valores nulos en distintos atributos. Para abordarlos, se definió una métrica que mide el porcentaje de valores no nulos, implementando métodos aplicados específicos para cada atributo. En el modelo de CD referencia el enfoque fue análogo, utilizando métricas y métodos similares. Mientras que el modelo de CD de referencia utilizó un solo requerimiento que abarca un amplio número de atributos, en el modelo de contexto utilizado para el caso de estudio se utilizaron varios componentes específicos, lo que resultó en una cobertura menor para algunos atributos. Además, aunque los componentes de contexto implicaron la definición de métricas similares, los porcentajes de aceptación definidos en los requerimientos de CD variaron en cada modelo de contexto.

La tabla de la Tabla 7.6 presenta la métrica, el método y los atributos cuya CD fue medida. El color verde indica la coincidencia total entre ambos modelos de CD, mientras que el amarillo representa una coincidencia parcial.

Métrica	**Método**	**Método Aplicado**
Non Null Value Ratio Granularity: Column Result domain: [0,1]	*calculateNonValueRatio* Input: List Output: Float Logic: devuelve el porcentaje de valores no nulos	Type: Measurement Applied to: IR, Estado, Día.Últ.Dosis, Posología, Fecha, Crea, Diálisis

Tabla 7.6: Métrica, método y métodos aplicados para el factor de CD *Density*

DQ Factor: Coverage

Este factor, que es exclusivo del modelo de CD construido en este trabajo, surgió de un análisis más detallado de la completitud. A partir de un requerimiento de CD que, además de la condición de nulidad, incluía otras condiciones que los valores de un atributo no debían presentar, se decidió medir más allá del caso de los nulos. Para ello, se definió una métrica que mide el ratio de valores válidos con un método cuyo algoritmo permite, en las implementaciones específicas, contemplar estas condiciones de valores esperados. Estos resultados se muestran en la Tabla 7.7.

Métrica	**Método**	**Método Aplicado**
Valid Value Ratio Granularity: Column Result domain: [0,1]	*calculateValidValueRatio* Input: List Output: Float Logic: devuelve el ratio de valores dentro de los valores esperados	Type: Measurement Applied to: Vía

Tabla 7.7: Métrica, método y métodos aplicados para el factor de CD *Coverage*. Las celdas en color rojo se deben a que no se encuentran en el modelo de CD de referencia.

DQ Dimension: Accuracy

La dimensión *Accuracy*, que se enfoca en la veracidad y la representación correcta de las entidades del mundo real, fue definida en ambos modelos de CD. En el modelo de CD construido en este trabajo, se abordaron los factores *Syntactic Accuracy*, que también fue incluido en el modelo de CD de referencia, y *Semantic Accuracy* y *Precision*, que fueron definidos en este caso de estudio.

DQ Factor: Syntactic Accuracy

Tanto en el modelo de CD construido con la herramienta como en el modelo de CD de referencia, este factor surgió de requerimientos de CD que establecen un formato específico para las fechas.

Respecto a la métrica definida, si bien en ambos modelos se busca medir los mismos atributos, en el caso de estudio se definieron métricas orientadas a una medición general, mientras que el modelo de CD de referencia siguió un enfoque mas especifico, con de granularidad por celda y resultados *boolean*. Estas diferencias se observan en la tabla 7.8.

Métrica	**Método**	**Método Aplicado**
Syntax Compliance Ratio Granularity: Column Result domain: [0,1]	*calculateSyntaxComplianceRatio* Input: List Output: Float Logic: Devuelve el ratio de entradas que cumplen con el formato adecuado	Type: Measurement Applied to: Fecha, DíaÚltDosis

Tabla 7.8: Métrica, método y métodos aplicados para el factor de CD *Syntactic Accuracy*

DQ Factor: Semantic Accuracy

Este factor fue definido exclusivamente en el modelo de CD construido con la herramienta. Su inclusión surgió a partir de una regla de negocio que establece una relación temporal lógica entre dos fechas relacionadas a la dosificación de medicamento. La cual, si bien también se encuentra en el modelo de contexto de referencia, para este caso de estudio se interpretó que dicha regla busca validar que la secuencia temporal de la dosificación refleje correctamente los hechos del mundo real, lo cual es la esencia de la precisión semántica. Adicionalmente, se detectaron problemas de CD referentes a la falta de claridad en la representación de los datos.

Para medir este factor, se definió la métrica *Semantic Rule Compliance*. Con una granularidad por tupla y un dominio de resultado *boolean* indicando el cumplimiento o no, de una condición dada (Tabla 7.9).

Si bien la necesidad del factor Semantic Accuracy surgió tanto de reglas de negocio como de problemas de CD, los métodos solo se aplicaron sobre los atributos relacionados con los componentes de contexto que tenían reglas claras y formales. Los problemas de CD relacionados con las concentraciones, a pesar de justificar la necesidad del factor, no contaban con reglas o patrones válidos definidos, lo que impidió la aplicación de una métrica para su evaluación.

Métrica	**Método**	**Método Aplicado**
Semantic Rule Compliance Granularity: Tuple Result domain: Boolean	*calculateSemanticRuleCompliance* Input: Row Output: Boolean Logic: Devuelve true o false dependiendo cierta condición	Type: Measurement Applied to: Fecha, DíaÚltDosis

Tabla 7.9: Métrica, método y métodos aplicados para el factor de CD *Semantic Accuracy*

DQ Factor: Precision

Este factor, que es exclusivo del modelo de CD del caso de estudio, surgió de un análisis del dominio de aplicación y la tarea realizada con los datos analizados. Dado que el modelo de CD se da en un dominio médico y los datos se utilizan para estudios e investigaciones con fines estadísticos, se dedujo que la precisión era un requisito implícito y fundamental para la evaluación de los datos. Además, esta deducción demuestra que los factores de CD pueden surgir de una interpretación de los componentes de contexto más generales, y no únicamente de reglas de negocio o requerimientos de CD.

Para medir este factor, se definió una métrica que aborda los problemas de ambigüedad en las concentraciones, previamente identificados. El método asociado evalúa cada registro para determinar si un valor de concentración está presente en los datos sin la aclaración en el campo de `Comentarios`. Esta métrica, a nivel de fila, busca cuantificar la falta de nivel de detalle que compromete la precisión del dato. Los resultados se muestran en la tabla 7.10.

Métrica	**Método**	**Método Aplicado**
Detail Level Score By Row Granularity: Tuple Result domain: Boolean	*detailLevelScoreByRow* Input: Fila Output: Boolean Logic: Devuelve true o false dependiendo de cierta condicion	Type: Measurement Applied to: Conc.Valle, Conc.Pico, Conc.Cont, Conc.PreHD, Conc.PostHD, Conc.LCR, Conc, Comentarios

Tabla 7.10: Métrica, método y métodos aplicados propuestos por la herramienta para el factor de CD *Precision*

DQ Dimension: Uniqueness

La dimensión *Uniqueness*, que indica la ausencia de entradas duplicadas en un conjunto de datos, fue definida exclusivamente en el modelo de CD construido con la herramienta. Para de esta dimensión, se incluyó el factor *No-duplication*.

DQ Factor: No-duplication

Este factor surgió de la necesidad de garantizar la unicidad de cada registro de paciente. Su justificación se basa en un problema de CD que identifica al atributo `registro` como el atributo que identifica al paciente, y se apoya en un componente de contexto de filtrado de datos que requiere la identificación de un único paciente.

Para medir este factor, se definió una métrica de granularidad de tabla que mide el porcentaje de registros únicos (no duplicados). Si bien el campo registro identifica al paciente, una entrada se considera única solo si no comparte los mismos valores en los atributos `registro` y `fecha` con ninguna otra fila, lo que establece una clave compuesta sobre la cual se aplica el método. Los detalles se presentan en la tabla 7.11

Métrica	**Método**	**Método Aplicado**
Duplicate Non Entry Ratio Granularity: Table Result domain: [0,1]	*calculateNonDuplicateEntryRatio* Input: List Output: Float Logic: Devuelve el porcentaje de entradas no duplicadas	**Atributos:**Fecha,Registro

Tabla 7.11: Métrica, método y métodos aplicados propuestos por la herramienta para el factor de CD *No-duplication*

DQ Dimension: Freshness

Otra dimensión definida en el caso de estudio, que no incluye el modelo de CD referencia, es *Freshness*. Si bien no se identificó un requerimiento de CD que implicara su inclusión, se decidió abordarla con el fin de indagar en la actualidad de los datos a través del factor *Currency*.

DQ Factor: Currency

Para medir la vigencia de los datos, se definió una métrica que sirva para evaluar la antigüedad de los registros a nivel de fila. Si bien un componente de filtrado de datos hacía referencia a un rango de fechas, no se especificó un umbral de tiempo concreto para determinar la validez de los datos. Dada esa ambigüedad, se tomó la decisión pragmática de establecer un umbral de 10 años para considerar a los datos como desactualizados. De esta manera, se proporcionó un criterio definido para la medición, manejando un margen de tiempo considerable antes de que los registros se consideren no vigentes. Esto se muestra en la tabla 7.12

Métrica	**Método**	**Método Aplicado**
Recent Data Granularity: Cell Result domain: Boolean	*isRecentData* Input: Fecha Output: Boolean Logic: devuelve true o false si la fecha del registro es mayor a cierto umbral	**Atributos:**Fecha

Tabla 7.12: Métrica, método y métodos aplicados propuestos por la herramienta para el factor de CD *Currency*

Observación:

- En las dimensiones *Consistency* y *Accuracy*, el modelo de CD de referencia implementa métodos individuales para cada caso, mientras que el modelo de CD construido con la herramienta emplea una plantilla general de código *SQL*, cuya implementación final varía en cada método aplicado según la/s columna/s seleccionada/s.

Utilización de la funcionalidad de IA

Como una mención importante, este caso de estudio ofreció el escenario ideal para probar la funcionalidad de IA que genera sugerencias de pares de dimensiones-factores de CD basándose en los componentes de contexto y los problemas de CD. En contraste al caso de estudio anterior, donde el contexto fue definido manualmente y de manera más superficial, en este caso se disponía de un modelo de contexto detallado, producto de la ejecución de la herramienta para la Fase 1 de CaDQM. La exhaustividad de este contexto permitió probar y verificar la utilidad de la funcionalidad de IA.

Dado lo anterior, se realizó una ejecución completa de la Etapa 4 de la Fase 2 - *DQ Assessment* de CaDQM con la herramienta, utilizando la funcionalidad. Las sugerencias de la IA se utilizaron como un punto de partida para la construcción del modelo de CD, el cual fue finalizado tras realizar ajustes necesarios en cada recomendación, y tomando decisiones sobre la inclusión o no, de cada factor de CD recomendado.

Tras el análisis realizado entre el modelo de CD definido con la herramienta y el modelo de CD de referencia, un hallazgo interesante se observó en el orden en que se presentaron las sugerencias de la herramienta. Los primeros factores de CD recomendados fueron aquellos que surgen del modelo de contexto, como *Density* de *Completeness*, e *Intra-relationship Integrity* y *Domain Integrity* de *Consistency*. De hecho, si se consideran todos los factores de CD definidos en el modelo de CD de referencia, que fueron 4, estos fueron incluidos en las primeras 7 sugerencias. Las 3 sugerencias restantes dentro de este primer grupo (*Semantic Accuracy*, *Precision* y *Coverage*) también fueron resultaron formando parte del

modelo de CD definido. Es decir, las primeras 7 sugerencias de la herramienta fueron todas consideradas para la construcción del modelo de CD.

Además, el factor *Currency* de *Freshness* fue sugerido en los últimos puestos, lo cual se alinea a la explicación brindada respecto a su justificación de inclusión: no contaba con evidencia clara en ningún componente de contexto y surgió más bien de una decisión personal y una necesidad identificada en un análisis manual más amplio, lo cual explica su baja prioridad en las sugerencias de la herramienta.

7.2.4. Medición y Evaluación de la CD

Esta sección complementa la ejecución de la Fase 2 - *DQ Assessment* en la herramienta, enfocándose en la medición (Etapa 5) y la evaluación de la CD (Etapa 6), a partir del del modelo de CD construido.

Enfoque de la Medición de CD

En el modelo de CD construido con la herramienta, la mayoría de las métricas se definieron con un enfoque de medición de tipo ratio. Este enfoque se adoptó para obtener un primer panorama general de la CD, capturando su estado global en una medición inicial.

Análisis de la Evaluación de CD

En lo que respecta a la evaluación, se destaca que, por la propia construcción de la herramienta, se definió un sistema de umbrales en el que un valor alto siempre representa una mejor calidad, y uno bajo, una calidad inferior. Esta convención se estableció desde la definición de las métricas y los métodos; por ejemplo, para conocer la cantidad de datos nulos, se implementaron métodos que devuelven la cantidad de datos no nulos, donde un 100 % representa la calidad máxima. Se observó, además, que la evaluación de la CD depende fuertemente del tipo de usuario que la define y de los requerimientos de CD específicos definidos en el modelo de contexto.

7.2.5. Conclusiones Generales

A traves de la ejecución de este caso de estudio y su respectivo análisis, se lograron los objetivos y se validaron aspectos clave de la herramienta implementada. Las principales conclusiones son:

- Interoperabilidad Verificada: Se demostró con éxito la interoperabilidad entre las herramientas que implementan la Fase 1 - *DQ Planning* y la Fase 2 - *DQ Assessment*. Si bien el proceso presentó desafíos iniciales debido a inconsistencias de implementación, estos se resolvieron mediante una comunicación constante y colaboración entre los equipos. Este esfuerzo permitió obtener una base de datos común que pudo ser configurada como predeterminada en el *backend* de ambos sistemas, para un uso de tipo *plug-and-play*. Además, este enfoque, se validó así que la arquitectura, apoyada por la entidad `Proyecto`, permite una integración semántica de las fases y un flujo de ejecución continuo de la Fase 1 a la Fase 2 de CaDQM a través de ambas herramientas.
- Influencia del Contexto: Se mostró cómo el contexto influye fuertemente en la definición del modelo de CD. Se constató que las coincidencias entre los modelos de contexto permitieron cubrir la totalidad de las dimensiones y factores que se encontraron en el modelo de CD de referencia. De hecho, el uso de la herramienta permitió extender el modelo de CD de referencia, definiendo nuevas dimensiones y factores a partir de un análisis más exhaustivo que consideró no solo requerimientos de CD o reglas de negocio, sino también otros componentes de contexto como el dominio de la aplicación y los tipos de usuario y las tareas. No obstante, las diferencias contextuales y la subjetividad de cada usuario que actuó como experto en CD llevaron a variaciones en las métricas, métodos y, en consecuencia, en la cobertura de los atributos medidos. En este último aspecto, se reflejó una relación directa entre el tamaño de cada modelo de contexto y la cantidad de métodos aplicados para medir aquellos factores que coincidieron en ambos modelos de CD.
- Análisis de Medición y Evaluación: Se concluyó que la elección del enfoque de medición (porcentaje vs. granularidad por celda) depende en buena parte del objetivo del experto en CD a cargo, lo

que subraya la necesidad de flexibilidad en la herramienta. Además, se reafirmó que la evaluación de la CD es una tarea que se basa fuertemente en los requerimientos de CD y reglas de negocio específicas del contexto de los datos.

- Pertinencia de las recomendaciones utilizando IA: Adicionalmente, se constató que la funcionalidad de IA, para la recomendación de dimensiones y factores en función del modelo de contexto y los problemas de CD, demostró ser un agregado importante y útil para la definición del modelo de CD. Sus sugerencias se alinearon con la evidencia presente en el modelo de contexto, sugiriendo en primer lugar los factores que surgen de los componentes del contexto.

Capítulo 8

Conclusiones y Trabajo Futuro

8.1. Conclusiones

El objetivo principal del proyecto consistió en diseñar y desarrollar una herramienta para dar soporte a los expertos en CD, en la aplicación de la Fase 2 - *DQ Assessment* de la metodología CaDQM.

Mediante un proceso de validación con pruebas y casos de estudio, la herramienta implementada demostró ser un prototipo efectivo para la ejecución integral de la Fase 2 de CaDQM. Fue posible llevar a cabo todas las etapas de la Fase 2: (i) Etapa 4: Construcción del modelo de CD basado en el contexto y siguiendo la jerarquía de conceptos; (ii) Etapa 5: Medición de la CD mediante la ejecución de métodos de CD aplicados, almacenando los valores de CD obtenidos en una base de datos dedicada (*DQ Metadata*); y (iii) Etapa 6: Evaluación de la CD, utilizando umbrales definidos según los componentes de contexto, para habilitar un diagnóstico cualitativo de la CD.

En particular, la definición del modelo de CD se identificó como la etapa con mayor carga de trabajo y complejidad para el usuario dentro de la Fase 2. La herramienta abordó esta dificultad guiando al usuario paso a paso en la definición jerárquica de dimensiones, factores, métricas y métodos de CD, lo que permitió mantener la trazabilidad con los componentes de contexto y con los problemas de CD.

Adicionalmente, se integró IA en la herramienta, mediante un generador de recomendaciones de dimensiones y factores de CD a partir de componentes de contexto y problemas de CD, y del autocompletado en la definición de métodos de CD. A través de experimentaciones y la aplicación de los casos de estudio, utilizando modelos de lenguajes de uso libre, se verificó por un lado, que se generaron recomendaciones de dimensiones y factores de CD coherentes, mientras que la funcionalidad de los métodos de CD mostró una capacidad de producir algoritmos *SQL* útiles y fácilmente adaptables para la implementación de los métodos de CD aplicados. Esta asistencia permitió agilizar el proceso de definición del modelo de CD, reduciendo el esfuerzo del usuario sin sustituir el criterio experto.

Una vez finalizada la implementación completa de la aplicación, se verificó con éxito la interoperabilidad con la herramienta implementada por otro proyecto de grado, dedicada a la Fase 1 de CaDQM. A través de un caso de estudio, se corroboró que el modelo de contexto y los problemas de CD producidos en la Fase 1, almacenados en una base de datos común, pudieran ser consumidos directamente por la actual herramienta (Fase 2), para definir el modelo de CD y ejecutar las mediciones de CD sobre el mismo *data at hand*. De esta manera, se cumplió con éxito otro de los objetivos propuestos para el proyecto de grado, asegurando una completa compatibilidad entre los distintos artefactos generados por ambas aplicaciones desarrolladas y verificando, además, la continuidad metodológica entre ambas fases de CaDQM. La interoperabilidad lograda con otra herramienta demuestra la factibilidad de un desarrollo colaborativo y complementario entre distintos proyectos de grado, permitiendo así sentar bases firmes para avanzar hacia una herramienta unificada que aborde el ciclo completo de gestión de CD sensible al contexto, tal como fue propuesto por Serra en su tesis de doctorado.

La aplicación implementada se disponibilizó, junto con su documentación y manual de usuario, para ser utilizada en un módulo de taller dictado como curso de grado en la Facultad de Ingeniería de la Universidad de la República, iniciado en setiembre de 2025. Dicho curso se planteó con el propósito de probar la Fase 1 - *DQ Planning* y la Fase 2 -*DQ Assessment* de CaDQM.

Finalmente, se destaca la contribución didáctica producida mediante este proyecto de grado como un recurso académico valioso en el área de la gestión de la CD.

8.2. Trabajo Futuro

Si bien la implementación actual de la herramienta logró abarcar los objetivos propuestos del proyecto de grado y dio soporte completo a la ejecución de la Fase 2 de CaDQM, el alcance definido deja lugar a diversas líneas de trabajo futuro que podrían expandir sus funcionalidades y su utilidad. Estas líneas de trabajo se detallan a continuación.

- Unificación de las herramientas para la Fase 1 - *DQ Planning* y Fase 2 - *DQ Assessment*: Una clara linea de trabajo futuro, consiste en la unificación de la aplicación actual desarrollada para la Fase 2 con la herramienta implementada para la Fase 1 (producto de otro proyecto de grado paralelo), en una herramienta única y completamente integrada. Actualmente ambas aplicaciones ya son interoperables a nivel de datos (trabajan sobre una misma instancia de base de datos) y se diseñaron con estilos de interfaz similares *("look and feel")*, como punto de partida para dicha integración final.
- Incorporación de la Fase 3 - *DQ Improvement*: Complementando la unificación de las herramientas de la Fase 1 y Fase 2 en una aplicación única, se sentarían las bases para la incorporación de la Fase 3, centrada en la mejora de CD. Esto permitiría cubrir todas las fases de la metodología CaDQM dentro de una única solución que dé soporte a los expertos en CD para la gestión de la CD sensible al contexto, a lo largo de todo su ciclo de vida: planificación, evaluación y mejora de la CD.
- Reportes configurables para la visualización de resultados: Una funcionalidad proyectada para la herramienta actual consiste en incorporar reportes configurables que faciliten la visualización de los valores de CD obtenidos en las mediciones y evaluaciones. Si bien la herramienta ya permite visualizar estos resultados desde la propia interfaz (filtrando por granularidades, tablas o métodos de CD), los nuevos reportes permitirían consolidar múltiples resultados bajo diferentes enfoques. Por ejemplo, se podrían generar informes detallados para dimensiones específicas de CD o atributos de interés del *data at hand*, disponibles como archivos descargables, generando así artefactos que apoyen un eventual monitoreo continuo como parte de la gestión de la CD.
- Flexibilidad en la implementación de métodos de CD: Actualmente, la definición de métodos de CD en la herramienta se limita a implementaciones mediante código *SQL* puro, con el fin de permitir su posterior ejecución. No obstante, la arquitectura del sistema fue concebida contemplando la posibilidad de incorporar algoritmos más complejos, como consultas *SQL* compuestas o lógica de control embebida en un algoritmo *Python*. Esto le daría mayor flexibilidad a los expertos en CD para cubrir distintas necesidades, especialmente en dominios donde se requieren métodos de CD más avanzados para la medición de la CD.
- Extensión del uso de IA en la herramienta: Además de la asistencia actual en la definición de métodos de CD y la recomendación de dimensiones y factores de CD, se proyecta incorporar nuevas funcionalidades basadas en IA, enfocadas especialmente en la definición de métricas de CD y la sugerencia de umbrales de evaluación de CD. Esto permitiría extender la automatización de las distintas tareas que constituyen y facilitan la aplicación de la metodología CaDQM, sin perder el control por parte del experto.
- Fuentes de datos no relacionales: Actualmente, la herramienta se limita a trabajar con bases de datos relacionales. Una posibilidad de trabajo futuro podría ser ampliar el alcance de la herramienta para incluir también el uso de bases de datos no relacionales (*NoSQL*).

Bibliografía

[1] Amazon.com, Inc. *Amazon: Multinational Technology Company*. https://en.wikipedia.org/wiki/Amazon_(company). Fundada el 5 de julio de 1994; sede en Seattle, WA. 2025.

[2] Leopoldo Bertossi y Flavio Rizzolo. "Contexts and Data Quality Assessment". En: *arXiv, Cornell University* (mayo de 2012). DOI: 10.48550/arXiv.1608.04142.

[3] Centro de Evaluación de Biodisponibilidad y Bioequivalencia (CEBIOBE). *Centro de Evaluación de Biodisponibilidad y Bioequivalencia (CEBIOBE), Uruguay*. https://www.fq.edu.uy/?q=node/474. Institución uruguaya de investigación en farmacología y bioequivalencia. 2025.

[4] C. Cichy y S. Rass. "An Overview of Data Quality Frameworks". En: *IEEE Access* PP.c (2019), págs. 1-1. DOI: 10.1109/ACCESS.2019.2899751.

[5] Instituto de Computación y el Instituto de Agrimensura de la Facultad de Ingeniería - Universidad de la República. *Framework para la Gestión de la Calidad de Datos en Gobierno Digital*. Último acceso: 11 de noviembre de 2025. Dic. de 2019. URL: https://www.gub.uy/agencia-gobierno-electronico-sociedad-informacion-conocimiento/sites/agencia-gobierno-electronico-sociedad-informacion-conocimiento/files/documentos/publicaciones/Framework%20para%20la%20Gesti%C3%B3n%20de%20la%20Calidad%20de%20Datos%20v1.0.pdf.

[6] Walter Samá Danilo Ardagna Cinzia Cappiello y Monica Vitali. "Context-aware data quality assessment for big data". En: *ResearchGate* (2018). URL: https://www.researchgate.net/publication/326452333_Context-aware_data_quality_assessment_for_big_data.

[7] Django REST Framework Team. *Django REST Framework: Web APIs for Django*. Último acceso: 9 de octubre de 2025. 2025. URL: https://www.django-rest-framework.org/.

[8] Express.js Team. *Express: Fast, Unopinionated, Minimalist Web Framework for Node.js*. Último acceso: 9 de octubre de 2025. 2025. URL: https://expressjs.com/.

[9] Hadi Fadlallah et al. "Context-aware Big Data Quality Assessment: A Scoping Review". En: *Journal of Data and Information Quality* 15 (jun. de 2023). DOI: 10.1145/3603707.

[10] Django Software Foundation. *Django: Framework Web en Python*. Último acceso: 1 de abril de 2025. 2025. URL: https://www.djangoproject.com/.

[11] PostgreSQL Global Development Group. *PostgreSQL: Sistema de Gestión de Bases de Datos*. Último acceso: 1 de abril de 2025. 2025. URL: https://www.postgresql.org/.

[12] Groq Inc. *Groq: IA de Alto Rendimiento*. Último acceso: 1 de abril de 2025. 2025. URL: https://groq.com/.

[13] Groq Inc. *GroqCloud Models — Supported Models*. Último acceso: 9 de octubre de 2025. 2025. URL: https://console.groq.com/docs/models.

[14] Groq Inc. *Llama-3-70B-8192 — GroqCloud Models*. Último acceso: 9 de octubre de 2025. 2025. URL: https://console.groq.com/docs/model/llama3-70b-8192.

[15] Groq Inc. *Llama-3-8B-8192 — GroqCloud Models*. Último acceso: 9 de octubre de 2025. 2025. URL: https://console.groq.com/docs/model/llama3-8b-8192.

[16] Groq Inc. *Llama-3.1-8b-instant — GroqCloud Models*. Último acceso: 9 de octubre de 2025. 2025. URL: https://console.groq.com/docs/model/llama-3.1-8b-instant.

[17] Groq Inc. *Llama-3.3-70B-Versatile — GroqCloud Models*. Último acceso: 9 de octubre de 2025. 2025. URL: https://console.groq.com/docs/model/llama-3.3-70b-versatile.

[18] Shadi Iskander et al. "Quality Matters: Evaluating Synthetic Data for Tool-Using LLMs". En: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Ed.

por Yaser Al-Onaizan, Mohit Bansal y Yun-Nung Chen. Miami, Florida, USA: Association for Computational Linguistics, nov. de 2024, págs. 4958-4976. DOI: 10.18653/v1/2024.emnlp-main.285. URL: https://aclanthology.org/2024.emnlp-main.285/.

[19] Karolyn Kerr, Tony Norris y Rosemary Stockdale. "Data quality information and decision making: A healthcare case study". En: *ACIS 2007 Proceedings - 18th Australasian Conference on Information Systems* (ene. de 2007).

[20] LangChain. *LangChain Documentation*. Último acceso: 8 de octubre de 2025. 2025. URL: https://docs.langchain.com/.

[21] David Loshin. *Evaluating the Business Impacts of Poor Data Quality*. Inf. téc. Submitted by David Loshin, President, Knowledge Integrity, Inc. Silver Spring, MD, USA: Knowledge Integrity, Inc., 2020. URL: https://www.myecole.it/biblio/wp-content/uploads/2020/11/3_DK_2DS_Business_Impacts_Poor_Data_Quality.pdf.

[22] Meta Platforms, Inc. *React: A JavaScript library for building user interfaces*. https://react.dev. Meta Platforms, Inc. 2024.

[23] Node.js Foundation. *Node.js: JavaScript Runtime Built on Chrome's V8 JavaScript Engine*. Último acceso: 9 de octubre de 2025. 2025. URL: https://nodejs.org/.

[24] Oracle Corporation. *MySQL 8.4 Reference Manual*. https://dev.mysql.com/doc/refman/8.4/en/. Oracle Corporation. 2024.

[25] Pallets Projects. *Flask: A Lightweight WSGI Web Application Framework*. Último acceso: 9 de octubre de 2025. 2025. URL: https://flask.palletsprojects.com/.

[26] *Repositorio del Backend del Proyecto CaDQM Phase 2 - DQ Assessment*. Último acceso: 11 de noviembre de 2025. 2025. URL: https://github.com/maurixavi/CaDQM-phase2-dq-assessment-backend.

[27] *Repositorio del Frontend del Proyecto CaDQM Phase 2 - DQ Assessment*. Último acceso: 11 de noviembre de 2025. 2025. URL: https://github.com/maurixavi/CaDQM-phase2-dq-assessment-frontend.

[28] Guido van Rossum y the Python development team. *Python 3 Reference Manual*. https://www.python.org/doc/. Python Software Foundation. 2024.

[29] Flavia Serra. "Context-aware data quality management. Tesis de doctorado. Universidad de la República (Uruguay). Facultad de Ingeniería. Université de Tours (Francia)". En: (2025).

[30] Flavia Serra y Adriana Marotta. *Calidad de Datos e Información*. Último acceso: 29 de marzo de 2025. 2025. URL: https://eva.fing.edu.uy/course/view.php?id=1073&section=0.

[31] Flavia Serra et al. "Modeling context for data quality management". En: *Conceptual Modeling - 41st International Conference, ER 2022, Proceedings*. Ed. por J. Ralyté et al. Vol. 13607. Lecture Notes in Computer Science (LNCS). Springer, 2022, págs. 325-335.

[32] Angular Team. *Angular: Framework para Aplicaciones Web*. Último acceso: 1 de abril de 2025. 2025. URL: https://angular.io/.

Anexo A

Manual de Usuario

A.1. Instalación y Configuración del Sistema

Para poder utilizar la herramienta, es necesario configurar el entorno de trabajo para el servidor *backend* y el cliente *frontend*. A continuación, se detallan los pasos para la instalación de cada componente.

A.1.1. Configuración del Servidor Backend

El servidor *backend* gestiona toda la lógica de negocio y la comunicación con las bases de datos.

Requisitos

- *Python*: versión 3.8 o superior.
- *virtualenv*: herramienta para la creación de entornos virtuales de *Python*.
- *PostgreSQL*: versión 12 o superior.

Pasos de Instalación

1. **Clonar el repositorio:**

   ```
   git clone https://github.com/maurixavi/CaDQM-phase2-dq-assessment-backend
   ```

2. **Crear y activar entorno virtual:**

   ```
   python -m venv env
   source env/bin/activate
   En Windows: env\scripts\activate
   ```

3. **Instalar dependencias:**

   ```
   pip install -r requirements.txt
   ```

Configuración Bases de Datos

El sistema requiere dos bases de datos para su correcto funcionamiento: `cadqm_db` y `cadqm_dqmetadata_db`. Para su correcta configuración, seguir los siguientes pasos:

Consideraciones iniciales

Dado que la actual herramienta (Fase 2 - *DQ Assessment*) es interoperable con una herramienta externa que ejecuta la Fase 1 - *DQ Planning* de CaDQM con el uso de una misma BD común, se consideran los siguientes escenarios para la configuración de las bases de datos del *backend* de la Fase 2:

- Si se dispone de la BD proveniente de la aplicación de la Fase 1:
  - Para la BD de la aplicación (`cadqm_db`), que ya debe estar creada con las estructuras y datos de la Fase 1, solo se deben configurar las credenciales de conexión (Paso 2). Se pueden omitir los Pasos 1 (crear BD `cadqm_db` vacía) y 3 para esta BD.
  - Para la BD de metadatos (`cadqm_dqmetadata_db`), se deben seguir todos los pasos a continuación (1, 2, 3 y 4). Para el Paso 3:
    - Se recomienda la **Opción A**, al ser la opción mas simple y directa.
    - Si se selecciona la **Opción B** (Migraciones) entonces en el Paso 1 se debe ejecutar también la creación de la BD `cadqm_db`, requerida para que todas las migraciones se ejecuten correctamente. Una vez realizadas las migraciones, se deben actualizar las credenciales de la conexión de la BD `cadqm_db` para que apunten a la BD proveniente de la aplicación de la Fase 1.
- Si se desea ejecutar esta herramienta de forma independiente (prueba o desarrollo), se deben seguir todos los pasos a continuación para la configuración de ambas bases de datos (`cadqm_db` y `cadqm_dqmetadata_db`). La Opción A de restauración incluye en `cadqm_db` un proyecto de ejemplo por defecto para facilitar la ejecución y verificar el funcionamiento de la aplicación. Esta BD incluye, además, la carga de datos iniciales descrita en el Paso 4.

1. **Crear las bases de datos vacías:**

Accede a una instancia de *PostgreSQL* y crear las dos bases de datos necesarias.

```
CREATE DATABASE cadqm_db;
CREATE DATABASE cadqm_dqmetadata_db;
```

2. **Configurar las credenciales de conexión:**

El archivo `settings.py`, ubicado en el directorio `myproject/`, ya incluye la estructura para la conexión. Debes actualizar y completar los campos de `USER`, `PASSWORD`, `HOST` y `PORT` con los datos de su entorno de trabajo.

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'cadqm_db',
        'USER': 'your_user',
        'PASSWORD': 'your_password',
        'HOST': 'localhost',
        'PORT': '5432',
    },
    'metadata_db': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'cadqm_dqmetadata_db',
        'USER': 'your_user',
        'PASSWORD': 'your_password',
        'HOST': 'localhost',
        'PORT': '5432',
    }
}
```

3. **Cargar estructuras de tablas requeridas:**

Una vez creadas las bases de datos vacías y configuradas las credenciales de conexión, se ofrecen dos alternativas para el proceso de carga inicial con la estructura de tablas.

Opción A: Restauración a partir de *dumps SQL*

En el directorio `db_dumps/` del repositorio se disponibilizan dos archivos *pg_dump*, `cadqm_db.sql` y `cadqm_dqmetadata_db.sql`. Esta opción permite obtener una base de datos lista para su uso de forma inmediata. Ejecutar los siguientes comandos para generar las estructuras de tabla requeridas para cada base.

```
psql -U <tu_usuario> -d cadqm_db -f db_dumps/cadqm_db.sql
psql -U <tu_usuario> -d cadqm_dqmetadata_db -f db_dumps/cadqm_dqmetadata_db.sql
```

Opción B: Ejecutar migraciones *Django*

Con este enfoque, el usuario construirá la estructura de la base de datos desde cero. Ejecutar las siguientes migraciones en estricto orden.

```
# Crear tablas django
python manage.py migrate contenttypes
python manage.py migrate auth
python manage.py migrate sessions

# Crear tablas DQ Model (--fake evita creación de tablas en bases incorrectas)
python manage.py migrate dqmodel 0001
python manage.py migrate dqmodel 0001 --database=metadata_db --fake
python manage.py migrate dqmodel 0002 --database=metadata_db
python manage.py migrate dqmodel 0002 --fake

# Crear tablas Project
python manage.py migrate project 0001

# Crear tablas Admin
python manage.py migrate admin
```

4. **Cargar datos iniciales:**

El sistema requiere una carga inicial (en la BD `cadqm_db`) de un conjunto de dimensiones y factores de CD base predefinidos. Esta acción es clave para el uso adecuado de la herramienta, y necesario para la generación de recomendaciones mediante IA. Se debe ejecutar el siguiente comando desde el directorio raíz del proyecto *backend*:

```
python manage.py load_dqtemplate --template preset_dq_dimensions_factors_base
```

Configuración de variables de entorno

Para las funcionalidades de IA, es necesaria una clave de acceso a la *API* de *Groq*. Esta debe configurarse en el archivo `.env`, ubicado dentro del directorio `myproject/`:

```
GROQ_API_KEY=tu_api_key_aqui
```

Para obtener una clave *API*:

- Acceder a `https://console.groq.com/keys`
- Crear una cuenta (gratuita) en caso de no disponer de una.
- Generar una nueva *API Key* y copiarla.

A.1.2. Configuración del Cliente Frontend

El cliente *frontend* es la interfaz de usuario de la aplicación. Se conecta automáticamente al servidor *backend* una vez que este esté en ejecución.

Requisitos

- *Node.js*: versión 18.x o superior.
- *npm*: versión 9.x o superior.
- *Angular CLI*: versión 18.0.3 o superior.

Pasos de Instalación

1. **Clonar el repositorio:**

   ```
   git clone https://github.com/maurixavi/CaDQM-phase2-dq-assessment-frontend
   ```

2. **Instalar dependencias:**

   ```
   npm install
   ```

A.1.3. Ejecución de la Aplicación

Para iniciar la herramienta, se debe ejecutar ambos servidores en terminales separadas, uno para el *backend* y otro para el *frontend*.

1. **Iniciar el servidor backend:** En el directorio raíz del proyecto *backend*, ejecutar el siguiente comando:

   ```
   python manage.py runserver
   ```

   El servidor estará corriendo en `http://127.0.0.1:8000/` o `http://localhost:8000/`.

2. **Iniciar el cliente frontend:** En el directorio raíz del proyecto *frontend*, ejecutar el siguiente comando:

   ```
   ng serve
   ```

Una vez que el servidor y el cliente estén en ejecución, se podrá acceder a la aplicación desde el navegador web, la cual estará disponible en la dirección `http://localhost:4200/`

A.2. Interfaces y Navegación

Esta sección describe las distintas interfaces de la aplicación, cómo interactuar con ellas y navegar a través de las diferentes funcionalidades.

A.2.1. Inicio y Selección de Proyecto

La aplicación inicia con esta interfaz (Figura A.1). Aquí se listan todos los proyectos existentes en el sistema, con sus detalles, incluyendo la etapa (*Stage*) y estado (*Status*) en la que se encuentra. El usuario debe seleccionar un proyecto (botón *Select*) para iniciar o continuar su ejecución de la Fase 2 - *DQ Assessment*.

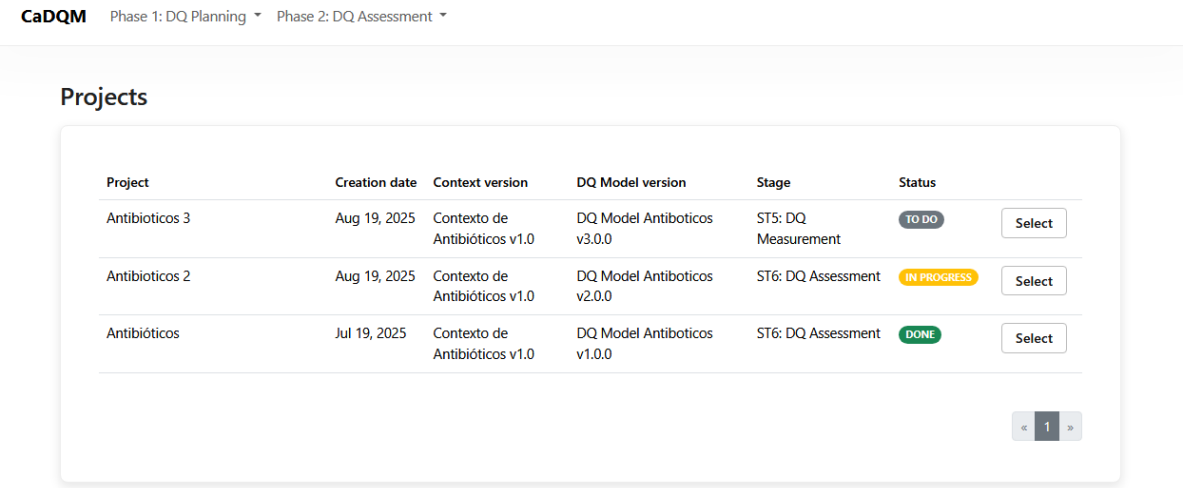

Figura A.1: Pantalla inicial de la Aplicación

A.2.2. Dashboard del Proyecto

Una vez seleccionado el proyecto, se accede a un Dashboard que centraliza la información principal proyecto. Desde esta interfaz el usuario puede realizar las siguientes acciones:

- Si el proyecto no tiene un modelo de CD, el usuario debe crear un modelo de CD para comenzar la ejecución de la Etapa 4 de la Fase 2 de CaDQM. Para crear un nuevo modelo de CD debe seleccionar el botón *Create DQ Model* del dashboard de la Figura A.2 para que se despliegue un modal donde debe insertar el nombre del modelo de CD. Para confirmar debe presionar el botón *Create DQ Model* del modal de la Figura A.3.

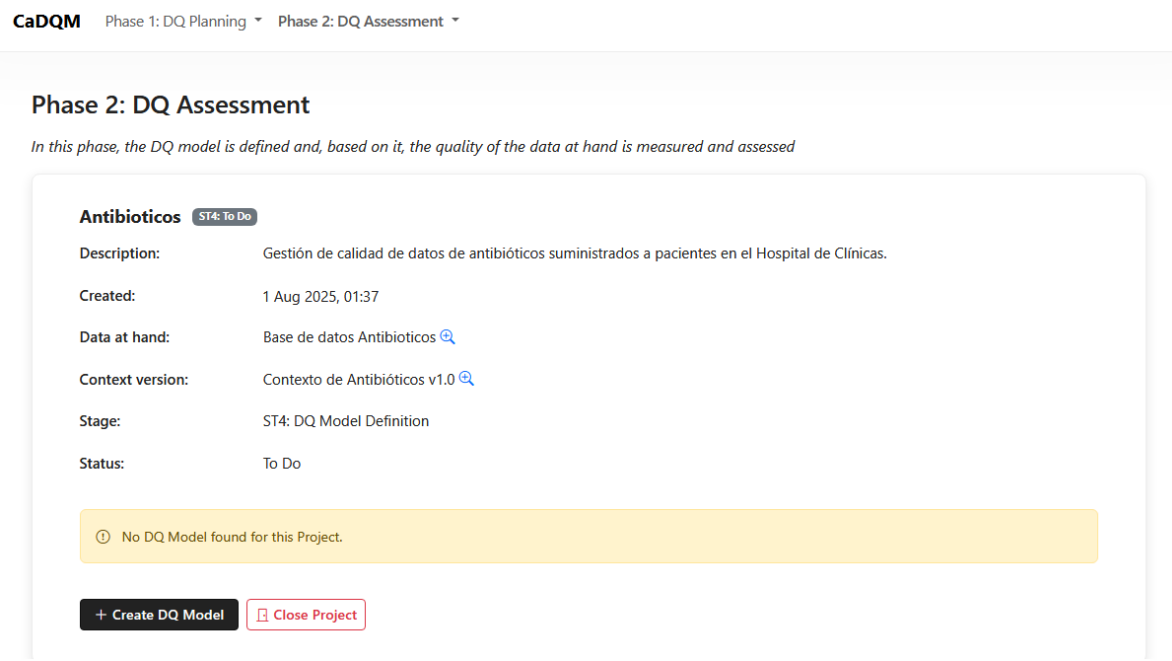

Figura A.2: Dashboard de un proyecto sin *DQ Model*.

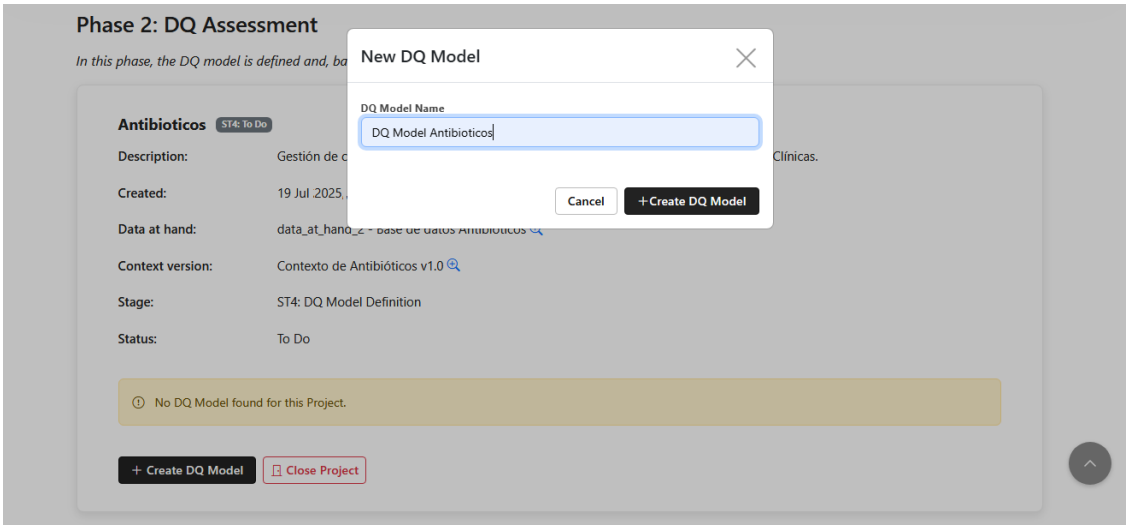

Figura A.3: Interfaz para la creación de un modelo de CD.

- Si el proyecto tiene un modelo de CD finalizado, en esta interfaz puede definir una nueva versión de un modelo de CD. Para esto debe seleccionar el botón *New DQ Model Version* que aparece en la parte inferior del dashboard, como se observa en Figura A.4. Se desplegará un modal con dos opciones:
  - *DQ model from Scratch*: se crea un nuevo modelo de CD vacío para su definición desde cero.
  - *New DQ Model Version*: se crea un una copia completa del actual modelo de CD para su definición desde dicho template.

  Al crear una nueva versión un modelo de CD, el usuario pasará ejecutar un proyecto nuevo al cual fue asociado.

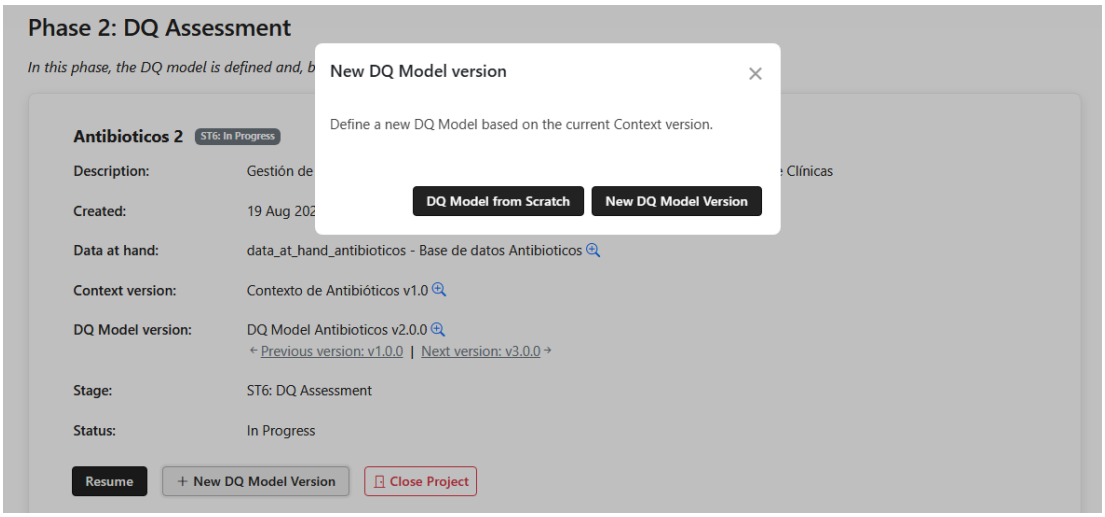

Figura A.4: Interfaz del dashboard para la creación de una nueva versión de un modelo de CD.

- Desde el menú superior de la aplicación, puede navegar a las interfaces de las etapas habilitadas según la ejecución de la Fase 2. En la Figura A.5 se observa que se encuentra habilitado solamente la Etapa 4, pues aún continua pendiente de finalización.
- Presionando el botón *Resume* en la Figura A.5 puede reanudar la ejecución de la etapa, navegando a la interfaz de la actividad pendiente correspondiente.
- Para salir del proyecto actual el usuario debe seleccionar botón *Close Project* ubicado en la parte inferior la interfaz de la Figura A.5, mediante el cual navegará hacia la página de Inicio para poder seleccionar un proyecto nuevo.

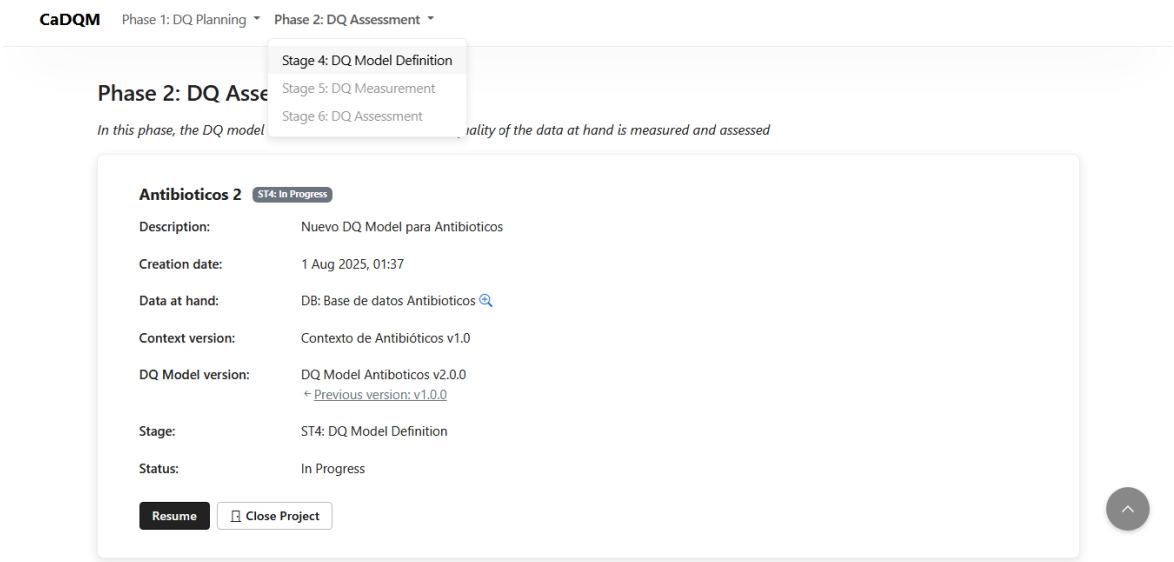

Figura A.5: Dashboard de un proyecto que se encuentra ejecutando la Etapa 4.

- En el Dashboard y en toda interfaz de la Etapa 4, se incluye un botón fijo en la esquina derecha inferior para la visualización de los componentes de contexto (Figura A.7) y problemas de CD (priorizados) del proyecto (Figura A.6).

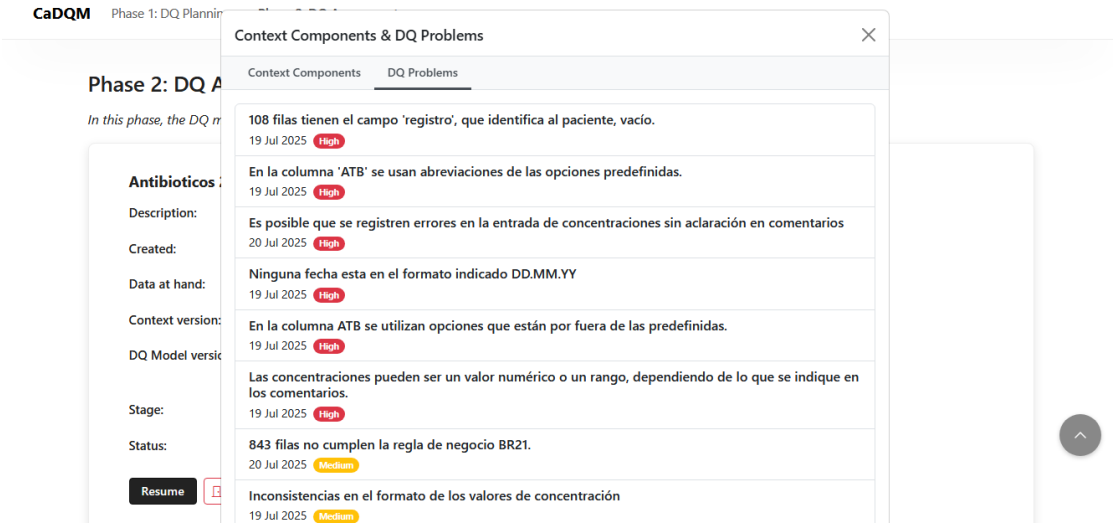

Figura A.6: Componentes de contexto para el proyecto en ejecución.

Figura A.7: Problemas de CD (priorizados) disponibles para el proyecto en ejecución.

A.2.3. Priorización de Problemas de Calidad de Datos

La Etapa 4 de la Fase 2 comienza con la Priorización de Problemas de CD, cuya interfaz se presenta en la Figura A.8. En esta interfaz, el usuario debe establecer la importancia de cada problema de CD disponible en el proyecto. El usuario dispone de las siguientes acciones en esta interfaz (Figura A.8):

- Consultar la lista de problemas de CD disponibles, organizados en columnas que representan las prioridades: *High*, *Medium* y *Low*.
- Para cambiar la prioridad, se presentan dos vías:
  - Reordenar los problemas de CD moviendo los ítems entre las columnas de prioridad (*drag and drop*).
  - Seleccionar la prioridad deseada directamente desde el desplegable (opción *select*) de cada problema.
- Guardar los cambios de la priorización seleccionando el botón *Save Priorization*, ubicado en la esquina inferior izquierda (Figura A.8), y confirmando dicha acción en un modal correspondiente.

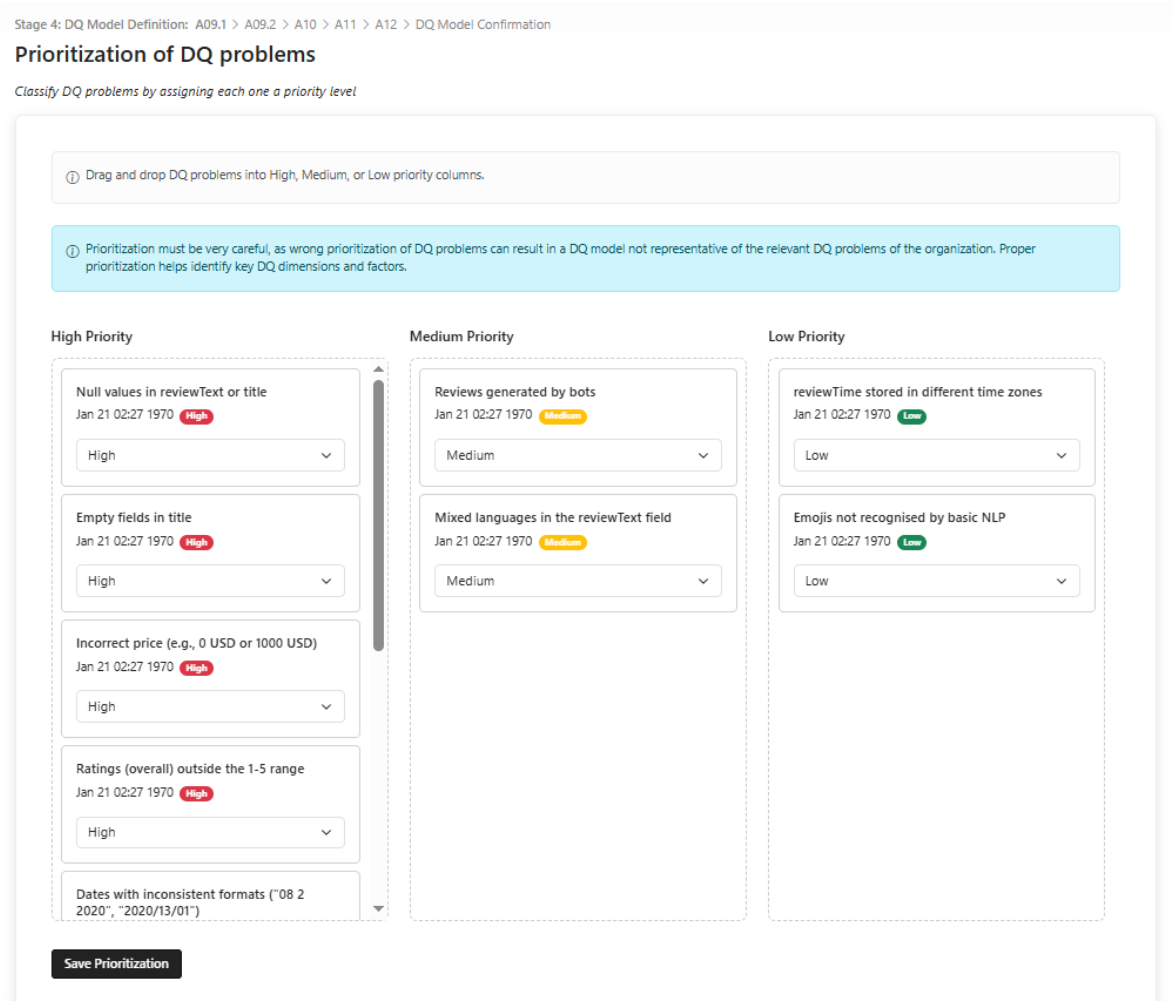

Figura A.8: Priorización de los Problemas de CD

A.2.4. Selección de Problemas de Calidad de Datos

Luego de priorizar los problemas de CD, el usuario debe seleccionar cuáles serán considerados en la definición del modelo de CD. La interfaz para la *Selección de Problemas de CD* se presenta en la Figura A.9, y permite al usuario realizar las siguientes acciones:

- Visualizar y seleccionar los problemas de CD disponibles (Figura A.9), los cuales están categorizados y organizados por su prioridad (*High*, *Medium*, *Low*).
- En la columna *Available Prioritized DQ Problems* (Figura A.9), el usuario puede incluir problemas de CD mediante dos formas:
  - Individualmente, mediante el botón + asociado a cada problema de CD.
  - Seleccionando todos los problemas de CD de una misma de prioridad con el botón *Select All*.
- Revisar la selección en la columna *DQ Problems Selection* (Figura A.9). Desde aquí, el usuario puede remover un problema de CD del conjunto seleccionado utilizando el botón correspondiente con el ícono de remoción.
- Guardar la selección mediante el botón *Save Selection*, ubicado en la esquina inferior izquierda de la interfaz (Figura A.9). Para poder avanzar, es obligatorio haber seleccionado al menos un problema de CD.

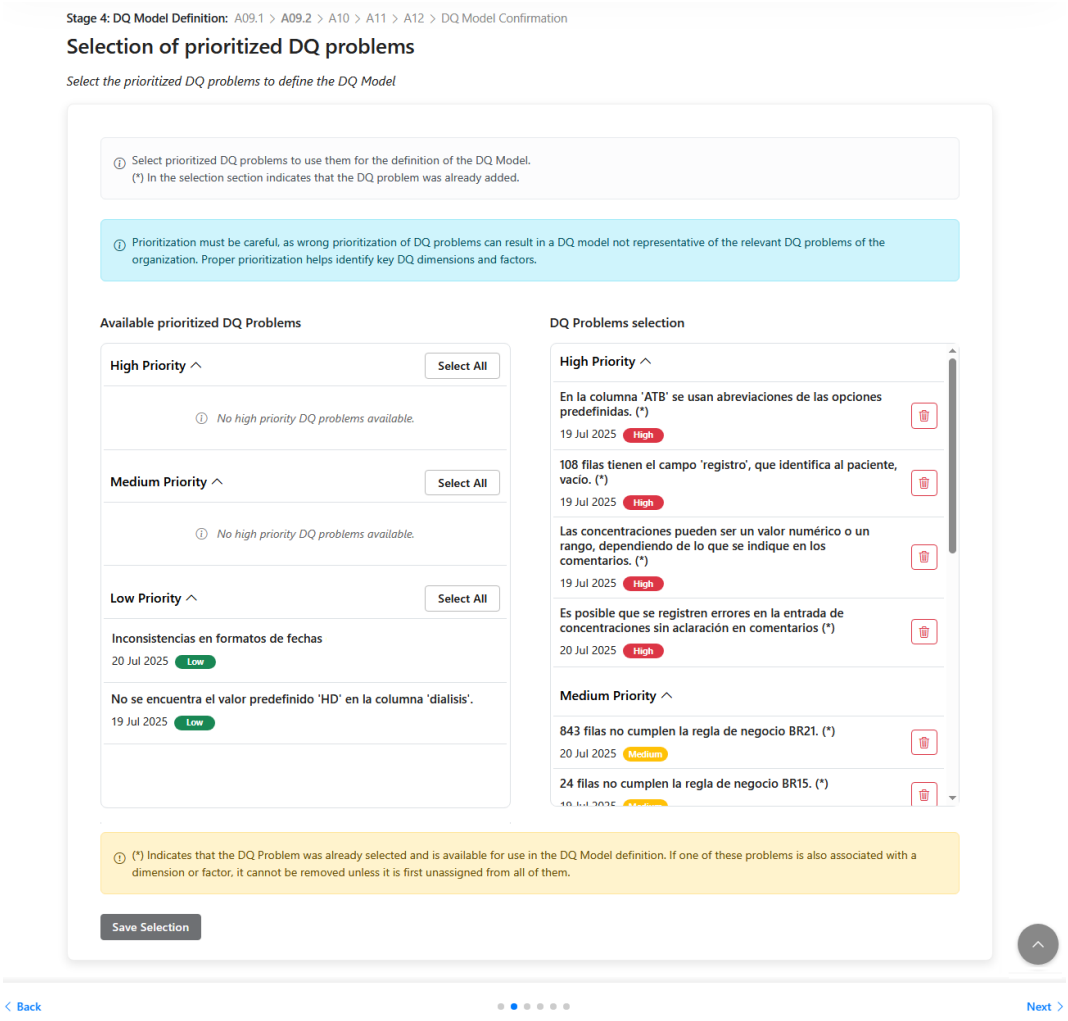

Figura A.9: Selección de los Problemas de CD

A.2.5. Definición de Dimensiones y Factores de Calidad de Datos

En este paso, el usuario debe comenzar a definir los conceptos de CD para el modelo de CD, seleccionando las dimensiones y factores de CD que desea agregar. Esta selección se apoya teniendo en consideración los componentes de contexto del proyecto y los problemas de CD priorizados y seleccionados en los pasos previos. La interfaz para esta actividad se presenta en la Figura A.10.

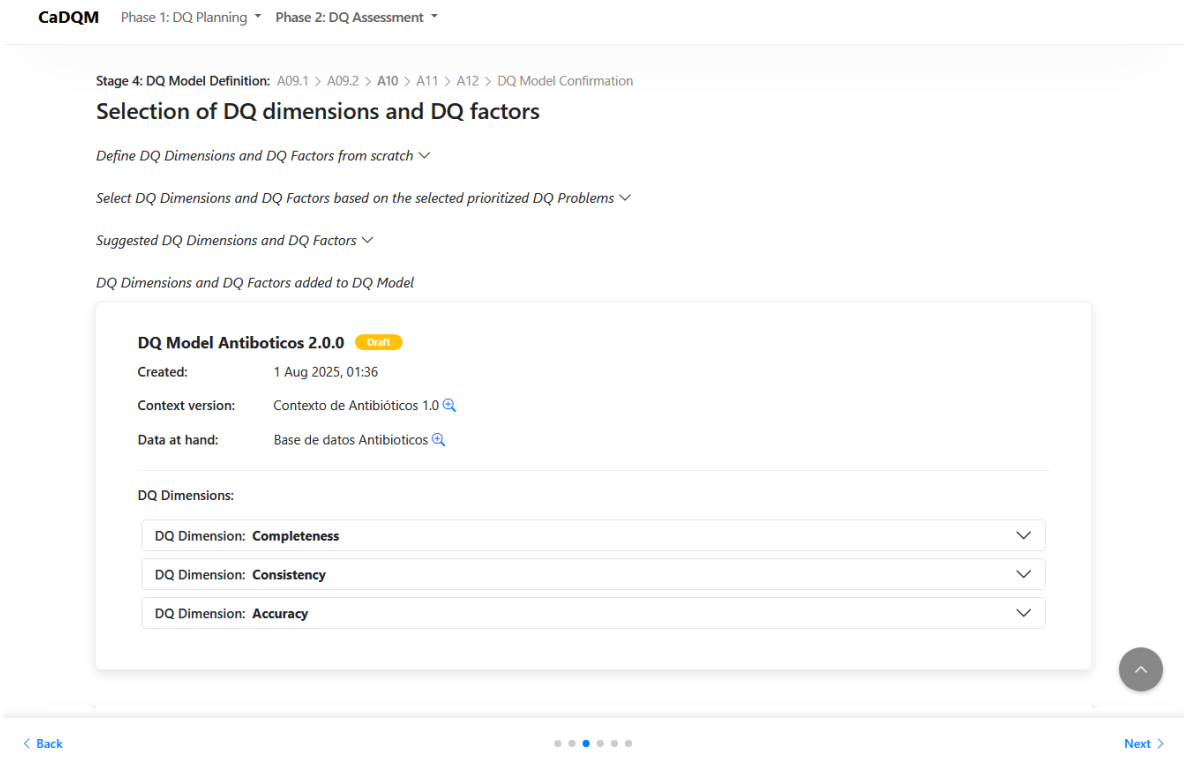

Figura A.10: Interfaz para la Selección de dimensiones y factores de CD.

Como se observa en la Figura A.10, la selección de dimensiones y factores de CD está organizada en tres secciones principales que ofrecen distintas alternativas para su realización:

- **Definición de Dimensiones y Factores de CD desde Cero** (*Sección Define DQ Dimensions and DQ Factors from scratch*): Como se observa en la Figura A.12, la interfaz especifica de esta sección se divide en dos vistas principales: una para agregar las dimensiones de CD (*Add DQ Dimension*) y otra para agregar los factores de CD (*Add DQ Factor*).
  - Dimensiones de CD (Vista *Add DQ Dimension*) (Figura A.12): Al seleccionar la opción *Add DQ Dimension*, se presentan las siguientes opciones:
    - Crear una nueva dimensión de CD: Mediante el botón *+ New DQ Dimension* (Figura A.12), se despliega un formulario (Figura A.11). El usuario debe completar los campos requeridos y la nueva dimensión de CD aparecerá en la lista desplegable.
    - Agregar dimensión de CD existente: El usuario selecciona una dimensión de CD desde la lista desplegable (*DQ Dimension*), la cual contiene todas las dimensiones predefinidas en la aplicación y las que el usuario haya creado.
    - Trazabilidad y Confirmación: Una vez seleccionada, se muestra la definición de la dimensión de CD, los componentes de contexto y los problemas de CD disponibles. El usuario asocia aquellos que sugieran la selección de la dimensión para la trazabilidad de las de-

cisiones en el modelo de CD. Para agregar la dimensión al modelo de CD, se selecciona el botón *Add to DQ Model* (ubicado en la esquina inferior izquierda, Figura A.12).

- Eliminar dimensión de CD: El usuario puede eliminar una dimensión de CD que haya creado previamente mediante un botón de eliminación que se habilita junto al de *+ New DQ Dimension*. Este botón no aparece para las dimensiones de CD predefinidas, las cuales no pueden ser borradas.

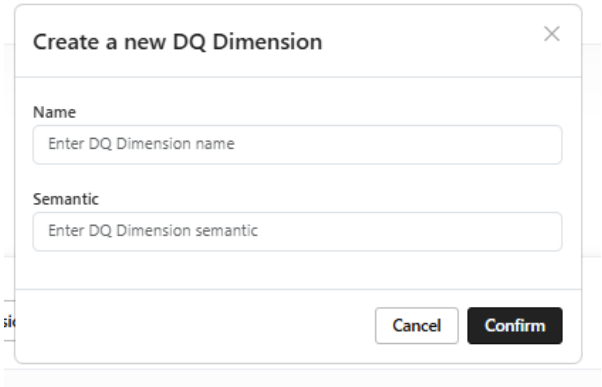

Figura A.11: Crear una nueva dimensión de CD.

- Factores de CD (Vista *Add DQ Factor*) (Figura A.13): Al seleccionar la opción *Add DQ Factor*, se presentan las siguientes opciones:
  - Seleccionar dimensión de CD: El usuario debe primero seleccionar una dimensión de CD desde una lista desplegable que solo incluye las dimensiones de CD que ya han sido agregadas al modelo de CD, para la cual desea definir o seleccionar un factor de CD.
  - Crear o agregar factor de CD: Una vez seleccionada la dimensión de CD, se listan los factores de CD asociados disponibles en la herramienta (Figura A.13).
    - Agregar factor de CD existente: El usuario puede seleccionar un factor de CD ya existente desde la lista.
    - Crear un nuevo factor de CD: Usando el botón *+ New DQ Factor*, se sigue un procedimiento análogo al de la creación de dimensiones de CD, completando el formulario correspondiente.
  - Trazabilidad y Confirmación: Una vez seleccionado o creado el Factor de CD, se muestra su definición, los componentes de contexto y los problemas de CD disponibles (Figura A.13). El usuario asocia aquellos que sugieran la selección del factor de CD para la trazabilidad de las decisiones en el modelo de CD. Para agregar el factor de CD al modelo de CD (asociado a la dimensión seleccionada), se selecciona el botón *Add to DQ Model*.
  - Eliminar factor de CD: El usuario puede eliminar un factor de CD que haya creado previamente mediante un botón de eliminación que se habilita junto al de *+ New DQ Factor* (Figura A.13). Este botón no aparece para los factores de CD predefinidos.

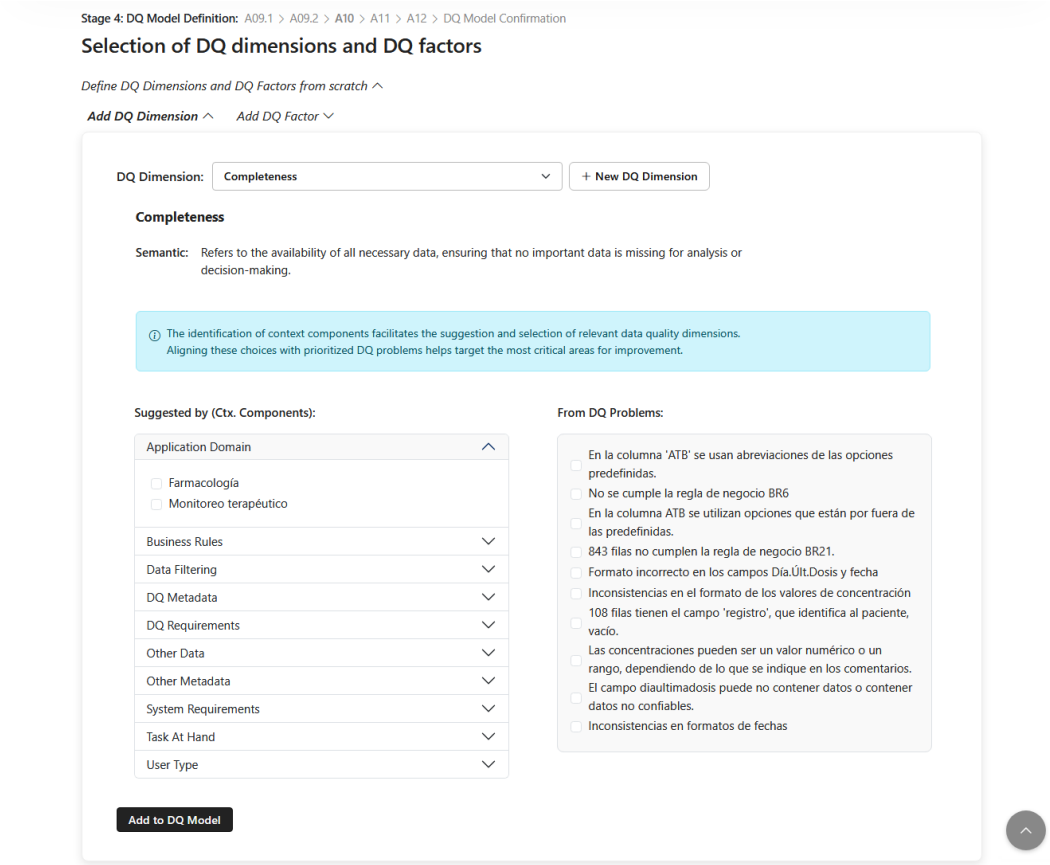

Figura A.12: Selección de dimensión de CD para agregar al modelo de CD.

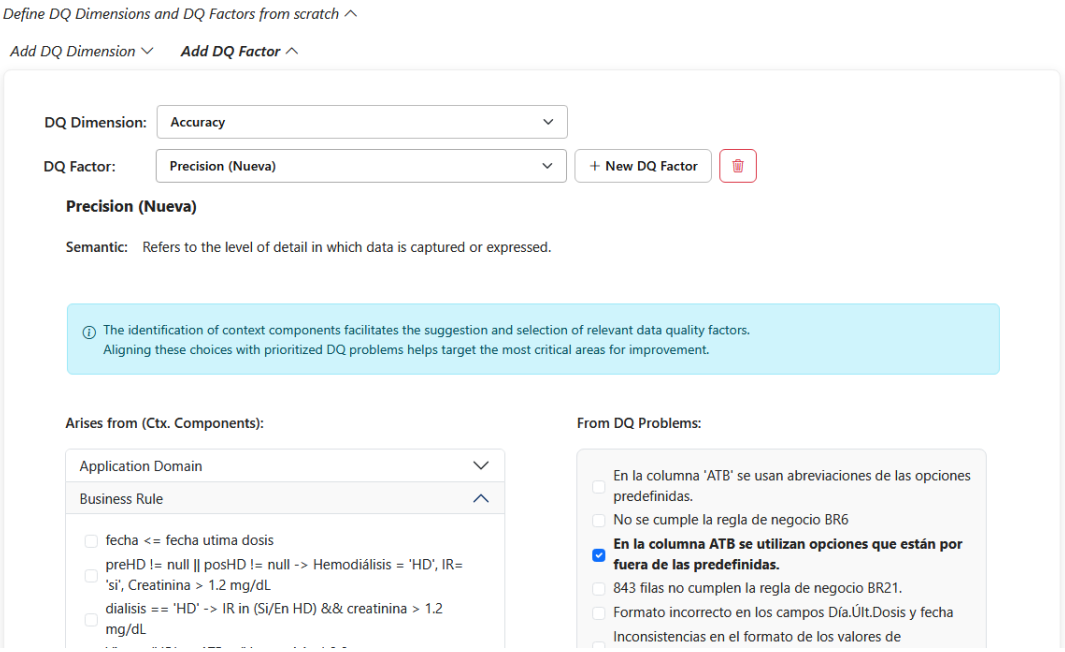

Figura A.13: Vista parcial de interfaz para selección de factor de CD para agregar al modelo de CD.

- **Selección de Dimensiones y Factores de CD a partir de Problemas de CD** (Sección *Select DQ Dimensions and DQ Factors based on the selected prioritized DQ Problems*):
  - Esta sección permite agregar una dimensión y un factor de CD a partir de un problema de CD priorizado (Figura A.14).
  - El usuario selecciona primero un problema de CD y luego selecciona las dimensiones y los factores de CD sugeridos por dicho problema de CD, que desea agregar al modelo de CD y asociar a dicho problema de CD.

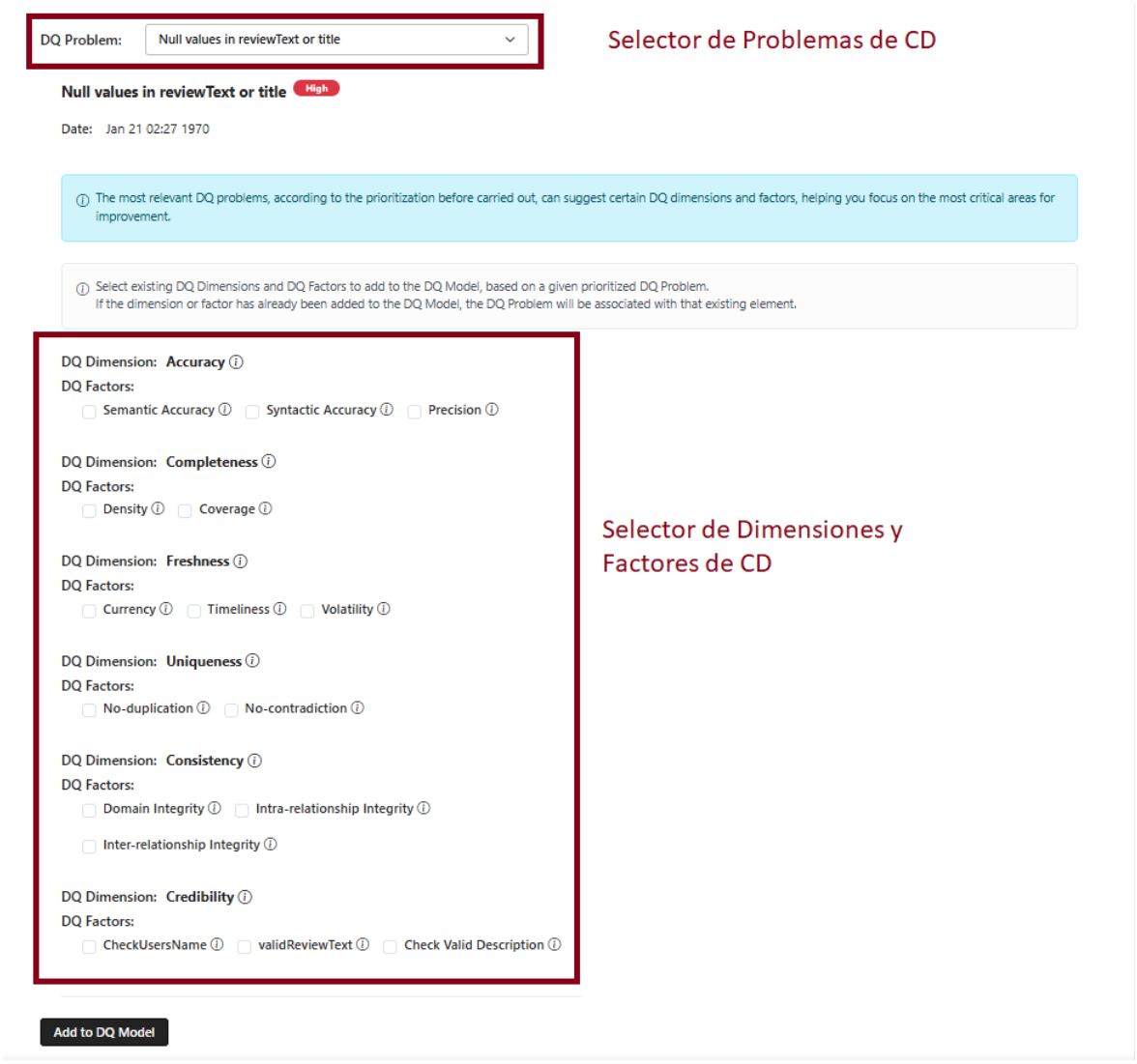

Figura A.14: Agregar una dimensión y factor de CD a partir de un problema de CD.

- Sugerencias generadas por IA (Sección *Suggested DQ Dimensions and DQ Factors*):
  - Al seleccionar el botón *Generate*, se inicia la asistencia de la IA. Primero, se presenta un modal de configuración que permite al usuario filtrar las dimensiones y respectivos factores de CD, que desea excluir de la sugerencia (Figura A.15).

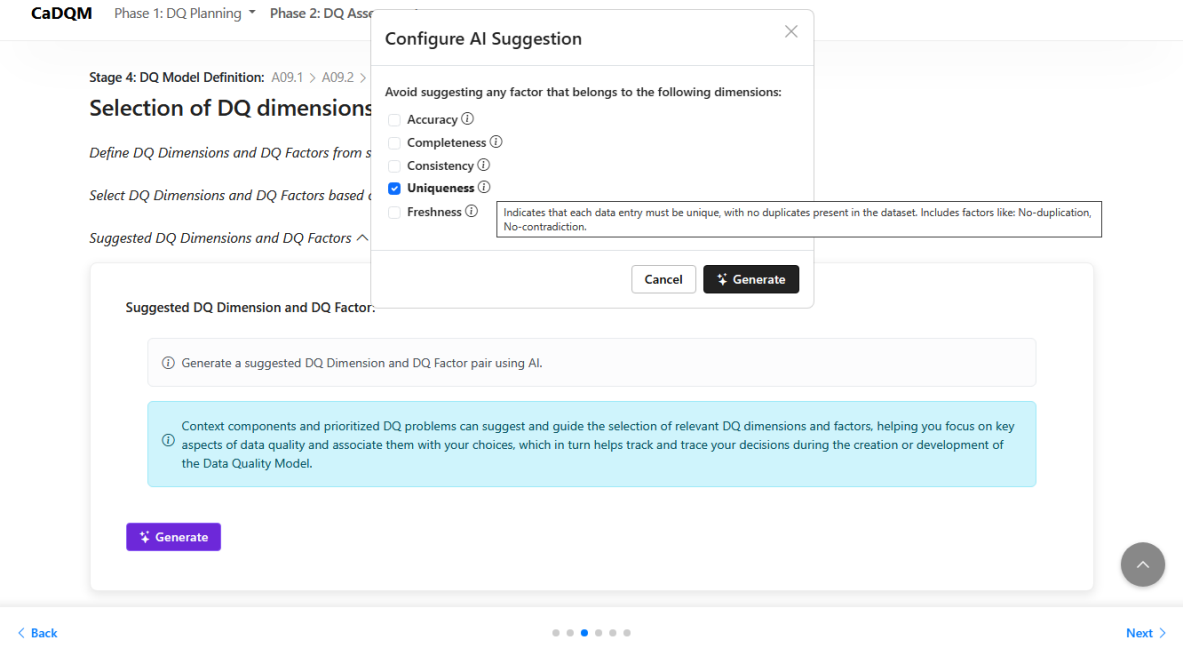

Figura A.15: Filtro para la selección de dimensiones de CD a evitar ser sugerida por la IA.

  - Posteriormente, la interfaz muestra la recomendación generada (par dimensión-factor de CD) y su fundamentación (Figura A.16). En dicha interfaz, se presentan en formato *checkbox* los componentes de contexto y problemas de CD usados para sugerir el factor de CD y su dimensión. Esta selección es completamente personalizable por el usuario, pudiendo seleccionar nuevos componentes o problemas de CD para la recomendación, así como también eliminar los utilizados, desde la interfaz de la Figura A.16. Finalmente seleccionando el botón *Add to DQ Model* puede incluir en el modelo de CD la recomendación generada.

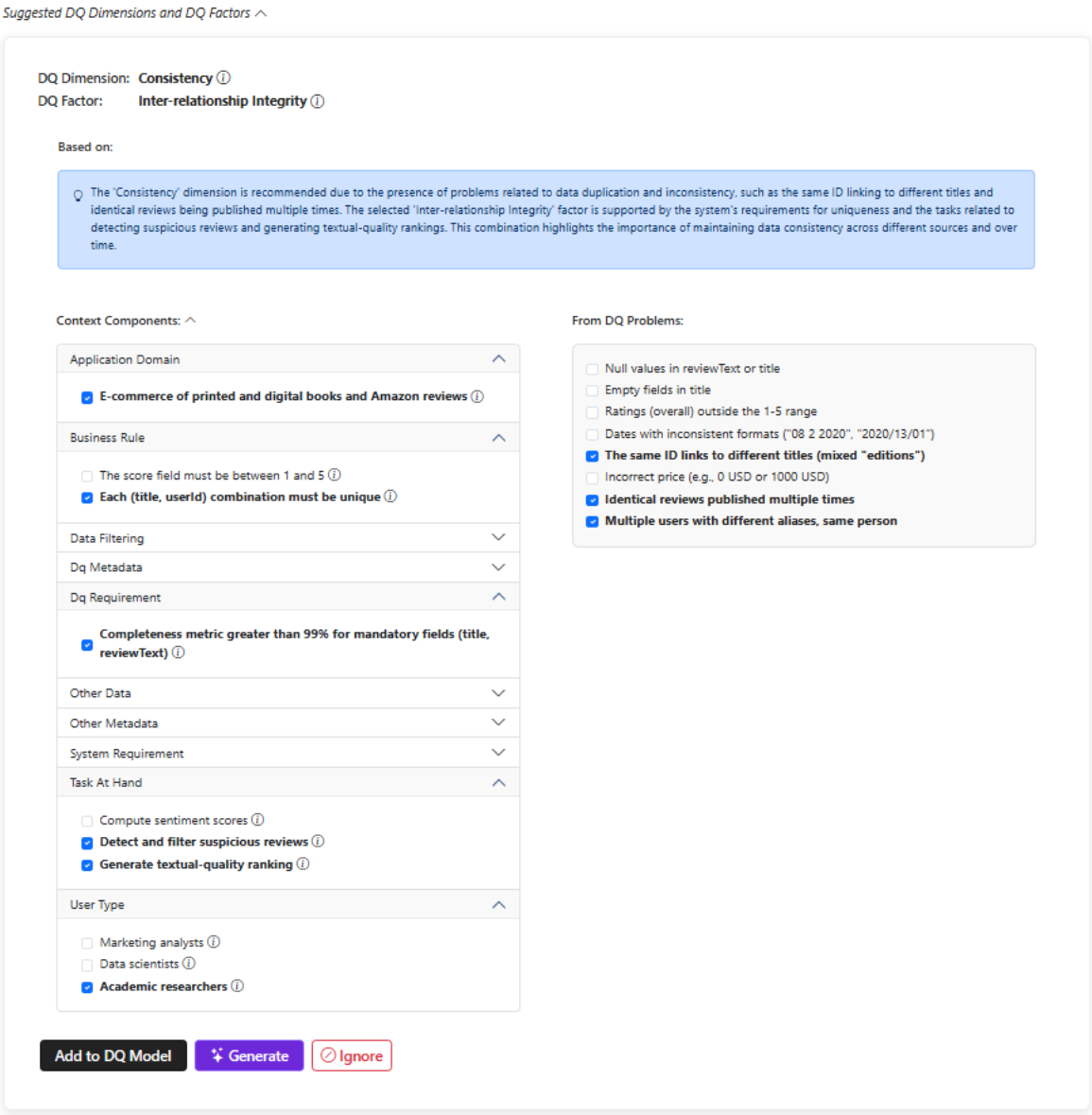

Figura A.16: Par dimensión-factor de CD recomendado por IA.

Importante: Para habilitar y poder avanzar a la siguiente actividad (*Next*), el usuario debe agregar al menos una dimensión y un factor de CD al modelo de CD. Luego podrá volver a este paso (*Back*) y seguir trabajando en esta actividad. Esta regla aplica para los siguientes pasos también, debiendo siempre definir al menos un concepto de CD en cada actividad correspondiente.

Una vez agregados los conceptos de CD al modelo de CD, el usuario puede gestionar las dimensiones y factores de CD en la sección *Dimensions and Factors added to DQ Model*, presentado en la Figura A.17, mediante las siguientes acciones disponibles:

- Revisión de la selección: En la última sección de la interfaz (Figura A.17), se muestran de forma anidada todas las dimensiones y dentro sus factores de CD, que han sido agregados al modelo de CD.

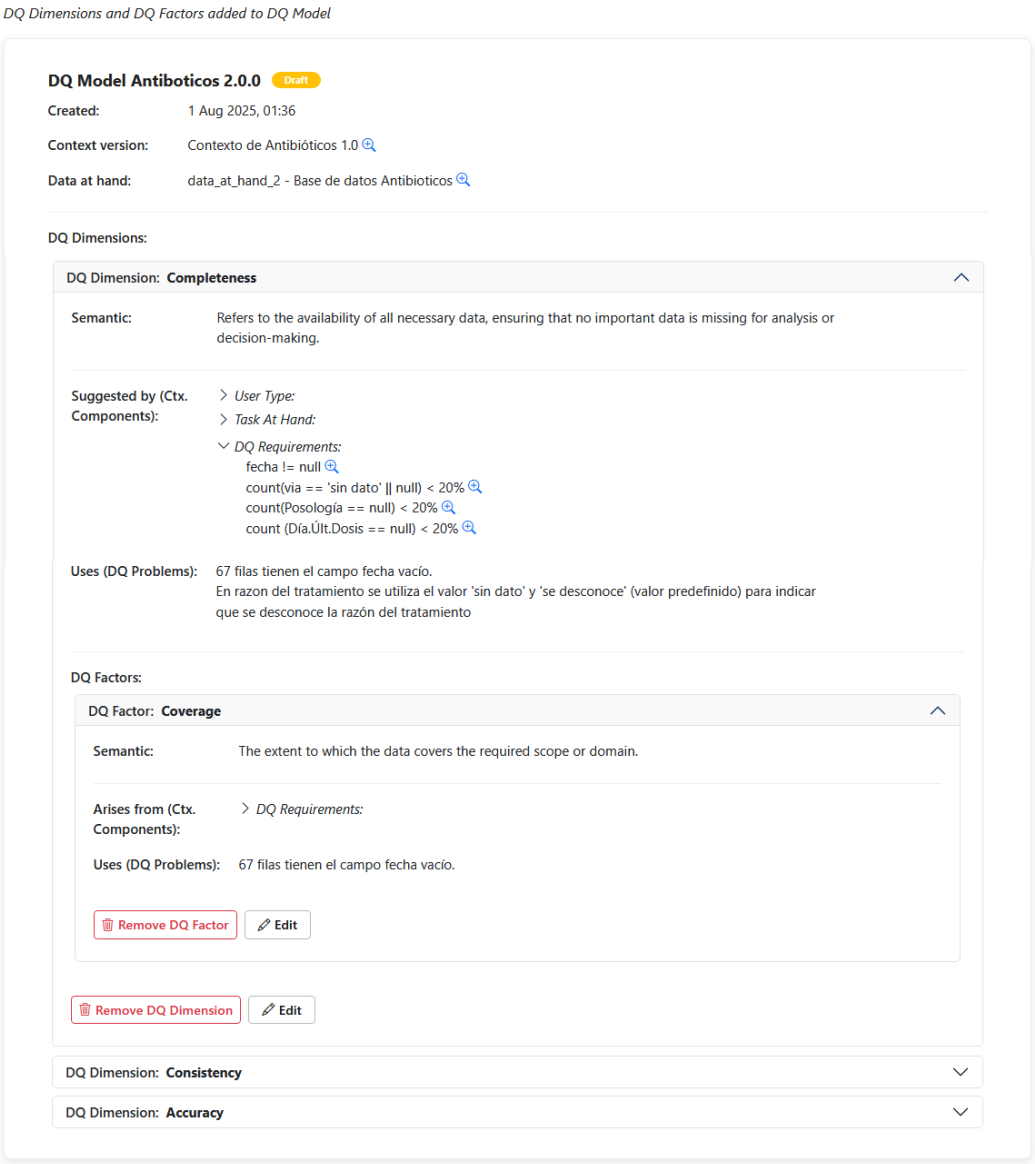

Figura A.17: Visualización de dimensión y factores de CD para modelo de CD en construcción.

- Edición: El botón *Edit* habilita la modificación de los componentes de contexto y problemas de CD relacionados con un factor o dimensión específicos (Figura A.18). Los cambios se guardan usando el botón *Save* dentro del diálogo de edición.
- Eliminación: Los botones *Remove DQ Dimension* y *Remove DQ Factor* (Figura A.17) permiten quitar la dimensión o el factor de CD correspondiente del modelo de CD.
- Confirmación de cambios: Cada acción de adición, edición o eliminación debe ser confirmada presionando la opción *Save* dentro del diálogo o formulario presentado.

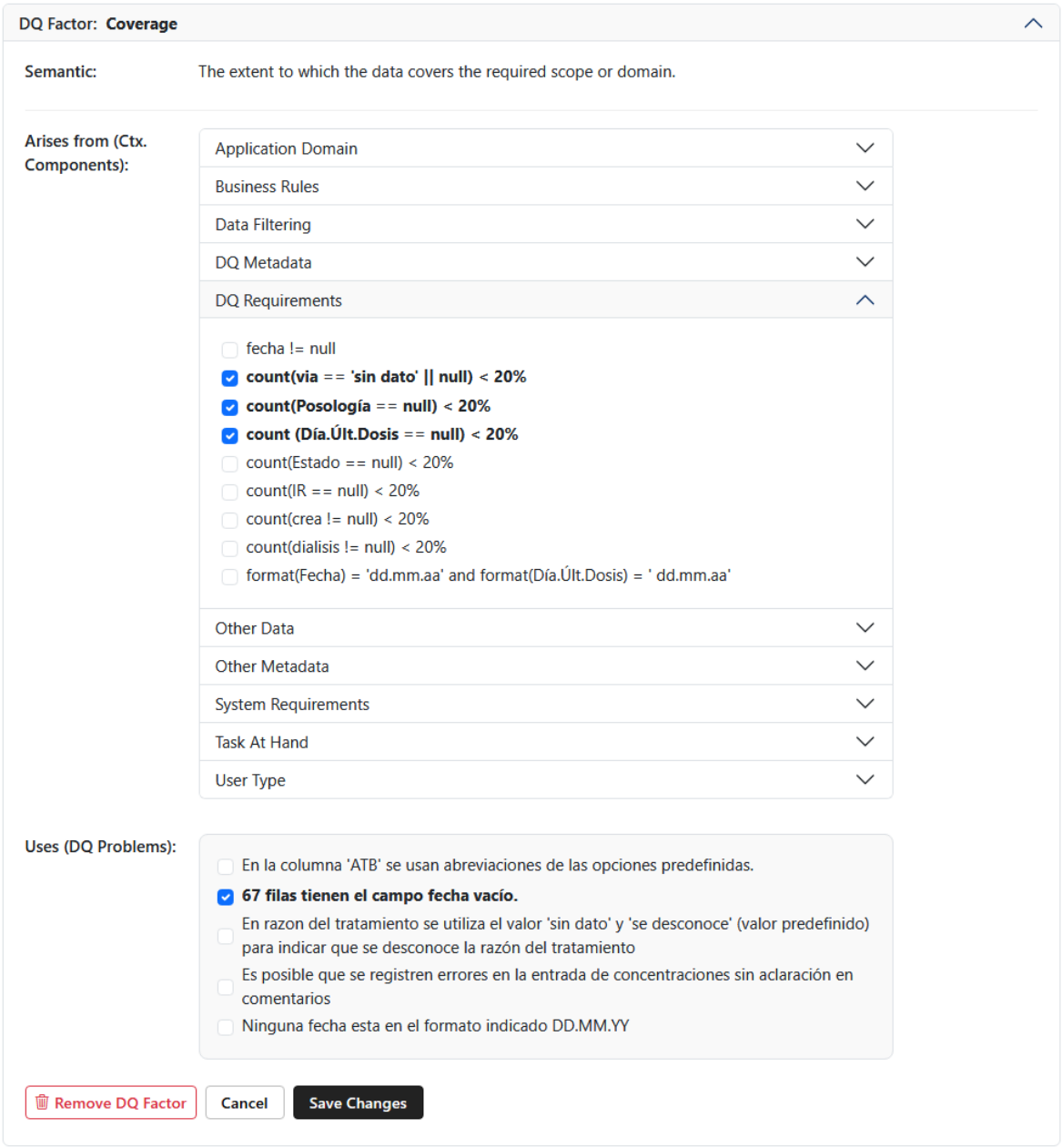

Figura A.18: Edición de componentes de contexto y problemas de CD de un factor de CD.

A.2.6. Selección de Métricas de Calidad de Datos

El próximo paso para la construcción del modelo de CD es la definición de las métricas de CD que medirán los factores de CD seleccionados en la actividad anterior.

Para agregar métricas de CD al modelo de CD, el usuario debe seguir la siguiente secuencia de acciones desde esta interfaz:

- Primero, seleccionar una dimensión de CD y luego el factor de CD específico para el cual se desea definir la métrica.
- Seleccionar una métrica de CD existente desde el selector (*select*) disponible (Figura A.19).

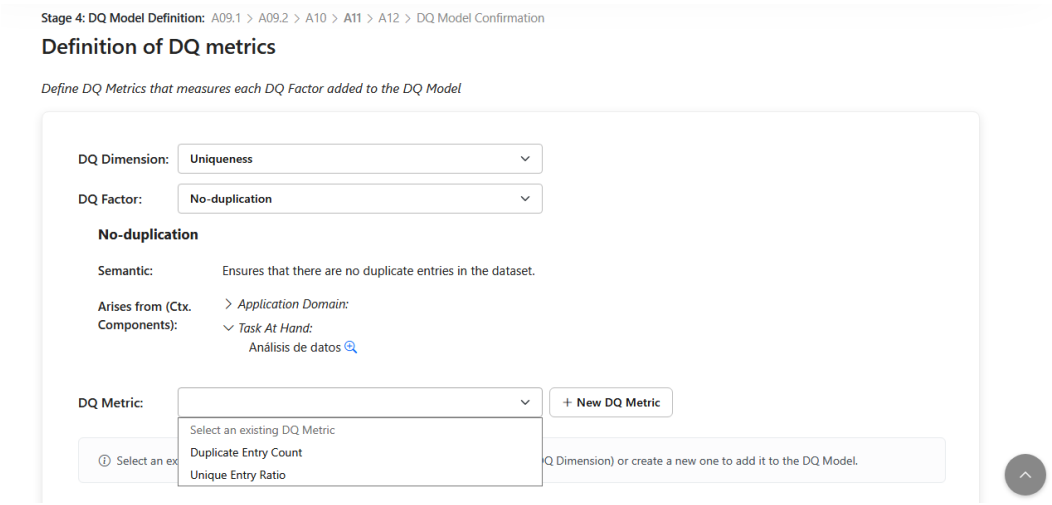

Figura A.19: Selección de Métricas de CD existentes

- Si no existe la métrica deseada, crear una nueva métrica de CD utilizando el botón *+ New DQ Metric*. Esta acción desplegará un formulario de creación (Figura A.20) donde el usuario deberá insertar todos los campos necesarios para definir la métrica de CD.

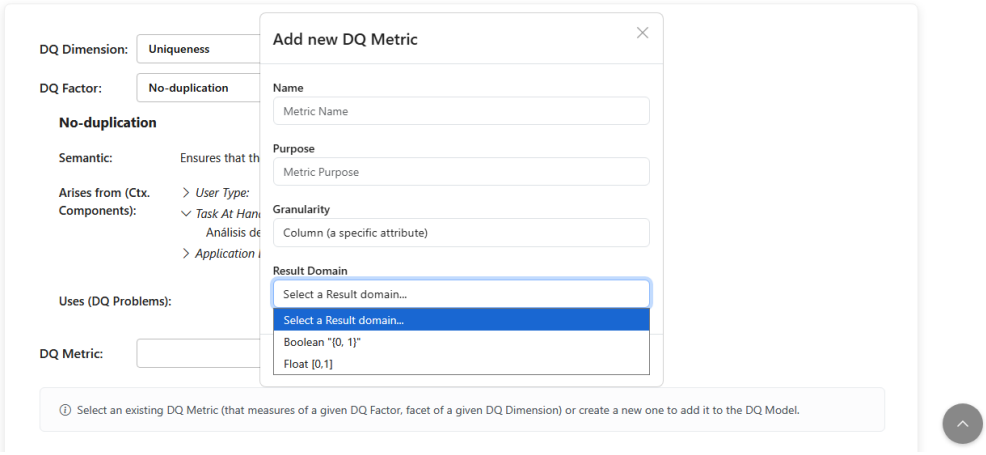

Figura A.20: Creación de Métricas de CD

- Una vez seleccionada la métrica de CD a agregar al modelo de CD, el usuario puede seleccionar los componentes de contexto (inicialmente se muestran los utilizados para el factor de CD) que de los que surge dicha selección (Figura A.21).
- Por último, el usuario debe confirmar y agregar la métrica de CD al modelo de CD utilizando el botón *Add to DQ Model*, ubicado en la esquina inferior izquierda de la interfaz de la Figura A.21.

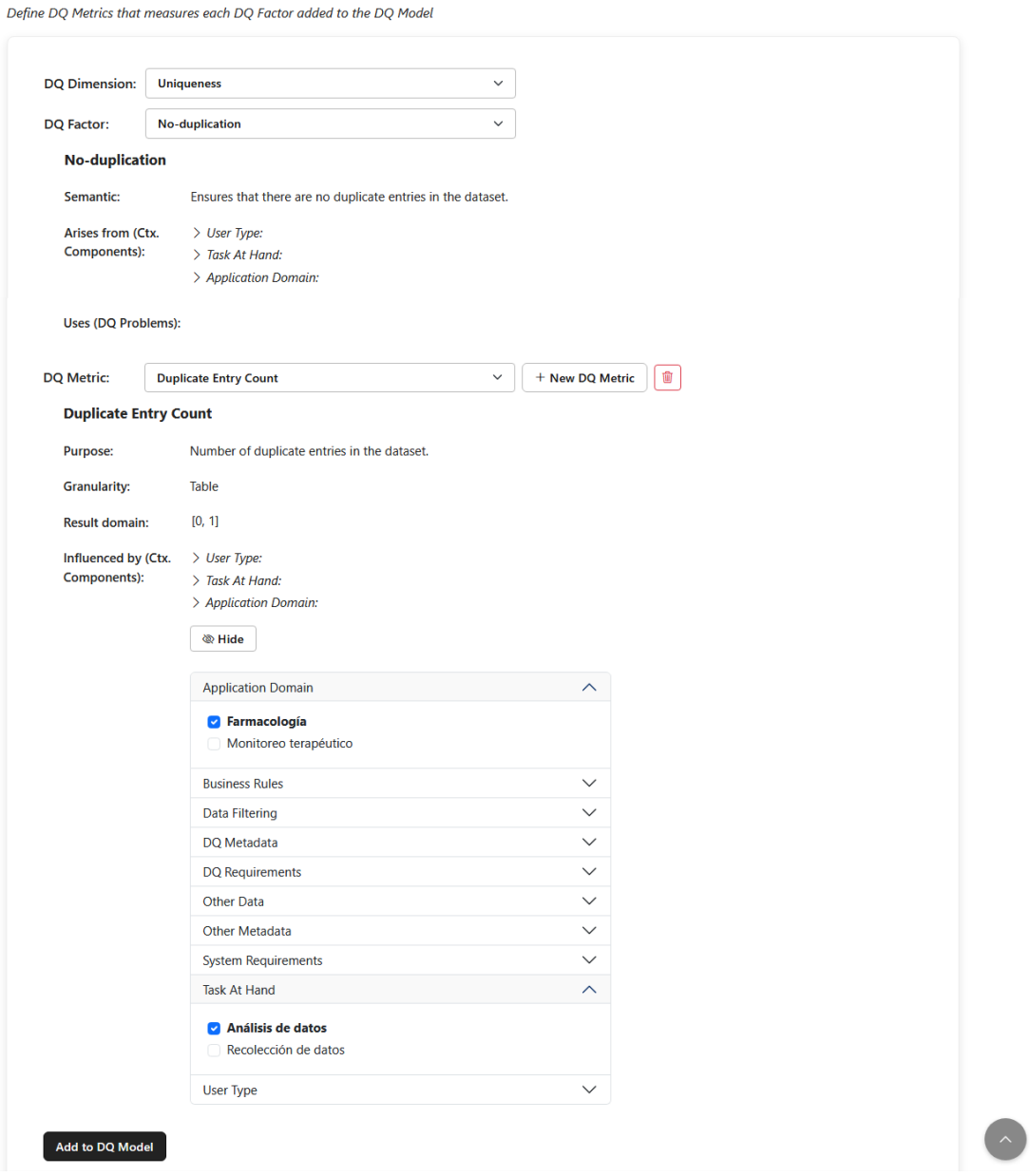

Figura A.21: Definición de una métrica de CD

Al igual que en la interfaz (Figura A.17) para la definición de dimensiones y factores de CD, se incluye una sección (Figura A.22) en la interfaz de la definición de métricas de CD que permite:

- Revisar el modelo de CD parcial: Se muestran todos los factores de CD y las métricas de CD asociadas. Esta visualización permite identificar cualquier factor de CD que haya quedado sin una métrica de CD definida.
- Edición: El usuario puede modificar los componentes de contexto y problemas de CD asociados a una métrica de CD, utilizando el botón *Edit*.
- Eliminación: Es posible quitar una métrica de CD del modelo, utilizando el botón *Remove DQ Metric*.
- Confirmación de Cambios: Toda actualización de las métricas de CD (adición, edición o eliminación) debe ser confirmada mediante el botón *Save*.

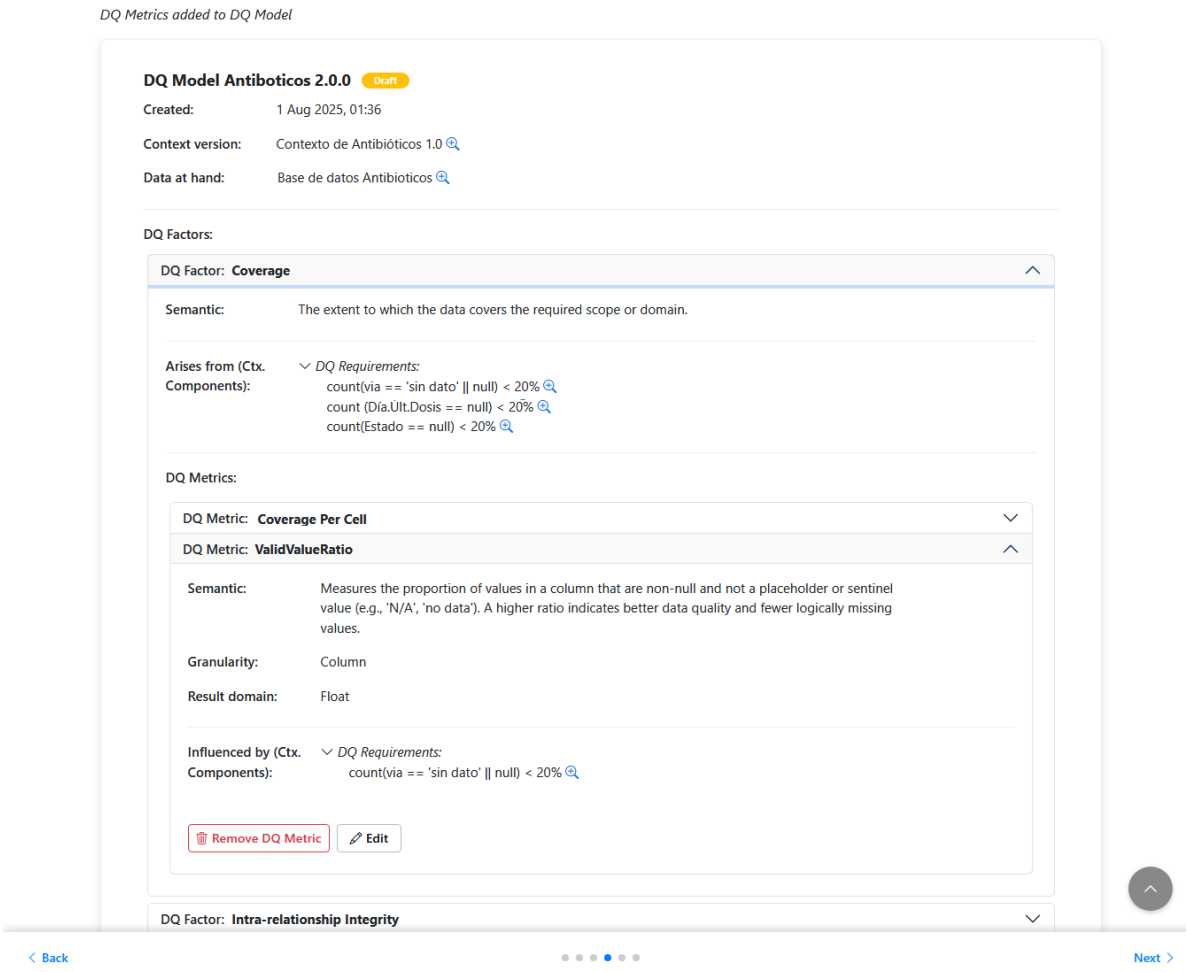

Figura A.22: Visualización de factores de CD y sus métricas de CD definidas a modelo de CD en construcción

A.2.7. Selección de Métodos de Calidad de Datos

El paso final para la construcción del modelo de CD es la definición de los métodos de CD, que implementan las métricas de CD definidas.

Para agregar un método de CD al modelo, el usuario debe seguir la siguiente secuencia de acciones desde esta interfaz (Figura A.23):

- Primero, seleccionar una dimensión, luego un factor, y finalmente la métrica de CD la cual se desea implementar mediante un método de CD.
- Agregar el método de CD mediante los siguientes pasos:
  - Seleccionar un método de CD existente asociado a la métrica de CD seleccionada, desde el selector asociado a la etiqueta *DQ Method* ubicado en la interfaz presentada en Figura A.23.

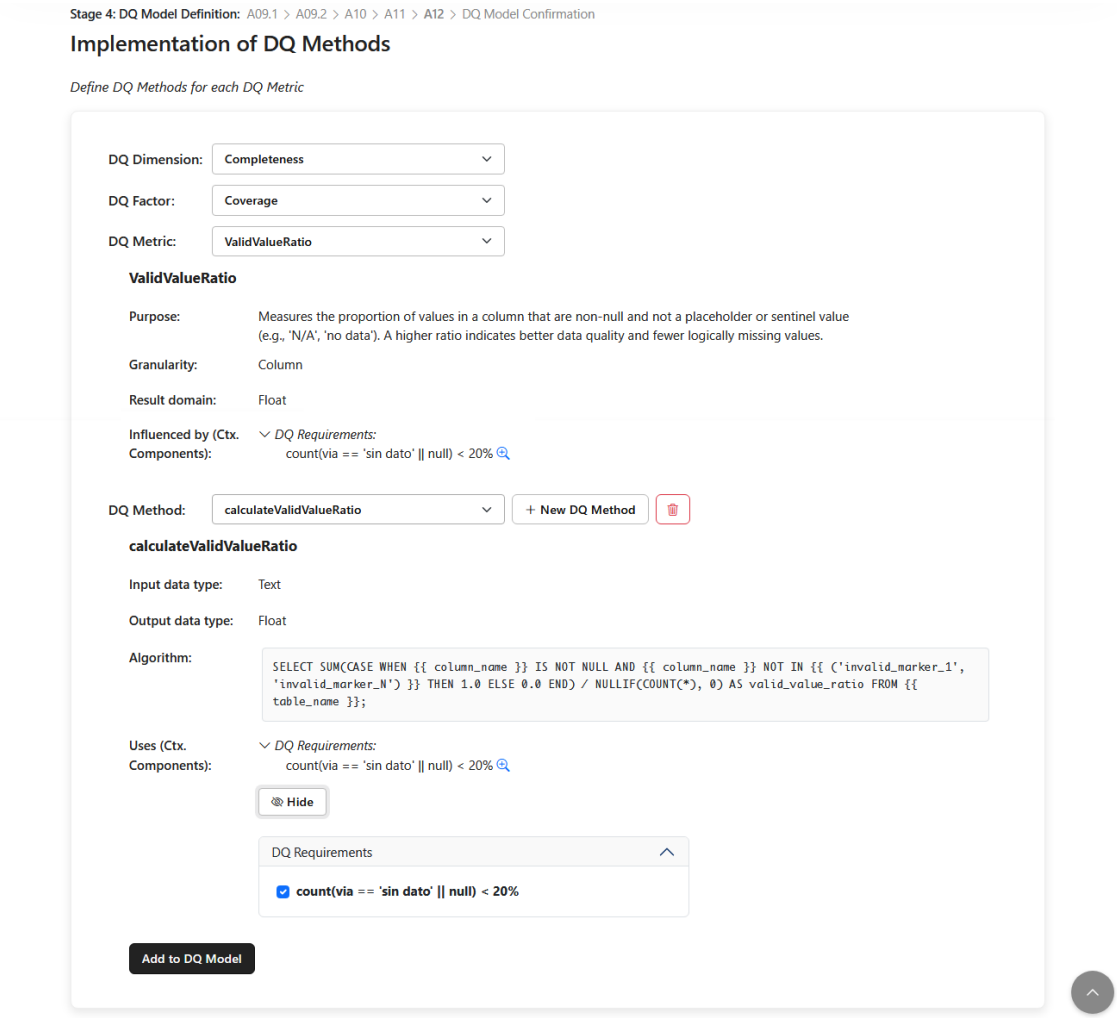

Figura A.23: Selección de métodos de CD

- Si no existe el método de CD deseado, crear uno nuevo utilizando el botón *+ New DQ Method*. Esta acción desplegará un formulario de creación (Figura A.24) donde el usuario deberá insertar todos los campos necesarios para la definición del método de CD. Se cuenta con la opción de generar o completar campos sugeridos por IA a partir de la definición de la métrica, mediante el botón *Generate* (Figura A.24), pudiendo realizar múltiples intentos, previo a confirmar la creación con el botón *Confirm*.

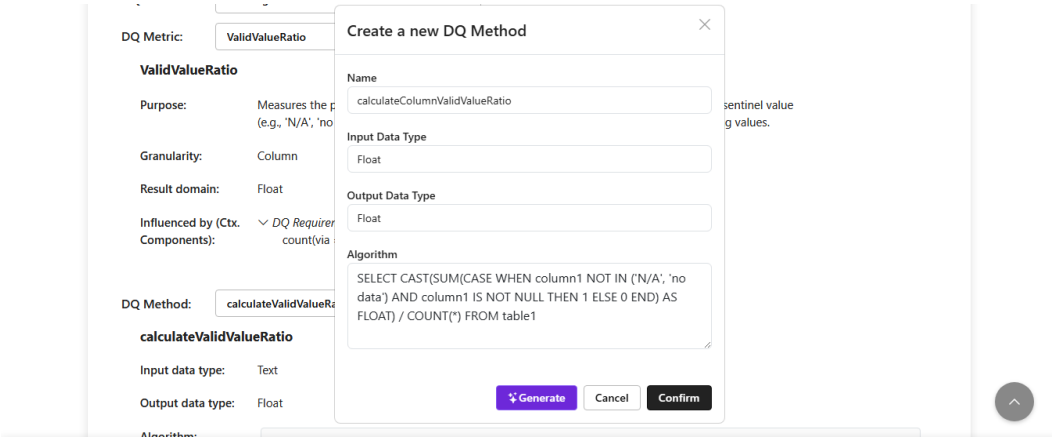

Figura A.24: Creación de un métodos de CD, con autocompletado generado por IA

- Una vez seleccionado el método de CD a agregar, el usuario puede seleccionar los componentes de contexto (inicialmente se muestran los utilizados para la métrica de CD) que sugieren dicha selección (Figura A.23).
- Por último, el usuario debe confirmar y agregar el método de CD al modelo de CD utilizando el botón *Add to DQ Model*, ubicado en la esquina inferior izquierda de la interfaz (Figura A.23).

Definición de Métodos de CD Aplicados: Luego de definir los métodos de CD, el usuario debe definir los métodos de CD aplicados. Esta acción se realiza en la sección *Define Applied DQ Methods for each DQ Methods* de la interfaz (Figura A.25).

- Primero, el usuario debe seleccionar una dimensión de CD. A continuación, se visualizarán los diferentes métodos de CD asociados a esa dimensión (Figura A.25).

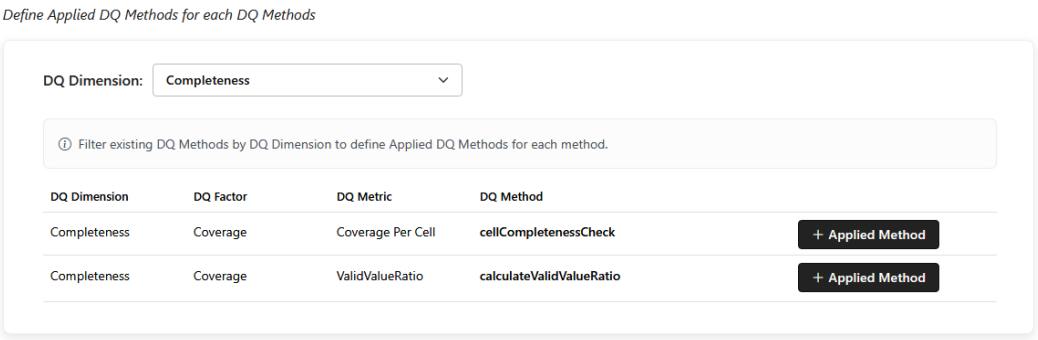

Figura A.25: Sección de la interfaz para definición de métodos de CD aplicados.

- Para definir un método de CD aplicado, el usuario debe seleccionar un método de CD mediante el botón *+ Applied Method* (Figura A.25). Esta acción despliega un formulario de creación (Figura A.26) donde se deben completar todos los campos que definen el método de CD aplicado.

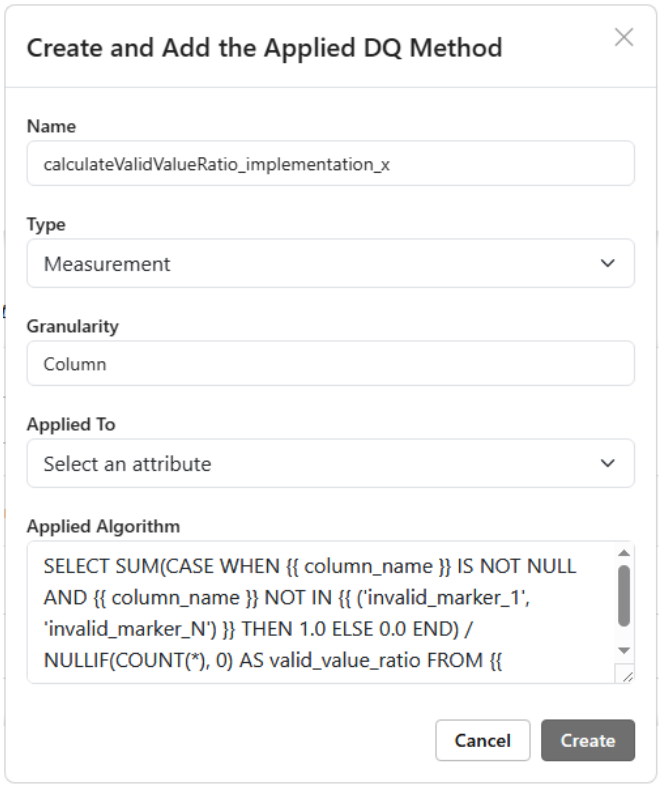

Figura A.26: Formulario de creación de un método de CD aplicado.

- En el formulario de creación del método de CD aplicado (Figura A.26) el usuario debe seleccionar o definir:
  - Tipo de método (*Measurement* o *Aggregation*).
  - Atributo o atributos del *dataset* a los que se aplicará el método.
  - Consulta `SQL` como el algoritmo instanciado que implementa el método de CD.
- El método de CD aplicado se confirma y se agrega al modelo de CD mediante el botón *Create* dentro del formulario (Figura A.26).

Al igual que en las interfaces de las actividades anteriores para la definición de dimensiones y factores (Figura A.17), y las metricas de CD (Figura A.22), la intefaz de métodos de CD incluye una sección análoga a las mencionadas que permite revisar el modelo de CD de forma parcial, pero ahora desde las métricas de CD, mostrando dentro sus métodos de CD y, en un nivel anidado, los métodos de CD aplicados. También se permite la eliminación de los métodos de CD y métodos aplicados, así como también la edición de los componentes de contexto asociados a los métodos de CD.

A.2.8. Visualización y Confirmación del Modelo de Calidad de Datos

La última actividad de la Etapa 4 consiste en la confirmación del modelo de CD. En esta interfaz (Figura A.27), el usuario realiza una revisión final del modelo de CD y lo confirma para dar por finalizada la Etapa 4 y habilitar la Etapa 5 (*DQ Measurement*). Desde esta interfaz el usuario dispone de las siguientes acciones:

- Visualizar el modelo de CD completo. La interfaz (Figura A.27) organiza los conceptos de CD primero por dimensión de CD en ítems que pueden ser expandidos. Al desplegar una dimensión, se muestra su jerarquía anidada completa: factores, métricas, métodos, y métodos de CD aplicado. Esto permite inspeccionar la definición detallada de cada concepto de CD (Figura A.2.8).
- Finalizar el modelo de CD: Una vez que el modelo de CD ha sido revisado, debe ser finalizado y confirmado utilizando el botón *Confirm DQ Model* (Figura A.27). Esta acción solo es posible si el usuario ha completado la definición de toda la jerarquía del modelo de CD incluyendo: al menos una dimensión, al menos un factor para cada dimensión, una métrica para cada factor, al menos un método de CD para cada métrica, y al menos un método de CD aplicado para cada método.
- Descargar reporte: Es posible obtener un reporte en formato PDF que contiene toda la información del modelo de CD, seleccionando el botón *Download DQ Model* (Figura A.27).

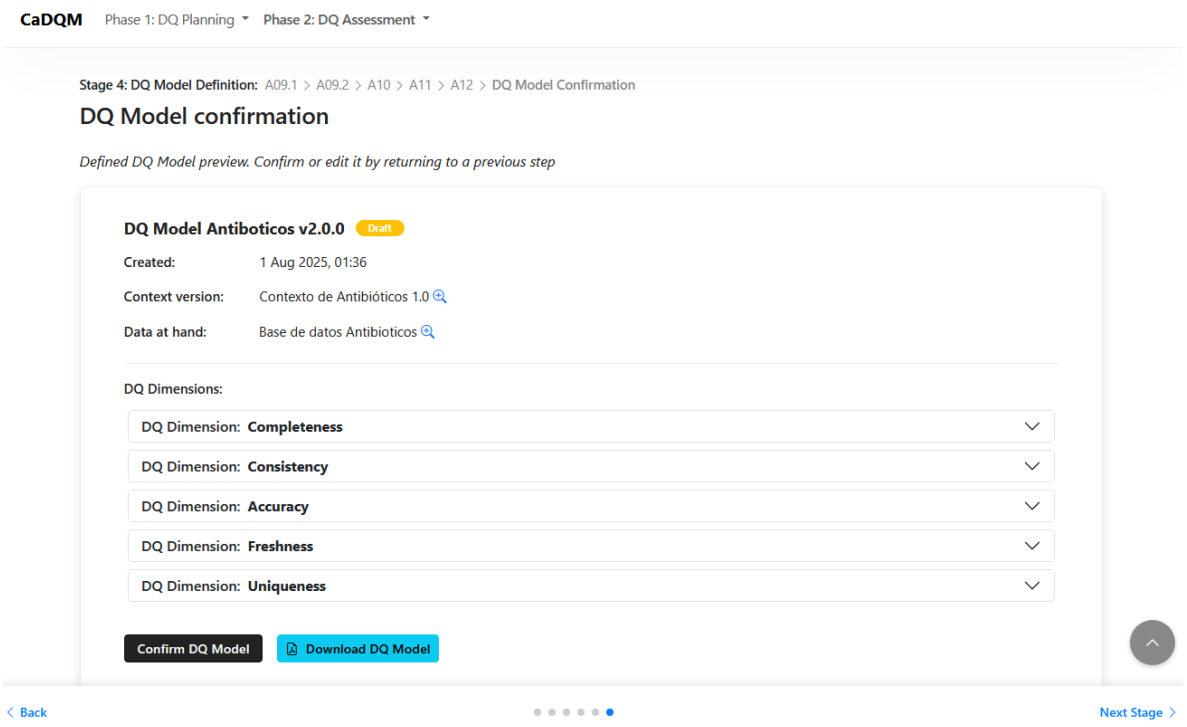

Figura A.27: Visualización del modelo de CD con las dimensiones de CD colapsadas.

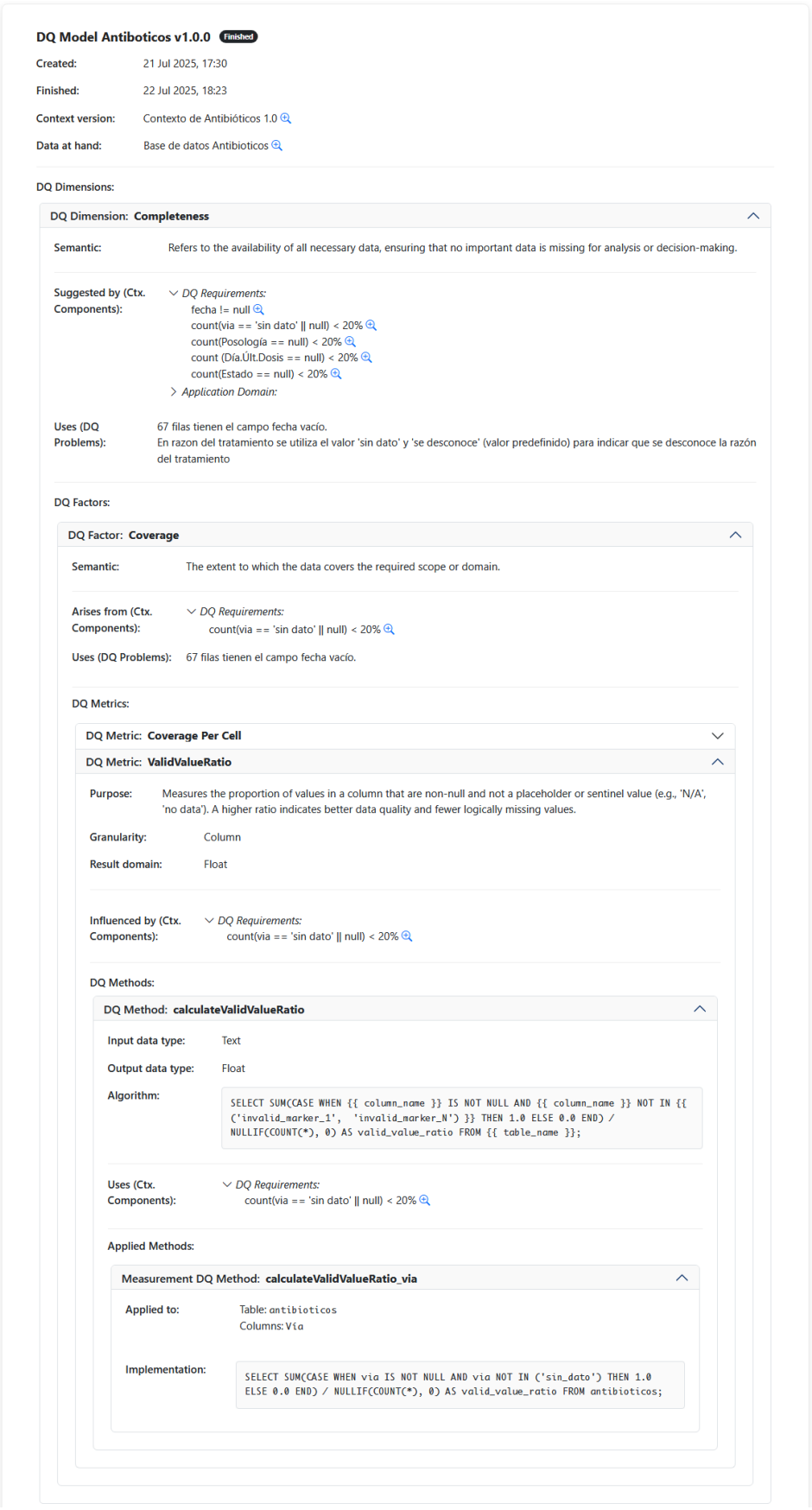

Figura A.28: Visualización completa de una dimensión de CD expandida (factores, métricas, métodos y métodos de CD aplicados) para modelo de CD finalizado.

A.2.9. Ejecución de la Medición de la de Calidad de Datos

Esta actividad marca el inicio de la Etapa 5, enfocada en la medición de la CD.

Para iniciar el proceso de medición de la CD, el usuario debe seleccionar el botón *Start Execution* en la interfaz principal (Figura A.29) para habilitar la ejecución de los métodos de CD aplicados definidos en el modelo.

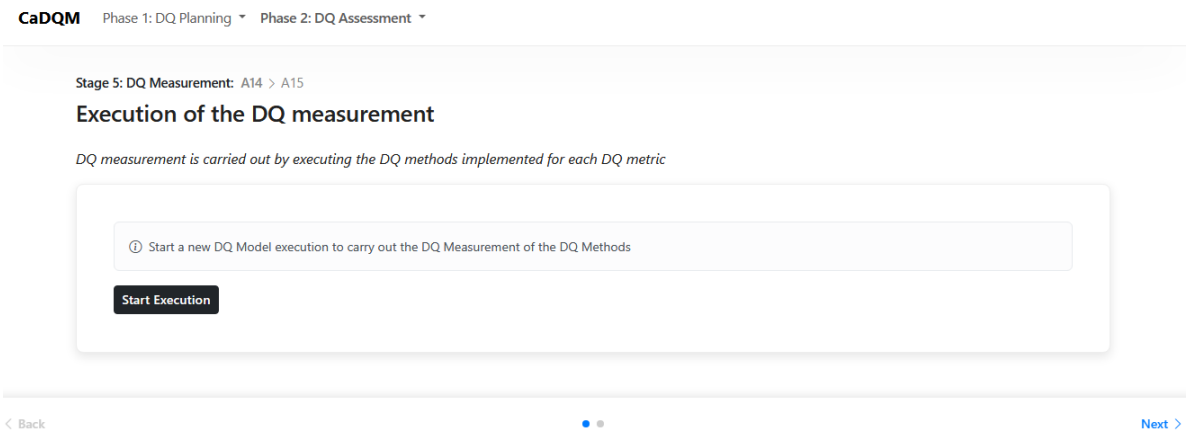

Figura A.29: Interfaz inicial de la Etapa 5 para la medición de la CD

El usuario dispone de las siguientes acciones para gestionar la ejecución:

- Filtrar Métodos de CD Aplicados: Los métodos de CD aplicados se organizan y se filtran por su estado de ejecución (*Execution Status*) en dos vistas: *Pending* y *Completed* (Figura A.30).
- Al seleccionar el filtro *Pending*, se muestran todos los métodos de CD aplicados que aún no han sido ejecutados. El usuario puede seleccionar uno o varios métodos de CD aplicados para su ejecución, utilizando el *checkbox* asociado a cada ítem.
- Para ejecutar los métodos de CD aplicados seleccionados, se pulsa el botón *Execute* (Figura A.30). Durante y al termino de la ejecución, se muestra un indicador de estado del proceso de ejecución (Figura A.31).

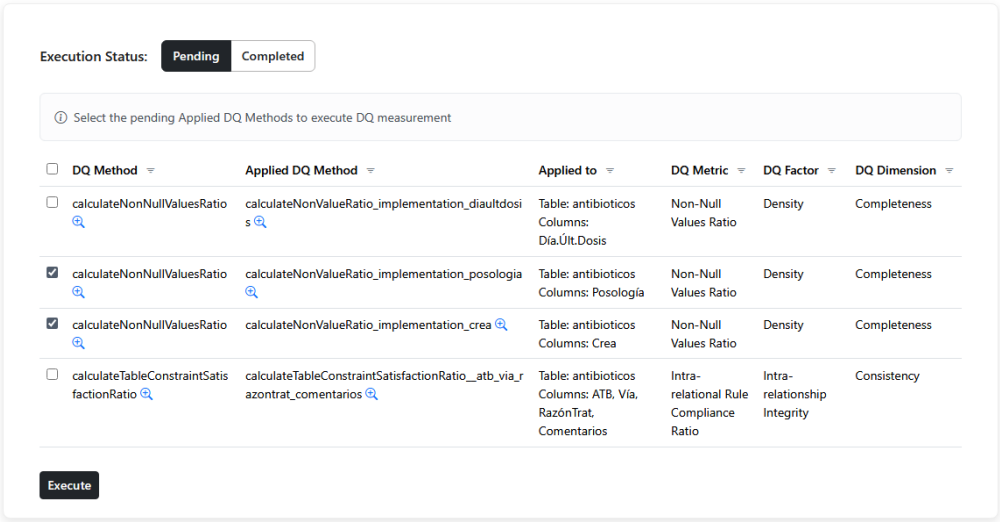

Figura A.30: Métodos de CD aplicados a ejecutar

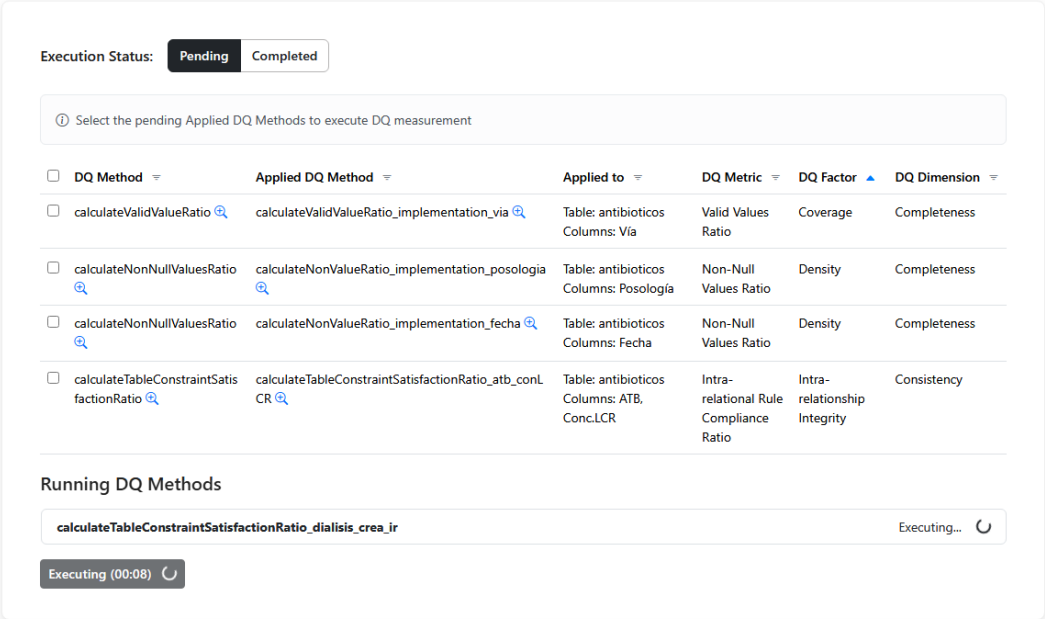

Figura A.31: Método de CD aplicado en ejecución

- Inspeccionar y Editar Método de CD Aplicado (Vista *Pending*): Desde la vista *Pending*, al hacer clic en el nombre de un método de CD aplicado (Figura A.31), se despliega un modal informativo (Figura A.32). Este muestra los detalles del método de CD aplicado y de los demás conceptos de CD relacionados según la jerarquía del modelo. Desde esta vista, es posible modificar el código que implementa el método aplicado utilizando el botón *Edit* (Figura A.32).

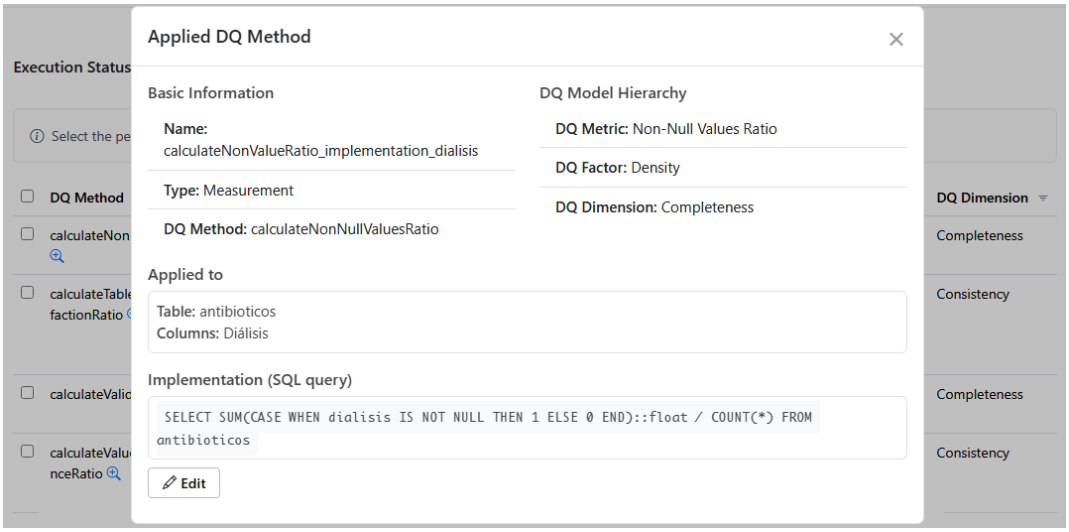

Figura A.32: Vista en modal para Método de CD aplicado sin ejecutar

- Historial de Ejecuciones (Vista *Completed*): Al presionar la opción *Completed*, el usuario puede acceder a un listado con todos las métodos de CD aplicados con ejecuciones finalizadas.

El usuario podrá acceder a la siguiente interfaz de la Etapa 5 para visualizar los resultados obtenidos aún cuando no haya completado la medición de la CD ejecutando todos los de los métodos de CD aplicados.

A.2.10. Resultados de la Ejecución de la Medición de la Calidad de Datos

En este paso se presenta una interfaz (Figura A.33) para la visualización de los resultados de las mediciones de CD ejecutadas en la actividad anterior de la Etapa 5. La vista presenta los resultados obtenidos de la medición de CD, sean estos parciales o completos (se ejecutaron todos los métodos de CD aplicados en el paso anterior).

El usuario dispone de dos vías principales para filtrar y visualizar los resultados (*DQ Measurement results*):

- Filtrar por Método de CD Aplicado: Primero se selecciona la opción(*By Applied Method*) en la interfaz (Figura A.33). Luego el usuario debe seleccionar la granularidad (*Granularity*) del método de CD y, por último, elegir el método deseado desde una lista desplegable.

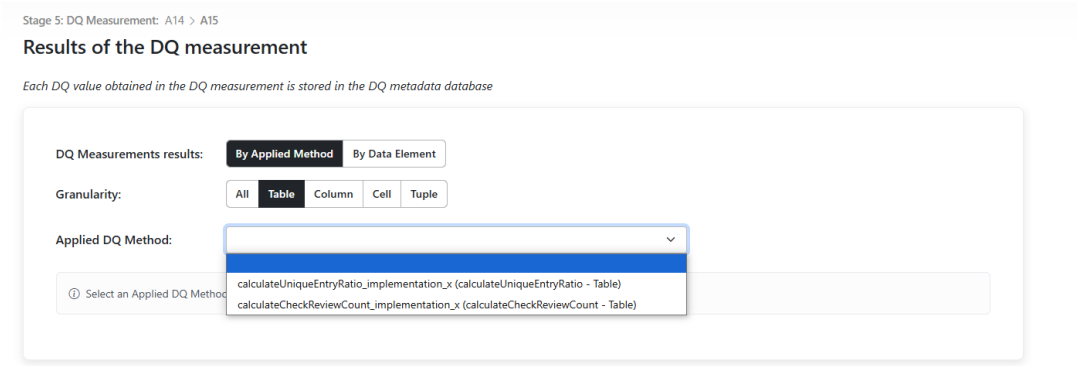

Figura A.33: Selección de un método de CD aplicado con resultados disponibles partiendo del filtro *By Applied Method*

- Filtrar por elementos del *Dataset*: Primero se selecciona la opción(*By Data Element*) en la interfaz (Figura A.34). Luego el usuario debe seleccionar la granularidad (*Granularity*), y a continuación una tabla específica del *dataset* que se está midiendo. Por último, para el cual se mostran en una tabla todos los métodos de CD que cumplan con los filtros aplicados, para la selección de uno.

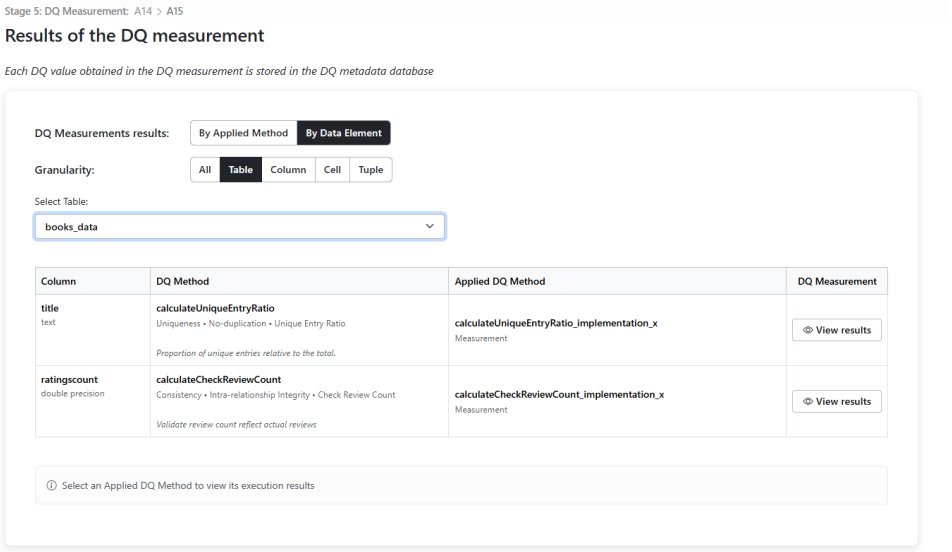

Figura A.34: Métodos de CD aplicados con resultados disponibles partiendo del filtro *By Data Element*

Una vez que se ha seleccionado un método de CD aplicado, la interfaz presenta los resultados detallados. Se muestra la información general del método (algoritmo, fecha de ejecución, granularidad y dominio del resultado) y la trazabilidad completa (dimensión, factor, métrica y método de CD).

Visualización de Resultados por Dominio: Los resultados de la ejecución se presentan de maneras distintas dependiendo el dominio de la métrica de CD:

- Métricas con dominio *Boolean* ($\{0, 1\}$): El resultado consiste en múltiples valores de CD (N), uno para cada fila del *dataset*. Inicialmente se muestran la cantidad de valores de CD en la tabla resumida con todos los conceptos de CD y una columna **DQ Value** que muestra la cantidad de valores de CD y para acceder a los resultados detallados, el usuario debe seleccionar el item *N values*, y entonces se desplegaran los valores de CD presentados en una tabla (Figura A.35) que incluye el identificador de la fila (*row id*) junto a la tabla y columna, y el valor de CD obtenido (*DQ Value*). Si el método fue aplicado sobre múltiples columnas (una tupla), la tabla solo muestra un atributo incluido como representativo.
- Métricas con dominio *Float* ($[0, 1]$): Se muestra el valor del porcentaje como un número real entre 0 y 1. Este valor se observa en la columna *DQ Value* de la tabla (Figura A.36).

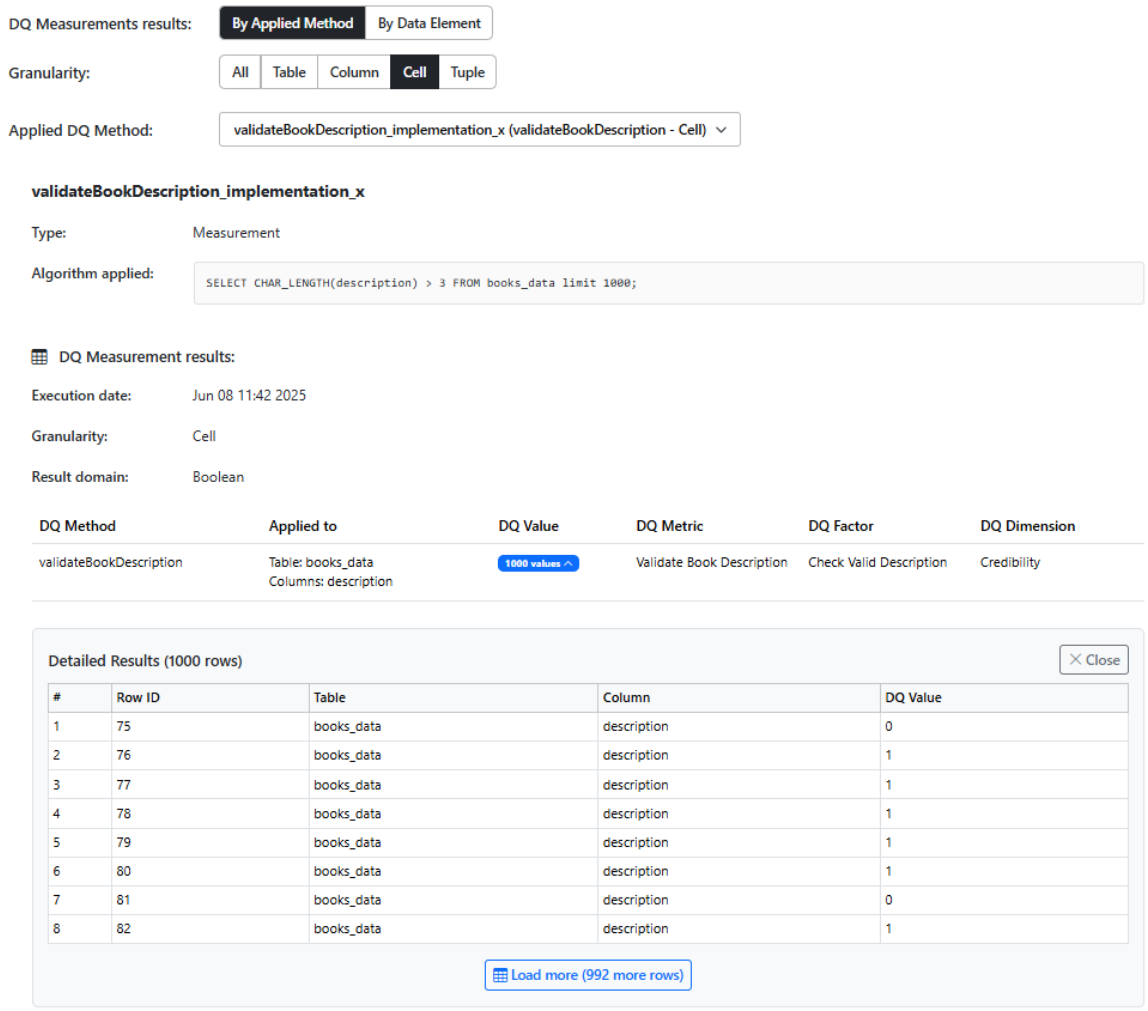

Figura A.35: Resultado de un método de CD aplicado con dominio *Boolean*.

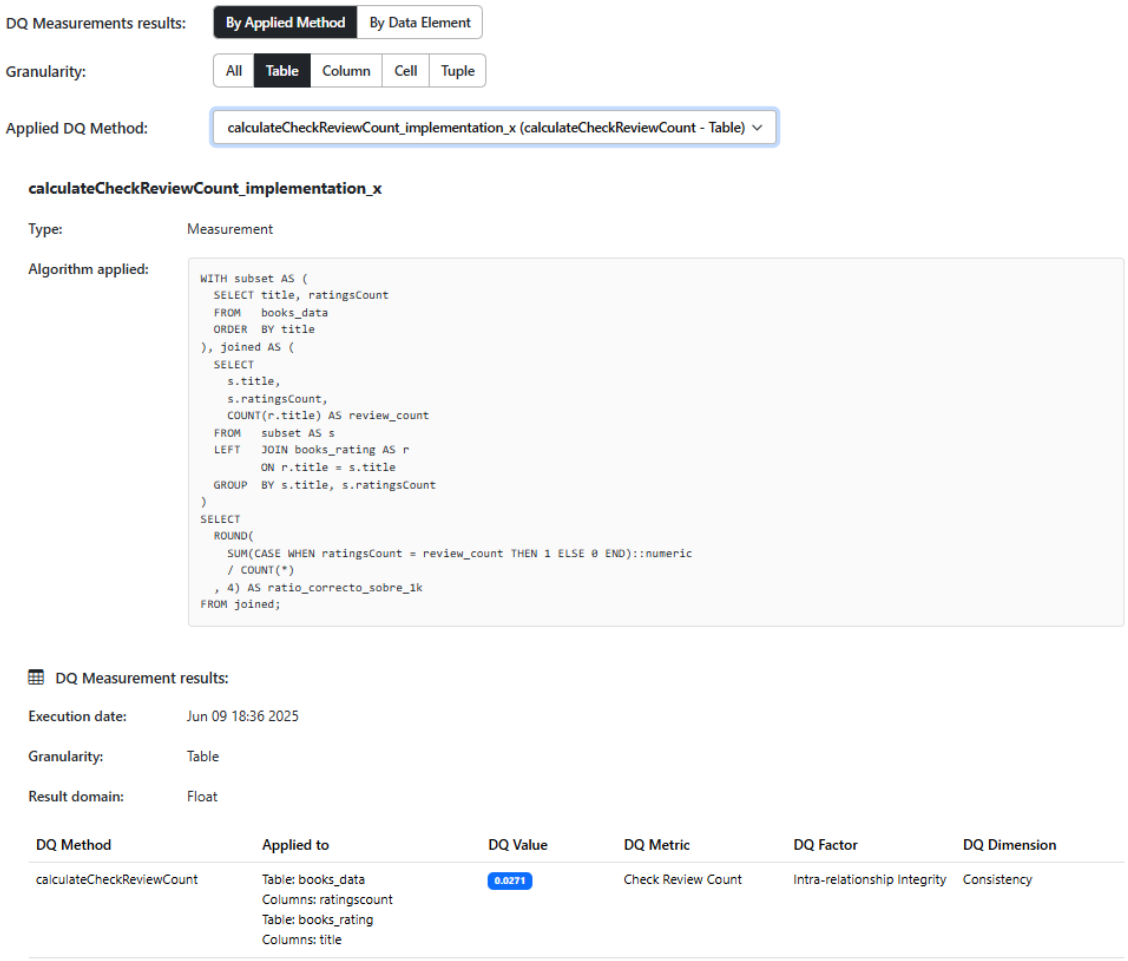

Figura A.36: Resultado de un método de CD aplicado con dominio *Float*.

A.2.11. Definición de Umbrales de Calidad de Datos

Una vez completada la medición de CD para todos los métodos de CD aplicados en la Etapa 5, se habilita la Etapa 6. Esta interfaz (Figura A.37) es donde se definen los umbrales de CD (*thresholds*) que serán utilizados para la evaluación de la CD de los resultados de las mediciones de CD.

La interfaz se divide en dos vistas según el estado de definición de los umbrales (*Thresholds definition Status*) donde el usuario puede filtrar por *Pending* (métodos de CD aplicados sin umbrales) y *Defined* (métodos de CD aplicados con umbrales).

- Para comenzar la definición de los umbrales de CD el usuario debe seleccionar el filtro *Pending* y luego elegir un método de CD aplicado desde la lista seleccionable (*Applied DQ Method*) (Figura A.37) para definir un umbral de evaluación asociado .

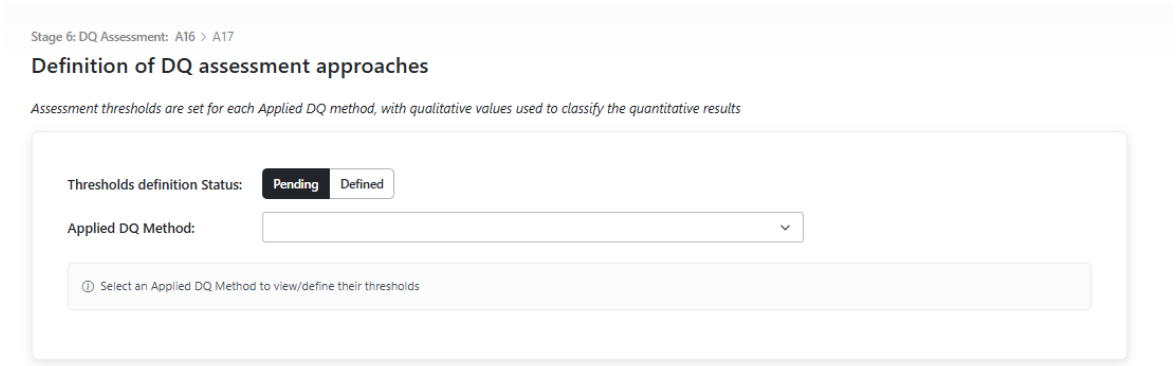

Figura A.37: Pantalla inicial para la definición de umbrales de CD

- Al seleccionar el método, se habilita una interfaz con información detallada del método, los resultados de la medición obtenidos y la sección *DQ Assessment approaches*, que se observa en la Figura A.38. En esta sección, los umbrales se definen asignando un nombre para la evaluación cualitativa (*Assessment score*) asociado a un valor mínimo (*Min Value*) y un valor máximo (*Max Value*).
- El usuario puede agregar nuevos umbrales mediante el botón *New thresholds* o quitarlos con el botón de borrar.
- El proceso definición de los umbrales varía según el dominio del resultado:
  - Umbrales de tipo *Float* (Figura A.38): Se deben introducir los intervalos deseados. Se requieren al menos dos umbrales y es obligatorio cubrir el rango completo entre 0 y 1 sin dejar valores sin clasificar (ejemplo: es válido $0 - 0{,}59$, $0{,}60 - 0{,}79$, $0{,}80 - 1$, pero no es válido $0{,}60 - 0{,}78$ y $0{,}80 - 1$).
  - Umbrales de tipo *Boolean* (Figura A.39): Se permiten solo dos umbrales predefinidos (0 y 1). El usuario solo puede ajustar el *Assessment score* (etiqueta cualitativa).
- Al hacer clic en el botón *Save Thresholds*, se confirma y guarda la definición de los umbrales. Este método de CD aplicado pasa automáticamente a la sección *Defined*.

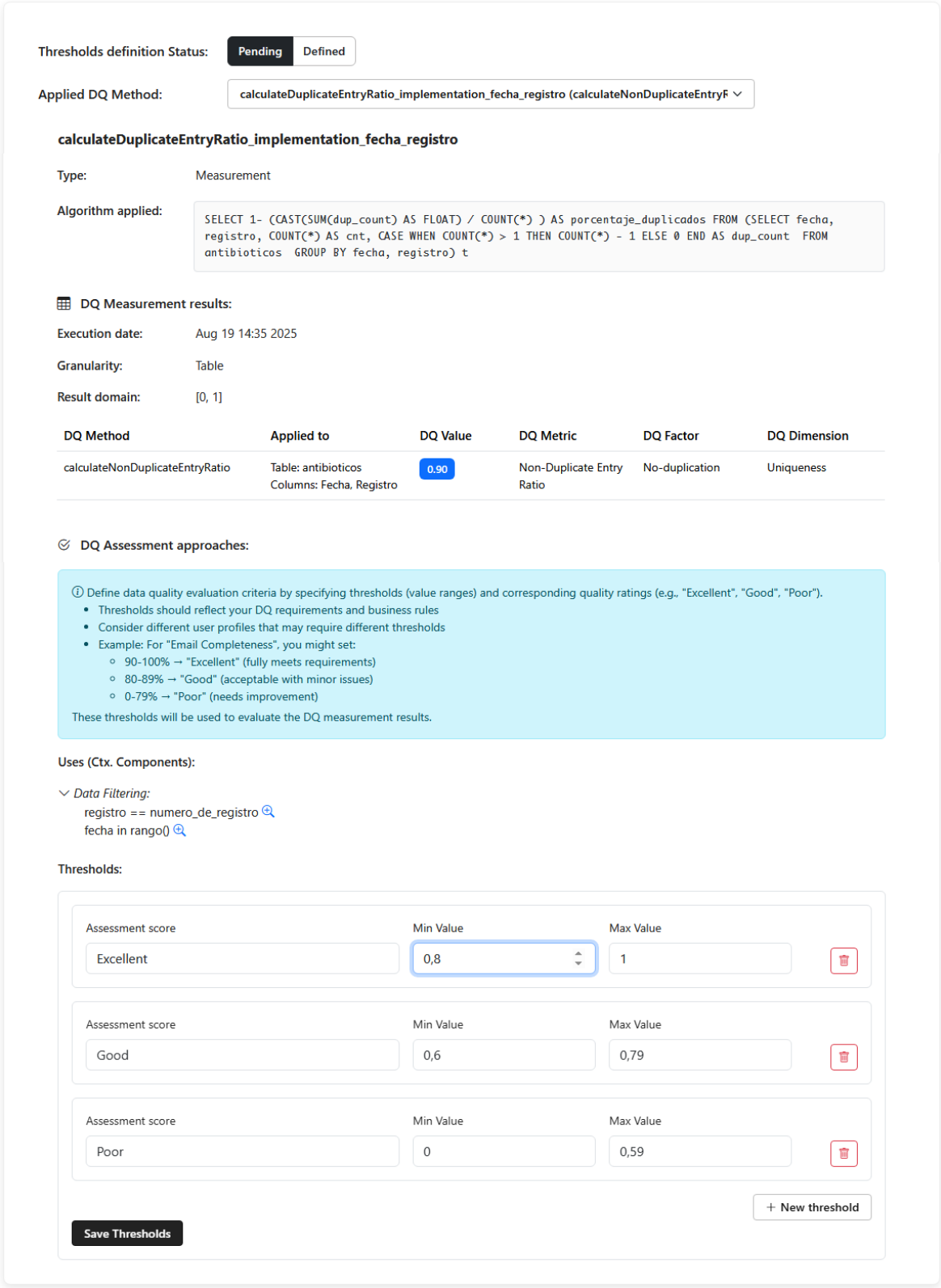

Figura A.38: Umbrales de CD de tipo *Float*

- En la sección *Defined* (Figura A.39), el usuario puede revisar y editar los umbrales definidos previo a su ejecución (ya habilitada en la interfaz de la actividad siguiente de la Etapa 5). La edición de los umbrales sigue exactamente el mismo procedimiento que el de su creación.

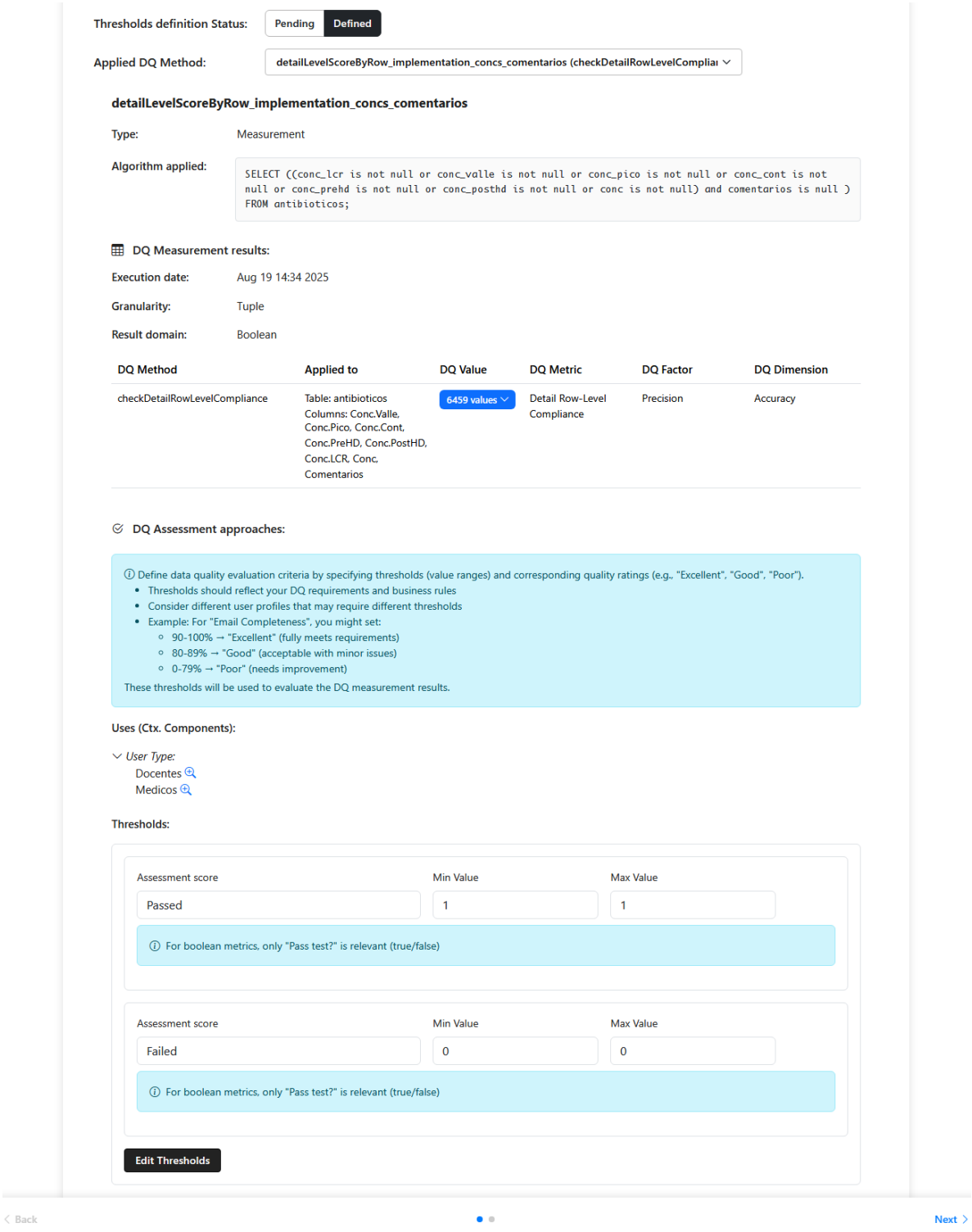

Figura A.39: Edición Umbrales de CD de tipo *Boolean* (vista *Defined*)

A.2.12. Ejecución de la Evaluación de la Calidad de Datos

Esta es la última actividad de la Etapa 6 y marca el fin de la Fase 2 - *DQ Assessment* de CaDQM. Desde esta interfaz se realiza la evaluación de la CD y se accede a la visualización de los resultados obtenidos. Para esto, la interfaz de esta actividad se divide en dos vistas según el estado de ejecución la evaluación de la CD (*Assessment Status* de los métodos de CD aplicados: *Pending* y *Completed*.

Ejecución de la Evaluación (Vista *Pending*) Esta vista permite ejecutar la evaluación de la CD para los métodos de CD aplicados que ya cuentan con un umbral definido. El usuario debe seguir los siguientes pasos:

- Primero el usuario debe seleccionar el filtro *Pending*. Luego debe elegir el método de CD aplicado a evaluar desde la lista seleccionable (*Applied Method*).
- Al seleccionar el método de CD, la interfaz (Figura A.40) se actualiza mostrando información del método, una vista previa de los resultados de medición de la CD, y la sección *DQ Assessment Configuration* con los umbrales definidos en el paso anterior.
- Para ejecutar la evaluación, en la sección *DQ Assessment Configuration* el usuario debe seleccionar el botón *Execute Assessment* (Figura A.40), a partir de lo cual se comparan los valores de CD obtenidos con los umbrales definidos para determinar el resultado de la evaluación.

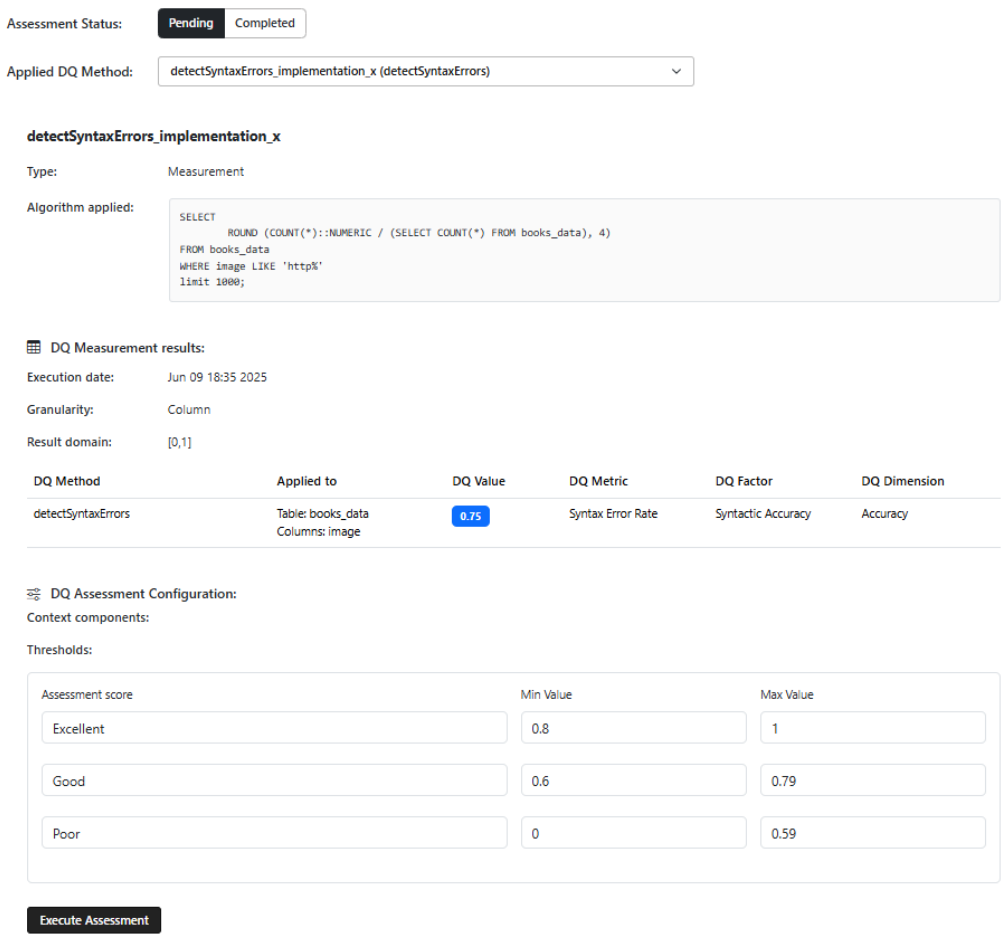

Figura A.40: Interfaz para la ejecución de la evaluación de CD de un método de CD aplicado.

Visualización de Resultados de la Evaluación de CD (Vista *Completed*): Una vez completada la evaluación, el método de CD aplicado pasa automáticamente a la sección *Completed*. Desde esta vista (Figura A.41), el usuario puede visualizar los resultados de las evaluaciones de CD realizadas. Para acceder a los resultados el usuario debe seguir los siguientes pasos:

- Primero debe seleccionar el filtro *Completed* y filtrar el resultado deseado por la opción *Granularity*.
- Luego, selecciona el método de CD aplicado cuyo resultado de evaluación de CD desea ver.
- Para la selección, se muestra una interfaz (Figura A.41) que mantiene la información del método de CD aplicado y una vista previa de las mediciones anteriores (*DQ Measurement results*), y a continuación presenta la sección *DQ Assessment results* con los resultados de la evaluación.
- La interfaz de los resultados de la evaluación de la CD varía según el dominio de la métrica asociada al método de CD:
  - Resultados dominio *Float* ($[0, 1]$): En la la sección *DQ Assessment results* (Figura A.41) se muestra la fecha de la evaluación (*Assessment date*) y el resultado de la evaluación (*Assessment Score*), que es el nombre cualitativo del umbral alcanzado (ejemplo: *Good*). El usuario puede verificar la condición del umbral en la sección *DQ Assessment Configuration*, que muestra los *Thresholds* utilizados.

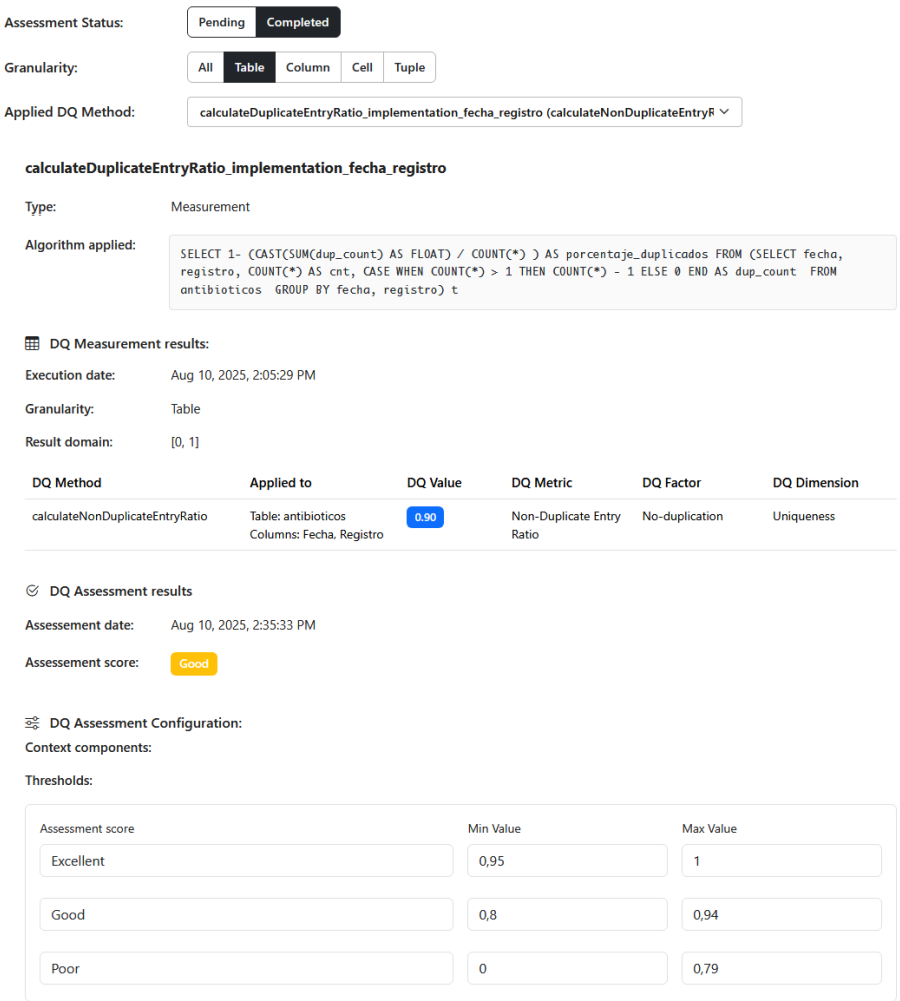

Figura A.41: Resultados de la evaluación de CD para un método de CD aplicado con dominio *Float*

- Métodos de tipo *Boolean* ($\{0, 1\}$): En la sección *DQ Assessment results* (Figura A.42) se muestra la fecha de la evaluación y una tabla con las columnas *Row ID*, *Table*, *Column(s)*, *DQ Value*, y la nueva columna *Assessment Score*. Esta última muestra la evaluación cualitativa correspondiente, utilizando un código de color (verde para resultados exitosos, rojo para no exitosos).

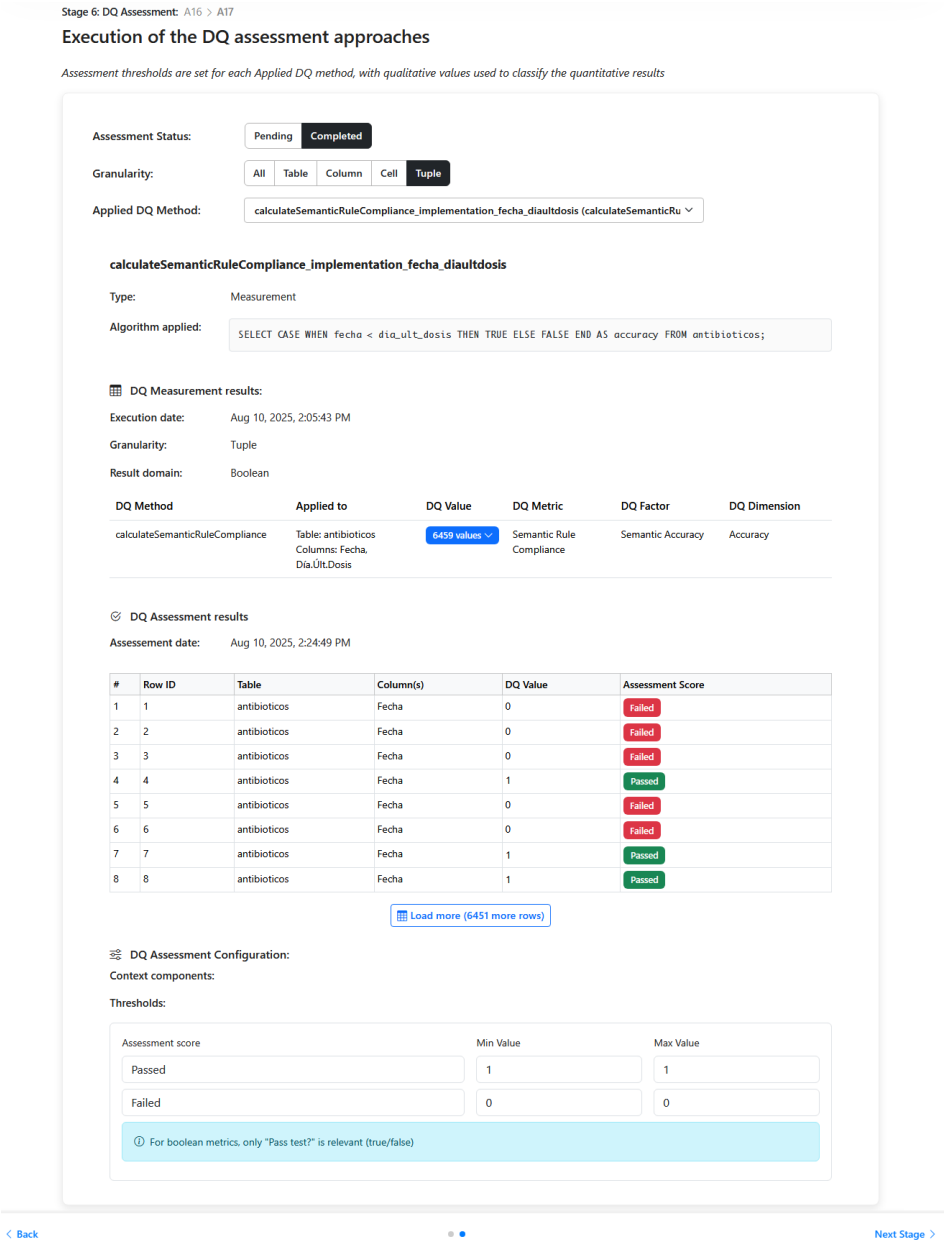

Figura A.42: Resultados de la evaluación de CD para un método de CD aplicado con dominio *Boolean*

Una vez que se ha finalizado la evaluación de CD luego de ejecutar la evaluación de todos los métodos de CD aplicados utilizando los umbrales definidos, se da por finalizada la Etapa 6 y la Fase 2 - *DQ Assessment* de CaDQM. El usuario podrá navegar, mediante el botón *Next Stage* ubicado en la esquina inferior derecha de la interfaz (Figura A.42), al *Dashboard* para decidir sus próximos pasos en la herramienta.

A.2.13. Fin de la Ejecución de la Fase 2

La ejecución de la Fase 2 - *DQ Assessment* de CaDQM, a través de la herramienta, se considera finalizada cuando el usuario ha ejecutado exitosamente las tres etapas de la fase:

- **Definición del Modelo de CD**: Se ha finalizado el modelo de CD completo (Etapa 4).
- **Medición de la CD**: Se han ejecutado todos los métodos de CD aplicados definidos en el modelo de CD (Etapa 5).
- **Evaluación de la CD**: Se ha ejecutado la evaluación de todos los umbrales de CD definidos para cada método de CD aplicado (Etapa 6).

Una vez completada la Fase 2, el usuario puede revisar todas las actividades de las etapas ejecutadas accediendo a ellas desde el menú superior de la aplicación. Para iniciar la ejecución de otro proyecto, el usuario debe dirigirse al *Dashboard* y seleccionar el botón *Close Project*. Esto le presentará la opción de seleccionar un proyecto nuevo y comenzar nuevamente el flujo de ejecución descrito en este manual de usuario.

Anexo B

Documentación Técnica

El presente anexo ofrece una visión técnica de la arquitectura e implementación de la herramienta como una aplicación web *full-stack*. La documentación se ha estructurado en dos secciones principales: la primera detalla la estructura y la lógica del servidor *backend*, mientras que la segunda se centra en la arquitectura y la interfaz del cliente *frontend*.

B.1. Backend

Esta sección describe la arquitectura y los principales componentes del servidor *backend*. Desarrollado con el *framework Django* [10] de *Python* [28], utilizando la librería *Django REST Framework* [7], este servidor provee la *API RESTful* para la aplicación. A continuación, se detalla su estructura principal, aspectos de configuración, los modelos de datos, construcción de los *endpoints* de la *API* y la integración con servicios externos.

B.1.1. Arquitectura General y Estructura del Proyecto

El *backend* consiste de un proyecto *Django*, que sigue el patrón de arquitectura de aplicaciones propuesto por el *framework*. La lógica del *backend* está modularizada en dos aplicaciones principales, lo que permite una clara separación de responsabilidades:

- `dqmodel`: Contiene y gestiona toda la lógica necesaria para la definición de modelos de CD. Además, se encarga de la ejecución de las mediciones, la evaluación de los resultados y su almacenamiento en una base de datos independiente.
- `project`: Integra la lógica común y transversal de la aplicación, gestionando la entidad principal "Proyecto"que articula y relaciona todos los componentes centrales: los modelos de CD (definidos en `DQModel`), la definición del contexto, los problemas de CD y los datos fuente. Sirve como contenedor lógico para el flujo de trabajo completo de las fases 1 y 2 de CaDQM.

La configuración global y la orquestación de estas aplicaciones se gestionan desde el directorio `myproject/`.

Estructura Interna de las Aplicaciones *Django* (`apps`)

Cada aplicación *Django* sigue el patrón Modelo-Vista-Template (MVT) adaptado para una *API REST*, lo que resulta en la siguiente estructura de archivos:

- `models.py`: Define las entidades del dominio (ej: `User`, `Project`, `DQModel`), que se mapean automáticamente a tablas en la base de datos mediante el ORM *(Object-Relational Mapper)* de *Django*.

- `serializers.py`: Contiene los serializadores de *Django* REST Framework, que transforman los datos de los modelos en formato *JSON* para la *API*, y viceversa (validación y deserialización de datos de entrada).
- `views.py`: Gestiona las vistas, que implementan la lógica de controlador para procesar las peticiones *HTTP*, interactuar con los modelos y devolver las respuestas *JSON* correspondientes.
- `urls.py`: Define los patrones de URL específicos de la aplicación y enruta las peticiones entrantes hacia sus vistas correspondientes.

B.1.2. Configuración Global

La configuración del servidor se centraliza en el directorio `myproject/`, siendo el archivo principal `settings.py`. Este archivo define el comportamiento fundamental del proyecto y se complementa con otros archivos clave de configuración.

- `settings.py`: Define la configuración global del proyecto, incluyendo las aplicaciones instaladas (`INSTALLED_APPS`), la conexión a múltiples bases de datos (`DATABASES`), middlewares, y políticas de seguridad y *CORS*.
- `urls.py`: Configura el enrutamiento URL principal del proyecto, integrando y distribuyendo las peticiones hacia los archivos de URLs de cada aplicación (`project/urls.py`, `dqmodel/urls.py`).
- `wsgi.py` & `asgi.py`: Puntos de entrada para el despliegue del servidor de aplicaciones.

Bases de Datos

El sistema utiliza dos bases de datos *PostgreSQL* con propósitos distintos:

- `default`: Base de datos por defecto de la aplicación, que almacena todos los datos operativos y de configuración del sistema.
- `metadata_db`: Base de datos dedicada exclusivamente para almacenar de forma separada los resultados generados durante las mediciones y evaluaciones de CD.

Para gestionar el acceso específico a la base de datos de metadatos, se implementó un router personalizado (`MetadataRouter.py`) que dirige las operaciones de los modelos relacionadas a la ejecución de medición y evaluación de los datos, hacia `metadata_db`, mientras que el resto de los modelos utilizan la base de datos por defecto.

B.1.3. *Endpoints* de la *API*

La *API* expone *endpoints* RESTful organizados según las entidades gestionadas por cada aplicación. Los *endpoints* se estructuran bajo el prefijo `/api/` e incluyen principalmente operaciones CRUD estándar para cada recurso, implementadas mediante `ViewSets` de *Django REST Framework*, así como otros *endpoints* personalizados para diferentes propósitos.

Los *endpoints* son generados en cada aplicación (`project` o `dqmodel`) según la lógica de negocio correspondiente. Entre algunos de los *endpoints* principales se presentan:

- Aplicación `project`: Gestiona los recursos centrales del sistema y el contexto de los datos. Sus *endpoints* principales bajo `/api/` incluyen:
  - `/projects/`: CRUD de proyectos.
  - `/contexts/`: Gestión de las versiones de contexto definidas para su asociación a proyectos.
  - `/contexts/<id>/context-components/`: Obtención de los componentes de contexto asociados a una versión específica.
  - `/dq-problems/`: Registro y gestión del catálogo de problemas de CD disponibles.

    - `/projects/<id>/quality-problems/`: Gestión de los problemas de CD específicos asociados a un proyecto particular.
    - `/data-at-hand/`: CRUD de conexiones a bases de datos fuente cargadas en el sistema.
- `dqmodel`: Se especializa en la definición y evaluación de modelos de CD. Sus *endpoints* bajo `/api/` se organizan en dos grupos:
    - Definición de Modelos:
        - `/dqmodels/`: CRUD de los modelos de CD.
        - `/dqmodels/<id>/full/`: Endpoint especial que recupera la estructura completa de un modelo, incluyendo todas sus dimensiones, factores, métricas y métodos asociados, así como los IDs de los componentes de contexto y problemas de CD vinculados.
        - `/dimensions-base/`, `/factors-base/`, `/metrics-base/`, `/methods-base/`: CRUD de los conceptos de CD reutilizables para construir modelos de CD.
    - Ejecución y Resultados: *endpoints* para ejecutar mediciones y recuperar resultados (por ejemplo, `/dqmodels/<id>/applied-dq-methods/<id>/execute/`).

Patrones y Características de Diseño

- Anidamiento: La *API* utiliza frecuentemente rutas anidadas para reflejar las relaciones entre entidades (e.g., `/dqmodels/<dqmodel_id>/dimensions/<dimension_id>/factors/...`).
- Acciones Personalizadas: Además de las operaciones CRUD estándar, los recursos exponen acciones personalizadas como `/generate-dqmethod-suggestion/` (que utiliza el motor de IA) o `/start-dq-model-execution/`.
- Desacople: El enrutamiento está modularizado; cada aplicación tiene su propio archivo `urls.py` que es luego incluido en el punto de entrada principal (`myproject/urls.py`).
- Documentación Automática: La *API* incluye documentación interactiva automática generada mediante CoreAPI, disponible en el *endpoints* `/docs/`. Esta documentación lista todos los *endpoints* disponibles, sus parámetros, métodos HTTP permitidos y estructuras de datos esperadas.

B.1.4. Integración con Servicios Externos

El sistema se integra con servicios externos para ampliar sus capacidades y habilitar funcionalidades avanzadas. La integración principal se realiza con una plataforma de inferencia de modelos de lenguaje (*LLMs*) para dar soporte a las funcionalidades de inteligencia artificial.

Motor de IA

El sistema utiliza la plataforma *Groq* [12] para la ejecución acelerada de los modelos de lenguaje. La integración se implementó mediante la librería `langchain-groq` de *Python*, que facilita la conexión con la lógica de las funcionalidades de IA.

La lógica de negocio de estas funcionalidades mediante IA se define en un directorio específico (`ai_modules`), que centraliza las operaciones de generación de sugerencias para métodos de CD y recomendaciones de dimensiones y factores. Los parámetros del cliente *Groq*, como los modelos de lenguaje a utilizar, se configuran directamente en esta lógica de implementación.

La configuración sensible, como la clave de *API* (`GROQ_API_KEY`), se gestiona mediante variables de entorno definidas en un archivo `.env` en el directorio `myproject/`, y se accede mediante la librería `python-decouple`, garantizando la separación de la configuración del código fuente.

B.1.5. Carga de Artefactos de CD Predefinidos

El sistema incorpora un mecanismo para la carga de definiciones de conceptos de CD, esenciales para inicializar la base de datos y enriquecer la experiencia de uso la herramienta. Para este proceso se brinda un comando de gestión de *Django* que procesa archivos de plantilla (*templates* definidos manualmente en formato *Markdown*.

Proceso de Carga

La carga se ejecuta desde la línea de comandos del servidor, utilizando el siguiente comando:

```
python manage.py load_dqtemplate --template <nombre_template>
```

donde `<nombre_template>` corresponde al nombre del archivo (sin extensión) ubicado en el directorio `dqmodel/templates/definitions/`.

Estructura del Archivo Template

Cada *template* debe seguir un formato estandarizado, respetando estructura jerárquica definida por los niveles de encabezado de *Markdown*, que el *parser* del sistema utiliza para identificar y relacionar automáticamente cada concepto con su entidad padre. A continuación se presenta un ejemplo de como se definen

```
## DQ Dimension: [Nombre]
**Semantic:** [Descripción semántica]

### DQ Factor: [Nombre]
**Semantic:** [Descripción semántica]
**Facet of (DQ Dimension):** [Dimensión padre]

#### DQ Metric: [Nombre]
**Purpose:** [Propósito]
**Granularity:** [Nivel granular]
**Result Domain:** [Dominio resultado]
**Measures (DQ Factor):** [Factor padre]

##### DQ Method: [Nombre]
**Name:** [Nombre método]
**Input data type:** [Tipo entrada]
**Output data type:** [Tipo salida]
**Algorithm:** [Código algoritmo]
**Implements (DQ Metric):** [Métrica padre]
```

El sistema controla la no duplicación de conceptos ya existentes y asegura la correcta asociación jerárquica al momento de la carga.

Template Inicial

El sistema requiere una carga mínima de artefactos base para un funcionamiento adecuado de la herramienta, los cuales consisten en las cinco dimensiones clásicas presentadas en el marco teórico (*Accuracy*, *Completeness*, *Consistency*, *Uniqueness*, *Freshness*) y sus factores asociados. Este conjunto inicial se encuentra definido en un archivo `preset_dq_dimensions_factors_base` definido en el marco de este trabajo, y se carga ejecutando el siguiente comando:

```
python manage.py load_dqtemplate --template preset_dq_dimensions_factors_base
```

Sin esta carga, la aplicación carece de los conceptos de CD necesarios para la para la generación de recomendaciones mediante el motor de IA en la construcción de modelos de CD.

Carga de Templates Adicionales

Además del conjunto de dimensiones y factores requeridos, es posible ampliar el repositorio de conceptos cargando plantillas personalizadas adicionales. Como un apoyo adicional para el usuario, se proporciona una plantilla extendida, definida mediante un archivo `preset_dq_concepts_base_template.md`, que incluye un conjunto de métricas comunes para los factores predefinidos, y algunos métodos genéricos que faciliten sus implementaciones. Estos artefactos predefinidos permiten la reutilización de definiciones comunes para construir nuevos modelos de CD desde cero.

Este enfoque permite que el sistema se inicialice con un repositorio de conceptos de CD estandarizados, definiendo una base de conocimiento con elementos reutilizables que agilizan el proceso de construcción de modelos de CD. Además, al emplear un formato simple y legible como *Markdown*, se abre la posibilidad de un uso colaborativo, donde los usuarios pueden crear y compartir sus plantillas para construir repositorios de conceptos comunes o enriquecidos, que sirvan como un punto de partida sólido para el uso de la herramienta.

B.2. Frontend

Esta sección del anexo documenta la arquitectura y los componentes principales del cliente web del sistema, desarrollado con el *framework Angular*.

B.2.1. Arquitectura y Estructura del Proyecto

El *frontend* de aplicación web adopta una arquitectura modular basada en los principios de *Angular*, organizando el código en carpetas por funcionalidad dentro del directorio src/app/ para promover la separación de responsabilidades, la reutilización de código y la mantenibilidad.

- `app.module.ts`: Módulo raíz que importa y declara las dependencias globales de la aplicación.
- `app-routing.module.ts`: Módulo dedicado a la configuración del enrutamiento principal y la navegación entre las vistas del sistema.
- `components/`: Directorio para componentes de interfaz de usuario genéricos, reutilizables (modales, notificaciones, etc.).
- `pages/`: Contenedor de los componentes que representan las vistas principales o páginas completas del sistema, asociadas a rutas específicas.
- `shared/`: Componentes complejos reutilizables con lógica compartida.
- `services/`: Capa de servicios inyectables que encapsulan la lógica de negocio, el estado de la aplicación y la comunicación con la *API backend*.

B.2.2. Capa de Presentación: Componentes y Vistas

La interfaz de usuario se construye mediante una jerarquía de componentes `smart` (`pages`) y `dumb` (`presentationals`).

Vistas de la Aplicación (`pages/`)

En el directorio `pages/`, cada componente se corresponde con una ruta específica y una vista particular de la aplicación. Estas se encuentras organizadas en subdirectorios según las diferentes etapas de la Fase 2 de CaDQM, brindando separación clara de responsabilidades mediante una estructura modular.

- `home/`: Página de inicio y *landing page* de la aplicación.
- `dashboard/`: Panel de control del proyecto actual.
- `stage4-dq-model-definition/`: Contiene las páginas para la definición del modelo de CD:

    - `dqproblems-priorization/`
    - `dqproblems-selection/`
    - `dq-dimensions-factors-selection/`
    - `dq-methods-definition/`
    - `dq-metrics-definition/`
    - `dqmodel-confirmation/`
- `stage5-dq-measurement/`: Contiene las páginas para la ejecución de mediciones de CD sobre los datos fuente:
    - `dq-measurement-execution/`
    - `dq-measurement-results/`
- `stage6-dq-assessment/`: Contiene los páginas para la definición de umbrales y evaluación de los resultados obtenidos de las mediciones CD:
    - `dq-assessment-approaches-definition/`
    - `dq-assessment-execution/`

Componentes de Interfaz de Usuario (`/components`)

Componentes de interfaz reutilizables y genéricos, centrados en la presentación y con mínima lógica interna. Incluye, por ejemplo, modales para confirmaciones y visualización de contenido auxiliar o genéricos, y notificaciones *toast*.

Componentes Compartidos (`/shared`)

Componentes que encapsulan lógica de negocio relevante, y son reutilizados en múltiples vistas. Incluye, por ejemplo, un asistente de pasos para procesos complejos (`step-navigator`) y modales interactivos para exploración de datos como componentes de contexto y problemas de CD, así como de los datos del proyecto.

B.2.3. Capa de Lógica de Negocio y Servicios

Los servicios (`services/`) en *Angular* implementan la capa de lógica de negocio y comunicación con el *backend*. Su propósito principal es desacoplar la interfaz de usuario de la persistencia de datos, centralizando tanto las operaciones de acceso como el estado de la aplicación.

- `dq-model.service.ts`: Gestiona la definición de los modelos de CD, la medición y evaluación de los datos. Incluye manejo de funcionalidades de generación mediante IA.
- `project.service.ts`: Gestión general de proyectos (CRUD).
- `project-data.service.ts`: Gestiona el estado global de la aplicación para el proyecto en uso, centralizando el acceso a sus datos.
- `notification.service.ts`: Provee un sistema unificado de notificaciones para toda la aplicación.

A partir de los requerimientos y el análisis realizados sobre el proyecto a ejecutar, se consideraron varias tecnologías para las distintas áreas de desarrollo.

Anexo C

Experimentación IA

Este Anexo presenta algunos de los diferentes resultados obtenidos durante el proceso de experimentación y validación de las funcionalidades de IA integradas en la herramienta desarrollada.

C.1. Comparativa Técnica de Modelos de Lenguaje

Para establecer el punto de partida en el proceso de selección del modelo de lenguaje, en la Tabla C.1 se presenta una comparación de las principales características técnicas de las cuatro variantes de modelos *Llama 3* disponibles en *GroqCloud* como modelos de producción [13], consideradas para la experimentación.

Característica	**Llama-3-70B-8192**	**Llama-3-8B-8192**	**Llama-3.1-8b-instant**	**Llama-3-70B-Versatile**
Parámetros	70B	8B	8B	70B
Ventana contexto (*tokens*)	8192	8192	128K	128K
Costo/1M *tokens* Input (USD)	$0.59	$0.05	$0.05	$0.59
Costo/1M *tokens* Output (USD)	$0.79	$0.08	$0.08	$0.79
Velocidad (*tokens*/s)	∼100	∼200	∼250	∼90
Caso de usos recomendados	Tareas complejas, generación de alta calidad	Balance velocidad-calidad	Respuestas rápidas, tareas simples	Versatilidad en múltiples dominios

Tabla C.1: Comparativa técnica de modelos *Llama 3* en *GroqCloud* [13]

C.2. Resultados Experimentación Recomendaciones de Dimensiones/Factores de CD

Esta sección presenta los resultados de la experimentación centrada en analizar el orden de los factores de CD recomendados usando los tres *LLMs* viables. Para ello, se ejecutó un ciclo de cinco repeticiones del *script* de prueba con cada *LLM*, registrando la posición de las 13 recomendaciones en cada ejecución.

C.2.1. Análisis Orden Factores de CD Recomendados

Ejecuciones *Llama-3.1-8b-instant*:

Como se observa en la Tabla C.2.1, los resultados obtenidos para el modelo *Llama-3.1-8b-instant* permiten observar una clara tendencia hacia la temprana recomendación de ciertos factores de CD a partir de los componentes de contexto y problemas de CD disponibles. En particular, el factor *Inter-relationship Integrity* se destacó como el más relevante, ocupando el primer lugar en cuatro de las cinco ejecuciones y el segundo lugar en la restante. Junto con *Domain Integrity*, otro factor de la dimensión *Consistency*, completaron el top 2 para todas las ejecuciones realizadas. Por su parte, el factor *Density* (*Completeness*) también fue otro de temprana recomendación de manera consistente, siendo el tercero en ser sugerido en cuatro de las cinco ejecuciones.

Dimensión	**Factor**	**Ej.1**	**Ej.2**	**Ej.3**	**Ej.4**	**Ej.5**	**AVG**	**Ranking Gral.**	**Desv. Est.**
Accuracy	*Semantic Accuracy*	7°	6°	7°	6°	7°	6.6	6°	0.55
	Syntactic Accuracy	11°	12°	11°	12°	11°	11.4	11°	0.55
	Precision	10°	10°	8°	8°	8°	8.8	10°	1.10
Completeness	*Density*	3°	5°	3°	3°	3°	3.4	3°	0.89
	Coverage	6°	4°	6°	5°	6°	5.4	5°	0.89
Freshness	*Currency*	8°	7°	9°	9°	9°	8.4	8°	0.89
	Timeliness	12°	11°	12°	13°	12°	12.0	12°	0.71
	Volatility	13°	13°	13°	10°	13°	12.4	13°	1.20
Consistency	*Domain Integrity*	**1°**	2°	2°	2°	2°	1.8	2°	0.45
	Intra-rel. Integrity	5°	8°	5°	11°	5°	6.8	7°	2.48
	Inter-rel. Integrity	2°	**1°**	**1°**	**1°**	**1°**	1.2	**1°**	0.45
Uniqueness	*No-duplication*	4°	3°	4°	7°	4°	4.4	4°	1.20
	No-contradiction	9°	9°	10°	4°	10°	8.4	9°	2.30

Tabla C.2: Ranking de factores de CD recomendados usando *Llama-3.1-8b-instant*

En general, con este *LLM* las recomendaciones mostraron un comportamiento relativamente estable a lo largo de las diferentes ejecuciones, devolviendo los mismos factores de CD en posiciones similares. Sin embargo, también se observaron casos de inestabilidad en la recomendación de ciertos factores. En particular, *Intra-relationship Integrity* y *No-contradiction* presentaron desviaciones estándar elevadas (2.48 y 2.30 respectivamente). Esta inestabilidad se refleja en la variabilidad de su orden de aparición a lo largo de las distintas ejecuciones.

Ejecuciones *Llama-3.3-70b-versatile*:

Como se observa en la Tabla C.2.1, el modelo *Llama-3.3-70b-versatile* mostró una alta estabilidad en la mayoría de sus recomendaciones, evidenciada por múltiples factores de CD con una desviación estándar cero. Sumado a ese comportamiento, se destaca la temprana recomendación de los factores de la dimensión *Completeness*: el factor *Density* apareció consistentemente en el primer puesto en cuatro de cinco ejecuciones, y *Coverage* que promedió un tercer puesto en el orden de las recomendaciones generadas.

Dimensión	**Factor**	**Ej.1**	**Ej.2**	**Ej.3**	**Ej.4**	**Ej.5**	**AVG**	**Ranking Gral.**	**Desv. Est.**
Accuracy	*Semantic Accuracy*	5°	5°	3°	5°	4°	4.4	5°	0.75
	Syntactic Accuracy	10°	10°	10°	10°	10°	10.0	10°	0.00
	Precision	6°	6°	5°	6°	6°	5.8	6°	0.40
Completeness	*Density*	1°	1°	2°	1°	1°	1.2	1°	0.40
	Coverage	3°	4°	4°	4°	3°	3.6	3°	0.49
Freshness	*Currency*	13°	13°	13°	13°	13°	13.0	13°	0.00
	Timeliness	11°	11°	11°	11°	11°	11.0	11°	0.00
	Volatility	12°	12°	12°	12°	12°	12.0	12°	0.00
Consistency	*Domain Integrity*	4°	7°	7°	7°	5°	6.0	7°	1.22
	Intra-rel. Integrity	2°	2°	6°	2°	8°	4.0	4°	2.28
	Inter-rel. Integrity	8°	3°	1°	3°	2°	3.4	2°	2.06
Uniqueness	*No-duplication*	7°	8°	8°	8°	7°	7.6	8°	0.49
	No-contradiction	9°	9°	9°	9°	9°	9.0	9°	0.00

Tabla C.3: Ranking de factores de CD recomendados usando *Llama-3.3-70b-versatile*

Por el contrario, los factores de CD que mayor variación mostraron a lo largo de todas las ejecuciones fueron los de la dimensión *Consistency*. Se observaron resultados altamente inestables en *Inter-relationship Integrity* y *Intra-relationship Integrity*, llegando a recomendar estos factores desde el primer puesto hasta el octavo.

Finalmente, se observa los factores de CD correspondientes a la dimensión *Freshness*: *Currency*, *Timeliness* y *Volatility*, fueron sistemáticamente recomendados en los últimos puestos, con nula variación entre ejecuciones.

Ejecuciones *Llama3-70b-8192*:

En la Tabla C.2.1 se presentan los resultados obtenidos para el modelo *Llama3-70b-8192*. En este caso, se destaca *No-duplication* (*Uniqueness*) como el principal factor de CD en términos de temprana sugerencia, siendo recomendado primero en cuatro de cinco ejecuciones.

Completando los primeros puestos del ranking general de factores de CD, aparecen *Semantic Accuracy* (*Accuracy*), *Inter-relationship Integrity* (*Consistency*) y *Density* (*Completeness*), los cuales alternaron consistentemente en las posiciones iniciales.

En contraste, aunque reforzando la noción de estabilidad de los resultados obtenidos, la dimensión *Freshness* aparece claramente relegada, ya que todos sus factores (*Currency*, *Timeliness* y *Volatility*) ocuparon sistemáticamente las últimas posiciones en cada ejecución, con nula variación.

Dimensión	**Factor**	**Ej.1**	**Ej.2**	**Ej.3**	**Ej.4**	**Ej.5**	**AVG**	**Ranking Gral.**	**Desv. Est.**
Accuracy	*Semantic Accuracy*	2°	2°	3°	3°	2°	2.4	2°	0.49
	Syntactic Accuracy	8°	6°	10°	8°	8°	8.0	8°	1.26
	Precision	7°	9°	7°	6°	6°	7.0	7°	1.10
Completeness	*Density*	3°	3°	4°	4°	3°	3.4	4°	0.49
	Coverage	5°	7°	6°	7°	7°	6.4	6°	0.80
Freshness	*Currency*	11°	11°	11°	11°	11°	11.0	11°	0.00
	Timeliness	12°	12°	12°	12°	12°	12.0	12°	0.00
	Volatility	13°	13°	13°	13°	13°	13.0	13°	0.00
Consistency	*Domain Integrity*	9°	8°	8°	9°	9°	8.6	9°	0.49
	Intra-rel. Integrity	4°	4°	2°	2°	1°	2.6	3°	1.36
	Inter-rel. Integrity	10°	10°	9°	10°	10°	9.8	10°	0.40
Uniqueness	*No-duplication*	1°	1°	1°	1°	4°	1.6	1°	1.20
	No-contradiction	6°	5°	5°	5°	5°	5.2	5°	0.40

Tabla C.4: Ranking de factores de CD recomendados usando *Llama3-70b-8192*

C.2.2. Validación de las Justificaciones de las Recomendaciones

Métrica de CD Recomendada: *Density (Completeness)*

El factor *Density* se ubicó consistentemente entre los primeros puestos en las recomendaciones de los tres modelos, aunque cada uno justificó su elección de manera distinta. Para analizar estas diferencias, se presentan en la Tabla C.5 los problemas de CD y componentes de contexto que cada modelo de lenguaje utilizó para sugerir dicho factor de CD.

Density (Completeness)	**3-70b-8192**	**3.3-70b-versatile**	**3.1-8b-instant**
Problemas de CD			
PC1: Valores nulos en `reviewText` o `title`	✓	✓	✓
PC7: Reseñas duplicadas			✓
PC13: Entradas redundantes			✓
PC14: Relaciones faltantes			✓
Componentes de Contexto			
AD1: Comercio electrónico de libros y reseñas en Amazon		✓	✓
BR2: `(title, userId)` debe ser único		✓	✓
DF1: Excluir reseñas con menos de 20 caracteres		✓	✓
DF2: Incluir solo reseñas en inglés desde 2018		✓	✓
DQR2: Completitud $\geq$95 % en campos obligatorios	✓	✓	✓
T1: Calcular *sentiment scores*	✓	✓	✓
T2: Detectar reseñas sospechosas		✓	✓
T3: Generar ranking de calidad textual		✓	
UT1: Analistas de marketing		✓	✓
UT2: Científicos de datos		✓	✓
UT3: Investigadores académicos		✓	

Tabla C.5: Problemas de CD y componentes de contexto utilizados por modelo para generar la recomendación del factor *Density*.

Análisis Detallado: Modelo *3.3-70b-versatile*

- **Uso de Problemas de CD**: El modelo asocia correctamente el factor *Density* con el problema de valores nulos en `reviewText` y `title` (PC1), reconociendo que estos nulos afectan la cantidad de datos completos disponibles para el análisis.
- **Uso de Componentes de Contexto**: La recomendación surge de una amplia cobertura de los componentes disponibles. Entre ellos, se apoyó adecuadamente en el requerimiento de CD (DQR2), que establece un umbral de completitud mínima del $\geq$95 % en los campos obligatorios, lo que permite asociar directamente este componente con el factor *Density*. Si bien componentes como el dominio de aplicación (AD1) y las tareas de cálculo de sentiment scores o generación de rankings (T1, T2, T3) no explican la elección del factor Density por sí solos, la recomendación se sustenta en que estas tareas dependen de datos completos para asegurar el análisis y la toma de decisiones. Por otro lado, la recomendación también incluye componentes de filtrado de datos (DF1, DF2) que no muestran una relación explícita con el factor o la dimensión recomendada.
- **Justificación**:

> "La dimensión de Completitud, específicamente el factor de Densidad, es crucial en este contexto, ya que impacta directamente la disponibilidad de los datos necesarios para el análisis y la toma de decisiones. Los valores nulos en campos requeridos y en campos clave indican brechas en los datos, afectando su densidad. El dominio de aplicación de e-commerce y reseñas de clientes depende en gran medida de datos completos para asegurar análisis y recomendaciones precisas. Las reglas de negocio, como evitar múltiples reseñas por usuario y por libro, y los filtros de datos, como incluir solo reseñas en inglés publicadas desde 2018, enfatizan aún más la necesidad de datos densos y completos. El requisito de una métrica de completitud de al menos el 95 % para los campos obligatorios también respalda la importancia de la densidad. Por lo tanto, asegurar una alta densidad de datos es esencial para la integridad y fiabilidad de los procesos basados en datos en este contexto de e-commerce y análisis de reseñas."

El texto explicativo integra todos los problemas de CD y componentes de contexto utilizados, siendo claro al relacionar el factor Density con el problema de CD considerado (PC1) y fundamenta adecuadamente la métrica de completitud mínima (DQR2) asociada al factor. La explicación sobre componentes como los filtros de datos (DF1, DF2) y algunas tareas presenta argumentos menos directos y no necesariamente relevantes para el factor.

Análisis Detallado: Modelo *3-70b-8192*

- **Uso de Problemas de CD**: El modelo asocia correctamente el factor *Density* con el problema de valores nulos en `reviewText` y `title` (PC1).
- **Uso de Componentes de Contexto**: Se apoya en una cobertura contextual acotada, utilizando el requerimiento de CD (DQR2), que define un umbral de completitud del 95 %, lo cual constituye una asociación directa y adecuada con el factor *Density*. También incluye la tarea de análisis de sentimiento (T1), sugiriendo una posible asociación con la necesidad de datos completos para realizar dicho análisis.
- **Justificación**:

  > "El factor de Densidad se considera relevante ya que los valores faltantes en los campos de texto principales afectan la completitud del dataset, reduciendo su utilidad para el análisis de sentimiento y la generación de métricas de CD. Mantener un nivel de completitud de al menos 95 % en los campos requeridos asegura que los análisis posteriores sean representativos y consistentes."

El texto explicativo generado conecta de forma clara la presencia de valores nulos (PC1) con la necesidad de mantener una alta completitud (DQR2). Además, incorpora el componente de la tarea (T1) para reforzar la relevancia del factor en el contexto de análisis de sentimiento, mostrando coherencia entre los componentes utilizados y el razonamiento presentado.

Análisis Detallado: Modelo *3.1-8b-instant*

- **Uso de Problemas de CD**: Si vincula adecuamente el factor *Density* con el problema de valores nulos (PC1), tambien se apoya en los problemas relacionados con reseñas duplicadas (PC7), entradas redundantes (PC13) y relaciones faltantes (PC14). El uso de estos otros problemas de CD no se interpreta como el más adecuado, ya que los problemas seleccionados no reflejan una asociación bien fundamentada con la definición del factor *Density*. En particular, las reseñas duplicadas (PC7) y las entradas redundantes (PC13) remiten más a aspectos de unicidad o redundancia que a una ausencia de datos propiamente dicha. Si bien el problema de relaciones faltantes (PC14) podría vincularse a la dimensión *Completeness*, en tanto puede implicar ausencia de ciertos datos necesarios, su relación específica con el factor *Density* no es del todo clara.
- **Uso de Componentes de Contexto**: El modelo emplea prácticamente el mismo conjunto de componentes de contexto que el modelo *versatile*, destacándose nuevamente el requerimiento de CD (DQR2), que establece un umbral de completitud mínima del $\geq$95 % en los campos obligatorios y permite vincular directamente la recomendación con el factor *Density*. Asimismo, incluye componentes como los filtros de datos (DF1, DF2) y el dominio de aplicación (AD1), que carecen de una relación explícita que justifique la elección del factor.
- **Justificación**:

  > "La completitud es una dimensión clave en este contexto debido a la presencia de valores nulos en campos obligatorios, registros duplicados, entradas redundantes y marcas de tiempo faltantes. El factor Density es particularmente relevante, ya que mide la proporción de entradas de datos reales en comparación con el total de posibles entradas, lo cual se ve afectado por los problemas mencionados anteriormente. Los componentes de contexto de apoyo, como las reglas de negocio y los requerimientos de datos, refuerzan aún más la importancia de la completitud y la densidad para garantizar la integridad de los datos."

La justificación establece una conexión parcialmente válida entre el factor *Density* y la presencia de valores nulos, aspecto que efectivamente incide en la proporción de datos existentes respecto al total esperado. Sin embargo, el texto amplía indebidamente el alcance del factor al incorporar problemas como la duplicación de registros o la integridad general de los datos, que no guardan

una relación directa ni fundamentada con la densidad. Además, la referencia a los componentes de contexto resulta ambigua y genérica, sin ofrecer una vinculación explícita entre los elementos empleados y el razonamiento presentado.

Métrica de CD Recomendada: *Currency (Freshness)*

El factor *Currency*, correspondiente a la dimensión *Freshness*, fue consistentemente sugerido en las últimas posiciones del ranking general de recomendaciones generadas por los distintos modelos. Este comportamiento motivó el análisis de sus respectivas salidas para evaluar si la inclusión del factor respondió a una justificación sólida o si, por el contrario, se trató de una recomendación residual, posiblemente derivada de un proceso de descarte. La Tabla C.6 resume los problemas de CD y los componentes de contexto que cada modelo utilizó para fundamentar esta recomendación.

Currency (Freshness)	**3-70b-8192**	**3.3-70b-versatile**	**3.1-8b-instant**
Problemas de CD			
PC15: Timestamps faltantes	✓	✓	✓
PC16: Valores fuera de límites establecidos			✓
PC17: Valores inconsistentes entre fuentes	✓		✓
Componentes de Contexto			
AD1: Comercio electrónico de libros impresos y digitales en Amazon		✓	✓
BR1: Validación de rango 1-5 para score		✓	✓
DF2: Incluir solo reseñas en inglés desde 2018		✓	✓
DQR2: Completitud mínima 99 % en campos obligatorios	✓	✓	✓
T1: Cálculo de *sentiment scores*	✓	✓	✓
T2: Detección y filtrado de reseñas sospechosas		✓	✓
UT1: Analistas de marketing		✓	✓
UT2: Científicos de datos		✓	✓
UT3: Investigadores académicos		✓	✓

Tabla C.6: Problemas de CD y componentes de contexto utilizados por modelo para generar la recomendación del factor *Currency*.

Análisis Detallado: Modelo *3.3-70b-versatile*

- **Uso de Problemas de CD**: El modelo asocia el factor *Currency* con el problema de marcas de tiempo faltantes (PC15), vinculando la ausencia de *timestamps* con la capacidad de determinar la actualidad de los datos.
- **Uso de Componentes de Contexto**: La recomendación utiliza múltiples componentes de contexto. Entre ellos, el componente de filtrado de datos que restringe las reseñas a partir de 2018 (DF2), aporta una conexión explícita con la necesidad de datos actualizados, en línea con el factor *Currency*. También se incluyen el dominio de aplicación (AD1) y los diferentes tipos de usuario (UT1, UT2, UT2), que podrían implicar una necesidad indirecta de vigencia temporal. Sin embargo, se considera que las tareas T1 y T2, así como el requerimiento DQR2, no presentan una relación clara con el concepto de CD, y su inclusión no contribuye directamente a justificar la recomendación.
- **Justificación**:

  > "La dimensión y el factor recomendados están bien respaldados por la presencia de marcas de tiempo faltantes, lo que se relaciona directamente con la actualidad de los datos. El dominio de aplicación de e-commerce y reseñas de clientes en Amazon, donde la frescura de los datos es crucial para el análisis en tiempo real y la toma de decisiones, apoya aún más esta recomendación. Además, el filtro de datos que incluye solo reseñas en inglés publicadas desde 2018 y el requisito de baja latencia y altas métricas de completitud también enfatizan la importancia de tener datos actualizados."

  La justificación establece una conexión válida entre el factor *Currency* y la falta de marcas de tiempo (PC15), destacando su impacto en la vigencia de los datos. Luego, si bien intenta incorporar los demás componentes utilizados al texto explicativo, la argumentación de los mismos introduce

ruido, intentando justificar la inclusión de estos en escenarios de necesidad de actualidad de los datos, en lugar de suponer una relación directa.

Análisis Detallado: Modelo *3-70b-8192*

- **Uso de Problemas de CD**: El modelo asocia el factor *Currency* con problema de *timestamps* faltantes (PC15), estableciendo una relación entre la ausencia de marcas temporales y la imposibilidad de evaluar la actualidad de los datos. Por otro lado, también utiliza el problema de valores inconsistentes entre fuentes (PC17) el cual carece relación directa con el factor recomendado.
- **Uso de Componentes de Contexto**: Se apoya en dos componentes de contexto que no presentan relación directa con el factor *Currency*, mientras que omite el componente de filtrado de datos de reseñas desde 2018 (DF2), que sí se vincula directamente con la frescura de los datos.
- **Justificación**:

  > "Los problemas de CD de marcas de tiempo faltantes y valores inconsistentes entre fuentes indican que los datos no están actualizados, lo cual es un aspecto clave del factor Currency. El requisito de garantizar la integridad de los datos mediante una métrica de completitud de al menos el 95 % para los campos obligatorios también apoya este factor."

  El texto explicativo reconoce la relevancia de mantener los datos actualizados, pero su fundamentación es poco precisa. Aunque identifica problemas potencialmente relacionados con la vigencia (como los *timestamps* faltantes), los interpreta de manera incorrecta al asumir que su ausencia implica desactualización. La incorporación de valores inconsistentes y de un requerimiento de completitud refuerza la confusión conceptual, al mezclar dimensiones de CD distintas.

Análisis Detallado: Modelo *3.1-8b-instant*

- **Uso de Problemas de CD**: El modelo utiliza el problema de marcas de tiempo faltantes (PC15), cuya inclusión se considera adecuada, ya que la carencia de esta información impediría verificar la actualidad de los datos. Por otro lado, emplea los problemas de valores fuera de límites establecidos (PC16) y valores inconsistentes entre fuentes (PC17), cuya justificación no se vincula claramente con el factor *Currency*.
- **Uso de Componentes de Contexto**: El modelo se apoya en los mismos componentes de contexto que el modelo *versatile*, incluyendo el dominio de aplicación (AD1), los filtros de datos (DF2) y los requerimientos de completitud (DQR2). En particular, el componente DF2 podría vincularse con la necesidad de datos actualizados, al restringir las reseñas a partir de 2018. Sin embargo, los demás componentes mencionados no guardan una relación directa con el factor *Currency* ni con la dimensión *Freshness*, mostrando así las mismas debilidades observadas en el modelo *versatile*.
- **Justificación**:

  > "La dimensión seleccionada 'Freshness' está respaldada por problemas relacionados con la obsolescencia de los datos y las inconsistencias, los cuales son abordados por el factor 'Currency'. Este factor también cuenta con el apoyo del requisito del sistema de ofrecer respuestas en tiempo real y de la necesidad de garantizar la integridad de los datos. Los componentes de contexto, como el dominio de la aplicación, las reglas de negocio y los filtros de datos, refuerzan además la importancia de mantener los datos actualizados."

  El texto explicativo intenta justificar la recomendación aludiendo a la obsolescencia e inconsistencias de los datos, pero presenta una comprensión imprecisa del factor *Currency*. Aunque reconoce la importancia de mantener los datos actualizados, mezcla conceptos de integridad y capacidad de respuesta en tiempo real que pertenecen a otras dimensiones o requisitos del sistema. Además, la conexión entre los problemas utilizados y los componentes del contexto se presenta de forma ambigua, sin demostrar una relación clara o fundamentada.

C.3. Resultados Experimentación Generación de Métodos de CD

Esta sección presenta los resultados obtenidos de la experimentación de la funcionalidad de generación automática de métodos de CD para las métricas de CD: *Valid Score Range* y *Secure URLs Ratio*.

C.3.1. Métodos de CD Generados por Métrica

Métrica de CD: *Valid Score Range*

Propósito: Verificar que cada puntuación de reseña esté dentro del rango esperado (1-5).
Granularidad: Fila (por registro individual).
Dominio de resultado: Booleano {0,1}

Salidas Generadas

De las 10 ejecuciones realizadas, se obtuvo 3 variantes de métodos de CD que se agruparon según el código SQL generado:

- **Variante Principal (5/10):** En el Bloque C.1 se presenta el método de CD generado para la variante principal.

```
{
    "name": "validateScorePerRow",
    "inputDataType": "column",
    "outputDataType": "{0,1}",
    "algorithm": "SELECT CASE WHEN column1 BETWEEN 1 AND 5 THEN 1 ELSE 0 END AS
    is_valid FROM table1",
    "implements": 2
}
```

Bloque C.1: Método de CD generado – Variante principal (70 %)

- **Variante Secundaria (3/10):** En el Bloque C.2 se muestra el método de CD generado para esta variante.

```
{
    "name": "validateScoreWithFullRow",
    "inputDataType": "table",
    "outputDataType": "table",
    "algorithm": "SELECT *, CASE WHEN column1 BETWEEN 1 AND 5 THEN 1 ELSE 0 END AS
    is_valid FROM table1"
}
```

Bloque C.2: Método de CD generado – Variante secundaria (20 %)

- **Variante Alternativa (2/10):** En el Bloque C.3 se presenta el método de CD generado para esta última variante.

```
{
    "name": "validateScoreCompact",
    "inputDataType": "column",
    "outputDataType": "{0,1}",
    "algorithm": "SELECT column1, CASE WHEN column1 BETWEEN 1 AND 5 THEN 1 ELSE 0
    END AS is_valid FROM table1"
}
```

Bloque C.3: Método de CD generado – Variante alternativa (10 %)

Validación Práctica:
Adaptación para esquema real (*dataset Amazon Reviews*):

- `table1` → `books_rating`
- `column1` → `reviewscore`

A partir de esta sustitución de parámetros por elementos específicos del *dataset* a medir, se ejecutaron las variantes de consultas generadas para la métrica de CD, resumiendo los resultados obtenidos en la Tabla C.3.1.

Consulta SQL (Método Aplicado)	Tipo Resultado	Tiempo	Count	Correctitud

Variante Principal
`SELECT`
`  CASE WHEN review_score`
`  BETWEEN 1 AND 5`
`  THEN 1 ELSE 0 END`
`FROM books_rating`
	Valor binario $\{0, 1\}$	00:04:26.205	3mill	✓

Variante Secundaria
`SELECT *,`
`  CASE WHEN review_score`
`  BETWEEN 1 AND 5`
`  THEN 1 ELSE 0 END`
`FROM books_rating`
	Tabla completa (incluyendo Valor binario $\{0, 1\}$)	00:08:30.755	3mill	✓

Variante Alternativa
`SELECT review_score,`
`  CASE WHEN review_score`
`  BETWEEN 1 AND 5`
`  THEN 1 ELSE 0 END`
`FROM books_rating`
	Columna `reviewscore` + Valor binario $\{0, 1\}$	00:00:42.777	3mill	✓

Tabla C.7: Resultados de ejecución de los métodos de CD generados para la métrica *Valid Score Range*

Análisis de las Variantes *SQL* Generadas:

Todas las variantes implementan correctamente la lógica del método de CD, de acuerdo con el propósito definido por la métrica de CD, aunque presentan diferencias en el diseño de las consultas y, como consecuencia, en los resultados obtenidos. Dado que la granularidad de la métrica es a nivel de *fila*, cada variante genera múltiples resultados. Para garantizar una comparación adecuada, se verificó que todas las consultas evaluaran la misma cantidad de filas.

- Variante Principal: devuelve únicamente el resultado binario, cumpliendo con el `resultDomain` ($\{0, 1\}$) esperado. Sin embargo, presenta un rendimiento inesperadamente bajo (4m 26s) pese a la simplicidad de la consulta.
- Variante Secundaria: la consulta proyecta todas las columnas de la tabla innecesariamente, a convierte en la más lenta (8m 30s) debido a la transferencia excesiva de datos.
- Variante Alternativa: incluye solo la columna evaluada y el resultado binario. Fue la más eficiente, con un tiempo de ejecución de apenas 42.7s.

Conclusiones:

- Todos los algoritmos generados implementan correctamente la métrica de CD, siendo fácilmente adaptables para su ejecución mediante la sustitución de parámetros por los elementos específicos del *dataset* a evaluar.

- Si bien todas las variantes generadas para las consultas *SQL* producen resultados válidos para el propósito de la medición, difieren en su diseño y eficiencia. La Variante Principal genera un resultado más fiel al dominio esperado, devolviendo únicamente la columna binaria correspondiente, mientras que la Variante Alternativa, aunque incluye columnas adicionales, retorna el valor de CD esperado con un rendimiento significativamente superior.

Métrica de CD: *Secure URLs Ratio*

Propósito: Medir el porcentaje de URLs seguras (HTTPS) en una columna.
Granularidad: Columna.
Dominio de resultado: Porcentaje entre [0,1].

Salidas Generadas

De las 10 ejecuciones realizadas, se obtuvo 3 variantes de métodos de CD que se agruparon según el código SQL generado:

- **Variante 1 (4/10):** En el Bloque C.4 se presenta el método de CD generado para esta variante.

```
{
    "name": "calculateSecureURLsRatio",
    "inputDataType": "table1",
    "outputDataType": "float",
    "algorithm": "SELECT CAST(SUM(CASE WHEN column1 LIKE '%https%' THEN 1 ELSE 0
    END) AS FLOAT) / COUNT(*) FROM table1"
}
```

Bloque C.4: Variante de patrón amplio

- **Variante 2 (4/10):** En el Bloque C.5 se muestra el método de CD generado para esta variante.

```
{
    "name": "calculateSecureURLsRatio",
    "inputDataType": "table1",
    "outputDataType": "float",
    "algorithm": "SELECT CAST(SUM(CASE WHEN column1 LIKE '%https://%' THEN 1 ELSE 0
     END) AS FLOAT) / COUNT(*) FROM table1"
}
```

Bloque C.5: Variante de patrón específico

- **Variante 3 (2/10):** En el Bloque C.6 se presenta el método de CD generado para esta última variante.

```
{
    "name": "calculateSecureURLsRatio",
    "inputDataType": "table1",
    "outputDataType": "float",
    "algorithm": "SELECT CAST(COUNT(CASE WHEN column1 LIKE '%https://%' THEN 1 END)
     AS FLOAT) / COUNT(*) FROM table1"
}
```

Bloque C.6: Variante con COUNT

Validación Práctica

Adaptación para esquema real (*dataset Amazon Reviews*):

- `table1` → `books_data`
- `column1` → `infolink`

A partir de esta sustitución de parámetros por elementos específicos del *dataset* a medir, se ejecutaron las variantes de consultas generadas para la métrica de CD, resumiendo los resultados obtenidos en la Tabla C.3.1.

Consulta SQL (Método Aplicado)	Resultado	Tiempo	Correctitud
Variante 1: `SELECT CAST(SUM(CASE WHEN infolink LIKE '%https%' THEN 1 ELSE 0 END) AS FLOAT)/COUNT(*) FROM books_data`	0.2027	00:00:11.428	✓
Variante 2: `SELECT CAST(SUM(CASE WHEN infolink LIKE '%https://%' THEN 1 ELSE 0 END) AS FLOAT)/COUNT(*) FROM books_data`	0.2027	00:00:01.288	✓
Variante 3: `SELECT CAST(COUNT(CASE WHEN infolink LIKE '%https://%' THEN 1 END) AS FLOAT)/COUNT(*) FROM books_data`	0.2027	00:00:01.111	✓

Tabla C.8: Resultados de ejecución de los métodos de CD generados para la métrica *SecureURLsRatio*

Análisis de las Variantes *SQL* Generadas:

Todas las variantes para las consultas *SQL* generadas implementan correctamente la lógica de la métrica de CD, produciendo resultados equivalentes. Sin embargo, difieren en el patrón de búsqueda y en la forma de conteo utilizada, lo cual incide en su precisión y rendimiento.

- Variante 1 (Patrón de búsqueda amplio): busca cualquier ocurrencia de la cadena `https`, pudiendo incluir falsos positivos (por ejemplo, si la cadena aparece en texto no correspondiente a una URL). Presentó el mayor tiempo de ejecución (11.4s).
- Variante 2 (Patrón de búsqueda específico): restringe la búsqueda a `https://`, detectando únicamente URLs seguras reales. Fue mucho más eficiente (1.28s) sin perder precisión.
- Variante 3 (Patrón de búsqueda específico + conteo alternativo): reemplaza `SUM(CASE WHEN ...)` por `COUNT(CASE WHEN ...)`, logrando el mejor tiempo (1.11s) con resultados equivalentes.

Conclusiones:

- Los tres algoritmos propuestos implementan correctamente la métrica de CD, obteniendo resultados consistentes (ratio HTTPS de 0.2027).
- Si bien la Variante 1 presenta un patrón de búsqueda más amplio (`%https%`) que podría generar falsos positivos, su diseño constituye un excelente punto de partida que solo requeriría un ajuste mínimo (cambiar a `%https://%`) para perfeccionar su precisión.
- Según los resultados obtenidos, tanto la Variante 2 (patrón `%https://%` con `SUM`) como la Variante 3 (mismo patrón con `COUNT`) pueden considerarse soluciones óptimas, al presentar rendimientos comparables y ofrecen implementaciones igualmente válidas, donde la elección entre `SUM` y `COUNT` responde más a preferencias de estilo de código que a diferencias funcionales.

Anexo D

Caso de Estudio 1: Funcionalidad

En esta sección se detallan los componentes de contexto y problemas de CD definidos para el Caso de Estudio 1 donde se analizan las funcionalidades de la herramienta. Todos los objetos han sido etiquetados con un identificador único para ser referenciados de manera clara. En la tabla D.1 se encuentra la información correspondiente.

Categoría	**Componente de contexto**
Application domain	AD1: Comercio electrónico de libros impresos y digitales, junto con reseñas de clientes en Amazon[1].
User types	UT1: Analistas de marketing que monitorean la reputación de títulos y editoriales.
	UT2: Científicos de datos dedicados a entrenar modelos de recomendación.
	UT3: Investigadores académicos que examinan posibles sesgos lingüísticos.
Tasks at hand	T1: Cálculo de *sentiment scores* por libro.
	T2: Detección y filtrado de reseñas sospechosas (*review spam*).
	T3: Generación de un *ranking* de calidad de contenidos textuales.
Data filtering needs	DF1: Se excluyen reseñas con menos de 20 caracteres.
DQ requirements	DQR1: Latencia inferior a 5 segundos para consultas interactivas.
	DQR2: Métrica de completitud mínima del 99 % para campos obligatorios (`title`, `reviewText`).
Business rules	BR1: El campo `score` debe estar entre 1 y 5.
	BR2: La combinación `(title, userId)` debe ser única (una reseña por usuario y libro).
Problemas de CD	PC1: Valores nulos en `reviewText` o `title`.
	PC2: Títulos vacíos.
	PC3: Calificaciones (`overall`) fuera del rango 1–5.
	PC4: Fechas con formato inconsistente (08/2/2020, 2020/13/01).
	PC5: Precios inverosímiles (p. ej., 0 USD o 1000 USD).
	PC6: Identificadores que enlazan a títulos distintos (ediciones mezcladas).
	PC7: Reseñas idénticas publicadas múltiples veces.
	PC8: Múltiples alias de usuario que corresponden a la misma persona.
	PC9: Reseñas generadas por *bots*.
	PC10: Campo `reviewTime` almacenado en zonas horarias heterogéneas.
	PC11: Uso de emojis no contemplados por el procesamiento NLP básico.
	PC12: Mezcla de idiomas en `reviewText`.

Tabla D.1: Componentes de contexto y problemas de CD identificados en el dominio de reseñas de libros para el caso de estudio 1, donde se analizan las funcionalidades de la herramienta

Anexo E

Caso de Estudio 2: Interoperabilidad

En este anexo se presenta el conjunto de datos, el modelo de contexto y el modelo de CD usados en el caso de estudio 2. En la tabla E.1 se muestra la descripción de los datos. Por otro lado en la tabla E.2 se presenta el modelo de contexto generado por el equipo encargado de la Fase 1. Ademas, en las tablas E.3 y E.4 se muestra una comparación entre el modelo de contexto de referencia y el modelo de contexto definido por el proyecto de grado encargado de la Fase 1 - *DQ Planning*. Finalmente, en la tabla E.6 se presenta el modelo de CD generado durante la Fase 2 - *DQ Assessment*.

E.1. Conjunto de Datos

En la siguiente tabla se pueden observar los atributos de la tabla antibióticos, sobre la cual se realizo este caso de estudio.

Atributos	**Descripción**
Fecha	Fecha de dosificación del antibiótico (concentración de un fármaco en un fluido biológico en un momento dado).
Registro	Número de registro que identifica al paciente.
ATB	Antibiótico suministrado: Amikacina (Amika), Vancomicina (Vanco) o Gentamicina (Genta).
Posología	Dosis en la que se administran los medicamentos.
Vía	Vía de administración de la droga.
Día.Últ.Dosis	Día de la última dosis de antibiótico.
RazónTrat	Razón del tratamiento (texto libre).
Estado	Estado clínico del paciente.
IR	Indica si el paciente presenta insuficiencia renal.
Crea	Valor de Creatinina.
Diálisis	Indica si el paciente está en diálisis.
Conc.Valle	Concentración en valle (Cmin), previa a la siguiente dosis.
Conc.Pico	Concentración en pico (Cmáx), máxima alcanzada.
Conc.Cont	Concentración continua.
Conc.PreHD	Concentración antes de hemodiálisis.
Conc.PostHD	Concentración después de hemodiálisis.
Conc.LCR	Concentración registrada solo cuando el paciente recibe Vancomicina y el fluido biológico es LCR.
Conc	Concentración cuando se desconoce el momento de extracción.
Comentarios	Comentarios realizados al momento de la dosificación (texto libre, puede estar vacío).

Tabla E.1: Descripción de los atributos del dataset usado en el caso de estudio 2

E.2. Modelo de Contexto

En la tabla E.2 se presenta el modelo de contexto proporcionado por el de proyecto de grado encargado de la Fase 1, definido mediante la ejecución de su herramienta. Este modelo de contexto fue el utilizado para la ejecución del caso de estudio 2, donde analizamos la interoperabilidad entre las 2 herramientas.

Categoría	**Componente de contexto**
Application domain	**AD1: Farmacología**
	AD2: Monitoreo terapéutico
Business rules	BR3: fecha $\leq$ fecha última dosis
	BR6: preHD $\neq$ null $\vee$ posHD $\neq$ null $\rightarrow$ Hemodiálisis = 'HD', IR = 'si', Creatinina $>$ 1.2 mg/dL
	BR7: dialisis = 'HD' $\rightarrow$ IR $\in$ Si, En HD $\wedge$ Creatinina $>$ 1.2 mg/dL
	BR15: Via = 'VO' $\rightarrow$ ATB = 'Vancomicina' $\wedge$ razontrat = 'clostridium' $\wedge$ (comentarios LIKE ïndetectable"$\vee$ comentarios LIKE "no cuantificable")
	BR16: Conc.LCR $\neq$ null $\rightarrow$ Fluido biológico = 'LCR'
	BR17: Posología LIKE '%BIC%' $\rightarrow$ ATB = 'vanco'
	BR18: (IR = Si $\vee$ IR = En HD) $\wedge$ Crea = null $\rightarrow$ dialisis $\neq$ null
	BR19: 0.17 $<$ crea $<$ 20 mg/dL
	BR20: crea $>$ 1.2 mg/dL $\rightarrow$ IR = 'si'
	BR21: Conc.Cont $\neq$ null $\rightarrow$ posologia LIKE '%BIC%'
	BR22: Conc.LCR $\neq$ null $\rightarrow$ ATB = 'vanco'
	BR23: Conc.LCR $\leq$ 10 mg/L
Tasks at hand	T4: Análisis de datos - Se utiliza como fuente de datos para estudios o investigaciones, con fines estadísticos en análisis descriptivos
	T12: Recolección de datos - Recolección de datos brindados por los médicos
Data filtering needs	DF8: Filtrar por fecha
	DF9: Filtrar por número de registro
	DF10: Datos que se encuentran en la columna Conc
Users characteristics	UC5: Estudiantes - Estudiantes que estén haciendo trabajos por créditos. Estudiantes que estén haciendo el practicantado (pasantía final de la carrera)
	UC13: Docentes - El grupo docente, mayormente grados 1
	UC14: Médicos - Médicos que proveen los datos
DQ requirements	DQR24: fecha $\neq$ null
	DQR25: count(via = 'sin dato' $\vee$ null) $<$ 20 %
	DQR26: count(Posología = null) $<$ 20 %
	DQR27: count(Día.Últ.Dosis = null) $<$ 20 %
	DQR28: count(Estado = null) $<$ 20 %
	DQR29: count(IR = null) $<$ 20 %
	DQR30: count(crea $\neq$ null) $<$ 20 %
	DQR31: count(dialisis $\neq$ null) $<$ 20 %
	DQR32: format(Fecha) = 'dd.mm.aa' and format(Día.Últ.Dosis) = 'dd.mm.aa'
System requirements	SR11: Espacio de almacenamiento de datos - Base de datos para almacenar información de análisis de dosificación de antibióticos

Tabla E.2: Componentes de contexto identificados con la herramienta de Fase 1 - DQ Planning.

La Tabla E.3 presenta el modelo de contexto de referencia, comparado con el modelo de contexto definido con la herramienta del equipo encargado de la Fase 1. Las distintas celdas de la tabla se encuentran coloreadas de verde, amarillo o rojo, que significa que el componente de contexto se encuentra en el modelo de contexto definido por la herramienta de la Fase 1, si tiene alguna diferencia o si no esta presente, respectivamente.

Categoría	**Componente de contexto**
Application domain	**AD: Health (Farmacología, Monitoreo terapéutico)**
Business rules	BR1: ATB = "amika" (Amikacina) or "vanco" (Vancomicina) or "genta" (Gentamicina)
	BR2: Fluido biológico = "Plasma" or "Líquido Cefalorraquídeo" (LCR)
	BR3: If notNull(Conc.LCR) → Fluido biológico = "LCR"
	BR4: Registros duplicados representan al mismo paciente
	BR5: If "BIC" in Posología → ATB = "vanco"
	BR6: via = "VO", "IV", "bolsa peritoneal", "intraperitoneal", "intraventricular", "intratecal", "intramuscular"
	BR7: If via = "VO" → ATB = "vanco"
	BR8: If notNull(Conc.LCR) → ATB = "vanco"
	BR9: Día.Últ.Dosis > fecha
	BR10: Distance(Día.Últ.Dosis, fecha) $\leq$ 1 week
	BR11: If (fecha Día.Últ.Dosis) > 3 → Posología = "Suspendida"
	BR12: diálisis = "hemodiálisis" → (IR = "Si" or IR = "En HD") or (Crea $\iff$ 1.2 mg/dL)
	BR13: 0.17 mg/dL $\leq$ Crea $\leq$ 20 mg/dL
	BR14: Crea $\leftrightarrow$ 1.2 mg/dL → IR $\neq$ "Si"
	BR15: If notNull(Conc.Cont) → "BIC" in Posología
	BR16: If notNull(Conc.preHD) → Diálisis = "hemodiálisis" and (IR = "si" OR IR = "en HD")
	BR17: If notNull(Conc.postHD) → Diálisis = "hemodiálisis" and (IR = "si" OR IR = "en HD")
	BR18: If diálisis = "diálisis peritoneal" → Conc.PreHD = NULL and Conc.PosHD = NULL
	BR19: dosis de vanco: múltiplos de 250 mg
	BR20: dosis de amika: múltiplos de 100 mg o 250 mg
	BR21: dosis de genta: múltiplos de 20 mg
	BR22: Todas las concentraciones se miden en mg/L
Users characteristics	UC1: docentes
	UC2: estudiantes
	UC3: médicos
Tasks at hand	T1: registro de datos
	T2: análisis de datos para investigación
	T3: análisis de la evolución estadística (nº de dosificaciones, calidad de la info recibida)
	T4: proveer datos
Data filtering needs	DF1: consultar si un paciente tiene dosificaciones previas
	DF2: proporción de datos que se encuentran en la columna Conc.
	DF3: para Via="VO" y RazónTrat="clostridium", cuántos pacientes presentan concentración
DQ requirements	DQR1: el atributo fecha tiene formato DD.MM.YY
	DQR2: el atributo Día.Últ.Dosis tiene formato DD.MM.YY
	DQR3: 100 % de los registros debe tener el campo fecha
	DQR4: representatividad mínima de variables (ej. 50 % vs. 80 %)
	DQR5: si un paciente tiene 2 registros con "diálisis peritoneal" y "hemodiálisis", debe aclararse en comentarios
	DQR6: If IR = "En HD" → Crea not NULL
	DQR7: If notNull(Conc.preHD) → Crea 1.2 mg/dL
	DQR8: If notNull(Conc.postHD) → Crea 1.2 mg/dL
	DQR9: "error de extracción" sin "hora de extracción" → inválido
	DQR10: válido si solo un valor no nulo entre Conc.Valle, Conc.Pico, Conc.Cont, Conc.PreHD, Conc.PostHD, Conc.LCR, Conc
	DQR11: (ATB=vanco y "<3.0" en comentarios) → concentración=3.0
	DQR12: (ATB=amika y "<2.3" en comentarios) → concentración=2.3
	DQR13: (ATB=genta y "<0.3" en comentarios) → concentración=0.3

Tabla E.3: Comparación entre el modelo de contexto de referencia y el modelo de contexto definido por la herramienta encargada de la Fase 1 (colores: verde = coincide, amarillo = diferencias, rojo = no presente).

Hay algunos componentes de contexto que fueron identificados por el equipo encargado de la Fase 1 que no se encontraban en el modelo de contexto de referencia, estos se presentan en la Tabla E.4.

Categoría	**Componente de contexto**
Business rules	BR18: If (IR = Si or IR = En HD) and Crea null → diálisis not null
	BR23: Conc.LCR $\leq$ 10 mg/L
System requirements	SR11: Espacio de almacenamiento de datos.
Data filtering needs	DF8: Filtrar por fecha
	DF10: Datos que se encuentran en la columna Conc.

Tabla E.4: Componentes de contexto presentes en el modelo de contexto definido en la Fase 1, no presentes en el modelo de contexto de referencia.

Problemas de CD

En la Tabla E.5 se presentan los problemas de CD, junto con su identificador y la prioridad asignada durante la ejecución de la herramienta para la Etapa 4 de CaDQM. Considerando que dichos problemas surgieron a través de un proceso de análisis exhaustivo en la Fase 1 de CaDQM, todos fueron seleccionados para brindar apoyo en la definición del modelo de CD.

Id	**Descripción**	**Prioridad**
PC1	Inconsistencias en el formato de los valores de concentración	Alta
PC2	Formato incorrecto en los campos Día.Últ.Dosis y fecha	Alta
PC3	En la columna ATB se utilizan opciones que están por fuera de las predefinidas.	Media
PC4	Las concentraciones pueden ser un valor numérico o un rango, dependiendo de lo que se indique en los comentarios.	Baja
PC5	En la columna ‘ATB‘ se usan abreviaciones de las opciones predefinidas.	Media
PC6	Ninguna fecha está en el formato indicado DD.MM.YY	Alta
PC7	67 filas tienen el campo fecha vacío.	Alta
PC8	108 filas tienen el campo ‘registro‘, que identifica al paciente, vacío.	Alta
PC9	No se encuentra el valor predefinido ‘HD‘ en la columna ‘dialisis‘. En su lugar se encuentra ‘hemodiálisis‘.	Media
PC10	No se cumple la regla de negocio BR6	Alta
PC11	24 filas no cumplen la regla de negocio BR15.	Media
PC12	Inconsistencias menores en formatos de fechas	Media
PC13	El campo ‘diaultimadosis‘ puede no contener datos o contener datos no confiables.	Media
PC14	Es posible que se registren errores en la entrada de concentraciones sin aclaración en comentarios	Media
PC15	En razón del tratamiento se utiliza el valor ‘sin dato‘ y ‘se desconoce‘ (valor predefinido) para indicar que se desconoce la razón del tratamiento.	Baja
PC16	Hay casos particulares donde el usuario está en diálisis peritoneal y hemodiálisis, pero el campo “Diálisis” solo tiene un valor. Esos casos se aclaran en la columna “Comentarios”.	Baja
PC17	843 filas no cumplen la regla de negocio BR21.	Alta

Tabla E.5: Problemas de CD identificados con la herramienta de la Fase 1 - *DQ Planning* de CaDQM para el dataset de antibióticos, priorizados con la herramienta de la Fase 2 - *DQ Assessment*.

E.3. Modelo de CD

En las Tablas E.2 y E.3.1 se presenta el modelo de CD definido mediante la aplicación de la herramienta propuesta . Con el fin de facilitar la lectura del modelo de CD, este se divide en dos tablas, una para dimensiones y factores, y otra para métricas, métodos y métodos aplicados.

Dimensión	**Factor**	**"Surgen de" (Problemas de CD / Componentes de Contexto)**
Completeness	*Density*	PC8, PC7; DQR27, DQR26, DQR28, DQR29, DQR24, DQR30, DQR31; UC13, UC14; SR9
	Coverage	DQR25; UC13, UC14
Consistency	*Domain Integrity*	BR19, BR23
	Intra-relationship Integrity	PC17, PC11, PC10; BR6, BR7, BR15, BR16, BR17, BR18, BR21, BR22, BR20
Accuracy	*Syntactic Accuracy*	PC2, PC12; DQR32
	Semantic Accuracy	PC14, PC4; UC14; BR3; T4
	Precision	PC14, PC4; AD1, AD2; T4
Uniqueness	*No-duplication*	PC7, PC8; DF9, DF8; UC6, UC4
Freshness	*Currency*	PC12; DF8

Tabla E.6: Dimensiones y factores del modelo, vinculados a problemas de CD y componentes de contexto identificados en la Fase 1 - *DQ Planning*.

E.3.1. Métricas, Métodos y Métodos de CD Aplicados

Por cada factor seleccionado se definieron métricas y métodos en base a los distintos componentes de contexto y problemas de CD. En la Tabla E.3.1 se presentan todos estos conceptos de CD, indicando a qué factor están asociados, cuál es su granularidad y tipo de resultado, la lógica de su algoritmo y finalmente a qué atributos se aplicaron los métodos.

Tabla E.7: Métricas con componentes de contexto y detalle de métodos (incluye algoritmo) y su aplicación

Métrica: *Non Null Value Ratio* (Factor: *Density*)	
Granularidad: Column Dominio resultado: Float [0,1] Componentes de Contexto: U13,U14; DQ27,DQ26,DQ28,DQ29,DQ24,DQ30,DQ31.	
Método	**Métodos aplicados**

Continued on next page

Tabla E.7: Métricas con componentes de contexto y detalle de métodos (incluye algoritmo) y su aplicación (Continued)

calculateNonValueRatio: proporción de valores no nulos por atributo. *Componentes :* U13,U14; DQ27,DQ26,DQ28,DQ29,DQ24,DQ30,DQ31. *Algoritmo:* `SELECT SUM(CASE WHEN {{field}} IS NOT NULL THEN 1 ELSE 0 END) / COUNT(*) FROM {{table name}}`	*Aplicado a:* `antibioticos.IR` *Implementación:* `SELECT SUM(CASE WHEN ir IS NOT NULL THEN 1 ELSE 0 END)::float / COUNT(*) FROM antibioticos;`
Métrica: *Valid Value Ratio* (Factor: *Coverage*)	
Granularidad: Column Dominio resultado: Float [0,1] Componentes de Contexto: U13,U14; DQ25.	
Método	**Métodos aplicados**
calculateValidValueRatio: porcentaje de valores válidos (no nulos y distintos de "no hay dato"). *Componentes :* U13,U14; DQ25. *Algoritmo:* `SELECT SUM(CASE WHEN {{condition}} IS NOT NULL THEN 1 ELSE 0 END) / COUNT(*) FROM {{table name}}`	*Aplicado a:* `antibioticos.Vía` *Implementación:* `SELECT SUM(CASE WHEN via IS NOT NULL AND TRIM(via) <> 'no hay dato' THEN 1 ELSE 0 END)::float / NULLIF(COUNT(*),0) FROM antibioticos;`
Métrica: *Value Range Compliance Ratio* (Factor: *Domain Integrity*)	
Granularidad: Column Dominio resultado: Float [0,1] Componentes de Contexto: BR19,BR23.	
Método	**Métodos aplicados**
calculateValueRangeComplianceRatio: porcentaje de valores dentro del rango esperado. *Componentes :* BR19,BR23. *Algoritmo:* `WITH limits AS ( SELECT {{lower_limit}} AS lower_limit, {{upper_limit}} AS upper_limit ) SELECT 1 - COUNT(*) * 1.0 / (SELECT COUNT(*) FROM {{table}}) AS {{out_of_range_ratio_alias}} FROM {{table}} a CROSS JOIN limits l WHERE a.{{field}} < l.lower_limit OR a.{{field}} > l.upper_limit;`	*Aplicado a:* `antibioticos.Crea` *Implementación:* `WITH limits AS (SELECT 0.17 AS lower_limit, 2.0 AS upper_limit) SELECT 1 - COUNT(*) * 1.0 / (SELECT COUNT(*) FROM antibioticos) FROM antibioticos a CROSS JOIN limits l WHERE a.crea < l.lower_limit OR a.crea > l.upper_limit;`
Métrica: *Constraint Satisfaction Ratio* (Factor: *Intra-relationship Integrity*)	
Granularidad: Table Dominio resultado: Float [0,1] Componentes de Contexto: BR6,BR7,BR15,BR16,BR17,BR18,BR21,BR22.	
Método	**Métodos aplicados**

Continued on next page

Tabla E.7: Métricas con componentes de contexto y detalle de métodos (incluye algoritmo) y su aplicación (Continued)

calculateConstraintSatisfactionRatio: reglas de consistencia entre columnas. *Componentes :* BR6, BR7, BR15, BR16, BR17, BR18, BR21,BR22. *Algoritmo:* `SELECT 1 - {{violaciones}} / NULLIF(COUNT(*),0)`	*Aplicado a:* `(IR, Crea, Diálisis, Conc.PreHD, Conc.PostHD)` *Implementación:* `SELECT 1 - SUM(CASE WHEN (dialisis NOT IN ('hemodiálisis','HD','hd','en hd','en hemodiálisis') AND ir <> 'si' AND crea < 1.2) AND (conc_prehd IS NOT NULL OR conc_posthd IS NOT NULL) THEN 1.0 ELSE 0.0 END) / COUNT(*) FROM antibioticos;`
Métrica: *Syntax Compliance Ratio* (Factor: *Syntactic Accuracy*)	
Granularidad: Column Dominio resultado: Float [0,1] Componentes de Contexto: DQ32.	
Método	**Métodos aplicados**
calculateSyntaxComplianceRatio: valida patrón de fecha esperado. *Componentes :* DQ32. *Algoritmo:* `SELECT SUM(CASE WHEN {{condition}} IS NOT NULL THEN 1 ELSE 0 END) / COUNT(*) FROM {{table name}}`	*Aplicado a:* `antibioticos.Fecha` *Implementación:* `SELECT SUM(CASE WHEN fecha ~ '^(0?[1-9]|1[0-2])/(0?[1-9]|[12][0-9]|3[01])/\d{4}$' THEN 1 ELSE 0 END )::float / NULLIF(COUNT(*),0) FROM antibioticos;`
Métrica: *Semantic Rule Compliance* (Factor: *Semantic Accuracy*)	
Granularidad: Column Dominio resultado: Float [0,1] Componentes de Contexto: U14; T4; BR3.	
Método	**Métodos aplicados**
calculateSemanticRuleCompliance: consistencia entre fechas (fecha vs. última dosis). *Componentes :* U14; T4; BR3. *Algoritmo:* `SELECT CASE WHEN {{condition}} THEN TRUE ELSE FALSE END FROM {{table name}}`	*Aplicado a:* `(Fecha, Día.Últ.Dosis)` *Implementación:* `SELECT CASE WHEN fecha < dia_ult_dosis THEN TRUE ELSE FALSE END FROM antibioticos;`
Métrica: *Detail Level By Row* (Factor: *Precision*)	
Granularidad: Tuple Dominio resultado: Boolean Componentes de Contexto: U13,U14; AD1,AD2; T4.	
Método	**Métodos aplicados**

Continued on next page

Tabla E.7: Métricas con componentes de contexto y detalle de métodos (incluye algoritmo) y su aplicación (Continued)

detailLevelByRow: verifica si se cumple condición según presencia de mediciones y comentarios. *Componentes :* U13,U14; AD1,AD2; T4. *Algoritmo:* `SELECT {{condition}} FROM {{table name}}`	*Aplicado a:* `(Conc.Valle, Conc.Pico, Conc.Cont, Conc.PreHD, Conc.PostHD, Conc.LCR, Conc, Comentarios)` *Implementación:* `SELECT ((conc_lcr IS NOT NULL OR conc_valle IS NOT NULL OR conc_pico IS NOT NULL OR conc_cont IS NOT NULL OR conc_prehd IS NOT NULL OR conc_posthd IS NOT NULL OR conc IS NOT NULL) AND comentarios IS NULL) FROM antibioticos;`
Métrica: *Non Duplicate Entry Ratio* (Factor: *No-duplication*)	
Granularidad: Table Dominio resultado: Float [0,1] Componentes de Contexto: U6,U4; DF9,DF8.	
Método	**Métodos aplicados**
calculateNonDuplicateEntryRatio: porcentaje de entradas no duplicadas. *Componentes :* U6,U4; DF9,DF8. *Algoritmo:* `SELECT 1- select key / total amount FROM table name`	*Aplicado a:* `(Fecha, Registro)` *Implementación:* `SELECT 1 - (CAST(SUM(dup_count) AS FLOAT) / COUNT(*)) FROM (SELECT fecha, registro, COUNT(*) AS cnt, CASE WHEN COUNT(*) > 1 THEN COUNT(*) - 1 ELSE 0 END AS dup_count FROM antibioticos GROUP BY fecha, registro) t;`
Métrica: *Recent Data* (Factor: *Currency*)	
Granularidad: Tuple Dominio resultado: Boolean Componentes de Contexto: DF8.	
Método	**Métodos aplicados**
isRecentData: marca si la tupla es reciente según un umbral temporal. *Componentes :* DF8. *Algoritmo:* `SELECT date field > threshold date FROM table name`	*Aplicado a:* `antibioticos.Fecha` *Implementación:* `SELECT TO_DATE(fecha,'MM/DD/YY') > '01/01/2015' FROM antibioticos;`