

INSTITUTO DE COMPUTACIÓN  
FACULTAD DE INGENIERÍA

UNIVERSIDAD DE LA REPÚBLICA



# CONSTRUCCIÓN DE HERRAMIENTA PARA LA EVALUACIÓN DE CARRERAS.

PROYECTO DE GRADO

ANA CLARA ESPONDA POU  
DIEGO DASTUGUE SADI  
MARINA ACOSTA PEDEMONTE

TUTORES

LUCÍA CAMILLONI  
DIEGO VALLESPÍR  
LAURA GONZÁLEZ

URUGUAY, MONTEVIDEO.

AGOSTO 2018







## RESUMEN

---

La evaluación de carreras es la recopilación, revisión y utilización de los datos para el mejoramiento continuo de los programas educativos. El objetivo de evaluar una carrera, es enfocarse en mejorar el aprendizaje y desarrollo de los estudiantes. Hoy en día, es común que las universidades se interesen por evaluar si sus programas de estudio cumplen con los objetivos educativos y los resultados de aprendizaje esperados.

En el marco de una tesis de posgrado de la Maestría en Ingeniería de Software dictada en la Universidad de la República, se define un método de evaluación de cubrimiento temático, enfocado en evaluar una carrera con respecto a su cuerpo de conocimiento. El cuerpo de conocimiento de una disciplina es el compendio del dominio de una profesión. En él se organizan los conceptos, términos, técnicas y actividades que constituyen al dominio aceptado por la comunidad profesional relevante en el campo. En este sentido, el método definido tiene por objetivo evaluar si un plan de estudio cubre los elementos definidos en el cuerpo de conocimiento.

Las principales organizaciones profesionales han creado guías curriculares para algunas disciplinas de la computación. En el caso de la Ingeniería de Software, existe una guía curricular para programas de posgrado, que define un cuerpo de conocimiento. El método de evaluación de cubrimiento temático antes mencionado, fue aplicado para evaluar el programa de la Especialización en Ingeniería de Software de la UDELAR, con respecto a este cuerpo de conocimiento.

La evaluación de la Especialización se llevó a cabo desde su creación en el 2012 a través del registro manual de horas dedicadas a los elementos del cuerpo de conocimiento. Debido al volumen creciente de datos y a la dificultad de mantenerlos, surge la necesidad de contar con una herramienta que facilite este proceso y permita agregar nuevas facilidades a la evaluación.

En el marco de este proyecto de grado, se construyó un sistema web, que da soporte a la evaluación de cubrimiento temático. El sistema se amolda a la evaluación realizada para la Especialización, y su base de datos se carga con toda la información de cubrimiento recolectada hasta el momento. Además, el diseño se realizó de forma genérica, de modo que la herramienta puede ser utilizada para evaluar otras carreras con respecto a otros cuerpos de conocimiento.

**PALABRAS CLAVE:** Evaluación de carreras, cuerpos de conocimiento, métodos de evaluación, cubrimiento temático, Core Body Of Knowledge, Especialización en Ingeniería de Software, Construcción herramienta web.

## ÍNDICE GENERAL

---

1	INTRODUCCIÓN	1
1.1	Objetivos	2
1.2	Aportes del proyecto	2
1.3	Organización del documento	3
2	MARCO CONCEPTUAL	5
2.1	Cuerpos de conocimiento	5
2.1.1	SWEBOK	6
2.2	Guías curriculares	9
2.3	GSwE2009	10
2.3.1	Core Body of Knowledge	11
2.4	Evaluación de carreras	13
2.5	Cubrimiento temático	14
3	ANÁLISIS DE LA REALIDAD	17
3.1	Aplicación de la evaluación de cubrimiento temático para la EIS	20
3.1.1	Cubrimiento temático	20
3.1.2	Esfuerzo temático	23
3.1.3	Balance temático	26
4	REQUISITOS DEL SISTEMA	29
4.1	Sistema a construir	29
4.2	Usuarios del sistema	31
4.3	Requerimientos funcionales	32
4.4	Requerimientos no funcionales	37
4.5	Interfaces de usuario	38
5	AQUITECTURA PROPUESTA	39
5.1	Casos de uso críticos	39
5.2	Vista lógica del sistema	45
5.3	Vista de distribución del sistema	51
5.4	Vista de Implementación del sistema	52
5.5	Vista de seguridad	53
5.6	Vista de datos	53
6	CONSTRUCCIÓN DEL SISTEMA	57
6.1	Tecnología y herramientas utilizadas	57
6.1.1	Servidor de Back-End	57
6.1.2	Servidor de Front-End	60
6.2	Implantación	63

6.3	Alternativas estudiadas	63
7	CARGA DE DATOS	67
7.1	Datos existentes	67
7.2	Automatización de la carga	71
7.3	Trabajo Realizado	72
7.3.1	Dificultades encontradas	75
8	GESTIÓN DEL PROYECTO	77
8.1	Planificación	77
8.1.1	Fases del proyecto	80
8.1.2	Ejecución real respecto a la planificación inicial	81
8.2	Proceso utilizado	85
8.2.1	Gestión de incidencias	86
8.3	Gestión de riesgos	87
8.4	Gestión de la configuración	91
8.5	Gestión de la calidad	92
9	CONCLUSIONES DEL TRABAJO REALIZADO	95
9.1	Lecciones aprendidas	97
9.2	Trabajo a futuro	100
BIBLIOGRAFÍA 103		
A	DOCUMENTO DE REQUISITOS	105
A.1	Sistema a construir	105
A.2	Usuarios	106
A.3	Requerimientos funcionales	107
A.4	Requerimientos no funcionales	119
A.5	Restricciones	120
A.6	Interfaces	120
B	DICCIONARIO BASE DE DATOS	123
B.1	Estructura y tablas	123
B.2	Código para insertar elementos	148
C	MANUAL DE INSTALACIÓN	153
C.1	Máquina virtual	153
C.2	Instalación del sistema en sistema operativo Ubuntu	158
C.2.1	Instalación de la base de datos	158
C.2.2	Instalación del <i>Back End</i>	160
C.2.3	Instalación del <i>Front End</i>	161
C.2.4	Inicializar el sistema	161

## ÍNDICE DE FIGURAS

---

Figura 1	Cubrimiento temático para área Testing.	21
Figura 2	Cubrimiento temático para área Ética y conducta profesional.	22
Figura 3	Cubrimiento temático a nivel de unidades para cada área del CBOOK.	23
Figura 4	Cubrimiento temático considerando todas las unidades del CBOOK.	23
Figura 5	Esfuerzo temático para la EIS 2012-2013 contra el CBOOK.	25
Figura 6	Balance temático para la EIS 2012-2013 contra el CBOOK.	27
Figura 7	Balance temático para la EIS 2012-2013 contra el CBOOK.	28
Figura 8	Casos de uso críticos para la definición de la arquitectura	41
Figura 9	Vista lógica	45
Figura 10	Componentes vista lógica	46
Figura 11	Diagrama de interacción - Importar dedicaciones	48
Figura 12	Diagrama de interacción - Mapeo de dedicaciones	49
Figura 13	Diagrama de interacción - Visualización de esfuerzo temático	50
Figura 14	Vista de distribución	51
Figura 15	Vista de implementación	52
Figura 16	Modelo entidad relación	54
Figura 17	Diagrama de interacción de las tecnologías y herramientas utilizadas.	62
Figura 18	Transformación realizada para cargar planillas con registro directo sobre tópicos.	74
Figura 19	Transformación realizada para cargar planillas con detalle de temas dictados.	75
Figura 20	Planificación inicial del proyecto.	78
Figura 21	Ejecución real del proyecto.	79
Figura 22	Diagrama de flujo de los estados que atraviesan las incidencias.	86
Figura 23	Porcentaje de cubrimiento mínimo - Cuerpo conocimiento	114

Figura 24	Porcentaje de cubrimiento mínimo - Análisis cubrimien- to tópico	116
Figura 25	Porcentaje de cubrimiento mínimo - Análisis cubrimien- to unidad	117
Figura 26	Diagrama de Base de datos	124
Figura 27	Pantalla de importación de una máquina virtual.	154
Figura 28	Mensaje de error sobre preferencias de red.	155
Figura 29	Panel de preferencias de red.	155
Figura 30	Mensaje de éxito al levantar el sistema.	156
Figura 31	Escritorio de la máquina virtual una vez que se inicia sesión.	157
Figura 32	Resultado esperado de la lista usuarios y permisos.	159
Figura 33	Configuración para crear la conexión a la base de da- tos.	160

## ÍNDICE DE CUADROS

---

Cuadro 1	Áreas de conocimiento SWEBOK 2004	8
Cuadro 2	Áreas de conocimiento SWEBOK v3	9
Cuadro 3	Porcentajes sugeridos para las áreas del CBOOK	24
Cuadro 4	Esfuerzo: Dedicación en horas esperada para las áreas del CBOOK	25
Cuadro 5	Balance: Porcentaje esperado para las áreas del CBOOK	27
Cuadro 6	Especificación de caso de uso: Importación de datos desde planillas Excel	42
Cuadro 7	Especificación de caso de uso: Mapeo de dedicaciones a ítems de un cuerpo de conocimiento.	43
Cuadro 8	Especificación de caso de uso: visualización de esfuerzo temático	44
Cuadro 9	Funcionalidades pendientes.	83
Cuadro 10	Funcionalidades que incluye el sistema desarrollado.	84
Cuadro 11	Tipos de datos Postgres	125
Cuadro 12	Tabla <i>authorities</i>	126
Cuadro 13	Tabla <i>bodies_of_knowledge</i>	126
Cuadro 14	Tabla <i>bodies_of_knowledge_careers</i>	127
Cuadro 15	Tabla <i>bodies_of_knowledge_careers_responsible_of_mapping</i>	127
Cuadro 16	Tabla <i>bodies_of_knowledge_study_programme</i>	128
Cuadro 17	Tabla <i>bodies_of_knowledge_study_programme_hours_configuration.</i>	128
Cuadro 18	Tabla <i>careers</i>	129
Cuadro 19	Tabla <i>careers_responsible</i>	129
Cuadro 20	Enumerado de tipos de hora del sistema.	130
Cuadro 21	Tabla <i>configuration_profiles</i>	130
Cuadro 22	Tabla <i>configuration_profiles_knowledge_items</i>	131
Cuadro 23	Tabla <i>configuration_profiles_levels_effort_and_balance_applies</i>	132
Cuadro 24	Tabla <i>configuration_profiles_parent_level</i>	132
Cuadro 25	Tabla <i>configuration_profiles_parent_level_coverage_ranges</i>	133
Cuadro 26	Tabla <i>courses</i>	133
Cuadro 27	Tabla <i>course_editions</i>	134
Cuadro 28	Tabla <i>courses_edition_themes</i>	135
Cuadro 29	Tabla <i>course_editions_executions_of_implementations</i>	135
Cuadro 30	Tabla <i>courses_plan_implementations</i>	136

Cuadro 31	Tabla <i>coverage_ranges</i>	137
Cuadro 32	Tabla <i>dedications</i>	138
Cuadro 33	Tabla <i>executions_of_implementation</i>	139
Cuadro 34	Tabla <i>knowledge_item_hours</i>	140
Cuadro 35	Tabla <i>knowledge_items</i>	141
Cuadro 36	Tabla <i>knowledge_levels</i>	142
Cuadro 37	Tabla <i>minimum_hour_criteria_leaf_level</i>	143
Cuadro 38	Tabla <i>people</i>	144
Cuadro 39	Tabla <i>plan_implementations</i>	144
Cuadro 40	Tabla <i>study_programmes</i>	145
Cuadro 41	Tabla <i>themes</i>	146
Cuadro 42	Tabla <i>type_of_hour_configuration</i>	146
Cuadro 43	Tabla <i>users</i>	147
Cuadro 44	Tabla <i>users_authorities</i>	147
Cuadro 45	Tabla <i>users_careers</i>	148

## INTRODUCCIÓN

---

La evaluación de carreras es la recopilación, revisión y utilización de la información sobre los programas educativos y tiene por objetivo mejorar el aprendizaje y el desarrollo de los estudiantes. En este sentido, la Facultad de Ingeniería de la Universidad de la República (UDELAR) desea construir un sistema que de soporte a la evaluación de sus carreras.

Un enfoque utilizado para evaluar una carrera consiste en considerar lo que los estudiantes aprenden en los cursos en lugar de lo que les es enseñado. Con este enfoque se busca evaluar si se cumple con los resultados esperados al egreso y con los objetivos educativos del programa, permitiendo que los profesores tengan mayor flexibilidad en la forma de plantear los cursos.

De todos modos, es importante conocer qué sucede realmente con los temas impartidos durante la carrera. De esta forma, interesa poder definir qué temas se desean impartir en una carrera y a su vez tener indicadores sobre los niveles de conocimiento o habilidad que se están adquiriendo.

En el año 2012 en la Facultad de Ingeniería de la UDELAR comenzaron a dictarse una Especialización en Ingeniería de Software (EIS) y una Maestría en Ingeniería de Software (MIS). Ambos planes de estudio están basados en la guía curricular para maestrías profesionales en Ingeniería de Software (GSWE2009) y han sido evaluados desde sus comienzos para detectar oportunidades de mejora.

En este sentido, se definió un método que permite evaluar el cubrimiento temático de una carrera con respecto a un cuerpo de conocimiento de referencia. Este método es aplicado desde el 2012 para evaluar la EIS y podría ser aplicable a otras carreras. Además en el 2015 se realizaron encuestas a los estudiantes y egresados para conocer su percepción respecto al grado de cumplimiento de los resultados esperados del plan de estudio.

Para almacenar todos estos datos se utilizan planillas electrónicas, que resultan difíciles de mantener a medida que el volumen de datos crece. Por este motivo, se desea contar con un sistema de información de acceso web, que soporte los distintos aspectos de la evaluación de carreras, y cuente con una base de datos relacional como repositorio de datos. Se espera que la base de datos contenga toda la información recolectada hasta el momento y que el sistema

sea flexible como para poder ser usado al evaluar otras carreras de la UDELAR e incluso de otras universidades del mundo.

### 1.1 OBJETIVOS

El proyecto tiene dos objetivos generales: proponer una herramienta que permita dar soporte a la evaluación de carreras, y utilizar la información de evaluaciones previas de la EIS para generar nuevos resultados a través del sistema.

El equipo debe proponer una herramienta que permita realizar la evaluación de carreras considerando las distintas perspectivas, como ser el cubrimiento temático y el cumplimiento de los resultados esperados al egreso. La herramienta debe ser flexible como para soportar la evaluación de distintos tipos de carreras, y a su vez, ajustarse a la evaluación realizada para la EIS.

En lo que respecta a la evaluación de la EIS utilizando la herramienta, se espera generar nueva información a través de cruzar los datos de cubrimiento temático con datos de otras evaluaciones, como ser el cumplimiento de los resultados esperados. Para ello, es necesario cargar la información de evaluaciones anteriores en la base de datos del sistema.

### 1.2 APORTES DEL PROYECTO

El principal aporte del proyecto es brindar una herramienta web flexible que permita realizar las distintas evaluaciones de cubrimiento temático de una carrera con respecto a un cuerpo de conocimiento. Dentro de la flexibilidad que ofrece el sistema se permite configurar distintos parámetros para adaptar las evaluaciones según las necesidades de la carrera y el cuerpo que se estén evaluando. Además la herramienta permite simular cambios en los cursos de un plan de estudio, y de este modo evaluar posibles mejoras.

Por otra parte, el proyecto deja disponible una base de datos relacional cargada con la información de cubrimiento temático recolectada, que surgió de las evaluaciones realizadas hasta el momento. La base de datos relacional se entrega junto con el código de la solución en una máquina virtual, además de estar disponible en el entorno web de producción. En este sentido, entre los aportes del proyecto se encuentra la validación de todos los datos, su organización en tablas, el arreglo de las inconsistencias existentes, y el acceso a dicha información a través de consultas SQL y de la interfaz web.

Por último, se consideran como aportes del proyecto hacia el equipo, el aprendizaje de tecnologías que se utilizan actualmente en la industria y la experiencia en el desarrollo de una aplicación web.

### 1.3 ORGANIZACIÓN DEL DOCUMENTO

El presente documento se organiza en nueve capítulos. En el capítulo 2 se explican los conceptos teóricos relacionados a la evaluación de carreras con respecto a cuerpos de conocimiento, es decir, la realidad sobre la que actúa el sistema. En el capítulo 3 se realiza un análisis de la realidad, presentando la EIS y el método utilizado para llevar a cabo su evaluación con respecto al *Core Body Of Knowledge* (CBOK).

Los capítulos 4, 5 y 6 se utilizan para presentar el sistema desarrollado. En primera instancia se presenta el análisis de requisitos, luego la arquitectura propuesta, y por último los detalles de construcción del sistema.

Luego de presentar el sistema, se destina el capítulo 7 a presentar el trabajo realizado para la carga de datos. Primero se introducen las panillas electrónicas resultantes de la evaluación de la EIS disponibles hasta el momento, y luego se explica el mecanismo utilizado para incluir estos datos en el esquema de la base de datos relacional consumida por la herramienta.

En el capítulo 8 se presenta toda la información relativa a la gestión del proyecto. En una primera sección se presenta la separación del proyecto en etapas, su planificación inicial, y la ejecución real de las actividades en el tiempo. Luego se muestra el proceso de trabajo definido y el trabajo realizado con respecto a la gestión de riesgos. Por último, se destinan dos secciones a explicar cómo se realiza la gestión de la configuración y la gestión de la calidad.

Por último, en el capítulo 9 se exponen las conclusiones del trabajo realizado, las lecciones aprendidas y el trabajo a futuro.

Al final de este documento, se encuentran los anexos A, B y C que muestran respectivamente el documento con los requisitos del sistema, un diccionario explicativo de las tablas de la base de datos, y el manual de instalación con guías de cómo levantar el sistema.



## MARCO CONCEPTUAL

---

El objetivo de este capítulo es presentar los conceptos teóricos necesarios para permitir la comprensión del trabajo realizado en el marco de este proyecto. Para lograrlo, se presentan los conceptos involucrados en la evaluación de carreras, específicamente en la evaluación del cubrimiento temático de una carrera en relación a un cuerpo de conocimiento.

En la sección 2.1 se introducen los cuerpos de conocimiento, específicamente el *Software Engineering Body of Knowledge* (SWEBOK) que es la guía para el cuerpo de conocimiento de la Ingeniería de Software.

Luego se presentan las guías curriculares y la guía *Curriculum Guidelines for Graduate Degree Programs in Software Engineering*(GSWE2009) orientada a carreras de posgrado en Ingeniería de Software en las secciones 2.2 y 2.3 respectivamente.

Por último, se introduce la evaluación de carreras en la sección 2.4, explicando en más detalle la evaluación del cubrimiento temático en la sección 2.5. Estas últimas secciones, se basan fuertemente en la investigación realizada por Lucía Camilloni en su tesis de Maestría en Ingeniería de Software de la UDELAR en el año 2015 [Camilloni, 2015].

### 2.1 CUERPOS DE CONOCIMIENTO

El cuerpo de conocimiento de una disciplina es el compendio del dominio de una profesión. En él se organizan los conceptos, términos, técnicas y actividades que constituyen al dominio aceptado por la comunidad profesional relevante en el campo <sup>1</sup>. Debido a que el dominio completo de una disciplina puede ser muy extenso, y encontrarse en diversas bibliografías, surge la necesidad de crear las guías para los cuerpos de conocimiento. Éstas tienen como objetivo dar soporte al cuerpo de conocimiento organizándolo en un único documento. Generalmente, cuando se menciona a un cuerpo de conocimiento, en realidad se hace referencia a su guía. Esta convención se utilizará de ahora en más para el resto de este documento.

Los cuerpos de conocimiento pueden tener distintos enfoques. A pesar de ello, todos mantienen el objetivo de estructurar el conocimiento, dándole un

---

<sup>1</sup> [https://www.regulacioninformatica.org/wiki/index.php?title=Corpus\\_de\\_Conocimiento](https://www.regulacioninformatica.org/wiki/index.php?title=Corpus_de_Conocimiento)

orden a los elementos del dominio, y ayudando a dar madurez al desarrollo de la profesión. Entre las distintas características que pueden tener, los cuerpos de conocimiento pueden enfocarse en facilitar el acceso a la principal bibliografía de la profesión, así como enfocarse en definir las competencias y habilidades necesarias para el desarrollo de los profesionales [IEEE-CS, 2017b].

Generalmente, los cuerpos de conocimiento son definidos por las asociaciones profesionales relevantes en el área. En la web de la IEEE-CS, se encuentran disponibles tres cuerpos de conocimiento para tres disciplinas de la computación: el SWEBOK, el EITBOK y el SEBOK [IEEE-CS, 2017b]. El SWEBOK describe el conocimiento generalmente aceptado para la Ingeniería de Software. El EITBOK (*Guide to the Enterprise Information Technology Body of Knowledge*) resume y describe a alto nivel las áreas que son generalmente requeridas para brindar servicios profesionales de Tecnología de la Información en forma exitosa. El SEBOK (*Guide to the Systems Engineering Body of Knowledge*), por su parte, analiza qué se incluye en la Ingeniería de Sistemas, cómo se puede estructurar este conocimiento para facilitar la comprensión y qué documentos son más importantes para la disciplina [SERC, INCOSE, IEEE-CS, 2017].

En las descripciones anteriores se observan los distintos enfoques que pueden tener los cuerpos de conocimiento, manteniendo el objetivo de proveer de manera organizada el conocimiento relativo a su disciplina. Por su naturaleza, los cuerpos de conocimiento son utilizados como base para la construcción de guías curriculares.

En conclusión, la existencia de un cuerpo de conocimiento ayuda al desarrollo de la profesión, dándole madurez a la disciplina. A continuación, se presenta un extracto del Capítulo V: La formación en Ingeniería de Software del Libro Blanco [Camilloni and Vallespir, 2013] que afirma la importancia de contar con cuerpos de conocimiento para el desarrollo de una profesión desde el punto de vista de la educación.

*“(...) una disciplina madura debe contar con un cuerpo de conocimiento, porque sin él es difícil consensuar la formación, otorgar licencias, crear certificaciones y acreditar planes de estudio, que garanticen la adecuada formación de profesionales competentes.”*

### 2.1.1 SWEBOK

El SWEBOK es la guía para el cuerpo de conocimiento de la Ingeniería de Software. Es un proyecto que comienza en 1997 cuando el *Software Engineering Coordinating Committee* comienza a definir el conocimiento generalmente aceptado de la Ingeniería de Software [IEEE-CS, a]. En el 2001 se publica la versión de prueba del SWEBOK y en el 2004 la IEEE-CS aprueba la publicación formal

de la primera versión. La versión actual es la versión 3 y fue publicada en el 2014.

Entre los objetivos del SWEBOK se encuentran: [IEEE-CS, b]

- Brindar una caracterización del cuerpo de conocimiento de la Ingeniería de Software.
- Proveer el acceso por tópicos al cuerpo de conocimiento de la Ingeniería de Software.
- Promover una visión consistente de la Ingeniería de Software en el mundo.
- Clarificar qué lugar ocupa la Ingeniería de Software con respecto a otras disciplinas como ciencias de la computación, gestión de proyectos, ingeniería en computación y matemáticas.
- Proveer las bases para el desarrollo de currículos, certificaciones individuales y material para el licenciamiento.

La guía apunta a ser usada por distintas entidades. Se espera que sea usada en organizaciones públicas y privadas para promover una visión consistente de la Ingeniería de Software en situaciones como capacitaciones, clasificación del trabajo y definición de políticas de evaluación. A su vez, busca ser usada en la práctica por ingenieros de software, creadores de políticas, licencias y guías profesionales, asociaciones profesionales participantes en la definición de guías curriculares, acreditaciones y certificaciones, así como por estudiantes y educadores de la disciplina.

El propósito del SWEBOK es brindar una visión consensuada de cuáles son los límites de la Ingeniería de Software, así como proveer el acceso por tópicos al cuerpo de conocimiento. La guía organiza las áreas de conocimiento de forma jerárquica. Cada área es dividida en unidades y luego, cada unidad se subdivide en tópicos.

Cada área ocupa un capítulo del SWEBOK, donde se sumarizan los conceptos básicos y se brindan referencias a otras fuentes de información más detalladas. En los cuadros 1 y 2, se listan las áreas de conocimiento para la primera y tercera versión del SWEBOK respectivamente.

Áreas de conocimiento SWEBOK 2004
Requisitos de Software
Diseño de Software
Construcción de Software
Pruebas de Software
Mantenimiento de Software
Gestión de la configuración de Software
Gestión de la Ingeniería de Software
Procesos de la Ingeniería de Software
Herramientas y métodos de la Ingeniería de Software
Calidad de Software

Cuadro 1: Áreas de conocimiento SWEBOK 2004

Áreas de conocimiento SWEBOK v3
Requisitos de Software
Diseño de Software
Construcción de Software
Pruebas de Software
Mantenimiento de Software
Gestión de la configuración de Software
Gestión de la Ingeniería de Software
Procesos de la Ingeniería de Software
Modelos y métodos de la Ingeniería de Software
Calidad de Software
Práctica profesional de la Ingeniería de Software
Economía de la Ingeniería de Software
Fundamentos de computación
Fundamentos matemáticos
Fundamentos de ingeniería

Cuadro 2: Áreas de conocimiento SWEBOK v3

## 2.2 GUÍAS CURRICULARES

Desde la década del '60, la *Association for Computing Machinery (ACM)*, en conjunto con otras sociedades profesionales como la *IEEE-CS*, llevan a cabo el proyecto *Computing Curricula*. Dicho proyecto se enfoca en generar guías curriculares que sirvan como referencia para las distintas carreras del área de computación [ACM, 2017]. Debido a que el campo de la informática sigue evolucionando y van surgiendo nuevas disciplinas, el proyecto *Computing Curricula* acompaña esta evolución a través de la creación y actualización de las guías.

Las guías curriculares generadas en el proyecto *Computing Curricula* definen recomendaciones hacia las universidades de lo que debería ser incluido en los planes de estudio, considerando las necesidades y opiniones de la academia, la industria y las sociedades profesionales del área.

Entre las recomendaciones que plantean las guías se encuentran los conocimientos que debe tener un estudiante que ingresa a la carrera. Además se

describen las capacidades que deberían tener los estudiantes cuando se gradúan de un programa que cumple con las recomendaciones de la guía. Estos son los resultados esperados al egreso, y reflejan los conocimientos y habilidades que se consideran necesarios para desempeñarse de manera exitosa en la práctica profesional. Finalmente, las guías curriculares definen un cuerpo de conocimiento central que debería ser incluido como mínimo en sus respectivas carreras.

A la fecha, se han creado guías para cinco disciplinas del área de la informática. Estas son: Ingeniería en Computación, Ciencias de la Computación, Sistemas de Información, Tecnología de la Información e Ingeniería de Software.

La IEEE-CS define la Ingeniería de Software como la disciplina dedicada al diseño y desarrollo sistemático de productos de software y a la gestión del proceso de software. Como su principal objetivo, se encuentra la producción de programas dentro del plazo y presupuesto estipulado [IEEE-CS, 1980].

Para esta disciplina se creó una guía curricular enfocada a carreras de grado y otra a carreras de posgrado. En lo que refiere al nivel de grado, se definió la guía *Software Engineering 2004* (SE2004). La misma fue actualizada en el 2014 denominándose *Software Engineering 2014* (SE2014). El SE2004 y el SE2014 definen un cuerpo de conocimiento central de estructura jerárquica que recibe el nombre *Software Engineering Education Knowledge* (SEEK).

Para los programas de posgrado se define la guía GSwE2009 [SIT, 2009]. El GSwE2009 también define un cuerpo de conocimiento central de estructura jerárquica, el CBOK. Tanto el SEEK como el CBOK están basados en el SWEBOK.

### 2.3 GSwE2009

La guía GSwE2009 comenzó su desarrollo en el 2007 en el marco del proyecto *Integrated Software and Systems Engineering Curriculum* (iSEEC) [SIT, 2009]. Esta guía curricular se propone como referencia para programas de posgrados en Ingeniería de Software. Luego, en el 2009 la IEEE-CS y la ACM adoptaron el GSwE2009 como parte de sus recomendaciones para la creación de planes de estudio de posgrado en Ingeniería de Software.

El GSwE2009 se basa en el cuerpo de conocimiento para la Ingeniería de Software SWEBOK [IEEE-CS, 2004b], y en la guía SE2004 [IEEE-CS, 2004a] para carreras de grado. El GSwE2009 en su capítulo "*Guidance for the Construction and Maintenance of GSwE2009*" explica que se influencia en los principios del SE2004, distinguiéndose principalmente en definir un mayor nivel de educación esperado, y haciendo foco en la relación que tiene la Ingeniería de Software con otras disciplinas.

El GSwE2009 se estructura en contenido preparatorio, contenido central, contenido específico de la Universidad, contenido electivo y una experiencia final. El contenido preparatorio es lo que debe ser dominado por el estudiante al comenzar la maestría. Cada programa se encarga de definir cómo nivelar a los estudiantes que no cuenten con estos conocimientos. El contenido central son las habilidades y conocimientos fundamentales identificados por el GSwE2009 que un graduado de una maestría en Ingeniería de Software debería tener. El contenido específico, es el contenido que le permite a la universidad adaptar el plan de estudios para cumplir con los objetivos esperados al egreso. Una universidad podría agregar cursos obligatorios sobre contenidos que están por fuera del GSwE2009, lo que se considera como contenido específico de la universidad. Luego se tiene el contenido electivo, que es el que permite al estudiante enfocarse en intereses particulares dentro del programa. Por último, se cuenta con una experiencia final, que puede ser un proyecto, práctica o tesis.

### 2.3.1 *Core Body of Knowledge*

El CBOK es el cuerpo de conocimiento central definido en el GSwE2009 y brinda una descripción de los principales conocimientos y habilidades que se esperan de un estudiante al egresar de una carrera de posgrado en Ingeniería de Software.

El CBOK posee una estructura jerárquica de tres niveles: áreas de conocimiento, unidades y tópicos, siendo las áreas el nivel de mayor jerarquía y los tópicos los ítems más específicos. Se mantiene una jerarquía estricta, lo que significa que cada unidad pertenece a una única área de conocimiento, al igual que cada tópico pertenece a una única unidad. A continuación se describen las áreas de conocimiento del CBOK.

**ÉTICA Y CONDUCTA PROFESIONAL** Se enfoca en la conducta del punto de vista ético y profesional que debe tener un Ingeniero de Software, considerando que los productos que produce pueden afectar a diversos ámbitos de la sociedad como ser salud, finanzas, educación, entre otros.

**INGENIERÍA DE SISTEMAS** Es la disciplina que engloba las actividades relacionadas a los sistemas, siendo éstos un conjunto de componentes interconectados que existen en un entorno e interaccionan con él. Por lo general, los sistemas tienen un componente de software por lo que el Ingeniero de Software es parte de los equipos de Ingeniería en Sistemas.

**INGENIERÍA DE REQUISITOS** Es el área encargada de la obtención, análisis, especificación y validación de los requerimientos del sistema y del software.

**DISEÑO DE SOFTWARE** Refiere a las actividades en las que se analizan los requerimientos para definir la estructura arquitectónica de la solución. Esto es, definir sus componentes e interfaces para servir como base a la construcción.

**CONSTRUCCIÓN DE SOFTWARE** Abarca las actividades llevadas a cabo para la realización de software: codificación, verificación, pruebas unitarias, pruebas de integración y *debugging*.

**TESTING** Involucra las actividades que prueban el correcto funcionamiento de un producto de software evaluando y mejorando su calidad. Esto implica la identificación de defectos a través de la selección y ejecución de casos de prueba.

**MANTENIMIENTO DE SOFTWARE** Refiere al ciclo de vida del software, luego de que está en funcionamiento. Involucra la gestión de los cambios requeridos, ya sean correcciones de defectos, adaptaciones de entorno o nuevas funcionalidades.

**GESTIÓN DE LA CONFIGURACIÓN** Es la disciplina cuyo objetivo es mantener la integridad y trazabilidad del sistema en todo su ciclo de vida.

**GESTIÓN DE LA INGENIERÍA DE SOFTWARE** Engloba todas las actividades relacionadas a la gestión como ser: planificación, coordinación, medición y monitoreo, que aseguran un desarrollo y mantenimiento sistemático, disciplinado y cuantificado.

**PROCESOS DE LA INGENIERÍA DE SOFTWARE** Área que se centra en el meta-nivel de lo que refiere a los procesos de software. Esto es, la definición, ejecución, evaluación, medición, gestión, cambio y mejora del ciclo de vida del proceso de software.

**CALIDAD DEL SOFTWARE** Área que incluye las actividades de gestión de la calidad, buscando asegurar que el producto de software cumpla con los requerimientos.

## 2.4 EVALUACIÓN DE CARRERAS

La evaluación de carreras es la recopilación, revisión y utilización de los datos para el mejoramiento continuo de los programas educativos. El objetivo de evaluar una carrera es enfocarse en mejorar el aprendizaje y desarrollo de los estudiantes.

En el año 1997, la organización dedicada a la acreditación de programas curriculares *Accreditation Board for Engineering and Technology* (ABET) <sup>2</sup> creó el *Engineering Criteria 2000* (EC2000). El EC2000 es un criterio de acreditación para carreras de ingeniería que propone evaluar los resultados esperados al egreso, fijándose en lo que los estudiantes aprenden en los cursos, y no en lo que es enseñado.

Hoy en día, como repercusión del EC2000, es común que los programas evalúen si cumplen con los objetivos educativos y los resultados de aprendizaje. ABET define estos dos conceptos de la siguiente manera: los objetivos educativos son amplias proposiciones que describen lo que espera la institución que los graduados obtengan pocos años después de graduarse. Los resultados de aprendizaje describen los conocimientos, habilidades y comportamientos que los alumnos deben tener incorporados en el momento de la graduación.

Existen distintos métodos para llevar a cabo la evaluación de una carrera. Los métodos se pueden catalogar en directos o indirectos. Los métodos directos se enfocan en evaluar el trabajo de los estudiantes. Es decir, consideran el desempeño en pruebas, proyectos u otros medios que permitan medir el logro de los resultados esperados de aprendizaje. Los métodos indirectos se concentran en tomar indicadores de aprendizaje. Para este caso se infieren las capacidades desarrolladas y los conocimientos adquiridos de los estudiantes a través de encuestas, entrevistas o estadísticas. Este tipo de métodos se basa en impresiones y opiniones más que en pruebas concretas como el caso de los métodos directos.

Para evaluar los resultados esperados de aprendizaje y el cumplimiento de los objetivos educativos del programa, se pueden utilizar distintos tipos de evaluaciones que aplican a diferentes momentos del ciclo de aprendizaje. Es decir, hay métodos que realizan la evaluación durante la realización de la carrera, así como otros métodos se deben aplicar unos años después de egresar.

Un ejemplo de un método de evaluación de resultados de aprendizaje es la evaluación basada en cursos. Ésta consiste en identificar los cursos que mejor se relacionen con los resultados esperados, y luego determinar trabajos que

---

<sup>2</sup> <http://www.abet.org/about-abet/>

sean realizados por los estudiantes para estos cursos, y sirvan para evaluar el programa.

Otro ejemplo de evaluación de resultados de aprendizaje son las encuestas a egresados, que buscan obtener su punto de vista en cuanto a si el programa cubre las capacidades, competencias y resultados esperados. Los profesores también pueden ser encuestados, y de este modo evaluar el rendimiento de los estudiantes.

Por otra parte, la evaluación de los objetivos educativos también puede hacerse a través de encuestas. Un ejemplo es la realización de encuestas a exalumnos que se graduaron en los últimos cinco años, para relevar su percepción sobre el nivel de preparación que les dio el programa para su situación actual. Otro ejemplo de encuestas para hacer la misma evaluación son las encuestas a los empleadores, pero suelen ser difíciles de llevar a cabo.

## 2.5 CUBRIMIENTO TEMÁTICO

Desde la creación del criterio EC2000 se ha dedicado esfuerzo en evaluar los resultados esperados de aprendizaje y objetivos educativos de los planes de estudio. De esta forma, se evalúa lo que los alumnos aprenden en lugar de lo que es enseñado, lo que da flexibilidad a los profesores en la forma de plantear los cursos. No obstante, es de interés conocer qué temas se están impartiendo realmente.

Es importante definir los temas que se desean impartir en una carrera, además de tener indicadores sobre los niveles de conocimiento o habilidad que se están adquiriendo en la enseñanza de los mismos.

Las guías de conocimiento SE2004 y GSwE2009 definen un cuerpo de conocimiento central que indican lo que debería ser abarcado como mínimo. En el caso del SE2004 se define para cada Unidad del SEEK el tiempo mínimo en horas que debería ser dedicado para lograr la profundidad deseada. En el caso del GSwE2009, el dominio del CBOOK se define como un resultado esperado al egreso y se brindan recomendaciones en tiempo para sus distintas áreas.

Contar con información sobre los temas que se imparten en una carrera permite hacer una conexión entre los resultados de aprendizaje y lo que realmente sucede en los cursos. Conocer qué se cubre temáticamente en una carrera puede verse como una lista de temas del cuerpo de conocimiento donde la medida de cubrimiento es binaria: se cubre o no se cubre.

Por otra parte, para evaluar cómo se cubre temáticamente una carrera se debe evaluar el nivel de conocimiento de cada tema. Una forma de medir el nivel de conocimiento, es utilizando la taxonomía de Bloom.

La taxonomía de Bloom es una herramienta para establecer los objetivos de aprendizaje. Luego de realizar un proceso de aprendizaje, el estudiante debe haber adquirido nuevas habilidades y conocimientos. Benjamín Bloom identifica tres categorías del saber. La primera es el campo cognoscitivo, que abarca las actividades de conocimiento, comprensión, aplicación, análisis, síntesis y evaluación. Luego identifica el campo psicomotriz, que refiere a las destrezas y hábitos adquiridos. Por último, Bloom identifica el campo afectivo, que toma en cuenta el grado de interiorización que el individuo revela en su conducta. [Bloom et al., 1956].

Para obtener el nivel de conocimiento alcanzado, una opción difícil de llevar a cabo sería evaluar al estudiante luego de finalizar la carrera considerando cada tema del cuerpo de conocimiento con la dificultad esperada. Otra opción sería dejar de lado los conocimientos y realizar una evaluación de habilidades del estudiante al egreso. En dicho caso, cada docente de cada curso sería responsable de evaluar si se está cubriendo o no el cuerpo de conocimiento. La forma de evaluar el nivel de conocimiento alcanzado también debe ser cuidadosamente definida. Distintas técnicas son sugeridas según el nivel de la taxonomía que se desee evaluar. Para evaluar el nivel de “Conocimiento” de la taxonomía es recomendable el uso de preguntas múltiple opción, mientras que para el nivel de “Comprensión” corresponde utilizar preguntas abiertas de desarrollo o presentaciones. Conocer el nivel de Bloom alcanzado por el estudiante para los distintos temas del cuerpo de conocimiento requiere un gran esfuerzo de gestión en torno a la evaluación.

Otra alternativa, más sencilla de llevar a cabo para evaluar el cubrimiento temático, es considerar las horas reales dedicadas a cada uno de los temas del cuerpo de conocimiento. De esta forma, se puede tener una idea de qué conocimientos se enseñan, y cuánto tiempo se le dedica a los mismos. No obstante, la información resultante debería cruzarse con los resultados de las pruebas a los estudiantes, para corroborar que realmente se están logrando los niveles de conocimiento esperados. En este sentido, Camilloni define un método para evaluar el cubrimiento temático de una carrera con respecto a un cuerpo de conocimiento que se tome como referencia [Camilloni, 2015].

El método propuesto no utiliza el estimado de horas que define el programa de cada curso, sino que se basa en obtener la información precisa de las horas reales que los alumnos dedican a cada curso. De esta forma, el método sirve para evaluar el cubrimiento temático de un conjunto de ediciones de cursos que cumplen con el plan de estudios, respecto a un cuerpo de conocimiento de referencia. Esto es, evaluar los conocimientos del cuerpo que son abarcados por un alumno que cumple con el plan de estudio.

El método definido es genérico para evaluar el cubrimiento temático de la ejecución de la implementación de un plan de estudios con respecto a cualquier cuerpo de conocimiento de estructura jerárquica y consta de cuatro etapas:

- Registrar horas dedicadas a cada uno de los temas impartidos en los cursos que se dictan.
- Asignar las correspondencias entre los temas impartidos en los cursos y los elementos del cuerpo de conocimiento. Para cada tema impartido en el curso se reparte su tiempo de dedicación entre los elementos del cuerpo con los que se corresponde.
- Revisión de la asignación con el docente responsable. El docente valida que la asignación entre los temas del curso y los elementos del cuerpo de conocimiento sea correcta, y que la distribución del tiempo sea adecuada con la realidad.
- Procesamiento de los datos para realizar la evaluación de cubrimiento temático.

Para llevar a cabo el método, se definen un conjunto de planillas formateadas e instructivos. Específicamente, se define una planilla de cálculo denominada planilla consolidada para asignación de horas a tópicos, con lógica agregada, que sirve como herramienta para obtener de forma automática distintos tipos de información acerca del cubrimiento temático. Esta planilla es una hoja de cálculo electrónica que se rellena con la información consolidada del cubrimiento de tópicos para todas las ediciones de los cursos dictados en una carrera. Al rellenar la planilla, se realizan los cálculos a través de la lógica agregada en el pre-formateo, y se visualiza la siguiente información:

- El cubrimiento temático obtenido para cada área de conocimiento, unidad y tópico del CBOOK, considerando horas de contacto y horas totales.
- Qué áreas de conocimiento son tratadas en cada edición de curso.
- El esfuerzo temático alcanzado. Esto es, la comparación en horas entre el tiempo dedicado a cada ítem del cuerpo con respecto al esperado en la guía GSwE2009.
- El balance temático por área de conocimiento. Esto es la distribución en porcentajes del tiempo dedicado entre las distintas áreas del cuerpo, y su comparación con lo sugerido en el GSwE2009.

## ANÁLISIS DE LA REALIDAD

---

En la sección 2.5 del marco teórico, se introdujo el método definido para la evaluación del cubrimiento temático. En el presente capítulo, se expone el trabajo que se ha estado realizando al aplicar este método para evaluar una carrera de posgrado en UDELAR.

La Facultad de Ingeniería de la UDELAR construyó en el año 2011 un Plan de estudios para una Especialización en Ingeniería de Software (EIS) y otro para una Maestría en Ingeniería de Software (MIS) basados en la guía curricular GSwE2009. Ambas carreras comenzaron a dictarse en el 2012 y desde su comienzo han sido evaluadas, con el objetivo de mejorarlas continuamente. La evaluación de la EIS se realizó a través del método de cubrimiento temático para evaluarla con respecto al cuerpo de conocimiento de la GSwE2009, es decir, el CBOOK.

Como se explica anteriormente, el método propone llevar a cabo el registro de tiempo dedicado por los alumnos a los temas impartidos, y a través de la correspondencia entre temas y tópicos, obtener el tiempo dedicado a cada elemento del CBOOK. Luego, con la información de tiempo dedicado a los tópicos, se evalúan distintas consultas de interés: el esfuerzo temático, el balance temático y el cubrimiento temático.

Como primera etapa de la evaluación, se seleccionan qué cursos se van a evaluar. Para el caso de la generación 2012-2013 se seleccionan los cursos obligatorios, y el conjunto de cursos electivos más relevantes para la Ingeniería de Software, aunque este conjunto no cumple con los créditos exigidos por el programa de la EIS. Los cursos seleccionados son:

- Prácticas de Desarrollo de IS - Requisitos
- Prácticas de Desarrollo de IS - Diseño de Software
- Prácticas de Desarrollo de IS - Construcción
- Prácticas de Desarrollo de IS - Testing
- Arquitectura de Software
- Costos para Ingeniería de Software
- Gestión de la Configuración

- Modelado y Simulación de Procesos de Negocio
- Introducción al CMMI-DEV
- Inspección de Software: El proceso de Inspección
- Estimación de Proyectos de Desarrollo y Mantenimiento de SW
- Testing en el proceso de desarrollo de software
- Mantenimiento de Software
- Construcción de Software
- Gestión de Proyectos de Software
- Introducción al CMMI-ACQ

Luego, para cada uno de estos cursos, se selecciona un alumno responsable de registrar el tiempo que le lleva realizar el curso. El alumno registra para cada tema dictado, el tiempo dedicado al mismo, discriminado entre los distintos tipos de hora:

- Horas de teórico presencial
- Horas de práctico presencial
- Horas de práctico no presencial
- Horas de laboratorio presencial
- Horas de laboratorio no presencial
- Horas de evaluación
- Horas de estudio

Las horas no presenciales de práctico y laboratorio registradas por el alumno, se consideran una estimación de la dedicación real promedio del resto de los estudiantes. Sin embargo, para el caso de las horas de estudio, se utiliza una estimación, considerando que un estudiante dedica una hora de estudio por cada hora de teórico presencial.

Para realizar la tarea de registro de tiempo, se utiliza el “Instructivo para registro de dedicación por tema de un curso”. En el instructivo se presentan un conjunto de guías que indican cómo completar la “Planilla para la asignación

de temas y tópicos de un curso”, la cual debe contener la información general del curso y la información asociada a las dedicaciones.

Con respecto al curso interesa conocer el nombre del curso, el docente responsable, la edición (año), los créditos, y la dedicación estimada en base a los créditos. Para el caso de las dedicaciones, interesa conocer su fecha, una descripción del tema dictado, el subtema, la fuente, la cantidad de horas, y el tipo de hora.

Luego de contar con el registro de tiempos por curso, interesa llevar a cabo la correspondencia entre los temas registrados con los tópicos del CBOK, asignándole a éstos el tiempo que les corresponda. Para esta actividad se cuenta con un instructivo llamado “Instructivo para asignación de horas por tópico en base a temas”. El instructivo guía cómo debe hacerse la asignación de correspondencias entre temas y tópicos del cuerpo de referencia. Entre las guías que plantea el instructivo se encuentran:

- Cómo se debe realizar la asignación cuando las horas se corresponden a más de un tópico de referencia.
- Cómo se debe realizar la asignación cuando un tema no se corresponde con ningún tópico del cuerpo de conocimiento, pero sí con algún ítem de nivel superior.
- Cómo registrar las horas que no se corresponden con ningún ítem del cuerpo de conocimiento de referencia.
- Cómo realizar la asignación para las horas de evaluación.

Una vez realizada la asignación de temas a tópicos, se utiliza la última hoja de la planilla de asignación de temas y tópicos de un curso para registrar la consolidación de las horas de cada tipo dedicadas a cada tópico del CBOK.

Otro caso contemplado en el proceso son los cursos bajo gestión de la configuración. Estos refieren a ediciones de cursos que presentan cambios menores con respecto a una edición anterior. Para estos casos, se define la “Planilla para registro de modificaciones temáticas de un curso”, en la que se registran únicamente las modificaciones en cuanto a los temas impartidos con respecto a la edición anterior del curso. Las modificaciones que sufren los temas se discriminan en nuevos, modificados o eliminados y para cada tipo de modificación interesa registrar distintos datos. En este sentido, el proceso define un instructivo que facilita la re-utilización del registro de temas y guía cómo asignar las horas por tópico en base a las modificaciones.

Al contar con el tiempo dedicado a los temas, se puede conocer cómo se cubre temáticamente el CBOK. Esto es, conocer los temas cubiertos, y las horas

dedicadas a cada uno. Para obtener esta información, se utiliza la “Planilla consolidada para asignación de horas a tópicos” mencionada previamente en la sección 2.5 del marco teórico. La planilla sirve como herramienta para obtener la evaluación de los distintos tipos de consulta, a partir de los datos registrados para cada curso.

La planilla consolidada se adapta para contemplar un cuerpo de conocimiento de tres niveles, y así soportar el caso específico de evaluar la EIS con respecto al CBOOK. En este sentido, la planilla utilizada consta de siete hojas. La primera de ellas, denominada CBOOK, se completa de forma manual con la información sumariada de tiempos por tópico, proveniente del registro para las distintas ediciones de los cursos.

A partir de la información cargada en la primera hoja, la planilla genera de forma automática la información de cubrimiento temático alcanzado a nivel de área de conocimiento, unidad y tópico para un conjunto de cursos seleccionados. A su vez, el análisis se puede realizar considerando distintos tipos de hora. Para la EIS se realiza el análisis de cubrimiento considerando las horas de contacto y las horas totales.

### 3.1 APLICACIÓN DE LA EVALUACIÓN DE CUBRIMIENTO TEMÁTICO PARA LA EIS

A continuación se presentan los distintos tipos de consultas que se obtienen al evaluar el cubrimiento para la primera generación de estudiantes de la EIS que comenzaron sus cursos en el 2012 y los culminaron en el 2013. Para esta evaluación se consideraron las ediciones de cursos más relevantes, ya que al tener un conjunto de cursos electivos no todos los estudiantes realizan los mismos cursos.

#### 3.1.1 *Cubrimiento temático*

La consulta de cubrimiento temático busca determinar para cada tópico, unidad y área del CBOOK si está cubierto o no. Para ello se define el siguiente criterio:

- Un tópico se considera cubierto si para alguna edición de curso se le han asignado horas de al menos una dedicación. Si un tópico está cubierto se lo identifica en color verde. Por el contrario, si no se tienen horas registradas para el tópico, en ninguna de las ediciones seleccionadas, se utiliza el color anaranjado para indicar que no está cubierto.

- Para determinar el cubrimiento a nivel de unidad se definen franjas de cubrimiento, en las que se ubican las unidades según el porcentaje de tópicos hijos que tengan cubiertos. Entre un 0 y un 15 % de hijos cubiertos, se aplica color anaranjado indicando que la unidad no está cubierta, entre 15 % y 50 % color amarillo indicando cobertura parcial, entre 50 % y 85 % verde claro indicando que la unidad está ampliamente cubierta, y entre 85 % y 100 % verde, lo cual indica cobertura total.
- Para el caso de las áreas de conocimiento, se definen las mismas franjas de cubrimiento que para el nivel de las unidades, y cada área se cataloga en una de las franjas según el porcentaje de unidades hijas que tengan al menos parcialmente cubiertas. Es decir, el porcentaje de unidades que contengan más de un 15 % de sus tópicos cubiertos.

En las figuras 1 y 2 se muestra el resultado de evaluar el cubrimiento temático para dos áreas del CBOK, Testing y Ética y conducta profesional.

F - Testing	
1 - Fundamentos del testing	1.1 - Testing de sistema y testing de software 1.2 - Terminología relacionada al testing 1.3 - Cuestiones claves 1.4 - Relaciones entre testing y otras actividades
2 - Niveles de pruebas	2.1 - El blanco del testing 2.2 - Los objetivos del testing 2.3 - Pruebas de componentes 2.4 - Pruebas de integración 2.5 - Pruebas de sistema 2.6 - Pruebas de aceptación
3 - Técnicas del testing	3.1 - Basadas en la intuición y experiencia del verificador 3.2 - Basadas en la especificación 3.3 - Basadas en el código 3.4 - Basadas en las fallas 3.5 - Basadas en el uso 3.6 - Basadas en la naturaleza de la aplicación 3.7 - Selección y combinación de técnicas
4 - Mediciones relacionadas a las pruebas	4.1 - Evaluación del programa o sistema bajo prueba 4.2 - Evaluación de las pruebas realizadas
5 - Proceso de Prueba	5.1 - Referente a la gestión 5.2 - Actividades de pruebas

Figura 1: Cubrimiento temático para área Testing.

En el caso del área Testing, se cuenta con cinco unidades, de las cuales cuatro están totalmente cubiertas ya que el 100 % de sus tópicos hijos están cubiertos, y la unidad Fundamentos del testing se encuentra ampliamente cubierta ya

A - Ética y Conducta Profesional	
1 - Cuestiones sociales, legales e históricas	1.1 - Confidencialidad de datos y seguridad, vigilancia y privacidad 1.2 - Cuestiones de desarrollo histórico, sexo, minorías y culturas 1.3 - Contratos de responsabilidad, propiedad intelectual y libertad de la información 1.4 - Delito informático y la aplicación de la ley
2 - Códigos de ética y conducta profesional	2.1 - Responsabilidad frente a la sociedad 2.2 - Modelos para el profesionalismo y Sociedades Profesionales 2.3 - Código de ética y práctica
3 - La naturaleza y el rol de los estándares de la ingeniería de software	3.1 - Naturaleza y rol de los estándares 3.2 - Estándares internacionales, estándares y organizaciones de armonización 3.3 - Cuerpos de conocimiento, prácticas aceptadas y mejoras prácticas

Figura 2: Cubrimiento temático para área Ética y conducta profesional.

que de cuatro tópicos hijos, uno no está cubierto, lo que genera que la unidad tenga un 75 % de cobertura, quedando ampliamente cubierta. Luego, a nivel de área, se considera que todas las unidades están cubiertas ya que están al menos parcialmente cubiertas, por lo que el área Testing está ampliamente cubierto según el análisis de cubrimiento y los criterios fijados para la EIS.

Para el caso del área Ética y Conducta Profesional, se observa que una unidad está ampliamente cubierta, ya que el 66,6 % de sus tópicos están cubiertos, y las otras dos unidades están no cubiertas. Esto implica que a nivel de área, Ética y Conducta Profesional cuenta con un 33,3 % de sus unidades cubiertas, quedando parcialmente cubierta.

Por otro lado, es de interés visualizar el cubrimiento considerando las unidades. Esto se presenta en las figuras 3 y 4. En la primera se muestra mediante gráficos de torta las franjas de cubrimiento alcanzadas por las unidades de cada área de conocimiento. En el segundo caso, se realiza una sumarización de lo anterior, y se presentan las franjas alcanzadas por todas las unidades del CBOK, permitiendo visualizar el porcentaje de unidades totalmente cubiertas, ampliamente cubiertas, parcialmente cubiertas y no cubiertas.

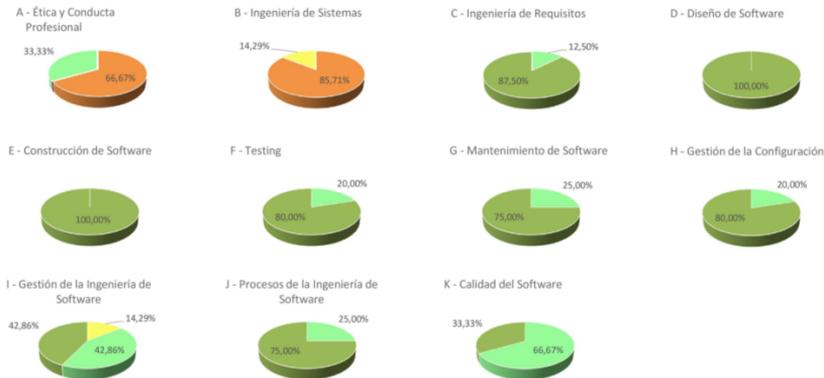


Figura 3: Cubrimiento temático a nivel de unidades para cada área del CBOK.

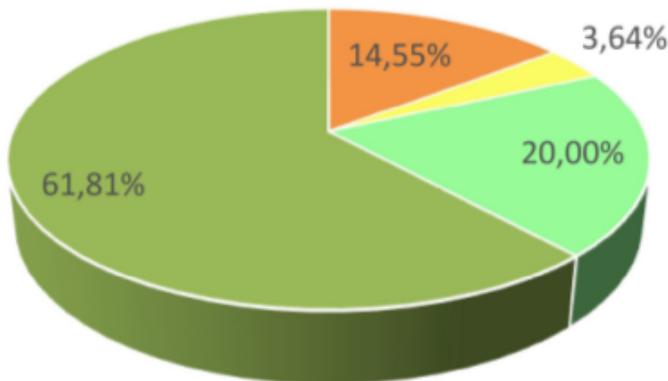


Figura 4: Cubrimiento temático considerando todas las unidades del CBOK.

### 3.1.2 Esfuerzo temático

El esfuerzo temático permite visualizar cuánto se le ha dedicado a cada ítem del CBOK. Se compara la dedicación considerando las horas totales y las horas de contacto, contra las sugeridas por el GSwE2009.

Para el CBOK, se definen porcentajes esperados de dedicación a nivel de áreas de conocimiento presentadas en la tabla 3, y se estipula un total de 200 horas de contacto para impartir el CBOK, más dos o tres veces esta dedicación

de estudio individual, a modo de adquirir la profundidad necesaria. Esto deriva en una dedicación total de entre 600 y 800 horas.

KA	% Mínimo	% Máximo
Ética y conducta profesional (ECP)	1 %	2 %
Ingeniería de sistemas (ISis)	2 %	3 %
Ingeniería de requisitos (IR)	6 %	8 %
Diseño de software (DS)	9 %	11 %
Construcción de software (CS)	1 %	3 %
Pruebas (P)	4 %	6 %
Mantenimiento de software (MS)	3 %	4 %
Gestión de la configuración (GC)	2 %	3 %
Gestión de la ingeniería de software (GIS)	7 %	9 %
Proceso de la ingeniería de software (PIS)	3 %	4 %
Calidad del software (CalS)	3 %	4 %
Contenido electivo o específico de la Universidad	50 %	50 %

Cuadro 3: Porcentajes sugeridos para las áreas del CBOK

Aunque el análisis de esfuerzo podría hacerse también a nivel de unidad o tópico, para el CBOK no se cuenta con parámetros que definan una dedicación esperada para otro nivel de no ser para las áreas de conocimiento.

Con los valores anteriores se calculan las horas esperadas máximas y mínimas para cada elemento. El mínimo por área, se calcula como el resultado de aplicar el porcentaje máximo del área definido por el CBOK, a la dedicación total esperada mínima para adquirir el CBOK (600 horas), mientras que la dedicación máxima esperada para el área se calcula como el resultado de aplicar el porcentaje máximo también, pero aplicado al total de horas máximas esperadas para adquirir el CBOK (800 horas). En la tabla 4 se puede observar el mínimo y máximo en horas que se espera sea dedicado a cada área.

Luego, se comparan los mínimos y máximos calculados contra el total de horas registradas para cada área del CBOK. El resultado de la comparación se representa utilizando el color rojo cuando la dedicación registrada en la EIS está por debajo de la dedicación sugerida para 600 horas totales, el color amarillo cuando la dedicación registrada para el elemento es mayor que la esperada

KA	% por KA	600 hs (hs)	800 hs (hs)
Ética y conducta profesional (ECP)	2 %	12,0	16,0
Ingeniería de sistemas (ISis)	3 %	18,0	24,0
Ingeniería de requisitos (IR)	8 %	48,0	64,0
Diseño de software (DS)	11 %	66,0	88,0
Construcción de software (CS)	3 %	18,0	24,0
Pruebas (P)	6 %	36,0	48,0
Mantenimiento de software (MS)	4 %	24,0	32,0
Gestión de la configuración (GC)	3 %	18,0	14,0
Gestión de la ingeniería de software (GIS)	9 %	54,0	72,0
Proceso de la ingeniería de software (PIS)	4 %	24,0	32,0
Calidad del software (CalS)	4 %	24,0	32,0

Cuadro 4: Esfuerzo: Dedicación en horas esperada para las áreas del CBOK

para una dedicación total de 600 horas pero menor que para la de 800 horas, y color verde cuando la dedicación en la EIS es superior a lo esperado para 800 horas de dedicación total. En la figura 5 se puede observar el resultado de evaluar la EIS en los años 2012-2013.

KA	% por KA	Basado en 600 hs (hs)	Basado en 800 hs (hs)	EIS (hs)
ECP	2	12,0	16,0	3,0
ISis	3	18,0	24,0	1,3
IR	8	48,0	64,0	44,8
DS	11	66,0	88,0	156,8
CS	3	18,0	24,0	67,5
TST	6	36,0	48,0	108,3
MS	4	24,0	32,0	55,0
GC	3	18,0	24,0	68,5
GIS	9	54,0	72,0	228,6
PIS	4	24,0	32,0	80,8
CalS	4	24,0	32,0	82,0

Figura 5: Esfuerzo temático para la EIS 2012-2013 contra el CBOK.

### 3.1.3 Balance temático

El balance temático permite evaluar cómo se distribuye la carga horaria dedicada entre las distintas áreas del CBOK y compararlo con la distribución recomendada por el GSwE2009.

La guía define el rango en términos de porcentaje de tiempo que debe serle dedicado a cada área de conocimiento. Por ejemplo, el GSwE2009 define que para el área Calidad del Software se debe dedicar entre un 3 % y 4 % del tiempo total, mientras que para el área Ingeniería de Requisitos la dedicación debe ser entre un 6 % y un 8 %. Esto significa, que en un total de 100 horas dedicadas, entre 3 y 4 horas deben ser dedicadas a temas relacionados a la Calidad del Software, y entre 6 y 8 horas deben dedicarse a la Ingeniería de Requisitos, para así cumplir con el balance de carga horaria estipulado por el GSwE2009.

En la figura 3 se presentan los porcentajes mínimos y máximos definidos por la guía para cada área de conocimiento. También se observa que el GSwE2009 deja aproximadamente un 50 % del tiempo dedicado a cubrir contenido electivo o específico de la universidad. Esto es, el contenido que cada Universidad podrá incluir para adaptar su programa y cumplir con sus objetivos esperados al egreso.

En este sentido, para evaluar el balance temático de la EIS con respecto al CBOK se entiende que no corresponde considerar contenido específico de la universidad, ya que el plan de estudios de la EIS está basado en el GSwE2009, y la totalidad del tiempo dedicado está enfocado a cubrir el CBOK.

Para efectuar la evaluación, se extiende el porcentaje definido para cada área de conocimiento a modo de dejar cubierto el 100 % del tiempo. Debido a que el GSwE2009 otorga un 50 % al contenido específico de la universidad, es decir, aproximadamente la mitad del tiempo, la forma de extender los porcentajes esperados para las áreas de conocimiento es multiplicándolos por dos. En la tabla 5 se pueden visualizar los porcentajes resultantes.

La evaluación del balance temático de la EIS con respecto al CBOK se realiza a nivel de áreas de conocimiento ya que para los niveles restantes no se cuenta con un porcentaje definido. En las figuras 6 y 7 se muestra el porcentaje del tiempo dedicado a cada área de conocimiento, para las materias de la ejecución 2012-2013 seleccionadas, y la comparación con los mínimos y máximos antes mencionados, de forma tabular y gráfica respectivamente.

KA	% Mínimo	% Máximo
Ética y conducta profesional (ECP)	2 %	4 %
Ingeniería de sistemas (ISis)	4 %	6 %
Ingeniería de requisitos (IR)	12 %	16 %
Diseño de software (DS)	18 %	22 %
Construcción de software (CS)	2 %	6 %
Pruebas (P)	8 %	12 %
Mantenimiento de software (MS)	6 %	8 %
Gestión de la configuración (GC)	4 %	6 %
Gestión de la ingeniería de software (GIS)	14 %	18 %
Proceso de la ingeniería de software (PIS)	6 %	8 %
Calidad del software (Cals)	6 %	8 %

Cuadro 5: Balance: Porcentaje esperado para las áreas del CBOK

Área de Conocimiento	%Min	%Max	% EIS	Comparación
A - Ética y Conducta Profesional	2	4	0,0	▼
B - Ingeniería de Sistemas	4	6	0,0	▼
C - Ingeniería de Requisitos	12	16	5,0	▼
D - Diseño de Software	18	22	17,0	▼
E - Construcción de Software	2	6	8,0	▲
F - Testing	8	12	12,0	✓
G - Mantenimiento de Software	6	8	6,0	✓
H - Gestión de la Configuración	4	6	8,0	▲
I - Gestión de la Ingeniería de Software	14	18	26,0	▲
J - Procesos de la Ingeniería de Software	6	8	9,0	▲
K - Calidad del Software	6	8	9,0	▲

Figura 6: Balance temático para la EIS 2012-2013 contra el CBOK.

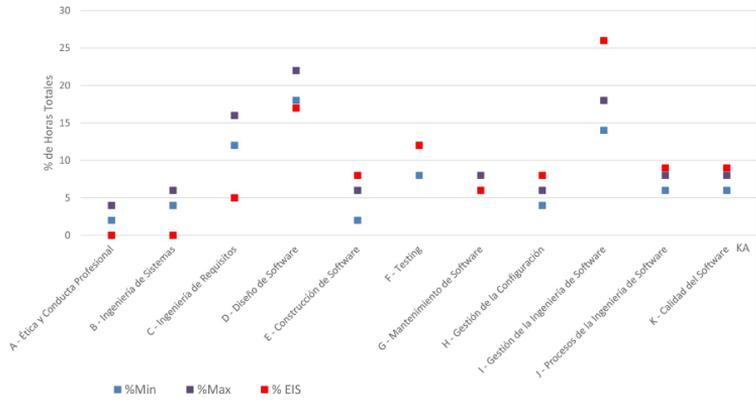


Figura 7: Balance temático para la EIS 2012-2013 contra el CBOK.

## REQUISITOS DEL SISTEMA

---

En este capítulo se presenta el sistema a construir durante este proyecto de grado. En la sección 4.1 se describe brevemente el sistema a construir. Luego, en la sección 4.2 se presentan los distintos actores del sistema. Por último, en las secciones 4.3 y 4.4 se presentan los requisitos funcionales y no funcionales del sistema respectivamente.

### 4.1 SISTEMA A CONSTRUIR

La Facultad de Ingeniería de la UDELAR desea construir un sistema de información con una base de datos relacional como repositorio de datos, que permita dar soporte a la evaluación de carreras con respecto a guías curriculares de referencia. En este sentido, el sistema brindará las herramientas necesarias para evaluar el cubrimiento temático de un plan de estudio con respecto a uno o varios cuerpos de conocimientos de referencia a partir del registro de horas reales que fueron ingresadas al sistema.

El sistema permitirá el registro de horas reales dedicadas a las ediciones de los cursos y la realización de la correspondencia de dedicaciones a elementos de los cuerpos de conocimiento de referencia.

Se considera que un curso se encuentra bajo gestión de la configuración para un cuerpo de conocimiento cuando ya se tiene el registro de cubrimiento temático de una edición del curso para dicho cuerpo de conocimiento. En este caso, el sistema permitirá reutilizar el registro de horas reales dedicadas y su correspondiente asignación a los elementos del cuerpo de conocimiento. De esta forma solo se deberán registrar las modificaciones realizadas en la edición del curso actual y realizar la correspondencia para dichas modificaciones.

A su vez, el sistema será extensible a incorporar un módulo de encuestas que permita evaluar el grado de cumplimiento de los resultados esperados al egreso definidos por el plan de estudios, desde el punto de vista de los distintos participantes en el área: estudiantes, docentes, entre otros.

Con respecto al análisis de la información registrada en la herramienta, es de gran interés que se pueda visualizar dicha información de forma gráfica y tabular, permitiendo evaluar y proponer mejoras en la implementación de una carrera.

La EIS debe utilizarse como caso de estudio, ya que se ha evaluado desde su inicio y se cuenta con el registro de dedicaciones completo para todas sus ediciones de curso y la información resultante del análisis de cubrimiento temático. Los datos registrados de ediciones anteriores de la Especialización permiten validar el sistema y evaluar qué otras funcionalidades resultan interesantes agregar.

Además de soportar las evaluaciones presentadas en el capítulo 3 “*Análisis de la realidad*”, el sistema debe ser genérico de forma que pueda ser utilizado para la evaluación de distintas carreras con respecto a distintas guías curriculares. Esto se logra a través de parametrizar los criterios de evaluación, y permitir que estos parámetros sean configurables por el usuario. Las configuraciones se podrán guardar en forma de perfiles de configuración, que serán utilizados al momento de realizar las consultas dándole valor a los distintos parámetros. En el marco de este proyecto de grado, se restringe la realidad a considerar únicamente cuerpos de conocimiento con estructura jerárquica.

El sistema debe soportar la gestión de carreras, planes de estudio, implementaciones de planes de estudio, ejecuciones, cursos y ediciones. Cada carrera tiene asociado un código, un nombre, un responsable, y da una certificación a quienes egresan de la misma. Los planes de estudio por su parte, definen distintos aspectos que un estudiante debe cumplir para egresar de la carrera. Esto es, créditos, duración, horas totales esperadas, objetivos educativos, resultados esperados al ingreso y al egreso. A lo largo de los años, las carreras suelen actualizar sus planes de estudio, por lo que pueden existir varios planes asociados a una carrera.

Para cumplir con un plan de estudios, un estudiante debe aprobar un conjunto de cursos. Cada curso cuenta con un código y un nombre que lo identifica, y tiene asociado una cantidad de horas totales esperadas, una cantidad de créditos y un programa. Los cursos pueden estar compartidos por más de una carrera, siendo obligatorio o electivo según el plan de estudios. Cada vez que se dicta un curso se genera una nueva edición del curso. La edición de un curso cuenta con un responsable y se identifica por el año en el que se dicta.

Una implementación es una instancia del plan de estudios. Se simplifica como un conjunto de cursos, algunos de los cuales el estudiante debe realizar en forma obligatoria y otros de forma electiva para acceder a la titulación. Es decir, una implementación de un plan de estudio es el conjunto de cursos obligatorios y electivos que cumplen con el plan de estudio.

Una ejecución de la implementación de un plan de estudio es el dictado de los cursos que implementan el plan. Es decir, es el subconjunto de ediciones de cursos, que cumplen con la implementación del plan de estudio. Puede verse co-

mo las ediciones cursadas por un estudiante, que cumplen con las restricciones del plan de estudio. Eventualmente puede ser distinta a la de otro estudiante que haya elegido materias electivas distintas.

El sistema debe soportar la gestión de cuerpos de conocimiento y su relación con los planes de estudio. Cada cuerpo tiene un nombre, una descripción y se compone por elementos estructurados de forma jerárquica arborescente. Los cuerpos se pueden relacionar con uno o más planes de estudio, y para cada edición de curso perteneciente al plan se deben poder registrar las horas reales de cada tipo dedicadas a los distintos temas, y la correspondencia de estos temas con los elementos del cuerpo de conocimiento. Además, se debe poder reutilizar el registro y correspondencia para aquellas ediciones que se encuentren bajo gestión de la configuración.

Es de interés que el sistema brinde flexibilidad en la selección de ediciones que conforman la ejecución a evaluar. Se desean evaluar ejecuciones reales del plan de estudio, y a su vez, interesa evaluar posibles mejoras a través de la simulación de ejecuciones. Una ejecución real es el conjunto de las ediciones de todos los cursos obligatorios, más un subconjunto de ediciones de cursos electivos que cumplen con los requisitos del plan de estudios. Las ejecuciones reales, eventualmente pueden coincidir con el conjunto de ediciones que cursó un alumno que aprobó el plan de estudio. Una ejecución simulada consiste en una selección de ediciones de cursos que no necesariamente hayan sido dictados para la misma generación de estudiantes, o no necesariamente cumplan con las restricciones del plan de estudio.

Con el objetivo de adecuarse a la forma de trabajo actual, el sistema debe permitir registrar dedicaciones a una edición a través de la importación de hojas de cálculo con un formato dado similar al usado actualmente. Además toda la información de interés generada al hacer las consultas debe poder ser exportada a este formato.

Es deseable que el sistema colabore en mostrar cruces de información que son difíciles de visualizar con la forma manual de evaluación. En este sentido, es deseable visualizar qué ediciones no tienen completa la asignación de dedicaciones a los elementos del cuerpo de conocimiento. Esto ayuda a conocer si hay evaluaciones a ejecuciones que estén parcialmente registradas.

## 4.2 USUARIOS DEL SISTEMA

El sistema debe poder ser utilizado por distintos tipos de usuarios, que se detallan a continuación:

**SUPER ADMINISTRADOR DEL SISTEMA**

El super administrador puede acceder a todas las funcionalidades del sistema.

**ADMINISTRADOR DEL SISTEMA**

Es el principal responsable del sistema. Se encarga de crear carreras y de administrar los usuarios responsables de las mismas. A excepción de la gestión de usuarios, tiene acceso al resto de las funcionalidades del sistema.

**RESPONSABLE DE LA CARRERA**

Usuario encargado de la administración de una carrera, de ingresar la información correspondiente a la misma y de administrar los usuarios que podrán acceder a ciertas funcionalidades vinculadas a la carrera. Entre las principales funciones de este rol se encuentran: definir cuerpos de conocimiento y vincularlos con distintos planes de estudios, registrar implementaciones de planes, definir los cursos con sus ediciones correspondientes. Además, este usuario administra a los usuarios responsables de asignar cubrimiento de la carrera, los usuarios responsables de registrar horas de una edición de un curso y los usuarios que pueden realizar consultas. El responsable de la carrera también se encarga del manejo de las encuestas y de visualizar los resultados obtenidos.

**RESPONSABLE DE ASIGNAR CUBRIMIENTO**

Este usuario se asigna por carrera, y es el encargado de realizar la asignación del cubrimiento temático, es decir de mapear los temas a los elementos del o los cuerpos de conocimiento de referencia asociados a la carrera.

**RESPONSABLE DEL REGISTRO DE TIEMPO POR TEMAS**

Usuario encargado de registrar las horas reales que se dedican a los temas, dictados en una edición de un curso. Este usuario se asigna por edición de un curso.

**USUARIO DE CONSULTAS**

Este rol se asigna por carrera, y puede visualizar los resultados obtenidos de los diferentes análisis realizados para la carrera.

**4.3 REQUERIMIENTOS FUNCIONALES**

En esta sección se detallan los requerimientos funcionales del sistema.

**ADMINISTRAR USUARIOS DEL SISTEMA**

El sistema debe permitir administrar los usuarios responsables de las carreras. El responsable de una carrera tendrá la responsabilidad de crear, modificar y eliminar los usuarios responsables de asignar cubrimiento, los usuarios responsables del registro de tiempos por tema para una edición de un curso y los usuarios que pueden realizar consultas sobre la carrera.

**CREACIÓN DE UNA CARRERA**

El sistema debe permitir crear carreras definiendo su nombre, código y responsable.

**CREACIÓN DE UN PLAN DE ESTUDIO**

El sistema debe permitir especificar un plan de estudio perteneciente a una carrera indicando su nombre, código, horas totales esperadas y duración. Además se podrán definir los conocimientos esperados al ingreso, el conjunto de conocimientos esperados al egreso y los objetivos educativos del plan.

**CREACIÓN DE UN CURSO**

Se requiere poder definir nuevos cursos indicando nombre, código, cantidad de créditos, horas totales esperadas y programa.

**CREACIÓN DE UNA EDICIÓN DE UN CURSO**

Se debe permitir ingresar las distintas ediciones para los cursos. Para cada edición se indica el año, docente responsable y se asigna un responsable de registro de tiempo.

**CREACIÓN DE UNA IMPLEMENTACIÓN DE PLAN DE ESTUDIO**

Se debe poder crear una implementación sobre un plan de estudio, indicándole un nombre, código y la selección de cursos electivos y obligatorios.

**REGISTRAR EJECUCIÓN DE UNA IMPLEMENTACIÓN**

El sistema debe permitir dar de alta una nueva ejecución de una implementación de un plan. Cada ejecución corresponde a una implementación y se le asigna un nombre, una descripción, un rango de años, y un conjunto de ediciones de cursos. Las ediciones de curso deben pertenecer al rango de años de la ejecución, y corresponderse a cursos que pertenecen a la implementación.

**REGISTRO DE DEDICACIONES**

El sistema permite registrar la dedicación horaria sobre las ediciones de

cursos. Para cada dedicación se detalla su tema, la cantidad de horas dedicadas, el tipo de hora, la fecha, una descripción, y de manera opcional, la fuente utilizada y un enlace a la misma.

#### DEFINICIÓN DE UN CUERPO DE CONOCIMIENTO

Se debe poder definir un cuerpo de conocimiento, indicando su nombre y descripción. Dado que el sistema soporta solamente aquellos cuerpos de conocimiento que tengan una estructura jerárquica arborescente, se debe indicar la cantidad de niveles en los que se estructura, y el nombre de cada nivel. Luego, para cada nivel, se deben agregar los ítems que pertenecen a dicho nivel, indicando su nombre, código e ítem padre en caso de corresponder.

#### RELACIONAR UN PLAN DE ESTUDIO CON UN CUERPO DE CONOCIMIENTO

Para realizar las evaluaciones el sistema debe permitir relacionar los planes de estudio a uno o más cuerpos de conocimiento de referencia. Las evaluaciones pueden realizarse considerando las horas de un subconjunto de los tipos de hora definidos en el sistema o considerando lo que se denomina horas totales. Las horas totales son la sumatoria de todos los tipos de hora cada una multiplicada por su respectiva ponderación:

$$\text{HorasTotales} = \sum_{i=1}^n p_i * x_i$$

Los tipos de hora definidos en el sistema son:

- Teórico presencial
- Práctico presencial
- Práctico no presencial
- Laboratorio presencial
- Laboratorio no presencial
- Evaluación
- Estudio

Luego, en la relación cuerpo de conocimiento con plan de estudio se define la ponderación para cada tipo de hora. A modo de ejemplo, si se tuviese la siguiente ponderación: 3 para las horas de teórico presencial, 1 para las horas prácticas no presenciales, y 0 para el resto, y se realizara la evaluación de cubrimiento temático considerando horas totales, entonces el tiempo real se calcularía como:

$$\begin{aligned} \text{HorasTotales} &= 3 * \text{horasTeóricoPresencial} \\ &+ 1 * \text{horasPrácticasNoPresenciales.} \end{aligned} \quad (1)$$

#### MAPEO DE DEDICACIONES A ÍTEMS DEL CUERPO DE CONOCIMIENTO

Se deben poder asignar las horas registradas en la edición de un curso, a los ítems de los cuerpos de conocimiento asociados a los planes de estudio en los que se dictó la edición.

Al mapear una dedicación con un ítem de la jerarquía, la dedicación se considera mapeada con todos los ancestros del ítem. En este sentido, las horas registradas para un ítem contabilizan para la suma de horas de todos sus ancestros.

Cada dedicación puede mapearse a uno o más elementos del cuerpo, y se debe registrar cómo se distribuyen las horas de la dedicación entre los distintos elementos con los que se corresponde. Existen tres tipos de relación con la que se puede modelar cada correspondencia entre una dedicación y un ítem:

**TIPO NORMAL** El ítem es un elemento hoja del cuerpo, por lo que las horas se asignan directamente al ítem.

**TIPO VARIAS (VA)** El ítem tiene descendientes, y la dedicación se corresponde con varios o todos los hijos. En este caso, el tiempo dedicado a cada ítem descendiente no es significativo por sí solo, de modo que la dedicación se vincula directamente al padre.

**TIPO NINGUNA (NG)** El ítem tiene descendientes, y la dedicación no se corresponde con ningún hijo.

El sistema también permitirá indicar que un tema no pertenece al cuerpo de conocimiento, es decir, que no se corresponde ni está incluido en ninguno de sus ítems.

#### REUTILIZAR REGISTRO: CURSO BAJO GESTIÓN DE LA CONFIGURACIÓN

El sistema debe permitir registrar de forma sencilla los tiempos por tema y el mapeo de temas a tópicos para los cursos que están bajo gestión de la configuración para un cuerpo de conocimiento.

En el caso que sea un curso bajo gestión de la configuración el sistema permitirá:

- Ingresar registro de horas y mapeo de temas como si fuese un nuevo curso.

- Seleccionar una edición anterior del curso y reutilizar sólo su registro de horas.
- Seleccionar una edición anterior del curso y reutilizar su registro de horas y el mapeo de temas para uno o varios cuerpos de conocimiento.

#### CONFIGURAR PERFILES PARA ANALIZAR EJECUCIONES

El sistema debe modelar la evaluación de cubrimiento temático de forma genérica, permitiendo personalizar los criterios de evaluación. En este sentido, se identifican distintos aspectos que pueden ser configurables. Se permite personalizar el criterio de horas mínimas que determina si un elemento hoja del cuerpo está cubierto o no y las franjas de cubrimiento, que determinan el nivel de cubrimiento para el resto de los elementos. Por otro lado, se puede configurar la dedicación mínima y máxima para cada elemento del cuerpo y los tipos de hora a considerar para el análisis. Las configuraciones deben poder guardarse en forma de perfiles, para luego ser reutilizadas en otras evaluaciones. En el anexo A "*Documento de requisitos*" se explica de forma detallada y ejemplificada como se modela esta generalización en el sistema.

#### VISUALIZACIÓN DE RESULTADOS OBTENIDOS

El sistema debe permitir observar el resultado de evaluar el cubrimiento temático de la ejecución de un plan de estudio, o de la selección personalizada de ediciones de cursos respecto a un cuerpo de conocimiento. En el caso de que se cuente con más de un cuerpo vinculado al plan de estudio el usuario podrá seleccionar con cual desea realizar la evaluación. Además, se debe seleccionar el perfil de configuración a utilizar para que el sistema realice las evaluaciones de cubrimiento, esfuerzo y balance temático.

El cubrimiento temático busca determinar para cada elemento, si está cubierto o no. Para ello, el sistema utiliza la configuración de cada nivel para evaluar sus elementos, determinar a qué franja de cubrimiento pertenecen y luego los colorea según la configuración de la franja. Para cada elemento se podrá visualizar en un gráfico de torta cómo se cubren sus elementos descendientes.

Por otro lado, se debe evaluar el esfuerzo temático. Esto es, la comparación en horas del tiempo real dedicado a cada ítem con respecto a su tiempo mínimo y máximo esperado. En este sentido, el sistema permite configurar para qué niveles aplica este estudio, y colorear los ítems según

si la dedicación se encuentra por debajo del mínimo, en la franja, o por encima del máximo.

Análogamente al esfuerzo, el sistema permite definir un porcentaje mínimo y un porcentaje máximo para evaluar el balance temático de los ítems de aquellos niveles que el usuario indique que corresponda. Para los ítems de mayor nivel, es decir, los que no tienen padre, se calcula el porcentaje de tiempo que se les dedica tomando como 100% la suma de horas de todos los ítems de mayor nivel. Para los ítems de niveles inferiores, se calcula su porcentaje a partir de tomar el tiempo dedicado a su padre como 100%. El sistema coloreará cada ítem según se encuentre por debajo, entre, o por encima de su franja de porcentaje esperado.

Para todas las consultas mencionadas, el usuario podrá navegar por la jerarquía del cuerpo de conocimiento, permitiendo visualizar el resultado de la evaluación para los elementos de los distintos niveles. Todos los resultados se deben poder exportar a formato *xlsx* y las gráficas podrán ser descargables. Además, para cada elemento del cuerpo el sistema debe permitir visualizar el detalle de las dedicaciones que afectan su cubrimiento.

#### 4.4 REQUERIMIENTOS NO FUNCIONALES

En esta sección se detallan los requerimientos no funcionales del sistema.

##### UTILIZACIÓN DE SOFTWARE LIBRE

El sistema debe utilizar herramientas que no requieran la compra de licencias. En particular, para la implementación del servidor de *back-end* se debe utilizar *Java Platform*

##### DOCUMENTACIÓN

El sistema deberá contar con un manual de usuario, de instalación y un diccionario con el esquema de la base de datos.

##### NAVEGADORES

El sistema deberá ser compatible con los navegadores Chrome a partir de la versión 51.0.2704.63 y Firefox a partir de la versión 47.0, ya que estas son las últimas versiones estables.

##### DESCARGA DE RESULTADOS OBTENIDOS

El sistema permitirá exportar los resultados obtenidos de las consultas de forma tabular y gráfica.

**CARGA DE DATOS**

El sistema permitirá importar datos, como por ejemplo, registro de dedicaciones desde planillas odt y xlsx.

**PERFORMANCE**

El sistema deberá mostrar el resultado de las consultas en tiempos menores a 3 segundos.

**BASE DE DATOS**

El motor de base de datos debe ser relacional.

**4.5 INTERFACES DE USUARIO**

El sistema deberá proveer interfaces para los distintos usuarios. Los usuarios accederán al sistema mediante una interfaz web. Para poder ingresar al sistema todo usuario deberá autenticarse, de esta forma el sistema reconocerá que tipo de usuario es y listará las funcionalidades que le corresponden. La interfaz del usuario administrador deberá incluir todas las funcionalidades de las demás interfaces, además de un módulo que permita el manejo de usuarios.

## AQUITECTURA PROPUESTA

---

En este capítulo se describe la arquitectura propuesta para el sistema construido siguiendo el modelo de vistas 4+1 de Krutchen [Krutchen, 1995] que sugiere presentar el sistema desde distintas vistas, donde cada una tiene como objetivo presentarlo desde un enfoque diferente.

En la sección 5.1 se presenta la vista de casos de uso, mostrando los más relevantes para la definición de la arquitectura del sistema.

En la sección 5.2 se presenta la vista lógica, donde se describe la arquitectura del sistema presentando varios niveles de refinamiento, identificando sus módulos principales con sus responsabilidades y dependencias.

En la sección 5.3 se presentan aspectos físicos del sistema, como topología e infraestructura necesaria, este enfoque se conoce como vista de distribución.

Luego en la sección 5.4 se presenta la vista de implementación, con información de los componentes físicos relativa a la tecnología, incluyendo el software de base de datos, las plataformas sobre las cuales va a estar montado el sistema y los ejecutables que lo componen.

En la sección 5.5 se presenta la vista de seguridad, con una descripción de los principales aspectos relativos a la seguridad en el sistema.

Por último, en la sección 5.6 se presenta la vista de datos, donde se detalla el Modelo Entidad Relación propuesto con sus respectivas restricciones

### 5.1 CASOS DE USO CRÍTICOS

En esta sección se presentan los principales casos de uso del sistema, denominados casos de uso críticos. Este subconjunto de casos de uso se debe elegir cuidadosamente entre todos los casos de uso utilizando principalmente dos criterios:

- Que intervengan el mayor número de componentes arquitectónicos posibles.
- Que intervengan los componentes que se vislumbran como críticos o más complejos.

En base a los criterios mencionados y a los casos de uso descritos en el capítulo 4 "Requisitos del sistema", se identificaron 3 casos de uso críticos:

**IMPORTACIÓN DE DEDICACIONES DESDE PLANILLAS EXCEL** El caso de uso importación de dedicaciones se considera crítico para la arquitectura debido a que involucra módulos de terceros relativos a la importación de archivos en diferentes formatos.

**MAPEO DE TEMAS A TÓPICOS** Este caso de uso se considera crítico para la arquitectura debido a que es de los principales casos de uso del sistema e involucra una gran cantidad de entidades del modelo de dominio. A su vez en este caso de uso se valida el uso de un módulo de terceros para la implementación de grillas.

**VISUALIZACIÓN DE RESULTADOS OBTENIDOS** Las consultas sobre los datos se consideran críticas dado que es una de las funcionalidades más importantes del sistema y que su implementación involucra realizar operaciones sobre gran cantidad de datos, pudiendo afectar la performance del sistema. A su vez en este caso de uso se valida utilizar un módulo de terceros para mostrar las gráficas en la interfaz. Dado que todas las consultas implican los mismos riesgos, elegimos el caso de uso visualización de esfuerzo temático para mitigarlos.

La figura 8 presenta el diagrama de casos de uso críticos y su interacción con el sistema.

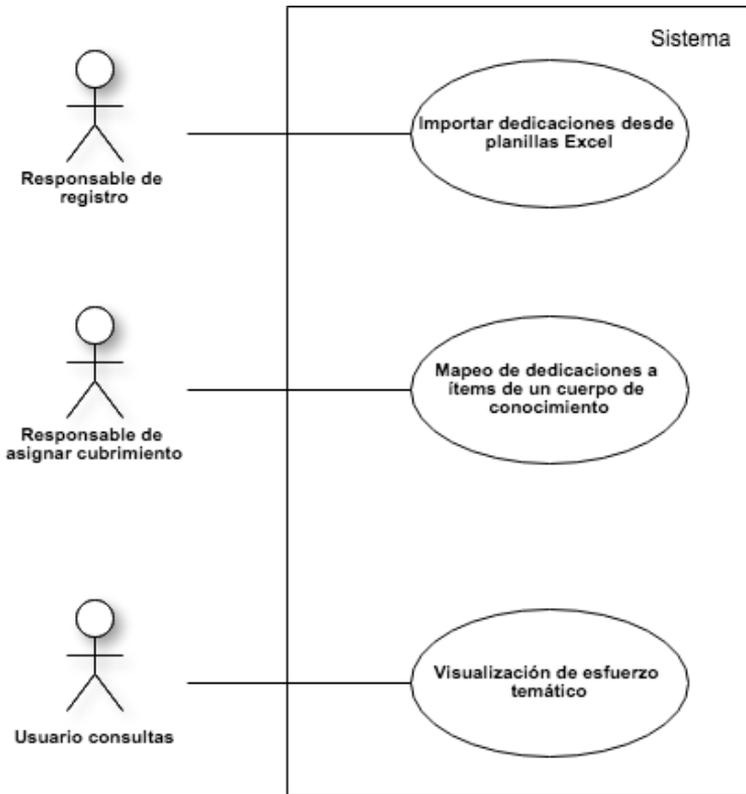


Figura 8: Casos de uso críticos para la definición de la arquitectura

Luego en los cuadros 6, 7 y 8 se especifican los casos de uso Importar dedicaciones desde planillas Excel, mapeo de dedicaciones a ítems de un cuerpo de conocimiento y visualización de esfuerzo temático respectivamente.

Nombre	Importar dedicaciones desde planillas Excel.
Objetivo	Importar desde una planilla en formato .xls o .odt dedicaciones para una edición de curso.
Actores	Responsable de registro.
Precondiciones	El curso debe existir en el sistema. El usuario debe conocer el formato de la planilla.
Descripción	El usuario selecciona un curso y una edición para la cual va a importar las dedicaciones subiendo la planilla en el formato acordado. El sistema guarda las dedicaciones.
Flujo normal	<ol style="list-style-type: none"> <li>1. El usuario selecciona curso y edición.</li> <li>2. El usuario selecciona la planilla a cargar.</li> <li>3. El sistema recibe la información y la planilla.</li> <li>4. El sistema realiza las validaciones de existencia, de formato y de tipos de datos.</li> <li>5. El sistema guarda las dedicaciones asociadas para esa edición.</li> </ol>
Flujo Alternativo	<p>4A. El formato de la planilla es inválido.</p> <ol style="list-style-type: none"> <li>1. El sistema despliega un mensaje de error.</li> <li>2. Vuelve a paso 2.</li> </ol>
Poscondiciones	Se crean en el sistema las dedicaciones asociadas a la edición seleccionada.

Cuadro 6: Especificación de caso de uso: Importación de datos desde planillas Excel

Nombre	Mapeo de dedicaciones a ítems de un cuerpo de conocimiento.
Objetivo	Mapear los registros de dedicaciones con ítems de un cuerpo de conocimiento.
Actores	Responsable de asignar cubrimiento.
Precondiciones	Deben existir dedicaciones en el sistema. Debe existir un cuerpo de conocimiento en el sistema.
Descripción	El usuario selecciona una dedicación y el/los ítems de conocimiento con el que corresponde mapear las horas. El sistema registra el mapeo.
Flujo normal	<ol style="list-style-type: none"> <li>1. El usuario selecciona curso, edición y un cuerpo asociado a alguna de las carreras que pertenece el curso.</li> <li>2. El Sistema lista las dedicaciones.</li> <li>3. El usuario selecciona una dedicación.</li> <li>4. El sistema lista los ítems del cuerpo seleccionado.</li> <li>5. El usuario selecciona uno o varios ítems indicando la cantidad de horas y el tipo de mapeo para ese ítem.</li> <li>6. El sistema valida que la suma de las horas asignadas no sea mayor a las horas de la dedicación.</li> <li>7. El sistema guarda los mapeos seleccionados para la dedicación.</li> <li>8A. El usuario elige volver al paso 3 para elegir otra dedicación.</li> <li>8B. El usuario elige terminar de mapear.</li> </ol>
Flujo Alternativo	<p>6A. La suma de horas asignadas es mayor a las horas de la dedicación.</p> <ol style="list-style-type: none"> <li>1. El sistema muestra un mensaje de error.</li> <li>2. Vuelve al paso 5.</li> </ol>
Poscondiciones	Se crean en el sistema los mapeos para las dedicaciones.

Cuadro 7: Especificación de caso de uso: Mapeo de dedicaciones a ítems de un cuerpo de conocimiento.

Nombre	Visualización de esfuerzo temático.
Objetivo	Visualizar gráficamente las horas dedicadas en comparación con las esperadas para un nivel jerárquico del cuerpo de conocimiento.
Actores	Usuario de consultas.
Precondiciones	Se cuenta con mapeos de dedicaciones a ítems de cuerpo de conocimiento en el sistema. Se cuenta con al menos un perfil de configuración.
Descripción	El usuario selecciona una ejecución de una implementación y un cuerpo de conocimiento. El sistema muestra gráficamente la comparación entre las dedicaciones esperadas y las reales por ítem de conocimiento, permitiendo navegar por niveles.
Flujo normal	<ol style="list-style-type: none"> <li>1. El usuario selecciona una carrera y un plan de estudio.</li> <li>2. El usuario selecciona una implementación del plan y una ejecución de la implementación.</li> <li>3. De los cuerpos de conocimiento asociados a la carrera, el usuario selecciona el que desea analizar.</li> <li>4. El sistema lista los perfiles configurados para el cuerpo de conocimiento.</li> <li>5. El usuario elige un perfil.</li> <li>6. El sistema muestra una gráfica donde se comparan las dedicaciones esperadas con las horas reales por ítem de conocimiento para el nivel de mayor jerarquía del cuerpo.</li> </ol>
Flujo Alternativo 1	<p>2A. El usuario simula una ejecución.</p> <ol style="list-style-type: none"> <li>1. El usuario elige un conjunto de ediciones de cursos para simular una ejecución.</li> <li>2. Sigue en el paso 3.</li> </ol>
Flujo Alternativo 2	<p>5A. El usuario elige un perfil.</p> <ol style="list-style-type: none"> <li>1. El usuario modifica parámetros del perfil.</li> </ol>
Flujo Optativo 1	7 A. El usuario selecciona un ítem y se vuelve al paso 6 pero para los hijos del ítem seleccionado.
Flujo Optativo 2	7 B. El usuario elige exportar la gráfica y la tabla que se muestra en pantalla.
Flujo Optativo 3	7 C. El usuario elige modificar los valores ya seleccionados, se vuelve al paso 1 pero con la información previamente cargada.
Poscondiciones	Se visualiza una gráfica con la comparación mencionada

Cuadro 8: Especificación de caso de uso: visualización de esfuerzo temático

## 5.2 VISTA LÓGICA DEL SISTEMA

La vista lógica permite describir el sistema en base a abstracciones fundamentales del diseño orientado a componentes para dar soporte a los requerimientos funcionales. En un enfoque *top-down*, se comienza por descomponer el sistema en un conjunto de subsistemas a partir de los cuales se realizan sucesivos refinamientos hasta llegar a las unidades lógicas más pequeñas [Taller de sistemas de información, 2015].

Se decidió utilizar un estilo arquitectónico en tres capas con el objetivo de desacoplar las partes que componen al sistema como se muestra en la figura 9. Esta separación permite dividir las responsabilidades de cada capa como se presenta a continuación.

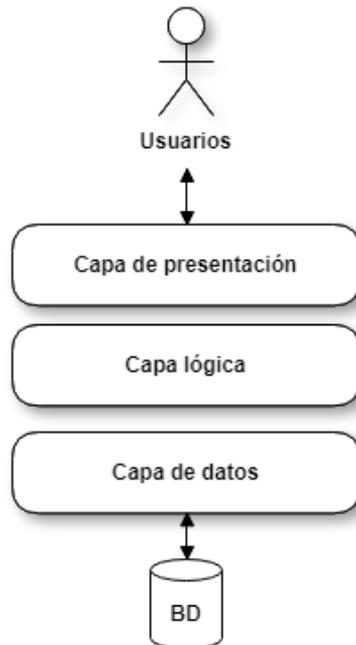


Figura 9: Vista lógica

La capa de presentación tiene como objetivo el manejo de la interfaz de usuario. Brinda servicios a los clientes y utiliza servicios de la capa lógica. Con el objetivo de brindar una experiencia más fluida a los usuarios se utiliza el pa-

trón de diseño *single page application* <sup>3</sup>. Este patrón consiste en cargar todo lo necesario para el funcionamiento de la página en la primera carga y luego el resto de los recursos a demanda.

En la capa lógica se implementa la lógica de negocio relevante a la aplicación. Consiste en componentes, los cuales exponen interfaces para que la capa superior y otros componentes utilicen. La figura 10 muestra el diagrama de componentes de la capa lógica.

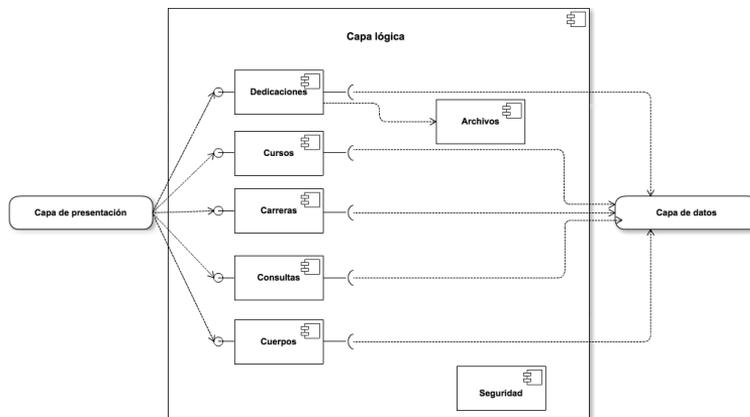


Figura 10: Componentes vista lógica

En el componente Cursos se desarrolla la lógica del sistema relativa al manejo de cursos. Entre sus interfaces se encuentran: alta, baja y modificación (ABM) de cursos y ediciones de cursos, crear edición de curso bajo gestión de la configuración, obtener cursos de un usuario, obtener ediciones de un curso, obtener cuerpos de conocimiento de un curso.

En el componente Consultas se desarrolla la lógica del sistema relativa a la visualización de consultas. Su principal funcionalidad es procesar los datos del sistema para las gráficas. El componente incluye las interfaces: obtener gráficas de esfuerzo, cubrimiento y balance temático para un perfil, un cuerpo y una ejecución dadas, además de la exportación de resultados obtenidos en el análisis.

En el componente Dedicaciones se desarrolla la lógica del sistema relativa a la gestión de tiempo dedicado a los temas de los cursos. Entre las funcionalidades de este componente se encuentran el ABM de dedicaciones, y sus mapeos

<sup>3</sup> <https://msdn.microsoft.com/en-us/magazine/dn463786.aspx>

con los ítems de conocimiento. Entre las principales interfaces se encuentran: obtener dedicaciones de una edición, guardar mapeo de una dedicación a ítems de un cuerpo, agregar dedicación a una edición e importar planilla con dedicaciones.

En el componente Cuerpos se desarrolla la lógica del sistema relativa a la gestión de los cuerpos de conocimiento. Entre sus interfaces se encuentran la alta, baja y modificación de los cuerpos de conocimiento y obtener perfiles de configuración disponibles para un cuerpo.

En el componente Archivos se desarrolla la lógica del sistema relativa al manejo de archivos. Este componente brinda interfaces distintas según la extensión del archivo permitiendo obtener dedicaciones a partir de formatos (.xls, .xlsx, .odt).

En el componente Carreras se desarrolla la lógica del sistema relativa a las carreras. Su principal funcionalidad es la gestión de carreras y planes de estudio. Entre sus interfaces encontramos: ABM de carreras, planes de estudio, implementaciones y ejecuciones, obtener carrera de un usuario, obtener plan de estudio de una carrera, obtener implementaciones de un plan de estudio, obtener Cuerpo de conocimiento de un plan de estudio.

Por último, la capa de datos es la capa inferior del sistema y su objetivo es el manejo de los datos y la interacción con la Base de Datos. Se utiliza la técnica de ORM (*Object-Relational mapping*)<sup>4</sup> para convertir los datos de la base relacional en el lenguaje orientado a objetos del sistema.

Para mostrar la interacción entre los componentes del sistema se realizaron diagramas de secuencia para los casos de uso críticos, que se presentan en las figuras 11, 12 y 13.

---

<sup>4</sup> <http://hibernate.org/orm/what-is-an-orm/>

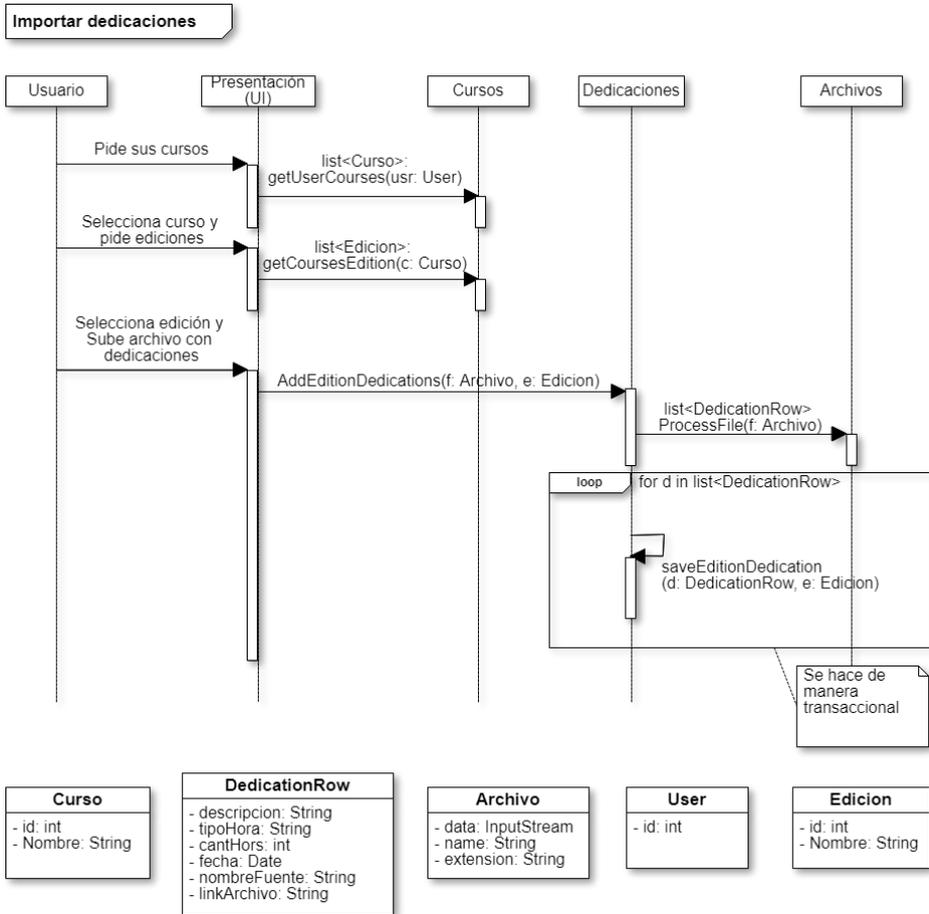


Figura 11: Diagrama de interacción - Importar dedicaciones

**Mapeo dedicaciones**

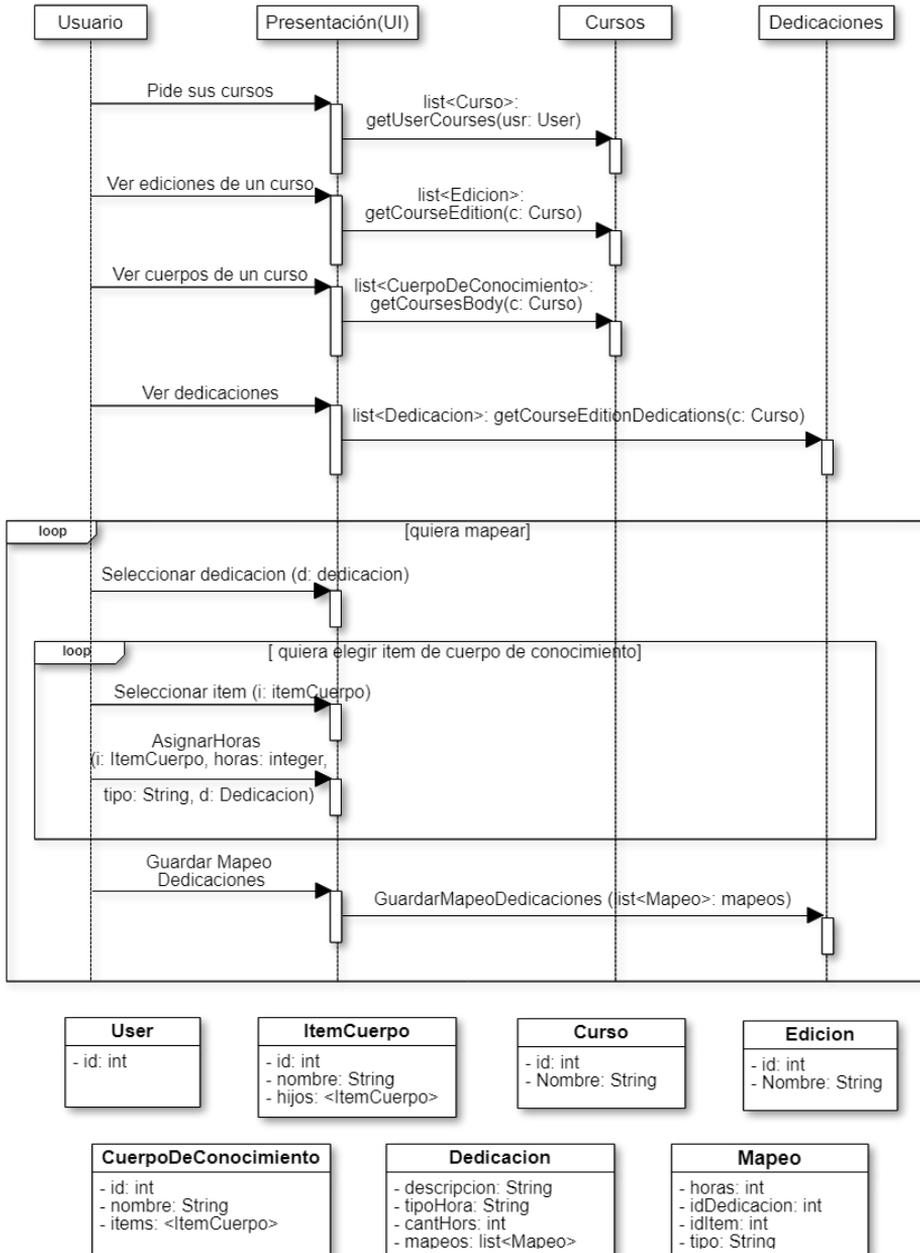


Figura 12: Diagrama de interacción - Mapeo de dedicaciones

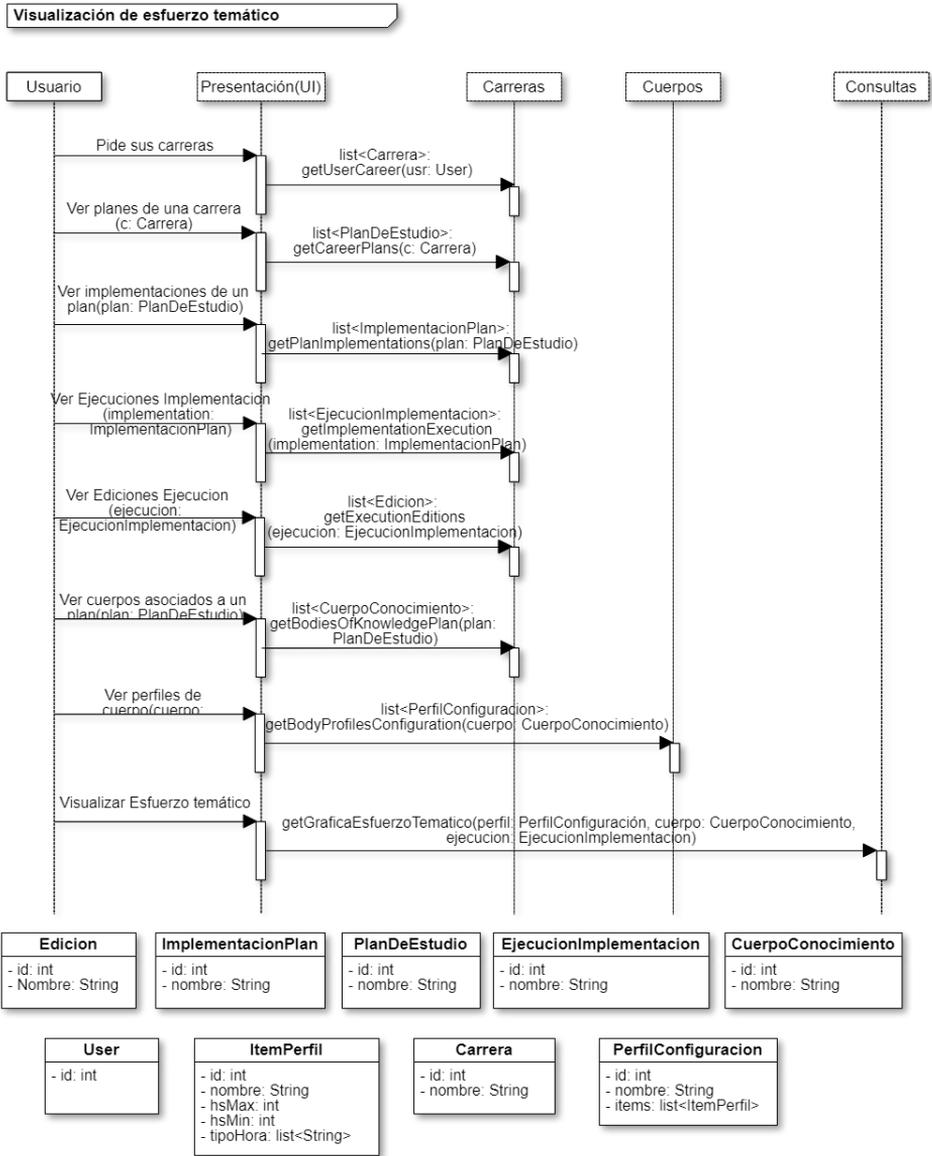


Figura 13: Diagrama de interacción - Visualización de esfuerzo temático

### 5.3 VISTA DE DISTRIBUCIÓN DEL SISTEMA

En esta vista se muestra la infraestructura necesaria para desplegar el sistema. Se presenta la arquitectura del sistema describiendo los nodos físicos que lo componen.

El sistema se compone de tres nodos como puede observarse en la figura 14

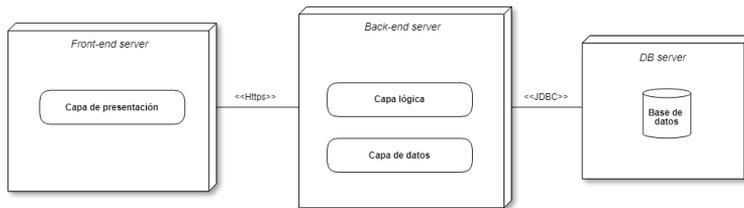


Figura 14: Vista de distribución

En el nodo *Front-end server* se tiene un servidor encargado de hospedar la capa de presentación. Realiza pedidos al servidor de *Back-end* utilizando el protocolo *HTTP* <sup>5</sup> sobre *SSL* <sup>6</sup>.

En el nodo *Back-end server* se tiene un servidor responsable de hospedar la lógica de negocio y atender los pedidos de la capa de presentación. Interactúa mediante el protocolo *JDBC* <sup>7</sup> con el servidor de base de datos.

En el nodo *Database server* se tiene un servidor de base de datos relacional donde se encuentra la base de datos del sistema.

Esta arquitectura distribuida busca separar las responsabilidades de cada nodo, permitiendo optimizar los recursos de acuerdo a sus necesidades. Por ejemplo, la base de datos va a requerir optimizar los accesos a disco mientras que los servidores de *front-end* y de *back-end* los tiempos de respuesta.

Por otra parte esta arquitectura permite escalar eficientemente al sistema para los nodos de *back-end* y *front-end*. En caso de picos de tráfico, basta agregar instancias en el nodo que no esté pudiendo atender todos los pedidos para no degradar la performance del sistema. Luego al disminuir el tráfico se puede bajar la cantidad de nodos y de esa forma disminuir el costo en infraestructura.

Otra ventaja de la separación en nodos es que cada uno se puede monitorear de forma independiente.

<sup>5</sup> <http://www.ietf.org/rfc/rfc2616.txt>

<sup>6</sup> <https://tools.ietf.org/html/rfc6101>

<sup>7</sup> [http://download.oracle.com/otn-pub/jcp/jdbc-4\\_1-mrel-spec/jdbc4.1-fr-spec.pdf](http://download.oracle.com/otn-pub/jcp/jdbc-4_1-mrel-spec/jdbc4.1-fr-spec.pdf)

Pensando en construir un sistema extensible, se utiliza el estilo de arquitectura *REST*<sup>8</sup> para modelar la interacción entre los proyectos de *front-end* y *back-end*. La arquitectura *REST* se basa en un conjunto de operaciones bien definidas para la comunicación entre dos nodos de forma de cliente-servidor. *REST* realiza pedidos *HTTP*, utilizando los métodos *GET*, *POST*, *PUT* y *DELETE* sobre los recursos del sistema. Además, los datos se transfieren en formato *JSON*<sup>9</sup>, por lo que cualquier cliente puede consumir los servicios sin importar en qué lenguaje estén implementados. De esta forma resulta sencillo agregar nuevas interfaces al sistema, por ejemplo una aplicación móvil que interactúe con la capa lógica.

Para la inicialización del sistema en el marco de este proyecto se utiliza una instancia de cada nodo.

#### 5.4 VISTA DE IMPLEMENTACIÓN DEL SISTEMA

La vista de implementación se enfoca en los componentes en tiempo de ejecución que forman el sistema, como se presenta en la figura 15.

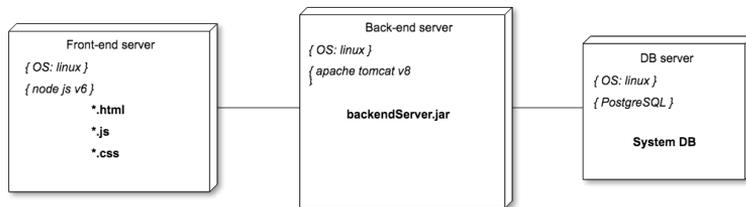


Figura 15: Vista de implementación

En el nodo *Front-end server* encontramos un servidor node js corriendo sobre un sistema operativo Linux. En el mismo se sirven todos los archivos .html, .js y .css correspondientes a la capa de presentación del sistema.

Por otro lado, en el nodo *Back-end server* corre un servidor apache tomcat sobre un sistema operativo Linux donde se ejecuta el backendServer.jar encargado de empaquetar la lógica de negocio.

Por último, en el nodo *Database server* se ejecuta un servidor postgresSQL, donde se aloja la base de datos del sistema.

<sup>8</sup> <https://spring.io/understanding/REST>

<sup>9</sup> <https://www.json.org/>

## 5.5 VISTA DE SEGURIDAD

En esta vista se describen los mecanismos utilizados para satisfacer las necesidades de seguridad del sistema: autenticación, autorización y comunicación segura de datos.

Respecto a la autenticación es necesario conocer la identidad de los usuarios que utilizan el sistema, por lo tanto se emplea un mecanismo de autenticación basado en usuario y contraseña. Para ello el sistema guarda la contraseña encriptada de cada usuario y una vez que el usuario ingresa sus credenciales, se verifica con la que se encuentra en la base de datos. En caso de ser iguales, el sistema retorna un *token* con fecha de expiración, el cual será utilizado por el usuario para realizar los siguientes pedidos al servidor de aplicaciones. Una vez que el usuario cierre sesión o el *token* expire, el sistema volverá a pedir sus credenciales.

Se utiliza un mecanismo de autorización basado en roles y en datos asociados a cada usuario, con el fin de determinar a qué recursos del sistema tienen acceso y qué operaciones pueden realizar los usuarios. Por ejemplo, un usuario con el rol "responsable de registro" puede acceder al sistema para ingresar los tiempos dedicados únicamente para aquellas ediciones de las cuales es responsable de su registro.

Por último la comunicación entre los navegadores y los servidores se realiza utilizando el protocolo HTTP sobre SSL. El protocolo permite que la información que se transfiere viaje encriptada. De esta forma se protegen los datos sensibles de un posible atacante que consiga interceptar la transferencia de datos.

## 5.6 VISTA DE DATOS

La figura 16 representa de forma gráfica el modelado de las distintas entidades de la realidad y sus relaciones. Este modelo, se propone como punto de partida para el diseño de la base de datos relacional que consume el sistema.



#### RESTRICCIONES DE AUTO-RELACIÓN

1. Un ítem de conocimiento no puede ser padre de sí mismo (irreflexiva).
2. Un ítem de conocimiento no puede ser padre de un ítem que es padre de él (asimétrica).
3. El padre de un ítem de conocimiento "X" es padre de todos los ítems de conocimiento de los que el ítem "X" es padre (transitiva).

#### RESTRICCIONES DE RELACIONES CIRCULARES

1. Una Dedicación "D" sólo se podrá mapear con una Hora por ítem de conocimiento "H", si "D" se registra para Ediciones de cursos que se dan en un Plan de estudio que esté relacionado con el ítem de conocimiento que contabiliza "H".

#### RESTRICCIONES DE NEGOCIO

1. Para cada ítem de conocimiento, el nivel asociado al mismo debe ser igual al nivel del ítem de conocimiento de su ítem padre más uno. O sea, se debe cumplir que  $\text{Nivel}(\text{ítem conocimiento}) = \text{Nivel}(\text{ítem conocimiento (padre)}) + 1$ . Si el ítem de conocimiento no tiene padre, entonces su nivel será 1.
2. La cantidad de horas de una Dedicación es mayor o igual a la suma de la cantidad de horas de las Horas por ítem de conocimiento que se mapean con dicha Dedicación, y que tienen igual tipo de hora.
3. Para cada relación entre Nivel de conocimiento de un cuerpo, y un Plan de estudio, se define el atributo "Horas cubrimiento mínimo" para aquellos niveles con mayor número de nivel, es decir, los que identifican a los ítems de conocimientos que no tienen hijos. Para el resto de los niveles se especifica el atributo "Porcentaje cubrimiento mínimo".
4. Las franjas de cubrimiento definidas para un Nivel de conocimiento y un Plan de estudio deben abarcar todo el espectro de 0-100 sin sobreponerse entre ellas, y dicho atributo tendrá sentido para aquellos niveles que identifican ítems de conocimiento que tienen hijos.
5. Los atributos esfuerzo horas mínimas, esfuerzo horas máximas, balance porcentaje mínimo y balance porcentaje máximo en la relación que se definen entre ítem de conocimiento y plan de estudio, sólo son requeridos para ítems que pertenecen a niveles de conocimiento que tienen el atributo aplica dedicación en verdadero.



## CONSTRUCCIÓN DEL SISTEMA

---

El objetivo de este capítulo es presentar aspectos relevantes a la construcción e implantación del sistema. En la sección 6.1 se presentan las tecnologías de desarrollo utilizadas, y los criterios con qué se seleccionaron. Luego, en la sección 6.2 se explica cómo se realiza la implantación del sistema. Por último en la sección 6.3 se presentan otras tecnologías analizadas.

### 6.1 TECNOLOGÍA Y HERRAMIENTAS UTILIZADAS

El objetivo de esta sección es presentar las tecnologías utilizadas para el desarrollo del proyecto.

En primer lugar, las herramientas y lenguajes seleccionados deben asegurar los siguientes requisitos no funcionales: ser de código abierto o uso libre, y que el programa resultante sea independiente de la plataforma en la que se vaya a ejecutar.

Además, interesa considerar tecnologías de uso actual en la industria, teniendo en cuenta la curva de aprendizaje de las distintas alternativas, y priorizando aquellas que resulten más fáciles de aprender. Esta perspectiva tiene dos motivos. Por un lado, se entiende que esta decisión facilita al futuro mantenimiento del proyecto. Por otra parte, se intenta contrarrestar la falta de experiencia en desarrollo del equipo.

Para realizar la selección, se investigan varias alternativas de forma teórica, leyendo sus respectivas documentaciones y foros de la comunidad. Luego, se realiza un prototipo de aprendizaje para validar que el conjunto de tecnologías seleccionado cumpla con los lineamientos antes mencionados, y sean compatibles entre sí.

#### 6.1.1 *Servidor de Back-End*

Se describen a continuación las herramientas y tecnologías utilizadas para construir el servidor de *back-end*.

JAVA PLATFORM, ENTERPRISE EDITION (JEE) <sup>10</sup> Es una plataforma de código abierto y orientado a objetos, utilizada para desarrollar y ejecutar *software* de aplicaciones. Permite crear aplicaciones portables, independientes a la plataforma en la que ejecutan, a partir de la especificación de varias interfaces de programación de aplicaciones (API, según sus siglas en inglés). También permite crear arquitecturas de N capas distribuidas, facilitando la escalabilidad. Dado que JEE viene en continua mejora desde el 2002, fecha en la cual fue distribuido el primer *software development kit* (SDK) y a la alta aceptación que tiene en el mercado, se puede asegurar que es una tecnología estable.

JAVA DATABASE CONNECTIVITY (JDBC) <sup>11</sup> Es la especificación de una API estándar que permite que los programas Java realicen consultas a múltiples sistemas de gestión de bases de datos SQL<sup>12</sup>. Permite al sistema realizar consultas de forma independiente al ambiente, tanto del sistema operativo como del sistema de gestión de bases de datos (DBMS) utilizado.

JAVA PERSISTENCE API (JPA) <sup>13</sup> Es una API de persistencia desarrollada para la plataforma Java que provee el mapeo entre los objetos de la programación orientada a objetos y los objetos relacionales de la base de datos independientemente del proveedor de bases de datos elegido. JPA crea un estándar para relacionar objetos con una o más tablas de una base de datos mediante el uso de metadatos asociado a objetos, los cuales pueden expresarse en forma de anotaciones o configuraciones XML. Permite al sistema actualizar, recuperar, eliminar o persistir objetos en una base de datos manejando la mayor parte de las operaciones, lo que hace que no sea necesario escribir código JDBC o SQL. Se utiliza la implementación de JPA *Hibernate*<sup>14</sup>. Este *framework* permite realizar el mapeo objeto-relacional (ORM) para la persistencia en bases de datos relacionales desde la plataforma Java utilizando la especificación JDBC. Facilita la correspondencia de atributos entre una base de datos y el modelo de objetos agregando anotaciones en los atributos de los objetos Java. En este sentido, permite aprovechar las ventajas de la programación orientada a objetos como herencia, polimorfismo, asociaciones, composiciones y colecciones aplicadas a la persistencia. Además, provee facilidades para la consulta y recu-

---

10 <http://www.oracle.com/technetwork/java/javaee/overview/index.html>

11 <http://www.oracle.com/technetwork/java/javase/jdbc/index.html>

12 <http://www.oracle.com/technetwork/java/javase/jdbc/index.html>

13 [https://www.ibm.com/support/knowledgecenter/es/SSAW57\\_8.5.5/com.ibm.websphere.wlp.nd.doc/ae/cwlp\\_jpa.html](https://www.ibm.com/support/knowledgecenter/es/SSAW57_8.5.5/com.ibm.websphere.wlp.nd.doc/ae/cwlp_jpa.html)

14 <http://hibernate.org/orm/>

peración de datos, generando llamadas SQL y libera al desarrollador del manejo y conversión de los resultados.

**POSTGRESQL** <sup>15</sup> Es un sistema de gestión de bases de datos relacional orientado a objetos y de código libre.

**APACHE MAVEN** <sup>16</sup> Es una herramienta de software para la gestión y construcción de proyectos Java. Facilita el manejo de dependencias y la compilación, además de brindar la posibilidad de generar proyectos modulares en lugar de un único proyecto monolítico. Se basa en utilizar un archivo POM<sup>17</sup> donde se describe el proyecto a construir, sus dependencias con otros módulos y el orden de construcción de los elementos. Maven existe desde el año 2002, y desde entonces cada año se ha liberado al menos una nueva versión de la herramienta.

**SPRING** <sup>18</sup> Es un *framework* que provee un modelo de programación y configuración que facilita el desarrollo de aplicaciones basadas en Java EE. Entre las principales funcionalidades que provee se encuentran: el manejo de vida de los objetos, manejo de transacciones, inyección de dependencias y el acceso a datos.

Spring presenta una arquitectura modular con una configuración inicial, donde el resto de los módulos son seleccionados dependiendo de las necesidades del sistema, permitiendo crear sistemas livianos y soportar distintas arquitecturas. También provee una estructura en capas bien delimitadas, motivando el uso de buenas prácticas de programación. A su vez, ofrece diversos módulos que solucionan problemas comunes de la programación.

Para el desarrollo de la solución se utiliza la versión 4.3.3 del *framework*. Uno de los módulos utilizados es *Spring Boot*<sup>19</sup>, diseñado para crear y colaborar con la infraestructura de aplicaciones *stand-alone*<sup>20</sup> que sean fáciles de correr y mantener. Esto libera al programador de realizar configuraciones referentes al despliegue de la aplicación, usando un servidor embebido, por defecto *Tomcat*<sup>21</sup>.

---

15 <https://www.postgresql.org/>

16 <https://maven.apache.org/>

17 <https://maven.apache.org/guides/introduction/introduction-to-the-pom.html>

18 <https://projects.spring.io/spring-framework/>

19 <https://projects.spring.io/spring-boot/>

20 <https://dictionary.cambridge.org/es/diccionario/ingles/standalone>

21 <https://tomcat.apache.org/>

También se utiliza el módulo *Spring Security*<sup>22</sup>, el cual provee mecanismos de autenticación y autorización por medio de la asignación de roles a grupos de usuarios. Permite gestionar la seguridad en varios niveles, desde las URL que se solicitan al servidor, hasta el acceso a métodos, clases Java, o instancias concretas de clases. El *framework* aporta la ventaja de separar la lógica de negocio de la lógica de control de seguridad de la aplicación, con la utilización de filtros para las peticiones al servidor y otros aspectos más avanzados como la seguridad de clases y métodos.

APACHE POI, APACHE ODF TOOLKIT Para el manejo de archivos se utilizaron las librerías provistas por *Apache POI*<sup>23</sup> y *Apache ODF Toolkit*<sup>24</sup>, las cuales ofrecen una API para aplicaciones Java que permite escribir, modificar y extraer datos de planillas de textos. *Apache POI* es una interfaz para archivos Microsoft Office con formato Word, PowerPoint y Excel, mientras que *Apache ODF Toolkit* es para archivos *Open Document Format* (ODF) con extensión odt.

### 6.1.2 Servidor de Front-End

Se describen a continuación las herramientas y tecnologías utilizadas para la construcción del servidor de *front-end*.

REACT<sup>25</sup> Es una librería de JavaScript para crear interfaces de usuarios, desarrollada por la empresa Facebook con el objetivo de facilitar la creación de componentes interactivos y reutilizables. Es declarativo, es decir, permite diseñar vistas para los distintos estados por los que va pasando la aplicación, reaccionando a la interacción del usuario y a la comunicación con el *back-end*. React ofrece mecanismos para reaccionar a las distintas interacciones, y actualizar la interfaz cuando el estado cambia. Las vistas declarativas facilitan la construcción de un código simple de entender y depurar. El diseño de React está basado en componentes que manejan su propio estado, promoviendo buenas prácticas, entre ellas modularidad y un flujo claro de datos y eventos. Al ser una librería, proporciona flexibilidad al desarrollador en cuanto a la estructura y diseño de la solución, dejando libertad para manejar aspectos como enrutamiento o mantenimiento de estado, entre otros. La elección de esta librería para la construcción de las

22 <https://docs.spring.io/spring-security/site/docs/5.0.3.RELEASE/reference/htmlsingle/>

23 <https://poi.apache.org/>

24 <https://incubator.apache.org/odftoolkit/>

25 <https://reactjs.org/>

interfaces de usuario del sistema se basa en las ventajas mencionadas anteriormente, junto con la madurez que ofrece, su popularidad y extensa comunidad hace que su documentación se encuentre en gran cantidad y calidad, asegurando también el mantenimiento de la librería. El uso de esta tecnología por empresas con requerimientos de mucho tráfico como Facebook, Airbnb, Netflix e Instagram brinda confianza en que no conlleva problemas de *performance*.

**NODEJS** <sup>26</sup> Es un entorno de ejecución de código abierto para JavaScript. Es asíncrono, y utiliza un modelo de operaciones de entrada y salida sin bloqueo orientado a eventos, que lo hace liviano y eficiente. NodeJS se utiliza como servidor del proyecto de *Front-end*.

**HIGHCHARTS** <sup>27</sup> Es una librería mantenida por la empresa Highsoft, escrita en JavaScript que ofrece una amplia variedad de gráficas interactivas para aplicaciones web. Es de libre uso para proyectos personales o educativos.

**REACT DATA GRID** <sup>28</sup> Esta librería provee un componente que permite crear grillas con las facilidades que ofrece una hoja de cálculo. Entre las facilidades que implementa se encuentran: ordenar, filtrar, agregar, navegar por las celdas, sugerencias de enumerados para rellenar una columna, extender y colapsar filas, entre otras. Por todas las funcionalidades que ofrece, y su semejanza con la usabilidad de las planillas electrónicas a las que estaban acostumbrados los usuarios del sistema se opta por agregar esta librería frente a otras alternativas parecidas. Además, React Data Grid posee documentación concisa y actualizada, y provee mecanismos para modificar sus funcionalidades preestablecidas.

**SHEETJS** <sup>29</sup> Para el manejo de archivos se utiliza la librería *SheetJS js-xlsx*, la cual permite al sistema de *Front-end* exportar información en hojas de cálculo, sin que sea necesario realizar un pedido al *Back-end*. Este enfoque se puede realizar en aquellos casos que se tenga toda la información necesaria a exportar en el estado de la aplicación y no requiera complejidad en el diseño de generar el archivo, incrementando así la *performance* con la que se realiza la exportación.

La figura 17 presenta un diagrama que permite visualizar cómo interactúan las principales tecnologías seleccionadas.

26 <https://nodejs.org/es/>

27 <https://www.highcharts.com/>

28 <http://adazzle.github.io/react-data-grid/>

29 <https://github.com/SheetJS/js-xlsx>

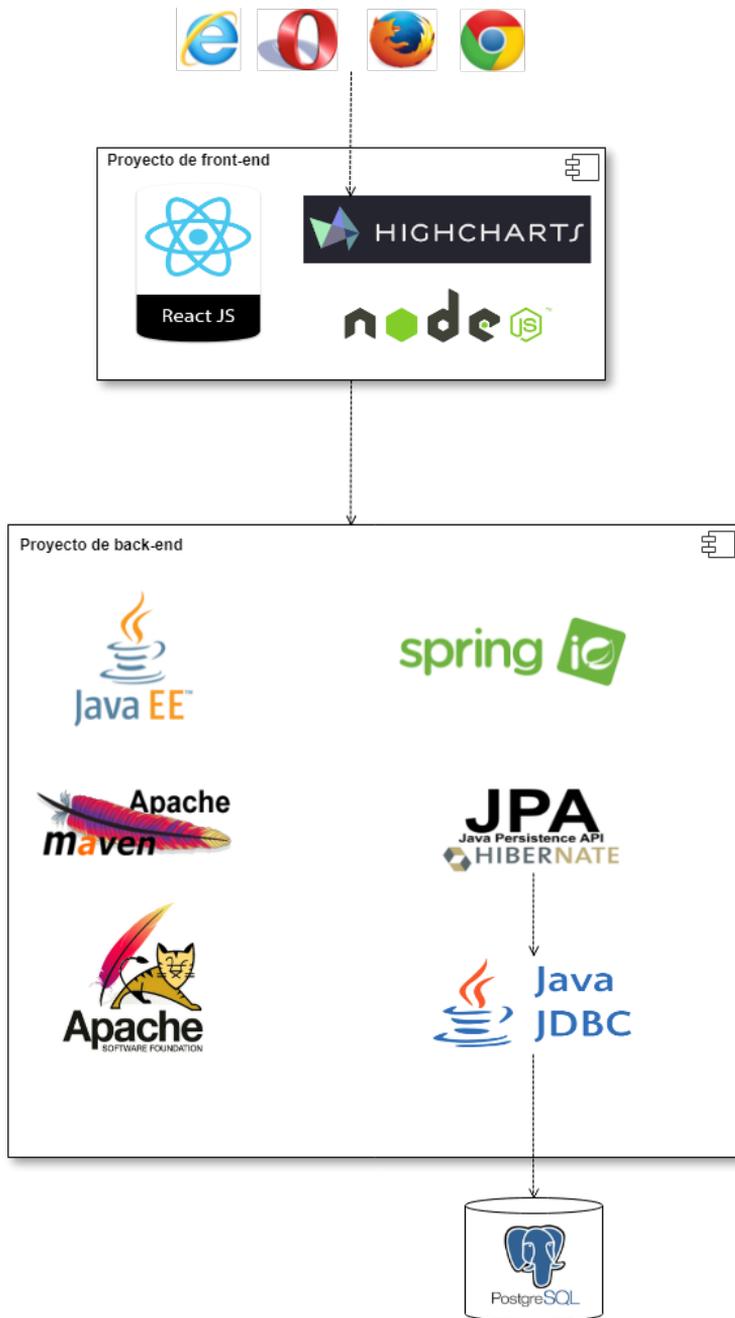


Figura 17: Diagrama de interacción de las tecnologías y herramientas utilizadas.

## 6.2 IMPLANTACIÓN

La liberación final del sistema se realiza en dos ambientes. Por un lado, el sistema se encuentra disponible en una plataforma como servicio<sup>30</sup> (PaaS), utilizando un plan gratuito con ciertas restricciones, a la que se accede a través de una URL pública. Por otro lado, se entrega una máquina virtual que actúa como servidor, permitiendo el acceso a la aplicación desde todos los dispositivos que se encuentren conectados a la misma red.

Se elige Heroku<sup>31</sup>, como entorno web para alojar y ejecutar el sistema en la nube. Dado que Heroku es una PaaS, además de brindar la infraestructura para poder alojar el sistema, también provee la configuración del sistema operativo, de la base de datos y el mecanismo de balanceo de carga si se extiende el plan a utilizar más de un nodo, entre otros servicios.

La principal limitación de utilizar el plan gratuito<sup>32</sup> de la plataforma es que la capacidad de la base de datos está restringida a diez mil filas. Además, cada ambiente cuenta como máximo con 512 MB de memoria RAM y las aplicaciones se duermen luego de treinta minutos de inactividad, pudiendo alcanzar un máximo de quinientas cincuenta horas de uso por mes. Estas restricciones motivan a la entrega de una máquina virtual al finalizar la construcción del sistema. De todos modos, se considera que es redituable mantener el ambiente en la nube, ya que permite liberaciones menos costosas para poder realizar la validación de cada iteración.

A pesar de las limitantes, se elige Heroku para realizar una construcción iterativa con liberaciones intermedias. Incluso, se considera que para un uso exhaustivo y extendido en el tiempo de la aplicación, es decir, poner el sistema en producción, una buena opción es extender el plan de Heroku. El plan puede ser ajustado a las necesidades de uso que se tengan, y de ello dependerá su costo.

## 6.3 ALTERNATIVAS ESTUDIADAS

Durante el proceso de selección de tecnologías, se analizaron alternativas en cuanto a las tecnologías para desarrollar el *front-end*, para la implementación de gráficas, y para el manejo de grillas. Además, se evaluaron varias posibilidades para dejar disponible la aplicación en un entorno web. A continuación, se explica cómo se realizaron las selecciones.

---

<sup>30</sup> <https://azure.microsoft.com/es-es/overview/what-is-paas/>

<sup>31</sup> <https://www.heroku.com/home>

<sup>32</sup> <https://www.heroku.com/pricing>

Al momento de elegir las tecnologías para desarrollar el *front-end* se evaluaron alternativas sobre Javascript, específicamente ReactJs y Angular, que son las más aceptadas y utilizadas en la industria. Angular es un *framework* creado por Google<sup>33</sup> para el desarrollo de aplicaciones *web* que brinda soluciones a distintos problemas comunes de implementación, como son el manejo de enrutamiento o la estructura del proyecto. Al brindar un marco de trabajo estándar tiene una curva de aprendizaje menor que React, que al ser una librería ofrece más flexibilidad al diseño de la solución, forzando a la toma de más decisiones por parte del desarrollador. Se realizó un prototipo con la versión dos de Angular para validar su funcionamiento pero en ese momento se encontraba en proceso de migración no retro-compatible, donde la versión dos no estaba lo suficientemente estable y no había certeza sobre el mantenimiento de la versión uno. Debido a esto se decide realizar un nuevo prototipo utilizando React que cumple con los requisitos del sistema.

Con respecto a la visualización de gráficas, se preseleccionaron dos alternativas para implementarlas, tomando en cuenta recomendaciones y popularidad en la industria: Highcharts <sup>34</sup> y ChartJS <sup>35</sup>. Tanto para Highcharts como para ChartJS existen *react-highcharts* <sup>36</sup> y *react-chartjs* <sup>37</sup> respectivamente, que son módulos que facilitan la integración en proyectos React. Aunque se encontró que ChartJS es más utilizada debido a que tiene mejor rendimiento, se optó por utilizar Highcharts ya que provee el mecanismo para descargar las gráficas en varios formatos, lo cual es un requisito del sistema.

En relación al manejo de grillas, existen varias alternativas para manejar datos en forma tabular que permiten un manejo similar al de la herramienta Excel. Se encontraron librerías para React con ejemplos *online* que fueron considerados, entre ellos *react-datatable* <sup>38</sup>, *react-spreadsheet-component* <sup>39</sup>, *reactabular* <sup>40</sup>, *react-bootstrap-table* <sup>41</sup>, *react-spreadsheet* <sup>42</sup> y *react-data-grid* <sup>43</sup>.

Aunque todas las opciones anteriores presentan funcionalidades como editar datos, ordenar, personalizar estilos y formatos de celda, sólo algunas ofrecen funciones más avanzadas como filtrar, agregar filas, sugerir enumerados para

---

33 <https://www.google.com/about/>

34 <https://www.highcharts.com/>

35 <https://www.chartjs.org/>

36 <https://github.com/kirjs/react-highcharts>

37 <https://github.com/reactjs/react-chartjs>

38 [wmira.github.io/react-datatable/](http://wmira.github.io/react-datatable/)

39 [felixrieseberg.github.io/React-Spreadsheet-Component](http://felixrieseberg.github.io/React-Spreadsheet-Component)

40 <https://reactabular.js.org/>

41 <http://allenfang.github.io/react-bootstrap-table/example>

42 <https://react.rocks/example/react-spreadsheet>

43 <http://adazzle.github.io/react-data-grid/>

una columna, o exportar a CSV. De todas las alternativas, se optó por la librería *react-data-grid* ya que se considera que es la más completa con respecto a las funcionalidades que ofrece, y cuenta con mayor documentación. A pesar de no ofrecer la exportación a planillas Excel, se realizó un prototipo de prueba en el que se agregó esta funcionalidad a través de la librería *SheetJS js-xlsx* de forma exitosa.

Para la implantación del sistema, además de Heroku, se consideraron otras alternativas como *Amazon Web Services*<sup>44</sup>, *Red Hat OpenShift*<sup>45</sup>, *Bluemix*<sup>46</sup> y dejar el sistema disponible en un servidor de Facultad. Esta última opción se descarta debido a que la Facultad no cuenta con servidores o un servicio para desplegar aplicaciones resultantes de proyectos de grado. En el caso de *Amazon Web Services*, es obligatorio el ingreso de una tarjeta de crédito, incluso en planes gratuitos, por lo que tampoco resulta ser una opción viable. También se descarta *Red Hat OpenShift* ya que si la cuenta se encuentra inactiva por cierta cantidad de tiempo, se liberan los recursos, entre ellos la base de datos. Esta limitante implicaría hacer un desarrollo extra para el respaldo de información solamente para poder utilizar *OpenShift*. Luego, para el caso de *Bluemix*, es requisito tramitar una licencia para su uso, lo cual no fue posible al momento que se necesitaba.

---

44 <https://aws.amazon.com/es/>

45 <https://www.redhat.com/es/technologies/cloud-computing/openshift>

46 <https://www.ibm.com/cloud-computing/bluemix/es>



## CARGA DE DATOS

---

Una parte fundamental de este proyecto es dejar disponible en el sistema toda la información relativa a la evaluación del cubrimiento temático de la EIS con respecto al CBOK.

En el presente capítulo se muestra el trabajo realizado para cargar en la base de datos del sistema todos los registros generados hasta el momento. En la sección 7.1 se exponen los datos a cargar. En la sección 7.2 se presentan las alternativas consideradas para llevar a cabo la carga, y se introduce la herramienta seleccionada para su automatización. Por último, en la sección 7.3 se explica el trabajo realizado y las dificultades encontradas a lo largo del proceso.

### 7.1 DATOS EXISTENTES

El cubrimiento temático de la EIS se evalúa desde su comienzo en el 2012, a través del registro del tiempo dedicado a los distintos temas, y el mapeo de estos temas a los tópicos del CBOK usando planillas electrónicas.

Hasta el 2016 se cuenta con el registro de dedicaciones de cuarenta y tres ediciones de cursos. Cada edición de curso cuenta con una planilla denominada “Planilla para la asignación de temas y tópicos de un curso” donde se detallan las dedicaciones registradas para la edición y sus correspondencias con respecto al CBOK. De las cuarenta y tres planillas disponibles, veintiséis corresponden a cursos nuevos, mientras que diecisiete pertenecen a cursos bajo gestión de la configuración. Esto significa que las últimas diecisiete planillas no incluyen el registro completo de temas, sino que describen cambios menores que se realizaron con respecto a una edición anterior del curso.

Las planillas para la asignación de temas y tópicos de un curso de distintas ediciones no cuentan con un mismo formato. En principio se clasifican las planillas en dos tipos: las que cuentan con detalle de registro por temas dictados, y las que registran el tiempo directamente sobre los elementos del cuerpo de conocimiento.

Para las primeras ediciones de cursos, las planillas muestran directamente las horas asignadas a los tópicos del CBOK, mientras que para el resto de las ediciones, las horas se asocian a los temas impartidos en el curso, y se asignan estos temas a los tópicos. A continuación, se presenta un ejemplo de planilla del

segundo caso, correspondiente al curso “Introducción al CMMI-ACQ” edición 2013.

**PESTAÑA CRÉDITOS MATERIA:** Incluye información de la edición.

**PESTAÑA TEMAS DADOS:** Registro de dedicaciones en base a temas del curso, estructurados en las siguientes columnas:

- **Día:** Día de la semana en la que se registra la dedicación.
- **Fuente:** Referencia al material relacionado con la dedicación.
- **Fecha:** Fecha en que se registra la dedicación con formato MM/DD/AAAA
- **Temas:** Descripción del tema dado.
- **Tópicos CBOK:** Tópicos o elementos del CBOK con los que se corresponde la dedicación. Para cada uno de ellos, se detalla la cantidad de horas y si se relaciona de forma normal, a varios hijos del elemento (VA) o a ninguno (NG).
- **Horas:** Cantidad de horas dedicadas.
- **Tipo:** Texto que identifica el tipo de hora (por ejemplo: “teórico”).
- **Subtema:** Descripción más detallada del tema.

**PESTAÑA HORAS EN BASE A CBOK:** Para cada elemento del CBOK al cual se le asignó al menos una correspondencia, se detalla la suma de horas que se le dedicaron en la edición. Esta información se estructura en las siguientes columnas:

- **KA (Área de conocimiento)** Nombre del área de conocimiento del CBOK.
- **Nro Unidad** Identificador numérico de la unidad del CBOK.
- **Unidad** Nombre de la unidad.
- **Nro Tópico** Identificador alfanumérico del tópico de CBOK. Se compone del número de unidad, y el número de tópico separados por un punto (por ejemplo: 1.1).
- **Tópico** Nombre del tópico.
- **Subtópico** Campo que existe para algunos tópicos. En general es una descripción que agrega información extra sobre las dedicaciones.
- **Comentarios** Campo utilizado para agregar comentarios similar a la columna "subtópico".

- **Horas teórico (presencial)** Cantidad de horas teóricas presenciales que se le dedicaron al tópico.
- **Horas práctico presencial** Cantidad de horas de práctico presenciales que se le dedicaron al tópico.
- **Horas práctico no presencial** Cantidad de horas de práctico no presenciales que se le dedicaron al tópico.
- **Horas laboratorio presencial** Cantidad de horas de laboratorio presenciales que se le dedicaron al tópico.
- **Horas laboratorio no presencial** Cantidad de horas de laboratorio no presenciales que se le dedicaron al tópico.
- **Horas evaluación** Cantidad de horas de evaluación que se le dedicaron al tópico.
- **Horas totales** Cantidad de horas totales que se le dedicaron al tópico, calculadas en base a ponderaciones descritas en la hoja Créditos Materia.

Por otra parte, para el caso del curso “Prácticas de Desarrollo de Ingeniería de Software: Diseño de Software” edición 2012 se cuenta con una planilla que no brinda detalle de los temas dictados en el curso. En este caso, se registran las horas directamente sobre los elementos del cuerpo, utilizando un formato similar al de la hoja Horas en base a CBOOK descrita anteriormente.

Además de las planillas correspondientes a cada edición, se cuenta con una única planilla consolidada que contiene la información de todas las ediciones de todos los cursos, cuyo objetivo es sumar el tiempo dedicado a cada elemento del CBOOK. La planilla consolidada cuenta con la lógica requerida para que el usuario pueda filtrar las ediciones de curso y los tipos de hora de su interés, y así obtener la evaluación del cubrimiento temático de las ediciones seleccionadas con respecto al CBOOK. La planilla consolidada recibe el nombre de “Planilla consolidada de asignación horas a tópicos” y tiene el siguiente formato:

**PESTAÑA CBOOK:** Hoja que contiene la sumarización por elementos del CBOOK para cada asignatura y cuenta con las siguientes columnas:

- **Asignatura (Edición):** Concatenación del nombre del curso y el año de la edición en el que se registra la información, con el formato  
 <nombre-curso> - <año>
- **Asignatura:** Nombre del curso.
- **Edición (Año):** Año de la edición.

- **Área de Conocimiento (KA):** Área de conocimiento con la que se corresponde el registro, con el siguiente formato  
     <código-area> - <nombre-área>
- **Nro Unidad:** Identificador de la unidad.
- **Unidad:** Nombre de la unidad.
- **Nro Tópico:** Identificador del tópico.
- **Tópico:** Nombre del tópico.
- **Subtópico:** Descripción extra del registro.
- **Comentarios:** Comentarios sobre el registro.
- **Horas teórico (presencial)** Cantidad de horas teóricas presenciales que se le dedicaron al ítem.
- **Horas práctico (presencial)** Cantidad de horas de práctico presenciales que se le dedicaron al ítem.
- **Horas práctico (no presencial)** Cantidad de horas de práctico no presenciales que se le dedicaron al ítem.
- **Horas laboratorio (presencial)** Cantidad de horas de laboratorio presenciales que se le dedicaron al ítem.
- **Horas laboratorio (no presencial)** Cantidad de horas de laboratorio no presenciales que se le dedicaron al ítem.
- **Horas evaluación** Cantidad de horas de evaluación que se le dedicaron al ítem.
- **Horas totales:** Cantidad de horas totales dedicadas al ítem.

PESTAÑA NIVEL KA Permite obtener el cubrimiento temático a nivel de áreas filtrando las ediciones que se desean tener en cuenta.

PESTAÑA NIVEL UNIDAD Permite obtener el cubrimiento temático a nivel de unidad filtrando las ediciones que se desean tener en cuenta.

PESTAÑA NIVEL TÓPICO Permite obtener el cubrimiento temático a nivel de tópico filtrando las ediciones que se desean tener en cuenta.

PESTAÑA KA POR ASIGNATURA Para cada asignatura se despliegan las áreas cubiertas para cada edición.

PESTAÑA ESFUERZO TEMÁTICO Análisis del esfuerzo temático.

PESTAÑA BALANCE Análisis del balance temático.

En el contexto de este proyecto, la planilla consolidada se utiliza como referencia para validar la carga de las planillas de registro. La validación consta de comparar para cada edición la suma del tiempo para cada ítem del cuerpo, corroborando que lo registrado en la base de datos del sistema coincide con la suma disponible en la planilla consolidada.

## 7.2 AUTOMATIZACIÓN DE LA CARGA

Toda la información presentada en la sección anterior debe incluirse en el esquema de la base de datos del sistema. Este esquema fue generado de forma automática a partir de las entidades modeladas en código, resultantes del diseño de la realidad. Aunque el modelado se realiza teniendo en cuenta los datos existentes, la estructura de las planillas electrónicas y la estructura de la base de datos no tienen una correspondencia directa, sino que los datos deben pasar por una serie de transformaciones antes de ser insertados.

Para hacer efectiva la migración se consideran las ventajas y desventajas de varias alternativas. Una primera opción es realizar la correspondencia entre la estructura de las planillas y el esquema de la base de forma manual, y generar un *script* con la secuencia de sentencias SQL necesarias para agregar todos los datos a la base. Una segunda alternativa es generar un código o programa que obtenga los datos de las planillas, le aplique las transformaciones necesarias, y acceda a la base de datos para almacenarlos en las tablas correspondientes. Otra posibilidad considerada es ingresar la información en la base de datos una vez que se cuenta con el sistema construido, a través del uso de su interfaz. Finalmente, se considera utilizar una herramienta ETL para automatizar el proceso de carga. A continuación se introduce este tipo de herramientas, y se mencionan las ventajas que presentan, y porqué es considerada la mejor alternativa para realizar la carga de datos en este proyecto.

Las herramientas ETL (*Extract, Transform, Load*) como su nombre lo indica tienen por objetivo facilitar el trabajo de extracción, transformación y carga de grandes cantidades de datos. La primera parte de un proceso ETL es la extracción de los datos desde múltiples fuentes. Las herramientas ETL proveen distintas interfaces para cada tipo de fuente, como por ejemplo una base de datos relacional, una base no relacional, una planilla Excel, un archivo en formato JSON, entre otros. Un mismo proceso puede consumir datos de más de una fuente, y combinarlos según sea necesario. Una vez extraídos los datos, comienza la fase de transformación. La transformación de un proceso ETL es la aplicación de una serie de reglas de negocio o funciones sobre los datos extraí-

dos para convertirlos en los datos que serán cargados. Entre las funciones que se pueden aplicar en el proceso de transformación se encuentran la selección de algunas columnas, la codificación o reemplazo de valores, la obtención de nuevos valores a partir de cálculos, la unión y separación de datos, normalización de columnas a filas, filtros, entre otras. Una vez que se llevan los datos a la estructura deseada, el proceso ETL concluye en la carga de los datos en el sistema de destino. El destino de un proceso ETL pueden ser una o más tablas de una base de datos relacional, archivos, u otras posibles salidas.<sup>47</sup>

Realizar la carga de datos a través de una herramienta ETL tiene varias ventajas en el contexto de las necesidades del proyecto con respecto a las otras alternativas consideradas. Por un lado, la incorporación de los datos en el sistema es fundamental para la validación del esquema de base de datos. Por este motivo, es necesario que la carga de datos se realice en simultáneo con la construcción del sistema, a modo de detectar de forma temprana los posibles cambios. Ante la posible necesidad de realizar ajustes sobre el esquema diseñado, es imprescindible que el mecanismo utilizado para la carga soporte cambios en la estructura de la base de datos sin implicar esfuerzo de re-trabajo. En este sentido, una vez programado el proceso ETL, puede ejecutarse tantas veces como sea necesario, adaptando la transformación a los cambios. Por otra parte, el volumen de datos disponibles requiere que el proceso de carga sea lo más automático posible. Cada planilla afecta a varias tablas, por lo que el trabajo de llevar control sobre claves foráneas resulta complejo de realizar de forma manual. A su vez, para las planillas que siguen un formato, se puede correr el mismo proceso reiteradas veces solamente modificando la fuente que consume.

Existen varias herramientas *open source* para la realización de procesos ETL. Para este proyecto se utilizó Kettle perteneciente a la suite de Pentaho Data Integration.<sup>48</sup>

### 7.3 TRABAJO REALIZADO

Debido a que aún dentro del mismo tipo de planillas se encuentran muchas variaciones entre las distintas ediciones, se define un nuevo formato homogéneo para cada tipo. El objetivo de este nuevo formato, es contar con una estructura estándar que concilie las diferencias entre las planillas de un mismo tipo. Al tener todas las planillas de un mismo tipo con el mismo formato, se puede

<sup>47</sup> <http://www.dataprix.com/blogs/respinosamilla/herramientas-etl-que-son-para-que-valen-productos-mas-conocidos-etl-s-open-sour>

<sup>48</sup> <http://www.pentaho.com/product/data-integration>

realizar una transformación ETL para el nuevo formato, y aplicarla a todas las planillas normalizadas.

La actividad de llevar cada planilla al formato estándar definido se realiza de forma manual, ya que las diferencias van desde pequeñas variaciones hasta cambios significativos en la estructura. Las planillas que registran tiempo sobre tópicos, se normalizan en las siguientes columnas:

- KA (Área de Conocimiento)
- Nro. Unidad
- Unidad
- Nro. Tópico
- Tópico
- Subtópico
- Horas teórico presencial
- Horas práctico presencial
- Horas práctico no presencial
- Horas laboratorio presencial
- Horas laboratorio no presencial
- Horas evaluación

Para cada elemento del cuerpo que tenga alguna hora registrada, se especifica el tiempo dedicado en las columnas que corresponda según el tipo de hora.

Luego, las planillas que presentan detalle de temas dictados se normalizan en las siguientes columnas:

- Día
- Tema
- Cbok (mantiene nomenclatura presentada para el caso CMMI -ACQ 2013)
- Horas
- Tipo de hora.

- Fuente
- Subtema

Para llevar los datos desde las planillas normalizadas hacia la base de datos del sistema, se definen dos transformaciones ETL, una para cada formato normalizado, que siguen un flujo similar al siguiente:

1. Leer los datos de la planilla fuente.
2. Filtrar aquellas filas que no correspondan agregar, por ejemplo si no tienen horas registradas.
3. Agregar información estática que no se consume de la planilla, como por ejemplo los datos de la edición que se está cargando.
4. Distribuir los datos de cada fila en las tablas de la base de datos que corresponda, gestionando las claves foráneas para todas las referencias a otros elementos.

Cada transformación agrega los pasos necesarios acordes a la estructura del formato fuente que le corresponda. Las transformaciones implementadas se pueden observar en las figuras 18 y 19.

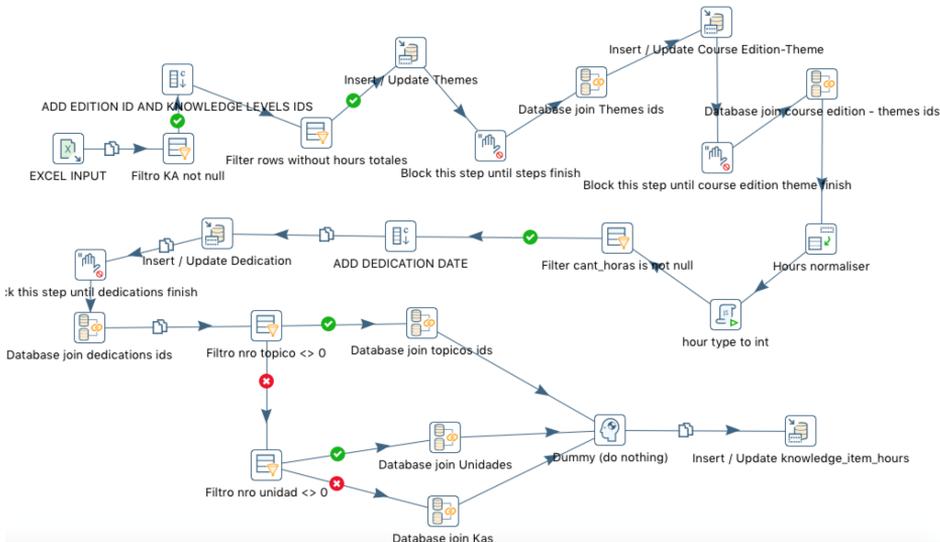


Figura 18: Transformación realizada para cargar planillas con registro directo sobre tópicos.

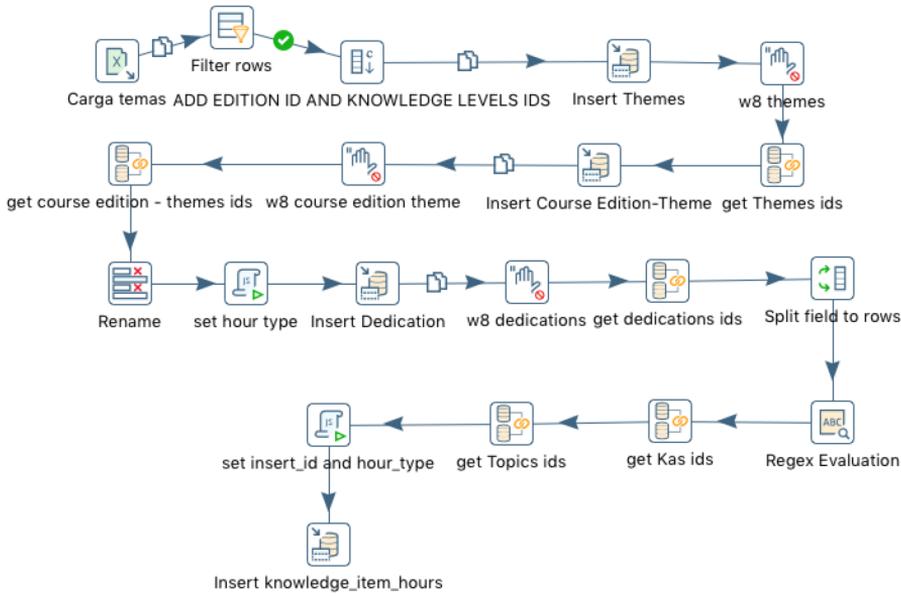


Figura 19: Transformación realizada para cargar planillas con detalle de temas dictados.

7.3.1 Dificultades encontradas

El primer problema enfrentado en el proceso de carga, es contemplar las diferencias que tienen planillas de un mismo tipo entre sí. Por ejemplo, para las ediciones de curso 2012 y 2013 se presentan diversas variaciones que se mencionan a continuación.

Algunas planillas separan los registros en diferentes pestañas según el tipo de hora, mientras que el resto separan los tipos de hora en columnas. Para las planillas que separan los registros en pestañas, el formato de las pestañas difieren entre sí.

Dos diferencias generales que se encuentran son la variedad en la nomenclatura de las columnas, y en la forma en la que se modela la correspondencia con los tópicos del CBOK. Existen variaciones considerables de una planilla a otra, y dentro de una misma planilla. En algunos casos, para cada tópico se indica su área, y unidad, mientras que en otros casos se indica únicamente el tópico. En otro caso puntual, se indica el área de conocimiento, y luego para cada área se mencionan los tópicos a los que se corresponde el registro.

A su vez, muchos registros presentan la cantidad de horas asociadas a cada tópico entre paréntesis, mientras que otros registros no. En algunos casos es porque se corresponden completamente a un tópico pero no es lo que sucede en todos los casos.

Otras diferencias encontradas, es que en algunos campos la correspondencia con cada tópico comienza con un asterisco, en otros casos comienza con la sigla "KA", mientras que en el resto de los casos directamente se nombra el elemento.

Las diferencias presentadas anteriormente imposibilitan la automatización directa de las planillas. Para enfrentar este problema, se realiza el esfuerzo manual de llevarlas al formato estándar definido.

Otra dificultad encontrada, es no contar con identificadores numéricos para los elementos del CBOK o temas dictados en las planillas. Esto obliga a encontrar las correspondencias por comparación de texto. En este sentido, se encuentran varios errores de escritura, faltas de ortografía y abreviaciones que conllevan a un esfuerzo manual mayor al normalizarlas.

Por último, en la etapa de carga se detectan varias inconsistencias en los datos. La carga se realiza a partir de las planillas de registro para cada edición, y se valida con respecto a la planilla consolidada. En este sentido, se detectan inconsistencias entre los registros de cada edición, y lo sumariado en la planilla consolidada. Las inconsistencias detectadas son consultadas con los tutores, que determinan los cambios a realizar. Estos cambios se agregan a las planillas normalizadas, y se vuelven a correr las transformaciones ETL para trasladar los arreglos a la base de datos.

## GESTIÓN DEL PROYECTO

---

El presente capítulo se enfoca en mostrar las actividades relacionadas con la gestión del proyecto. En la sección 8.1 se presentan las principales actividades que se deben llevar a cabo para la realización del proyecto, y las fases en las que se organizan estas actividades. Luego, se muestra cómo se planificaron en el tiempo, y cómo fueron llevadas a cabo realmente.

La sección 8.2 explica el proceso utilizado para la construcción del sistema y la generación de documentos. Además, se muestra el proceso mediante el cual se gestionan las incidencias reportadas para su correcto procesamiento.

En la sección 8.3 se presenta el trabajo realizado en relación a la gestión de riesgos. Por un lado, se presentan los riesgos detectados al inicio del proyecto, y cómo los mismos fueron mitigados. Luego, se explican otros riesgos que no fueron tomados en cuenta, o que no fueron mitigados correctamente y las consecuencias que generaron.

En la sección 8.4 se explica cómo se llevó a cabo la gestión de la configuración, presentando las herramientas y flujos utilizados para la correcta trazabilidad de los entregables. Por último, en la sección 8.5 se presentan las actividades realizadas y el plan de pruebas para asegurar la calidad del sistema construido y de los documentos generados.

### 8.1 PLANIFICACIÓN

Para cumplir con los objetivos del proyecto, se comienza por identificar las principales actividades a llevar a cabo. En la figura 20 se muestra la organización de las actividades en fases, y cómo éstas se planifican en el tiempo.

Luego, durante el transcurso del proyecto, surgen actividades que no estaban previstas en un inicio y se lleva a cabo una instancia de re-planificación, en la que algunas funcionalidades son removidas del alcance del proyecto. Entre estas funcionalidades, se encuentran las alta, baja, modificación y consulta (ABMC) de funcionalidades de baja prioridad, así como la incorporación del módulo de encuestas.

En la figura 21 se muestra gráficamente la ejecución real del proyecto. Esto es, las actividades que se llevan a cabo realmente, su separación en fases, y su distribución en el tiempo.

Fase	Actividad	03 /16	04 /16	05 /16	06 /16	07 /16	08 /16	09 /16	10 /16	11 /16	12 /16
Fase I	Lectura de documentación existente	■									
Fase I	Aprendizaje latex	■									
Fase I	Documentación: Análisis de la problemática planteada		■								
Fase I	Investigación herramientas de encuestas		■								
Fase I	Documentación: Descripción de la realidad		■								
Fase I	Requisitos: Definición y documentación		■	■							
Fase I	MER		■	■							
Fase II	Diagramas de secuencia CU críticos			■							
Fase II	Validación interfaces: Diseño de UI			■							
Fase II	Investigación: deploy en la nube			■							
Fase II	Documento arquitectura			■	■						
Fase II	Investigación de tecnologías e impacto en la arquitectura			■	■						
Fase II	Metodología de trabajo, gestión incidencias, plan de pruebas			■	■						
Fase II	Crear ambiente de trabajo y dejarlo disponible en la nube			■	■						
Fase II	Construcción: Desarrollo de casos de usos críticos					■	■	■	■		
Fase II	Back-end					■	■				
Fase II	Front-end					■	■				
Fase II	Funcionalidades restantes							■	■		
Fase III	Carga de datos									■	
Fase IV	Incorporación módulo encuestas										■
Fase V	Documentación final									■	■
Fase V	Documentación manual de usuario									■	■
Fase V	Manual de Instalación										■

Figura 20: Planificación inicial del proyecto.



Figura 21: Ejecución real del proyecto.

### 8.1.1 Fases del proyecto

La primera fase tiene por objetivo familiarizarse con la realidad sobre la que actúa el sistema y modelarla de forma genérica. La generalización debe ajustarse a soportar distintas carreras, cuerpos de conocimiento y configuraciones para llevar a cabo la evaluación.

Para lograr los objetivos, la principal actividad de esta fase es el aprendizaje. Se debe comprender cómo se compone la realidad académica y qué es lo que se debe tener en cuenta al momento de evaluar una carrera. Por otro lado, se debe aprender sobre los cuerpos de conocimiento y cómo se puede llevar a cabo la evaluación del cubrimiento temático de una carrera con respecto a un cuerpo.

Otra actividad que se debe llevar a cabo en esta fase es analizar los datos existentes de la evaluación del cubrimiento temático de la EIS. Dado que el sistema debe contar con estos datos, se debe conocer el formato, estructura, calidad y cantidad de esta información. Considerar los datos en la primera fase del proyecto permite lograr un modelado de la base de datos que los soporte, así como planificar la migración.

Luego de entender cómo funciona la realidad y conocer los datos, como parte de esta fase se releva qué es lo que se espera del sistema. Las salidas de esta fase son el documento de requisitos detallando las funcionalidades y características del sistema a construir, el modelo entidad-relación (MER) que plasma el diseño de la base de datos, y la planificación inicial del trabajo a realizar.

En segunda instancia, la fase II consta de la creación la herramienta para dar soporte a la evaluación de cubrimiento temático. Esta fase implica las actividades de investigación tecnológica, diseño, construcción, pruebas, y puesta en producción del software desarrollado.

Al comienzo de esta fase se realiza una búsqueda de alternativas existentes. Esto es, investigar herramientas existentes que puedan ser utilizadas o sirvan de guía para el desarrollo de la parte del cubrimiento temático y la evaluación a través de encuestas.

Otra parte fundamental de esta fase es la investigación y selección de tecnologías a utilizar. Esta actividad tiene como objetivo colmar el interés del equipo en aprender tecnologías y herramientas a las que apunta la industria, así como asegurar que éstas cumplen con los requisitos del sistema. Dentro de esta fase se incluyen actividades de realización de prototipos que logren dichos objetivos.

Además de las actividades relacionadas a la investigación previa, se realizan documentos que acompañan el proceso de construcción y validación:

- Diagramas de secuencia.

- Diseño y maquetado de UI.
- Documento de arquitectura.
- Documento de metodología de trabajo, gestión de incidencias y plan de pruebas.
- Manual de usuario para las funcionalidades creadas.

Por último, esta fase incluye las iteraciones para realizar la construcción, esto es, realizar el código, las pruebas y la validación del sistema.

En paralelo a la construcción de la herramienta, se realiza la fase III, correspondiente a la carga de datos. La fase III tiene por objetivo incorporar los datos generados al evaluar la EIS con respecto al CBOK desde el año 2012 hasta el 2016 a la base de datos del sistema. Las principales actividades de esta fase son:

- Conocer la estructura con la que se almacenan los datos en las planillas Excel.
- Identificar dificultades.
- Evaluar posibles alternativas para realizar la migración.
- Migrar los datos.
- Validar los datos migrados, y corregirlos cuando sea necesario.

Una vez que se cuenta con el sistema construido y validado, y la carga de datos completada, se considera una última fase, cuyo objetivo es la preparación de los entregables del proyecto. Éstos son: la creación del presente documento, el manual de instalación, una máquina virtual con la solución desarrollada, la base de datos cargada y un instructivo de su esquema. Por último, de forma opcional se entrega un entorno web de producción en funcionamiento.

### 8.1.2 *Ejecución real respecto a la planificación inicial*

Existen dos grandes diferencias entre lo que se había planificado inicialmente para el proyecto y lo que se lleva a cabo realmente: el tiempo asociado a cada actividad y el alcance del proyecto.

Las fases se extendieron de tres a cinco veces más del tiempo estimado inicialmente. Se identifican algunos factores que influyeron en la mala estimación.

- Falta de experiencia en estimación de proyectos. En consecuencia, se subestima el tamaño del proyecto, y el tiempo disponible.

- Falta de experiencia en desarrollo y en las tecnologías seleccionadas. Se dedica gran esfuerzo en tiempo al aprendizaje de tecnologías y herramientas, que aportó a los objetivos personales del equipo pero no así a los del proyecto.
- Planificación optimista, que busca alcanzar todos los objetivos en el lapso de tiempo disponible, sin considerar la factibilidad de realizar las distintas actividades en el tiempo asignado.
- Subestimación del tiempo dedicado a documentación. En la planificación inicial no se conocían todos los documentos que iban a ser demandados, por lo que no se consideró el tiempo asociado a su realización. A su vez, cada documento oficia de validación para las actividades, por lo que el nivel de detalle exigido fue mayor al esperado inicialmente. Esto, sumado a la falta de experiencia en redacción formal de documentos conllevó a que cada uno requiera varias versiones.
- Subestimación de la cantidad y calidad de datos a migrar.
- Esfuerzo en avances de funcionalidades que quedaron por fuera del alcance del proyecto. Específicamente, al inicio de la fase de construcción, se avanzó en código para dejar funcionalidades planteadas que finalmente no fueron entregadas. Este es el caso de la consideración de los distintos roles para el manejo de vistas y los ABMC de perfiles de configuración y cuerpos de conocimiento. El avance en la autorización, se realiza en el comienzo de la construcción, como forma de mitigar el riesgo técnico que conlleva, lo cual a pesar de no haber sido entregado, se considera que tiene valor ya que el sistema es fácilmente extensible a soportar los distintos roles. Por el contrario, el avance en los ABMCs de perfiles y cuerpos realizaron con el objetivo de dejar planteada la navegación del sistema, así como parte del aprendizaje de la tecnología dejando todos los ABMCs planteados, pero sin tener las funcionalidades completas, es código inutilizado que fue removido de la solución, por lo que se considera trabajo desperdiciado.

Se logra entregar un sistema web que soporta la evaluación del cubrimiento temático de la EIS con respecto al CBOK, que cuenta con todos los datos de evaluación disponibles hasta el momento y que además es lo suficientemente genérico como para ser usado con otros planes de estudio y cuerpos de conocimiento de estructura jerárquica.

De esta forma, se considera que el sistema cumple con los principales objetivos propuestos, dejando uno por fuera: el soporte a la evaluación de carreras a

través de encuestas. Al inicio del proyecto, se realizó la investigación pertinente a la incorporación de esta funcionalidad. Se encontraron una gran variedad de herramientas para realizar encuestas disponibles en la web, que ofrecen además servicios de integración. Aunque no se llegó a desarrollar esta integración, se considera que el sistema es fácilmente extensible a agregarla, debido a la arquitectura *REST* con la que se plantea el *back-end*.

Además del manejo de encuestas, otras funcionalidades consideradas de menor prioridad quedaron por fuera del sistema por razones de tiempo. En las tablas 9 y 10 se listan respectivamente las funcionalidades que quedaron por fuera del alcance del proyecto y las que sí fueron incluidas en la solución final.

Funcionalidades pendientes
Alta, modificación y consulta de cuerpos de conocimiento.
Alta, modificación y consulta de perfiles de configuración.
Modificar ponderaciones.
Manejo de módulo de encuestas para cruzamiento de datos.
Aplicar permisos de autorización al resto de las funcionalidades.

Cuadro 9: Funcionalidades pendientes.

Funcionalidades incluidas en la solución
Autenticación
Alta, modificación y consulta de carreras
Alta, modificación y consulta de planes de estudio
Alta, modificación, baja y consulta de implementaciones de un plan de estudio
Alta, modificación, baja y consulta de ejecuciones de una implementación
Alta, modificación y consulta de cursos
Alta, modificación y consulta de ediciones
Alta edición bajo gestión de la configuración (reutilizar registros y mapeos)
Importar registro de dedicaciones desde planilla Excel
Registrar, editar, y eliminar dedicaciones
Exportación de registro de dedicaciones a planilla Excel
Asignar dedicaciones a ítems de un cuerpo
Exportación de dedicaciones con mapeos a los ítems de un cuerpo de conocimiento
Personalizar las consultas, utilizando perfiles de configuración
Visualización consulta esfuerzo temático
Exportación gráfica y tabular de la consulta de esfuerzo temático
Visualización consulta balance temático
Exportación gráfica y tabular de la consulta de balance temático
Visualización consulta cubrimiento temático
Exportación gráfica y tabular de la consulta de cubrimiento temático
Consultas anteriores sobre ejecuciones simuladas
Visualizar detalle de registro por ítem de conocimiento
Exportación tabular de los registros asociados a un ítem de conocimiento
Modificar configuración de la cuenta de usuario
Manual de usuario
Alta, modificación y consulta de usuarios desde un rol super-admin

Cuadro 10: Funcionalidades que incluye el sistema desarrollado.

## 8.2 PROCESO UTILIZADO

El proyecto se realiza siguiendo un proceso de desarrollo iterativo incremental, con liberaciones al término de cada iteración. Para cada iteración se define un alcance y de acuerdo al trabajo a realizar se estima su duración. Este proceso aplica tanto para las actividades de construcción de la herramienta, así como para la creación de otros artefactos entregables como ser la documentación previa, y final. Cada iteración incluye las siguientes etapas:

**IDENTIFICACIÓN DE TAREAS** Esta etapa se realiza al comienzo de cada iteración, y consiste en separar las actividades a realizar en un conjunto de tareas cuyo alcance quede bien definido, ordenadas según su prioridad y el o los criterios de aceptación que se deben cumplir para considerar cada tarea finalizada.

**EJECUCIÓN** Los miembros del equipo toman responsabilidad sobre las tareas, y llevan a cabo la realización de la misma. Esto puede implicar una investigación, la realización de un documento, o el desarrollo de una funcionalidad. El responsable de la actividad debe asegurarse de que la tarea cumple con los criterios de aceptación definidos en el primer paso. Esto puede ser, llegar a una conclusión y documentarla en el caso de las investigaciones, incluir los puntos establecidos por el equipo en el caso de la realización de un documento, o cumplir con los resultados esperados en el caso de la implementación de una funcionalidad.

**PRUEBAS** El resto del equipo valida el trabajo realizado para cada tarea. Si se encuentran incidencias, se vuelve al paso anterior o si existen motivos para no atacarlos en la iteración, se reportan como errores conocidos para ser atacados posteriormente.

**LIBERACIÓN** Una vez culminadas las etapas anteriores para todas las tareas de la iteración, se realiza una revisión de los elementos generados y se entregan a los tutores para su revisión.

**REPORTE DE INCIDENCIAS** En esta etapa, los tutores reportan incidencias sobre los entregables de la iteración, y las mismas serán atacadas en posteriores iteraciones siguiendo el mismo proceso.

El proyecto consta de cuatro fases, explicadas previamente en la sección 8.1. A priori, no se definen la cantidad de iteraciones que implica cada fase, sino que surgen a demanda según las actividades pendientes y las incidencias reportadas.

## 8.2.1 Gestión de incidencias

Se define un proceso para llevar a cabo la gestión de incidencias en lo que refiere a la construcción del sistema, que asegura la correcta comunicación y ciclo de vida de las mismas. El proceso consta en mover las incidencias entre distintos estados manteniendo estas transiciones visibles a todos los miembros del equipo y tutores.

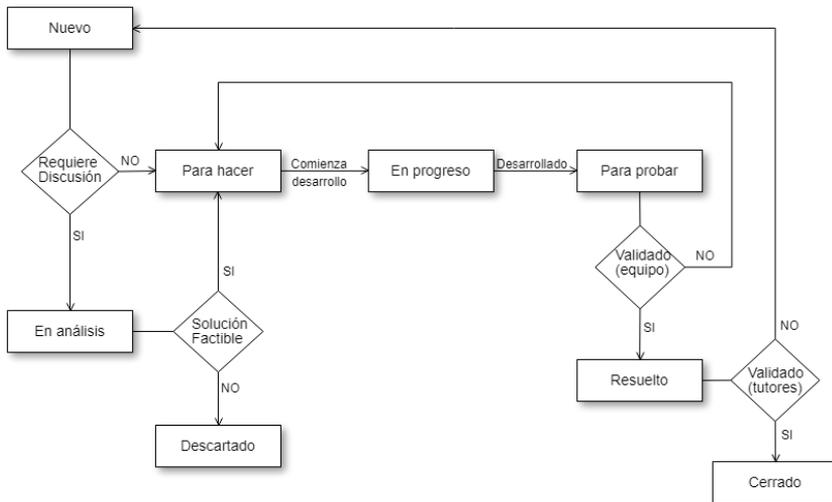


Figura 22: Diagrama de flujo de los estados que atraviesan las incidencias.

La figura 22 muestra un diagrama de flujo con los estados por los que pasan los incidentes reportados. A continuación se explican los distintos estados y qué motiva el cambio de un estado a otro.

1. **Nuevo:** Incidencias creadas. Tanto los miembros del equipo como los tutores reportan incidencias que surjan en las etapas de prueba del sistema. Cada incidencia lleva asociada una descripción, y los archivos o imágenes adjuntos necesarios para su correcta comprensión y reproducción. Además, las incidencias se etiquetan según su nivel de criticidad siguiendo el siguiente criterio:
  - **Crítica:** Incidencia que pone en riesgo el uso del sistema, y debe resolverse de forma inmediata.
  - **Alta:** Incidencia requerida para el correcto funcionamiento del sistema.

- Media: Incidencia que aporta valor a la calidad del sistema pero que no imposibilita el correcto funcionamiento del mismo.
  - Baja: Mejoras a realizar.
2. **En análisis:** Estado en el que se encuentran las incidencias que requieren de un intercambio de comentarios entre el equipo y los tutores para definir la solución a realizar. Puede ser debido a que haya que elegir entre varias opciones, o porque la solución pedida requiera demasiado esfuerzo y deba buscarse una alternativa.
  3. **Para hacer:** Estado en el que se encuentran todas las incidencias cuya solución está definida, pero aún no se comenzó a implementar.
  4. **En progreso:** Estado en el que se encuentran las incidencias cuando están siendo atacadas, esto es, que uno o más miembros del equipo están desarrollando la solución.
  5. **Para probar:** Estado en el que se ubica una incidencia cuando el responsable de desarrollar la solución considera que está resuelta, y debe ser probada por el resto del equipo.
  6. **Resuelto:** Incidencias cuya solución han pasado las pruebas definidas por el equipo y se encuentran liberadas para ser verificadas por los tutores.
  7. **Cerrado:** Estado en el que se encuentran las incidencias que ya no están presentes porque su solución fue validada.
  8. **Descartado:** Incidencias que no requieren una solución. Esta situación se puede dar cuando lo reportado forma parte de funcionalidades que aún no han sido desarrolladas, su solución no es factible, no se han podido reproducir o se encuentran duplicadas, es decir, están incluidas en otras incidencias.

### 8.3 GESTIÓN DE RIESGOS

En esta sección se presenta el trabajo realizado en torno a la gestión de riesgos. Se mencionan los distintos riesgos identificados al comienzo del proyecto, y para éstos se explica el impacto que generan en caso de que ocurran, la probabilidad de ocurrencia, el plan con el que se mitigan, y qué sucede con ellos durante el proyecto.

El primer riesgo identificado, es que el sistema no contemple de forma correcta las necesidades del usuario. Este riesgo puede aparecer en cualquier punto

del proyecto, sobretodo en etapas tardías cuando el sistema se deja disponible para los usuarios, lo cual implica un riesgo grave, pues en ese momento el impacto de realizar modificaciones es muy alto. Existen varios factores por los que se considera probable que ocurra este riesgo. Por un lado, los conceptos asociados a la realidad sobre la que actúa el sistema son nuevos para los integrantes del equipo, deben aprenderse de cero y ser percibidos como los usuarios esperan. Por otra parte, no existe ningún sistema que haya sido utilizado previamente y pueda tenerse como referencia para alinear expectativas. Incorporar un sistema de *software* a un proceso ya definido que se realiza de forma manual genera cambios difíciles de prever y por lo tanto difíciles de validar en la forma de trabajo. Por último, es riesgoso conseguir un sistema genérico, ya que la generalización se crea a partir del uso específico con el que se plantea el método de cubrimiento temático y a priori no se conocen otras realidades académicas para validarla. La forma de reducir al máximo el impacto de este riesgo es identificar posibles diferencias conceptuales y en los requerimientos entre lo que entiende el equipo y lo que esperan los tutores de forma temprana. Para ello, se realizan diversas validaciones en todas las etapas del proyecto. Por un lado, se realizan dos documentos para validar la correcta comprensión de la realidad: descripción de la realidad y análisis de la solución. Luego se validan las necesidades del usuario a través de un documento con la especificación de requisitos relevados. Luego se presentan varios documentos con respecto al diseño de la solución: un modelo entidad relación que representa el modelado genérico de la realidad, documento de arquitectura, diagramas de secuencia, bosquejos de la interfaz de usuario de los casos de uso principales, documentos independientes explicando decisiones tomadas para realizar la generalización y cómo se personalizan las distintas configuraciones. Una vez validados todos estos documentos, se comienza con una construcción iterativa e incremental, que cuenta con liberaciones al término de cada iteración para poder obtener de forma temprana comentarios sobre las funcionalidades entregadas y poder planificar cambios sugeridos en próximas iteraciones. Las validaciones resultaron eficientes para mitigar el riesgo, ya que las correcciones surgen a medida que se va realizando cada actividad, siendo poco costosa su corrección. Como resultado la herramienta entregada contempla las necesidades requeridas por el usuario, aunque no se ha corroborado dentro del alcance del proyecto que la herramienta sirva para contemplar evaluaciones de otras carreras con respecto a otros cuerpos de conocimiento.

El siguiente riesgo identificado es seleccionar tecnologías no apropiadas para la construcción del sistema. Este riesgo puede resultar en varios problemas, como ser una curva de aprendizaje muy costosa, incompatibilidad entre las

tecnologías seleccionadas, o que éstas no cumplan con los requerimientos del sistema. En este sentido, por las distintas consecuencias que tiene, se considera que el impacto que puede causar una mala selección de tecnologías va de medio a alto. A su vez, la probabilidad de ocurrencia de este riesgo se considera alta, ya que el equipo no tiene experiencia suficiente en desarrollo de aplicaciones, ni en el uso de tecnologías actuales. Como plan para mitigar el riesgo, y reducir el impacto, se dedica tiempo al inicio del proyecto a la investigación de distintas tecnologías para poder seleccionar las adecuadas. Además del estudio teórico validado a través de un documento de selección de tecnologías, se realizaron prototipos en busca de evaluar la curva de aprendizaje y conocer el funcionamiento de cada una. Estos prototipos, incluyen la interacción entre distintas herramientas, como por ejemplo la librería de gráficas y la plataforma para dejar el prototipo en la nube. De esta forma, se logra mitigar el riesgo de que las tecnologías seleccionadas no cumplan con los requisitos del sistema, o no sean compatibles entre sí. A pesar del éxito logrado con respecto a evitar los riesgos identificados, durante el inicio del proyecto se enfrentaron problemas de compatibilidad con respecto a actualizaciones de las tecnologías. Como plan de contingencia a este riesgo que no estuvo mitigado, se toma la decisión de cambiar la selección inicial del *framework* seleccionado para el *front-end* por una librería cuya versión actual es más estable. A su vez, se opta por fijar las versiones utilizadas, dejándolas explícitamente indicadas para cada dependencia del proyecto, evitando su actualización automática.

El sistema utiliza módulos de terceros para resolver la importación de planillas Excel, la visualización y descarga de gráficas, y el manejo de datos a través de grillas con comportamiento similar a Excel. Se identifica como riesgo que los módulos de terceros utilizados no contemplen todos los requerimientos, generen problemas de rendimiento. Debido a que todos estos módulos influyen en casos de uso fundamentales en lo que refiere a la realidad sobre la que actúa el sistema, se considera que el impacto que puede tener un problema es muy alto. Se toman tres medidas para evitar o al menos reducir la ocurrencia de este riesgo. El primer paso es relevar los requisitos que debe cumplir el componente externo para colmar las expectativas de la aplicación. En segundo lugar, se investigan alternativas y se selecciona la que se adapte a las necesidades relevadas. Por último se identifican las funcionalidades que utilizan estos módulos como funcionalidades críticas y se desarrollan en la primera iteración de la construcción. De esta forma, se puede validar la arquitectura y la selección de éstos módulos de forma temprana. Como resultado de esta gestión, la incorporación de los módulos de gráficas e importación de planillas Excel resultó exitosa. En el caso de la librería seleccionada para modelar grillas resulta

exitosa en el caso de uso crítico de mapeo de temas a tópicos, pero al utilizarlo para el registro manual de dedicaciones algunas facilidades esperadas por el usuario se vieron limitadas. En este caso, se considera que el error fue de relevar en alto nivel que el módulo seleccionado debía cumplir un funcionamiento similar a Excel, sin validar los detalles de usabilidad con el usuario. Otro riesgo identificado es la imposibilidad de dejar disponible el sistema en un entorno web ya que la implantación del sistema en la nube presenta los siguientes requisitos incompatibles entre sí: plataforma gratuita, segura y con capacidad ilimitada. Este riesgo se hizo presente en el proyecto, y fue enfrentado a través de la priorización de los requisitos anteriores. Se priorizan los requisitos de costo y seguridad, sacrificando la capacidad de base de datos disponible. Luego, se investigan las distintas opciones disponibles. Dentro de la investigación, se contemplaron opciones como dejar el sistema disponible en un servidor dentro de la Facultad, conseguir licencias en IBM para utilizar Bluemix y utilizar plataformas comerciales como Heroku o Openshift. Las dos primeras opciones no fueron factibles por razones externas al proyecto, por lo que se elige trabajar con el plan no pago de la plataforma comercial que ofrece mejores condiciones. De este modo, resultó posible utilizar un entorno web de forma gratuita y segura durante el desarrollo del sistema para realizar las distintas etapas de pruebas y validación. Eventualmente para el entorno de producción se puede optar por extender el plan, y de esta forma la capacidad de la base de datos, a medida que el volumen de datos crezca con el uso del sistema. Por otra parte, para asegurar una entrega segura del sistema construido se agrega una máquina virtual con la solución desarrollada como parte de la entrega final.

En cuanto a la carga de datos en la base del sistema, se considera un riesgo que la estructura de la base de datos no soporte los datos a cargar. Si la carga de datos se realiza una vez finalizada la construcción del sistema el impacto de este riesgo es muy alto, pues puede resultar en la imposibilidad de agregar algunos datos al sistema, o en su defecto, requerir cambios en la estructura de la base y en el código de la aplicación que la consume. El esquema de la base de datos se diseñó teniendo en cuenta las planillas de datos disponibles, por lo que la probabilidad de ocurrencia es baja. De todos modos, debido al alto impacto que puede ocasionar un cambio en la base de datos se planifica una estrategia para reducirlo. Esto se logra detectando de forma temprana los ajustes que se deban realizar sobre el esquema diseñado, para que tanto el código desarrollado como las transformaciones en el proceso de migración los contemplen lo antes posible. En este sentido, se opta por comenzar el trabajo de migración de datos en paralelo con la construcción del sistema. Esta planificación resulta exitosa, ya que efectivamente fueron requeridas algunas modificaciones en la

estructura diseñada motivadas tanto por la migración de datos como por la implementación de las funcionalidades. En todos los casos se coordinaron los cambios para ser contemplados desde estos dos puntos de vista con el menor perjuicio posible.

Un problema que no se identificó correctamente como riesgo es la subestimación del tamaño y tiempo de realización del proyecto. Se planifica el proyecto en base a ubicar todas las actividades identificadas en un año lectivo, que es el tiempo original estimado para un proyecto de grado. El error cometido es aceptar el alcance completo del proyecto en el tiempo disponible, sin considerar la factibilidad de cumplirlo. No sólo surgieron documentos y actividades que no estaban previstas durante el proyecto, sino que el tiempo original asignado a cada actividad resultó escaso en la mayoría de los casos. El no lograr cumplir con la planificación inicial es un riesgo que podría haber sido identificado y mitigado desde un principio. Surge en consecuencia a la falta de experiencia en estimación y la forma en la que se realiza la planificación inicial, y genera que el proyecto se haya retrasado en el plazo, y acotado su alcance.

#### 8.4 GESTIÓN DE LA CONFIGURACIÓN

Un aspecto de interés en la realización del proyecto, es mantener la correcta trazabilidad y versionado de los distintos entregables.

En lo que refiere a la documentación del proceso, cada documento se identifica con un nombre, y lleva adjunto un número de versión, el cual incrementa en cada iteración.

En lo que refiere al código, se mantiene un repositorio remoto con la última versión disponible, al que se van incorporando las nuevas funcionalidades. Para mantener este repositorio se utiliza la herramienta GitLab, que provee un mecanismo de control de versiones GIT<sup>49</sup>. La herramienta provee un acceso al historial de modificaciones, permitiendo restaurar versiones anteriores.

Con el objetivo de mantener la trazabilidad de las funcionalidades liberadas, y el aislamiento de los datos, se definen cuatro ambientes independientes para desarrollo, pruebas, pre-producción y producción. El ambiente de desarrollo es un entorno orientado exclusivamente al desarrollo, diseño e integración de nuevas funcionalidades. El entorno de pruebas es dónde el equipo de desarrollo prueba las nuevas funcionalidades y su integración con el resto del sistema utilizando datos ficticios. Pre-producción es el ambiente en el que el usuario prueba y valida el avance del sistema utilizando datos ficticios. Por último, producción es el entorno final en el se trabaja con los datos reales de la organización.

---

49 <https://git-scm.com/>

Una vez que cada funcionalidad esta implementada y probada en el entorno de desarrollo, se incorpora al ambiente de pruebas, donde se prueba su integración al resto del sistema como se explica en la sección 8.2. Cuando este proceso se cumple para todas las funcionalidades de una iteración, y sus respectivas pruebas resultan satisfactorias, los cambios se trasladan al entorno de pre-producción dónde queda disponible a los tutores para que realicen sus pruebas. Al finalizar la construcción, el sistema desarrollado se traslada al ambiente de producción para que quede habilitado su uso con datos reales.

## 8.5 GESTIÓN DE LA CALIDAD

Para asegurar que la documentación y el software creados a lo largo de este proyecto cumplen con un nivel aceptable de calidad, se define un conjunto de pruebas a realizarse sobre los distintos artefactos generados.

**REVISIÓN DE DOCUMENTOS** Todos los documentos se revisan de forma cruzada. Es decir, todos los integrantes del equipo que no hayan participado en la redacción del documento lo revisan y envían sus comentarios al responsable de la redacción. La revisión consiste en chequear que el documento es coherente con el trabajo realizado, y no contenga errores de sintaxis.

**REVISIÓN DE CÓDIGO** Para funcionalidades críticas en cuanto a *performance*, complejidad, o tamaño se realizan revisiones de código cruzadas. La revisión de código debe validar las siguientes cualidades: código comprensible, utilización de comentarios, resolución óptima de los problemas evitando procesamiento innecesario, código separado en módulos que favorezca su re-utilización, nomenclatura consistente, aplicación de buenas prácticas de programación, uso de convenciones y correcta sintaxis.

**PRUEBAS DE INTEGRACIÓN Y CRITERIOS DE ACEPTACIÓN** Al comienzo de cada iteración se definen los criterios de aceptación de cada tarea y pruebas que verifiquen el flujo normal de cada funcionalidad, los escenarios alternativos más comunes y casos borde. Al finalizar cada iteración, se prueban todos los escenarios definidos de forma manual, para asegurar que las funcionalidades liberadas cumplan con los resultados esperados. La ejecución de las pruebas se documenta y forma parte de los entregables en cada iteración.

**COMPATIBILIDAD CON NAVEGADORES** Al desarrollar una funcionalidad nueva, el desarrollador prueba al menos el escenario principal en todos los navegadores a soportar.

**PRUEBA DE INTERFAZ DE USUARIO** Se realizan revisiones cruzadas que validan características que debe cumplir la interfaz como ser: uso de convenciones habituales utilizadas en sitios web, vocabulario representativo de la realidad en la que se basa el sistema, acceso a todas las funcionalidades de forma intuitiva, indicar si hay un proceso en marcha los pasos que faltan para terminar, mostrar en qué lugar de la aplicación se encuentra el usuario, prevenir errores mediante validaciones y ayudas, verificar que si el usuario se equivoca tiene mecanismos para recuperarse del error y corroborar que todo lo que aparece en la interfaz tiene un sentido.

**PRUEBA DE SEGURIDAD Y CONTROL DE ACCESO** En cada liberación, se chequea que únicamente usuarios registrados y con permiso sobre las funcionalidades pueden acceder a ellas.

**PRUEBAS EXPLORATORIAS** En cada liberación, se realizan pruebas exploratorias sobre el ambiente de pre-producción, para asegurar que la liberación fue exitosa, y para dar lugar a la detección de errores aleatorios.

**PRUEBAS DE REGRESIÓN** En cada liberación, se realizan pruebas del sistema, verificando el correcto funcionamiento del flujo normal de cada funcionalidad liberada hasta el momento y los escenarios alternativos más comunes. De esta forma, se valida que las nuevas funcionalidades incorporadas no generan problemas inesperados.



## CONCLUSIONES DEL TRABAJO REALIZADO

---

En el marco de este proyecto se creó una plataforma web, que sirve como herramienta para realizar la evaluación de cubrimiento temático de carreras con respecto a cuerpos de conocimiento. La herramienta entregada soporta el método definido para evaluar la EIS con respecto al CBOK y permite realizar ajustes sobre los criterios de evaluación con el objetivo de adaptarse a las necesidades de otras carreras y otros cuerpos de conocimiento de estructura jerárquica. De este modo, se cumple con el objetivo de dejar disponible un sistema que brinda soporte a la evaluación de carreras desde el punto de vista del cubrimiento temático, modelado de forma genérica para brindar flexibilidad a distintos criterios de evaluación.

En lo que respecta a la evaluación de la EIS a través del uso de la herramienta, se realizó la carga de todos los registros de dedicaciones disponibles hasta el momento en planillas electrónicas. Esta carga permite tener disponible en la base de datos del sistema toda la información necesaria para evaluar el cubrimiento temático de la EIS con respecto al CBOK. A pesar de haber utilizado los datos cargados para realizar distintas consultas de cubrimiento, balance y esfuerzo temático sobre varias ejecuciones de la EIS, estas actividades se enfocaron en la validación del sistema y no en el objetivo inicial, que buscaba generar nueva información.

En un principio, se esperaba que el sistema incluyera la evaluación de los resultados esperados de un plan de estudios a través del uso de encuestas. Además, se deseaba evaluar la EIS teniendo en cuenta el cruzamiento de información de este enfoque, con los resultados obtenidos del cubrimiento temático. En etapas tempranas del proyecto, se evaluaron alternativas para la incorporación de un módulo de encuestas, y se planteó una arquitectura extensible a soportarlas. El esfuerzo que llevó el desarrollo de la herramienta generó la necesidad de priorizar y re-planificar las actividades restantes en acuerdo con los tutores, dejando la integración de las encuestas por fuera del alcance del proyecto. Por este mismo motivo, el objetivo de evaluar la EIS pasó a quedar obsoleto, ya que su sentido estaba en realizar el cruzamiento de la información recolectada en las encuestas con la información del cubrimiento temático.

Entre los aportes del proyecto, se destaca la creación de una herramienta de software que brinda soporte al método de cubrimiento temático. De hecho, no

se encontraron otros sistemas similares que pudieran ser utilizados para hacer este tipo de análisis, por lo que es la primera vez que se realiza un modelado de esta realidad, situación que llevó a tomar decisiones de generalización o abstracción que antes no habían sido necesarias. Como resultado se entrega un sistema en condiciones para ser utilizado en la evaluación de la EIS con respecto al CBOK, y que puede ser extendido para evaluar otras carreras con respecto a otros cuerpos de conocimiento. Además de facilitar las funcionalidades principales que se llevaban a cabo a través de planillas electrónicas, como ser el registro de dedicaciones, el mapeo al cuerpo, y la realización de las consultas de esfuerzo, balance y cubrimiento, el sistema agrega nuevas funcionalidades. Entre estos aportes se encuentra la facilidad para visualizar si el mapeo de las ediciones de una ejecución se encuentra completo para un cuerpo de conocimiento y así evitar realizar consultas sobre datos parcialmente mapeados. Por otro lado, se ofrece la opción de visualizar qué registros de tiempo influyen en el cubrimiento de un elemento específico del cuerpo, y de qué ediciones provienen estas dedicaciones. En este sentido, es fácil visualizar qué modificaciones serían necesarias para mejorar los resultados de evaluación. Por otra parte, la flexibilidad del sistema para personalizar los criterios de evaluación y la posibilidad de simular ejecuciones abren muchas posibilidades a probar alternativas y plantear mejoras a la carrera evaluada. Por último, el hecho de que el registro de dedicaciones pueda ser importado desde planillas electrónicas, y que todos los resultados puedan ser exportados de forma tabular y gráfica ofrece la posibilidad de conservar las ventajas que brindan las hojas de cálculo.

Dentro del trabajo realizado, se considera que el proyecto tiene varios puntos fuertes. En primer lugar, el trabajo realizado para dejar disponibles todos los datos de las planillas electrónicas en la base de datos relacional que consume el sistema. En el proceso de migración de los datos, se detectaron errores e inconsistencias que fueron corregidos, mejorando la calidad de la información. De ahora en más, los datos procedentes de la evaluación de la EIS desde el inicio de su dictado, se encuentran estructurados en la base de datos, y son accesibles a través de la interfaz del sistema construido o directamente desde consultas a la base de datos, solucionando los problemas adyacentes al creciente volumen de datos.

Por otra parte, se considera un punto fuerte haber dejado el sistema disponible en Heroku. Además de permitir un proceso iterativo con liberaciones intermedias poco costosas, permite mantener ambientes independientes de desarrollo, pre-producción y producción. El ambiente de producción se entrega con la base de datos inicializada con los datos reales de las planillas. Aunque el plan gratuito con el que se entrega el sistema ofrece limitantes en la capacidad de la

base de datos, y tiene restricciones de disponibilidad, es muy sencillo aumentar las condiciones de suscripción a través de la interfaz que provee Heroku.

Otro punto fuerte a destacar es la mantenibilidad del sistema construido. Se considera que es un sistema fácil de mantener por personas que no necesariamente hayan estado involucradas en la construcción, principalmente por tres razones: las tecnologías utilizadas, documentación actualizada, y estructura del código. El sistema se desarrolló utilizando lenguajes conocidos y estables como son Java y Javascript, además de incorporar herramientas reconocidas en la industria como son Spring y React, que hacen el proyecto más atractivo para encontrar interesados en mantenerlo. Por otra parte, se cuenta con documentación actualizada de todas las etapas del proceso. Se entrega acceso a un directorio en Google Drive con la última versión de todos los documentos generados a lo largo del proyecto. Esta documentación sirve de guía para toda aquella persona que se involucre en el proyecto. Por último, el código se encuentra estructurado siguiendo prácticas conocidas para las tecnologías elegidas, manteniendo el código separado en componentes reutilizables. Este aspecto es indispensable para que se puedan agregar nuevas funcionalidades o arreglar errores que puedan aparecer.

El último punto a destacar, es que dentro del marco del proyecto se pudieron contemplar los intereses personales del equipo en generar experiencia de desarrollo de aplicaciones web, y aprender tecnologías de uso en el mercado laboral actual. Este enfoque en lo tecnológico generó complejidad y puso en riesgo el alcance del proyecto, pero también brindó recompensas en lo que refiere a la motivación y experiencia profesional.

## 9.1 LECCIONES APRENDIDAS

Este proyecto tiene tanto fortalezas como debilidades, y se considera interesante dedicar esta sección a analizar las decisiones que resultaron exitosas, y los aspectos que se podrían haber atacado de mejor manera.

Entre los puntos fuertes, se destaca que durante todas las etapas del proyecto se hicieron entregas, las cuales permitieron validar continuamente el sistema y desde sus inicios. No sólo la construcción fue iterativa e incremental con liberaciones al término de cada iteración, sino que previamente se realizaron varios documentos con el análisis de la realidad y la solución los cuales también fueron validados. Las validaciones sirvieron para detectar errores e inconsistencias que se debían arreglar, además de dar lugar a una comunicación más fluida con los tutores, que también son usuarios, sobre las decisiones tomadas. Poner a prueba el sistema con quienes lo van a usar desde las primeras funcionalidades

construidas permite alinear las expectativas de los distintos interesados, lo cual es necesario, ya que incorporar una herramienta de software a una forma de trabajo existente hace que esta realidad se modifique. Además, detectar incidencias y mejoras de forma temprana resulta imprescindible para poder atacarlas y obtener un proyecto exitoso.

Otra decisión acertada, fue la de no postergar la fase de carga de datos hacia el final de la construcción. Por un lado, ambas etapas se extendieron en el tiempo, por lo que posiblemente durante la construcción se hubiese seguido avanzando en funcionalidades menos prioritarias que la carga de datos, y luego esta actividad se hubiese visto comprometida en el alcance del proyecto. Por otra parte, y más importante, durante la migración de datos se detectaron correcciones a realizar en el esquema de base de datos planteado. Estos cambios afectan directamente al sistema construido, por lo que resultaron menos costosos los cambios en tiempo de construcción, que si se hubiesen realizado después de tener el sistema terminado y entregado.

Dentro de las carencias del proyecto, se considera que la forma en la que se trabajó podría haber sido mejor. A pesar de haber priorizado las funcionalidades a desarrollar, se realizaron actividades de análisis y diseño a bajo nivel de funcionalidades que resultaron quedar por fuera del alcance del proyecto, y que no eran tan prioritarias. En algunos casos, estos análisis dan certeza de que el sistema es extensible, como es el caso del manejo de roles y encuestas, pero en otros casos, se considera que el tiempo dedicado a estas actividades no se ve reflejado en el proyecto entregado. Si el proyecto se hiciera de nuevo, se consideraría disminuir el esfuerzo dedicado a iterar sobre algunas actividades, por ejemplo el refinamiento de documentos intermedios. Es decir, los documentos deben servir para generar entendimiento entre los tutores y el equipo, pero para algunos casos no fue valioso iterar en correcciones para perfeccionarlos, ya que no formaron parte de los entregables.

En esta misma línea, se considera que para algunos análisis, resulta más difícil validarlos a través de su explicación por escrito, que si por ejemplo se hubiesen prototipado. Esto queda en evidencia al momento de construcción, pues se detectaron inconsistencias en los bosquejos de interfaz de usuario, diagramas de secuencia, y modelado de perfiles para el esfuerzo temático que ya habían sido validados a través de documentos. Esto hace reafirmar la importancia de poner en tela de juicio el nivel de detalle que debe tener el análisis inicial, y enfocarse en que todas las actividades que se realizan generen valor, teniendo en cuenta que el nivel de detalle se puede refinar en las sucesivas iteraciones.

En cuanto a la priorización de tareas, hay dos aspectos en los que se podría haber trabajado mejor. En primer lugar, se comenzó a trabajar con un enfoque

horizontal, es decir, realizando las actividades base de todas las funcionalidades en lugar de enfocarse en el flujo completo de cada una. La segunda opción hubiese permitido finalizar antes la primera iteración, y evitar plantear trabajo que finalmente no fue utilizado. En segundo lugar, se decidió priorizar lo más riesgoso en lugar de lo más valioso. En este sentido, se planteó un sistema más complejo de lo que era necesario, por ejemplo, al incluir los manejos de perfiles de configuración.

Otro aspecto a mejorar es la repartición de tareas. Al comenzar las fases de construcción y carga de datos, se repartieron las tareas entre los miembros del equipo según si pertenecían a tareas de *back-end*, *front-end* o carga de datos. Este enfoque resultó en una menor curva de aprendizaje al inicio de la fase, pero generó dos inconvenientes. Por un lado, el desarrollo de cada funcionalidad requería más esfuerzo de coordinación y sincronización que si el *back-end* y el *front-end* los hiciera la misma persona. Por otro lado, la cantidad de actividades resultó desbalanceada para la repartición planteada. Esto generó la necesidad de que cada miembro del equipo aprenda de las otras áreas para poder avanzar en las tareas restantes. Este cruce de responsabilidades llevó más tiempo de lo que podría haber llevado si todos hubiesen trabajado sobre el total del proyecto desde un principio.

Desde el punto de vista de desarrollo, se considera una carencia no haber incorporado pruebas automáticas. Para cada liberación se realizaron pruebas de regresión en las que se probaron manualmente los principales flujos de cada actividad verificando sus criterios de aceptación. Esta verificación manual consume más tiempo que mantener un conjunto de pruebas automáticas, y debido al tiempo que lleva, se realiza previo a cada liberación. Realizar la validación de todas las funcionalidades al final de la iteración, generó la necesidad de postergar muchas de las liberaciones debido a que se encontraban incidencias que se debían atacar. De haber tenido un conjunto de pruebas automáticas, se podrían haber detectado los errores al momento de construir las funcionalidades, haciendo que el arreglo sea menos costoso.

En cuanto a la selección de tecnologías, se encuentran algunos puntos que vale la pena mencionar. Enfocarse en tecnologías de uso actual en la industria tuvo buenos resultados, así como basar la elección en las ventajas y desventajas de cada una, considerando recomendaciones, documentación concisa y curva de aprendizaje. De todos modos, un aspecto que no se tuvo en cuenta, fue que utilizar la última versión de una tecnología puede resultar que la elección no sea lo suficientemente estable, y luego deba dedicarse tiempo a acompañar las actualizaciones.

Por último, se considera que hubiese sido necesario realizar técnicas de estimación con mayor nivel de detalle y evaluar el alcance del proyecto desde un principio. En la descripción y análisis inicial del proyecto no se consideraron aspectos importantes como inconsistencia en los datos, documentación del proceso, dificultad de llevar el método de cubrimiento a una realidad modelable, complejidad de la generalización y de agregar otros métodos de evaluación, que sumado a la falta de experiencia en estimación del equipo, llevó a que el alcance fuera subestimado. En este sentido, de haber tenido una mejor estimación, se podría haber hecho un enfoque más acertado de las tareas a realizar.

## 9.2 TRABAJO A FUTURO

Se identifican varias actividades interesantes de llevar a cabo como continuación de este proyecto, algunas de ellas enfocadas a darle completitud al sistema construido, y otras enfocadas a seguir avanzando en el área de la evaluación de carreras.

Hay varias tareas que se deben realizar para mejorar la solución propuesta y poder extender el uso del sistema a distintas carreras y usuarios. En lo que refiere a funcionalidades, se deben implementar los alta, consulta y modificación de cuerpos de conocimientos, planes de estudios y carreras. La lógica por parte del *back-end* de estas funcionalidades ya se encuentra implementada, debe hacerse accesible a través de la interfaz de usuario.

Por otra parte, se deben agregar los roles y permisos de los distintos tipos de usuario de la aplicación. Tanto en el proyecto de *back-end* como en el de *front-end* se cuenta con la estructura planteada para realizar la autenticación y autorización de las distintas funcionalidades.

Otra tarea que se debe realizar en caso de querer extender el uso del sistema, es el ajuste del plan de suscripción de Heroku. Para ello es recomendable cuantificar el uso previsto de la herramienta, realizar pruebas de rendimiento, estimar la capacidad necesaria de la base de datos, y ajustar el plan para que éste contemple los requerimientos.

En lo que refiere a la evaluación de carreras, se pueden realizar varias actividades de interés. En primer lugar, parece atractiva la idea de aplicar el método de evaluación de cubrimiento temático a la evaluación de otras carreras de UDELAR con respecto a sus correspondientes cuerpos de conocimiento, siempre que sean de estructura jerárquica. Para lograrlo, se deben conocer las sugerencias que puedan plantear las guías de referencia, y llevar estos requisitos al modelo generalizado que plantea el sistema.

El otro campo sobre el que se puede avanzar, es la incorporación de la evaluación de los resultados esperados a la herramienta. Esto puede lograrse agregando el manejo de encuestas, analizando los datos resultantes, y cruzando estos datos con los provenientes del cubrimiento temático. En este sentido, interesa plantear un método genérico que permita generar información valiosa a partir del cruce de información de las evaluaciones de cubrimiento temático y resultados esperados. La EIS servirá como caso de estudio ya que se cuenta con información de los dos tipos de evaluación.

Otro aspecto en el que se puede trabajar es mejorar el mecanismo para determinar las horas reales dedicadas a cada tema de las ediciones de curso. Hoy en día se tiene en cuenta el registro de un único alumno, pero este tiempo podría llegar a diferir del tiempo dedicado por otros alumnos para una misma edición. La principal diferencia se puede dar en las horas de estudio, para las cuales el método plantea no considerar el registro, y calcularlas como dos o tres veces las horas de teórico. Aunque el sistema permite modelar esta definición, resulta interesante pensar en alternativas para obtener un valor para las horas de estudio que resulte representativo de la realidad de la mayoría de los estudiantes. Para ello, el sistema debería extenderse para soportar el registro de dedicaciones de varios alumnos para cada edición de curso, y además proveer un mecanismo para conciliar esta información.

En relación al último punto, y considerando la arquitectura planteada para la herramienta que mantiene la capa de presentación independiente de la lógica de negocio, podría resultar interesante construir una aplicación para dispositivos móviles que le permita a los estudiantes rellenar el registro de horas dedicadas a una edición de curso en tiempo real a través de sus celulares. Esta facilidad podría incluso influir en conseguir datos más precisos ya que podría pensarse hasta en cronometrar el tiempo de estudio.



## BIBLIOGRAFÍA

---

- ACM, Association for Computing Machinery. Curricula Recommendations, 2017. URL <http://www.acm.org/education/curricula-recommendations>.
- Bloom, Benjamin S; Engelhart, Max D; Furst, Edward J; Hill, Walker H; Krathwohl, David R, and others, . Taxonomy of educational objectives: The classification of educational goals. handbook i: Cognitive domain. new york: David mckay company. 1956.
- Camilloni, Lucía. Evaluación de las carreras de posgrado en Ingeniería de Software de la Universidad de la República. Technical report, Facultad de Ingeniería, Universidad de la República, 2015.
- Camilloni, Lucía and Vallespir, Diego. *Capítulo V: La formación en Ingeniería de Software del Libro Blanco de la Ingeniería de Software en América Latina*. Instituto Antioqueño de Investigación, Medellín, Colombia, 2013. URL <https://www.fing.edu.uy/sites/default/files/biblio/24438/libroblancocapituloeducacionis.pdf>.
- IEEE-CS, ACM. Curriculum guidelines for undergraduate degree programs in software engineering, 2004a. URL <http://sites.computer.org/ccse/SE2004Volume.pdf>.
- IEEE-CS, SERC - Systems Engineering Research Center , INCOSE - International Council on Systems Engineering,. Body of Knowledge and Curriculum to Advance Systems Engineering Project (BKCASE), 2017a. URL <http://www.bkcase.org/>.
- IEEE-CS, The Institute of Electrical and Electronics Engineers. About swebok, a. URL <https://www.computer.org/web/swebok>.
- IEEE-CS, The Institute of Electrical and Electronics Engineers. SWEBOK Objectives, b. URL <https://www.computer.org/web/swebok/objectives>.
- IEEE-CS, The Institute of Electrical and Electronics Engineers. The management of software engineering, Part I: Principles of software engineering, 1980. URL <http://ieeexplore.ieee.org/document/5387924/>.

IEEE-CS, The Institute of Electrical and Electronics Engineers. Guide to the Software Engineering Body of Knowledge, 2004b. URL <http://www.math.unipd.it/~tullio/IS-1/2007/Approfondimenti/SWEBOK.pdf>.

IEEE-CS, The Institute of Electrical and Electronics Engineers. Bodies of Knowledge, 2017b. URL <https://www.computer.org/web/education/bodies-of-knowledge>.

Kruchten, Philippe B. The 4+ 1 view model of architecture. *IEEE software*, 12(6): 42–50, 1995.

SIT, Stevens Institute of Technology. Graduate software engineering 2009 (gswe2009), 2009. URL <http://www.acm.org/binaries/content/assets/education/gsew2009.pdf>.

Taller de sistemas de información, FING. Documento de arquitectura de software, 2015. URL <https://eva.fing.edu.uy/mod/resource/view.php?id=69262>.



## DOCUMENTO DE REQUISITOS

---

En este Anexo se detalla el documento de requisitos realizado previo a la construcción del sistema.

### A.1 SISTEMA A CONSTRUIR

El sistema a construir es un sistema de información que permite dar soporte a la evaluación de carreras con respecto a guías curriculares de referencia. En este sentido, el sistema brindará las herramientas necesarias para evaluar el cubrimiento temático de un plan de estudio con respecto a uno o varios cuerpos de conocimientos de referencia a partir del registro de horas reales que fueron ingresadas al sistema.

Para registrar el cubrimiento temático, se deben realizar dos etapas, el ingreso de las horas de dedicación sobre temas dados en ediciones de cursos, y el mapeo de las horas dedicadas a temas con los ítems de conocimiento pertenecientes a los cuerpos de conocimientos que se desean analizar.

El sistema permitirá el registro de horas reales dedicadas a las ediciones de los cursos y la realización de la correspondencia de dedicaciones a ítems de conocimiento de los cuerpos de conocimiento de referencia.

Se considera que un curso se encuentra bajo gestión de la configuración para un cuerpo de conocimiento cuando ya se tiene el registro de cubrimiento temático de una edición del curso para dicho cuerpo de conocimiento. En este caso, el sistema permitirá reutilizar el registro de horas reales dedicadas y su correspondiente mapeo con las horas pertenecientes al cuerpo de conocimiento. De esta forma solo se deberán registrar las modificaciones realizadas en la edición del curso actual y realizar el correspondiente mapeo para dichas modificaciones.

A su vez, el sistema será extensible a incorporar un módulo de encuestas que permita evaluar el grado de cumplimiento de los resultados esperados al egreso definidos por el plan de estudios, desde el punto de vista de los distintos participantes en el área: estudiantes, docentes, entre otros.

Con respecto al análisis de la información registrada en la herramienta, es de gran interés que se pueda visualizar dicha información de forma gráfica y

desde diferentes perspectivas para que luego se puedan sacar conclusiones que permitan evaluar y proponer mejoras en la implementación de una carrera.

El sistema a construir debe ser flexible de forma que pueda ser utilizado como soporte para la evaluación de distintas carreras. Se utilizará la Especialización en Ingeniería de Software de Facultad de Ingeniería de la UdelaR como caso de estudio donde se cuenta con registro de dedicaciones y análisis de cubrimiento temático. Los datos registrados de ediciones anteriores de la Especialización permitirán validar el sistema y evaluar qué otras funcionalidades puedan resultar de interés para añadir al sistema.

## A.2 USUARIOS

**ADMINISTRADOR DEL SISTEMA** Es el principal responsable del sistema. Se encarga de crear carreras y de administrar los usuarios responsables de las mismas. El administrador puede acceder a todas las funcionalidades del sistema.

**RESPONSABLE DE LA CARRERA** Usuario encargado de la administración de una carrera, de ingresar la información correspondiente a la misma y de administrar los usuarios que podrán acceder a ciertas funcionalidades vinculadas a la carrera. Entre las principales funciones de este rol se encuentran: definir cuerpos de conocimiento y vincularlos con distintos planes de estudios, registrar implementaciones de planes, definir los cursos con sus ediciones correspondientes. Además, este usuario administra a los usuarios responsables de asignar cubrimiento de la carrera, los usuarios responsables de registrar horas de una edición de un curso y los usuarios que pueden realizar consultas. El responsable de la carrera también tiene acceso al manejo de las encuestas y la visualización de los resultados obtenidos.

**RESPONSABLE DE ASIGNAR CUBRIMIENTO** Este usuario se asigna por carrera, y es el encargado de realizar la asignación del cubrimiento temático, es decir de mapear los temas a los tópicos del o los cuerpos de conocimiento de referencia asociados a la carrera.

**RESPONSABLE DEL REGISTRO DE TIEMPO POR TEMAS** Usuario encargado de registrar las horas reales que se dedican a los temas, dictados en una edición de un curso. Este usuario se asigna por edición de un curso.

**USUARIO DE CONSULTAS** Este rol se asigna por carrera, y puede visualizar los resultados obtenidos de los diferentes análisis realizados para la carrera.

### A.3 REQUERIMIENTOS FUNCIONALES

En esta sección se describen las principales funcionalidades del sistema.

**ADMINISTRAR USUARIOS DEL SISTEMA** El administrador del sistema será el encargado de administrar los usuarios responsables de las carreras. Además tendrá permiso de “super-admin”, es decir, no tendrá restricciones para usar las funcionalidades del sistema. El responsable de una carrera tendrá la responsabilidad de crear, modificar y eliminar los usuarios responsables de asignar cubrimiento, los usuarios responsables del registro de tiempos por tema para una edición de un curso y los usuarios que pueden realizar consultas sobre la carrera.

**CREACIÓN DE UNA CARRERA** El administrador del sistema debe poder crear carreras definiendo el nombre, código y responsable de la misma.

**CREACIÓN DE UN PLAN DE ESTUDIO** El usuario responsable de la carrera debe poder especificar un plan de estudio perteneciente a la carrera indicando el nombre, código, horas totales esperadas y la duración del mismo. Además deberá definir los conocimientos esperados al ingreso, el conjunto de conocimiento esperados al egreso y los objetivos educativos del plan.

**CREACIÓN DE UN CURSO** El usuario responsable de la carrera debe poder definir nuevos cursos indicando el nombre, código, cantidad de créditos, horas totales esperadas y programa. Luego, se podrán ingresar los temas correspondientes al curso, indicando su nombre.

**CREACIÓN DE UNA EDICIÓN DE UN CURSO** El usuario responsable de la carrera debe poder ingresar una nueva edición de un curso, indicando el año, docente responsable y asignándole un responsable de registro de tiempo. Aunque la edición sea creada por un responsable de carrera, la edición deberá poder compartirse entre carreras.

**CREACIÓN DE UNA IMPLEMENTACIÓN DE PLAN DE ESTUDIO** El usuario responsable de la carrera debe poder crear una implementación sobre un plan de estudio, indicándole un nombre, código y seleccionando para el mismo un conjunto de cursos electivos y obligatorios.

**REGISTRAR EJECUCIÓN DE UNA IMPLEMENTACIÓN** El responsable de la carrera podrá dar de alta una nueva ejecución de una implementación de un plan que consta de una selección de ediciones de cursos, un nombre

y una descripción de esta ejecución. Las ejecuciones pueden ser reales o simuladas. Para el caso de las ejecuciones reales, es decir, que corresponden al subconjunto de ediciones de cursos que cursó un alumno, se puede especificar el rango de años en que se lleva a cabo dicha ejecución. El sistema también permitirá simular ejecuciones de una implementación, permitiendo dejar este último campo vacío.

**REGISTRO DE DEDICACIONES** Permite al responsable de registro de tiempo, registrar dedicación horaria sobre los temas impartidos en una edición de curso. Cada dedicación sobre un tema detalla la cantidad de horas, el tipo de hora, la fecha, la fuente y una descripción de la misma.

**DEFINICIÓN DE UN CUERPO DE CONOCIMIENTO** El usuario administrador del sistema o cualquier responsable de carrera puede definir un cuerpo de conocimiento, indicando su nombre y descripción. El sistema permitirá analizar aquellos cuerpos de conocimiento que tengan una estructura jerárquica arborescente, que se modelará al crear el cuerpo, agregando la cantidad de niveles en los que se estructura, y el nombre de cada nivel. Luego, para cada nivel, se deben agregar los ítems que pertenecen a dicho nivel, indicando su nombre, código e ítem padre en caso de corresponder.

**RELACIONAR UN PLAN DE ESTUDIO CON UN CUERPO DE CONOCIMIENTO** Un responsable de carrera podrá relacionar los planes de estudio de sus carreras asociadas a cuerpos de conocimientos existentes en el sistema. Para cada relación cuerpo de conocimiento con plan de estudio se debe definir la ponderación para los tipos de hora usada como horas totales para la visualización de resultados obtenidos. Los tipos de horas a ponderar serán los siguientes:

- Teórico presencial
- Práctico presencial
- Práctico no presencial
- Laboratorio presencial
- Laboratorio no presencial
- Evaluación
- Estudio

Es decir, suponiendo que se cuenta con la siguiente ponderación: 3 para las horas de teórico presencial, 1 para las horas prácticas no presenciales,

y cero para el resto, entonces al realizar la evaluación de cubrimiento considerando horas totales, las mismas se calcularán como:

$$\begin{aligned} \text{HorasTotales} &= 3 * \text{horasTeóricoPresencial} \\ &+ 1 * \text{horasPrácticasNoPresenciales.} \end{aligned} \quad (2)$$

Además, en este paso se deberá definir al menos un usuario responsable de la asignación de cubrimiento, quien será experto sobre el cuerpo, y responsable de realizar las correspondencias entre temas dados en la ejecución del plan con ítems del cuerpo. Posteriormente, los responsables de carrera podrán editar la relación cuerpo-plan modificando los atributos definidos en un principio, o agregando y quitando responsables de asignación de cubrimientos.

**MAPEO DE DEDICACIONES A ÍTEMS DEL CUERPO DE CONOCIMIENTO** El responsable de la asignación de cubrimiento deberá poder realizar la correspondencia entre los temas impartidos en una edición de curso, con los ítems de los cuerpos de conocimiento asociados a los planes de estudio relacionados a dicha edición. Se deberá poder definir una correspondencia del tema a ítems del cuerpo de conocimiento (por cada cuerpo de conocimiento que se desee analizar el cubrimiento temático obtenido). Para ello el sistema listará el registro de horas de dedicaciones, permitiendo modificar mapeos existentes y agregar los que aún no se hayan realizado. El responsable de asignación seleccionará un cuerpo de conocimiento sobre el cual se quiera definir las correspondencias, y luego para cada dedicación el sistema permitirá registrar la correspondencia con uno o más ítems de conocimiento que se encuentren en cualquier nivel de la jerarquía. Al mapear un tema con un ítem de la jerarquía, el mismo se mapea con todos los ancestros de dicho ítem, por lo que las horas registradas para este ítem se tienen en cuenta también para sus ancestros. Cuando un tema se mapea con más de un ítem, se debe poder indicar cuánto tiempo se asigna a cada uno de los ítems. Luego, cuando un tema se mapea a ítems de conocimiento que no son hojas del árbol de jerarquía, el sistema soportará las siguientes 3 situaciones:

- El tema se mapea con uno o más ítems hijos, y es significativo registrar la distribución de horas a los ítems hijos. El total de horas del ítem padre debe ser superior a la suma total de los horas asignadas a los ítems hijos. Este caso se puede ejemplificar con la siguiente situación: un tema denominado  $t_1$  se vincula con la unidad llamada

unidad 1, la misma está compuesta por los tópicos t1 y 2. La dedicación horaria del tema t1 se corresponden con 3 horas prácticas asociadas a la unidad 1, que se dividen en 1 hora práctica al tópico 1 y 2 horas prácticas al tópico dos.

- El tema se mapea con varios o todos los ítems hijos, y no resulta significativo asignar registros a cada uno de los ítems hijos. Esta situación se puede ver reflejada por ejemplo con horas de dedicación de una prueba, la cual cubre muchos ítems descendientes de un ítem padre. Las horas registradas a la prueba se asignan al ítem padre dado que el tiempo dedicado a cada ítem descendiente de él no resulta significativo por sí solo. Esta relación deberá especificarse como de tipo VA.
- El tema no se mapea con ningún ítem hijo. En este caso el tema se vincula al ítem padre pero no se vincula directamente con ninguno de sus ítems hijos. Esta relación deberá especificarse como de tipo NG.

El sistema también permitirá indicar que un tema no pertenece al cuerpo de conocimiento (no se corresponde ni está incluido en ningún ítem del mismo).

**REUTILIZAR REGISTRO: CURSO BAJO GESTIÓN DE LA CONFIGURACIÓN** Esta funcionalidad permitiría registrar de forma más sencilla los tiempos por tema y el mapeo de temas a tópicos para los cursos que están bajo gestión de la configuración para un cuerpo de conocimiento. En el caso que sea un curso bajo gestión de la configuración el sistema permitirá:

- Ingresar registro de horas y mapeo de temas como si fuese un nuevo curso.
- Seleccionar una edición anterior del curso y reutilizar sólo su registro de horas.
- Seleccionar una edición anterior del curso y reutilizar su registro de horas y el mapeo de temas para uno o varios cuerpos de conocimiento.

Además, luego de realizar la copia o reutilización, el sistema permite al administrador, responsable de carrera o el usuario responsable de cubrimiento modificar estas dedicaciones, permitiendo eliminar o agregar temas, además de modificar la carga horaria de los distintos registros de horas de dedicación sobre cierto tema. Cuando se agrega un nuevo tema

con su respectivas horas de dedicación se debe definir la correspondencia entre las horas registradas a temas y las horas registradas a ítems de conocimiento. Si se modifica la carga horaria de un tema ya existente, el mapeo a ítems se mantiene, pero se deben modificar los tiempos por ítem, repitiendo la validación por parte del sistema, en que se chequea que la suma de las horas por ítem no superen las horas dedicadas al tema. Por último, en el caso que se eliminen las horas de dedicación de un tema, las horas correspondientes a los ítems de conocimiento mapeados a las mismas se deberán también eliminar.

**CONFIGURACIÓN DE PERFILES** El administrador del sistema o el responsable de la carrera deberán poder definir parámetros a utilizar para flexibilizar el análisis de una ejecución de un plan de estudio contra un determinado cuerpo de conocimiento asociado a dicho plan. El sistema permitirá definir estos parámetros asociándolos a un cuerpo de conocimiento existente ya ingresado y guardarlos en forma de perfiles a modo de mantener las configuraciones reutilizables entre los distintos planes de estudio que se vayan a evaluar contra dicho cuerpo. Estas configuraciones o perfiles, tienen por objetivo parametrizar el criterio de horas mínimas, dedicación mínima y máxima, y el porcentaje de cubrimiento temático, explicados a continuación.

**CRITERIO DE HORAS MÍNIMAS** El sistema soportará la definición y almacenamiento de un conjunto de tuplas, del tipo <CTH, HM>, donde CTH corresponde a un conjunto de tipo de horas y HM son las horas mínimas requeridas para ese conjunto. Esta solución permite definir mínimos por cada tipo de hora, o para la agrupación de varios tipos de horas en casos que no se considere conveniente granularizar la configuración. A modo de ejemplo, suponemos que se cuenta con los siguientes tipos de hora:

- Teórico Presencial (TP)
- Teórico No Presencial (TNP)
- Prácticas Presencial (PP)
- Práctico No Presencial (PNP)

Entonces, CTH sería un subconjunto de TP, TNP, PP, PNP, HT, donde TP, TNP, PP, PNP son los tipos de horas definidos, y HT son las horas totales, que se calculan automáticamente como la suma de los distintos tipos de horas, con sus respectivas ponderaciones definidas por la relación plan de estudio con cuerpo de conocimiento:

$$HT = \text{ponderacionTP} \times \text{horasTP} + \text{ponderacionTNP} \times \text{horasTNP} \\ + \text{ponderacionPP} \times \text{horasPP} + \text{ponderacionPNP} \times \text{horasPNP} \quad (3)$$

Luego, para cada CTH se especifican las horas mínimas requeridas.

Ejemplos:

- La siguiente configuración permite asignar mínimos por tipo de hora:

$$\text{Config1} = \{ [ TP, 3], [ TNP, 5 ], [ PP, 2 ], [ PNP, 4 ] \}$$

Un ítem hoja de la estructura arborescente del cuerpo de conocimiento será considerado cubierto si cumple todas las condiciones siguientes.

- Total Horas Teóricas Presenciales  $\geq 3$
- Total Horas Teóricas No Presenciales  $\geq 5$
- Total Horas Prácticas Presenciales  $\geq 2$
- Total Horas Prácticas NO Presenciales  $\geq 2$

- La siguiente configuración permite asignar mínimos diferenciando por horas practicas y teóricas.

$\text{Config2} = \{ [ TP, TNP, 3], [ PP, PNP, 2 ] \}$  Un ítem hoja de la estructura arborescente del cuerpo de conocimiento será considerado cubierto si cumple todas las condiciones siguientes:

- $(\text{Total Horas TP}) + (\text{Total Horas TNP}) \geq 3$
- $(\text{Total Horas PP}) + (\text{Total Horas PNP}) \geq 2$

- La siguiente configuración permite asignar un mínimo de horas presenciales.

$\text{Config3} = \{ [ TP, PP, 3] \}$  Un ítem hoja de la estructura arborescente del cuerpo de conocimiento será considerado cubierto si cumple todas las condiciones siguientes:

- $(\text{Total Horas TP}) + (\text{Total Horas PP}) \geq 3$

- La siguiente configuración permite asignar un mínimo de horas de tipo Teórico Presenciales.

$\text{Config4} = \{ [ TP, 10] \}$  Un ítem hoja de la estructura arborescente del cuerpo de conocimiento será considerado cubierto si cumple todas las condiciones siguientes:

- (Total Horas TP)  $\geq 10$
- Para el caso donde se considera una configuración donde la suma de todas las horas sea mayor a un valor sin tener en cuenta ponderaciones se define la siguiente configuración.  
Config5 = { [ TP, TNP, PP, PNP, 12 ] } Un ítem hoja de la estructura arborescente del cuerpo de conocimiento será considerado cubierto si cumple todas las condiciones siguientes:
  - (Total Horas TP) + (Total Horas TNP) + (Total Horas PP) + (Total Horas PNP)  $\geq 12$
- Por último, para configurar el mínimo de horas totales, es decir, la suma de todos los tips de horas multiplicados por sus respectivas ponderaciones, se debe realizar la siguiente configuración:  
Config6 = { [ HT, 1 ] } Un ítem hoja de la estructura arborescente del cuerpo de conocimiento será considerado cubierto si cumple la siguiente condición:
  - (Ponderación TP)  $\times$  (Total Horas TP) + (Ponderación TNP)  $\times$  (Total Horas TNP) + (Ponderación PP)  $\times$  (Total Horas PP) + (Ponderación PNP)  $\times$  (Total Horas PNP)  $\geq 1$

Luego, esta configuración podrá ser reutilizada para evaluar cualquier plan de estudio con respecto al mismo cuerpo de conocimiento

#### PORCENTAJE DE CUBRIMIENTO MÍNIMO Y FRANJAS DE CUBRIMIENTO

Las franjas de cubrimientos se especifican como un conjunto de límites o cotas, con un nombre y color asociados. Los límites no se pueden solapar, y en su unión, deben cubrir los valores de 0 a 100. Esto se debe a que los límites definen porcentajes mínimos y máximos de hijos cubiertos para determinar distintas franjas de cubrimiento, en las que se busca ubicar a los ítems al ser evaluados. Cuando un ítem tiene X% de hijos cubiertos, se buscan el nombre y color asociados a la franja del rango al que pertenece X, por ejemplo “parcialmente cubierto” y color anaranjado, entonces al hacer la evaluación se dice que el ítem está parcialmente cubierto y se pinta de anaranjado.

Las franjas de cubrimiento se definen para todos los niveles de la jerarquía, excepto para el nivel inferior, donde el resultado de la evaluación no es una franja, sino un binario cubierto o no, sin valor de porcentaje asociado, ya que se calcula como una comparación simple teniendo en cuenta si cumplen con el criterio de horas mínimas.

Para el caso de los ítems del segundo nivel, éstos calculan el porcentaje de ítems hijos cubiertos, y con éste valor determinan su franja de cubrimiento. Por último, para los ítems de un tercer nivel o mayor, es decir, que tienen al menos dos niveles descendientes, es necesario definir el porcentaje de cubrimiento, el cual permite determinar la franja de cubrimiento.

Para un ítem de tercer nivel en adelante, el porcentaje de cubrimiento mínimo es un valor que define el porcentaje de cubrimiento que tienen que tener al menos los ítems hijos para ser considerados como cubiertos por su padre (ítem perteneciente al nivel analizado). Luego de comparar los porcentajes de cubrimiento de los hijos con este porcentaje mínimo, el ítem padre contabiliza el porcentaje de hijos cubiertos, para determinar en qué franja de cubrimiento se encuentra.

A continuación se presenta un ejemplo que permite aplicar los conceptos anteriores.

Suponemos un cuerpo de conocimiento compuesto por tres niveles: Tópicos, Unidades y Áreas con la siguiente estructura:

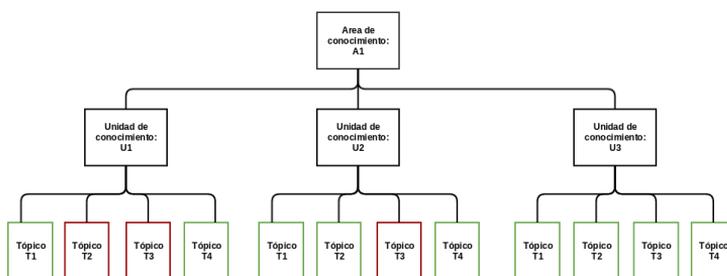


Figura 23: Porcentaje de cubrimiento mínimo - Cuerpo conocimiento

En la figura, aparecen coloreados en verde los tópicos que alcanzan el mínimo de horas requeridas en base al criterio definido para horas mínimas, y en rojo los que no. Además supongamos que se definen las siguientes franjas de cubrimiento para las unidades:

- No cubierto (Rojo): 0 y 20 %
- Poco Cubierto (Anaranjado): entre +20 % y 40 %

- Cubierto (Verde Claro): entre +40 % y 60 %
- Cubierto Ampliamente (Verde Oscuro): mas de 60 %

Luego, para el ejemplo tiene sentido definir el porcentaje de cubrimiento mínimo sólo para las áreas de conocimiento dado que es el único nivel con al menos dos niveles descendientes. Se utilizará una configuración del 60 % como porcentaje de cubrimiento mínimo para este nivel. De modo que el área de conocimiento  $A_1$  considerará una unidad hija como cubierta si la misma tiene un porcentaje de cubrimiento mayor o igual al 60 %. En este caso particular, implica que la unidad tenga al menos el 60 % de los tópicos que la componen cubiertos, o lo que es equivalente, que la unidad esté “Cubierta ampliamente”.

Para este ejemplo analizaremos cómo se determina el porcentaje de cubrimiento del área de conocimiento  $A_1$ , en base al cubrimiento de las unidades que la componen. De misma manera analizaremos la franja a la cual pertenece cada unidad.

- Unidad  $U_1$ 
  - La unidad  $U_1$  tiene como porcentaje de cubrimiento temático 50 % ya que la mitad de los tópicos que la componen están cubiertos.
  - $U_1$  se considera “Cubierta” y pintada de verde claro para la evaluación.
  - $A_1$  no considera a  $U_1$  cubierta, ya que el porcentaje de cubrimiento es menor al 60 % definido como mínimo, o lo que es equivalente, no llega a estar “Cubierta ampliamente”.
- Unidad  $U_2$ 
  - La unidad  $U_2$  tiene como porcentaje de cubrimiento temático 75 % ya que es el porcentaje de tópicos que la componen que están cubiertos.
  - $U_2$  se considera “Cubierta Ampliamente” y se pinta de verde oscuro al realizar la evaluación.
  - $A_1$  considera cubierta a  $U_2$ , dado que su porcentaje de cubrimiento es mayor al 60 % definido como mínimo, o lo que es equivalente, está “Cubierta Ampliamente”.
- Unidad  $U_3$

- La unidad U<sub>3</sub> tiene como porcentaje de cubrimiento temático 100 % ya que todos los tópicos que la componen están cubiertos.
- U<sub>3</sub> se considera “Cubierta Ampliamente” y se pinta de verde oscuro al realizar la evaluación.
- A<sub>1</sub> considera cubierta a U<sub>3</sub>, dado que su porcentaje de cubrimiento es mayor al 60 % definido como mínimo, o lo que es equivalente, está “Cubierta Ampliamente”.

Luego de realizar el análisis de cubrimiento del nivel tópico, el árbol que representa el cuerpo de conocimiento queda de la siguiente manera:

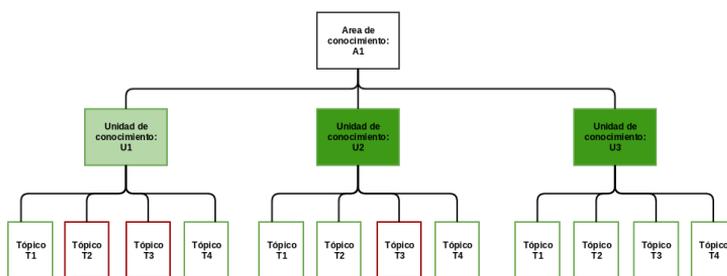


Figura 24: Porcentaje de cubrimiento mínimo - Análisis cubrimiento tópico

Se observa que el área de conocimiento A<sub>1</sub> tiene un 66.67 % de unidades hijas consideradas cubiertas. A partir de este valor, A<sub>1</sub> calcula su franja de cubrimiento. Suponiendo que se definieron las siguientes franjas de cubrimiento para las áreas de conocimiento:

- No cubierto (Naranja): 0 y 15 %
- Cubierto Parcialmente (Amarillo): entre +15 % y 50 %
- Cubierto Ampliamente (Verde Claro): entre +50 % y 85 %
- Cubierto Ampliamente (Verde Oscuro): mas de 85 %

Entonces, A<sub>1</sub> pertenece a la franja Cubierto Ampliamente, y se colorea de verde claro.

Por último, el sistema deberá soportar definir los parámetros utilizados para definir tanto el porcentaje mínimo de cubrimiento, cómo las

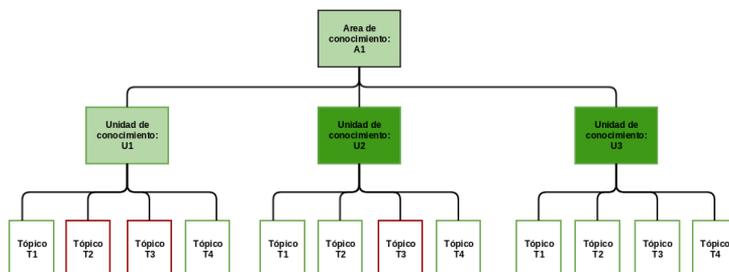


Figura 25: Porcentaje de cubrimiento mínimo - Análisis cubrimiento unidad

franjas de cubrimiento, que serán reutilizadas para evaluar cualquier plan de estudios sobre un mismo cuerpo de conocimiento.

**DEDICACIONES MÍNIMAS Y MÁXIMAS** Para los ítems de cualquier nivel de la jerarquía del cuerpo de conocimiento se podrán definir la dedicación mínima y máxima esperada, utilizando una metodología similar a la anterior, especificando un conjunto de tuplas <CTH, HMin, HMax>, donde CTH es un conjunto de tipo de horas, HMin son las horas mínimas y HMax las horas máximas esperadas para un ítem. Como esta información puede ser relevante únicamente para los ítems de algunos niveles de la jerarquía, primero se deberá definir para qué niveles aplica ingresar ésta dedicación esperada. Luego, esta configuración podrá ser reutilizada para evaluar cualquier plan de estudio con respecto al mismo cuerpo de conocimiento.

**VISUALIZACIÓN DE RESULTADOS OBTENIDOS** El usuario administrador, los usuarios responsables de carreras, responsables de asignación de cubrimiento y los usuarios de consultas, deberán poder observar los resultados obtenidos para las diferentes ejecuciones de implementaciones de un plan de estudio o selecciones de ediciones de cursos. En el caso de que se cuente con más de un cuerpo de conocimiento vinculado al plan de estudio el usuario podrá seleccionar el cuerpo de conocimiento para el cual desea visualizar la información de cubrimiento. Una vez que se seleccionó el cuerpo de conocimiento, se deberá seleccionar el perfil de configuración asociado al cuerpo que se desea utilizar para evaluar la ejecución de la implementación del plan de estudio. Al seleccionar el perfil, se utilizarán los parámetros definidos en el mismo para realizar las evaluaciones de

cubrimiento temático, distribución del esfuerzo temático, y balance temático desarrollados luego en esta sección. Luego de seleccionar el perfil, el sistema permitirá seleccionar los tipos de horas que se desean considerar para visualizar los resultados, además de permitir usar la ponderación definida para la carrera y el cuerpo de conocimiento, o modificarla para así calcular lo que se desea considerar como total de horas. Para los ítems de conocimiento que se define una dedicación mínima y máxima por tipo de hora, el sistema permitirá visualizar la comparación entre las horas reales registradas para el ítem, en las distintas ediciones de los cursos que se consideran en la consulta, con las esperadas para el perfil definido para el cuerpo de conocimiento. El sistema permitirá exportar los resultados obtenidos, además de los cruzamientos analizados a formato *xlsx*. Además en todas las consultas se deberá poder navegar a través de cómo se compone la dedicación para cada ítem, visualizando los temas y cursos respectivos a las dedicaciones, a modo de lograr una mayor comprensión del resultado de la evaluación para el ítem.

**CUBRIMIENTO TEMÁTICO** El sistema mostrará el cubrimiento temático de los ítems de conocimiento, coloreándolos según el porcentaje de cubrimiento obtenido. El sistema permitirá navegar a través de los niveles de la jerarquía del cuerpo de conocimiento, para luego mostrar el cubrimiento temático de los ítems que componen el nivel seleccionado. Se utilizará la configuración del nivel para saber si los ítems de conocimiento pertenecientes al mismo están cubiertos o no y a cuál franja de color le corresponde acuerdo al cubrimiento alcanzado. El sistema mostrará estos ítems gráficamente con los colores que corresponden a esta configuración. A su vez, se podrá hacer este análisis, a partir de seleccionar un ítem del cuerpo, y así visualizar el cubrimiento de los ítems que lo componen.

**ESFUERZO TEMÁTICO** El sistema permitirá estudiar el esfuerzo temático, que es la dedicación temática expresada en horas, seleccionando el nivel jerárquico que se desea estudiar. A modo de ejemplo, si el cuerpo de conocimiento de referencia es el CBOK, si se selecciona el nivel áreas de conocimiento se visualizará los resultados obtenidos sumariados por áreas de conocimiento, si se selecciona unidades se visualizará los resultados obtenidos por las unidades y si se seleccionan los tópicos se visualizará el esfuerzo por tópico. También se podrá seleccionar un ítem específico del cuerpo de conocimiento, y visualizar el esfuerzo temático de sus ítems hijos. Al tener el esfuerzo temático por nivel, se podrá realizar un informe exportable con la información de esfuerzo para dicho nivel, por ejemplo,

cuántos ítems del nivel están en cada color de los definidos y cuáles son. El sistema permitirá elegir sobre qué tipo de horas mostrar el esfuerzo temático, además, para los niveles que cuenten con dedicación mínima y máxima esperada, se va a permitir evaluar gráficamente distinguiendo por colores si está por debajo, o no del esfuerzo esperado. El análisis se podrá realizar sobre un solo tipo de hora, o sobre un conjunto seleccionado de tipos de hora.

**BALANCE TEMÁTICO** Para aquellos ítems de conocimiento que cuentan con la información de horas mínimas y máximas esperadas, el sistema mostrará el resultado obtenido de evaluar el porcentaje de tiempo dedicado a cada ítem en comparación a los porcentajes mínimos y máximos esperadas. Se presentará esta información con gráficas de puntos indicando la carga de horas porcentual dedicadas al ítem de conocimiento, además de graficar los porcentajes mínimos y máximos esperados. El sistema permitirá seleccionar sobre qué tipo de horas trabajar, además de permitir seleccionar qué nivel jerárquico se desea considerar.

**COMPARACIÓN DE RESULTADOS** Se debe poder comparar los resultados obtenidos entre dos o más ejecuciones de implementaciones de un plan, ya sean reales, o simuladas seleccionando un conjunto de ediciones de cursos, mostrando los resultados desde todos los puntos de vista que se describieron anteriormente.

**LISTAR ÍTEMS DE CONOCIMIENTO COMPARTIDOS POR CURSOS** El sistema debe permitir visualizar de forma fácil, el solapamiento en el dictado de ítems de conocimientos entre distintos cursos. Esto implica poder visualizar qué ítems son dados en más de una edición de curso de la implementación del plan o conjunto de ediciones seleccionadas, y a su vez, visualizar cuáles son estas ediciones.

#### A.4 REQUERIMIENTOS NO FUNCIONALES

En esta sección se detallan los requerimientos no funcionales del sistema.

**UTILIZACIÓN DE SOFTWARE LIBRE** El sistema debe utilizar herramientas que no requieran la compra de licencias.

**DOCUMENTACIÓN** El sistema deberá contar con un manual de usuario, de instalación y de desarrollo.

**NAVEGADORES** El sistema deberá ser compatible con los navegadores Chrome a partir de la versión 51.0.2704.63 y Firefox a partir de la versión 47.0, ya que estas son las últimas versiones estables.

**DESCARGA DE RESULTADOS OBTENIDOS** El sistema permitirá exportar los resultados obtenidos de las consultas y cruce de información en formato pdf, odt y xlsx.

**CARGA DE DATOS** El sistema permitirá importar datos, como por ejemplo, registro de dedicaciones desde planillas xlsx.

**PERFORMANCE** El sistema deberá mostrar el resultado de las consultas en tiempos menores a 10 segundos.

#### A.5 RESTRICCIONES

En esta sección se describen las restricciones del sistema.

**PROCESO DE SOFTWARE** Se comenzará con el desarrollo de las funcionalidades principales, entre ellas, el registro de horas por temas, mapeo de temas a tópicos y visualización de los resultados. Para conocer el proceso de desarrollo de software y la metodología a utilizar para el mismo referirse al documento “ Metodología de configuración, trabajo y testing del sistema” en su versión 2.

**DURACIÓN** Se estima que la duración del desarrollo de la herramienta corresponde a 6 meses con una dedicación de 45 horas semanales (15 horas por cada integrante del equipo), considerándose en este tiempo las actividades de diseño, construcción, testing y validación. Luego se estima un mes para la realización de la migración de datos en planillas .xls existentes a la base de datos relacional del sistema, y el análisis de los mismos.

**BASE DE DATOS** El motor de base de datos debe ser relacional.

#### A.6 INTERFACES

En esta sección se describen las interfaces que el sistema debe proveer.

**INTERFACES DE USUARIOS** El sistema deberá proveer interfaces para los distintos usuarios. Los usuarios accederán al sistema mediante una interfaz web. Para poder ingresar al sistema todo usuario deberá autenticarse, de

esta forma el sistema reconocerá que tipo de usuario es y listará las funcionalidades que le corresponden. La interfaz del usuario administrador deberá incluir todas las funcionalidades de las demás interfaces, además de un módulo que permita el manejo de usuarios.

**INTERFACES CON SOFTWARE** El sistema deberá contar con un módulo para el análisis de las encuestas.



## DICCIONARIO DE LA BASE DE DATOS

---

El presente documento tiene como objetivo describir las estructuras de datos utilizadas en el sistema. En la sección B.1 se presenta el esquema de la base de datos y se describe cada una de las tablas con sus respectivos atributos. Luego, en la sección B.2 se incluye el código necesario para ingresar a la base de datos aquellos elementos que su ingreso a través del sistema quedó por fuera del alcance del proyecto.

### B.1 ESTRUCTURA Y TABLAS

En la figura 26 se observa el esquema de la base de datos del sistema. En él, se pueden observar las tablas que constituyen la base de datos, y cómo se relacionan entre sí. Para simplificar la visualización, se presentan únicamente los nombres de cada tabla, y luego se presentan en orden alfabético, indicando para cada una su descripción, sus columnas, y sus claves primarias y foráneas.



Antes de describir cada tabla, se explican los distintos tipos de datos utilizados para modelar las entidades. En el cuadro 11 se describe cada uno de estos tipos de datos <sup>50</sup>.

Tipo de columna	Descripción
<i>BIGINT</i>	Entero con signo de 8 bytes. El valor mínimo es -2147483648 to y el máximo +2147483647
<i>BOOLEAN</i>	Tipo de dato que representa los valores lógicos: verdadero y falso.
<i>BYTEA</i>	Arreglo de Bytes de longitud variable.
<i>INTEGER</i>	Entero con signo de 4 bytes. El rango del valor esta limitado por el mínimo -2147483648 y el máximo +2147483647.
<i>VARCHAR(n)</i>	Cadena de caracteres de largo variable con límite. Por defecto el máximo largo de la cadena es de 255, en los casos que se especifique otro valor, este se indicará de la siguiente forma <i>máximo especificado</i> ,
<i>TEXT</i>	Cadena de caracteres de largo variable sin límite.
<i>REAL</i>	Valor numérico real.
<i>TIMESTAMP</i>	Tipo de dato que contiene una fecha y hora.

Cuadro 11: Tipos de datos Postgres

Entre los aspectos generales a destacar de la definición de tablas, se encuentra la utilización de claves primarias simples en lugar de compuestas en los casos que se debe incluir atributos asociados a la relación. Este enfoque facilita el manejo de las entidades con las herramientas utilizadas para la abstracción de la base de datos al código.

<sup>50</sup> <https://www.postgresql.org/docs/9.2/static/datatype.html>

*Tabla authorities*

La tabla *authorities* contiene las autoridades para el manejo de la autorización de los usuarios a los distintos recursos del sistema.

Nombre de columna	Tipo de datos	Descripción
<i>id</i>	<i>BIGINT</i>	Clave primaria. Auto incrementa.
<i>created_at</i>	<i>TIMESTAMP</i>	Fecha de creación.
<i>modified_at</i>	<i>TIMESTAMP</i>	Fecha de última modificación.
<i>name</i>	<i>VARCHAR(n)</i>	Nombre de la autoridad o rol.

Cuadro 12: Tabla *authorities**Tabla bodies\_of\_knowledge*

La tabla *bodies\_of\_knowledge* contiene la información sobre los cuerpos del conocimiento.

Nombre Columna	Tipo de datos	Descripción
<i>id</i>	<i>BIGINT</i>	Clave primaria.
<i>created_at</i>	<i>TIMESTAMP</i>	Fecha de creación.
<i>modified_at</i>	<i>TIMESTAMP</i>	Fecha de última modificación.
<i>name</i>	<i>VARCHAR(n)</i>	Nombre del cuerpo de conocimiento.
<i>description</i>	<i>VARCHAR(n)</i>	Descripción del cuerpo de conocimiento.

Cuadro 13: Tabla *bodies\_of\_knowledge**Tabla bodies\_of\_knowledge\_careers*

La tabla *bodies\_of\_knowledge\_careers* es la fuente de información de la relación *N-N* entre las entidades Carrera y Cuerpo de conocimiento. Forma parte de la relación triple que dada la dupla carrera y cuerpo de conocimiento asocia uno o más responsables de mapear el cuerpo de conocimiento.

Nombre Columna	Tipo de datos	Descripción
<i>id</i>	<i>BIGINT</i>	Clave primaria.
<i>created_at</i>	<i>TIMESTAMP</i>	Fecha de creación.
<i>modified_at</i>	<i>TIMESTAMP</i>	Fecha de última modificación.
<i>body_of_knowledge_id</i>	<i>BIGINT</i>	Clave foránea que referencia a la tabla <i>body_of_knowledge</i> .
<i>career_id</i>	<i>BIGINT</i>	Clave foránea que referencia a la tabla <i>career</i> .

Cuadro 14: Tabla *bodies\_of\_knowledge\_careers*

*Tabla bodies\_of\_knowledge\_careers\_responsible\_of\_mapping*

La tabla *bodies\_of\_knowledge\_careers\_responsible\_of\_mapping* contiene la información que relaciona una persona, con la dupla formada por un cuerpo de conocimiento y una carrera, estableciendo la persona responsable de mapeo para esa dupla.

Nombre Columna	Tipo de datos	Descripción
<i>bodies_careers_id</i>	<i>BIGINT</i>	Clave foránea que referencia a la tabla <i>bodies_of_knowledge_careers</i> .
<i>responsible_id</i>	<i>BIGINT</i>	Clave foránea que referencia a la tabla <i>users</i> .

Cuadro 15: Tabla *bodies\_of\_knowledge\_careers\_responsible\_of\_mapping*

*Tabla bodies\_of\_knowledge\_study\_programme*

La tabla *bodies\_of\_knowledge\_study\_programme* contiene las duplas de la relación *N-N* entre las entidades Cuerpo de conocimiento y Plan de estudio. Se define para cada dupla la configuración de horas en la tabla: *bodies\_of\_knowledge\_study\_programme\_hours\_configuration*.

Nombre Columna	Tipo de datos	Descripción
<i>id</i>	<i>BIGINT</i>	Clave primaria.
<i>created_at</i>	<i>TIMESTAMP</i>	Fecha de creación.
<i>modified_at</i>	<i>TIMESTAMP</i>	Fecha de última modificación.
<i>body_of_knowledge_id</i>	<i>BIGINT</i>	Clave foránea que referencia a la tabla <i>bodies_of_knowledge</i> .
<i>study_programme_id</i>	<i>BIGINT</i>	Clave foránea que referencia a la tabla <i>study_programmes</i> .

Cuadro 16: Tabla *bodies\_of\_knowledge\_study\_programme**Tabla bodies\_of\_knowledge\_study\_programme\_hours\_configuration*

Relaciona una dupla conformada por un cuerpo de conocimiento con un plan de estudio a una o más configuraciones de horas.

Nombre Columna	Tipo de datos	Descripción
<i>id</i>	<i>BIGINT</i>	Clave primaria.
<i>bodies_programmes_id</i>	<i>BIGINT</i>	Clave foránea que referencia a la tabla <i>bodies_of_knowledge</i> .
<i>hours_configuration_id</i>	<i>BIGINT</i>	Clave foránea que referencia a la tabla <i>type_of_hour_configuration</i> .

Cuadro 17: Tabla *bodies\_of\_knowledge\_study\_programme\_hours\_configuration*.*Tabla careers*

La tabla *careers* contiene la información relacionada a las carreras registradas en el sistema.

Nombre Columna	Tipo de datos	Descripción
<i>id</i>	<i>BIGINT</i>	Clave primaria.
<i>created_at</i>	<i>TIMESTAMP</i>	Fecha de creación.
<i>modified_at</i>	<i>TIMESTAMP</i>	Fecha de última modificación.
<i>name</i>	<i>VARCHAR(n)</i>	Nombre de la carrera.
<i>code</i>	<i>VARCHAR(n)</i>	Código de la carrera.
<i>director</i>	<i>VARCHAR(n)</i>	Nombre del director de carrera.

Cuadro 18: Tabla *careers**Tabla careers\_responsible*

La tabla *careers\_responsible* relaciona las carreras definidas en el sistema con los usuarios responsables de cada una.

Nombre Columna	Tipo de datos	Descripción
<i>careers_id</i>	<i>BIGINT</i>	Clave foránea que referencia a la tabla <i>careers</i> .
<i>responsible_id</i>	<i>BIGINT</i>	Clave foránea que referencia a la tabla <i>users</i> .

Cuadro 19: Tabla *careers\_responsible**Tabla configuration\_profiles*

Contiene la información de los perfiles de configuración que se utilizan en las consultas. Los perfiles definen el subconjunto de horas que se van a tomar en cuenta para realizar la evaluación. En este sentido, para almacenar la selección de tipos de horas de forma óptima, se define un número primo para identificar cada tipo de hora, y se guarda el producto de los identificadores de cada tipo de hora seleccionado, evitando así crear una fila para cada relación. En el cuadro 20 se observa el identificador asociado a cada tipo de hora.

Tipo de hora	Identificador
Teórico presencial	2
Práctico presencial	3
Práctico no presencial	5
Laboratorio presencial	7
Laboratorio no presencial	11
Evaluación	13
Estudio	17

Cuadro 20: Enumerado de tipos de hora del sistema.

La tabla *configuration\_profiles* se compone de la siguientes columnas:

Nombre Columna	Tipo de datos	Descripción
<i>id</i>	<i>BIGINT</i>	Clave primaria.
<i>created_at</i>	<i>TIMESTAMP</i>	Fecha de creación.
<i>modified_at</i>	<i>TIMESTAMP</i>	Fecha de última modificación.
<i>name</i>	<i>VARCHAR(n)</i>	Nombre del perfil de configuración.
<i>hour_configuration</i>	<i>BIGINT</i>	Valor numérico que representa el conjunto de horas que se consideran en el perfil. Se calcula como la multiplicación de todos los identificadores de los tipos de horas seleccionados.
<i>include_weighing</i>	<i>BOOLEAN</i>	Valor booleano que indica si se deben incluir las ponderaciones definidas para los tipos de hora en la tabla <i>type_of_hour_configuration</i> en el procesamiento de las consultas.

Cuadro 21: Tabla *configuration\_profiles*

Tabla *configuration\_profiles\_knowledge\_items*

La tabla *configuration\_profiles\_knowledge\_items* define para un perfil, los valores mínimos y máximos esperados para evaluar el esfuerzo y balance temático de un ítem de conocimiento.

Nombre Columna	Tipo de datos	Descripción
<i>id</i>	<i>BIGINT</i>	Clave primaria.
<i>balance_minimum_percentage</i>	<i>REAL</i>	Valor que representa el porcentaje mínimo esperado en la consulta de balance temático.
<i>balance_maximum_percentage</i>	<i>REAL</i>	Valor que representa el porcentaje máximo esperado en la consulta de balance temático.
<i>effortt_minimum_hours</i>	<i>REAL</i>	Valor que representa la cantidad de horas mínimas de esfuerzo esperado.
<i>effort_maximum_hours</i>	<i>REAL</i>	Valor que representa la cantidad de horas máximas de esfuerzo esperado.
<i>configuration_profile_id</i>	<i>BIGINT</i>	Clave foránea que referencia a la tabla <i>configuration_profile</i> .
<i>knowledge_item_id</i>	<i>BIGINT</i>	Clave foránea que referencia a la tabla <i>knowledge_items</i> .

Cuadro 22: Tabla *configuration\_profiles\_knowledge\_items*Tabla *configuration\_profiles\_levels\_effort\_and\_balance\_applies*

La tabla *configuration\_profiles\_levels\_effort\_and\_balance\_applies* define para un perfil de configuración, sobre qué niveles corresponde establecer los valores de balance y esfuerzo temático esperados.

Nombre Columna	Tipo de datos	Descripción
<i>id</i>	<i>BIGINT</i>	Clave primaria.
<i>applies</i>	<i>REAL</i>	Valor que especifica si se pueden establecer los valores esperados de balance y esfuerzo temático para los ítems de ese nivel.
<i>configuration_profile_id</i>	<i>BIGINT</i>	Clave foránea que referencia a la tabla <i>configuration_profile</i> .
<i>knowledge_level_id</i>	<i>BIGINT</i>	Clave foránea que referencia a la tabla <i>knowledge_levels</i> .

Cuadro 23: Tabla *configuration\_profiles\_levels\_effort\_and\_balance\_applies*

#### Tabla *configuration\_profiles\_parent\_level*

La tabla *configuration\_profiles\_parent\_level* define para un perfil de configuración y un nivel de conocimiento, cuál es el porcentaje mínimo de hijos cubiertos que debe tener un elemento de ese nivel, para considerarse cubierto.

Nombre Columna	Tipo de datos	Descripción
<i>id</i>	<i>BIGINT</i>	Clave primaria.
<i>minimum_coverage_percentage</i>	<i>REAL</i>	Porcentaje mínimo de cubrimiento de ítems hijos.
<i>configuration_profile_id</i>	<i>BIGINT</i>	Clave foránea que referencia a la tabla <i>configuration_profile</i> .
<i>knowledge_level_id</i>	<i>BIGINT</i>	Clave foránea que referencia a la tabla <i>knowledge_levels</i> .

Cuadro 24: Tabla *configuration\_profiles\_parent\_level*

*Tabla configuration\_profiles\_parent\_level\_coverage\_ranges*

La tabla *configuration\_profiles\_parent\_level\_coverage\_ranges* define para un perfil de configuración, las franjas de cubrimiento que aplican a cada nivel con hijos.

Nombre Columna	Tipo de datos	Descripción
<i>id</i>	<i>BIGINT</i>	Clave primaria.
<i>configuration_profile_level_id</i>	<i>BIGINT</i>	Clave foránea que referencia a la tabla <i>configuration_profiles_parent_levels</i> .
<i>coverage_range_id</i>	<i>BIGINT</i>	Clave foránea que referencia a la tabla <i>coverage_ranges</i> .

Cuadro 25: Tabla *configuration\_profiles\_parent\_level\_coverage\_ranges*

*Tabla courses*

La tabla *courses* contiene la información sobre los cursos del sistema.

Columna	Tipo de datos	Descripción
<i>id</i>	<i>BIGINT</i>	Clave primaria, se incrementa automáticamente.
<i>created_at</i>	<i>TIMESTAMP</i>	Fecha de creación.
<i>modified_at</i>	<i>TIMESTAMP</i>	Fecha de última modificación.
<i>code</i>	<i>VARCHAR(n)</i>	Código del curso. El valor no puede ser nulo.
<i>credits</i>	<i>INTEGER</i>	Cantidad de créditos que proporciona el curso.
<i>expected_hours</i>	<i>INTEGER</i>	Cantidad de horas de dedicación estimadas.
<i>name</i>	<i>VARCHAR(n)</i>	Nombre del curso. El valor no puede ser nulo.
<i>programme</i>	<i>TEXT</i>	Descripción del programa que es dictado en el curso.

Cuadro 26: Tabla *courses*

*Tabla course\_editions*

La tabla *course\_editions* contiene la información sobre las ediciones de cursos del sistema.

Nombre Columna	Tipo de datos	Descripción
<i>id</i>	<i>BIGINT</i>	Clave primaria.
<i>created_at</i>	<i>TIMESTAMP</i>	Fecha de creación.
<i>modified_at</i>	<i>TIMESTAMP</i>	Fecha de última modificación.
<i>year</i>	<i>INTEGER</i>	Año en que fue dictada la edición
<i>course_id</i>	<i>BIGINT</i>	Clave foránea que referencia a la tabla <i>courses</i> . Conecta la edición al curso que pertenece
<i>responsible_id</i>	<i>BIGINT</i>	Clave foránea que referencia a la tabla <i>people</i> . Conecta la edición al curso con la persona responsable de esa edición.
<i>time_tracker_id</i>	<i>BIGINT</i>	Clave foránea que referencia a la tabla <i>users</i> . Conecta la edición de curso con la persona responsable de registrar las horas de temas dictados. A diferencia del encargado del curso, el responsable de registro debe ser un usuario del sistema

Cuadro 27: Tabla *course\_editions**Tabla courses\_edition\_themes*

La tabla *courses\_edition\_themes* relaciona las ediciones de cursos con los temas dictados. Se crea una clave primaria y no se utiliza una compuesta debido a que se asocian un conjunto de dedicaciones a la dupla formada.

Nombre Columna	Tipo de datos	Descripción
<i>id</i>	<i>BIGINT</i>	Clave primaria, se incrementa automáticamente.
<i>created_at</i>	<i>TIMESTAMP</i>	Fecha de creación.
<i>modified_at</i>	<i>TIMESTAMP</i>	Fecha de última modificación.
<i>course_edition</i>	<i>BIGINT</i>	Clave foránea que referencia a la tabla <i>course_edition</i> .
<i>theme_id</i>	<i>BIGINT</i>	Clave foránea que referencia a la tabla <i>themes</i> .

Cuadro 28: Tabla *courses\_edition\_themes**Tabla course\_editions\_executions\_of\_implementations*

La tabla *course\_editions\_executions\_of\_implementations* contiene la información de la relación *N-N* entre las ediciones de los cursos y las ejecuciones de implementaciones.

Nombre Columna	Tipo de datos	Descripción
<i>edition_id</i>	<i>BIGINT</i>	Clave foránea que referencia a la tabla <i>course_editions</i>
<i>execution_id</i>	<i>BIGINT</i>	Clave foránea que referencia a la tabla <i>executions_of_implementation</i> .

Cuadro 29: Tabla *course\_editions\_executions\_of\_implementations**Tabla courses\_plan\_implementations*

La tabla *courses\_plan\_implementations* contiene la información de la relación *N-N* conformada entre los cursos y las implementaciones de planes.

Nombre Columna	Tipo de datos	Descripción
<i>id</i>	<i>BIGINT</i>	Clave primaria.
<i>created_at</i>	<i>TIMESTAMP</i>	Fecha de creación.
<i>modified_at</i>	<i>TIMESTAMP</i>	Fecha de última modificación.
<i>elective_course</i>	<i>BOOLEAN</i>	Valor booleano que representa si el curso referenciado es electivo para la implementación del pla.
<i>course_id</i>	<i>BIGINT</i>	Clave foránea que referencia a la tabla <i>courses</i>
<i>plan_implementation_id</i>	<i>BIGINT</i>	Clave foránea que referencia a la tabla <i>plan_implementations</i>

Cuadro 30: Tabla *courses\_plan\_implementations**Tabla coverage\_ranges*

La tabla *coverage\_ranges* contiene la información de los rangos de cobertura definidos en el sistema.

Nombre Columna	Tipo de datos	Descripción
<i>id</i>	<i>BIGINT</i>	Clave primaria.
<i>created_at</i>	<i>TIMESTAMP</i>	Fecha de creación.
<i>modified_at</i>	<i>TIMESTAMP</i>	Fecha de última modificación.
<i>name</i>	<i>VARCHAR(n)</i>	Nombre del rango.
<i>color</i>	<i>VARCHAR(n)</i>	Secuencia de caracteres que representa el valor hexadecimal asociado al color que se define para el rango, por ejemplo #FF8000.
<i>minimum</i>	<i>REAL</i>	Valor mínimo del rango que debe ser superado por el valor evaluado para ser considerado como perteneciente al rango.
<i>maximum</i>	<i>REAL</i>	Valor máximo del rango que no debe ser superado por el valor evaluado para ser considerado como perteneciente al rango.

Cuadro 31: Tabla *coverage\_ranges**Tabla dedications*

La tabla *dedications* contiene la información acerca de las dedicaciones registradas en el sistema.

Nombre Columna	Tipo de datos	Descripción
<i>id</i>	<i>BIGINT</i>	Clave primaria.
<i>created_at</i>	<i>TIMESTAMP</i>	Fecha de creación.
<i>modified_at</i>	<i>TIMESTAMP</i>	Fecha de última modificación.
<i>date</i>	<i>TIMESTAMP</i>	Fecha de registro de la dedicación.
<i>hours_quantity</i>	<i>REAL</i>	Cantidad de horas que se registraron para la dedicación.
<i>type_of_hour</i>	<i>INTEGER</i>	Valor numérico que representa el tipo de hora, definido en 20
<i>source_name</i>	<i>VARCHAR(n)</i>	Nombre del archivo que fue utilizado para dictar el tema.
<i>source_file</i>	<i>VARCHAR(n)</i>	Enlace al archivo que fue utilizado en la dedicación.
<i>subtheme</i>	<i>TEXT</i>	Descripción de la dedicación, en este campo se pueden agregar detalles que no correspondan a ninguna de las columnas anteriores.
<i>course_edition_theme_id</i>	<i>BIGINT</i>	Clave foránea que referencia a la tabla <i>courses_edition_themes</i> , conectando de esta forma la dedicación con la agregación compuesta por una edición de curso y un tema dictado.

Cuadro 32: Tabla *dedications**Tabla executions\_of\_implementation*

La tabla *executions\_of\_implementation* contiene la información relacionada a las ejecuciones de una implementación de un plan de estudio.

Nombre Columna	Tipo de datos	Descripción
<i>id</i>	<i>BIGINT</i>	Clave primaria.
<i>created_at</i>	<i>TIMESTAMP</i>	Fecha de creación.
<i>modified_at</i>	<i>TIMESTAMP</i>	Fecha de última modificación.
<i>name</i>	<i>VARCHAR(n)</i>	Nombre de la ejecución.
<i>description</i>	<i>VARCHAR(n)</i>	Descripción de la ejecución.
<i>start_year</i>	<i>INTEGER</i>	Año de comienzo de la ejecución.
<i>ending_year</i>	<i>INTEGER</i>	Año de finalización de la ejecución.
<i>plan_implementation_id</i>	<i>BIGINT</i>	Clave foránea que referencia a la tabla <i>plan_implementations</i> , conecta la ejecución con la implementación del plan a la cual pertenece.

Cuadro 33: Tabla *executions\_of\_implementation**Tabla knowledge\_item\_hours*

En la tabla *knowledge\_item\_hours* se guardan los mapeos de dedicaciones a los ítems de cuerpos de conocimiento.

Nombre Columna	Tipo de datos	Descripción
<i>id</i>	<i>BIGINT</i>	Clave primaria.
<i>created_at</i>	<i>TIMESTAMP</i>	Fecha de creación.
<i>modified_at</i>	<i>TIMESTAMP</i>	Fecha de última modificación.
<i>dedication_id</i>	<i>BIGINT</i>	Clave foránea que referencia a la tabla <i>dedications</i> , representa la clave primaria de la dedicación sobre la cuál se realiza el mapeo.
<i>knowledge_item_id</i>	<i>BIGINT</i>	Clave foránea que referencia a la tabla <i>knowledge_items</i> , conecta con el ítem de conocimiento al cuál se le asocian las horas.
<i>type_of_mapping</i>	<i>VARCHAR(n)</i>	Tipo de relación entre la dedicación y el ítem. El valor <b>NORMAL</b> indica que las horas mapeadas aplican directamente al ítem de conocimiento, <b>VA</b> (Varios) representa que las horas se asocian con todos los ítems hijos del ítem referenciado y por último el valor <b>NG</b> (Ninguno) significa que las horas mapeadas se asocian con el ítem referenciado pero no lo hacen con ninguno de sus hijos.
<i>hours_quantity</i>	<i>REAL</i>	Cantidad de horas de la dedicación que corresponden al ítem de conocimiento referenciado.

Cuadro 34: Tabla *knowledge\_item\_hours**Tabla knowledge\_items*

En la tabla *knowledge\_items* se tiene la información de los ítems del cuerpo de conocimiento.

Nombre Columna	Tipo de datos	Descripción
<i>id</i>	<i>BIGINT</i>	Clave primaria.
<i>created_at</i>	<i>TIMESTAMP</i>	Fecha de creación.
<i>modified_at</i>	<i>TIMESTAMP</i>	Fecha de última modificación.
<i>code</i>	<i>VARCHAR(n)</i>	Código del ítem de conocimiento.
<i>name</i>	<i>VARCHAR(n)</i>	Nombre del ítem de conocimiento.
<i>knowledge_level_id</i>	<i>BIGINT</i>	Clave foránea que referencia a la tabla <i>knowledge_levels</i> , uniendo al ítem con el nivel al que pertenece.
<i>parent_id</i>	<i>BIGINT</i>	Clave foránea que referencia al ítem padre del ítem de conocimiento.

Cuadro 35: Tabla *knowledge\_items**Tabla knowledge\_levels*

En la tabla *knowledge\_levels* se tiene la información relacionada a los niveles del cuerpo de conocimiento.

Nombre Columna	Tipo de datos	Descripción
<i>id</i>	<i>BIGINT</i>	Clave primaria.
<i>created_at</i>	<i>TIMESTAMP</i>	Fecha de creación.
<i>modified_at</i>	<i>TIMESTAMP</i>	Fecha de última modificación.
<i>name</i>	<i>VARCHAR(n)</i>	Nombre del nivel de conocimiento.
<i>level_number</i>	<i>INTEGER</i>	Valor numérico que se le asocia al nivel de acuerdo a su jerarquía en el cuerpo. El nivel de mayor valor de abstracción tendrá el valor numérico más bajo.
<i>body_of_knowledge_id</i>	<i>BIGINT</i>	Clave foránea que referencia a la tabla <i>bodies_of_knowledge</i> con la clave primaria del cuerpo de conocimiento al cual pertenece.

Cuadro 36: Tabla *knowledge\_levels**Tabla minimum\_hour\_criteria\_leaf\_level*

La tabla *minimum\_hour\_criteria\_leaf\_level* contiene para un perfil de configuración, el criterio de horas mínimas definido la evaluar si los ítems hojas del cuerpo están cubiertos.

Nombre Columna	Tipo de datos	Descripción
<i>id</i>	<i>BIGINT</i>	Clave primaria.
<i>created_at</i>	<i>TIMESTAMP</i>	Fecha de creación.
<i>modified_at</i>	<i>TIMESTAMP</i>	Fecha de última modificación.
<i>hours_quantity</i>	<i>REAL</i>	Cantidad mínima de horas, de los tipos de hora seleccionados, que deben estar asociadas al ítem para considerarlo cubierto.
<i>types_of_hour</i>	<i>BIGINT</i>	Tipos de horas a considerar. Se almacena como el producto de los números primos que identifican a los tipos de hora disponibles en el cuadro 20.
<i>configuration_profile_id</i>	<i>BIGINT</i>	Clave foránea que referencia a la tabla <i>configuration_profiles</i> , conecta el criterio con el perfil de configuración al cual pertenece.

Cuadro 37: Tabla *minimum\_hour\_criteria\_leaf\_level*

#### Tablas *oauth\_access\_token* y *oauth\_client\_details*

Las tablas *oauth\_access\_token* y *oauth\_client\_details* contienen la información necesaria para la implementación del estándar *OAuth* utilizando el *framework Spring Security* <sup>51</sup>. Se opta por no incluir la definición de estas tablas debido a que son definidas y manejadas por el módulo del *framework* y no deben ser alteradas por ningún usuario.

#### Tabla *people*

En la tabla *people* se tiene la información de todas las personas registradas, sean del tipo usuarios del sistema o no. Esta entidad tiene mayor jerarquía ya que guarda la información común a los dos tipos, y luego, la información específica de los usuarios se almacena en la tabla *users*.

<sup>51</sup> <http://projects.spring.io/spring-security-oauth/docs/oauth2.html>

Nombre Columna	Tipo de datos	Descripción
<i>person_id</i>	<i>BIGINT</i>	Clave primaria.
<i>created_at</i>	<i>TIMESTAMP</i>	Fecha de creación.
<i>modified_at</i>	<i>TIMESTAMP</i>	Fecha de última modificación.
<i>name</i>	<i>VARCHAR(255)</i>	Nombre completo de la persona.
<i>email</i>	<i>VARCHAR(50)</i>	Email asociado a la persona.

Cuadro 38: Tabla *people**Tabla plan\_implementations*

La tabla *plan\_implementations* contiene la información que refiere a las implementaciones de un plan de estudio.

Nombre Columna	Tipo de datos	Descripción
<i>id</i>	<i>BIGINT</i>	Clave primaria.
<i>created_at</i>	<i>TIMESTAMP</i>	Fecha de creación.
<i>modified_at</i>	<i>TIMESTAMP</i>	Fecha de última modificación.
<i>code</i>	<i>VARCHAR(n)</i>	Código de la implementación.
<i>name</i>	<i>VARCHAR(n)</i>	Nombre de la implementación.
<i>study_programme_id</i>	<i>BIGINT</i>	Clave foránea que referencia a la tabla <i>study_programmes</i> , conectando la implementación con el plan de estudio al cual pertenece.

Cuadro 39: Tabla *plan\_implementations**Tabla study\_programmes*

La tabla *study\_programmes* contiene la información de los planes de estudio.

Nombre Columna	Tipo de datos	Descripción
<i>id</i>	<i>BIGINT</i>	Clave primaria.
<i>created_at</i>	<i>TIMESTAMP</i>	Fecha de creación.
<i>modified_at</i>	<i>TIMESTAMP</i>	Fecha de última modificación.
<i>code</i>	<i>VARCHAR(255)</i>	Código del plan de estudio.
<i>name</i>	<i>VARCHAR(n)</i>	Nombre del plan de estudio.
<i>credits</i>	<i>INTEGER</i>	Total de créditos asignados al plan de estudio.
<i>expected_hours</i>	<i>INTEGER</i>	Horas totales esperadas para completar el plan.
<i>educational_objectives</i>	<i>BYTEA</i>	Objetivos educativos del plan de estudio.
<i>expected_knowledge_before_start</i>	<i>BYTEA</i>	Conocimientos esperados del alumno antes de comenzar el plan.
<i>expected_knowledge_after_graduating</i>	<i>BYTEA</i>	Conocimientos esperados del alumno después de finalizar el plan de estudio.
<i>months_duration</i>	<i>INTEGER</i>	Cantidad de meses de duración.
<i>career_id</i>	<i>BIGINT</i>	Clave foránea que referencia a la tabla <i>careers</i> . Relaciona el plan de estudio con la carrera que lo define.

Cuadro 40: Tabla *study\_programmes**Tabla themes*

La tabla *themes* contiene la información sobre los temas que se dictan en las ediciones de cursos.

Nombre Columna	Tipo de datos	Descripción
<i>id</i>	<i>BIGINT</i>	Clave primaria.
<i>created_at</i>	<i>TIMESTAMP</i>	Fecha de creación.
<i>modified_at</i>	<i>TIMESTAMP</i>	Fecha de última modificación.
<i>name</i>	<i>VARCHAR(n)</i>	Nombre del tema.

Cuadro 41: Tabla *themes**Tabla type\_of\_hour\_configuration*

La tabla *type\_of\_hour\_configuration* contiene las ponderaciones que se utilizan para cada tipo de hora. Para cada relación entre un plan de estudio y un cuerpo de conocimiento, se define el peso que tiene cada tipo de hora. Luego, este peso funciona como coeficiente para calcular las horas dedicadas al realizar la evaluación.

Nombre Columna	Tipo de datos	Descripción
<i>id</i>	<i>BIGINT</i>	Clave primaria.
<i>type_of_hour</i>	<i>INTEGER</i>	Valor numérico que representa uno de los tipos de horas definidos en el sistema.
<i>weight</i>	<i>REAL</i>	Ponderancia o peso que se asigna al tipo de hora.

Cuadro 42: Tabla *type\_of\_hour\_configuration**Tabla users*

La tabla *users* guarda los atributos específicos de la entidad usuario.

Nombre Columna	Tipo de datos	Descripción
<i>person_id</i>	<i>BIGINT</i>	Clave primaria, referencia a la tabla <i>people</i>
<i>username</i>	<i>VARCHAR(255)</i>	Nombre de usuario utilizado para ingresar al sistema.
<i>password</i>	<i>VARCHAR(500)</i>	Contraseña elegida por el usuario. Es guardada en forma encriptada en la base de datos.

Cuadro 43: Tabla *users**Tabla users\_authorities*

La tabla *users\_authorities* contiene los permisos asociados a cada usuario. Los roles definidos en el sistema son *ROLE\_ADMIN* y *ROLE\_SUPER\_ADMIN*. El primer rol permite realizar todas las operaciones del sistema menos las funcionalidades relacionadas a la gestión de usuarios. El acceso a estas funcionalidades es exclusivo para aquellos usuarios del sistema que poseen la autoridad *ROLE\_SUPER\_ADMIN*. El sistema es extensible a la definición de nuevas autoridades, insertando nuevos valores en la tabla *authorithies* y realizando las modificaciones correspondientes en el código de los controladores de peticiones de recursos.

Nombre Columna	Tipo de datos	Descripción
<i>users_person_id</i>	<i>BIGINT</i>	Clave foránea, referencia a la tabla <i>users</i> .
<i>authorities_id</i>	<i>BIGINT</i>	Clave foránea, referencia a la tabla <i>authorities</i> .

Cuadro 44: Tabla *users\_authorities**Tabla users\_careers*

La tabla *users\_careers* mantiene la relación entre las carreras y sus respectivos responsables.

Nombre Columna	Tipo de datos	Descripción
<i>users_person_id</i>	<i>BIGINT</i>	Clave foránea, referencia a la tabla <i>users</i> .
<i>career_id</i>	<i>BIGINT</i>	Clave foránea, referencia a la tabla <i>careers</i> .

Cuadro 45: Tabla *users\_careers*

## B.2 CÓDIGO PARA INSERTAR ELEMENTOS

En esta sección se incluye código *SQL* para ingresar la información que no puede ser ingresada a través de la herramienta, debido a que sus respectivas funcionalidades quedaron por fuera del alcance del proyecto. En este sentido, las siguientes entidades se deben manejar a través de la base de datos: cuerpos de conocimiento, carreras, planes de estudio y perfiles de configuración.

En el siguiente código se presentan las inserciones necesarias para ingresar un cuerpo de conocimiento en el sistema. A modo de ejemplo, se utilizan datos del cuerpo de conocimiento CBOK. Por detalles de implementación, se debe crear un área de conocimiento con código Ninguna para representar que una dedicación no se corresponde a ningún elemento del cuerpo.

```
-- Cuerpos de conocimiento

INSERT INTO bodies_of_knowledge (id, description, name)
VALUES (1, 'CBOK description', 'CBOK');

-- CBOK Niveles

INSERT INTO knowledge_levels (id, level_number, name, body_of_knowledge_id)
VALUES (3,1, 'Área de conocimiento', 1);

INSERT INTO knowledge_levels (id, level_number, name, body_of_knowledge_id)
VALUES (2,2, 'Unidad', 1);

INSERT INTO knowledge_levels (id, level_number, name, body_of_knowledge_id)
VALUES (1,3, 'Tópico', 1);

-- CBOK Items
```

```
INSERT INTO knowledge_items (id,created_at,modified_at,code,name,knowledge_
    level_id,parent_id)
VALUES (1,NOW(),NOW(),'Ninguna','Ninguna',3,NULL);

INSERT INTO knowledge_items (id,created_at,modified_at,code,name,knowledge_
    level_id,parent_id)
VALUES (2,NOW(),NOW(),'A','Ética y Conducta Profesional',3,NULL);

INSERT INTO knowledge_items (id,created_at,modified_at,code,name,knowledge_
    level_id,parent_id)
VALUES (13,NOW(),NOW(),' 1','Cuestiones sociales, legales e históricas',2,2)
;

INSERT INTO knowledge_items (id,created_at,modified_at,code,name,knowledge_
    level_id,parent_id)
VALUES (14,NOW(),NOW(),' 2','Códigos de ética y conducta profesional',2,2);

INSERT INTO knowledge_items (id,created_at,modified_at,code,name,knowledge_
    level_id,parent_id)
VALUES (15,NOW(),NOW(),' 3','La naturaleza y el rol de los estándares de la
    ingeniería de software',2,2);
```

El siguiente bloque incluye el código necesario para ingresar carreras y planes de estudio.

```
-- Carreras
INSERT INTO careers (id,code,director,name)
VALUES (1,'MIS','Diego Vallespir','Maestria en ingenieria de software');
--Plan de estudio

INSERT INTO study_programmes (code,credit,months_duration,expected_hours,
    name,career_id)
VALUES ('MIS-Plan',110,24,1650,'Plan de estudio de la maestría en ingeniería
    de software',1);
```

En el siguiente bloque de código se presentan las inserciones necesarias para configurar los perfiles de consultas, los tipos de hora y asociar planes de estudio con cuerpo de conocimiento.

```
-- Asociar Cuerpo a Plan de estudio
INSERT INTO bodies_of_knowledge_study_programmes (id, body_of_knowledge_id,
    study_programme_id) VALUES (1, 1,1);
```

```
-- Especificar ponderación por tipo de hora.
INSERT INTO public.type_of_hour_configuration(id, type_of_hour, weight)
VALUES (1, 2, 2),(2, 3, 1),(3, 5, 1),(4, 7, 1),(5, 11, 1),(6, 13, 1),(7,
      17, 1);

-- Establecer las configuraciones de tipo de hora por dupla conformada por
      cuerpo de conocimiento y plan de estudio.
INSERT INTO public.bodies_of_knowledge_study_programmes_hours_configuration(
      bodies_programmes_id, hours_configuration_id)
VALUES (1, 1),(1, 2),(1, 3),(1, 4),(1, 5),(1, 6),(1, 7);

--- Perfil

-- Crear perfil de configuración
INSERT INTO public.configuration_profiles(id, created_at, modified_at, hour_
      configuration, name, include_weighing)
VALUES (1, now(), now(), 30030, 'Configuración para CBOK utilizando todos
      los tipos de hora', true);

--Asociar perfil de configuración con planes de estudio
INSERT INTO public.configuration_profiles_study_programmes(configuration_
      profile_id, study_programme_id)
VALUES (1, 1);

-- Criterio horas minimas
INSERT INTO public.minimum_hour_criteria_leaf_level(id, configuration_
      profile_id, hours_quantity, types_of_hour)
VALUES (1, 1, 0.01, 30030);

-- Franjas de cubrimiento
INSERT INTO public.coverage_ranges(id, name, minimum, maximum, color)
VALUES (1, 'No cubierto', 0, 15, '#FF8000'),
      (2, 'Cubierto Parcialmente', 15, 50, '#FFFF00'),
      (3, 'Cubierto Ampliamente', 50, 85, '#ACFA58'),
      (4, 'Totalmente Cubierto', 85, 100, '#01DF3A');

-- Porcentaje de cubrimiento de hijos mínimos para considerar al ítem como
      cubierto
```

```
INSERT INTO public.configuration_profiles_parent_level(id, configuration_
  profile, knowledge_level_id, minimum_coverage_percentage, created_at,
  modified_at)
VALUES (1, 1, 2, 15, now(), now()),(2, 1, 3, 15, now(), now());

-- Asocia franjas de cubrimiento con niveles de cuerpo de conocimiento.
INSERT INTO public.configuration_profiles_parent_level_coverage_ranges(
  configuration_profile_level_id, coverage_range_id)
VALUES (1, 1), (1, 2), (1,3), (1,4), (2, 1), (2, 2), (2, 3), (2, 4);

-- Establecer para cuáles niveles se puede definir valores esperados de
  esfuerzo y balance.
INSERT INTO public.configuration_profiles_levels_effort_and_balance_applies(
  id, knowledge_level, configuration_profile, applies)
VALUES (1, 1, 1, FALSE),(2, 2, 1, FALSE),(3, 3, 1, TRUE);

Esfuerzo
-- Configuración de valores esperados de esfuerzo y balance mínimos
INSERT INTO public.configuration_profiles_knowledge_items(id, configuration_
  profile_id, knowledge_item_id, effort_minimum_hours, effort_maximum_
  hours,
  balance_minimum_percentage, balance_maximum_percentage)
VALUES (1, 1, 1, 258, 344, 0, 0),
(2, 1, 2, 12, 16, 2, 4),
(3, 1, 3, 18, 24, 4, 6),
(4, 1, 4, 48, 64, 12, 16),
(5, 1, 5, 66, 88, 18, 22),
(6, 1, 6, 18, 24, 2, 6),
(7, 1, 7, 36, 48, 8, 12),
(8, 1, 8, 24, 32, 6, 8),
(9, 1, 9, 18, 24, 4, 6),
(10, 1, 10, 54, 72, 14, 18),
(11, 1, 11, 24, 32, 6, 8),
(12, 1, 12, 24, 32, 6, 8);
```



## MANUAL DE INSTALACIÓN

---

Este manual tiene como propósito brindar las instrucciones necesarias para instalar y configurar el sistema Plataforma Evaluadora de Carreras.

En la sección C.1 se describe la máquina virtual entregada que posee el sistema Plataforma de Evaluación de Carreras instalado y disponible para su utilización, junto a las configuraciones e instrucciones necesarias para inicializar la máquina virtual en el software de virtualización.

En la sección C.2 se describen los requerimientos e instrucciones que fueron seguidos para crear la máquina virtual entregada y que son necesarios si se desea realizar la instalación en un sistema operativo Ubuntu. Para esta sección se asume que el lector está familiarizado con conceptos básicos de administración de sistemas Linux y que posee un usuario con permiso de administrador en el sistema sobre el cual se desea realizar la instalación.

### C.1 MÁQUINA VIRTUAL

Se entrega una máquina virtual con el sistema previamente instalado, con las siguientes características:

- Sistema operativo: Ubuntu 16.04.4 LTS de 64 bits
- Memoria RAM: 2048MB
- Disco duro: 20GB.
- Credenciales de Usuario
  - Nombre de usuario: evaluacioncarrera
  - Contraseña: admin1234

Se utiliza VirtualBox <sup>52</sup> como software de virtualización del sistema. En el enlace de descargas <sup>53</sup> se encuentran las versiones disponibles del programa. Se recomienda descargar la última versión, y seguir las instrucciones de instalación <sup>54</sup>.

---

<sup>52</sup> <https://www.virtualbox.org/>

<sup>53</sup> <https://www.virtualbox.org/wiki/Downloads>

<sup>54</sup> <https://download.virtualbox.org/virtualbox/5.2.10/UserManual.pdf>

La máquina virtual que se entrega es un archivo con extensión .ova (Open Virtual Appliance) <sup>55</sup>. Para completar la instalación e importación de la máquina virtual desde el archivo se pueden seguir las instrucciones *Importing an Existing Virtual Machine into VirtualBox* <sup>56</sup> disponibles en la página de Oracle, o abrir el archivo con el programa VirtualBox. Para el segundo caso, se abre una ventana como la disponible en la figura 27. En dicha pantalla se debe seleccionar la opción importar, y luego aceptar el resto de los pasos.

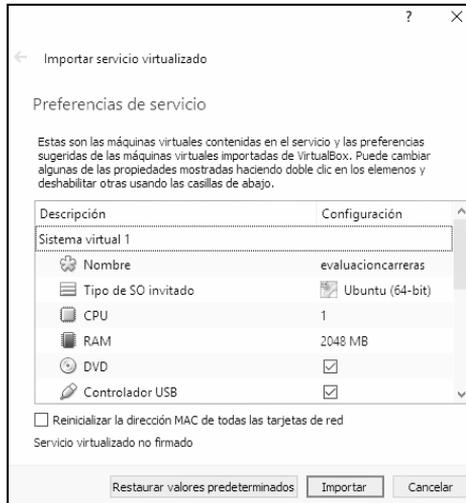


Figura 27: Pantalla de importación de una máquina virtual.

Si al iniciar la máquina aparece una pantalla de error con el mensaje que se ilustra en la figura 28, se debe seleccionar la opción cambiar preferencias de red.

<sup>55</sup> <https://www.file-extension.info/es/format/ova>

<sup>56</sup> [https://docs.oracle.com/cd/E26217\\_01/E26796/html/qs-import-vm.html](https://docs.oracle.com/cd/E26217_01/E26796/html/qs-import-vm.html)

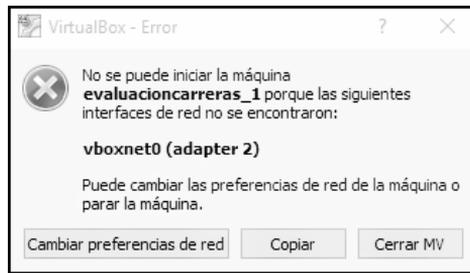


Figura 28: Mensaje de error sobre preferencias de red.

Al seleccionar esta opción se muestra la pantalla de configuración de red de la figura 29, en ella se debe seleccionar la opción *VirtualBox Host-Only Ethernet Adapter* dentro del panel llamado «Adaptador 2». Luego se debe hacer *click* en el botón aceptar.

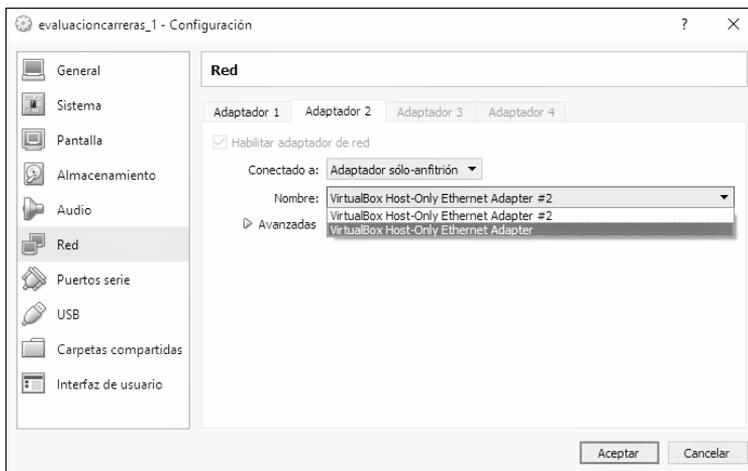


Figura 29: Panel de preferencias de red.

Luego de actualizar la configuración de red se debe reiniciar la máquina e iniciar sesión. El sistema de evaluación de carreras es levantado automáticamente, mostrando un mensaje de éxito como el de la figura 30.

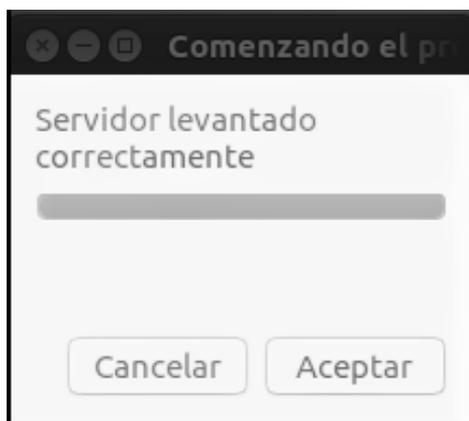


Figura 30: Mensaje de éxito al levantar el sistema.

La primera vez que se inicia sesión en la máquina virtual puede ocurrir un error al levantarse el sistema, por lo cual, si el mensaje que se muestra es distinto a «Servidor levantado correctamente», se debe hacer doble *click* sobre el ícono del escritorio «Parar Pec». Una vez que el sistema fue detenido exitosamente, ejecutar «Correr Pec» y esperar unos minutos a que el sistema se ponga en funcionamiento hasta obtener el mensaje de éxito indicado anteriormente. En la figura 31 se puede observar el escritorio de la máquina virtual, con los íconos antes mencionados.

El sistema de evaluación de carreras queda disponible en <http://localhost:9000>, y puede ser accedido tanto desde el interior de la máquina virtual, como desde la computadora dónde corre la máquina. Se recomienda utilizar el sistema en esta última modalidad, dadas las limitaciones de espacio que presenta la máquina virtual. A su vez, si las configuraciones de red lo permiten, el sistema podrá ser accedido desde otros dispositivos conectados a la misma red a través de la IP de la máquina que contiene la máquina virtual.

Cada vez que se inicie sesión en la máquina virtual, el sistema es levantado automáticamente, y restaura el estado de la última utilización. El sistema ya se encuentra inicializado, es decir, su base de datos está creada y cuenta con la información del cubrimiento disponible hasta el momento.



Figura 31: Escritorio de la máquina virtual una vez que se inicia sesión.

Los íconos del escritorio permiten realizar las siguientes acciones sobre el sistema:

- **Correr Pec:** Al ejecutar este ícono se pone en funcionamiento el sistema.
- **Inicializar / Re inicializar Pec :** Si se ejecuta este ícono, el sistema volverá a su estado base, es decir, este comando borra la base de datos actual del sistema, crea una nueva y realiza las inserciones necesarias de los elementos utilizados.
- **Parar Pec:** Detiene la ejecución del sistema.

Los comandos mencionados anteriormente son necesarios sólo en casos excepcionales, como por ejemplo, que el sistema deje de funcionar o se necesite borrar la base de datos para volver al estado original.

En la carpeta *logs* del Escritorio se encuentra un archivo con la salida del sistema.

Se puede acceder a la base de datos utilizando *PgAdmin* que se encuentra instalado en la máquina virtual con las siguientes credenciales:

- **Usuario de Postgres:** evaluacioncarrera
- **Contraseña:** admin1234

El sistema de evaluación de carreras tiene definidos los usuarios «Admin» y «Super Admin», con los siguientes nombre de usuario y contraseña respectivamente:

- **Admin:** admin / admin1234
- **Super Admin:** superAdmin / superAdmin1234

## C.2 INSTALACIÓN DEL SISTEMA EN SISTEMA OPERATIVO UBUNTU

El sistema está compuesto por tres grandes artefactos técnicos, el servidor de *Front End*, el servidor de *Back End* y el servidor de base de datos, los cuales podrían distribuirse en distintas máquinas. En el contexto de este manual se describen las configuraciones necesarias para localizar todos en una única máquina con las siguientes características:

- Sistema operativo: Ubuntu 16.04.4 LTS de 64 bits.
- Memoria RAM: 2048MB.

### C.2.1 Instalación de la base de datos

El sistema utiliza una base de datos *PostgreSQL*<sup>57</sup>. Se definen a continuación los pasos para instalarla:

1. Ir a la página de descargas de Postgres <sup>58</sup>.
2. Instalar Postgres ejecutando desde la terminal el comando:

```
sudo apt-get install postgresql-9.5
```

<sup>57</sup> <https://www.virtualbox.org/>

<sup>58</sup> <https://www.postgresql.org/download/>

3. Crear un usuario PostgreSQL con los permisos necesarios, con el siguiente comando:

```
sudo -u postgres createuser evaluacioncarrera -P -s
```

- El parámetro `-s` indica que el usuario tiene los permisos de un usuario super admin, puede crear roles y base de datos.
- El parámetro `-P` habilita un formulario para ingresar la contraseña del usuario.
- `evaluacioncarrera` representa el nombre del usuario seleccionado.

Para validar que el usuario se crea de forma exitosa abrir la terminal `psql` ejecutando:

```
sudo -u postgres psql
```

Desde la terminal `psql`, ejecutar el siguiente comando para listar todos los usuarios Postgres del sistema:

```
\du
```

El resultado del comando debe ser como el que se muestra a continuación.

```
postgres=# \du
                List of roles
Role name | Attributes | Member of
-----+-----+-----
evaluacioncarrera | Superuser, Create role, Create DB | {}
postgres | Superuser, Create role, Create DB, Replication, Bypass RLS | {}
postgres=#
```

Figura 32: Resultado esperado de la lista usuarios y permisos.

#### C.2.1.1 Instalación del cliente de datos

Se recomienda la utilización del cliente de base de datos *Postgres PgAdmin3*<sup>59</sup>. El siguiente comando se ejecuta para instalar *PgAdmin3* en la máquina virtual:

```
sudo apt-get install pgadmin3
```

Luego de la instalación, se debe crear la conexión a la base de datos utilizada por el sistema, ingresando la configuración que se muestra en la figura 33.

<sup>59</sup> <https://www.pgadmin.org/>

The image shows a 'New Server Registration' dialog box with the following fields and values:

- Name: Proyecto
- Host: localhost
- Port: 5432
- Service: (empty)
- Maintenance DB: postgres
- Username: evaluacioncarrera
- Password: (masked with dots)
- Store password:
- Colour: (empty)
- Group: Servers

Buttons at the bottom: Avuda, Aceptar, Cancelar.

Figura 33: Configuración para crear la conexión a la base de datos.

### c.2.2 Instalación del Back End

Para configurar el servidor de *Back End*, el ambiente debe contar con Java y Maven instalados. A continuación se explican los pasos a seguir para contemplar estos requerimientos.

- Instalar Java JDK 8, ejecutando el siguiente comando<sup>60</sup>:

```
sudo apt-get install openjdk-8-jre
```

- Instalar el manejador para la construcción y gestión de paquetes *Maven* con el siguiente comando:

```
sudo apt-get install maven
```

Una vez instalados Java y Maven, se deben establecer las siguientes variables de entorno:

<sup>60</sup> <http://openjdk.java.net/install/>

- **DB\_USERNAME** : Nombre de usuario para PostgreSQL, definido en la creación del usuario Postgres.
- **DB\_PASSWORD** : Contraseña definida para PostgreSQL.

Para configurar las variables de entorno en la máquina virtual se debe abrir el archivo *back-end-env* y asociarle valores a las variables antes mencionadas, de la siguiente manera:

```
DB_USERNAME=Nombre de usuario postgres
DB_PASSWORD=Contraseña de usuario
```

### c.2.3 Instalación del Front End

- Instalar *curl* <sup>61</sup> ejecutando el siguiente comando:

```
sudo apt install curl
```

- Instalar el manejador de versiones de Node, *NVM* <sup>62</sup> ejecutando el siguiente comando:

```
curl -o- https://raw.githubusercontent.com/creationix/nvm/v
0.33.9/install.sh | bash command -v nvm
```

- Luego, reiniciar la terminal e instalar la última versión estable de *Node* <sup>63</sup>. En el momento de la redacción del manual, la versión estable (*long term support*) es la 8. Para completar este paso ejecutar el siguiente comando:

```
nvm install 8
```

### c.2.4 Inicializar el sistema

Para poner el sistema en marcha, la máquina donde se desea realizar la instalación debe contar con los archivos ejecutables (*\*.sh*) que se entregan junto al código del proyecto:

<sup>61</sup> <http://openjdk.java.net/install/>

<sup>62</sup> <https://github.com/creationix/nvm>

<sup>63</sup> <https://nodejs.org/es/>

- *inicializar-pec.sh*: Ejecutable que levanta el sistema, junto a la base de datos y los datos iniciales. Debe utilizarse para iniciar por primera vez el sistema, o en caso que se desee volver al estado inicial de los datos.
- *correr-pec.sh*: Levanta el sistema, sin restaurar los datos de la base de datos. Una vez que el sistema se encuentra inicializado, debe ejecutarse este archivo para los subsiguientes usos.

Una vez que se realizaron las instalaciones y configuraciones mencionados en las secciones previas, se debe ejecutar el archivo *inicializar-pec.sh*. Antes de ejecutar cualquiera de los archivos, es necesario modificarlos, actualizando la ruta dónde se encuentra el código del proyecto. Al inicializar el sistema, el servidor de *Front End* queda disponible en <http://localhost:9000> y el servidor de *Back End* en <http://localhost:8080/>. Por último, para futuros usos del sistema, se debe ejecutar el archivo *correr-pec.sh*.