UNIVERSIDAD DE LA REPÚBLICA URUGUAY

FACULTAD DE INGENIERÍA

Predicción de la ejecución de procesos de negocio colaborativos inter-organizacionales

Informe de Proyecto de Grado presentado por

Ana Carolina Espino y Nicolás Ribero

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

Supervisores

Andrea Delgado
Daniel Calegari

Montevideo, 9 de diciembre de 2024

Resumen

Los procesos de negocio son un conjunto de actividades realizadas en coordinación para alcanzar un objetivo de negocio, cuya ejecución en sistemas de información tradicionales o basados en procesos, permite obtener una variedad de datos para su evaluación y mejora.

Las técnicas de minería de procesos permiten realizar análisis complejos de la ejecución real de los procesos, proveyendo a las organizaciones información valiosa sobre eficiencia, calidad o cumplimiento de normativas, para descubrir oportunidades de mejora basada en evidencia. Este tipo de análisis retrospectivo es muy útil para las organizaciones sentando la base para la mejora de los procesos basada en evidencia, aportando elementos para la toma de decisiones con base en la operativa diaria.

Sin embargo, más allá del análisis retrospectivo, las organizaciones están mostrando un creciente interés en la capacidad de realizar predicciones basadas en los datos históricos de ejecución de procesos. Utilizando técnicas avanzadas de minería de procesos, es posible anticipar posibles desviaciones, violaciones o demoras en los procesos en tiempo real, lo que permite la implementación de medidas preventivas, como la reasignación de recursos o la optimización de tiempos. Estas predicciones se basan en el análisis de las trazas registradas en los logs de eventos, lo que representa un enfoque proactivo para gestionar la ejecución de los procesos.

La minería de procesos sobre los datos de ejecución de procesos de negocio se ha enfocado principalmente en procesos de tipo orquestación que se realizan en una única organización (intra-organizacionales) y no en procesos colaborativos (inter-organizacionales).

El objetivo principal de este proyecto es analizar técnicas/enfoques existentes para la predicción de la ejecución de procesos de negocio con minería de procesos y definir/extender/adaptar a la predicción de procesos de negocio colaborativos.

Se propone una solución que extiende un enfoque existente para procesos no colaborativos, que permite a partir de un log de eventos extendido para procesos colaborativos, poder hacer predicciones relevantes a este contexto.

Para finalizar, se evalúa la aplicación sobre un conjunto de logs de procesos colaborativos existente en la comunidad obteniendo resultados satisfactorios.

Palabras clave: Minería de procesos, Monitoreo Predictivo de Procesos de Negocio, Proceso colaborativo

Índice general

Capítulo 1

Introducción

Los procesos de negocio son un conjunto de actividades realizadas en coordinación para alcanzar un objetivo de negocio (Weske, 2019), cuya ejecución en sistemas de información tradicionales o basados en procesos, genera una amplia variedad de datos útiles para su evaluación y mejora. Un proceso colaborativo involucra dos o más organizaciones que actúan de forma coordinada para llevar a cabo distintas partes de un proceso. Para lograr esta coordinación intercambian mensajes que permiten sincronizar sus acciones.

La minería de procesos (W. van der Aalst, 2016) es una disciplina dentro de la Ciencia de Datos que se basa en técnicas de minería de datos para analizar los logs de eventos asociados a la ejecución de procesos. Un área de análisis dentro de la minería de procesos es la mejora de procesos (W. van der Aalst y cols., 2012) dentro de la cual se incluye el monitoreo predictivo de procesos de negocio. Esta rama busca anticipar el comportamiento futuro de los procesos de negocio mediante el análisis de datos históricos y en tiempo real a través de técnicas avanzadas de minería de procesos, aprendizaje automático y modelos predictivos. Sin embargo, esta disciplina se ha enfocado principalmente en procesos que se realizan en una única organización (intra-organizacionales) y no en procesos colaborativos donde participan dos o más organizaciones (inter-organizacionales). Además, las soluciones actuales no incorporan elementos ni métodos colaborativos explícitos que faciliten su extensión a estos escenarios.
El objetivo principal de este proyecto es analizar técnicas/enfoques existentes para la predicción de la ejecución de procesos de negocio con minería de procesos y definir/extender/adaptar a la predicción de procesos de negocio colaborativos. Para ello, se propone una solución que amplía las capacidades de predicción de la herramienta ProcessTransformer para escenarios colaborativos. La solución se evaluará utilizando logs de eventos extendidos para procesos colaborativos y el desarrollo se realizará en Python, junto con el microframework Flask para el desarrollo de la aplicación web.

1.1. Motivación y objetivos

La motivación de este proyecto es la no existencia de propuestas que permitan realizar predicciones en el contexto de procesos de negocio colaborativos. Es por esta razón, que el objetivo general de este proyecto es desarrollar o extender una herramienta que permita realizar predicciones en procesos colaborativos.

Los objetivos específicos del proyecto son:

1. Estudiar y analizar técnicas, algoritmos, herramientas y propuestas existentes para predicción de la ejecución de procesos de negocio con minería de procesos

2. Generar propuesta/extensión de minería de procesos para la predicción de la ejecución de procesos de negocio colaborativos

3. Desarrollar/extender/adaptar herramienta prototipo de soporte a la propuesta

4. Evaluar la aplicabilidad de la propuesta a través de casos de estudio.

1.2. Aportes del proyecto

1. Definición de posibles tipos de predicción para procesos colaborativos.

2. Ampliación las capacidades de predicción de la herramienta ProcessTransformer para escenarios colaborativos.

3. Aplicación web que permite realizar predicciones para procesos colaborativos.

1.3. Contexto de trabajo

Este proyecto de grado se enmarca en el proyecto de investigación financiado por el Fondo María Viñas (FMV) de ANII 2021 “Minería de procesos y datos para la mejora de procesos colaborativos aplicada a e-Government” del grupo COAL del INCO.

1.4. Estructura documento

A continuación, se explica cómo está estructurado el resto del informe:

- Capítulo 2 - Marco teórico: describe de forma introductoria los conceptos de proceso de negocio, proceso colaborativo, minería de procesos, Log de eventos y monitoreo predictivo de procesos

- Capítulo 3 - Estado del arte: abarca la investigación realizada sobre el tema, incluyendo las metodologías empleadas para la búsqueda, los documentos clave encontrados y un resumen de algunas herramientas existentes.
- Capítulo 4 - Problemática y propuesta de solución: se describe la problemática que se quiere abordar y se plantea la posible solución/extensión.
- Capítulo 5 - Implementación de la solución: se detalla la solución implementada
- Capítulo 6 - Evaluación: se evalúa la solución desarrollada en profundidad para un caso de estudio. Además se realiza una evaluación comparativa para distintos logs de eventos.
- Capítulo 7 - Conclusiones y Trabajo Futuro: se exponen las conclusiones del proyecto junto con propuestas para futuros trabajos.

Capítulo 2

Marco teórico

2.1. Proceso de negocio

Un proceso de negocio consiste en un conjunto de actividades que se realizan de forma coordinada en un entorno organizacional y técnico. Estas actividades realizan conjuntamente un objetivo empresarial. Cada proceso de negocio es ejecutado por una única organización, pero puede interactuar con procesos de negocio ejecutados por otras organizaciones (Weske, 2019).
El ciclo de vida de los procesos de negocio consta de las siguientes fases interrelacionadas (ver Figura 2.1):

- **Diseño y Análisis:** Se identifican, validan y representan los procesos mediante modelos, utilizando herramientas de modelado y simulación para validar y mejorar los procesos.

- **Configuración:** Se configura el sistema y se realizan pruebas de integración y rendimiento para asegurar que el sistema cumpla con los requisitos.

- **Ejecución:** Los procesos se ejecutan para cumplir los objetivos de negocio. Los sistemas de gestión de procesos controlan y monitorean su ejecución, recopilando datos para su evaluación.

- **Evaluación:** Se analizan los datos de ejecución con técnicas de monitoreo y minería de procesos para mejorar los modelos de negocio.

- **Administración y Participantes:** Se gestionan los artefactos del proceso, como modelos y datos, y se organiza el conocimiento de los participantes y el entorno tecnológico.

Este ciclo permite gestionar, monitorear y mejorar continuamente los procesos de negocio en un entorno dinámico.

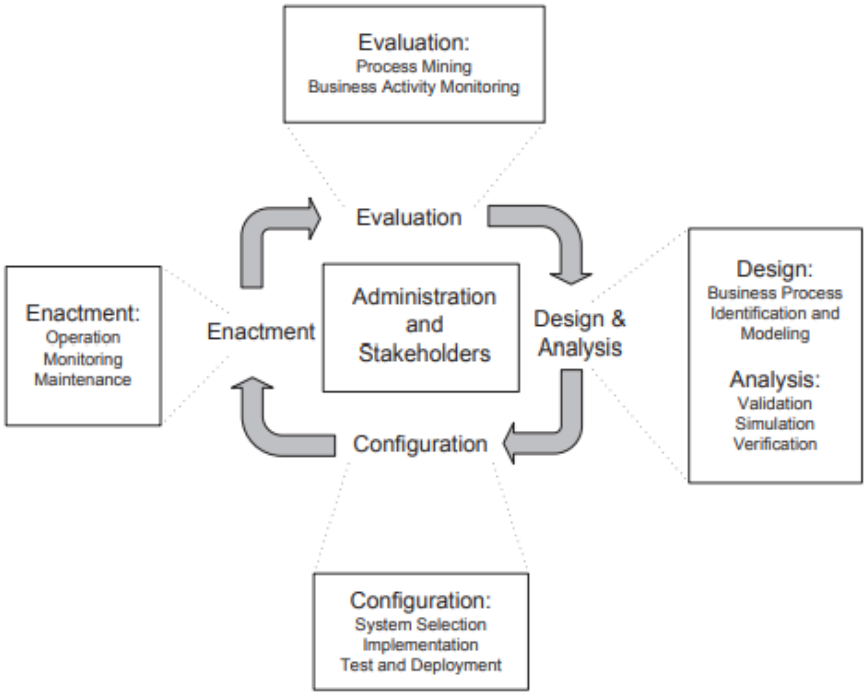


Figura 2.1: Ciclo de vida de un proceso de negocio (Weske, 2019)

2.2. Proceso colaborativo

La situación que se muestra en la Figura 2.2 se denomina proceso colaborativo, donde dos o más participantes interactúan para llegar a un objetivo común donde a su vez, cada uno de ellos tiene su propio proceso de negocio. Existen cinco formas de interactuar (W. M. P. van der Aalst, 2011):

- Capacidad compartida, el control del proceso de negocio está centralizado en una organización pero las tareas se encuentran distribuidas entre el resto de las organizaciones.

- Ejecución en cadena, el proceso se divide en varios subprocesos independientes que las organizaciones ejecutan de forma secuencial, donde una organización activa la ejecución de tareas en otra.

- Subcontratación, una organización subcontrata subprocesos a otras organizaciones.

- Transferencia de casos, las organizaciones tienen el mismo proceso de negocio y la carga de trabajo se distribuye entre las mismas.

- Débilmente acoplado, el proceso de negocio se divide en partes que pueden estar activas simultáneamente, donde cada organización tiene un subproceso y sólo conocen el protocolo utilizado para comunicarse.

Cuando se tiene un proceso colaborativo, existe una interacción entre las organizaciones la cual es enviando y recibiendo mensajes. A este intercambio de mensajes se le denomina coreografía de procesos. Para realizar una interacción correcta, las organizaciones deben acordar una coreografía común antes de interactuar (Weske, 2019)

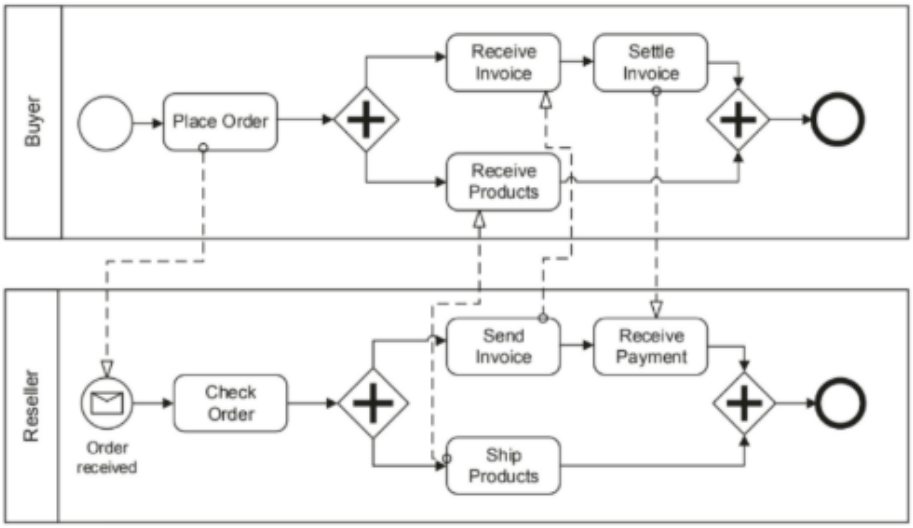


Figura 2.2: Ejemplo de un proceso colaborativo (Weske, 2019)

2.3. Minería de procesos

La minería de procesos (W. M. P. van der Aalst, 2011) es una disciplina que se encuentra entre la Ciencia de Datos y la Ciencia de Procesos. Esta disciplina incluye tres enfoques principales: descubrimiento automático de procesos, chequeo de conformidad de modelos y extensión/mejora de modelos de procesos de negocio reales mediante la extracción de la información que se obtiene de los registros de eventos de los sistemas involucrados.
La minería de procesos permite cerrar el ciclo de vida de un proceso de negocio, los datos que se registran en los sistemas de información pueden ser utilizados para mejorar el proceso de negocio haciendo un análisis de los datos para evaluar posibles desviaciones y en tal caso, mejorar el modelo de negocio (W. M. P. van der Aalst, 2011)

Se pueden identificar tres tipos de minería de procesos (W. M. P. van der Aalst, 2011). Por un lado el descubrimiento de modelos, que a partir de un registro de eventos produce un modelo de proceso de negocio (por ejemplo BPMN) sin usar ninguna otra información adicional. Otro tipo de minería de procesos es conformidad de modelos, donde dado un modelo de proceso y un registro de eventos con ejecuciones de ese proceso, se chequea o comprueba que el modelo se ajusta a la realidad registrada en el registro de eventos y viceversa.
La comprobación puede utilizarse para detectar y localizar desviaciones, y para medir la gravedad de las mismas.
Por último la extensión de modelos, donde la idea es ampliar o mejorar un modelo de proceso existente utilizando información sobre el proceso real registrado en el registro de eventos, teniendo en cuenta por ejemplo la duración, recursos involucrados o cuellos de botella.

2.4. Log de eventos

En la Figura 2.3 ilustra la información típica presente en un log de eventos utilizado para la minería de procesos. Suponemos que un registro de eventos contiene datos relacionados con un único proceso, es decir, el primer paso de delimitación de alto nivel en la Figura 2.4 debe asegurar que todos los eventos puedan vincularse a este proceso. Además, cada evento en el registro debe referirse a una única instancia del proceso, usualmente denominada caso.

Se especifica un proceso como una colección de actividades de tal manera que se describe el ciclo de vida de una instancia única. Por lo tanto, las columnas "id de caso" y "actividad" en la Figura 2.3 representan el mínimo necesario para la minería de procesos.

Los eventos dentro de un caso deben estar ordenados, sin información de orden, sería imposible descubrir dependencias causales en los modelos de procesos. Este ordenamiento usualmente se realiza en base a los timestamps de ejecución registrados para cada evento.

En la Figura 2.3 también se muestra que cada evento puede contener información adicional como por ejemplo *timestamp*, *resources*, *cost.*

Case id	Event id	Properties				
		Timestamp	Activity	Resource	Cost	...
1	35654423	30-12-2010:11.02	register request	Pete	50	...
	35654424	31-12-2010:10.06	examine thoroughly	Sue	400	...
	35654425	05-01-2011:15.12	check ticket	Mike	100	...
	35654426	06-01-2011:11.18	decide	Sara	200	...
	35654427	07-01-2011:14.24	reject request	Pete	200	...
2	35654483	30-12-2010:11.32	register request	Mike	50	...
	35654485	30-12-2010:12.12	check ticket	Mike	100	...
	35654487	30-12-2010:14.16	examine casually	Pete	400	...
	35654488	05-01-2011:11.22	decide	Sara	200	...
	35654489	08-01-2011:12.05	pay compensation	Ellen	200	...
3	35654521	30-12-2010:14.32	register request	Pete	50	...
	35654522	30-12-2010:15.06	examine casually	Mike	400	...
	35654524	30-12-2010:16.34	check ticket	Ellen	100	...
	35654525	06-01-2011:09.18	decide	Sara	200	...
	35654526	06-01-2011:12.18	reinitiate request	Sara	200	...
	35654527	06-01-2011:13.06	examine thoroughly	Sean	400	...
	35654530	08-01-2011:11.43	check ticket	Pete	100	...
	35654531	09-01-2011:09.55	decide	Sara	200	...
	35654533	15-01-2011:10.45	pay compensation	Ellen	200	...
4	35654641	06-01-2011:15.02	register request	Pete	50	...
	35654643	07-01-2011:12.06	check ticket	Mike	100	...
	35654644	08-01-2011:14.43	examine thoroughly	Sean	400	...
	35654645	09-01-2011:12.02	decide	Sara	200	...
	35654647	12-01-2011:15.44	reject request	Ellen	200	...
...	...	...	...	...	...	...

Figura 2.3: Fragmento de un log de eventos: cada línea corresponde a un evento. (W. van der Aalst, 2016)

La Figura 2.4 muestra la estructura en forma de árbol de un log de eventos reflejando las siguientes características:

- Un proceso consiste de casos.
- Un caso esta compuesto por eventos, de modo que cada evento esta relacionado con un único caso.
- Los eventos dentro de un caso están ordenados.
- Los eventos pueden tener atributos. Ejemplos de nombres típicos de atributos son actividad, tiempo, costos y recurso.

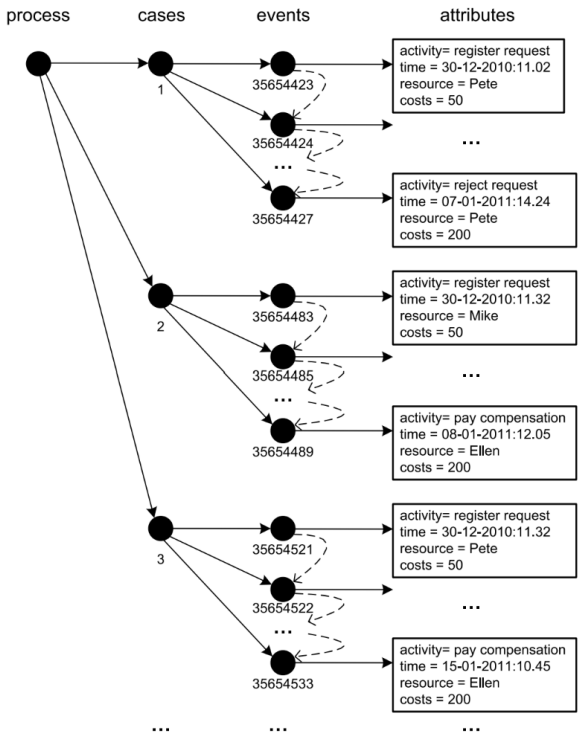


Figura 2.4: Estructura de los logs de eventos. (W. van der Aalst, 2016)

No todos los eventos necesitan tener el mismo conjunto de atributos. Sin embargo, típicamente, los eventos que se refieren a la misma actividad tienen el mismo conjunto de atributos. Para mejorar el entendimiento y reforzar los conceptos, se formalizan diversas nociones.

Se presentan definiciones formales para los conceptos presentados previamente (Bukhsh, Saeed, y Dijkman, 2021)

Definición 1: (Evento)

Sea A el conjunto de actividades, C el conjunto de casos, T el dominio de tiempo, y D_1,...,D_m el conjunto de atributos relacionados donde $m > 0$. Un evento es una tupla $e = (a, c, t, d_1, \ldots, d_m)$, donde $a \in A$, $c \in C$, $t \in T$ y $d_i \in \{D_i\}$ con $i \in [1, m]$.

Definición 2: (Traza, Log de Eventos)

Sean π_A, π_C y π_T funciones que mapean un evento $e = (a, c, t, d_1, \ldots, d_m)$ a una actividad, como $\pi_A(e) = a$, a un identificador de caso único, como

$\pi_C(e) = c$ y a un timestamp, como $\pi_T(e) = t$. Una traza se define como una secuencia finita no vacía de eventos $\sigma = \langle e_1, e_2, \ldots, e_n \rangle$, tal que para todo $e_i, e_j \in \sigma$ debe cumplirse que: los eventos dentro de una traza σ deben tener el mismo identificador de caso, es decir, $\pi_C(e_i) = \pi_C(e_j)$, y el tiempo debe ser no decreciente, es decir, $\pi_T(e_j) \geq \pi_T(e_i)$ para $j > i$. Decimos que una traza $\sigma = \langle e_1, e_2, \ldots, e_n \rangle$ tiene longitud n, denotada $|\sigma|$. Un log de eventos es una colección de trazas $L = \{\sigma_1, \sigma_2, \ldots, \sigma_l\}$. Decimos que una colección $L = \{\sigma_1, \sigma_2, \ldots, \sigma_l\}$ tiene tamaño l, denotado $|L|$.

2.4.1. Log de eventos extendido

En el contexto del proyecto, para poder trabajar con procesos colaborativos, es necesario agregar la extensión (González y Delgado, 2021) que se muestra en la Tabla 2.1. Esto, permite agregar las etiquetas correspondientes a los participantes (Participant) que están interactuando en el evento y al tipo de tarea que estan realizando (elemType).

Tabla 2.1: Extensión para procesos colaborativos

Extension	Key	Level	Type	Description
collab	`collab:elemType`	event	string	Especifica el tipo de evento.
collab	`collab:participant`	event	string	Especifica el participante que ejecuta la tarea.

A continuación vemos un ejemplo de un log que no contiene las etiquetas colaborativas y luego su correspondiente extensión.

```
<log xes.version="1.0" xes.features="nested-attributes" openxes.version="1.0RC7">
    <extension name="Organizational" prefix="org" uri="http://www.xes-standard.org/org.xesext"/>
    <extension name="Time" prefix="time" uri="http://www.xes-standard.org/time.xesext"/>
    <extension name="Concept" prefix="concept" uri="http://www.xes-standard.org/concept.xesext"/>
    <extension name="Identity" prefix="identity" uri="http://www.xes-standard.org/identity.xesext"/>
    <string key="concept:name" value="collog_healthcare"/>
    <trace>
        <string key="concept:name" value="case_44"/>
        <event>
            <string key="msgInstanceId" value="disease_47"/>
            <string key="msgType" value="send"/>
            <string key="org:group" value="Patient"/>
            <string key="concept:name" value="Communicate disease"/>
            <date key="time:timestamp" value="2021-06-17T11:06:35.562+02:00"/>
            <string key="msgProtocol" value="P2P"/>
            <string key="msgName" value="disease"/>
        </event>
        <event>
            <string key="msgInstanceId" value="disease_47"/>
            <string key="msgType" value="receive"/>
            <string key="org:group" value="Gynecologist"/>
            <string key="concept:name" value="Receive disease info"/>
            <string key="eventType" value="start"/>
            <date key="time:timestamp" value="2021-06-17T11:06:38.889+02:00"/>
            <string key="msgProtocol" value="P2P"/>
            <string key="msgName" value="disease"/>
        </event>
```

Figura 2.5: Ejemplo log original - Asistencia en salud (caso de estudio).

Figura 2.6: Ejemplo log extendido para procesos colaborativos- Asistencia en salud (caso de estudio).

2.5. Monitoreo predictivo de procesos de negocio

El Monitoreo Predictivo de Procesos de Negocio (PBPM, por sus siglas en inglés) es una disciplina que busca anticipar el comportamiento futuro de los procesos de negocio mediante el análisis de datos históricos y en tiempo real. A través de técnicas avanzadas de minería de procesos, aprendizaje automático y modelos predictivos, permite detectar patrones y prever posibles desviaciones que podrían afectar la eficiencia o el cumplimiento de los objetivos organizacionales. (Márquez-Chamorro, Resinas, y Ruiz-Cortés, 2018)

El objetivo del PBPM es proporcionar a las organizaciones la capacidad de actuar proactivamente, optimizando la toma de decisiones y evitando problemas antes de que estos se materialicen.
Para implementar PBPM es fundamental capturar datos relevantes del proceso en forma de eventos, a menudo a partir de registros de sistemas transaccionales o plataformas BPM. Estos datos se procesan mediante técnicas de minería de procesos para descubrir y analizar el comportamiento del flujo de trabajo. Posteriormente, se aplican modelos predictivos como árboles de decisión, redes neuronales o modelos probabilísticos que permiten generar predicciones útiles para los responsables del negocio. (Ceravolo, Comuzzi, De Weerdt, Di Francescomarino, y Maggi, 2024)

La integración de PBPM aporta una ventaja competitiva al permitir que

las organizaciones no solo reaccionen a problemas, sino que se anticipen a ellos, promoviendo una gestión ágil y eficiente de los procesos de negocio.

La Figura 2.7 presenta las principales fases de los enfoques típicos basados en aprendizaje automático. Estos enfoques suelen requerir que se extraigan prefijos de trazas a partir de las trazas de ejecución históricas (fase de extracción de prefijos). Esto se debe a que, en tiempo de ejecución, las predicciones se realizan sobre trazas incompletas, por lo que es necesario aprender, en la fase de entrenamiento, las correlaciones entre trazas incompletas y lo que se desea predecir (variables objetivo o etiquetas). Después de que se han extraído los prefijos, las trazas de prefijo y las etiquetas (es decir, la información que se desea predecir) se codifican en forma de vectores de características (fase de codificación). Las trazas codificadas se pasan luego a las técnicas de aprendizaje supervisado encargadas de aprender a partir de los datos codificados uno (o más) modelo(s) predictivo(s) (fase de codificación). En tiempo de ejecución, las trazas de ejecución incompletas, es decir, aquellas cuyo futuro es desconocido, también deben codificarse como vectores de características y utilizarse para consultar el(los) modelo(s) predictivo(s) con el fin de obtener la predicción (fase de predicción). (Di Francescomarino y Ghidini, 2022)

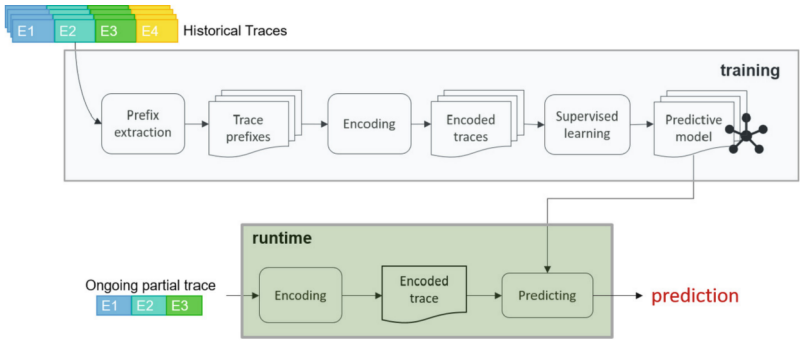


Figura 2.7: Resumen de las fases típicas de los enfoques basados en el aprendizaje automático (Di Francescomarino y Ghidini, 2022).

Capítulo 3

Estado del arte

3.1. Fuentes académicas

Para comenzar con el estudio del tema, tuvimos como objetivo poder revisar y analizar la literatura existente sobre la predicción de procesos colaborativos. Para ello, realizamos búsquedas en plataformas y bases de datos académicas como IEEE, Scopus, Springer y Science direct utilizando en las cadenas de búsquedas ciertas palabras claves.

En IEEE las búsquedas avanzadas no concatenan las palabras, sino que busca todas sin importar el orden. Primero buscamos solo en títulos de documentos y publicaciones con la siguiente búsqueda avanzada sin encontrar resultados:

("Document Title": ("collaborative" OR "collaboration" OR "choreograph") AND "business process" AND ("prediction" OR "predictive")) OR ("Publication Title": ("collaborative" OR "collaboration" OR "choreography") AND "business process*" AND ("prediction" OR "predictive"))*

Extendiendo la búsqueda a todos los campos de información vemos que aparecen 28 conferencias, 1 revista y 1 guía de términos de IEEE. Pero luego de revisar los títulos y resumen de la lista, vemos que ninguno tiene relación con la colaboración entre organizaciones para la predicción de procesos de negocio

En Scopus, buscando solo en títulos, resúmenes y palabras claves de la siguiente forma:

TITLE-ABS-KEY(("collaborative" OR "collaboration" OR "choreography") AND "business process" AND ("prediction" OR "predictive"))*

Obtenemos 129 resultados. Luego tratando de acotar esa cantidad filtrando por temas como Computer Science, Decision Science y Engineering. Tampoco logramos encontrar alguno que nos pueda ayudar.

En Springer, buscamos que en los títulos se encuentren los conceptos de

nuestra búsqueda sin resultados para la consulta completa, por eso decidimos quitar algunas palabras para ver qué artículos nos traería y si tenían alguna relación al tema.

Dejando *("collaborative" OR "collaboration" OR "choreography")* junto a *"business processes"* primero y luego junto a *("prediction" OR "predictive")* no obtenemos ningún resultado.

Luego para *"business processes" AND ("prediction" OR "predictive")* encontramos 97 conference paper y 22 artículos

Más allá de encontrar algún artículo relacionado a la predicción en los procesos de negocio, como era de esperarse por la reformulación de la búsqueda, no encontramos nada relacionado a procesos colaborativos. También se agregó la palabra inter-organizational pero los pocos resultados encontrados no son relacionados al tema.

En Science direct, realizamos la búsqueda en los Título, resumen o palabras clave especificadas por el autor ya que al hacerlo en todo el artículo devuelve 6193 resultados.

("collaborative" OR "collaboration" OR "choreography") AND "business processes" AND ("prediction" OR "predictive")

Esta búsqueda trae 11 resultados pero ninguno sobre el tema en cuestión. También como en IEEE, en esta librería también se buscan solo por términos.

3.2. Tipos de predicción existentes

Al ver que no encontrábamos ningún artículo o publicación que hable directamente de la predicción de procesos colaborativos decidimos enfocarnos en la búsqueda de todos los tipos de predicción que podíamos hacer de un proceso. Para ello, nos enfocamos en algunos papers para comenzar a tener una visión general del tema.

En (Di Francescomarino y Ghidini, 2022) se da una introducción al monitoreo predictivo de procesos con un ejemplo simple y las principales dimensiones que caracterizan a la familia de enfoques de Monitoreo Predictivo de Procesos para procesos de negocio. Se describen las codificaciones y enfoques típicos utilizados para la predicción de resultados, valores numéricos y secuencias de actividades.

El principal objetivo de (Di Francescomarino, Ghidini, Maggi, y Milani, 2018) es desarrollar un marco basado en valores para clasificar los trabajos existentes sobre el monitoreo predictivo de procesos. Este objetivo se logra identificando, categorizando y analizando sistemáticamente los enfoques existentes para la monitoreo predictivo de procesos. En la sección 4 del mismo, se ofrece una clasificación de los tipos de predicción en 3 dimensiones: numéricas, de resultado y próxima actividad/secuencia de actividades. Finalmente, el marco tiene en cuenta el tipo de predicción, los datos de entrada necesarios, la existencia de herramientas que le den soporte, técnica en que se basa el algoritmo de predicción y además proporciona una referencia al trabajo específico en la literatura. Esto nos dio una idea de la cantidad de variables que pueden haber

relacionado a la predicción de proceso y algunos de los distintos métodos que pueden utilizarse.

Para seguir buscando otros posibles enfoques, fue que también recurrimos a otro artículo (Márquez-Chamorro y cols., 2018) donde se resumen los enfoques más representativos para la predicción en tiempo de ejecución de procesos de negocio. Se categorizan métodos teniendo en cuenta diferentes tipos de técnicas de predicción computacional, como estadísticas o los enfoques de aprendizaje automático, y determinados aspectos como el tipo de valores predichos y las métricas de evaluación de la calidad.
Finalmente luego de analizar los papers mencionados, viendo las distintas dimensiones, métodos, dominios, tipos de entrada, aspectos de evaluación y los tipos de predicción que pueden hacerse, tomamos de (Di Francescomarino y Ghidini, 2022) la clasificación de los tipos de predicción como:

- **Predicciones basadas en resultados**: Predicciones relacionadas con valores de resultado categóricos o booleanos predefinidos.
- **Predicciones de valores numéricos**: Predicciones relacionadas con medidas de interés que toman valores numéricos o continuos. Ejemplos en esta categoría podrían ser:
  - *Predicción tiempo hasta próximo evento*: indica el tiempo que falta hasta que ocurra el próximo evento.
  - *Predicción de Tiempo Restante*: indica el tiempo que falta para que finalice el proceso.
- **Predicciones del próximo evento**: Predicciones relacionadas con secuencias de actividades futuras y sus atributos, por ejemplo:
  - *Predicción de próxima actividad*, donde la predicción indica cual será la próxima actividad a realizarse.

La figura 3.1 muestra un ejemplo de traza de ejecución que describe las actividades realizadas por Juan. Supongamos que son las 8:54 de la mañana. A las 8:00 a.m. Juan se ha registrado en el hospital para someterse a unos controles de salud, a las 8:10 lo han llevado al departamento de radiología, donde lo han visitado a las 8:15 y ahora le están haciendo radiografías. La monitorización predictiva de procesos nos permitiría responder a distintos tipos de preguntas sobre el futuro de John.

Por ejemplo, podríamos predecir si Juan se someterá a una ecografía en el futuro. La respuesta a esta pregunta concreta será un valor booleano (por ejemplo, es cierto que Juan se someterá a una ecografía en el futuro). Este es un ejemplo típico de **predicción basada en resultados**.

Otra pregunta típica que la Monitorización Predictiva de Procesos podría permitirnos responder sobre el futuro de Juan es, una vez que sabemos que se va a someter a una ecografía, en cuánto tiempo se la va a hacer. La respuesta a esta pregunta se proporciona generalmente en términos de un valor numérico

(por ejemplo, Juan se va a hacer una ecografía en 26 min) y es un ejemplo de **predicción de valor numérico**.

Por último, podríamos incluso predecir lo que Juan va a hacer a partir de ahora. La respuesta a esta pregunta es una secuencia de actividades futuras (por ejemplo, Juan se someterá a una ecografía, pedirá su factura y la pagará) y es un ejemplo de **predicción del próximo evento**.

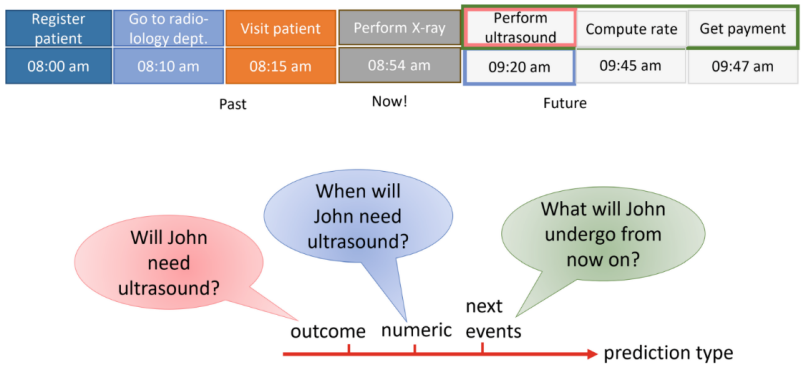


Figura 3.1: Tipos de predicción (Di Francescomarino y Ghidini, 2022).

3.3. Herramientas e implementaciones existentes

3.3.1. ProM

En la búsqueda de aplicaciones o herramientas existentes que permitan el monitoreo predictivo de procesos nos encontramos con ProM, una de las herramientas más utilizadas y conocidas en Minería de Procesos. Se trata de un marco de trabajo que reúne una serie de plugins, que funcionan de forma independiente unos de otros, y cada uno centrado en la ejecución de una tarea específica. Entre su variedad de plugins, ProM también recoge varios plugins que implementan técnicas para la predicción de resultados, para la predicción de valores numéricos, así como para la predicción de secuencias de actividades siguientes.

3.3.2. Apromore

Apromore es otra herramienta muy conocida y consolidada. Se trata de un repositorio avanzado de modelos de procesos que permite almacenar, analizar y reutilizar grandes conjuntos de modelos de procesos. La herramienta está basada en la web y, por lo tanto, permite la fácil integración de nuevos plug-ins de una manera orientada al servicio.

Si bien teníamos experiencia trabajando con ProM (descubrimiento de procesos, chequeo de conformidad, extensión de modelos) creemos que Apromore tiene la ventaja de ser una aplicación web, con todo lo que implica ello en relación a la accesibilidad, escalabilidad y compatibilidad con tecnologías emergentes. Por ello comenzamos a probar e investigar una versión de prueba de Apromore. El complemento de monitoreo de procesos predictivos de la Apromore incluye un módulo de "entrenamiento" y un módulo de "monitoreo del tiempo de ejecución". El módulo de entrenamiento utiliza algoritmos de aprendizaje automático para crear modelos predictivos a partir de un registro de eventos. Una vez que ha entrenado un modelo, Apromore le proporciona informes sobre el poder predictivo del modelo y explicaciones sobre cómo el modelo realiza sus predicciones. Luego el modelo entrenado puede utilizarse para generar cuadros de mando predictivos que ofrecen predicciones para cada caso abierto en el proceso. En esa versión de prueba solo teníamos disponibles las predicciones de tiempo restante y resultados del caso (pudiendo seleccionar los argumentos que nos interesaban). Además algunas funcionalidades e informes que menciona que tiene la herramienta no estaban disponibles.

3.3.3. ProcessTransformer

Continuando con la búsqueda de otras herramientas nos encontramos con el artículo (Bukhsh y cols., 2021). En este trabajo, los autores proponen ProcessTransformer, un enfoque para el aprendizaje de representaciones de alto nivel a partir de registros de eventos con una red basada en la atención. Aborda el problema de PBPM utilizando deep learning para aprender las funciones de predicción Θ_a, Θ_t y Θ_{rt} tal y como se definen en las definiciones 3-5. En particular, mientras que existen varios métodos de monitorización de procesos basados en deep learning, que aprenden un modelo predictivo basado en la variación de la longitud de los prefijos de las secuencias de eventos, ProcessTransformer es una red neuronal profunda que considera todos los prefijos posibles para el entrenamiento y la inferencia.

Para la implementación del processTransformer se ha utilizado TensorFlow (Tensorflow, s.f.), que es una plataforma end-to-end de código abierto para el aprendizaje automático (machine learning). Ofrece un ecosistema integral y flexible de herramientas, bibliotecas y recursos comunitarios que permiten a los investigadores avanzar en el estado del arte en aprendizaje automático, mientras que facilita a los desarrolladores la creación y despliegue de aplicaciones impulsadas por esta tecnología. Originalmente, TensorFlow fue desarrollado por investigadores e ingenieros del equipo de Inteligencia Artificial de Google Brain, con el objetivo de realizar investigaciones en aprendizaje automático y redes neuronales. Sin embargo, su versatilidad permite su aplicación en diversas áreas. La plataforma proporciona APIs estables en Python y C++.

A partir de la versión 2.0 (ProcessTransformer utiliza 2.4 o mayor), TensorFlow integra Keras (Keras, s.f.) como su interfaz principal, permitiendo a los desarrolladores definir, entrenar y evaluar modelos de forma sencilla. Esta integración no solo simplifica la construcción de modelos complejos mediante

una API clara y concisa, sino que también permite escribir código compacto y modular, facilitando tanto el desarrollo como la experimentación.

A continuación creemos importante especificar como define las predicciones que implementa el ProcessTransformer:

Definición 3: (Predicción de Actividad)

Sea σ una traza $\langle e_1, e_2, \ldots, e_n \rangle$ y $k \in [1, n-1]$ un número positivo escalar. El prefijo de eventos de longitud k, $hd^k(\sigma)$ puede definirse como: $hd^k(\sigma) = \langle e_1, e_2, \ldots, e_n \rangle$ El prefijo de actividades puede obtenerse aplicando la función de mapeo π_A como $\pi_A(hd^k(\sigma)) = (\pi_A(e_1), \pi_A(e_2), \ldots, \pi_A(e_k))$. La predicción de actividad es la definición de una función Θ_a que toma el prefijo de eventos $hd^k(\sigma)$ donde $k \in [1, n-1]$ y predice la siguiente actividad e' , es decir:

$$\Theta_a(hd^k(\sigma)) = \pi_A(e'_{k+1})$$

Definición 4: (Predicción tiempo hasta próximo evento)

Sea $\sigma = \langle e_1, e_2, \ldots, e_n \rangle$ una traza de largo n. Para extraer las características relacionadas con el tiempo del último evento en la traza, se definen las siguientes funciones:

$$fv_{t1}(\sigma) = \begin{cases} 0 & \text{si } |\sigma| = 1, \\ \pi_T(e_n) - \pi_T(e_{n-1}), & \text{en otro caso.} \end{cases}$$

$$fv_{t2}(\sigma) = \begin{cases} 0 & \text{si } |\sigma| \in [1, 2], \\ \pi_T(e_n) - \pi_T(e_{n-2}), & \text{en otro caso.} \end{cases}$$

$$fv_{t3}(\sigma) = \begin{cases} 0 & \text{si } |\sigma| = 1, \\ \pi_T(e_n) - \pi_T(e_0), & \text{en otro caso.} \end{cases}$$

La función fv_{t1} representa la diferencia de tiempo entre el evento anterior y el evento actual de una traza. La función fv_{t2} contiene la diferencia de tiempo entre el evento actual y el evento de dos eventos antes. Finalmente, fv_{t3} muestra el tiempo aproximado transcurrido desde que el caso ha comenzado. La predicción de tiempo del próximo evento , es la definición de una función Θ_t que toma el prefijo de eventos $hd^k(\sigma)$ donde $k \in [1, n-1]$ y predice el momento en que probablemente ocurrirá el siguiente evento , es decir:

$$\Theta_t(\sigma', fv_{t_1}(\sigma'), fv_{t_2}(\sigma'), fv_{t_3}(\sigma')) = \pi_T(e'_{k+1}), \quad \text{donde } \sigma' = hd^k(\sigma).$$

Definición 5: (Predicción de Tiempo Restante)

Sea $\sigma = \langle e_1, e_2, \ldots, e_n \rangle$ una traza de largo n.

La predicción de tiempo restante es la definición de una función Θ_{rt} que toma el prefijo de eventos$hd^k(\sigma)$ donde$k \in [1, n-1]$ y predice el tiempo restante del caso , es decir:

$$\Theta_{rt}(\sigma', fv_{t_1}(\sigma'), fv_{t_2}(\sigma'), fv_{t_3}(\sigma')) = \pi_T(e_n) - \pi_T(e_k), \quad \text{donde } \sigma' = hd^k(\sigma).$$

Métricas para predicciones

Para las predicciones de próxima actividad, se utilizan las siguientes métricas para evaluar el modelo:

Accuracy *(exactitud)*:

La exactitud es la proporción de todas las clasificaciones correctas, ya sean positivas o negativas. Se define matemáticamente de la siguiente manera:

$$Accuracy = \frac{\text{correct classifications}}{\text{total classifications}} = \frac{TP + TN}{TP + TN + FP + FN}$$

Indica la precisión global del modelo, es decir, la proporción de predicciones correctas sobre el total de muestras evaluadas. Puede ser engañosa en conjuntos de datos desbalanceados ya que no tiene en cuenta la distribución de las clases. En escenarios desbalanceados, donde una clase es mucho más frecuente que las otras, un modelo que siempre prediga la clase mayoritaria podría tener una alta accuracy sin ser realmente útil.

Un modelo perfecto no tendría ningún falso positivo ni ningún falso negativo y, por lo tanto, tendría una precisión de 1.0 o 100 %.

Precision: La precisión es la proporción de todas las clasificaciones positivas del modelo que realmente son positivas. Matemáticamente, se define de la siguiente manera:

$$Precision = \frac{\text{correctly classified actual positives}}{\text{everyhing classified as positive}} = \frac{TP}{TP + FP}$$

Mide la exactitud del modelo para las predicciones positivas. Indica cuántas de las predicciones positivas realizadas por el modelo fueron correctas.

Recall *(recuperación)*: La recuperación o tasa de verdaderos positivos (TPR) es la proporción de todos los positivos reales que se clasificaron correctamente como positivos. La recuperación se define matemáticamente de la siguiente manera:

$$Recall = \frac{\text{correctly classified actual positives}}{\text{all actual positive}} = \frac{TP}{TP + FN}$$

Evalúa la capacidad del modelo para identificar todas las instancias positivas.Indica qué tan completo es el modelo al encontrar todas las instancias positivas.

En un conjunto de datos desequilibrado en el que la cantidad de casos positivos reales es muy baja, por ejemplo, de 1 a 2 ejemplos en total, la recuperación es menos significativa y útil como métrica.

La recuperación mejora cuando disminuyen los falsos negativos, mientras que La precisión mejora a medida que disminuyen los falsos positivos. Sin embargo, aumentar el umbral de clasificación tiende a aumentar la cantidad de falsos negativos y disminuir la cantidad de falsos positivos, mientras que disminuir el umbral tiene los efectos opuestos. Como resultado, la recuperación y la precisión suelen mostrar una relación inversa, en la que mejorar uno de ellos empeora al otro.

F-Score: Es la media armónica entre la precisión y el recall. Esta métrica es útil para balancear la precisión y el recall, especialmente cuando ambas son importantes para la tarea. Se define matemáticamente de la siguiente manera:

$$F\text{-}score = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

En las predicciones de tiempo hasta el próximo evento y tiempo restante se utilizan:

MAE (Mean Absolute Error) – Error Absoluto Medio Como la salida es un valor continuo, se utiliza el MAE para medir el rendimiento. Esta métrica se calcula como el promedio de los errores absolutos entre los valores reales y los predichos, proporcionando una idea clara del margen de error del modelo.

$$\text{MAE} = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n}$$

donde y_i es el valor predicho por el modelo, $\hat{y}_i$ es el valor deseado, y n es el número total de muestras en el conjunto de prueba.

MSE (Mean Squared Error) – Error Cuadrático Medio El MSE calcula el promedio de los errores al cuadrado. A diferencia del MAE, el MSE amplifica los errores grandes debido al cuadrado. Esto puede ser útil si quieres que tu modelo evite cometer grandes errores.

RMSE (Root Mean Squared Error) – Raíz del Error El RMSE es simplemente la raíz cuadrada del MSE. Al igual que el MSE, penaliza más los errores grandes, pero resulta más fácil de interpretar al estar en las mismas unidades que los datos.

Capítulo 4

Problemática y propuesta de solución

4.1. Definición del problema

El monitoreo predictivo de procesos de negocio se ha enfocado principalmente en procesos que se realizan en una única organizacion (intra-organizacionales) y no en procesos colaborativos donde participan dos o más organizaciones (inter-organizacionales). Las implementaciones existentes no incluyen ningún elemento colaborativo, ni ningún método explicito que indique como poder extenderla para considerarlos. Por esta razón, se quiere contar con una propuesta y herramienta que permita realizar predicciones para procesos de negocio colaborativos.

En la Figura 4.1 se presenta gráficamente la problemática a resolver, que consiste en implementar una herramienta que recibe como dato de entrada un log extendido de colaboración y genera predicciones colaborativas.

El log extendido de colaboración del cual se parte incluye los datos los participantes que están interactuando en el evento y el tipo de tarea que estan realizando (González y Delgado, 2021).

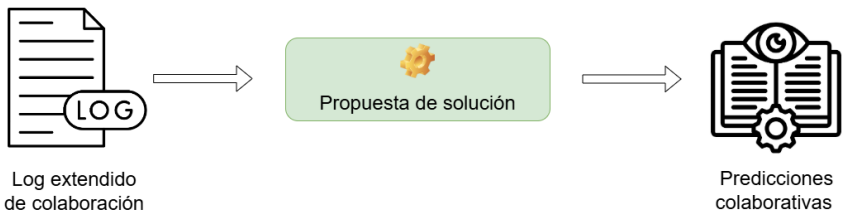


Figura 4.1: Representación de la problemática

4.2. Propuesta de solución

Se plantea desarrollar una aplicación que permita realizar predicciones para procesos colaborativos, partiendo de un log extendido de colaboración. En la figura 4.2 se ven a grandes rasgos las etapas que deberá tener la solución a implementar.

- **Preprocesamiento del log:** Los datos del log extendido serán extraídos y codificados para preparar la información necesaria para la generación del modelo.
- **Generación del modelo de predicción:** A partir de los datos preprocesados, se construirán los modelos correspondientes a los tipos de predicción definidos.
- **Ejecución de predicciones:** Una vez generado el modelo de predicción, será posible cargar trazas en curso para realizar predicciones sobre ellas.

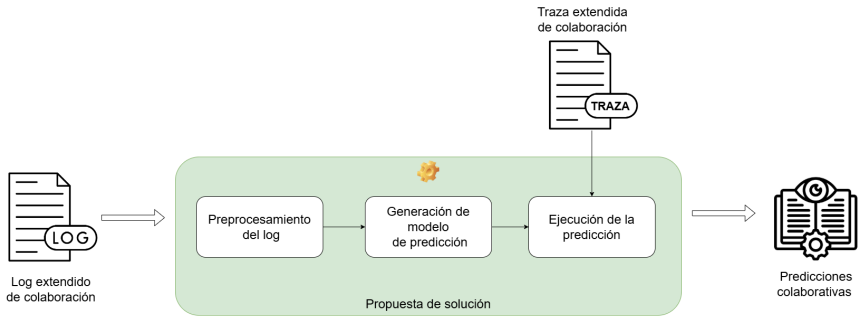


Figura 4.2: Propuesta de solución

4.2.1. Propuesta de extensión para procesos colaborativos

Partiendo de las predicciones para procesos no colaborativos categorizadas (3.2) e integrando los conceptos de participante y mensaje se define una lista preliminar de posibles tipos de predicciones que podrían ser interesantes implementar para procesos colaborativos.

Numéricos:

- Tiempo de finalización de un participante
- Tiempo restante de un participante
- Tiempo de retraso en participantes

    - Duración participante
    - Tiempo hasta el próximo mensaje a recibir/enviar
    - Probabilidad de que un participante partícipe del caso
    - Probabilidad de que se envíe/reciba un mensaje en particular
    - Cantidad de mensajes que enviará/ recibirá un participante.
    - Cantidad de mensaje restantes

- Futuras actividades
    - Qué participante realizará la próxima actividad
    - Qué participante enviará/ recibirá el próximo mensaje
    - próximo mensaje/ secuencia de próximos mensajes

Nuestra propuesta se basa en extender los tipos de predicciones implementados por el ProcessTranformer (Bukhsh y cols., 2021) a procesos colaborativos.

En la búsqueda de entender la herramienta, encontramos un código compartido en Google Colab (GitHub-Processtransformer, s.f.) que nos permitió rápidamente poder ejecutar las predicciones de siguiente actividad, el tiempo en que ocurrirá el próximo evento y el tiempo restante para un caso en ejecución.

Partiendo de este código compartido (y junto con el acceso al código de todo ProcessTransformer) comenzamos una lectura descendente (top-down reading) del mismo, para poder tener primero una visión general de la aplicación y comprender las secciones, siguiendo el camino natural de la ejecución, e ir bajando en los niveles de código para profundizar cuando consideramos necesario. Luego de esta lectura del código, y de muchas ejecuciones controladas y minuciosas, pudimos comenzar a pensar cómo extender esta herramienta para las distintas predicciones colaborativas, predicciones que veremos en la siguiente sección.

En la figura 4.3 es una modificación de la presentada en el marco teorico 2.7 donde se resumen las fases típicas de los enfoques basados en apredizaje automático, como el del *ProcessTranformer*.

En rojo se indican las etapas donde se agregan modificaciones para lograr la solución propuesta. En primer lugar, las trazas que se toman como entrada tanto para el entrenamiento como para la predicción. tienen el formato del log extendido. Además la modificación principal sobre la implementación del *ProcessTransformer* se encuentra en la etapa de extracción de los prefijos donde se modifican y se agregan funciones que permiten obtener los prefijos y etiquetas (lo que se quiere predecir) de los nuevos tipos de predicción.

Aunque el *ProcessTransformer* generaba predicciones, estas se realizaban sobre un porcentaje del log proporcionado que se destinaba específicamente para pruebas. Posteriormente, los datos obtenidos de dichas predicciones se empleaban para calcular las métricas del modelo, sin que se mostraran los resultados de las predicciones propiamente dichas.

La solución propuesta separa la generación del modelo de predicción de la realización de predicciones, permitiendo que estas últimas se lleven a cabo en cualquier momento mediante la carga de trazas en curso en el modelo previamente generado. Asimismo, permite la visualización no solo de las métricas asociadas al modelo, sino también de los resultados obtenidos para cada predicción realizada.

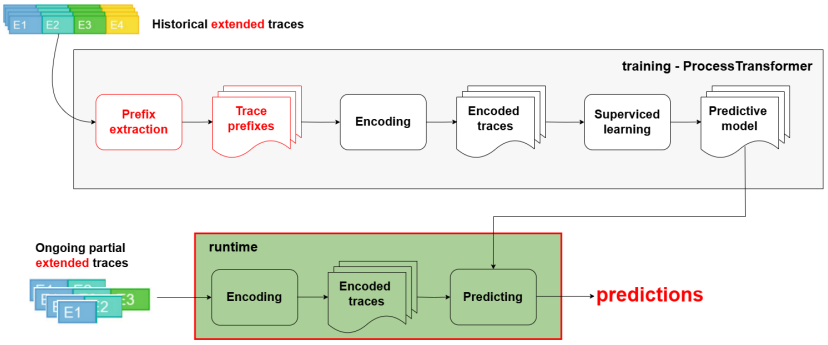


Figura 4.3: Representación de la problemática

Tipos de predicciones extendidas

En base a la lista 4.2.1 y analizando las posibilidades que nos brinda el *ProcessTransformer*, se agregan a las predicciones existentes las siguientes, integrando los elementos de procesos colaborativos:

- Basadas en Predicción de Actividad
  - ***Próximo participante que enviará un mensaje:*** La predicción indica únicamente el participante que es probable que envíe el próximo mensaje.
  - ***Próximo participante que enviará un mensaje (con actividad):*** La predicción indica la actividad y el participante que corresponden al posible próximo envío de mensaje.
  - ***Próxima actividad que ocurrirá y qué participante la realizará:***. La predicción indica la posible próxima actividad y el participante.
  - ***Próximo participante que realizará una actividad:*** La predicción indica únicamente el participante que es probable que realice la próxima actividad.
- Basadas en Predicción tiempo hasta próximo evento

- ***Tiempo hasta el próximo mensaje a enviar:*** La predicción indica el tiempo que falta hasta que ocurra el próximo envío de mensaje.

- Basadas en Predicción tiempo hasta próximo evento

  - ***Tiempo restante del participante:*** La predicción indica el tiempo que falta hasta que finalicen las actividades del participante en el caso.

Extensión de definiciones

A partir de las nuevas predicciones se extienden las definiciones de las funciones de predicción usadas en el *ProcessTransformer* (Bukhsh y cols., 2021)

Definición 1 extendida: (Evento) Sea A el conjunto de actividades, C el conjunto de casos, T el dominio de tiempo, P el conjunto de participantes, ET el conjunto {SendTask, ReceiveTask, task} y D_1,...,D_m el conjunto de atributos relacionados donde $m > 0$. Un evento es una tupla $e = (a, c, t, p, et, d_1, \ldots, d_m)$, donde $a \in A$, $c \in C$, $t \in T$, $p \in P$, $et \in ET$ y $d_i \in \{D_i\}$ con $i \in [1, m]$.

Definición 2 extendida: (Traza, Log de Eventos) Sean π_A, π_C, π_T, π_P y π_{ET} funciones que mapean un evento
$e = (a, c, t, p, et, d_1, \ldots, d_m)$ a una actividad, como $\pi_A(e) = a$, a un identificador de caso único, como $\pi_C(e) = c$, a un timestamp, como $\pi_T(e) = t$, a un participante, como $\pi_P(e) = p$ y a un elemtype como $\pi_{ET}(e) = et$. Una traza se define como una secuencia finita no vacía de eventos $\sigma = \langle e_1, e_2, \ldots, e_n \rangle$, tal que para todo $e_i, e_j \in \sigma$ debe cumplirse que: los eventos dentro de una traza σ deben tener el mismo identificador de caso, es decir, $\pi_C(e_i) = \pi_C(e_j)$, y el tiempo debe ser no decreciente, es decir, $\pi_T(e_j) \geq \pi_T(e_i)$ para $j > i$. Decimos que una traza $\sigma = \langle e_1, e_2, \ldots, e_n \rangle$ tiene longitud n, denotada $|\sigma|$. Un log de eventos es una colección de trazas $L = \{\sigma_1, \sigma_2, \ldots, \sigma_l\}$. Decimos que una colección $L = \{\sigma_1, \sigma_2, \ldots, \sigma_l\}$ tiene tamaño l, denotado $|L|$.

Además se definen π_{A_P}, π_{ET_P} y $\pi_{ET\_A\_P}$ que mapean un evento a e a la concatenación de actividad y participante, como $\pi_{A\_P}(e) = a\_p$, a la concatenación de elemtype y participante, como $\pi_{ET\_P}(e) = et\_p$, a la concatenación de elemtype, actividad y participante, como $\pi_{ET\_A\_P}(e) = et\_a\_p$

Definición 6: (Predicción de Próximo participante que realizará una actividad) De manera análoga a como se definieron las funciones para próxima actividad, definimos funciones para próximo participante. El prefijo de participantes puede obtenerse aplicando la función de mapeo π_P como $\pi_P(hd^k(\sigma)) = (\pi_P(e_1), \pi_P(e_2), \ldots, \pi_P(e_k))$. La predicción de próximo participante es la definición de una función Θ_p que toma el prefijo de eventos $hd^k(\sigma)$ donde $k \in [1, n-1]$ y predice el próximo participante e', es decir:

$$\Theta_p(hd^k(\sigma)) = \pi_P(e'_{k+1})$$

Definición 7: (Predicción de Próxima actividad que ocurrirá y qué participante la realizará) El prefijo de *actividad_participante* puede obtenerse aplicando la función de mapeo π_{A_P} como $\pi_{A\_P}(hd^k(\sigma)) = (\pi_{A\_P}(e_1), \pi_{A\_P}(e_2), \ldots, \pi_{A\_P}(e_k))$.

La predicción de próxima *actividad_participante* es la definición de una función Θ_{ap} que toma el prefijo de eventos $hd^k(\sigma)$ donde $k \in [1, n-1]$ y predice la próxima *actividad_participante* e' , es decir:

$$\Theta_{ap}(hd^k(\sigma)) = \pi_{A\_P}(e'_{k+1})$$

Definición 8: (Predicción de Próximo participante que enviará un mensaje) El prefijo de *elemtype_participante* puede obtenerse aplicando la función de mapeo π_{ET_P} como $\pi_{ET\_P}(hd^k(\sigma)) = (\pi_{ET\_P}(e_1), \pi_{ET\_P}(e_2), \ldots, \pi_{ET\_P}(e_k))$. La predicción de próximo participante que enviará un mensaje es la definición de una función Θ_{pm} que toma el prefijo de eventos $hd^k(\sigma)$ donde $k \in [1, n-1]$ y predice el próximo participante e' , es decir:

$$\Theta_{pm}(hd^k(\sigma)) = \pi_{ET\_P}(e'_{k+j}) \quad || \quad dummy$$

Siendo j la cantidad de eventos hasta encontrar el próximo envío de mensaje. *dummy* será el resultado cuando no haya un próximo envío de mensaje luego del evento actual.

Definición 9: (Predicción de Próximo participante que enviará un mensaje,con actividad) El prefijo de *elemtype_actividad_participante* puede obtenerse aplicando la función de mapeo $\pi_{ET\_A\_P}$ como $\pi_{ET\_A\_P}(hd^k(\sigma)) = (\pi_{ET\_A\_P}(e_1), \pi_{ET\_A\_P}(e_2),$
$\ldots, \pi_{ET\_A\_P}(e_k))$. La predicción de próximo participante que enviará un mensaje,con actividad, es la definición de una función Θ_{pma} que toma el prefijo de eventos $hd^k(\sigma)$ donde $k \in [1, n-1]$ y predice el próximo *actividad_participante* e' , es decir:

$$\Theta_{pma}(hd^k(\sigma)) = \pi_{ET\_A\_P}(e'_{k+j}) \quad || \quad dummy$$

Siendo j la cantidad de eventos hasta encontrar el próximo envío de mensaje. *dummy* será el resultado cuando no haya un próximo envío de mensaje luego del evento actual.

Definición 10: (Predicción de Tiempo hasta el próximo mensaje a enviar) Sea $\sigma = \langle e_1, e_2, \ldots, e_n \rangle$ una traza de largo n y $\hat{\sigma} = \langle e_a, \ldots, e_y, e_z \rangle$, $y < z <= n$ una subsecuencia de σ donde $\hat{\sigma}$ contiene solo los eventos que son de tipo envío de mensaje.

Para extraer las características relacionadas con el tiempo del último evento e_n de esa traza, definimos las siguientes funciones:

$$fv_{t1}(\sigma) = \begin{cases} 0 & \text{si } |\hat{\sigma}| <= 1, \\ \pi_T(e_n) - \pi_T(e_z), & \text{en otro caso.} \end{cases}$$

$$fv_{t2}(\sigma) = \begin{cases} 0 & \text{si } |\hat{\sigma}| <= 2, \\ \pi_T(e_n) - \pi_T(e_y), & \text{en otro caso.} \end{cases}$$

$$fv_{t3}(\sigma) = \begin{cases} 0 & \text{si } |\sigma| = 1, \\ \pi_T(e_n) - \pi_T(e_0), & \text{en otro caso.} \end{cases}$$

La función fv_{t1} representa la diferencia de tiempo entre el último envío de mensaje y el evento actual de una traza . La función fv_{t2} contiene la diferencia de tiempo entre el evento actual y el penúltimo envío de mensaje. Finalmente, fv_{t3} muestra el tiempo aproximado transcurrido desde que el caso ha comenzado. La predicción de tiempo hasta el próximo mensaje a enviar , es la definición de una función Θ_{tm} que toma el prefijo de eventos $hd^k(\sigma)$ donde $k \in [1, n-1]$ y predice el momento en que probablemente ocurrirá el siguiente envío de mensaje , es decir:

$$\Theta_{tm}(\sigma', fv_{t_1}(\sigma'), fv_{t_2}(\sigma'), fv_{t_3}(\sigma')) = \begin{cases} \pi_T(e_{k+s}) - \pi_T(e_k) & \text{si } k+s < n, \\ 1 & \text{en otro caso.} \end{cases}$$

donde $\sigma' = hd^k(\sigma)$ y s la cantidad de eventos hasta encontrar el próximo envío de mensaje si existe.

Definición 11: (Predicción de Tiempo Restante de un participante)

Sea $\sigma = \langle e_1, e_2, \ldots, e_n \rangle$ una traza de largo n y $\hat{\sigma} = \langle e_a, \ldots, e_y, e_z \rangle$, $y < z <= n$ una subsecuencia de σ donde $\hat{\sigma}$ contiene solo los eventos del participante p.

Para extraer las características relacionadas con el tiempo del último evento e_n de esa traza, definimos las siguientes funciones:

$$fv_{t1}(\sigma) = \begin{cases} 0 & \text{si } |\hat{\sigma}| <= 1, \\ \pi_T(e_n) - \pi_T(e_z), & \text{en otro caso.} \end{cases}$$

$$fv_{t2}(\sigma) = \begin{cases} 0 & \text{si } |\hat{\sigma}| <= 2, \\ \pi_T(e_n) - \pi_T(e_y), & \text{en otro caso.} \end{cases}$$

$$fv_{t3}(\sigma) = \begin{cases} 0 & \text{si } |\sigma| = 1, \\ \pi_T(e_n) - \pi_T(e_0), & \text{en otro caso.} \end{cases}$$

La función fv_{t1} representa la diferencia de tiempo entre el último evento del participante p y el evento actual de una traza . La función fv_{t2} contiene la

diferencia de tiempo entre el evento actual y el penúltimo evento del participante p. Finalmente, fv_{t3} muestra el tiempo aproximado transcurrido desde que el caso ha comenzado. La predicción de tiempo restante de un participante , es la definición de una función Θ_{rtp} que toma el prefijo de eventos $hd^k(\sigma)$ donde $k \in [1, n-1]$ y predice el momento en que probablemente ocurrirá el siguiente envío de mensaje , es decir:

$$\Theta_{rtp}(\sigma', fv_{t_1}(\sigma'), fv_{t_2}(\sigma'), fv_{t_3}(\sigma')) = \begin{cases} \pi_T(e_z) - \pi_T(e_k) & \text{si } k < z, \\ 0 & \text{en otro caso.} \end{cases}$$

donde $\sigma' = hd^k(\sigma)$ y e_z el último evento en el que actuará el participante con $z <= n$.

Capítulo 5

Implementación de la solución

5.1. Diseño conceptual

Se plantea desarrollar una aplicación que permita realizar predicciones para procesos colaborativos, partiendo de un log extendido de colaboración. Para ello, se parte de la herramienta ProcessTransformer (Bukhsh y cols., 2021) y se extienden las capacidades de predicción a procesos colaborativos. Dicha herramienta deberá permitir la carga del log, la generación del modelo de predicción, la carga de trazas en curso, ejecución de predicción y la visualización de los resultados.

La aplicación fue desarrollada utilizando Flask, un microframework de Python que facilita la creación de aplicaciones web. La arquitectura de la aplicación se basa en una estructura que combina tanto el backend, implementado en Python, como el frontend, que utiliza HTML y javascript para la presentación de los datos. La aplicación se divide en tres grandes módulos: Gestión de logs, Gestión de modelos y Predicciones, que permiten realizar el flujo completo desde que se carga un log, se genera el modelo para un tipo de predicción y se realiza finalmente la predicción para las trazas cargadas.

5.2. Aplicación Web

A continuación se describen los módulos que componen la aplicación implementada.

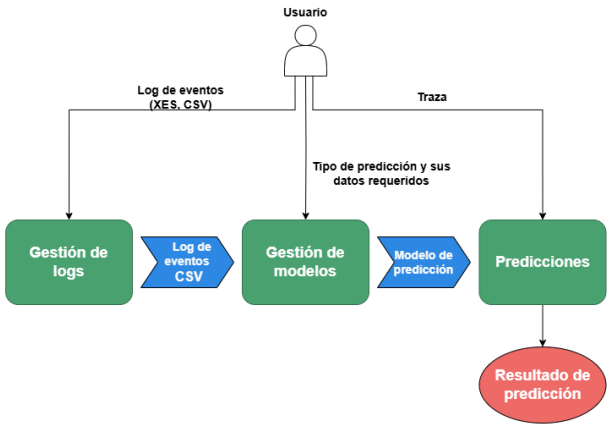


Figura 5.1: Diagrama de la aplicación

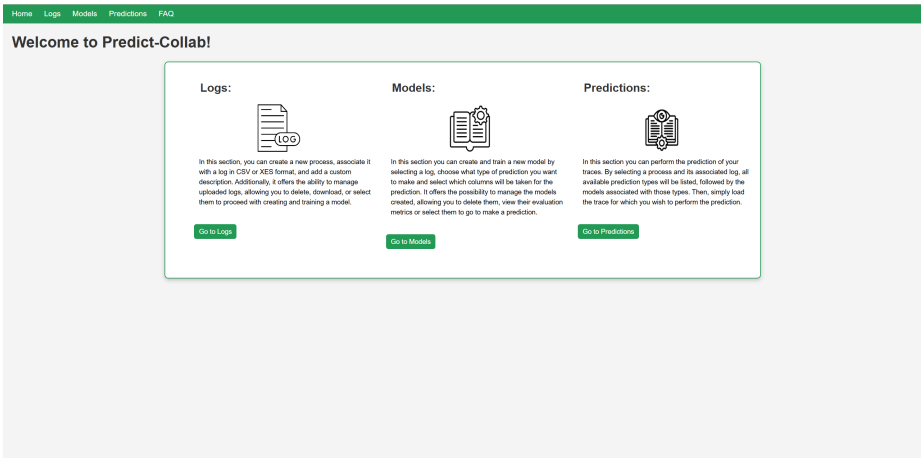


Figura 5.2: Pantalla de inicio de la aplicación

5.2.1. Gestión de logs

En este módulo se podrá cargar un nuevo log de eventos a la aplicación, tanto para un proceso que ya exista o generando una entrada para un nuevo proceso. La aplicación añade la posibilidad de cargar un log en formato .xes (un formato no soportado originalmente por ProcessTransformer) convirtiéndolo luego en .csv para utilizarlo en los siguientes módulos.

El nuevo log quedará asociado al proceso seleccionado y podrá ir acompañado de una descripción. Además se visualizan los logs ya cargados en el sistema, permitiendo las acciones de ir a generar un modelo con el log seleccionado, descargar y eliminar el log. En la figura 5.3 podemos ver la pantalla del módulo Gestión de logs.

Consideraciones:

- Los tipos de archivo aceptados para el log son .xes y .csv.
- Tanto el nombre de proceso como el nombre del archivo deben cumplir con el formato expresado con la expresión regular `^[a-zA-Z0-9-_]+$`
- En un mismo proceso no podrán cargarse dos logs con el mismo nombre, pero sí se pueden ingresar logs con mismo nombre a distintos procesos.
- Se debe tener en cuenta que solo se podrán eliminar logs que no tengan modelos asociados al momento de querer eliminarlo. Si se elige un log que tiene modelos asociados, el sistema indica que se deben eliminar previamente dichos modelos.

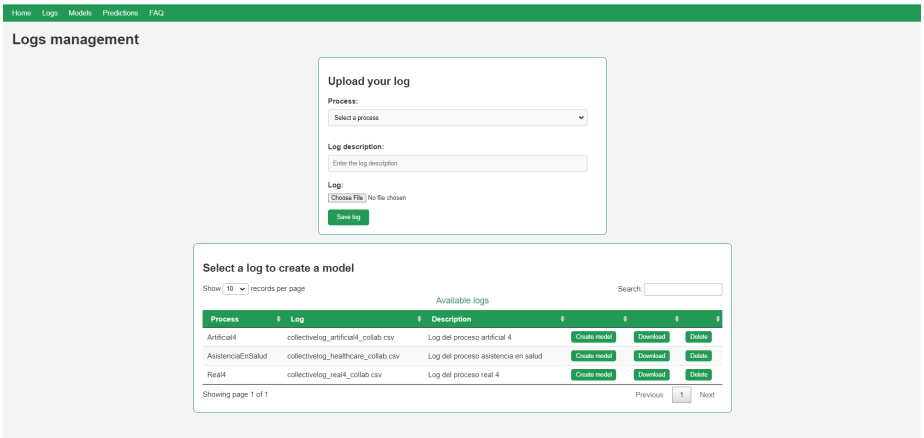


Figura 5.3: Pantalla del módulo Gestión de logs

5.2.2. Gestión de modelos

El módulo de gestión de modelos contiene la funcionalidad de mayor relevancia en la extensión propuesta. A partir de los datos que se ingresan en este módulo es que se genera el modelo de predicción que se utilizará para obtener las predicciones deseadas y es en esta etapa donde realizamos los cambios más significativos a lo provisto por el desarrollo del ProcessTransformer.

En esta sección se podrá crear y entrenar los modelos de predicción que quedarán asociados a un proceso, un log y un tipo de predicción. Además se visualizan los modelos ya cargados en el sistema, permitiendo las acciones de ir a generar una predicción con el modelo seleccionado, ver las métricas y eliminar el modelo. En la tabla 5.1 se detallan los campos requeridos en la creación de un modelo para los distintos tipos de predicción.

Consideraciones:

- Para que un modelo tenga sentido, cada traza del log de eventos debe contener un identificador del caso y un timestamp. Estos campos son imprescindibles en todas las predicciones.
- Las trazas deben contener sus eventos ordenados cronológicamente según el timestamp.
- Para las predicciones de tiempo se requiere elegir si el formato del mismo ($FormatoT$) será segundos, minutos, horas o días. Esto para que la predicción sea adecuada a la realidad del proceso. Este es un cambio respecto a lo implementado por el processTransformer donde todas las predicciones estaban realizadas en días. Con las pruebas realizadas se hizo evidente que esto no era adecuado para todos los procesos. Por ejemplo, en los logs utilizados las diferencias entre eventos, en su gran mayoría, era de segundos.

Tabla 5.1: Tipos de predicción y entradas requeridas

Tipo de predicción	**column1**	**column2**	**column3**	**FormatoT**	**Participante**
Próxima actividad	Actividad	-	-	NO	NO
Tiempo restante del proceso	Actividad	-	-	SI	NO
Tiempo hasta próximo evento	Actividad	-	-	SI	NO
Próximo participante que realizará una actividad	Participante	-	-	NO	NO
Tiempo hasta el próximo mensaje a enviar	*elemType*	-	-	SI	NO
Tiempo restante de un participante	Participante	-	-	SI	SI
Próximo participante que enviará un mensaje	*elemType*	Participante	-	NO	NO
Próxima actividad que ocurrirá y qué participante la realizará	Actividad	Participante	-	NO	NO
Próximo participante que enviará un mensaje (con actividad)	*elemType*	Actividad	Participante	NO	NO

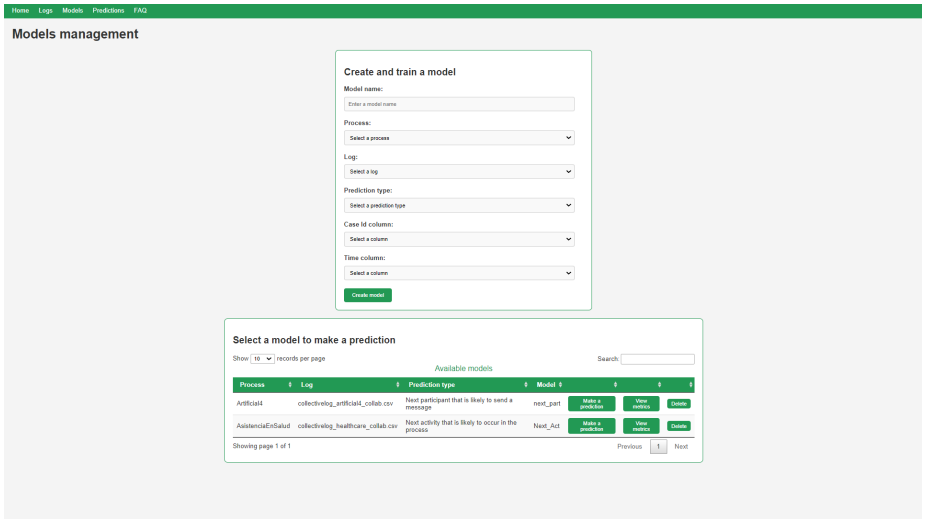


Figura 5.4: Pantalla del módulo Gestión de modelos

5.2.3. Predicciones

En este módulo es desde donde finalmente se podrán realizar las predicciones deseadas a partir de los modelos generados. La posibilidad de realizar una predicción a partir de trazas que no pertenecen al log original no estaba contemplada en el ProcessTransformer, por ello, se agrega esta funcionalidad. Adoptando los conceptos que el ProcessTransformer utiliza durante el preprocesamiento del log y la creación de las estructuras necesarias para generar el modelo de predicción, es que se pasa de tener una traza en el mismo formato que el log (csv o xes) a una estructura aceptada por el modelo para predecir. Se podrá ver mas en detalle esto en la sección 5.3. En la figura 5.5 podemos ver la pantalla del módulo Predicciones.

Para poder realizar una predicción se sigue el siguiente procedimiento:

1. **Selección del proceso:** Se elige el proceso sobre el cual se desea trabajar.
2. **Selección del log:** Se selecciona el log asociado al proceso elegido.
3. **Listado de tipos de predicciones:** Se mostrarán únicamente los tipos de predicciones para los cuales ya existen modelos creados.
4. **Selección del modelo:** A partir del proceso, log y tipo de predicción seleccionados, se podrá escoger un modelo previamente creado y entrenado.
5. **Carga de trazas:** Se debe subir un archivo que contenga la(s) traza(s) a predecir, considerando los siguientes requisitos:
   - **Formatos admitidos:** Los archivos deben ser .xes o .csv.
   - **Consistencia con el log:** El archivo debe tener el mismo formato que el log original, manteniendo los nombres de las columnas utilizadas durante la creación del modelo.
   - **Trazas incompletas:** Las trazas a predecir deben ser más cortas que la longitud máxima de los casos utilizados para entrenar el modelo.

Una vez realizadas las predicciones, los resultados se mostrarán en pantalla y estarán disponibles para descargar en un archivo .csv.

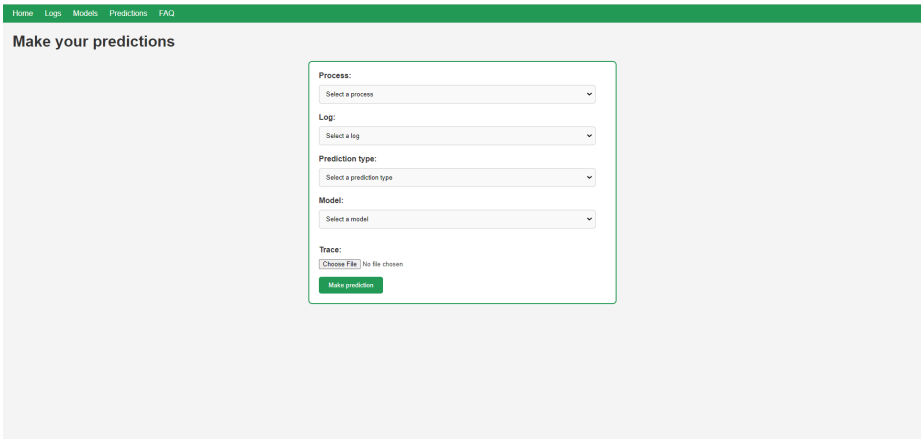


Figura 5.5: Pantalla del módulo Predicciones

5.3. Generación de modelo de predicción (Modelos.py)

La generación del modelo de predicción está compuesto por tres secciones que detallaremos a continuación.

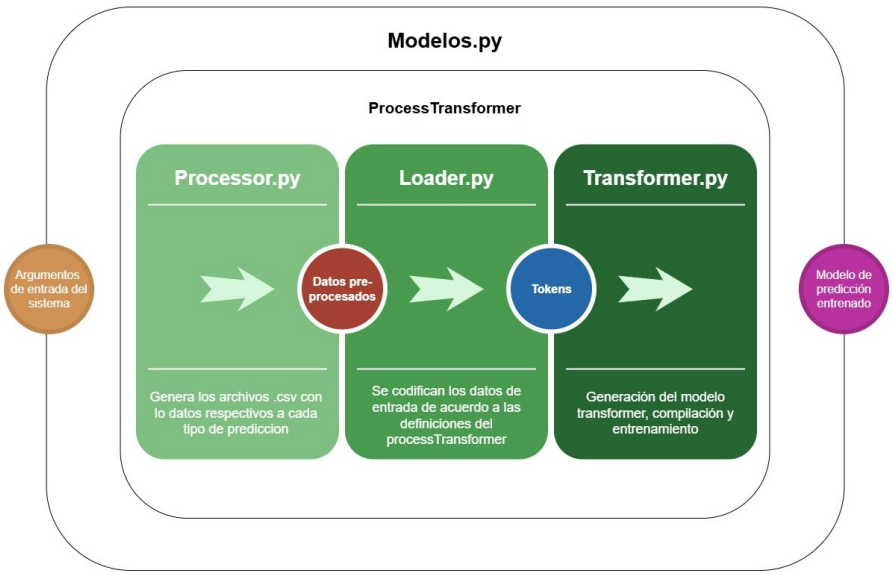


Figura 5.6: Secciones de la generación del modelo de predicción.

5.3.1. Preprocesamiento (Processor.py)

En esta sección se procesa el log y se generan los archivos .csv con los datos preprocesados para luego en el loader generar los tokens que utilizará el transformer para entrenar el modelo. Como estos archivos no son visualizados por el usuario y los datos contenidos en los mismos son utilizados posteriormente en funciones que no se modificaron, se mantiene el formato definido por el processTransformer (nombres cabezales y separación por ",").

Principales funciones:

- `data_processor = LogsDataProcessor(name='predict-collab', filepath=ruta_log, columns=[columnID, columnaFinal, columnT], dir_path=ruta_datasets, pool=4)`

  Esta función simplemente inicializa el entorno con el que va a empezar a procesar el log y no fue modificada en la extensión.

- `data_processor.process_logs(task=inputTask, participant=participant,formatoT=formatoT, sort_temporally=False)`

  Con el entorno generado se invoca a ésta función que pasará a procesar el log de acuerdo a los datos ingresados.

El valor `sort_temporally=False` es el que tenía por defecto el processTransformer por lo que lo dejamos incambiado y requerimos que las trazas vengan ordenadas por timestamp ya que no probamos el cambio de comportamiento al variar este atributo en el processTransformer.

Aquí se define el porcentaje de casos a tomar para train y test (definido en 80 % train y 20 % test en nuestro caso), se leen los datos del log (invocando a la función `_load_df` del processTransformer), y se extraen los metadatos que generan los diccionarios a usarse en la predicción (invocando a la función `_extract_logs_metadata` del processTrasformer).
Además se definen los dos subconjuntos de casos con el porcentaje indicado y se invoca a la función correspondiente según el tipo de predicción que se haya elegido (dada por el argumento "inputTask"), para realizar el preprocesamiento. Esta función fue modificada agregando los parámetros de entrada participant y formatoT, e internamente agregando invocaciones a las funciones definidas para los nuevos tipos de predicción.

A continuación se detallan las funciones para el preprocesamiento:

- Definidas por ProcessTransformer:
  - `_process_next_activity`
  - `_process_next_time`
  - `_process_remaining_time`
- Definidas en la extensión
  - `_process_next_message_send`
  - `_process_next_time_message`
  - `_process_remaining_time_participant`

Utilizamos el caso 459 para ejemplificar. Se muestran solo las columnas relevantes para las definiciones y se utilizará FormatoT en segundos para el cálculo de tiempos.

case:concept:name	time:timestamp	concept:name	collab:elemType	collab:participant
case_459	2021-06-17 07:17:07	Communicate disease	SendTask	Patient
case_459	2021-06-17 07:17:10	Receive disease info	ReceiveTask	Gynecologist
case_459	2021-06-17 07:17:13	Examine patient	task	Gynecologist
case_459	2021-06-17 07:17:18	Blood draw	task	Gynecologist
case_459	2021-06-17 07:17:21	Send blood sample	SendTask	Gynecologist
case_459	2021-06-17 07:17:25	Receive blood sample	ReceiveTask	Laboratory
case_459	2021-06-17 07:17:25	Analyse blood sample	task	Laboratory
case_459	2021-06-17 07:17:27	Send results	SendTask	Laboratory
case_459	2021-06-17 07:17:30	Receive blood analysis results	ReceiveTask	Gynecologist
case_459	2021-06-17 07:17:33	Send prescription	SendTask	Gynecologist
case_459	2021-06-17 07:17:38	Receive prescription	ReceiveTask	Patient

Figura 5.7: Columnas relevantes para ejemplo Caso 459

Preprocesamiento para predicción *Próxima Actividad:*

`_process_next_activity`:Esta función recibe un DataFrame con las columnas seleccionadas del log. Para cada traza (case_id) en el DataFrame, llama a la función auxiliar `_next_activity_helper_func`.

La función auxiliar descompone cada traza en X líneas,
con X = #eventos de la traza -1. Cada línea generada de la siguiente forma:

- **case_id:** Identificador único del caso.
- **prefix:** Las primeras $k+1$ actividades en el orden en que aparecen en el caso.
- **k:** Iterador que varía entre 0 y $X-1$.
- **next_act:** La actividad siguiente al prefijo, es decir, la actividad en la posición k + 2 de la traza.

Tabla 5.2: Ejemplo de salida de la función auxiliar *_next_activity_helper_func*

case_id	prefix	k	next_act
case_459	communicate-disease	0	receive-disease-info
case_459	communicate-disease receive-disease-info	1	examine-patient
case_459	communicate-disease receive-disease-info examine-patient	2	blood-draw
...	...	...	...
case_459	communicate-disease receive-disease-info examine-patient blood-draw send-blood-sample receive-blood-sample analyse-blood-sample send-results receive-blood-analysis-results	8	send-prescription
case_459	communicate-disease receive-disease-info examine-patient blood-draw send-blood-sample receive-blood-sample analyse-blood-sample send-results receive-blood-analysis-results send-prescription	9	receive-prescription

A partir de las líneas obtenidas para todos los casos, se generan los csv de train y test con el siguiente cabezal: case_id, prefix, k, next_act y los datos de las líneas separados por coma.

Para los casos de las nuevas predicciones:

- *Próximo participante que realizará una actividad*
- *Próxima actividad que ocurrirá y qué participante la realizará*

Se utiliza la misma función `_process_next_activity`, utilizando en el primer caso la columna Participante y el el segundo la nueva columna Actividad_Participante, producto de la concatenación de las columnas Actividad y Participante.

Tabla 5.3: Ejemplo de salida de función auxiliar para predicción *Próximo participante que realizará una actividad*

case_id	prefix	k	next_act
case_459	patient	0	gynecologist
case_459	patient gynecologist	1	gynecologist
case_459	patient gynecologist gynecologist	2	gynecologist
...	...	...	...
case_459	patient gynecologist gynecologist gynecologist gynecologist gynecologist laboratory laboratory laboratory gynecologist	8	gynecologist
case_459	patient gynecologist gynecologist gynecologist gynecologist gynecologist laboratory laboratory laboratory gynecologist gynecologist	9	patient

Tabla 5.4: Ejemplo de salida de función auxiliar para predicción *Próxima actividad que ocurrirá y qué participante la realizará*

case_id	prefix	k	next_act
case_459	communicate-disease_patient	0	receive-disease-info_gynecologist
case_459	communicate-disease_patient receive-disease-info_gynecologist	1	examine-patient_gynecologist
case_459	communicate-disease_patient receive-disease-info_gynecologist examine-patient_gynecologist	2	blood-draw_gynecologist
...	...	...	...
case_459	communicate-disease_patient receive-disease-info_gynecologist examine-patient_gynecologist blood-draw_gynecologist send-blood-sample_gynecologist receive-blood- sample_laboratory analyse-blood- sample_laboratory send-results_laboratory receive-blood-analysis- results_gynecologist	8	send-prescription_gynecologist
case_459	communicate-disease_patient receive-disease-info_gynecologist examine-patient_gynecologist blood-draw_gynecologist send-blood-sample_gynecologist receive-blood- sample_laboratory analyse-blood- sample_laboratory send-results_laboratory receive-blood-analysis- results_gynecologist send-prescription_gynecologist	9	receive-prescription_patient

Preprocesamiento para predicción *Tiempo hasta próximo evento:*

`_process_next_time`:Esta función recibe un DataFrame con las columnas seleccionadas del log. Para cada traza (case_id) en el DataFrame, llama a la función auxiliar `_next_time_helper_func`.

La función auxiliar descompone cada traza en X líneas,
con $X = \#\text{eventos de la traza } - 1$. Cada línea generada de la siguiente forma:

- **case_id:** Identificador único del caso.
- **prefix:** Las primeras $k + 1$ actividades en el orden en que aparecen en el caso.
- **k:** Iterador que varía entre 0 y $X - 1$.
- **time_passed:** Tiempo que pasó desde que inició el caso (def fv_{t3})
- **recent_time:** Tiempo entre la actividad actual y la previa a la anterior (def fv_{t2})
- **lastest_time:** Tiempo entre la actividad actual y la anterior(def fv_{t1})
- **next_time:** $\begin{cases} \text{Tiempo entre la actividad } k+1 \text{ y } k+2, & \text{cuando } k+1 < X, \\ 1, & \text{cuando } k+1 = X. \end{cases}$

A partir de las líneas obtenidas para todos los casos, se generan los csv de train y test con el siguiente cabezal: case_id,prefix,k,time_passed,recent_time, latest_time,next_time y los datos de las líneas separados por coma.

Tabla 5.5: Ejemplo de salida de la función auxiliar *_next_time_helper_func*

case_id	prefix	k	time_ passed	recent _time	latest _time	next _time
case_459	communicate-disease	0	0	0	0	3
case_459	communicate-disease receive-disease-info	1	3	0	3	3
case_459	communicate-disease receive-disease-info examine-patient	2	6	6	3	5
...	...	...	...	...	...	...
case_459	communicate-disease receive-disease-info examine-patient blood-draw send-blood-sample receive-blood-sample analyse-blood-sample send-results receive-blood-analysis-results send-prescription	9	26	6	3	5
case_459	communicate-disease receive-disease-info examine-patient blood-draw send-blood-sample receive-blood-sample analyse-blood-sample send-results receive-blood-analysis-results send-prescription receive-prescription	10	31	8	5	1

Preprocesamiento para predicción *Tiempo restante del proceso*

`_process_remaining_time`:Esta función recibe un DataFrame con las columnas seleccionadas del log. Para cada traza (case_id) en el DataFrame, llama a la función auxiliar `_remaining_time_helper_func`.

La función auxiliar descompone cada traza en X líneas,
con $X = \#\text{eventos de la traza } - 1$. Cada línea generada de la siguiente forma:

- **case_id:** Identificador único del caso.
- **prefix:** Las primeras $k + 1$ actividades en el orden en que aparecen en el caso.
- **k:** Iterador que varía entre 0 y $X - 1$.
- **time_passed:** Tiempo que pasó desde que inició el caso (def fv_{t3})
- **recent_time:** Tiempo entre la actividad actual y la previa a la anterior (def fv_{t2})
- **lastest_time:** Tiempo entre la actividad actual y la anterior(def fv_{t1})
- **remaining_time_days:** Tiempo entre la actividad $k+1$ y la última actividad del caso

Tabla 5.6: Ejemplo de salida de la función auxiliar *_remaining_time_helper_func*

case_id	prefix	k	time _passed	recent _time	latest _time	remaining _time_days
case_459	communicate-disease	0	0	0	0	31
case_459	communicate-disease receive-disease-info	1	3	0	3	28
case_459	communicate-disease receive-disease-info examine-patient	2	6	6	3	25
...	...	...	...	...	...	...
case_459	communicate-disease receive-disease-info examine-patient blood-draw send-blood-sample receive-blood-sample analyse-blood-sample send-results receive-blood-analysis-results send-prescription	9	26	6	3	5
case_459	communicate-disease receive-disease-info examine-patient blood-draw send-blood-sample receive-blood-sample analyse-blood-sample send-results receive-blood-analysis-results send-prescription receive-prescription	10	31	8	5	0

A partir de las líneas obtenidas para todos los casos, se generan los csv de train y test con el siguiente cabezal: case_id,prefix,k,time_passed,recent_time, latest_time,remaining_time_days y los datos de las líneas separados por coma

Preprocesamiento para predicción *Próximo participante que enviará un mensaje:*

`_process_next_message_send`: Esta función recibe un DataFrame con las columnas seleccionadas del log, en este caso cada item del prefijo proviene de la nueva columna concatenada `collab:elemType_collab:participant`. Para cada traza (case_id) en el DataFrame, llama a la función auxiliar `_next_message_send_helper_func`.

La función auxiliar descompone cada traza en X líneas,
con $X =$ #eventos de la traza -1. Cada línea generada de la siguiente forma:

- **case_id:** Identificador único del caso.
- **prefix:** Las primeras $k+1$ actividades en el orden en que aparecen en el caso.
- **k:** Iterador que varía entre 0 y $X-1$.
- **next_act:** $\begin{cases} \text{El ítem siguiente al prefijo con} & \\ \texttt{collab:elemType = sendtask} & \text{si existe} \\ \\ \texttt{dummy} & \text{en otro caso.} \end{cases}$

Tabla 5.7: Ejemplo de salida de la función auxiliar *_next_message_send_helper_func*

case_id	prefix	k	next_act
case_459	sendtask_patient	0	sendtask_gynecologist
case_459	sendtask_patient receivetask_gynecologist	1	sendtask_gynecologist
case_459	sendtask_patient receivetask_gynecologist task_gynecologist	2	sendtask_gynecologist
case_459	sendtask_patient receivetask_gynecologist task_gynecologist task_gynecologist	3	**sendtask_gynecologist**
case_459	sendtask_patient receivetask_gynecologist task_gynecologist task_gynecologist **sendtask_gynecologist**	4	sendtask_laboratory
...	...	...	...
case_459	sendtask_patient receivetask_gynecologist task_gynecologist task_gynecologist sendtask_gynecologist receivetask_laboratory task_laboratory sendtask_laboratory receivetask_gynecologist sendtask_gynecologist	9	dummy

Para el caso de la nueva predicción *Próximo participante que enviará un mensaje (con actividad)* se utiliza la misma función `_process_next_message_send`, donde cada ítem del prefijo proviene de la concatenación de las columnas `collab:elemType`, `concept:name` y `collab:participant`

Tabla 5.8: Ejemplo de salida de función auxiliar para predicción *Próximo participante que enviará un mensaje (con actividad)*

case_id	prefix	k	next_act
case_459	sendtask_communicate-disease_patient	0	sendtask_send-blood-sample_gynecologist
case_459	sendtask_communicate-disease_patient receivetask_receive-disease-info_gynecologist	1	sendtask_send-blood-sample_gynecologist
case_459	sendtask_communicate-disease_patient receivetask_receive-disease-info_gynecologist task_examine-patient_gynecologist	2	sendtask_send-blood-sample_gynecologist
case_459	sendtask_communicate-disease_patient receivetask_receive-disease-info_gynecologist task_examine-patient_gynecologist task_blood-draw_gynecologist	3	**sendtask_send-blood-sample_gynecologist**
case_459	sendtask_communicate-disease_patient receivetask_receive-disease-info_gynecologist task_examine-patient_gynecologist task_blood-draw_gynecologist sendtask_send-blood-sample_gynecologist	4	sendtask_send-results_laboratory
...	...	...	...
case_459	sendtask_communicate-disease_patient receivetask_receive-disease-info_gynecologist task_examine-patient_gynecologist task_blood-draw_gynecologist sendtask_send-blood-sample_gynecologist receivetask_receive-blood-sample_laboratory task_analyse-blood-sample_laboratory sendtask_send-results_laboratory receivetask_receive-blood-analysis-results_gynecologist sendtask_send-prescription_gynecologist	9	dummy

Preprocesamiento para predicción *Tiempo hasta el próximo mensaje a enviar:*

`_process_next_time_message:`Esta función recibe un DataFrame con las columnas seleccionadas del log. En este caso cada item del prefijo proviene de la columna `collab:elemType`. Para cada traza (case_id) en el DataFrame, llama a la función auxiliar `_next_time_message_helper_func.`

La función auxiliar descompone cada traza en X líneas,
con X = #eventos de la traza -1. Cada línea generada de la siguiente forma:

- **case_id:** Identificador único del caso.
- **prefix:** Las primeras $k+1$ actividades en el orden en que aparecen en el caso.
- **k:** Iterador que varía entre 0 y $X-1$.
- **time_passed:** Tiempo que pasó desde que inició el caso (def fv_{t3m})
- **recent_time:** Tiempo entre el ítem actual y el previo al anterior que sea ítem=`sendtask` (def fv_{t2m})
- **lastest_time:** Tiempo entre el ítem actual y el anterior ítem=`sendtask` (def fv_{t1m})
- **next_time:**
$$\begin{cases} \text{Tiempo entre el ítem } k+1 \\ \text{ y el siguiente ítem=}\texttt{sendtask} & \text{si existe} \\ \\ 1 & \text{en otro caso} \end{cases}$$

Tabla 5.9: Ejemplo de salida de la función auxiliar *_next_send_message_helper_func*

case_id	prefix	k	time _passed	recent_ time	latest_ time	next_ time
case_459	sendtask	0	0	0	0	3
case_459	sendtask receivetask	1	3	0	3	11
case_459	sendtask receivetask task	2	6	0	6	5
case_459	sendtask receivetask task task	3	11	0	11	3
case_459	sendtask receivetask task task sendtask	4	14	0	14	6
case_459	sendtask receivetask task task sendtask receivetask	5	18	18	4	2
...	...	...	...	...	...	...
case_459	sendtask receivetask task task sendtask receivetask task sendtask receivetask sendtask	9	26	12	6	1
case_459	sendtask receivetask task task sendtask receivetask task sendtask receivetask sendtask receivetask	10	31	11	5	1

Preprocesamiento para predicción *Tiempo restante del participante:*

`_process_remaining_time_participant:`Esta función recibe un DataFrame con las columnas seleccionadas del log.
En este caso cada item del prefijo proviene de la columna `collab:participant` y se define **p**=*participante seleccionado*. Para cada traza (case_id) en el DataFrame, llama a la función auxiliar `_remaining_time_participant_helper_func.`

La función auxiliar descompone cada traza en X líneas,
con X = #eventos de la traza -1. Cada línea generada de la siguiente forma:

- **case_id:** Identificador único del caso.
- **prefix:** Las primeras $k+1$ actividades en el orden en que aparecen en el caso.
- **k:** Iterador que varía entre 0 y $X-1$.
- **time_passed:** Tiempo que pasó desde que inició el caso (def fv_{t3p})

- **recent_time:** Tiempo entre el ítem actual y el previo al anterior que sea ítem=**p** (def fv_{t2p})
- **lastest_time:** Tiempo entre el ítem actual y el anterior ítem=**p** (def fv_{t1p})
- **remaining_time_days:** $\begin{cases} \text{Tiempo entre el ítem } k+1 \\ \text{y el último ítem=}\mathbf{p} & \text{si existe} \\ \\ \text{posición(ítem)} > k+1 & \text{en otro caso} \end{cases}$

A partir de las líneas obtenidas para todos los casos, se generan los csv de train y test con el siguiente cabezal: case_id,prefix,k,time_passed,recent_time,latest_time, next_time y los datos de las líneas separados por coma.

Tabla 5.10: Ejemplo de salida de la función auxiliar *_remaining_time_participant_helper_func*, tomando **p**=*gynecologist*

case_id	prefix	k	time _passed	recent_ time	latest_ time	remaining _time_days
case_459	patient	0	0	0	0	26
case_459	patient gynecologist	1	3	0	0	23
case_459	patient gynecologist gynecologist	2	6	0	3	20
case_459	patient gynecologist gynecologist gynecologist	3	11	8	5	15
case_459	patient gynecologist gynecologist gynecologist gynecologist	4	14	8	3	12
case_459	patient gynecologist gynecologist gynecologist gynecologist laboratory	5	18	7	4	8
...	...	...	...	...	...	...
case_459	patient gynecologist gynecologist gynecologist laboratory laboratory laboratory gynecologist gynecologist	9	26	12	3	0
case_459	patient gynecologist gynecologist gynecologist laboratory laboratory laboratory gynecologist gynecologist patient	10	31	8	5	0

Todas las funciones auxiliares (`helper_func`) se utilizan en conjunto con la clase Pool del módulo *multiprocessing* para ejecutar el procesamiento en paralelo sobre cada fragmento del DataFrame (cada fragmento es un caso). La clase Pool permite distribuir los fragmentos entre varios procesos, y cada proceso aplica la

función auxiliar sobre su fragmento correspondiente. De este modo, se optimiza el tiempo de ejecución al paralelizar la carga de trabajo.

5.3.2. Carga de los datos preprocesados (Loader.py)

A partir de los csv y diccionarios generados en el processor.py, este módulo carga los datos preprocesados y genera las estructuras necesarias para la compilación y entrenamiento del modelo.

Principales funciones:

- `data_loader = LogsDataLoader(name='predict-collab', dir_path=ruta_datasets)`

  Esta función inicializa `data_loader` indicando donde se encuentran los datos preprocesados.

- `(train_df, test_df, x_word_dict, y_word_dict, max_case_length, vocab_size, num_output) = data_loader.load_data(inputTask, participant)`

  En el módulo loader.py solo modificamos la función `load_data(self, task,participant)` para que acepte como parámetro al `participant` (necesario para la predicción Tiempo Restante de un participante) y que el parámetro `task` permita las nuevas predicciones agregadas. Además se agrega también al diccionario de predicción la opción "dummy" para las predicciones que involucran el envío de un próximo mensaje.

- `train_token_x, train_token_y = data_loader.prepare_data_next_activity (train_df,x_word_dict, y_word_dict, max_case_length)`

  En esta parte se separa el flujo de la ejecución dependiendo el tipo de predicción. Para cada tipo de predicción se invoca su respectiva función `prepare_data` que genera los tokens necesarios para entrenar el modelo de predicción. Se debe tener en cuenta que solo se dejan las tres funciones originales ya que la parte donde se agregan cambios a las nuevas predicciones ocurre en el preprocesamiento y ya a partir de acá se trabaja de acuerdo a lo implementado por el processTransformer.

Por lo Tanto, `prepare_data_next_activity` prepara datos para las predicciones:

- Próxima actividad
- Próximo participante que enviará un mensaje:
- Próximo participante que enviará un mensaje (con actividad)
- Próxima actividad que ocurrirá y qué participante la realizará
- Próximo participante que realizará una actividad

`prepare_data_next_time` prepara datos para las predicciones:

- Tiempo hasta el próximo evento
- Tiempo hasta el próximo mensaje a enviar

`prepare_data_remaining_time` prepara datos para las predicciones:

- Tiempo restante del caso
- Tiempo restante del participante

5.3.3. Generación del transformer, compilación y entrenamiento (Transformer.py)

En este sección se genera el modelo transformer (`transformer_model`) , con las funciones que prevé processTransformer para cada tipo de predicción (`get_next_activity_model` , `get_next_time_model` y `get_remaining_time_model`) usando la librería keras.

Una vez obtenido el modelo, se compila y se entrena con las funciones `compile` y `fit` de keras. Una vez realizado esto, se guarda el modelo con sus propiedades para que luego se puedan realizar predicciones con el modelo ya entrenado.

5.4. Evaluación de modelo

Con el modelo ya generado a partir de los datos de entrenamiento (train.csv), se procede a realizar el cálculo de métricas (3.3.3). Para ello, se cargan los datos en las variables necesarias para la predicción, pero esta vez tomando los datos para pruebas (test.csv).

Con los resultados de la predicción para los datos de prueba (`y_pred`) y con los datos reales de prueba, (`test_token_y` o `_test_y`) se calculan las métricas utilizando el módulo *metrics* de la biblioteca *scikit-learn*, que proporciona funciones para evaluar el rendimiento de modelos de machine learning.

Para predicciones basadas en próxima actividad:

- `accuracy = metrics.accuracy_score(test_token_y, y_pred)`
- `precision, recall, fscore = metrics.precision_recall_fscore_support(test_token_y, y_pred)`

Para predicciones de tiempo:

- `maes = metrics.mean_absolute_error(_test_y, _y_pred)`
- `mses = metrics.mean_squared_error(_test_y, _y_pred)`
- `rmses = metrics.mean_squared_error(_test_y, _y_pred)`

5.5. Preprocesamiento de trazas para la predicción

Con los datos ingresados en la aplicación para realizar la predicción, se carga el modelo elegido y sus propiedades asociadas para generar un preprocesamiento similar a lo realizado en la generación del modelo, pero para el archivo de trazas cargado.

Dependiendo del tipo de predicción seleccionada, si se necesita la concatenación de columnas, se realizará de igual forma a como se hizo previamente en la generación del modelo asociado.

Para cada traza obtenida del archivo, se toman los valores de la columna definida para componer el prefijo del caso, y junto con el diccionario, se traduce para generar la entrada del token correspondiente. En los casos de predicciones de tiempo, se agrega un token de tiempo que se genera tomando los valores de la columna de tiempo, de la misma manera que se hizo cuando se generó el modelo. Una vez que los tokens necesarios contienen las entradas para todas las trazas se procede a realizar la predicción invocando a la función predict de Keras, `modelo.predict([token_x,time_x])` para las predicciones de tiempo y `modelo.predict(token_x)` para las basadas en próxima actividad.

Capítulo 6

Evaluación

En este capítulo se analizarán los resultados obtenidos de aplicar la solución desarrollada a un log de eventos. Se realizaron todos los tipos de predicciones para el log extendido *collectivelog_healthcare_collab.xes* (extensión del log utilizado en (Corradini, Re, Rossi, y Tiezzi, 2022)). Se toma como entrada para las predicciones tres trazas que formaban parte del log extendido y que fueron quitadas para ser utilizadas en la experimentación.

También se evaluará la solución para otros logs de eventos, analizando las métricas obtenidas de la generación de distintos modelos.

6.1. Caso de estudio: Asistencia en salud

El modelo de colaboración de la Figura 6.1 ilustra un escenario en el que se combinan las actividades de un *Paciente*, un *Ginecólogo*, un *Laboratorio* y un *Hospital* de la siguiente manera. El *Paciente* proporciona detalles sobre su estado de salud y espera información relacionada con el tratamiento domiciliario o la hospitalización. El *Ginecólogo* coordina las actividades del *Laboratorio* y del *Hospital*, ocupándose de los análisis de sangre y de la hospitalización respectivamente. La colaboración comienza cuando la Paciente envía la información sobre la enfermedad al *Ginecólogo*. A continuación, el *Ginecólogo* examina a la *Paciente* y, paralelamente, extrae una muestra de sangre y la envía al *Laboratorio*. El *Laboratorio* analiza la muestra y devuelve los resultados al *Ginecólogo*. Una vez examinada la *Paciente* y recibidos los resultados de los análisis, el *Ginecólogo* decide si envía una receta médica u hospitaliza a la *Paciente*, e informa a la Paciente al respecto. Sólo en este último caso, el *Ginecólogo* pone en marcha el *Hospital* solicitando el ingreso de la *Paciente* y enviando los resultados de los análisis. Cuando el *Hospital* inicia su proceso, crea una historia clínica para la *Paciente*, y luego decide si considera los resultados del análisis de sangre ya realizado o solicita un nuevo análisis; en cualquier caso, luego envía la información de ingreso a la *Paciente*. (Corradini y cols., 2022)

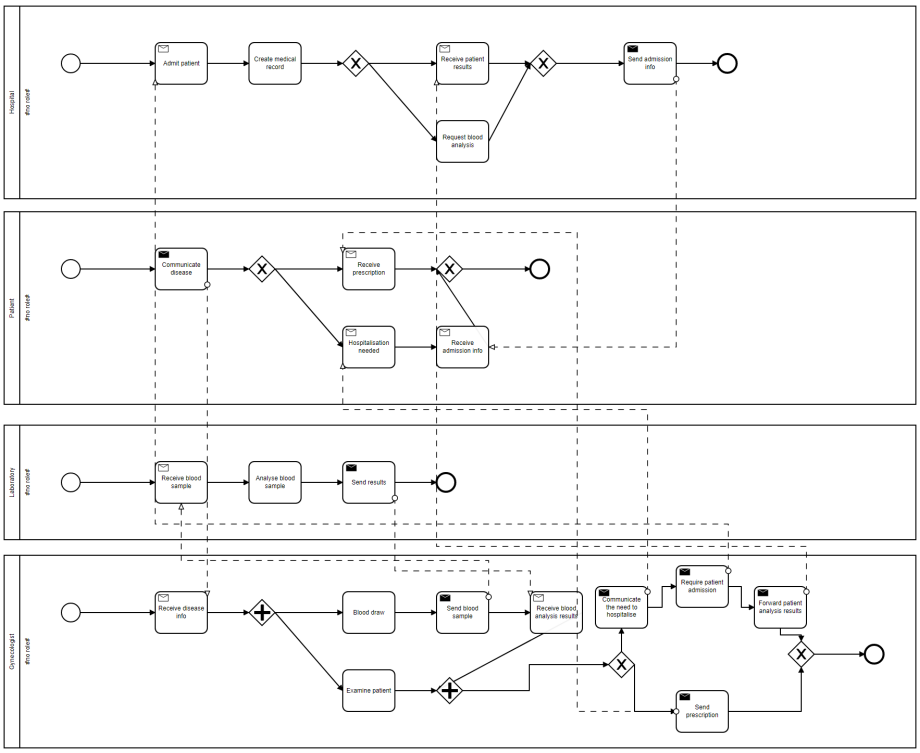


Figura 6.1: colaboración en el proceso de negocio Asistencia en salud

6.1.1. Predicciones sobre el caso de estudio

Analizando el log utilizado con la herramienta Disco(*Fluxicon Disco*, s.f.), obtuvimos algunos datos que nos parece relevante mencionar para la evaluación de los resultados:

- Contiene 100 casos en total, distribuidos en 47 variantes.
- Promedio duración de todos los casos: 49.1 segundos
- Las primeras 9 actividades se ejecutan en gran medida en el mismo orden en todos los casos, con mínimas variaciones.
- El 50 % de los casos tiene 11 eventos y el otro 50 % tiene 18.
- Promedio duración de casos de 18 eventos: 66.9 segundos

Para realizar la experimentación quitamos las trazas `case_44`, `case_9`, `case_209`, donde `case_44`, `case_9` eran de la variante 1 (29 casos), mientras que `case_209` de la variante 4 (3 casos). Se debe tener en cuenta que no se conoce previamente cuáles de los 97 casos restantes pertenecerán a los conjuntos de entrenamiento y de prueba.

Para visualizar mejor estos datos, observar las imágenes de la sección A.2 del anexo.

En la Figura 6.2 se muestran las trazas completas para estos casos, conteniendo sólo las columnas relevantes que se tomarán para las predicciones. En celeste se marca hasta que evento se cargará en las trazas incompletas, que serán la entrada de las pruebas. Para cada predicción se presentarán las entradas para la creación del modelo, las entradas para la generación de la predicción y los resultados de la misma.

case:concept:name	time:timestamp	concept:name	collab:elemType	collab:participant
case_44	2021-06-17 06:06:35	Communicate disease	SendTask	Patient
case_44	2021-06-17 06:06:38	Receive disease info	ReceiveTask	Gynecologist
case_44	2021-06-17 06:06:42	Examine patient	task	Gynecologist
case_44	2021-06-17 06:06:43	Blood draw	task	Gynecologist
case_44	2021-06-17 06:06:44	Send blood sample	SendTask	Gynecologist
case_44	2021-06-17 06:06:46	Receive blood sample	ReceiveTask	Laboratory
case_44	2021-06-17 06:06:48	Analyse blood sample	task	Laboratory
case_44	2021-06-17 06:06:51	Send results	SendTask	Laboratory
case_44	2021-06-17 06:06:52	Receive blood analysis results	ReceiveTask	Gynecologist
case_44	2021-06-17 06:06:55	Send prescription	SendTask	Gynecologist
case_44	2021-06-17 06:06:58	Receive prescription	ReceiveTask	Patient
case_9	2021-06-17 05:59:26	Communicate disease	SendTask	Patient
case_9	2021-06-17 05:59:27	Receive disease info	ReceiveTask	Gynecologist
case_9	2021-06-17 05:59:32	Examine patient	task	Gynecologist
case_9	2021-06-17 05:59:33	Blood draw	task	Gynecologist
case_9	2021-06-17 05:59:36	Send blood sample	SendTask	Gynecologist
case_9	2021-06-17 05:59:39	Receive blood sample	ReceiveTask	Laboratory
case_9	2021-06-17 05:59:40	Analyse blood sample	task	Laboratory
case_9	2021-06-17 05:59:40	Send results	SendTask	Laboratory
case_9	2021-06-17 05:59:44	Receive blood analysis results	ReceiveTask	Gynecologist
case_9	2021-06-17 05:59:44	Send prescription	SendTask	Gynecologist
case_9	2021-06-17 05:59:49	Receive prescription	ReceiveTask	Patient
case_209	2021-06-17 06:31:58	Communicate disease	SendTask	Patient
case_209	2021-06-17 06:32:00	Receive disease info	ReceiveTask	Gynecologist
case_209	2021-06-17 06:32:06	Examine patient	task	Gynecologist
case_209	2021-06-17 06:32:08	Blood draw	task	Gynecologist
case_209	2021-06-17 06:32:11	Send blood sample	SendTask	Gynecologist
case_209	2021-06-17 06:32:15	Receive blood sample	ReceiveTask	Laboratory
case_209	2021-06-17 06:32:16	Analyse blood sample	task	Laboratory
case_209	2021-06-17 06:32:20	Send results	SendTask	Laboratory
case_209	2021-06-17 06:32:28	Receive blood analysis results	ReceiveTask	Gynecologist
case_209	2021-06-17 06:32:31	Communicate the need to hospitalise	SendTask	Gynecologist
case_209	2021-06-17 06:32:40	Hospitalisation needed	ReceiveTask	Patient
case_209	2021-06-17 06:32:46	Require patient admission	SendTask	Gynecologist
case_209	2021-06-17 06:32:49	Admit patient	ReceiveTask	Hospital
case_209	2021-06-17 06:32:55	Create medical record	task	Hospital
case_209	2021-06-17 06:33:02	Forward patient analysis results	SendTask	Gynecologist
case_209	2021-06-17 06:33:07	Receive patient results	ReceiveTask	Hospital
case_209	2021-06-17 06:33:07	Send admission info	SendTask	Hospital
case_209	2021-06-17 06:33:11	Receive admission info	ReceiveTask	Patient

Figura 6.2: Principales variantes de collectivelog_healthcare_collab.xes

Próxima actividad

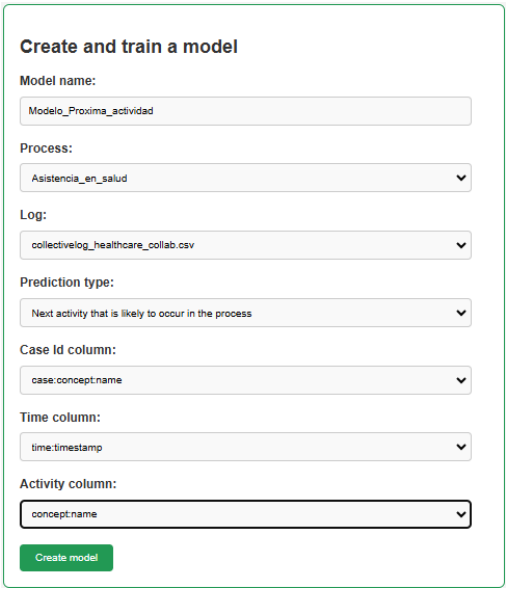


Figura 6.3: Creación de modelo - Próxima actividad

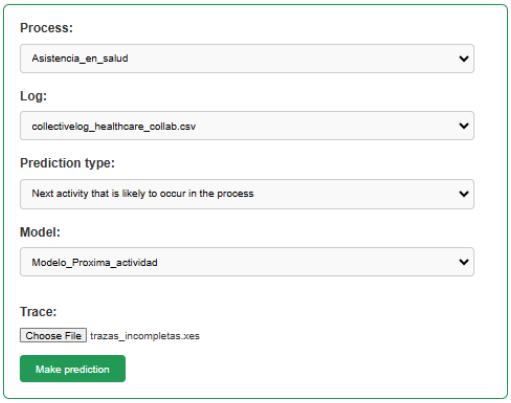


Figura 6.4: Generación de la predicción - Próxima actividad

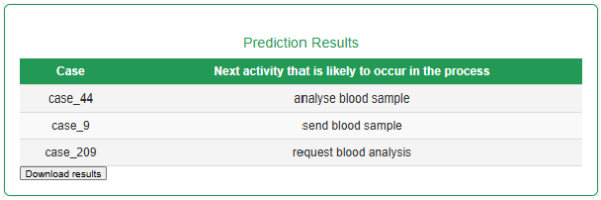


Figura 6.5: Resultado - Próxima actividad

Vemos que para los casos 44 y 9 la predicción coincide con lo esperado, mientras que para el caso 209 no. Entendemos que esto está relacionado a las variantes a las que pertenecen los casos y por lo tanto tiene sentido el resultado obtenido.

Tiempo hasta próximo evento

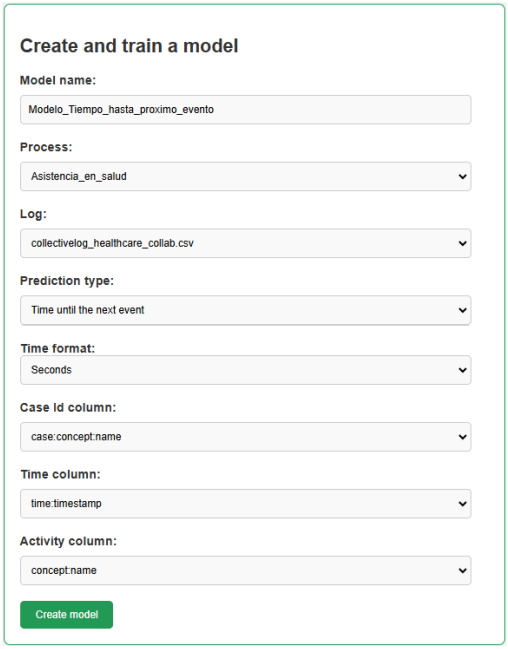
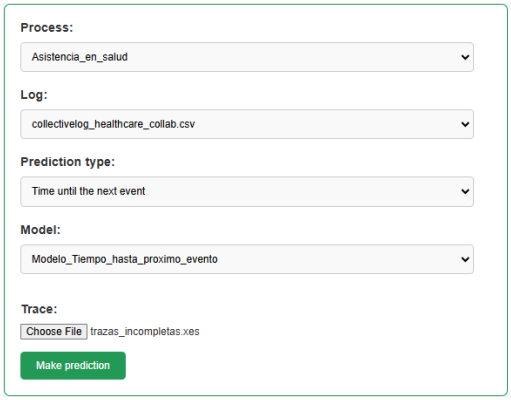


Figura 6.6: Creación de modelo - Tiempo hasta próximo evento



Generación de la predicción - Tiempo hasta próximo evento

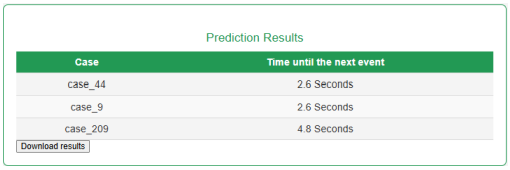


Figura 6.7: Resultado - Tiempo hasta próximo evento

Tabla 6.1: Comparativa para predicción Tiempo hasta próximo evento

Caso	**Valor esperado**	**Resultado**
Caso_44	2	2.6
Caso_9	3	2.6
Caso_209	5	4.8

Vemos que los resultados se encuentran en el orden de lo esperado ya que el MAE para este modelo es de 1.6 segundos.

Tiempo restante del proceso

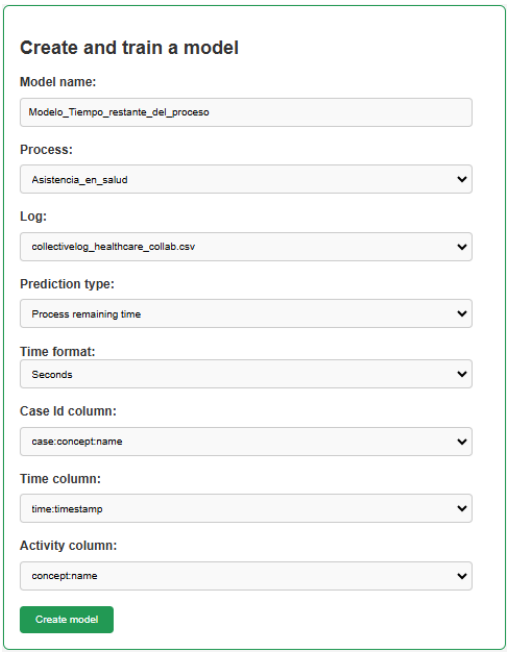


Figura 6.8: Creación de modelo - Tiempo restante del proceso

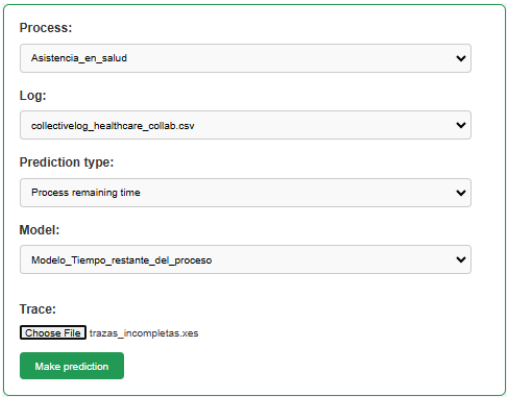


Figura 6.9: Generación de la predicción - Tiempo restante del proceso

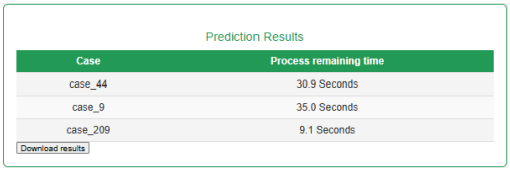


Figura 6.10: Resultado - Tiempo restante del proceso

Tabla 6.2: Comparativa para predicción Tiempo restante del proceso

Caso	**Tiempo actual**	**Tiempo restante (dato)**	**Tiempo restante (predicción)**
Caso_44	11	12	30.9
Caso_9	7	16	35
Caso_209	64	9	9.1

En los `caso_44` y `caso_9`, creemos que las diferencias observadas entre los valores reales y el resultado de la predicción se debe a la combinación de 2 factores: por un lado el tiempo promedio de todos los casos es de 49.1 segundos, por otro, como todos los casos comienzan con las mismas 9 actividades y las trazas 44 y 9 fueron cargadas con 6 y 4 actividades respectivamente. Por lo anterior, entendemos razonables los resultados obtenidos ya que los tiempos de finalización de cada caso no distan mucho del promedio: 11+30.9 = 41.9 para `caso_44` y 7+35= 44 para `caso_9`.

En el `caso_209`, la traza incompleta cuenta con 15 eventos, por lo tanto el resultado se ajusta más al promedio de duración de los casos de 18 eventos (66.9 segundos).

Próximo participante que realizará una actividad

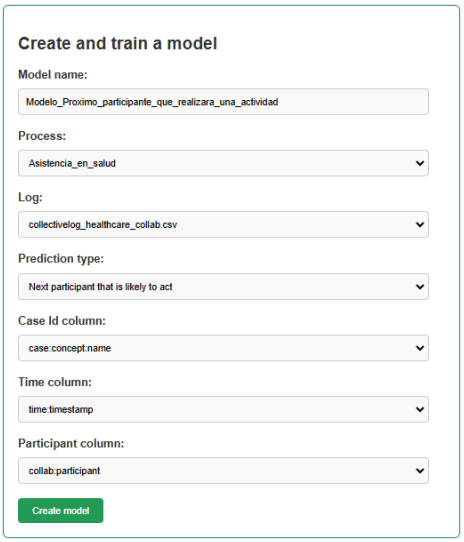


Figura 6.11: Creación de modelo - Próximo participante que realizará una actividad

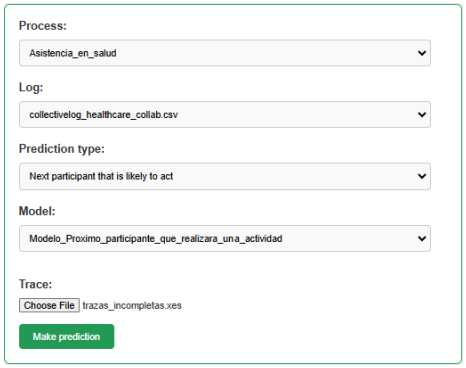


Figura 6.12: Generación de la predicción - Próximo participante que realizará una actividad

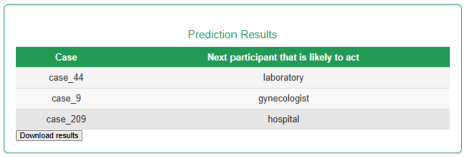


Figura 6.13: Resultado - Próximo participante que realizará una actividad

En estos casos la predicción coincide con los valores esperados.

Tiempo hasta el próximo mensaje a enviar

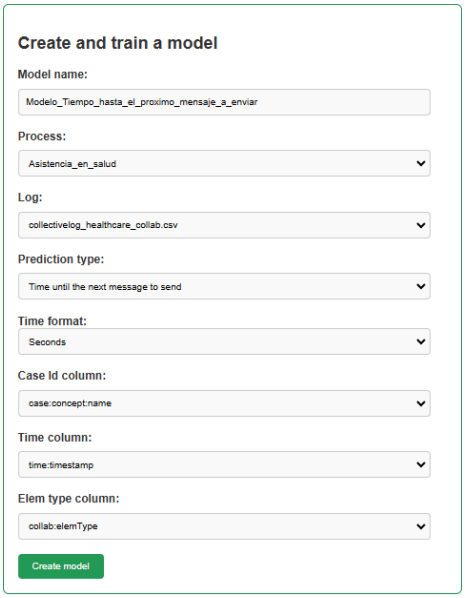


Figura 6.14: Creación de modelo - Tiempo hasta el próximo mensaje a enviar

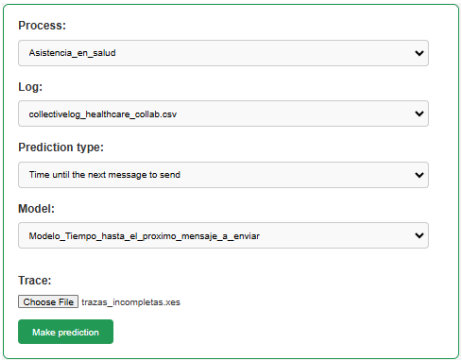
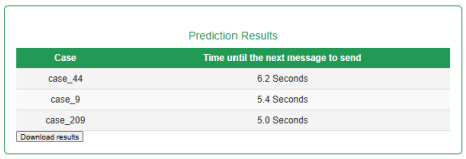


Figura 6.15: Generación de la predicción - Tiempo hasta el próximo mensaje a enviar



Resultado - Tiempo hasta el próximo mensaje a enviar

Tabla 6.3: Comparativa para predicción Tiempo hasta el próximo mensaje a enviar

Caso	**Valor esperado**	**Resultado**
Caso_44	5	6.2
Caso_9	3	5.4
Caso_209	5	5.0

Vemos que los resultados se encuentran en el orden de lo esperado ya que el MAE para este modelo es de 2.5 segundos.

Tiempo restante de un participante (Paciente)

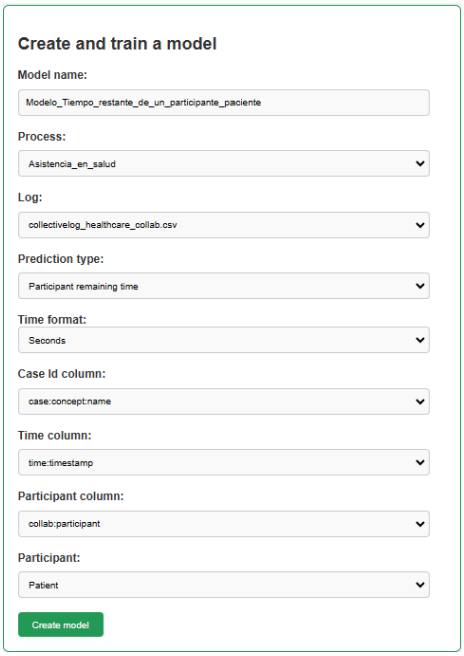


Figura 6.16: Creación de modelo - Tiempo restante de un participante (Paciente)

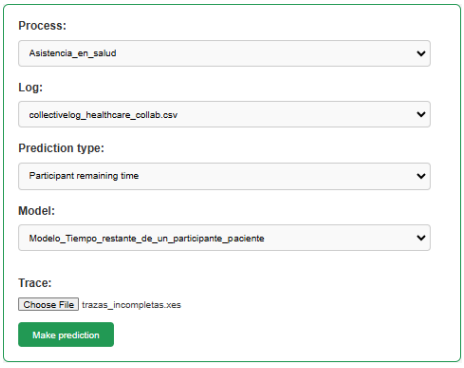


Figura 6.17: Generación de la predicción - Tiempo restante de un participante (Paciente)

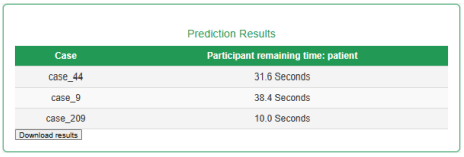


Figura 6.18: Resultado - Tiempo restante de un participante (Paciente)

Tabla 6.4: Comparativa para predicción Tiempo restante de un participante (Paciente)

Caso	**Tiempo actual**	**Tiempo restante Paciente (dato)**	**Tiempo restante Paciente (predicción)**
Caso_44	11	20	31.6
Caso_9	7	16	38.4
Caso_209	64	9	10

Como esta predicción depende del participante, agregamos un ejemplo para otro participante.

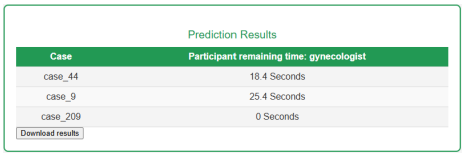


Figura 6.19: Resultado - Tiempo restante de un participante (Ginecólogo)

Tabla 6.5: Comparativa para predicción Tiempo restante de un participante (Ginecólogo)

Caso	**Tiempo actual**	**Tiempo restante Ginecólogo (dato)**	**Tiempo restante Ginecólogo (predicción)**
Caso_44	11	9	18.4
Caso_9	7	11	25.4
Caso_209	64	0	0

Entendemos que para las diferencias en los resultados de estas predicciones aplican las mismas observaciones realizadas para la predicción de Tiempo restante de proceso (6.1.1)

Próximo participante que enviará un mensaje

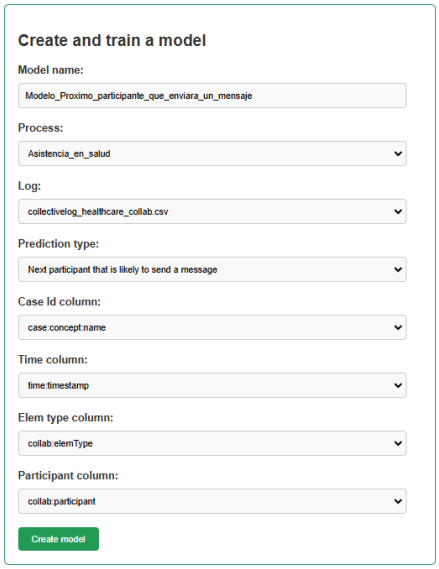


Figura 6.20: Creación de modelo - Próximo participante que enviará un mensaje

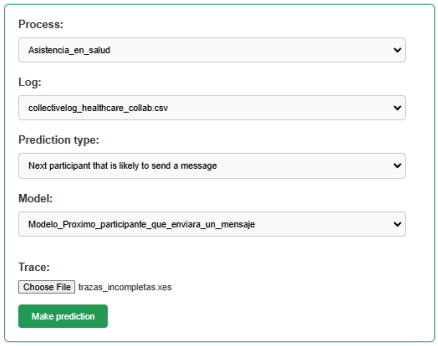


Figura 6.21: Generación de la predicción - Próximo participante que enviará un mensaje

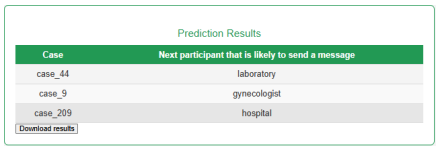


Figura 6.22: Resultado - Próximo participante que enviará un mensaje

En estos casos la predicción coincide con los valores esperados.

Próxima actividad que ocurrirá y qué participante la realizará

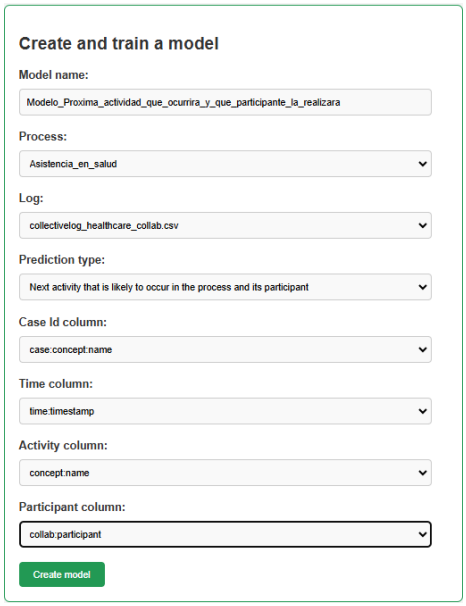


Figura 6.23: Creación de modelo - Próxima actividad que ocurrirá y qué participante la realizará

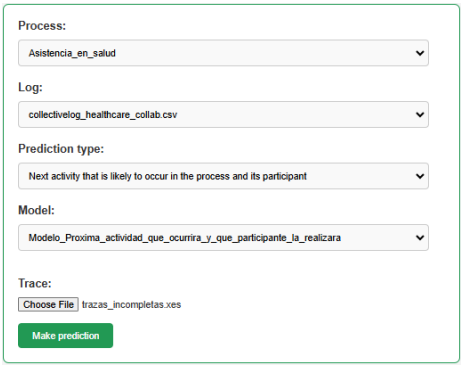


Figura 6.24: Generación de la predicción - Próxima actividad que ocurrirá y qué participante la realizará

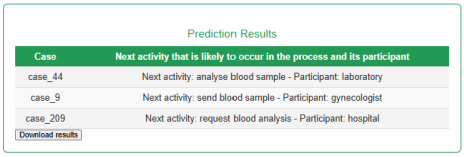


Figura 6.25: Resultado - Próxima actividad que ocurrirá y qué participante la realizará

En los casos 44 y 9 la predicción coincide con los valores esperados, mientras que para el 209 no. Si bien en este caso el resultado no es el esperado, vemos que tanto la actividad obtenida como la esperada son las que en su mayoría ocurren en el evento 16 de las trazas del log, ambas realizadas por el participante hospital.

Próximo participante que enviará un mensaje (con actividad)

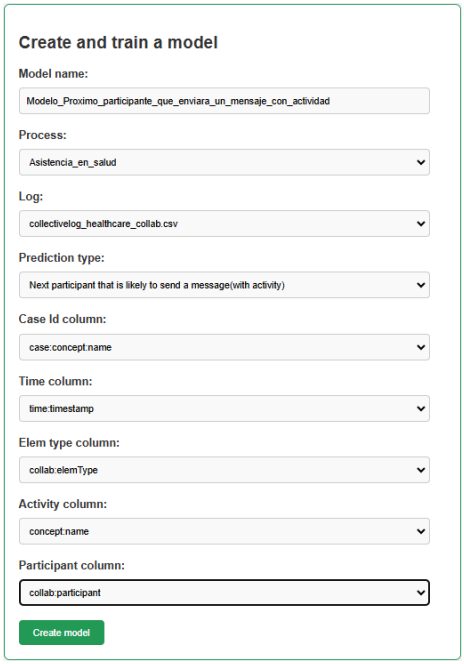


Figura 6.26: Creación de modelo - Próximo participante que enviará un mensaje (con actividad)

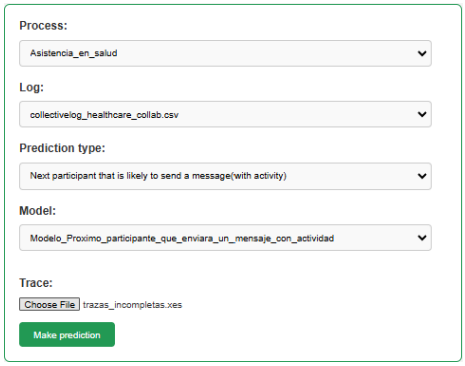


Figura 6.27: Generación de la predicción - Próximo participante que enviará un mensaje (con actividad)

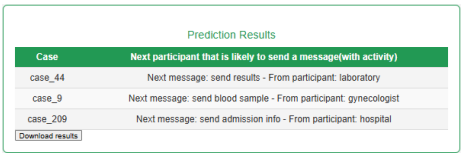


Figura 6.28: Resultado - Próximo participante que enviará un mensaje (con actividad)

En estos casos la predicción coincide con los valores esperados.

6.2. Evaluación comparativa de predicciones

Evaluamos la eficacia de las predicciones en cuatro logs de eventos extendidos a partir de los logs (Corradini y cols., 2022). En la Tabla 6.6 vemos las estadísticas descriptivas de los logs utilizados para las evaluaciones.
Log1: collectivelog_healthcare_collab.xes , Log2: collectivelog_artificial1_collab.xes, Log3: collectivelog_artificial4_collab.xes y Log4: collectivelog_real4_collab.xes

Tabla 6.6: Características de los logs

Logs	**Casos**	**Eventos**	**Variantes**	**Actividades**	**Largo máximo de casos**	**Largo promedio de casos**	**Duración máxima de caso**	**Duración promedio de caso**
Log1	97	1410	47	21	18	14.5	95s	49.4s
Log2	100	800	45	8	8	8	44.6s	29.4s
Log3	100	600	18	8	6	6	39.3s	19.5s
Log4	100	1800	77	19	18	18	99s	68.9s

A continuación presentamos los resultados experimentales de evaluar cada predicción para los logs mencionados:

Tabla 6.7: Resultados experimentales

Tipo de predicción		**Log1**	**Log2**	**Log3**	**Log4**
Próxima actividad	Accuracy	0.76	0.74	0.78	0.81
	F-score	0.70	0.64	0.69	0.75
Tiempo restante del proceso	MAE	9.44	4.10	3.20	6.20
Tiempo hasta próximo evento	MAE	1.67	1.76	1.53	1.93
Próximo participante que realizará una actividad	Accuracy	0.88	0.75	0.83	0.91
	F-score	0.83	0.65	0.77	0.90
Tiempo hasta el próximo mensaje a enviar	MAE	2.69	1.04	0.55	3.47
Tiempo restante de un participante	MAE	5.90 (Gyne-cologist)	4.10 (PartyA)	3.37 (PartyA)	5.85 (Zoo)
Próximo participante que enviará un mensaje	Accuracy	0.96	1.0	1.00	1.00
	F-score	0.95	1.0	1.0	1.0
Próxima actividad que ocurrirá y qué participante la realizará	Accuracy	0.81	0.71	0.78	0.74
	F-score	0.74	0.59	0.65	0.68
Próximo participante que enviará un mensaje (con actividad)	Accuracy	0.94	0.87	0.83	1.00
	F-score	0.92	0.81	0.77	1.00

Debido a la falta de métodos comparables que implementen este tipo de predicciones, no contamos con una referencia directa para contrastar nuestros resultados. No obstante, considerando las métricas utilizadas y los resultados observados en el caso de estudio, podemos afirmar que las predicciones muestran un rendimiento satisfactorio.

Es importante destacar que, al analizar las predicciones generadas por la extensión en relación con las predicciones originales del ProcessTransformer, encontramos que las métricas obtenidas son, en general, comparables o incluso mejores, lo que refuerza la efectividad de nuestro enfoque.

Capítulo 7

Conclusiones y Trabajo Futuro

El trabajo presentado ha logrado cumplir los objetivos planteados para el proyecto, avanzando en el estudio de la predicción de procesos colaborativos mediante la implementación de una herramienta que permite realizar predicciones relevantes en este contexto. La herramienta fue desarrollada a partir de la extensión de una solución preexistente para procesos no colaborativos.

En primer lugar se realizó el estudio y análisis de técnicas, algoritmos, herramientas y propuestas existentes para predicción de la ejecución de procesos de negocio con minería de procesos. Se aplicó una búsqueda en librerías, de dónde concluimos que si bien existían herramientas para predicción de procesos de negocio, ninguna de ellas tenían foco en los procesos colaborativos. Posteriormente a partir de estos resultados encontrados, se evaluaron las herramientas que fueron surgiendo de los documentos aunque no fueran para procesos colaborativos.

A partir de este primer análisis llegamos a una herramienta implementada para predicción de procesos de tipo no colaborativo que estaba presentada en (Bukhsh y cols., 2021) y tenía el código de la herramienta disponible (GitHub-Processtransformer, s.f.), dando lugar a su estudio y eventualmente su extensión/adaptación que nos permitió conseguir los siguientes objetivos: Generar propuesta/extensión de minería de procesos para la predicción de la ejecución de procesos de negocio colaborativos y Desarrollar/extender/adaptar herramienta prototipo de soporte a la propuesta. De esta extensión surge la herramienta que presentamos (predict-collab) donde logramos incorporar varios tipos de predicciones relevantes en el contexto de los procesos colaborativos.

La herramienta incluye el ciclo completo necesario para realizar cada una de las predicciones, desde la carga del log para entrenamiento, la creación de los modelos de predicciones y visualización de sus métricas, hasta la ejecución de las predicciones, visualización y descarga de los resultados obtenidos.

La herramienta fue evaluada mediante un caso de estudio obteniendo buenos resultados preliminares para todos los tipos de predicción implementados.

Además del caso de estudio que se enfoca en la visualización de los resultados de las predicciones, también se realizó una evaluación de las métricas de los modelos de predicción generados con distintos logs de eventos donde también se obtuvieron buenos resultados.

Al ser un primer enfoque a las predicciones de tipos colaborativo, queda abierta la posibilidad a varias mejoras en un trabajo a futuro.

En primer lugar de las predicciones implementadas, es directo agregar las siguientes predicciones:

- Próximo participante que recibirá un mensaje
- Próximo participante que recibirá un mensaje (con actividad)
- Tiempo hasta el próximo mensaje a recibir

Esto tiene que ver con el tipo de implementación realizada en la que estas predicciones serían análogas a las ya implementadas para el envío de mensaje pero cambiando el chequeo que se hace del campo *elemtype* de ser un *sendTask* a ser un *receiveTask*.

Entendemos importante que se sigan realizando pruebas sobre la aplicación con distintos tipos de logs, priorizando que en el caso de la generación de modelos de predicción, cuanto más rica sea la información en el entrenamiento, es esperable que mejoren las predicciones. Este punto viene de la mano con poder evaluar la herramienta con logs de aplicaciones de uso real que es posible cuenten con gran volumen de información para el entrenamiento.

Además a medida que pudieran surgir nuevas herramientas que aborden este tipo de predicciones se debería realizar un análisis comparativo de los distintos enfoques para ver que tan bien se comporta y si se puede mejorar de alguna manera este primer abordaje. La falta de este tipo de comparativos actualmente, limita las conclusiones que se pueden hacer sobre el desempeño de la herramienta.

El hecho de ser una implementación web, da lugar además a que se pudieran llegar a integrar otras funcionalidades que tengan que ver con la minería de procesos y tener en un mismo lugar acceso a herramientas que quizás hoy ya existen pero de manera independiente, un ejemplo de esto podría ser la integración de alguna herramienta de descubrimiento de procesos. Esto puede permitir tener una visión más general e integrada de todo el proceso de negocio.

En cuanto a la aplicación en sí, al haberse planteado como una aplicación web, entendemos conveniente que se introduzca el manejo de sesiones de usuario para que cada usuario pueda gestionar sus logs, modelos y predicciones de manera independiente. Esto viene de la mano de que generalmente el tipo de información que se puede ingresar en este tipo de sistemas sea de carácter confidencial y solo ciertos usuarios deban acceder a determinada información.

Referencias

Bukhsh, Z. A., Saeed, A., y Dijkman, R. M. (2021). Processtransformer: Predictive business process monitoring with transformer network. *ArXiv*, *abs/2104.00721*. Descargado de `https://api.semanticscholar.org/CorpusID:233004463`

Ceravolo, P., Comuzzi, M., De Weerdt, J., Di Francescomarino, C., y Maggi, F. (2024, 27 de Sep). Predictive process monitoring: concepts, challenges, and future research directions. *Process Science*, *1*(1), 2. Descargado de `https://doi.org/10.1007/s44311-024-00002-4` doi: 10.1007/s44311-024-00002-4

Corradini, F., Re, B., Rossi, L., y Tiezzi, F. (2022). A technique for collaboration discovery. En A. Augusto, A. Gill, D. Bork, S. Nurcan, I. Reinhartz-Berger, y R. Schmidt (Eds.), *Enterprise, business-process and information systems modeling* (pp. 63–78). Cham: Springer International Publishing.

Di Francescomarino, C., y Ghidini, C. (2022). Predictive process monitoring. En W. M. P. van der Aalst y J. Carmona (Eds.), *Process mining handbook* (pp. 320–346). Cham: Springer International Publishing. Descargado de `https://doi.org/10.1007/978-3-031-08848-3_10` doi: 10.1007/978-3-031-08848-3_10

Di Francescomarino, C., Ghidini, C., Maggi, F. M., y Milani, F. (2018). Predictive process monitoring methods: Which one suits me best? En M. Weske, M. Montali, I. Weber, y J. vom Brocke (Eds.), *Business process management* (pp. 462–479). Cham: Springer International Publishing.

Fluxicon disco. (s.f.). Descargado de `https://fluxicon.com/disco/`

GitHub-Processtransformer. (s.f.). *Zaharah/processtransformer: Transformer network for predictive business process monitoring tasks.* Descargado de `https://github.com/Zaharah/processtransformer`

González, L., y Delgado, A. (2021). Compliance requirements model for collaborative business process and evaluation with process mining. En *2021 xlvii latin american computing conference (clei)* (p. 1-10). doi: 10.1109/CLEI53233.2021.9640197

Keras. (s.f.). *Keras: The high-level api for tensorflow : Tensorflow core.* Descargado de `https://www.tensorflow.org/guide/keras`

Márquez-Chamorro, A. E., Resinas, M., y Ruiz-Cortés, A. (2018). Predictive monitoring of business processes: A survey. *IEEE Transactions on Services Computing*, *11*(6), 962-977. doi: 10.1109/TSC.2017.2772256

Tensorflow. (s.f.). *Tensorflow: An open source machine learning framework for everyone.* Descargado de `https://github.com/tensorflow/tensorflow`

van der Aalst, W. (2016). Data science in action. En *Process mining: Data science in action* (p. 3-23). Berlin, Heidelberg: Springer Berlin Heidelberg. Descargado de `https://doi.org/10.1007/978-3-662-49851-4_1` doi: 10.1007/978-3-662-49851-4_1

van der Aalst, W., Adriansyah, A., de Medeiros, A. K. A., Arcieri, F., Baier, T., Blickle, T., ... Wynn, M. (2012). Process mining manifesto. En F. Daniel, K. Barkaoui, y S. Dustdar (Eds.), *Business process management workshops* (pp. 169–194). Berlin, Heidelberg: Springer Berlin Heidelberg.

van der Aalst, W. M. P. (2011). Intra- and inter-organizational process mining: Discovering processes within and between organizations. En P. Johannesson, J. Krogstie, y A. L. Opdahl (Eds.), *The practice of enterprise modeling* (pp. 1–11). Berlin, Heidelberg: Springer Berlin Heidelberg.

Weske, M. (2019). *Business process management.* Springer Berlin, Heidelberg. Descargado de `https://doi.org/10.1007/978-3-662-59432-2`

Anexo A

Anexo

A.1. Aplicación Predict-Collab

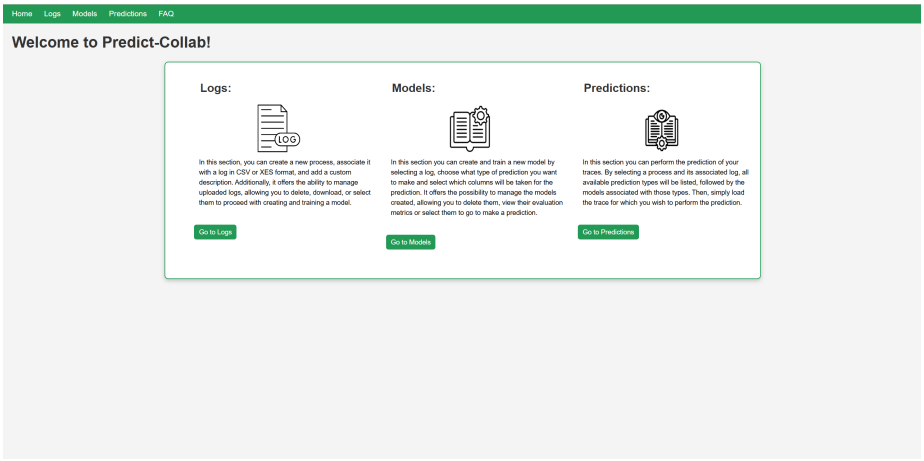


Figura A.1: Pantalla Home de la aplicación

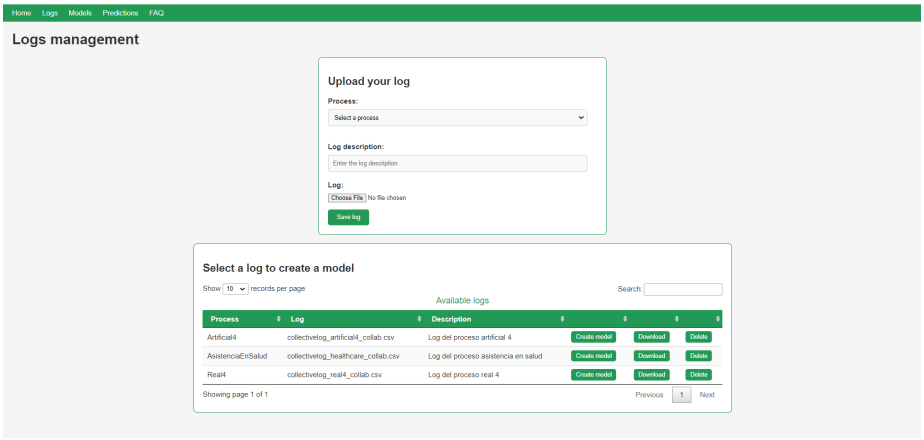


Figura A.2: Pantalla de Gestión de logs

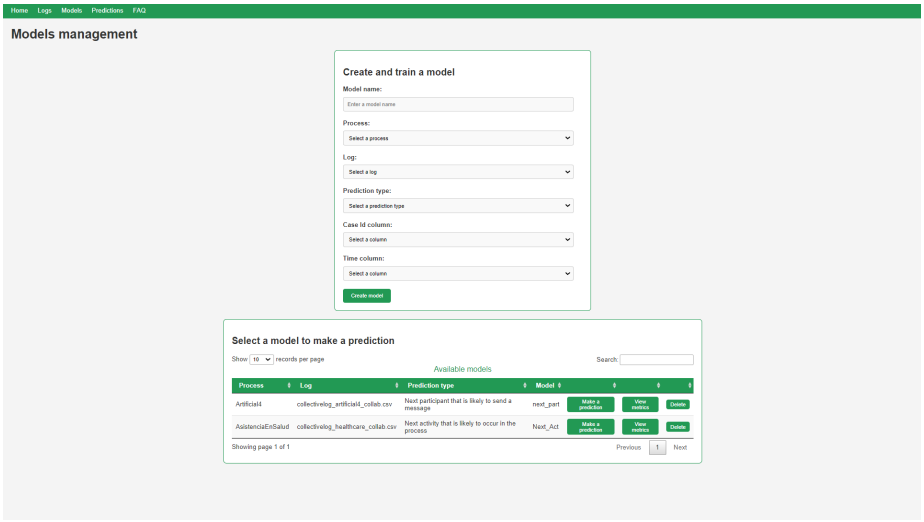


Figura A.3: Pantalla de Gestión de modelos

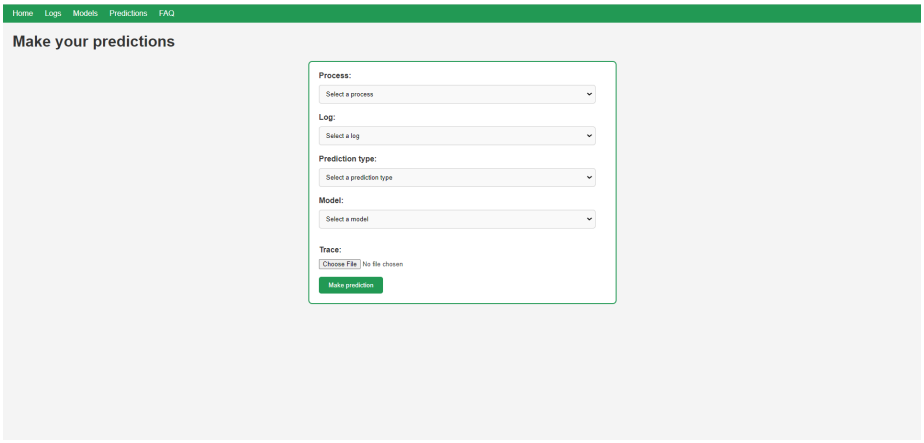


Figura A.4: Pantalla de Predicciones

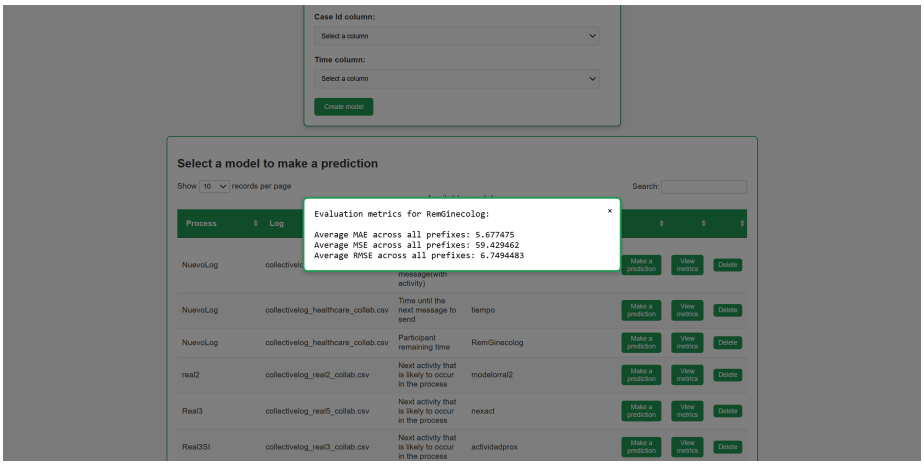


Figura A.5: Métricas de un modelo basado en próxima actividad

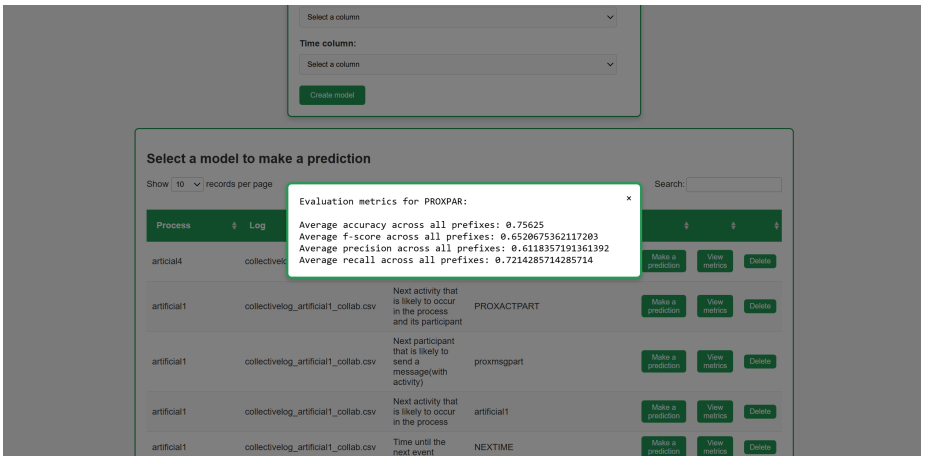


Figura A.6: Métricas de un modelo basado en tiempo

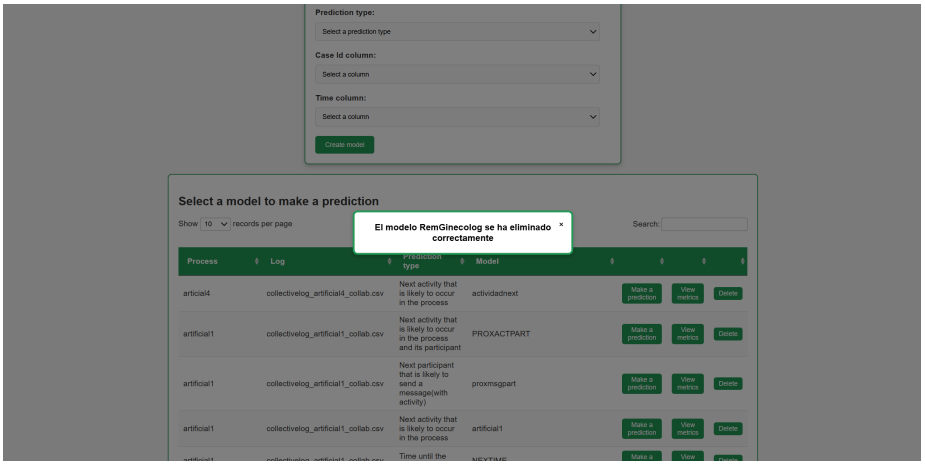


Figura A.7: Confirmación de eliminación de modelo

A.2. Datos del proceso

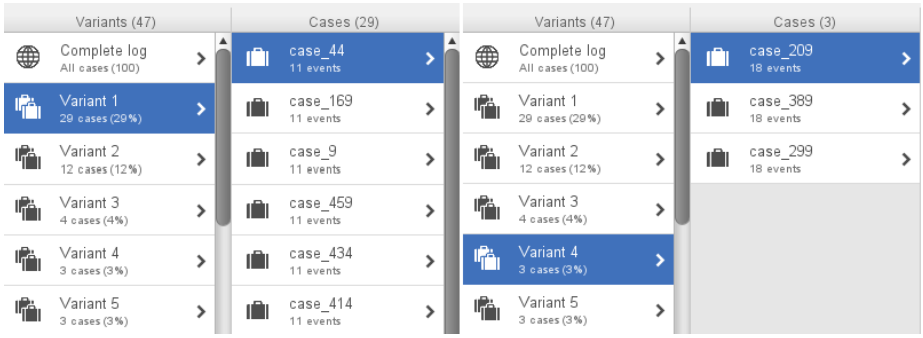


Figura A.8: Principales variantes de collectivelog_healthcare_collab.xes

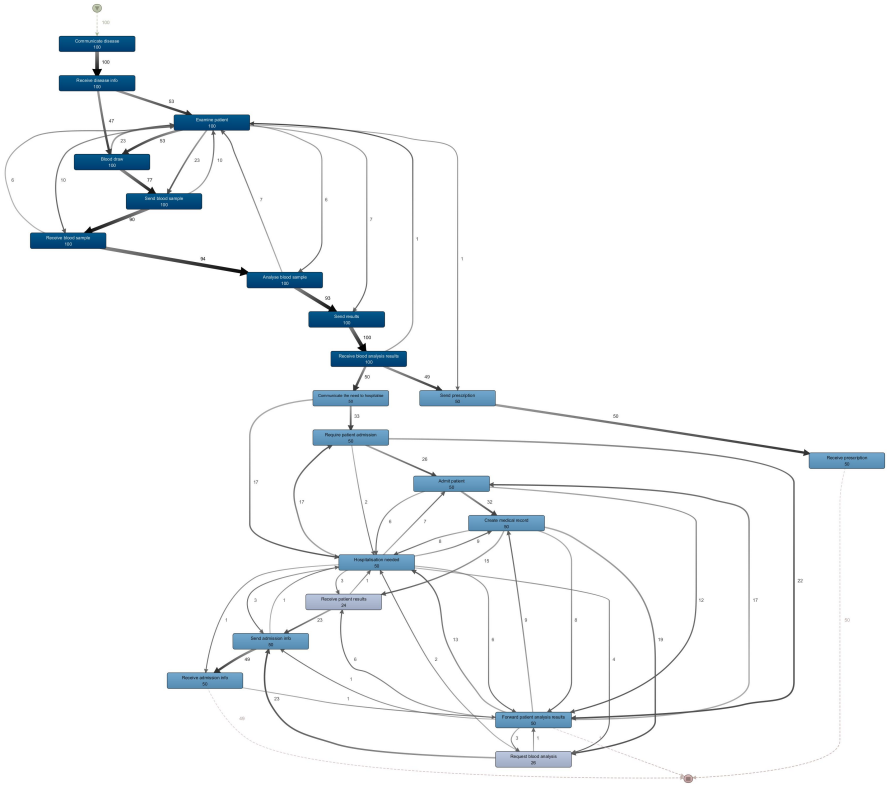


Figura A.9: Mapa del proceso Asistencia en salud obtenido de la herramienta Disco