

UNIVERSIDAD DE LA REPÚBLICA
FACULTAD DE INGENIERÍA



FACULTAD DE
INGENIERÍA



UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY

RECUPERACIÓN DE INFORMACIÓN Y RECOMENDACIONES EN LA WEB

INFORME DEL GRUPO 03 CORRESPONDIENTE A LA SEGUNDA ENTREGA DEL
LABORATORIO 2023

BRAIAN DE BARROS, IMANOL GONZALEZ Y GABRIEL BORGES

MONTEVIDEO, 25 DE NOVIEMBRE DE 2023

Índice general

1. Introducción	3
2. Problema	3
3. Enfoque de la solución	3
4. Arquitectura de la solución	3
5. Implementación	5
5.1. Frontend	6
5.2. Backend	6
5.3. Base de datos	7
5.4. Obtención e integración de datos	7
6. Funcionalidades y uso	7
6.1. Buscador con filtrado de previas	8
6.2. Información básica de la materia	8
6.3. Comentarios y puntaje	9
7. Evaluación y resultados	10
8. Conclusiones	10
9. Trabajo Futuro	11
Referencias	11

1. Introducción

En el siguiente proyecto, buscamos abordar la problemática que enfrentan los estudiantes a la hora de elegir qué materias cursar en un semestre de facultad. Durante este informe mostraremos como fue realizado el proyecto, así como las posibles mejoras propuestas que fueron identificadas como de valor, pero por el alcance del proyecto no pudieron ser implementadas.

La estructura del informe es la siguiente, en la sección 2 planteamos el problema que queremos resolver. A continuación se presenta en la sección 3 el enfoque de la solución para la problemática planteada. En las secciones 4 5.1 se plantea el diseño y la implementación de la solución. Luego se da una explicación de las principales funcionalidades en la sección 6. En la sección 7 se presentarán los principales resultados obtenidos en la ejecución de la solución. Para finalizar, se presentarán las conclusiones del trabajo elaborado en la sección 8 y una lista de puntos a seguir como trabajo a futuro en la sección 9

2. Problema

En la actualidad, los estudiantes se enfrentan a la toma de decisiones significativas a lo largo de su carrera académica. Estas decisiones incluyen la selección de materias optativas y/o la elección de horarios de cursos específicos ya que algunos ofrecen múltiples horarios, que suelen ser con profesores diferentes. Para llevar a cabo estas decisiones de máxima importancia, los estudiantes consultan la plataforma Eva de facultad (Udelar, 2023) para obtener información sobre qué profesores imparten cada materia. Una vez que tienen conocimiento de los nombres de los profesores de un curso en particular, los estudiantes tienen la opción de investigar las opiniones que tuvieron estudiantes pasados sobre esos profesores. Estas opiniones, que pueden ayudar a los alumnos a tomar las decisiones sobre qué materias cursar, entre las que se encuentran puntajes y comentarios, se pueden encontrar en sitios webs externos como lo es el sitio de MisProfesores (Anónimo, 2023). Además, es necesario darle herramientas a los usuarios para encontrar las materias de la carrera que están cursando y que requerimientos tienen que cumplir para poder cursarla.

3. Enfoque de la solución

Como solución se propone unificar esta tarea, creando una aplicación web que permita integrar estos conjuntos de datos: profesores, cursos, opiniones y calificaciones, de tal manera que los estudiantes puedan acceder a la información unificada para la toma de decisiones que podrán marcar la historia de su carrera. Para esto, se les brindara un buscador en el sitio web donde puedan consultar estos datos según el nombre de una materia. Además, se agregan mejoras a la búsqueda de los cursos que permita al usuario poder filtrar las materias para las cuales cumpla con las previas requeridas para poder cursar.

4. Arquitectura de la solución

Dado que no se tiene ningún control sobre los sitios de los que obtenemos la información, estos pueden cambiar repentinamente y desestabilizar nuestro sis-

tema. Por esto se plantea una arquitectura modular, como se puede observar en la figura 4.1, que permita el fácil intercambio de módulos. Sumado a esto, también se persisten los datos, para que nunca quede deshabilitado por problemas en los sitios externos.

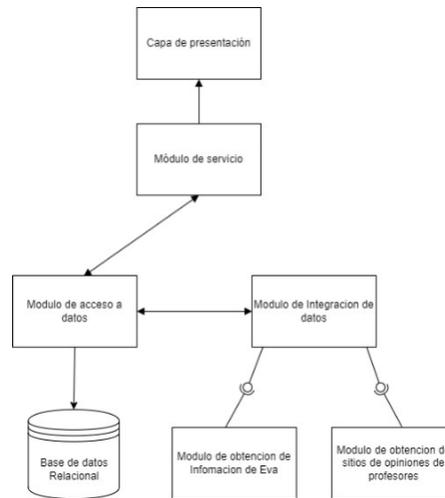


Figura 4.1: Arquitectura de la solución

La idea de la arquitectura es que el sistema utilice el módulo de integración de datos para nutrir periódicamente una base de datos relacional. Para esto utilizará módulos de obtención de datos, los cuales se encargaran de obtener la información de las diferentes fuentes y mapearlas a un esquema preestablecido que permita integrar dicha información. Esto con el objetivo de poder cambiar la fuente de los datos si así se desea sin la necesidad de afectar al módulo de integración. Además, la información almacenada en la base de datos, será consultada por el módulo de servicio, el cual la expone a la capa de presentación para poder ser accedida por los usuarios. A continuación, en la figura 4.2, se presenta un modelo de dominio con la información integrada de los diferentes sitios.

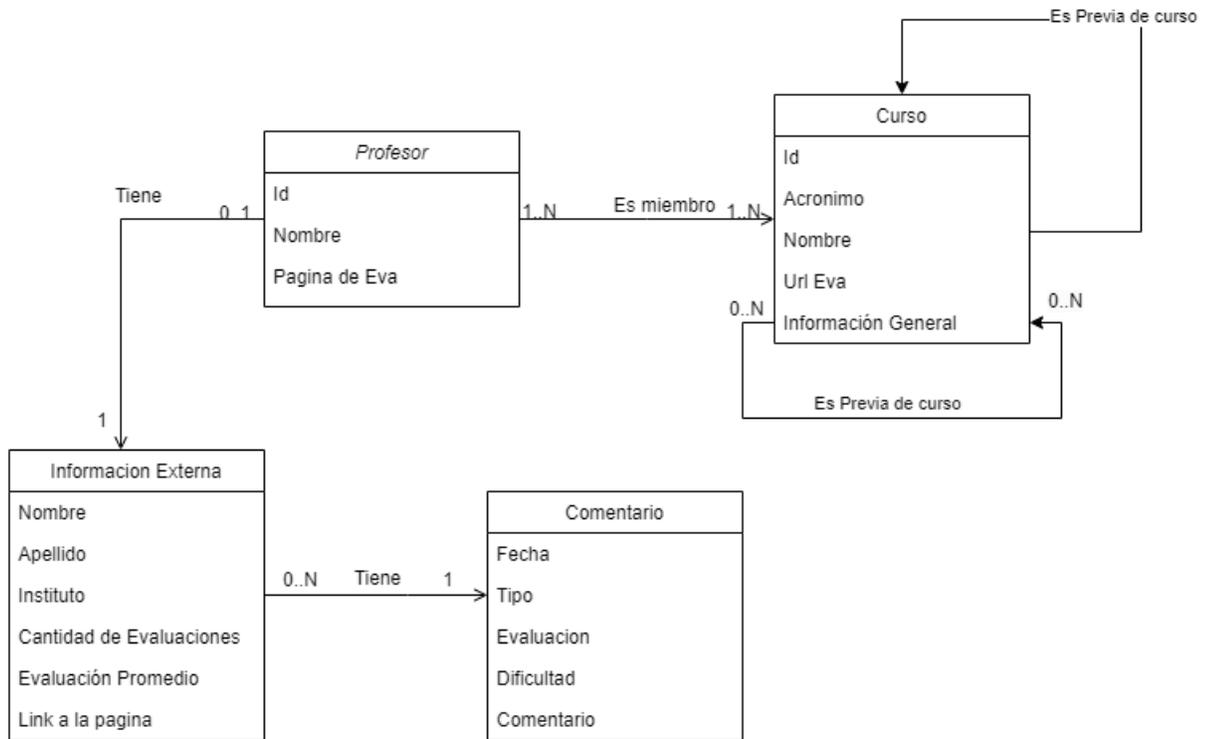


Figura 4.2: Integración de los diferentes datos

5. Implementación

Para la implementación utilizamos un conjunto de herramientas que se pueden observar en la figura 5.1. A continuación, en las siguientes secciones describiremos cada una de las partes del sistema y que herramienta utilizamos.

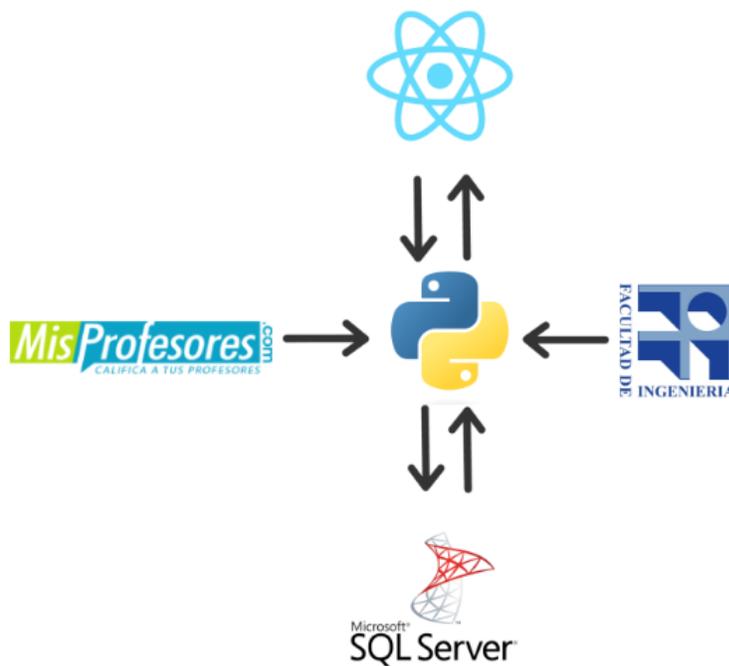


Figura 5.1: Herramientas

5.1. Frontend

En la construcción del Frontend, optamos por la biblioteca React mediante Vite para facilitar la integración a través de métodos HTTP con arquitectura REST, aprovechando las capacidades del lenguaje JavaScript.

5.2. Backend

Para la implementación de backend se utilizó el lenguaje python. Con él, se crearon los módulos de obtención e integración de datos que obtienen la información de los sitios externos, la procesan y la almacenan en una base de datos relacional. Además, se utilizó el framework de *Flask* para construir una API REST que le permita al frontend consultar los distintos datos.

5.3. Base de datos

La base de datos que se utilizó fue la de SQL Server. En particular, utilizamos el ORM *SQLAlchemy* para conectarnos desde python con la base de datos y así poder persistir la información.

5.4. Obtención e integración de datos

Es relevante realizar una sección aparte para la obtención e integración de los datos debido a que es el mayor problema que tuvimos que resolver y un punto clave para el funcionamiento del sistema.

La obtención de los datos se realizó utilizando la biblioteca *beautiful-soap*, con esta lográbamos obtener los HTML y obtener la información relevante, sumado a esto en ciertas ocasiones dado que se utilizaban llamadas Áyax dentro de las páginas fue necesario capturar dichas llamas, analizar que formato tenían y replicarlas en nuestro backend para poder obtener toda la información.

En particular se comenzó obteniendo la información del EVA, esto no es trivial, dado que EVA es nuestro sitio más confiable, únicamente queríamos obtener información de cursos y profesores que estuvieran allí y no llenar nuestro sistema con basura. El comenzar obteniendo la información de EVA nos permitió filtrar la de resto de sitios, obteniendo así una ampliación de la información existente ahí, pero no más. A su vez, esto ayuda mucho a la performance de la aplicación, el obtener las opiniones de los profesores implicaba múltiples peticiones al sitio, ya que algunos profesores tenían hasta 100 comentarios y estos estaban paginados, por lo que filtrar previamente qué profesores iban a ser necesarios reducía considerablemente el tiempo necesario para obtener toda la información.

En cuanto a la integración, en un principio se pensó tomar una estrategia de utilizar un modelo de aprendizaje automático para detectar cuando se hacía referencia al mismo profesor. Pero al momento de implementar, se detectó que el integrar únicamente por las coincidencias en los nombres de los profesores funcionaba realmente bien, pro lo que se decidió ir por esta estrategia mucho más simple.

6. Funcionalidades y uso

A continuación se explicarán las principales funcionalidades del sitio web que se implementó para resolver el problema de la sección 2.



Figura 6.1: Buscador con filtrado de previas

6.1. Buscador con filtrado de previas

Como se puede observar en la figura 6.2, tenemos el buscador de materias obtenidas del sitio eva de la facultad. Al buscador se le aplican los filtros de las materias de la parte superior, donde se cargan las diferentes materias aprobadas que tiene el estudiante. De esta forma solo se desplegarán las materias a las cuales el estudiante se puede anotar.

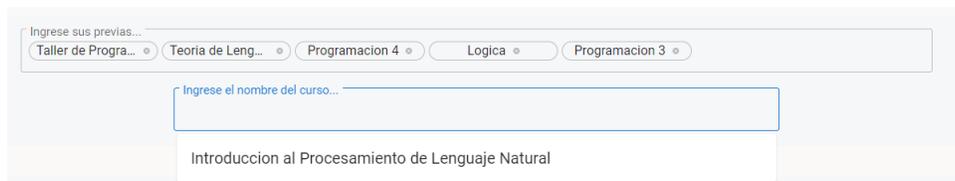


Figura 6.2: Buscador con filtrado de previas

6.2. Información básica de la materia

Observando la figura 6.3, tenemos alguna información básica de la materia, como sus objetivos y las previas de exámenes y cursos previos necesarios para cursar. Toda esta información obtenida del sitio eva de facultad.

<h2>Introducción al Procesamiento de Lenguaje Natural</h2>	<h2>Previas requeridas</h2>
<h3>Objetivos de la unidad curricular</h3>	<h3>Exámenes previos</h3>
<p>Presentar los principales problemas del área de PLN: traducción automática, resumen automático, reconocimiento del habla, recuperación de información, etc. Dar un panorama del estado de las aplicaciones existentes hoy día. Desarrollar los enfoques y técnicas comúnmente utilizados en el área: análisis morfológico, análisis sintáctico de superficie, análisis semántico, análisis pragmático y modelos del discurso, métodos de estado finito, aprendizaje automático, etc. Estudiar las limitaciones teóricas y materiales de los diferentes enfoques propuestos. Presentar algunas herramientas y recursos más utilizados: taggers, diccionarios, ontologías, gramáticas, corpus paralelos, etc. Utilizar estas herramientas en la resolución de un problema práctico.</p>	<p>Programacion 3</p>
	<p>Logica</p>
	<p>Programacion 4</p>
	<p>Teoria de Lenguajes</p>
	<h3>Cursos previos</h3>
	<p>Taller de Programacion</p>

Figura 6.3: Información básica de la materia

6.3. Comentarios y puntaje

A continuación, en la figura 6.4, se puede observar distintos comentarios y calificaciones que realizaron los estudiantes a un profesor de la materia. Con esta información, el estudiante podrá analizar si el profesor es bueno o no, y en función de esto saber si tiene que ir a sus clases.



Figura 6.4: Comentarios y puntaje

7. Evaluación y resultados

Luego de realizado el scrapping en los diferentes sitios, tanto en el sitio *Eva* como en el sitio web de *MisProfesores*, se obtuvieron 139 cursos con 207 profesores. De los cuales de 82 profesores se obtuvo información extra disponible en *MisProfesores*. Dicha información, incluía más de 1000 evaluaciones con comentarios.

8. Conclusiones

Se realizó un trabajo satisfactorio, donde se pudo incursionar en técnicas de scrapping que el equipo no conocía. El resultado final es una aplicación que ayuda a los estudiantes simplificando un poco su etapa estudiantil. Pese a que aún es necesario pulir algunos aspectos para un buen funcionamiento, entendemos que se tiene una muy buena base para la utilización de la aplicación, con información relevante para los estudiantes. A su vez, la arquitectura permite una



gran versatilidad a la hora de agregar nueva información o cambiar las fuentes. Esto permite, según las necesidades y sugerencias planteadas por los usuarios al momento de utilizarla, la aplicación puede adaptarse a ellas fácilmente. A su vez, durante el proceso de construcción se detectaron posibles mejoras, las cuales serán mencionadas en la sección.

9. Trabajo Futuro

Aunque estamos contentos con los resultados actuales, identificamos mejoras que podrían implementarse en el futuro para enriquecer aún más las funcionalidades y perfeccionar la experiencia del usuario.

- Obtención de datos de más foros públicos, pudiendo aplicar modelos de análisis de sentimiento.
- Obtener la información completa de las previas con un módulo que ejecute de manera automática, dado que la obtención mediante scrapping se nos hizo imposible para esto, como son dato bastante estáticos podrían ser actualizados por un administrador también.
- Ampliar a todas las materias de la facultad. Esto es muy sencillo, simplemente es ajustar para que se realice el scrapping para todos los institutos y no únicamente computación (no requiere desarrollos extras).
- Manejos de usuarios, permitiendo persistir las previas y enriquecer los datos, permitiendo agregar comentarios desde la página. A su vez, tener esta información nos serviría para implementar técnicas de clustering entre los usuarios, donde poder recomendar materias que le hayan gustado a usuarios que puntuaron de manera similar las materias anteriores.
- Mostrar los horarios de las clases, con el profesor que da la materia para elegir cuál es la de mayor conveniencia para el alumno.

Referencias

- Anonimo, H. (2023). *Misprofesores*. Descargado de <https://uruguay.misprofesores.com/escuelas/Facultad-de-Ingenieira-1146>
- Udelar. (2023). *Eva fing*. Descargado de <https://eva.fing.edu.uy/>