

REPORTE DEL DESARROLLO DE LA APLICACIÓN WEB

Desarrollado por Acosta Romero Martin Gustavo y Caballero Pérez Alberto.

Universidad Veracruzana, Región Boca del Rio, Veracruz.

13 de junio de 2024.

Proyecto final de la experiencia educativa Diseño de Aplicaciones Web.

Aplicación web desarrollada para el módulo de seguimiento y gestión de tutorías.

TABLA DE CONTENIDO

Tabla de ilustraciones.....	5
1 Introducción	7
1.1 Proposito	7
1.2 Alcance	7
1.3 Definiciones y acrónimos	7
2 Descripción General.....	9
2.1 Perspectiva del producto	9
2.2 Requerimientos	9
2.2.1 Requerimientos generales	9
2.2.2 Requerimientos específicos.....	10
2.3 Funciones Principales	11
2.3.1 Funciones para tutorados.....	11
2.3.2 Funciones para tutores.....	11
2.4 Aspectos de Funcionamiento	12
2.5 Restricciones	12
2.5.1 Dependencias	12
2.5.2 Funcionalidad de los Servlets.....	12
2.5.3 Configuración del Servidor:	13
2.6 Suposiciones y Dependencias	13
3 Planificación	15
4 Análisis y diseño del sistema	16
4.1 Metodología utilizada	16

	3
4.2 Diagramas UML	16
4.2.1 Diagrama general del sistema	17
4.2.2 Diagrama de clases.....	18
4.2.3 Diagrama de casos de uso	19
5 Diseño de la base de datos	22
6 Interfaz Gráfica del Usuario (GUI).....	25
6.1 Inicio de sesión	25
6.2 Registro	25
6.3 Vistas del tutor	27
6.3.1 Menú principal	27
6.3.2 Información personal de tutor	27
6.3.3 Mensajería para tutores	28
6.3.4 Sección para ver tutorados	30
6.3.5 Notas	32
6.3.6 Tutorías.....	33
6.3.7 Estadísticas	34
6.4 Vistas del tutorado.....	35
6.4.1 Menú principal	35
6.4.2 Información personal del tutorado	36
6.4.3 Mensajería de alumnos.....	36
7 Conclusiones	38
8. Referencias.....	39



		4
9	Anexos	40
9.1	DevOps	40

Tabla de ilustraciones

Ilustración 1. Diagrama General del Sistema	17
Ilustración 2. Diagrama de Clases	19
Ilustración 3. Actores	20
Ilustración 4. Casos de uso.....	20
Ilustración 5. Casos de uso del tutorado	21
Ilustración 6. Casos de uso del tutor	21
Ilustración 7. Interfaz para iniciar de sesión	25
Ilustración 8. Interfaz para registro	25
Ilustración 9. Interfaz para registro de tutorado	26
Ilustración 10. Interfaz para registro de tutor.....	26
Ilustración 11. Menú principal de tutores	27
Ilustración 12. Vista con información de tutores	27
Ilustración 13. Bandeja de entrada de tutores	28
Ilustración 14. Detalle de mensajes de tutores	28
Ilustración 15. Vista para redactar mensajes de tutores	29
Ilustración 16. Ejemplo para seleccionar remitente de mensajes	29
Ilustración 17. Listado de alumnos de tutores	30
Ilustración 18. Detalle de alumnos en interfaz de tutores	30
Ilustración 19. Ejemplo uno de reporte de estudiantes	31
Ilustración 20. Ejemplo dos de reporte de estudiantes.....	31
Ilustración 21. Vista de notas	32
Ilustración 22. Vista al seleccionar un alumno en notas	32
Ilustración 23. Registrar nueva nota	33
Ilustración 24. Detalle de tutorías	33

Ilustración 25. Registrar nueva tutoría.....	34
Ilustración 26. Estadísticas de la aplicación	34
Ilustración 27. Más detalle de estadísticas de la aplicación.....	35
Ilustración 28. Menú principal del tutorado.....	35
Ilustración 29. Información personal del tutorado	36
Ilustración 30. Bandeja de entrada del tutorado.....	36
Ilustración 31. Vista para redactar mensajes de tutorados	37

1 Introducción

El reporte y la aplicación presentados a continuación han sido desarrollados por Alberto Caballero Pérez y Martín Gustavo Acosta Romero como parte del proyecto final de la experiencia educativa Diseño de Aplicaciones Web.

1.1 Propósito

Este reporte tiene como propósito dar detalle de la forma en que fue desarrollada una aplicación web cuyo objetivo es administrar el módulo de seguimiento y gestión de tutorías, de la Universidad Veracruzana. En este reporte también se mencionarán los requerimientos generales y específicos que se nos asignaron para desarrollarla de forma eficaz, se mostrarán las funcionalidades y se analizará el diseño, las interfaces y los diferentes diagramas utilizados en el proceso de desarrollo.

1.2 Alcance

Como ya se mencionó, la aplicación se trata de un proyecto escolar que ayudará a administrar las tutorías de la universidad, por ende, su alcance está limitada a esta. No obstante, es una aplicación web que se podría adaptar para cubrir a todos los alumnos y tutores de todas las carreras y facultades de la Universidad Veracruzana.

Esto gracias a que la aplicación está automatizada para funcionar con la base de datos proporcionada, entonces lo único que limita su alcance es la base de datos con la que se cuenta.

1.3 Definiciones y acrónimos

Aplicación Web: Es un programa o servicio accesible a través de un navegador de internet, diseñado para realizar una o varias tareas específicas por medio de una interfaz web. Muchas personas suelen confundir una aplicación web con una página web, pero la diferencia es que las páginas web son estáticas y se utilizan más que nada para dar información, mientras que las aplicaciones web son dinámicas y permiten realizar tareas e interactuar.

Módulo de seguimiento y gestión de tutorías: Es una parte de el sistema tutorías que se encarga de monitorear y organizar la asignación de tutores, la programación de sesiones y el seguimiento del progreso de los estudiantes, entre otras acciones.

Requerimientos generales: Son las necesidades o condiciones esenciales que debe cumplir un sistema para ser operativo y eficaz, pero que no están vinculadas a características específicas del mismo.

Requerimientos específicos: Son las necesidades o condiciones detalladas que debe satisfacer un sistema o producto específico, relacionadas con funcionalidades o características particulares que deben ser implementadas.

Interfaz: Es el medio por el cual los usuarios interactúan con una máquina, un software o un dispositivo, y puede incluir elementos visuales, auditivos y táctiles.

Automatizado: Se refiere a un proceso o sistema que opera o se controla de manera automática, generalmente con poca o ninguna intervención humana.

Base de datos: Es una colección organizada de datos, generalmente almacenados y accesibles electrónicamente desde un sistema informático, un Sistema Gestor de Base de Datos, donde se pueden realizar operaciones de búsqueda, inserción, actualización y gestión de la información.

2 Descripción General

Para el desarrollo de esta aplicación, utilizamos el lenguaje de programación JAVA, con el entorno de desarrollo NetBeans. En él, desarrollamos las interfaces utilizando la tecnología JSP, la cual se emplea para crear aplicaciones web basadas en HTML y XML, entre otros tipos de documentos.

También utilizamos el sistema gestor de bases de datos PostgreSQL para desarrollar la base de datos necesaria.

2.1 Perspectiva del producto

La aplicación desarrollada pretende administrar el sistema de tutorías de la Universidad Veracruzana. La aplicación está enfocada en el módulo de seguimiento y gestión de tutorías, en el cual los tutores podrán dar seguimiento a la actividad de sus tutorados durante su estancia universitaria; asimismo, los alumnos podrán ingresar y consultar información sobre su trayectoria.

2.2 Requerimientos

2.2.1 Requerimientos generales

Los requerimientos generales son aquellos que se deberían implementar en cualquier aplicación web para que sea lo más eficiente posible, estos fueron proporcionados por la docente de esta experiencia educativa. A continuación, se presenta una lista no muy larga, pero que cubre los aspectos cruciales para el desarrollo.

- Velocidad y capacidad de respuesta: Se deben establecer objetivos de tiempo de respuesta y procesamiento.
- Escalabilidad: Se debe planificar la capacidad de la aplicación para crecer y manejar cargas de trabajo aumentadas.
- Medidas de protección: Implementar protocolos de seguridad para salvaguardar la aplicación.

- Gestión de riesgos: Identificar amenazas potenciales y desarrollar planes de mitigación.
- Interfaz de usuario: Diseñar una interfaz intuitiva y fácil de navegar.
- Accesibilidad: Asegurar que la aplicación sea accesible para todos los usuarios, incluidos aquellos con discapacidades.
- Modularidad: Fomentar una estructura que permita actualizaciones y mejoras eficientes.
- Compatibilidad: Garantizar que la aplicación funcione en diversas plataformas y dispositivos.
- Dependencias: Minimizar las dependencias externas para facilitar la portabilidad.

2.2.2 Requerimientos específicos.

Hablando de los requerimientos específicos, hacemos referencia a aquellos que fueron ideados específicamente para nuestra aplicación. Estos, por su parte, fueron proporcionados por nuestros stakeholders y tienen como objetivo cumplir con sus expectativas, sin dejar de lado la eficacia de la aplicación web. Los requerimientos específicos son los siguientes:

- Se debe poder revisar el expediente académico del tutorado.
- Se debe llevar un registro de notas, comentarios y observaciones referentes al progreso de los estudiantes.
- Debe verse el historial de tutorías realizadas, incluyendo detalles como la fecha, duración y temas tratados.
- Debe ser posible generar informes sobre el rendimiento individual de los tutorados, representados con estadísticas en comparación con lo ideal y/o lo real.

- Se deben realizar análisis y estadísticas sobre el uso de la plataforma, como: número de tutorías programadas, asistencia, participación, etc.
- Se deben generar también estadísticas de desempeño académico de los tutorados.
- Los tutores deben poder buscar alumnos específicos.
- Mensajería entre tutores y tutorados para facilitar la comunicación y coordinación, parecido a Eminus.

2.3 Funciones Principales

Al iniciar la aplicación, el usuario tiene la opción de registrarse, si es que no lo ha hecho antes; el usuario puede registrarse dependiendo si es tutor o tutorado, solo tiene que llenar sus datos y posteriormente iniciar sesión con el correo y contraseña que registró. A continuación se muestran las funciones que conforman la aplicación, según el tipo de usuario que se encuentre conectado, no sin antes mencionar que ambos tienen un apartado dónde podrán ver estadísticas sobre el uso de la plataforma.

2.3.1 Funciones para tutorados

Al iniciar sesión como tutorado, el usuario tiene las opciones de visualizar su avance académico, sus datos personales, ver su lista de mensajes y escribir uno nuevo para su tutor.

2.3.2 Funciones para tutores

Si es un tutor quién ha iniciado sesión, se le muestran aún más apartados, ya que es quién tiene más obligaciones en el sistema de tutorías. El tutor puede ver el avance académico y los datos personales de todos los tutorados que se le han asignado, puede enviarles mensajes, ver las tutorías realizadas y registrar nuevas, además, tiene un apartado en el cuál puede llevar un registro de notas para hacer sus observaciones del progreso de cada uno de sus tutorados.

2.4 Aspectos de Funcionamiento

Todas las funciones mencionadas anteriormente, contribuyen al proceso general de la aplicación, siendo cada una de estas una parte fundamental para su óptimo funcionamiento y siendo necesarias para la satisfacción del usuario. La eficiencia operativa de nuestra aplicación se sustenta en la interconexión fluida de sus funciones. Cada característica ha sido diseñada para ser una pieza esencial dentro de un engranaje que promueve la satisfacción del usuario final. Por ejemplo, la función de registro de tutorías se vincula directamente con nuestra base de datos PostgreSQL, lo que permite una sincronización inmediata y precisa de los datos académicos.

Para garantizar un funcionamiento óptimo, nuestra aplicación requiere de una conexión a internet estable. Durante la fase de desarrollo, utilizamos el entorno de desarrollo NetBeans, que simula el entorno de producción y ofrece herramientas robustas para la depuración y pruebas de nuestras interfaces JSP.

Esta aplicación también puede recibir actualizaciones con el paso del tiempo si es que son necesarias, esto dependerá de la satisfacción del cliente, quienes la probarán y darán su opinión respecto a la aplicación, no solo en la Interfaz si no también en el funcionamiento.

2.5 Restricciones

2.5.1 Dependencias

- Utiliza PostgreSQL, por lo que debe haber una base de datos PostgreSQL configurada y accesible.
- Depende de bibliotecas específicas como JFreeChart para la generación de gráficos y Gson para el procesamiento de JSON.

2.5.2 Funcionalidad de los Servlets

- Los servlets como AlumnoDetallesServlet manejan solicitudes GET y requieren parámetros específicos.

- Las respuestas se devuelven en formato JSON y deben manejarse adecuadamente en el frontend.

2.5.3 Configuración del Servidor:

- La aplicación está empaquetada como un WAR, por lo que necesita un servidor de aplicaciones compatible con Java EE.

2.6 Suposiciones y Dependencias

Para funcionar correctamente esta aplicación depende de componentes de terceros; a continuación, se mostrará una lista de las dependencias que utilizamos durante el desarrollo del proyecto.

- **PostgreSQL:** Utilizada para la conexión y gestión de la base de datos PostgreSQL.
- **Jfreechart:** Es una biblioteca de código abierto para la creación de gráficos y diagramas en Java.
- **Jcommon:** Es una biblioteca de soporte que proporciona utilidades y clases auxiliares que son utilizadas por JFreeChart.
- **Itexpdf:** Es una biblioteca para la creación y manipulación de documentos PDF en Java.
- **Gson:** Biblioteca para la serialización y deserialización de objetos Java a JSON y viceversa.
- **JUnit:** Framework de pruebas unitarias para Java.
- **javax.servlet-api:** API de servlet para Java.
- **javaee-web-api:** API de Java EE para aplicaciones web.
- **commons-dbcp2:** Biblioteca de Apache Commons para el pooling de conexiones de base de datos.
- **tomcat7-maven-plugin:** Plugin para integrar Apache Tomcat con Maven y facilitar el despliegue.



- **maven-compiler-plugin:** Plugin para configurar la compilación del código fuente de Java.
- **maven-war-plugin:** Plugin para construir archivos WAR (Web Application Archive).

3 Planificación

Para la selección del proyecto, nos aseguramos de escoger aquel que entendimos podríamos desarrollar de manera más eficiente, con el fin de entregar un trabajo que cumpla con las expectativas de los stakeholders asignados. Escogimos el Módulo de Seguimiento y Gestión de Tutorías porque es un tema con el que estamos muy familiarizados, ya que son temas que regularmente vemos con nuestros tutores.

Para llevar a cabo esta aplicación, realizamos una reunión con uno de los stakeholders asignados. En esta reunión se discutieron diversos puntos de vista e ideas para el desarrollo, y se elaboró una lista de requerimientos en la cual nos basamos para un desarrollo óptimo.

El proyecto comenzó con el análisis de la funcionalidad del sistema de tutorías. Posteriormente, diseñamos la estructura lógica y relacional de la base de datos para el registro de la información de la aplicación web propuesta.

Finalmente, diseñamos las interfaces de la aplicación web y les dimos el funcionamiento necesario para cada acción que realizarán los usuarios.

4 Análisis y diseño del sistema

4.1 Metodología utilizada

Para nuestro proyecto de desarrollo del sistema de gestión de tutorías, hemos trabajado con la metodología DevOps para optimizar el proceso de desarrollo y despliegue de software. DevOps nos ha permitido automatizar la construcción, prueba y despliegue de nuestras aplicaciones de manera eficiente, garantizando al mismo tiempo la estabilidad del sistema y la rápida entrega de nuevas funcionalidades.

Trabajamos de manera colaborativa con la herramienta GitHub Desktop, la cuál nos permitió manejar de manera visual y efectiva nuestros repositorios locales y remotos en GitHub. Facilita la realización de commits y la sincronización con el repositorio remoto, lo cual fue crucial para nuestro flujo de trabajo ágil y colaborativo.

Al integrar GitHub Desktop en nuestro proceso de desarrollo, se facilita la visibilidad y el orden de todos los cambios realizados en el código, así como la colaboración entre ambos colaboradores. Esto ha sido fundamental para mantener un flujo de trabajo ágil que se alinea con la metodología DevOps.

4.2 Diagramas UML

Los diagramas UML (Unified Modeling Language o Lenguaje Unificado de Modelado en español) dispone de multitud de diagramas que ayudan a comprender la complejidad del sistema en cuestión, permitiéndonos plasmar aquellas funcionalidades, requisitos y demás características que hemos detectado en el sistema.

Aunque UML define una gran cantidad de diagramas para representar los distintos aspectos del desarrollo de la aplicación, en este reporte nos hemos centrado solamente en el diagrama general del sistema, el diagrama de clases y el diagrama de casos de usos, cada uno de ellos se presentarán a continuación.

4.2.1 Diagrama general del sistema

El diagrama general del sistema proporciona una visión clara y concisa de cómo están estructurados los diferentes componentes y cómo interactúan entre sí para lograr los objetivos de la aplicación web; esto ayuda a los stakeholders a entender la arquitectura. A continuación, se detallan los componentes clave:

1. **Base de datos:** Esta es la base de toda la arquitectura pues es de donde se recopila toda la información que se mostrará posteriormente en la aplicación.
2. **Capa de acceso a datos:** Implementada con Java Beans y Java DAO's para la conexión y manipulación de datos en PostgreSQL, proporcionando la integración necesaria entre la base de datos y la lógica de la aplicación, por supuesto, también una clase Java para crear la conexión entre Java y PostgreSQL, que son el entorno de desarrollo integrado y el sistema gestor de base de dato que utilizamos.
3. **Interfaz de usuario:** Creamos interfáces con ayuda de la tecnología JSP (JavaServer Pages), la cual se emplea para crear aplicaciones web basadas en HTML y XML, entre otros tipos de documentos.
4. **Controlador del flujo de aplicación:** Hicimos uso de las clases Servlets, programas escritos en Java que admiten peticiones a través del protocolo HTTP. Estos, reciben peticiones desde un navegador web, las procesan y devuelven una respuesta al navegador, normalmente en HTML. Estos programas son los intermediarios entre el cliente (casi siempre navegador web) y los datos.



Ilustración 1. Diagrama General del Sistema

4.2.2 Diagrama de clases

El diagrama de clases describe la estructura del sistema mostrando las relaciones (asociación, agregación y composición) entre los distintos elementos que conforman la aplicación, expresados estos como clases con sus atributos. A continuación describiremos brevemente dichas clases:

- **Mensajes:** En esta clase se almacenan todos los mensajes que se han enviado, contiene el asunto, la fecha, el ID del tutor y el ID del alumno.
- **Materias:** Aquí se encuentran todas las materias que existen dentro del mapa curricular, contienen datos como el nombre de la materia, los créditos y el área a la que está asociada.
- **Materias_alumnos:** Esta clase conecta la clase materias con la clase alumnos para saber las materias que ha llevado el alumno, en ella se puede ver el tipo de inscripción de la materia y la calificación obtenida.
- **Tutorias:** Aquí se almacenan las tutorías que se han registrado, se puede ver la fecha, el contexto de la tutoría y la duración, además de el ID del tutor y tutorado que corresponde.
- **Notas:** En esta clase se almacenan todas las notas que vaya registrando el tutor.
- **Alumnos:** Como su nombre lo indica, en esta clase se encuentran todos los datos de todos los alumnos registrados.
- **Tutores:** Al igual que la clase de alumnos, en esta se encuentran registrados todos los datos de todos los tutores registrados.
- **Estatus:** Esta clase sirve para indicar el estado de el atributo a quien está referenciando.



- **Áreas:** Representa las diferentes áreas en que están divididas las materias, estas son:
Area de Formacion Basica, Iniciacion a la Disciplina, Area de Formacion Terminal y
Area de Formacion Eleccion Libre.

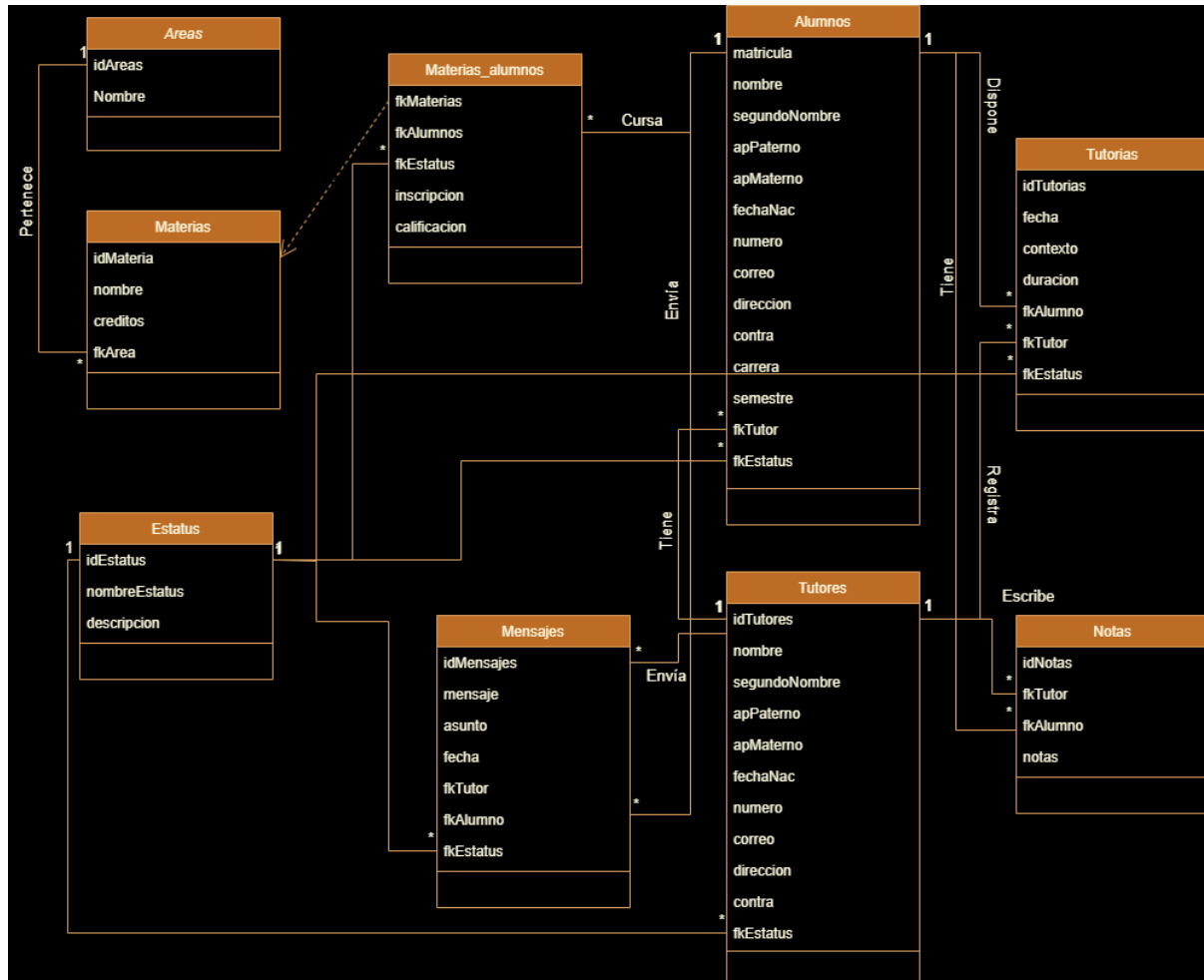


Ilustración 2. Diagrama de Clases

4.2.3 Diagrama de casos de uso

Los diagramas de casos de uso nos permiten diferenciar los actores que interactúan con nuestra aplicación, las relaciones entre ellos y las acciones que cada uno puede realizar dentro del sistema. Estos diagramas son fácilmente comprensibles tanto por clientes como por usuarios, representan los requisitos funcionales del sistema. Los diagramas de casos de uso tienen tres elementos:

- **Actores:** Son los usuarios del sistema. Un actor puede ser una persona, un conjunto de personas, un sistema hardware o un sistema software. Los actores representan un rol, que puede desempeñar alguien que necesita intercambiar información con el sistema.
- **Casos de uso:** Un caso de uso describe una forma concreta de utilizar parte de la funcionalidad de un sistema. La colección de todos los casos de uso describe toda la funcionalidad del sistema.
- **Comunicación entre actores y casos de uso:** Cada actor ejecuta un número específico de casos de uso en la aplicación. Por eso decimos que hay comunicación entre actores y casos de uso.

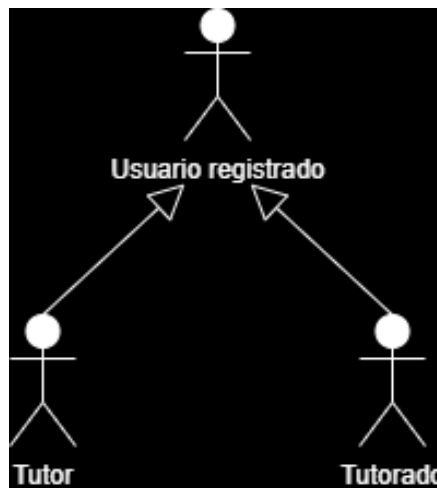


Ilustración 3. Actores

Los usuarios registrados son los tutores y tutorados de la universidad, que acceden a la aplicación mediante un correo y contraseña.

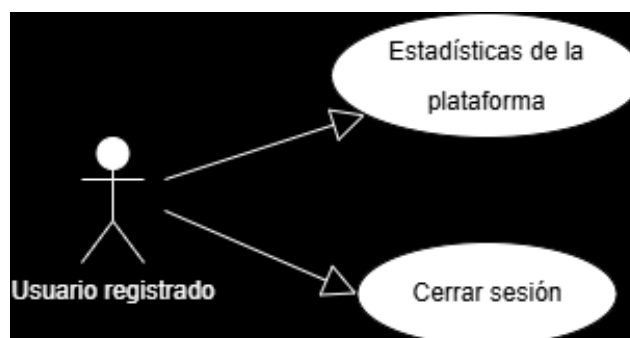


Ilustración 4. Casos de uso

Este tipo de usuario sólo puede realizar la acción de ver las estadísticas de la plataforma y cerrar su sesión porque es la única funcionalidad que tienen en común los usuarios tutor y tutorado.

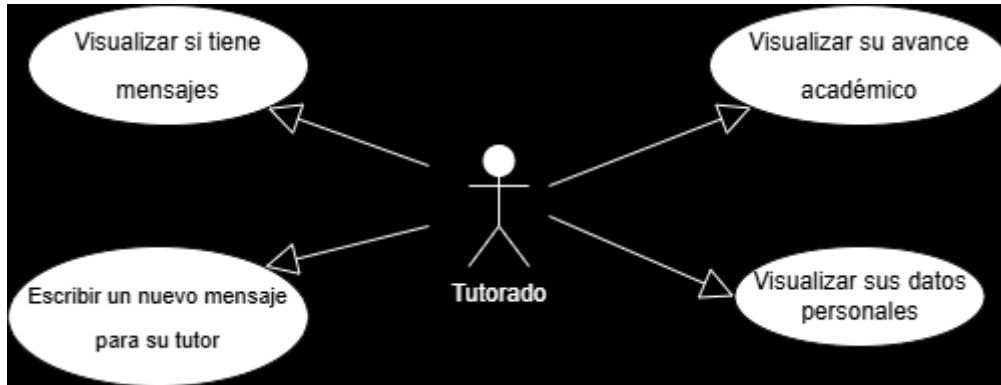


Ilustración 5. Casos de uso del tutorado

Este usuario representa a quien ha ingresado como tutorado, las acciones que puede realizar son: visualizar si tiene mensajes, escribir un nuevo mensaje para su tutor, visualizar su avance académico y visualizar sus datos personales.

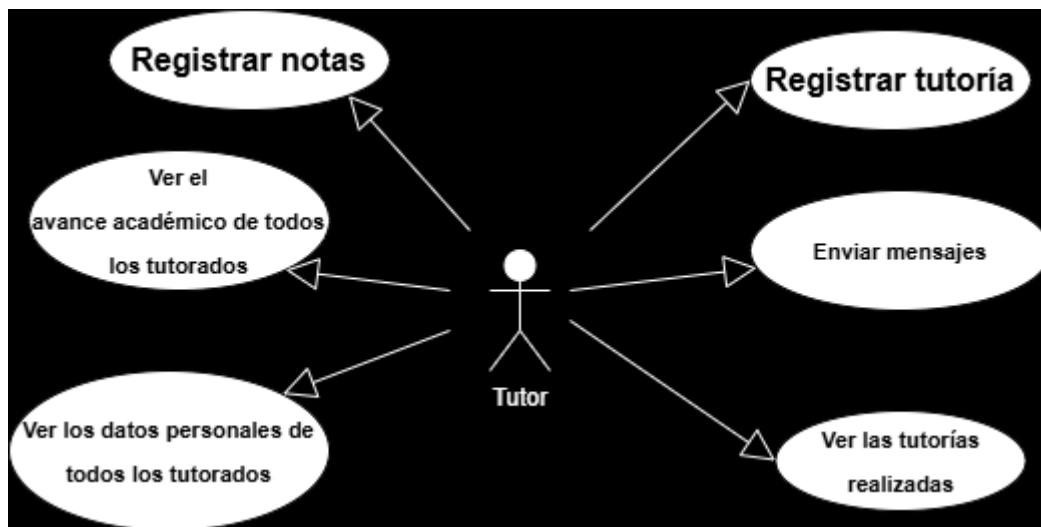


Ilustración 6. Casos de uso del tutor

Este usuario representa a quien ha ingresado como tutor, las acciones que puede realizar son: registrar notas, registrar y ver la lista de tutorías, enviar mensajes, visualizar el avance académico y los datos personales de sus tutorados.

5 Diseño de la base de datos

La base de datos está compuesta por varias tablas, cada una con una función específica y relaciones bien definidas para asegurar la integridad y coherencia de los datos.

Aquí se presenta un resumen de cada tabla y su propósito:

1. Tabla areas

Propósito: Almacena las áreas de formación de las materias.

Campos: idAreas (Clave primaria) y nombre.

2. Tabla estatus

Propósito: Contiene los diferentes estatus que pueden tener los atributos, como los alumnos, tutores, entre otros.

Campos: idEstatus (Clave primaria), nombreEstatus y descripcion.

3. Tabla tutores

Propósito: Guarda la información de los tutores.

Campos: idTutores (Clave primaria), nombre, segundoNombre, apPaterno (apellido parterno), apMaterno (apellido materno), fechaNac (fecha de nacimiento), numero, correo, direccion, contra (contraseña para iniciar sesión) y fkEstatus (Clave foránea a estatus)

4. Tabla alumnos

Propósito: Almacena la información de los alumnos.

Campos: matricula (Clave primaria), nombre, segundoNombre, apPaterno (apellido parterno), apMaterno (apellido materno), fechaNac (fecha de nacimiento), numero, correo, direccion, contra (contraseña para iniciar sesión), carrera, semestre, fkTutor (Clave foránea a tutores) y fkEstatus (Clave foránea a estatus).

5. Tabla materias

Propósito: Contiene la información de las materias disponibles.

Campos: idMaterias (Clave primaria), nombre, credits y fkArea (Clave foránea a areas).

6. Tabla notas

Propósito: Almacena las notas que los tutores dejan sobre los alumnos.

Campos: idNotas (Clave primaria), fkTutor (Clave foránea a tutores), fkAlumno (Clave foránea a alumnos) y notas.

7. Tabla tutorias

Propósito: Guarda los registros de las tutorías realizadas.

Campos: idTutorias (Clave primaria), fecha, contexto, duracion, fkAlumno (Clave foránea a alumnos), fkTutor (Clave foránea a tutores) y fkEstatus (Clave foránea a estatus).

8. Tabla materias_alumnos

Propósito: Relaciona a los alumnos con las materias que cursan, incluyendo su tipo de inscripción y calificación.

Campos: fkMaterias (Clave foránea a materias), fkAlumnos (Clave foránea a alumnos), fkEstatus (Clave foránea a estatus), inscripcion y calificacion.

9. Tabla mensajes

Propósito: Almacena los mensajes intercambiados entre tutores y alumnos.

Campos: idMensajes (Clave primaria), mensaje, asunto, fecha, fkTutor (Clave foránea a tutores), fkAlumno (Clave foránea a alumnos) y fkEstatus (Clave foránea a estatus).

El diseño de la base de datos ha sido cuidadosamente planificado para garantizar:

- Escalabilidad: Facilita la expansión de la base de datos con nuevas tablas y campos según sea necesario.

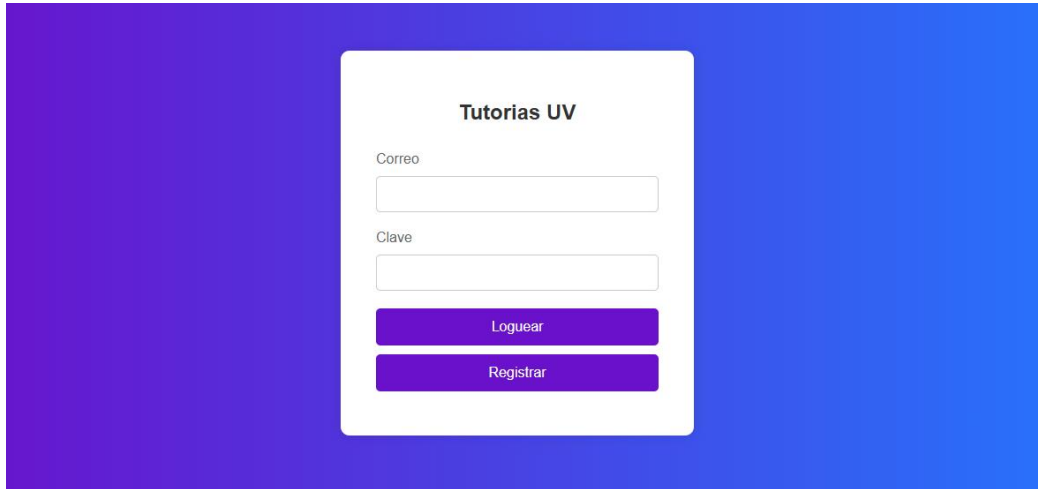


- Integridad de Datos: A través de claves foráneas y restricciones, se asegura que los datos sean consistentes y correctos.
- Eficiencia: Optimización de las consultas para un acceso rápido y eficiente a los datos.

Este diseño de base de datos soporta eficazmente las funcionalidades del módulo de seguimiento y gestión de tutorías, permitiendo una administración efectiva de los tutores, alumnos, materias, tutorías, etcétera.

6 Interfaz Gráfica del Usuario (GUI)

6.1 Inicio de sesión

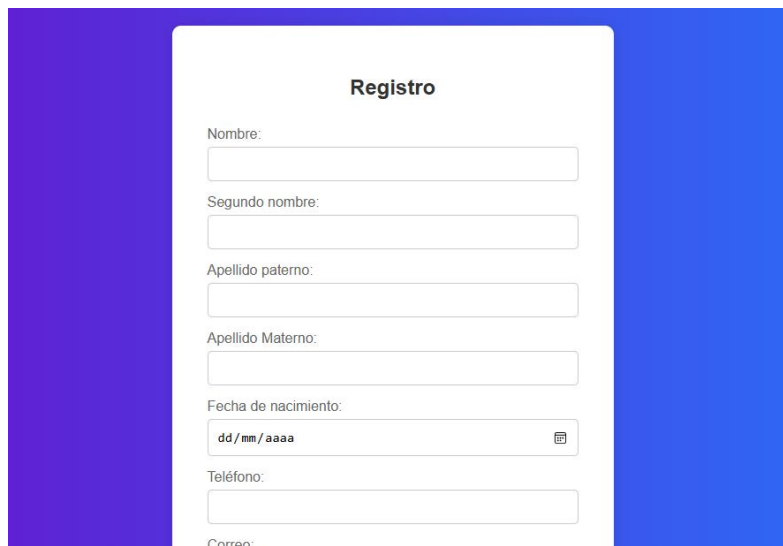


The screenshot shows a login form titled "Tutorías UV" centered on a blue gradient background. The form has two input fields: "Correo" (Email) and "Clave" (Password). Below these fields are two buttons: "Loguear" (Login) and "Registrar" (Register).

Ilustración 7. Interfaz para iniciar de sesión

Al ejecutar la aplicación web, la primer vista que ve el usuario es la vista para el inicio de sesión. Si el usuario ya se ha registrado en la base de datos, podrá iniciar con el correo y contraseña que proporcionó; de lo contrario, tendrá que dirigirse a la interfaz de registro.

6.2 Registro



The screenshot shows a registration form titled "Registro" centered on a blue gradient background. The form has several input fields: "Nombre:" (Name), "Segundo nombre:" (Second name), "Apellido paterno:" (Paternal surname), "Apellido Materno:" (Maternal surname), "Fecha de nacimiento:" (Date of birth) with a date picker showing "dd/mm/aaaa", "Teléfono:" (Phone), and "Correo:" (Email).

Ilustración 8. Interfaz para registro

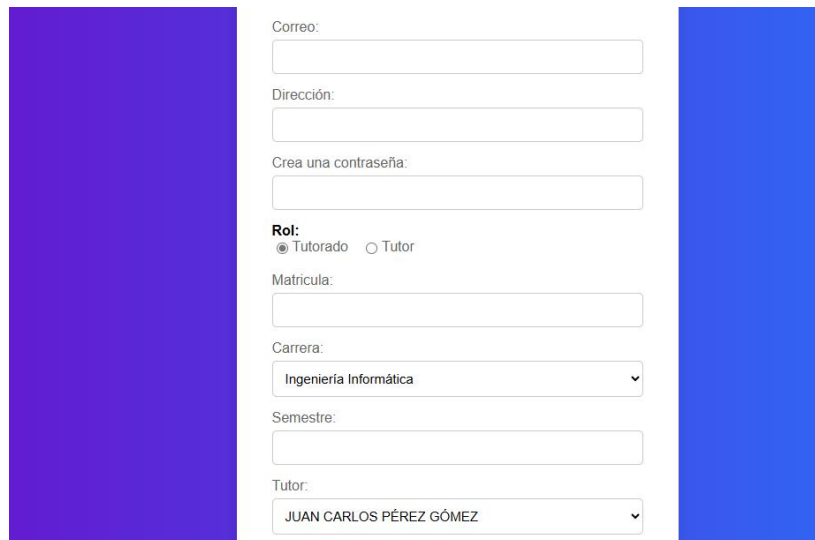


Ilustración 9. Interfaz para registro de tutorado

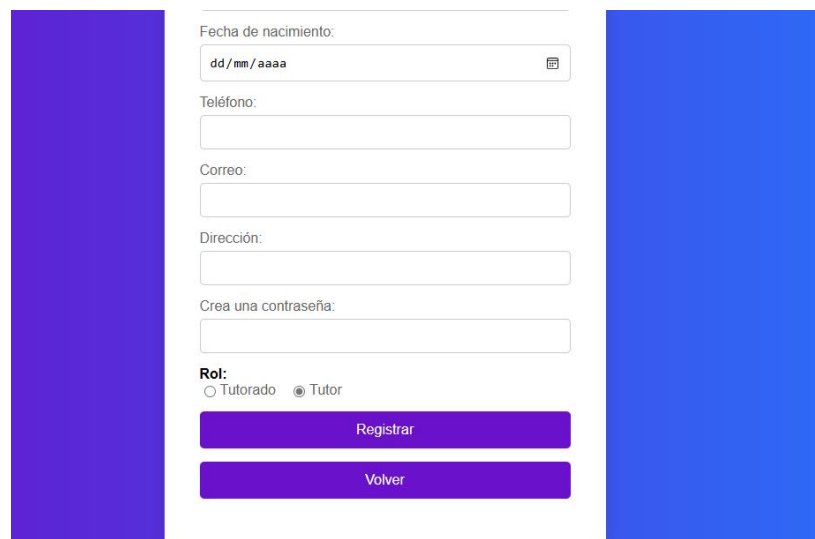


Ilustración 10. Interfaz para registro de tutor

Como ya se menciono, esta vista se utiliza cuando el usuario no se ha registrado anteriormente, y por ende, no puede iniciar sesión. En esta vista el usuario tendrá que llenar sus datos personales, junto con un correo y contraseña, y seleccionar si es tutor o tutorado. Al presionar el botón 'Registrar', los datos se almacenarán en la base de datos conectada a la aplicación web.

6.3 Vistas del tutor

6.3.1 Menú principal

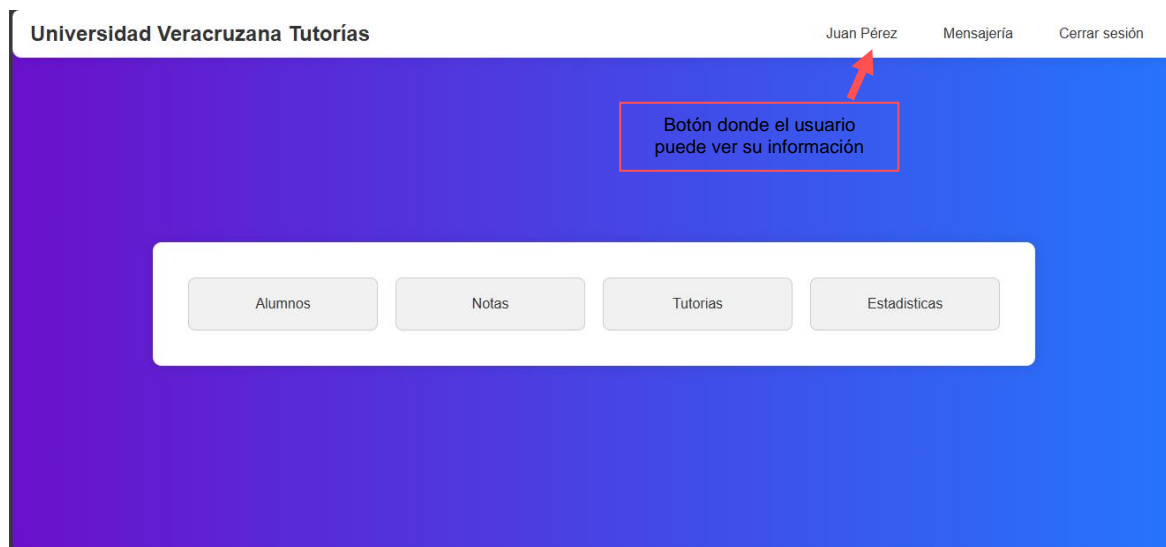


Ilustración 11. Menú principal de tutores

Esta es la interfaz que se muestra cuando un tutor inicia sesión. Aquí, el tutor podrá ver todas las funcionalidades con las que dispone, como la mensajería, las tutorías, las notas, entre otras que se mencionarán a continuación.

6.3.2 Información personal de tutor

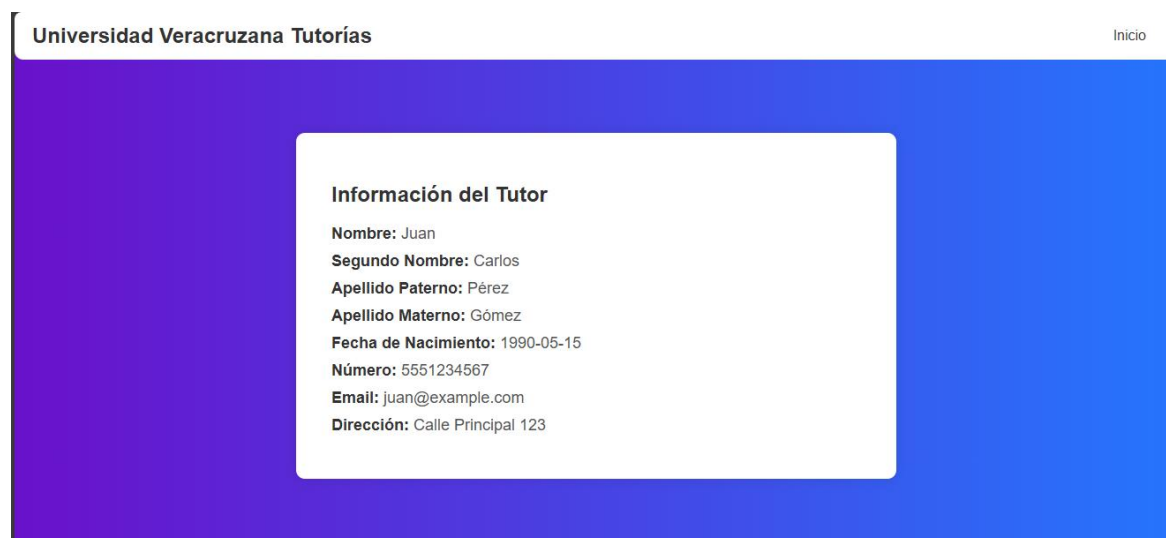


Ilustración 12. Vista con información de tutores

Esta es una vista muy simple, pero en la cual el tutor puede ver toda la información personal que proporcionó con anterioridad.

6.3.3 Mensajería para tutores

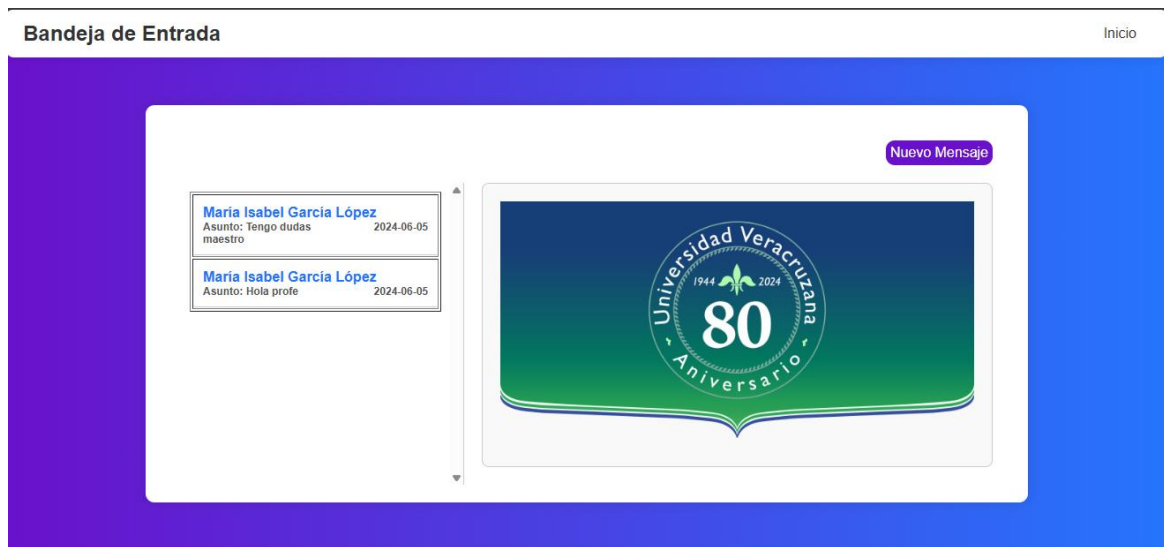


Ilustración 13. Bandeja de entrada de tutores

Esta es la interfaz que ve el alumno en el apartado de mensajería. Como se puede observar, en ella se encuentra su bandeja de entrada con todos los mensajes que le han mandado y un botón para redactar un nuevo mensaje.

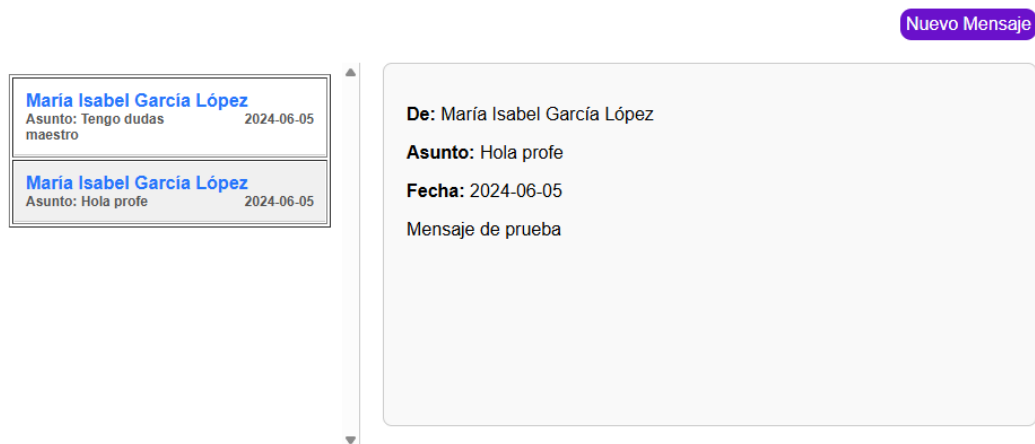


Ilustración 14. Detalle de mensajes de tutores

Cuando se selecciona un mensaje, en el contenedor se quita la imagen del principio y se ven los detalles del mensaje en cuestión.



Ilustración 15. Vista para redactar mensajes de tutores

Al presionar el botón para enviar un nuevo mensaje, la interfaz cambia a esta otra donde podrá seleccionar a quién le enviará el mensaje, y escribir el asunto y el cuerpo del mensaje. Esta vista también cuenta con un botón de enviar, el cuál almacena el mensaje en la base de datos, y un botón para cancelar y volver a la bandeja de entrada.

Ilustración 16. Ejemplo para seleccionar remitente de mensajes

Para seleccionar a quién le enviará el mensaje, el tutor cuenta con una lista de todos sus tutorados.

6.3.4 Sección para ver tutorados

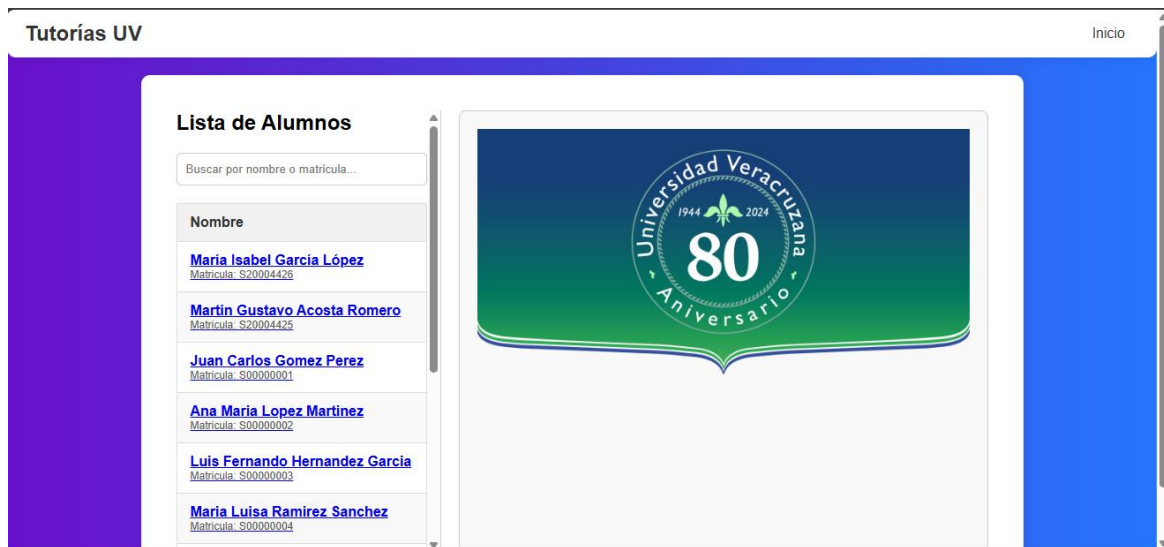


Ilustración 17. Listado de alumnos de tutores

En esta vista se muestra la lista de todos los tutorados con los que cuenta el tutor.

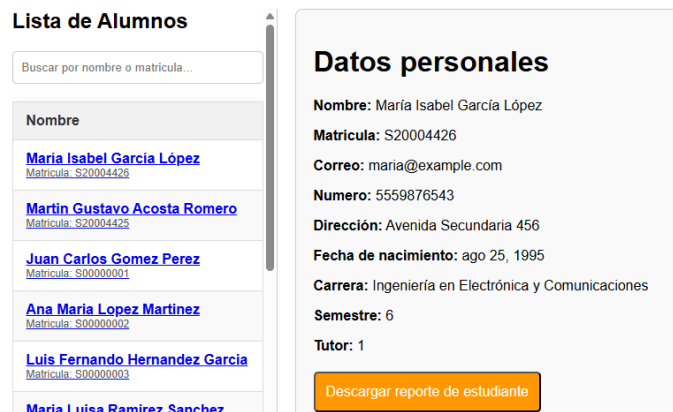


Ilustración 18. Detalle de alumnos en interfaz de tutores

Al seleccionar un alumno, el tutor podrá ver toda la información y podrá descargar el reporte de las materias aprobadas y su avance en formato PDF.

REPORTE DE ESTUDIANTE

Nombre: María Isabel García López
 Matricula: S20004426
 Correo: maria@example.com
 Numero: 5559876543
 Direccion: Avenida Secundaria 456
 Fecha de Nacimiento: 1995-08-25
 Carrera: Ingeniería en Electrónica y Comunicaciones
 Semestre: 6
 Tutor: 1

Area formación basica

Materia	Créditos	Inscripción	Calificación
Precalculo	6	1	90
Química	6	1	85
Física	6	1	88
Servicio Social	8	1	87
Cálculo Multivariable	8	1	89
Fund. de Redes de Computadoras	8	1	91
Programación Estructurada	8	1	92
Estructura de Datos	8	1	84

Ilustración 19. Ejemplo uno de reporte de estudiantes

Estudiante I	72	11	87.1
Lengua II	4	1	88

Promedio Total: 87.72

Area	Créditos Alumno	Créditos Totales
Area formación basica	294	338
Area formación disciplinaria	66	66
Area formación terminal	16	16
Area formación libre	8	8
Totales	384	428



Ilustración 20. Ejemplo dos de reporte de estudiantes

Esto es un ejemplo del archivo PDF que se descarga y su contenido.



6.3.5 Notas

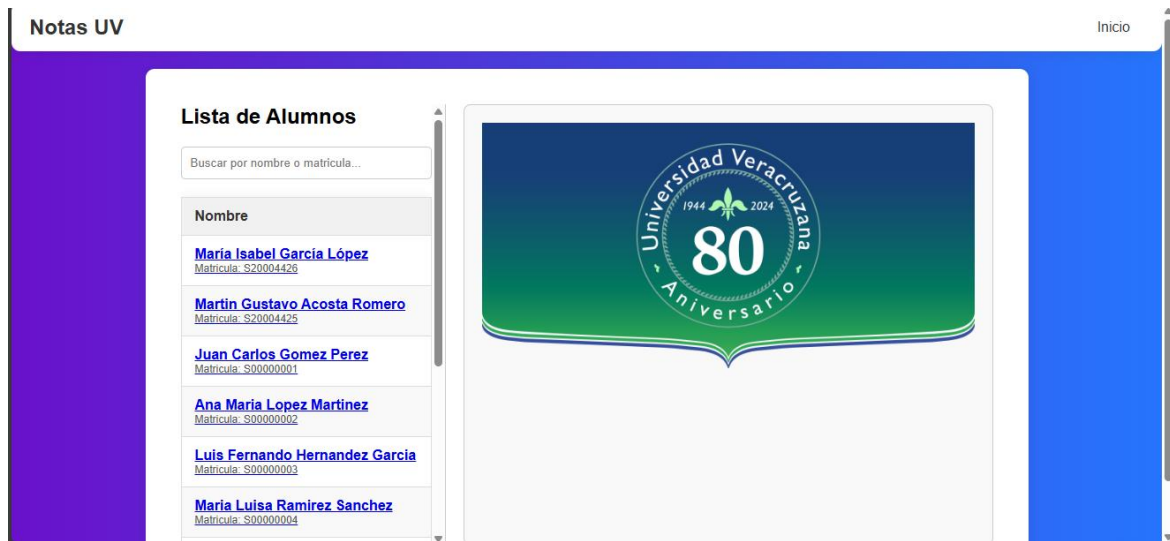


Ilustración 21. Vista de notas



Ilustración 22. Vista al seleccionar un alumno en notas

En la vista de notas, primero se muestra la lista de tutorados con los que cuenta el tutor, al seleccionar alguno, el tutor puede ver la lista de notas que ha redactado (si lo ha hecho) y un enlace que lo llevará al apartado para redactar una nueva.



The screenshot shows a web interface titled 'Sistema de Notas' with a 'Inicio' link in the top right. The main content area has a blue gradient background. In the center, there is a white box titled 'Registrar Nota'. Inside this box, there is a label 'Nota' above a large text input field. Below the input field, there are two buttons: 'Registrar' (blue) and 'Cancelar' (red).

Ilustración 23. Registrar nueva nota

En esta vista, el tutor puede redactar una nueva nota para el tutorado que anteriormente seleccionó, al presionar el botón ‘Registrar’ la nota se guardará en la base de datos y al presionar el botón ‘Cancelar’ volverá a la vista con la lista de alumnos.

6.3.6 Tutorías

The screenshot shows a web interface with two main sections. On the left, titled 'Lista de Alumnos', there is a search bar with the placeholder text 'Buscar por nombre o matrícula...'. Below the search bar is a table with the following data:

Nombre
Maria Isabel Garcia López Matrícula: S20004426
Martin Gustavo Acosta Romero Matrícula: S20004425
Juan Carlos Gomez Perez Matrícula: S00000001
Ana Maria Lopez Martinez Matrícula: S00000002

On the right, titled 'Tutorías', there is a link 'Registrar nueva tutoría' and a list of items:

- may 29, 2024 - Contexto de la tutoría (60 minutos)
- jun 5, 2024 - Otro contexto de la tutoría (45 minutos)

Ilustración 24. Detalle de tutorías

Al igual que en la vista de notas, en la vista de tutorías, primero se muestra la lista de tutorados con los que cuenta el tutor, al seleccionar alguno, el tutor puede ver la lista de tutorías que se han registrado y un enlace que lo llevará al apartado para registrar una nueva.



Sistema de Tutorías Inicio

Registrar tutoría

Matricula

Fecha

Contexto

Duración (minutos)

Ilustración 25. Registrar nueva tutoría

En esta vista, el tutor puede registrar una nueva tutoría, con la matricula del tutorado, la fecha y el contexto de lo que trató la tutoría, además de la duración. Al presionar el botón ‘Registrar’ la tutoría se almacenará en la base de datos y al presionar el botón ‘Cancelar’ volverá a la sección de tutorías.

6.3.7 Estadísticas



Ilustración 26. Estadísticas de la aplicación

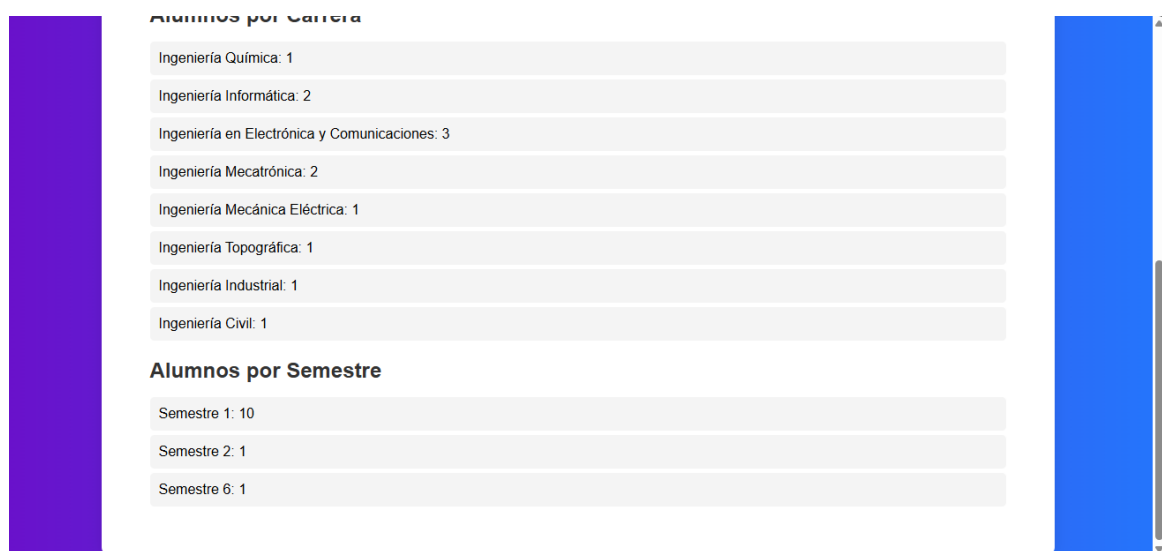


Ilustración 27. Más detalle de estadísticas de la aplicación

Por último, esta vista sirve para que el tutor pueda ver las estadísticas de la aplicación web. Esta vista incluye, entre otros detalles, el número de tutores, tutorados y tutorías programadas.

6.4 Vistas del tutorado

6.4.1 Menú principal

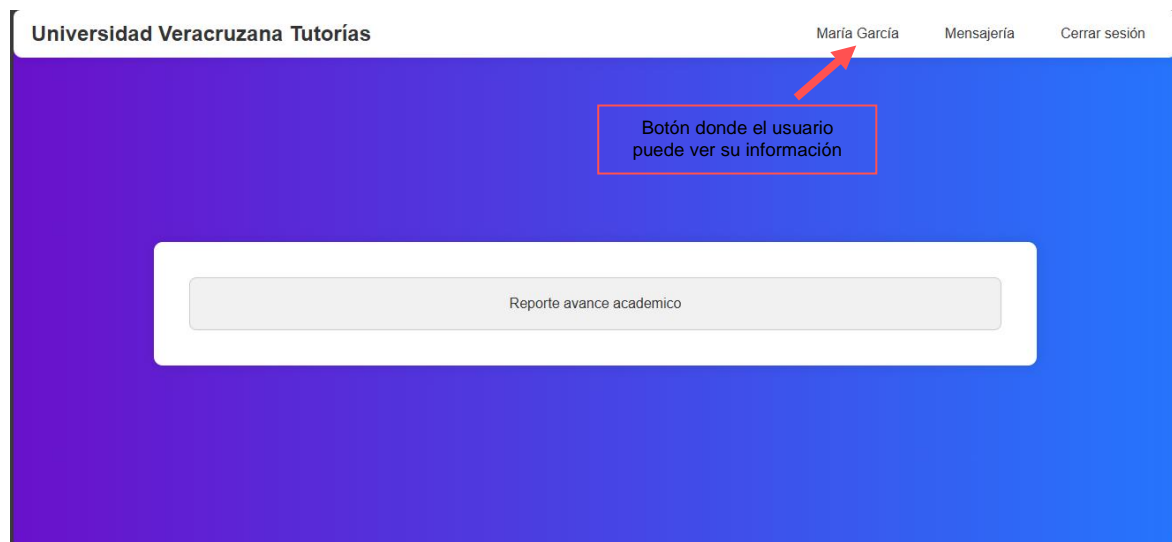


Ilustración 28. Menú principal del tutorado

Esta es la vista que se muestra cuando un usuario ingresa como tutorado, la vista contiene un botón para ver su información, otro para mensajería, otro para cerrar sesión y uno

más para generar un informe de sus calificaciones y avance en formato PDF, el cual ya fue mostrado anteriormente.

6.4.2 Información personal del tutorado



Ilustración 29. Información personal del tutorado

Como en la interfaz de tutores, en esta vista el tutorado puede ver toda la información personal que proporcionó con anterioridad.

6.4.3 Mensajería de alumnos



Ilustración 30. Bandeja de entrada del tutorado

En esta vista, el tutorado podrá ver la lista de los mensajes que su tutor le haya mandado, también cuenta con un botón en el cuál él podrá mandar un mensaje.



Redactar mensajes Inicio

Para:
juan@example.com

Asunto:
Tutorias...

Mensaje:
Escribe tu mensaje...

Enviar Mensaje

Cancelar

Ilustración 31. Vista para redactar mensajes de tutorados

Al presionar tal botón, se le enviará a la vista para redactar un nuevo mensaje. En esta vista el tutorado puede redactar un nuevo mensaje para su tutor presionando el botón 'Enviar Mensaje' después de haber llenado los campos de asunto y cuerpo de mensaje.

7 Conclusiones

Hemos diseñado y desarrollado una aplicación web que cumple con los requerimientos específicos y generales establecidos. Utilizando tecnologías como Java, JSP y PostgreSQL, hemos creado una plataforma que facilita la interacción entre tutores y tutorados, permitiendo un seguimiento detallado del progreso académico y una comunicación efectiva.

El diseño de la base de datos asegura la integridad y eficiencia de los datos, con tablas estructuradas para gestionar alumnos, tutores, materias, tutorías y mensajes de manera coherente y organizada.

La interfaz de usuario, desarrollada con JSP, garantiza una experiencia intuitiva y accesible, facilitando a los usuarios (tutores y tutorados) el acceso a información relevante y la realización de acciones como el registro de notas, la programación de tutorías y la comunicación a través de mensajes.

Durante la realización del proyecto he podido comprobar lo útiles que son los conocimientos adquiridos durante las asignaturas como Ingeniería del Software, Base de Datos, Paradigmas de Programación entre otras.

Cabe destacar que, por tratarse de un prototipo, la funcionalidad de la aplicación web puede no ser completa y queda pendiente para futuras actualizaciones con base a las recomendaciones que se nos brinden. Sin embargo, se implementaron todos los requerimientos específicos que se mencionaron con anterioridad.



8. Referencias

draw.io - free flowchart maker and diagrams online. (s. f.).

<https://app.diagrams.net/?src=about>

Lucid | the leading visual collaboration platform. (s. f.). <https://lucid.co/>

GitHub: Let's build from here. (2024). GitHub. <https://github.com/>

Mvn Repository. (2024). <https://mvnrepository.com/>

9 Anexos

9.1 DevOps

Al trabajar con la metodología DevOps, tuvimos que ponernos de acuerdo con otra pareja que estaba trabajando en un proyecto que intenta complementar el nuestro, ya que también trabajan con el sistema de tutorías de la Universidad Veracruzana; nos pusimos de acuerdo para hacer una retroalimentación mutua de nuestro trabajo y así poder mejorar aspectos de funcionamiento y/o estilo.

La retroalimentación que nos hicieron fue la siguiente:

“Hemos revisado su aplicación y consideramos que han hecho un muy buen trabajo respecto a las interfaces y funciones que en ella ofrecen. Sin embargo, tenemos algunas recomendaciones que quizá podrían ayudar a mejorarla. En primer lugar, podrían considerar un selector de fecha más intuitivo como un calendario desplegable a la hora de asignar la fecha de la tutoría, también podrían considerar añadir un efecto de hover para los botones para que a la hora de pasar el cursor sobre ellos cambie de color, por último, asegurar de que todos los elementos de la interfaz sigan un mismo estilo en cuanto a colores y tipografía.”

Por supuesto que recibimos esta retroalimentación de forma positiva y aplicamos los consejos que nos dieron para que nuestra aplicación sea aún mejor de como la planeamos.