



UNIVERSIDAD NACIONAL DE SAN AGUSTIN
FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y
SERVICIOS
DEPARTAMENTO ACADÉMICO DE INGENIERÍA
DE SISTEMAS E INFORMÁTICA
ESCUELA PROFESIONAL DE INGENIERÍA DE
SISTEMAS



Proyecto web: Desarrollo de una aplicación web para realizar la inscripción de estudiantes

Fecha: 2024/07/28



UNSA

UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE AREQUIPA

Proyecto: Sistema de Matrículas

INFORMACIÓN BÁSICA

ASIGNATURA:	Programación Web II
NOMBRE DEL PROYECTO:	SISMAT - UNSA

INTEGRANTES:

- Caceres Ruiz, Johann Andre - jcaceresruiz@unsa.edu.pe
- Mamani Céspedes, Jhonatan Benjamin - jmamanices@unsa.edu.pe
- Valdivia Segovia, Ryan Fabian - rvaldiviase@unsa.edu.pe

DOCENTES:

- Richart Smith Escobedo Quispe - rescobedoq@unsa.edu.pe

Sistema de Matrícula

License repo not found Downloads repo not found Last Commit repo not found



OBJETIVOS, TEMAS Y COMPETENCIAS

OBJETIVOS

- Desarrollar una aplicación web para la matrícula de cursos.
- Permitir a los estudiantes iniciar sesión, seleccionar cursos y grupos, y confirmar la matrícula.
- Generar un horario y una constancia de matrícula descargable.

TEMAS

- Autenticación de usuarios
- Manejo de estado en React
- Integración de frontend y backend con APIs RESTful
- Estilos y diseño responsivo con Tailwind y Material UI (MUI)
- Uso de SQLite para la base de datos
- Control de versiones con Git y GitHub

CONTENIDO DEL PROYECTO

MARCO CONCEPTUAL

- Desarrollo Web Full-Stack
 - Uso de Django para el backend y React con Vite para el frontend.
 - Diseño de componentes reutilizables con Material UI.
 - Estilización de la aplicación con Tailwind CSS.
 - Implementación de APIs RESTful para la comunicación entre frontend y backend.
 - Almacenamiento de datos con SQLite.
 - Control de versiones y colaboración con Git y GitHub.
- Herramientas y Tecnologías:
 - **Django:** Framework de backend para desarrollo web en Python.
 - **React:** Biblioteca de JavaScript para construir interfaces de usuario.
 - **Vite:** Herramienta de desarrollo rápida para proyectos de frontend.
 - **Tailwind CSS:** Framework de CSS utilitario para estilización rápida y eficiente.
 - **Material UI (MUI):** Biblioteca de componentes de UI para React.
 - **SQLite:** Sistema de gestión de bases de datos ligero y autónomo.
 - **Node.js:** Entorno de ejecución para JavaScript en el backend.
 - **Git:** Sistema de control de versiones distribuido.
 - **GitHub:** Plataforma para hospedaje de código fuente y colaboración.

INSTRUCCIONES DE INSTALACIÓN

1. Clonar el repositorio:

```
git clone https://github.com/RyanValdivia/sismat-unsa.git
cd sismat-unsa
```

2. Configurar el entorno virtual de Python:

```
python -m venv venv  
source venv/bin/activate
```

3. Instalar dependencias del backend (Django):

```
cd Backend/sismat  
pip install -r requirements.txt
```

4. Instalar dependencias del frontend (React y Vite):

```
cd Frontend/sismat  
npm install
```

5. Iniciar el servidor de desarrollo de Django:

```
python manage.py runserver
```

6. Iniciar el servidor de desarrollo de Vite:

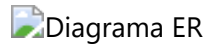
```
npm run dev
```

PROCESO SEGUIDO

Planificación

1. Definición de Requisitos:

- Realizamos varias reuniones iniciales para definir los objetivos y alcances del proyecto, asegurándonos de cubrir todas las necesidades de los usuarios.
- Identificación de las necesidades de los usuarios y especificaciones funcionales.
- Definición de las relaciones entre las entidades del sistema, utilizando diagramas ER para visualizar la estructura de la base de datos.
- En el proceso de planificación, colaboramos estrechamente para entender los requisitos y establecer prioridades claras para el desarrollo.
- Aquí está nuestro diagrama ER básico utilizado en este proceso:



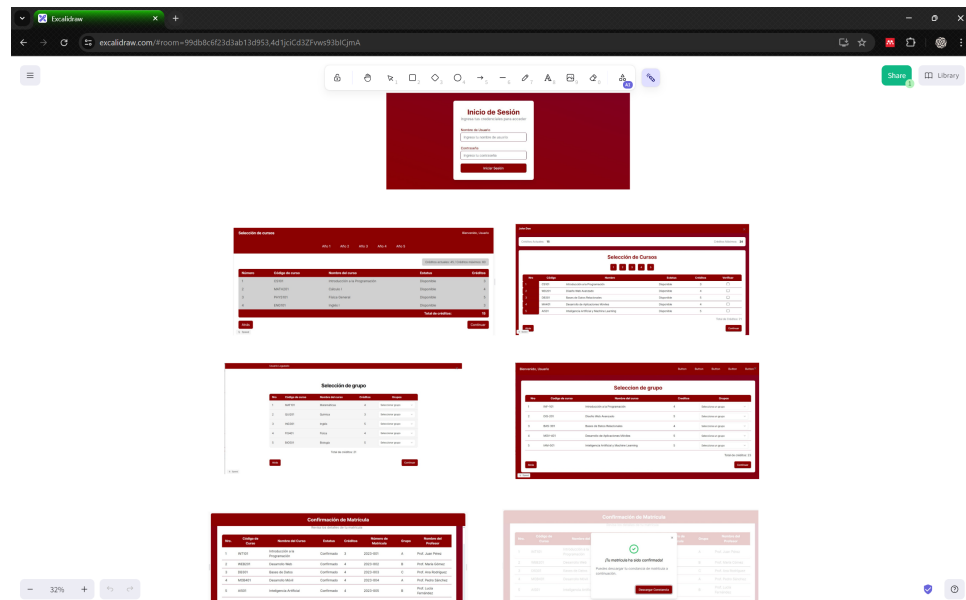
2. Análisis:

- Se hizo un estudio de factibilidad técnica para implementar en el código, validaciones y ciertos requerimientos.
- Se evaluaron las tecnologías a utilizar en el proyecto (Django, React, Vite, Tailwind, etc.).
- La elección de **Django** como backend se basó en su robustez, la rápida implementación de API RESTful y su capacidad para manejar la lógica del lado del servidor de manera eficiente.
- **React** fue seleccionado para el frontend debido a su flexibilidad, su capacidad para crear interfaces de usuario interactivas y su amplia comunidad de soporte.
- Se optó por **Vite** como el entorno de desarrollo debido a su velocidad y eficiencia en la compilación de proyectos React, mejorando significativamente los tiempos de desarrollo.
- **Tailwind CSS** se eligió para el diseño de la interfaz de usuario por su enfoque en clases utilitarias, lo que permite un diseño responsivo y personalizado sin sobrecargar el proyecto con CSS adicional.
- Estas tecnologías combinadas nos permitieron crear una aplicación escalable, mantenible y eficiente tanto en términos de rendimiento como de desarrollo.

Diseño

3. Diseño del Sistema:

- Creación de diagramas de arquitectura y diseño de base de datos.
- Realizamos los diseños iniciales y diagramas en Excalidraw para visualizar y planificar la estructura visual del sistema.
- **Diagrama de diseño:**



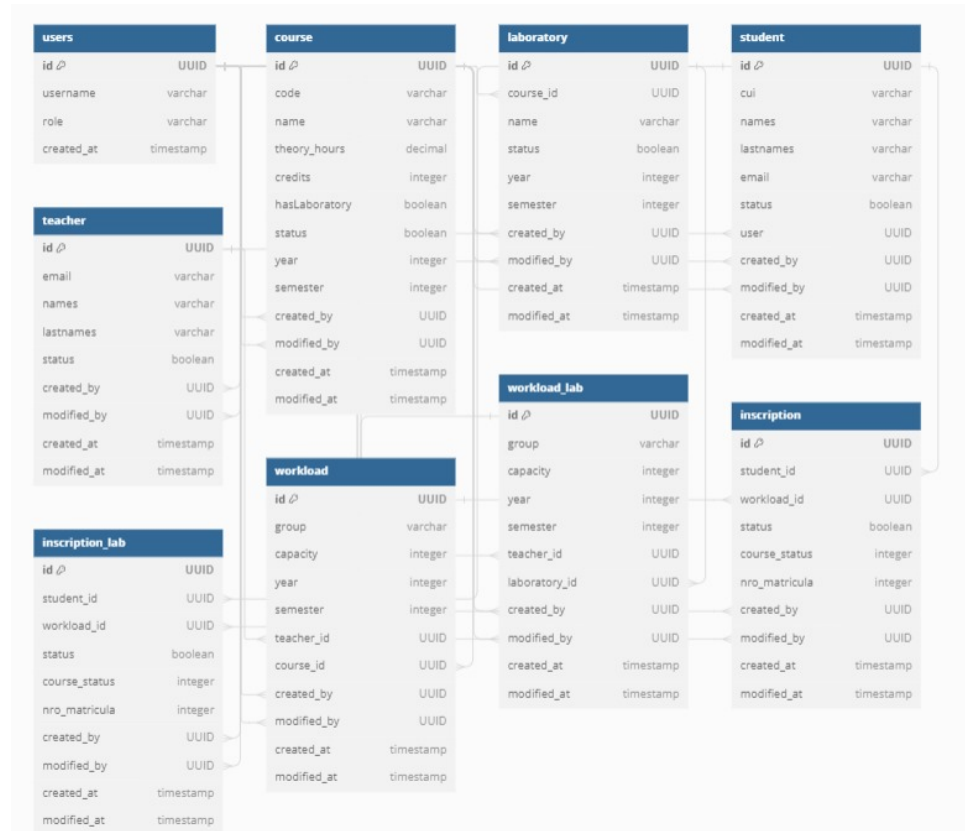
4. Planificación de Tareas:

- La división del trabajo se realizó en sprints y asignación de tareas específicas tras algunas sesiones Meet.
- Se hizo uso de herramientas de gestión de proyectos para determinar tiempos y tareas específicas.

Implementación

5. Desarrollo del Backend:

- Configuración de Django y creación de modelos, vistas, y controladores a partir del modelo de la base de datos que generamos:



- Implementación de la API RESTful para la comunicación con el frontend.

6. Desarrollo del Frontend:

- La configuración se realizó con React y Vite.
- Creación de componentes reutilizables y diseño responsivo con Tailwind y MUI.

Pruebas

7. Pruebas Unitarias:

- Se siguió una implementación de pruebas unitarias para asegurarse de que cada componente funciona correctamente.
- Se hizo uso de herramientas como Jest y Django Test.

8. Pruebas de Integración:

- Pruebas de flujo completo del sistema para asegurarse de que todos los componentes trabajan en conjunto.
- Validación de la interfaz de usuario y la API RESTful.
- Últimas observaciones para preparar el proyecto para hacer deploy.

Despliegue

9. Preparación para el Despliegue:

- Se realizó la configuración del entorno de producción.
- Se pensó sobre el uso de servicios de despliegue como Heroku, Vercel, o AWS.

10. **Despliegue:**

- Lanzamiento de la aplicación en el entorno de producción.
 - Monitoreo inicial y ajustes necesarios.
 - Nuestro sitio ya está desplegado y accesible en el siguiente enlace: [SISMAT-UNSA](#)
-

PDF Generado del README.md

- **Adjuntar PDF del README.md:** [Descargar PDF](#)

Lista de Videos

1. **BackEnd-Admin**

- [Video - API Rest BackEnd](#)

2. **FrontEnd-Framework**

- [Video - Framework FrontEnd](#)

3. **Proyecto en Producción**

- [Video - Proyecto en Producción](#)

RECURSOS

- [Documentación de Django](#)
- [Documentación de React](#)
- [Guía de Vite](#)
- [Documentación de Tailwind CSS](#)
- [Componentes de Material UI](#)
- [Documentación de SQLite](#)
- [Documentación de Node.js](#)
- [Documentación de Git](#)
- [Documentación de GitHub](#)

AUTORES

- [Caceres Ruiz, Johann Andre](#)
- [Mamani Céspedes, Jhonatan Benjamin](#)
- [Valdivia Segovia, Ryan Fabian](#)

LICENCIA

Este proyecto está licenciado bajo los términos de la licencia MIT.