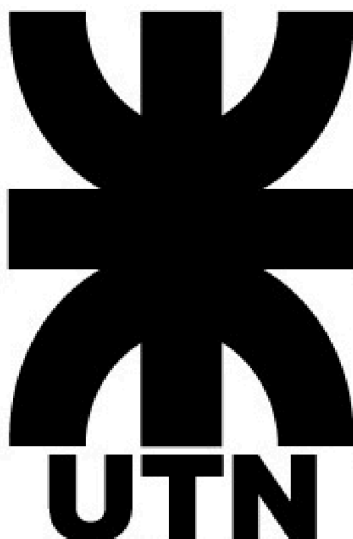


U.T.N. – Facultad Regional Resistencia

Ingeniería en Sistemas de Información



Trabajo Práctico Integrador

Equipo docente:

- Fernández, Jose Alejandro
- Villaverde, Jorge Eduardo

Grupo 4 - Integrantes:

- Benetucci, Ignacio Agustín - ignaciobenetucci@gmail.com
- Bravo, Juan Pablo - bravojuanpablo153@gmail.com
- Fernandez, Andrés Darío - androthepunk@ca.frre.utn.edu.ar
- Gallego, Adriel Fernando - gallegoadriel99@ca.frre.utn.edu.ar
- Ibaigorria, Ignacio Iván - ignacioibaigorria@gmail.com
- Nikcevich, Adriano - adri.nikce30@gmail.com
- Nikcevich, Iván Josué - ivan.nikcevich@hotmail.com

Cátedra: Desarrollo de Software

CICLO 2024 - SEGUNDO CUATRIMESTRE

ÍNDICE

Introducción.....	2
Arquitectura de la Aplicación.....	3
Front-End.....	3
Back-End.....	4
Base de Datos.....	5
Alcance de la Aplicación.....	6
Conclusión.....	7

Introducción

El presente informe describe el desarrollo de un sistema de gestión para la Bienal Internacional de Escultura del Chaco, diseñado como proyecto académico. El objetivo principal del sistema es optimizar la organización del evento mediante una plataforma que permita registrar y gestionar escultores, esculturas y actividades asociadas. Asimismo, se incorporaron aplicaciones complementarias para que el público pueda interactuar activamente durante el evento, dejando comentarios y participando en votaciones.

El desarrollo del proyecto incluyó la implementación de un front-end dinámico, desarrollado con React, para ofrecer una experiencia de usuario intuitiva y moderna, mientras que el back-end, basado en Django, asegura un procesamiento robusto de los datos y funcionalidades. La solución se apoya en una base de datos PostgreSQL, garantizando la confiabilidad y eficiencia del sistema.

Esta plataforma no solo facilita las tareas administrativas de los organizadores, sino que también potencia la interacción ciudadana, promoviendo una conexión más cercana entre los escultores, sus obras, y el público en general.

Arquitectura de la Aplicación

La arquitectura de la aplicación web ha sido diseñada para garantizar un sistema funcional y eficiente. A continuación, se muestra un diagrama de como fue implementada la arquitectura.

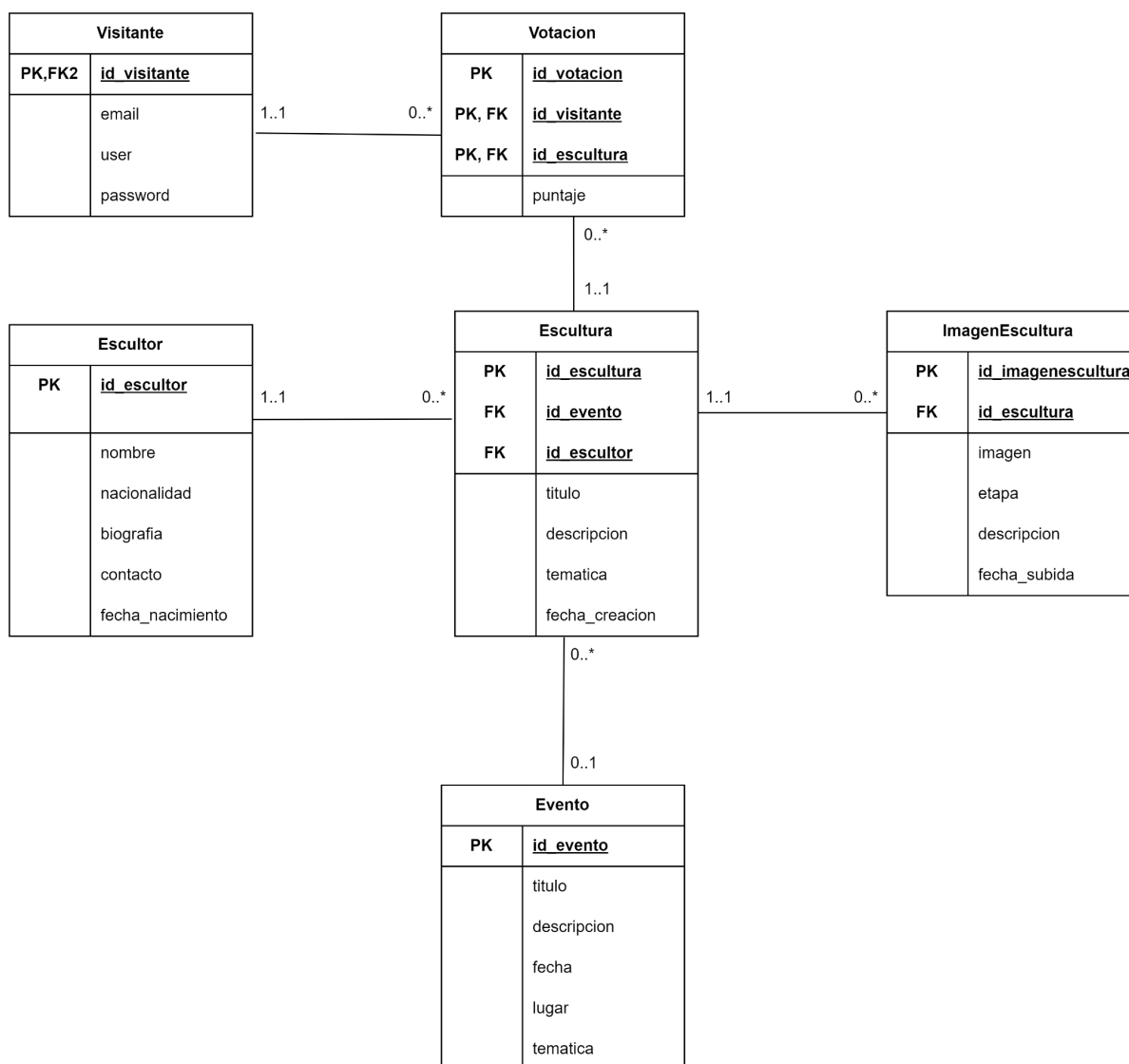


Diagrama de tablas

Front-End

Antes de comenzar con el desarrollo, se utilizó la herramienta Figma para diseñar y prototipar las pantallas de la aplicación. Esto permitió visualizar y validar la estructura de la

interfaz de usuario, garantizando una experiencia intuitiva y alineada con los objetivos del sistema.

La interfaz de usuario se ha desarrollado utilizando React, aprovechando su capacidad para construir componentes reutilizables y gestionar el estado de manera eficiente. Se empleó React Router para manejar la navegación entre las distintas secciones de esculturas, escultores y eventos, creando una experiencia fluida para los usuarios.

El diseño visual fue implementado con CSS y la biblioteca Tailwind CSS, logrando una apariencia moderna y personalizada, adaptada a las necesidades del usuario.

Back-End

El servidor ha sido implementado utilizando Django, un framework en Python que sigue el patrón Modelo-Vista-Controlador (MVC). Este enfoque organiza el sistema en capas bien definidas:

- Capa de presentación: procesa las solicitudes provenientes del Front-End y devuelve las respuestas en formato JSON mediante una API REST, asegurando una comunicación eficiente y estructurada entre las partes del sistema.
- Capa de lógica de negocio: maneja las reglas y operaciones específicas de la aplicación, garantizando que los datos se procesen correctamente según los requisitos funcionales, como la gestión de votos de esculturas y escultores o la validación de datos para la creación de nuevos eventos.
- Capa de acceso a datos: utiliza Django ORM para interactuar con la base de datos PostgreSQL, permitiendo realizar operaciones de lectura, escritura y actualización de manera segura y eficiente.

Para la autenticación y seguridad, se implementó **JSON Web Tokens (JWT)** mediante la biblioteca *djangorestframework-simplejwt*, permitiendo una autenticación basada en tokens que asegura el acceso adecuado a las funcionalidades. Esto facilita la protección de las operaciones críticas, como agregar, modificar o eliminar eventos, esculturas y escultores.

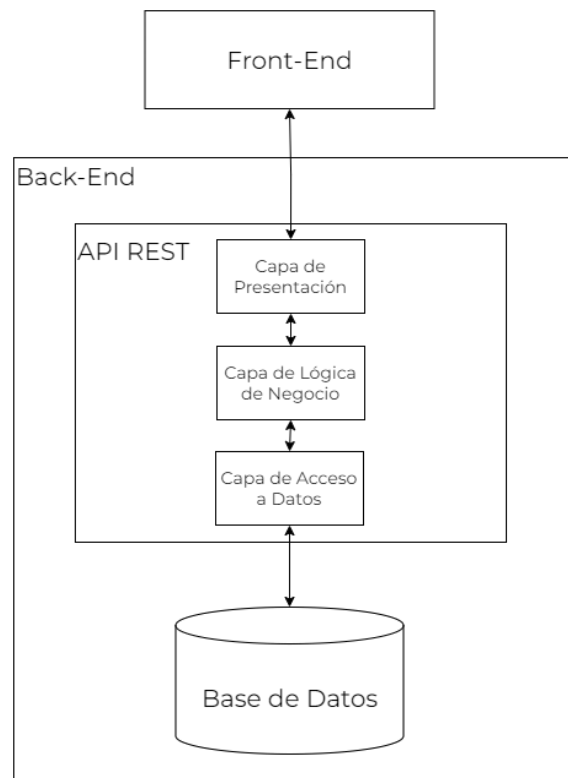
Además, se integró la librería **QRCode** para generar códigos QR dinámicos, lo que permite proporcionar a los usuarios una alternativa para votar por cada escultura. Estos códigos se generan directamente desde el servidor, asegurando su integridad y personalización.

La arquitectura modular no solo facilita el mantenimiento y la escalabilidad del sistema, sino que también asegura que las funcionalidades adicionales, como la autenticación y la

generación de QR, estén perfectamente integradas con las demás capas. Esto permite ofrecer una experiencia segura, eficiente y funcional tanto para los usuarios como para los administradores del sistema.

Base de Datos

El sistema de gestión de bases de datos PostgreSQL ha sido elegido por su confiabilidad, rendimiento y soporte para características avanzadas como tipos de datos personalizados y consultas complejas. La integración con el Back-End se realiza mediante el ORM de Django, que simplifica la interacción con la base de datos a través de modelos definidos en Python. La estructura de la base de datos ha sido diseñada cuidadosamente para cumplir con los requisitos de la aplicación, garantizando la organización eficiente de los datos, así como su integridad y seguridad.



Esquema de la arquitectura

Esta combinación de tecnologías permite construir una aplicación robusta, flexible y preparada para escalar según las necesidades del proyecto.

Alcance de la Aplicación

La aplicación podrá ser utilizada por dos tipos de usuarios: los visitantes (user) y los administradores del sitio (admin).

Desde la pantalla principal, los visitantes podrán navegar por el contenido disponible, mientras que los administradores tendrán acceso a las funcionalidades de gestión del sitio.

Los visitantes podrán:

- Explorar esculturas, escultores y eventos a través de una interfaz intuitiva.
- Ver información detallada de cada escultura, escultor o evento, accediendo a datos adicionales como descripciones, fechas o ubicaciones relevantes.
- Registrarse e iniciar sesión en el sitio.
- Votar esculturas, permitiendo destacar las obras y artistas favoritos, siempre y cuando el visitante se encuentre logueado en la aplicación.

Los administradores podrán:

- Gestionar eventos, esculturas y escultores. Esto incluye:
 - Agregar nuevas esculturas, escultores o eventos al sitio.
 - Editar información existente para mantener los datos actualizados.
 - Eliminar elementos que ya no sean relevantes o estén desactualizados.

La sección de administración permite una gestión sencilla, facilitando el mantenimiento continuo del contenido del sitio y asegurando que la información esté siempre actualizada para los visitantes.

De esta manera, la aplicación ofrece una experiencia completa para ambos tipos de usuarios, asegurando que los visitantes puedan interactuar con el contenido y los administradores puedan gestionar eficientemente el sitio.

Conclusión

El desarrollo del sitio web para la Bienal del Chaco representó un desafío técnico y organizativo que nos permitió aplicar conocimientos adquiridos durante la carrera. La integración de tecnologías modernas como React en el frontend, Django en el backend y PostgreSQL para la base de datos demostró ser efectiva para construir una plataforma funcional y escalable.

Este proyecto no solo cumplió con los requerimientos planteados, sino que también resaltó la importancia de la colaboración entre diseño y desarrollo, asegurando una experiencia fluida tanto para administradores como para usuarios finales. A futuro, podríamos considerar ampliar la funcionalidad del sistema de votación para incluir opciones adicionales, como comentarios o clasificaciones más detalladas.

- Link del repositorio: <https://github.com/FRRe-DS/2024-04-TPI>