



Tecnológico de Monterrey

STC0203

Diseño de componentes de software

César Ignacio Saucedo Rodríguez A01712245

04 / 06 / 2025

Construcción de software y toma de decisiones (Gpo 501)

Cuaderno de Apuntes – STC0203 Diseño de Componentes de Software

1. Especificación de Requisitos

Aprendimos a identificar y documentar requerimientos funcionales y no funcionales.

Utilizamos técnicas como los casos de uso para vincular los requisitos al diseño.

Se enfatizó la trazabilidad desde el requerimiento hasta el componente final.

2. Diseño de Componentes

Aplicamos principios como cohesión alta y acoplamiento bajo.

Diseñamos componentes reutilizables, independientes y bien definidos.

Abordamos la importancia del encapsulamiento, la abstracción y la separación de responsabilidades.

3. Diagramas UML

- Componentes: muestran la estructura modular del sistema y sus dependencias.
 - Secuencia: modelan el flujo de mensajes entre objetos o componentes.
 - Máquinas de estado: representan el comportamiento dinámico de una clase o módulo.
- Estos diagramas se generaron con herramientas como Lucidchart o Draw.io.

4. Persistencia e Integración

Se diseñaron interfaces para que los componentes pudieran acceder a bases de datos de forma segura.

Incorporamos reglas de integridad en la lógica de persistencia.

5. Pruebas y Validación

Se implementaron pruebas unitarias y de integración para validar el funcionamiento correcto de cada componente.

Las pruebas se alinearon con los requisitos especificados, asegurando calidad y funcionalidad.

6. Herramientas aplicadas

- Diseño: Lucidchart, Draw.io
- Versionamiento: Git, GitHub
- Tecnología: Python, HTML/CSS, JavaScript, Django (u otras según proyecto)
- Colaboración: reuniones periódicas de equipo, planificación de tareas y revisión de entregables

7. Reflexión Final

Este curso proporcionó las bases necesarias para estructurar sistemas de software robustos y bien diseñados.

A través de la documentación clara, la aplicación de principios de diseño y la integración de pruebas, desarrollamos competencias clave para proyectos reales de ingeniería de software.

Repositorio del equipo:

<https://github.com/saucedor/OBD>

Diagramas de diseño

El sistema está implementado con una arquitectura Cliente-Servidor, se basa en el Modelo-Vista-Controlador, el cliente interactúa con una vista en HTMX o HTML, CSS, JS (tradicional), El backend del sistema está implementado en Django y la parte del host se implementó en Railway, así mismo la base de datos.

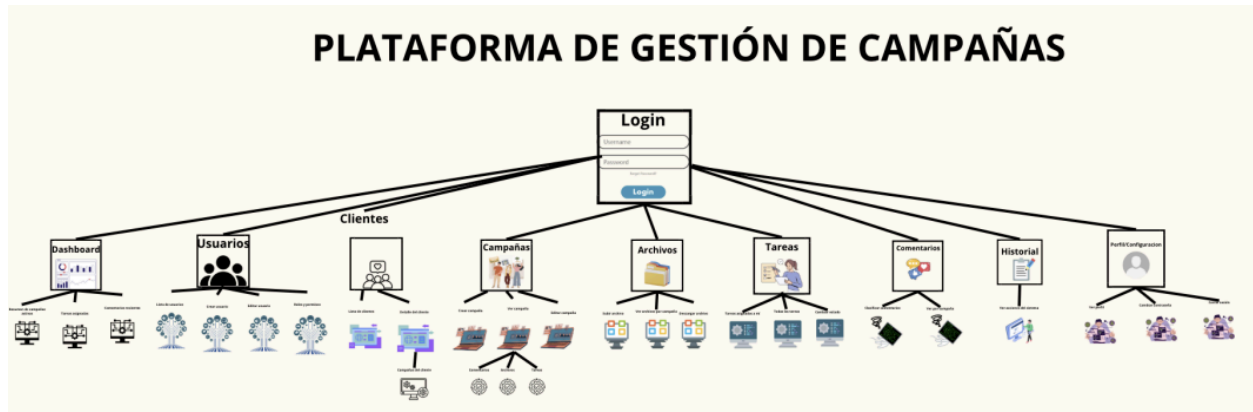


Figura 1. Diagrama de gestión de campañas elaborado por mi compañero Alex

El MER, pasó por varias versiones, en base a las necesidades del socio formador y las demandas del curso, la primera versión es confusa ya que no sabíamos bien la estructura y además todavía no empezábamos a programar el proyecto.

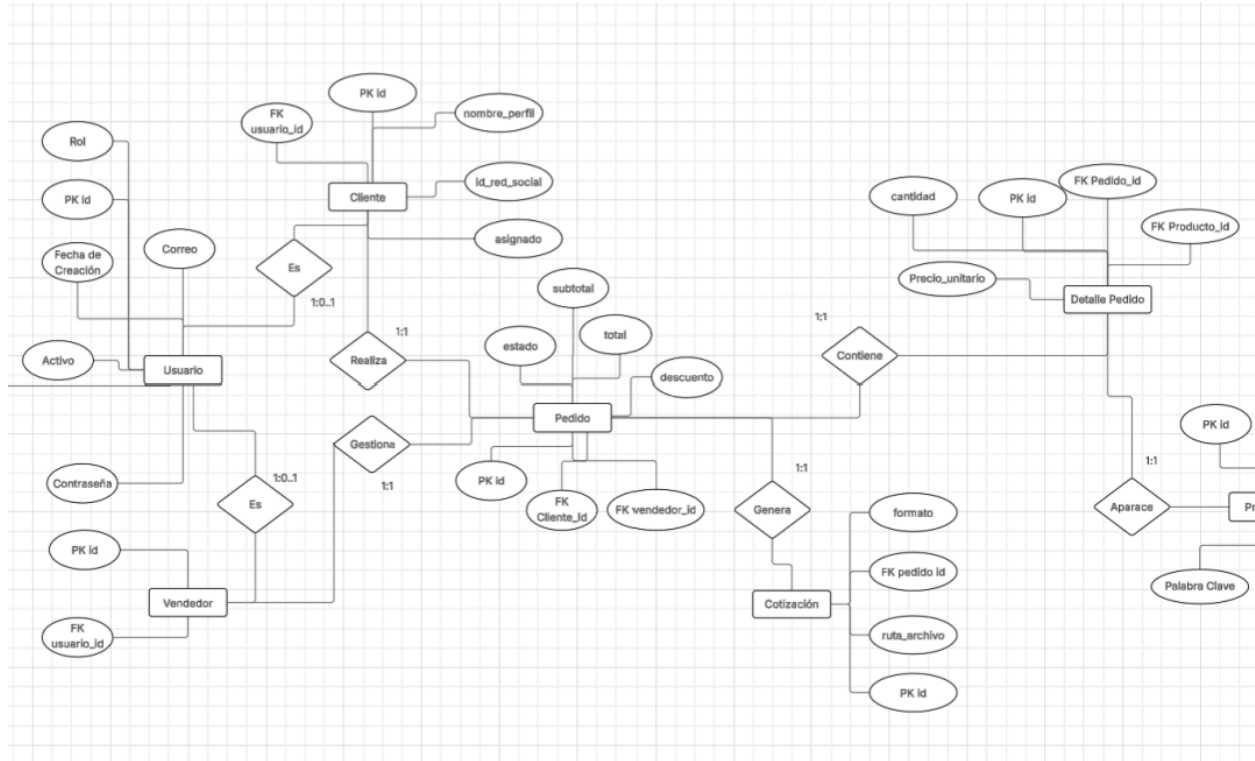


Figura 2. Primer versión del MER

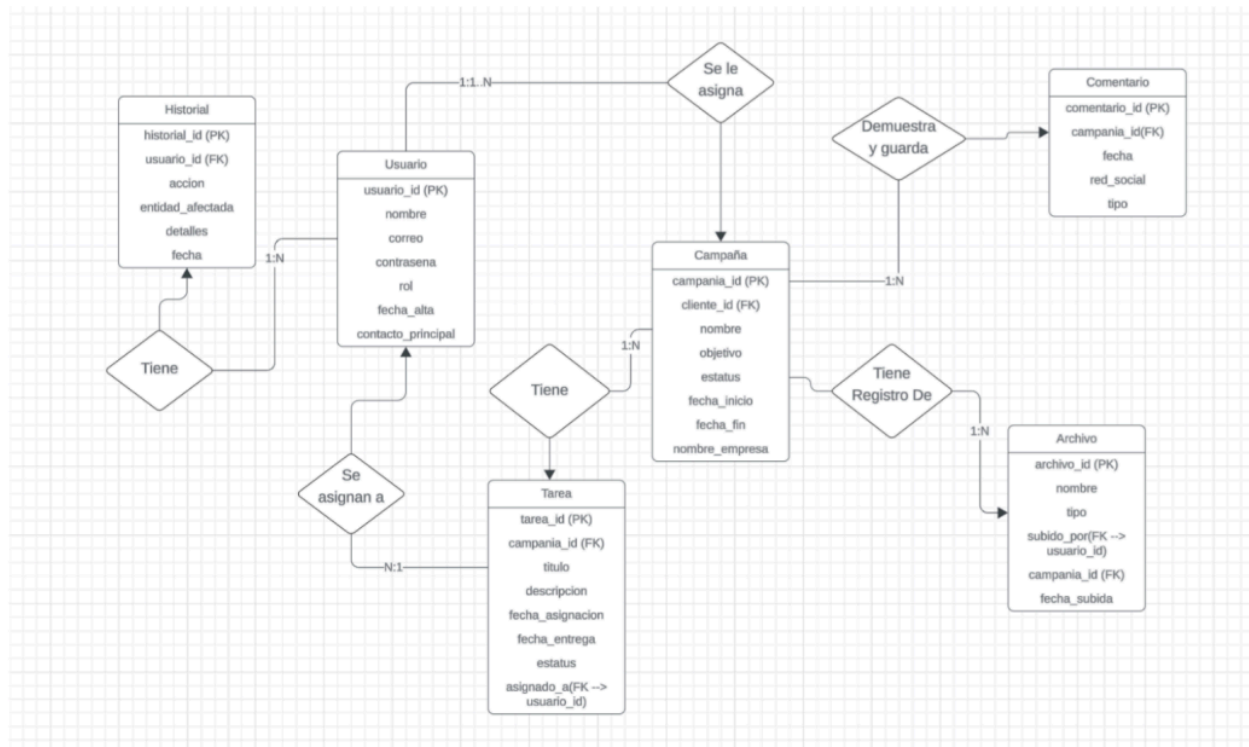


Figura 3. Segunda versión del MER

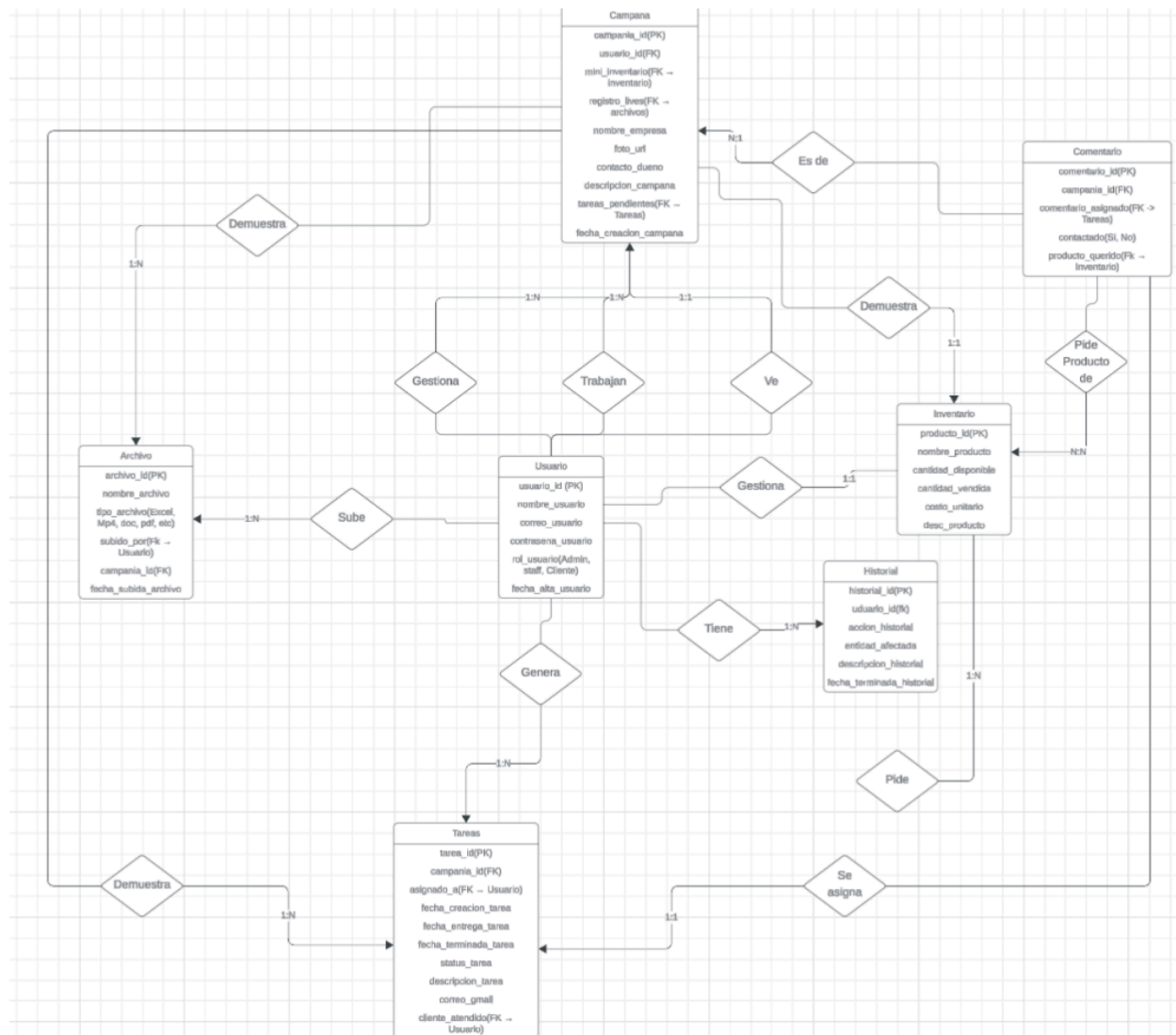


Figura 4. Versión 3 de MER

En las actualizaciones yo implementé el backend basándome en las necesidades del socio formador, como la implementación en campañas para subir los objetos de los ítems por medio de un archivo .csv

[illegible]

Figura 5. Excel de los ítems de socioformador

Basado en las necesidades del sistema y las demandas del curso, mis compañeros crearon bosquejos a manera de modelación del sistema a priori, desde esa premisa implementé backend del sistema y mis compañeros implementaron las vistas (frontend).

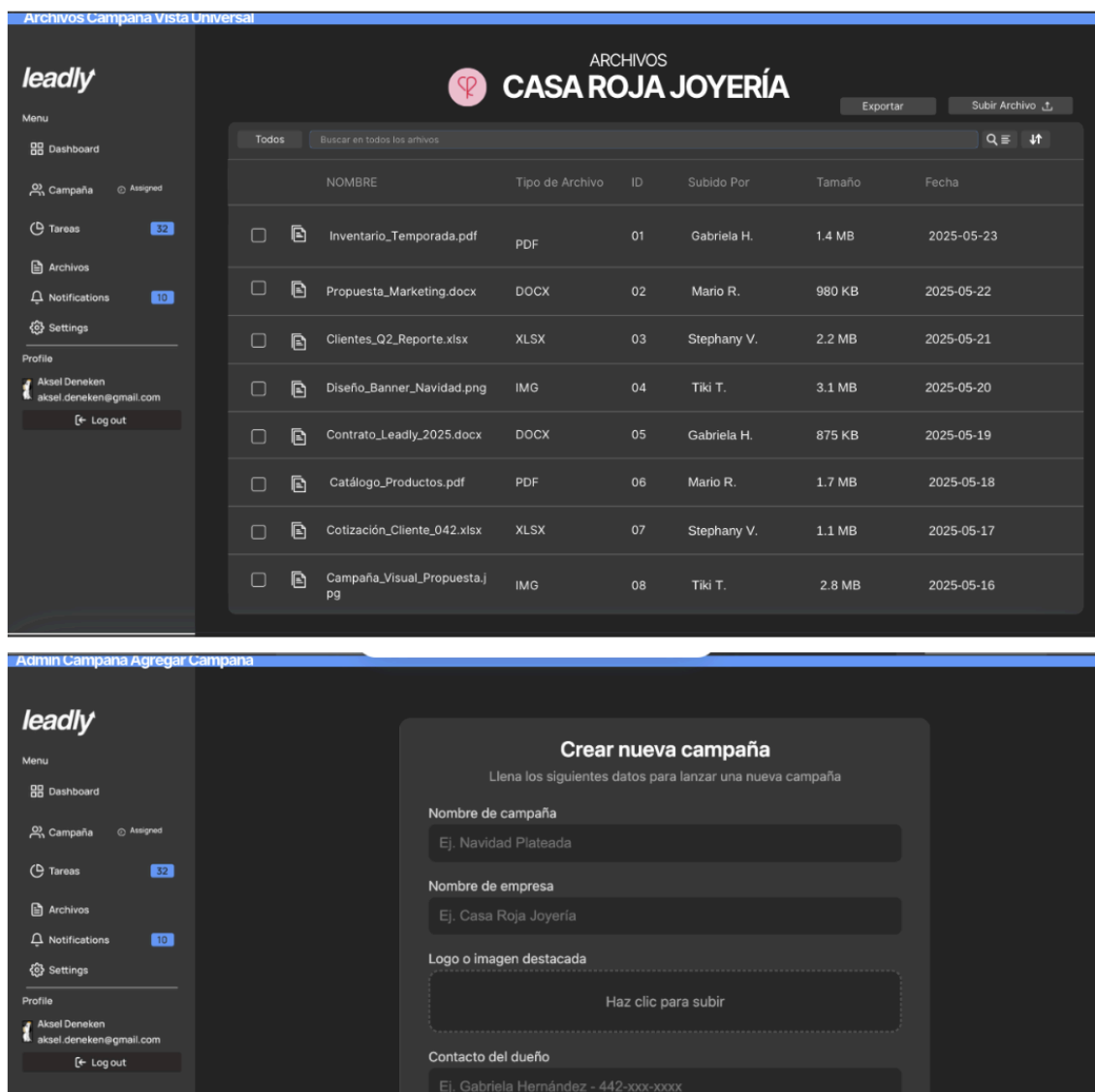


Figura 6. Bosquejos del sistema

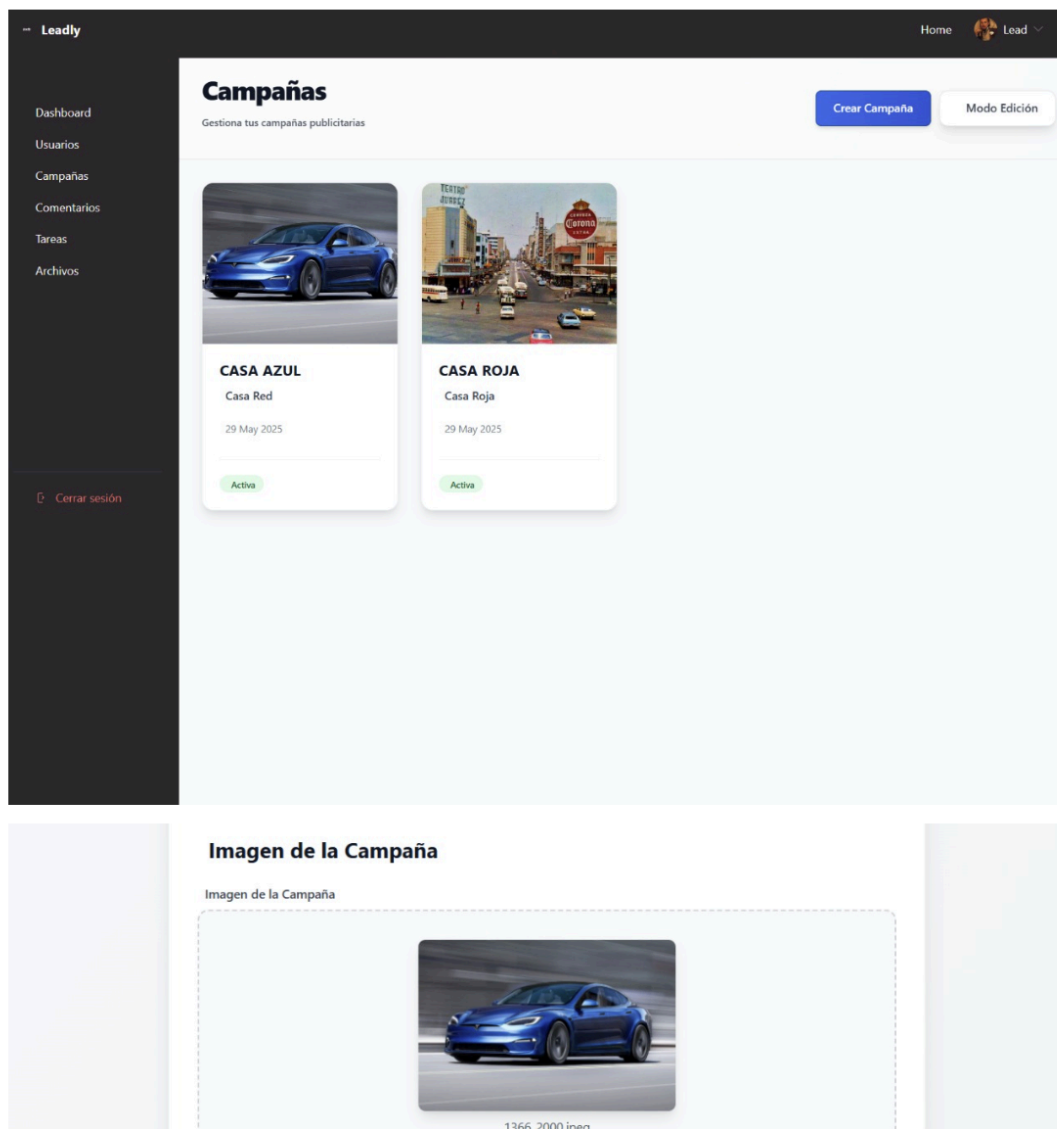


Figura 7. Visualización del sistema en etapa beta