

CINE
FUSION

¿Quiénes somos?



Alain Lozano García
Database Manager
API Assistant
Video Producer



Pablo Garrido Rubio
Api Manager
Database Assistant



Germán Romo Orta
Api Manager
Database Assistant



Jan Vizcaíno Boixadós
Web Developer
Domain Admin
Coordination Manager

OBJETIVO

Diseñar y desarrollar todos los
sistemas necesarios para gestionar
un cine

MÉTODO

trabajo
colaborativo

->

integración de diferentes
módulos y áreas.

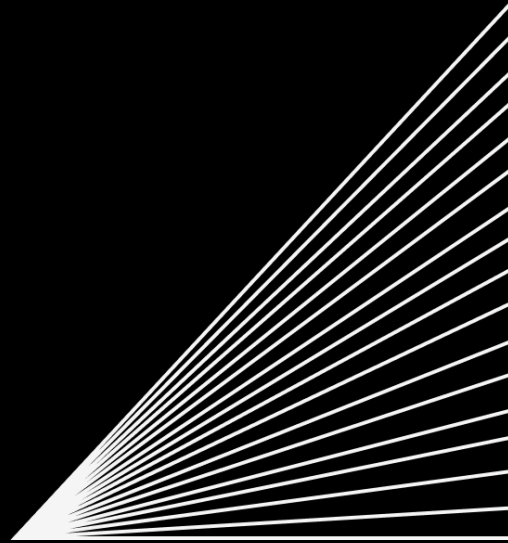
¿Cine?

Qué se necesita

Lógica
funcional



Infraestructura
Digital



UI/UX



Gestión de
información

Lógica funcional

Arquitectura por capas

Operaciones CRUD



Lógica funcional

Arquitectura por capas

- **Controller:** gestiona las peticiones HTTP
- **Service:** contiene la lógica de negocio
- **Repository:** se comunica con la base de datos

Operaciones CRUD

- **Model/Entity:** define la estructura de los datos
- **DTOs:** transferimos solo la información necesaria



Lógica funcional

Arquitectura por capas

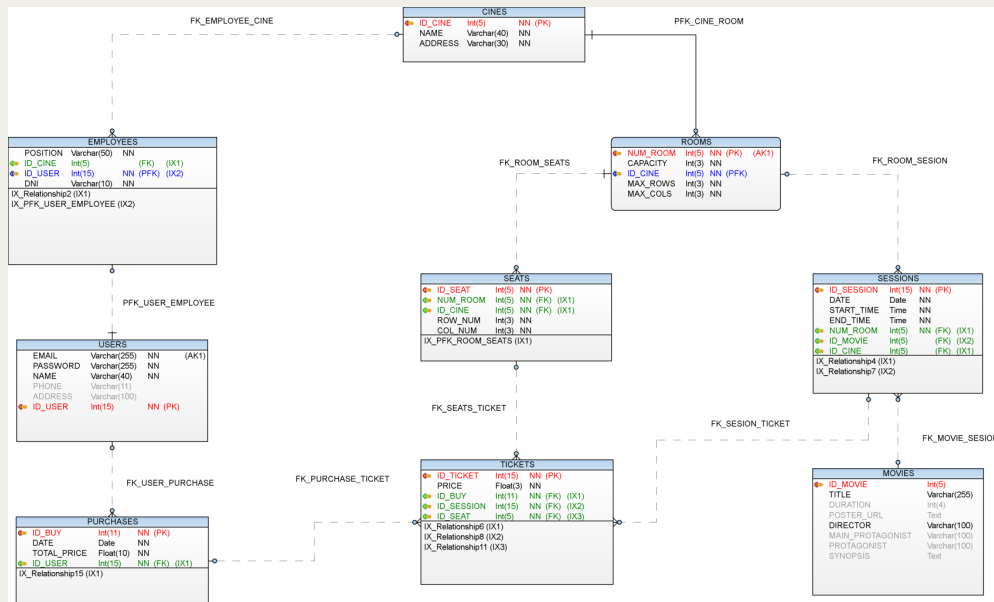
- **Create:** Permite crear **nuevos registros** en la base de datos. Su anotación en SpringBoot es **@PostMapping**.
- **Read:** Permiten **consultar los datos existentes** en la base de datos. Para devolver los registros se usa **@GetMapping**.

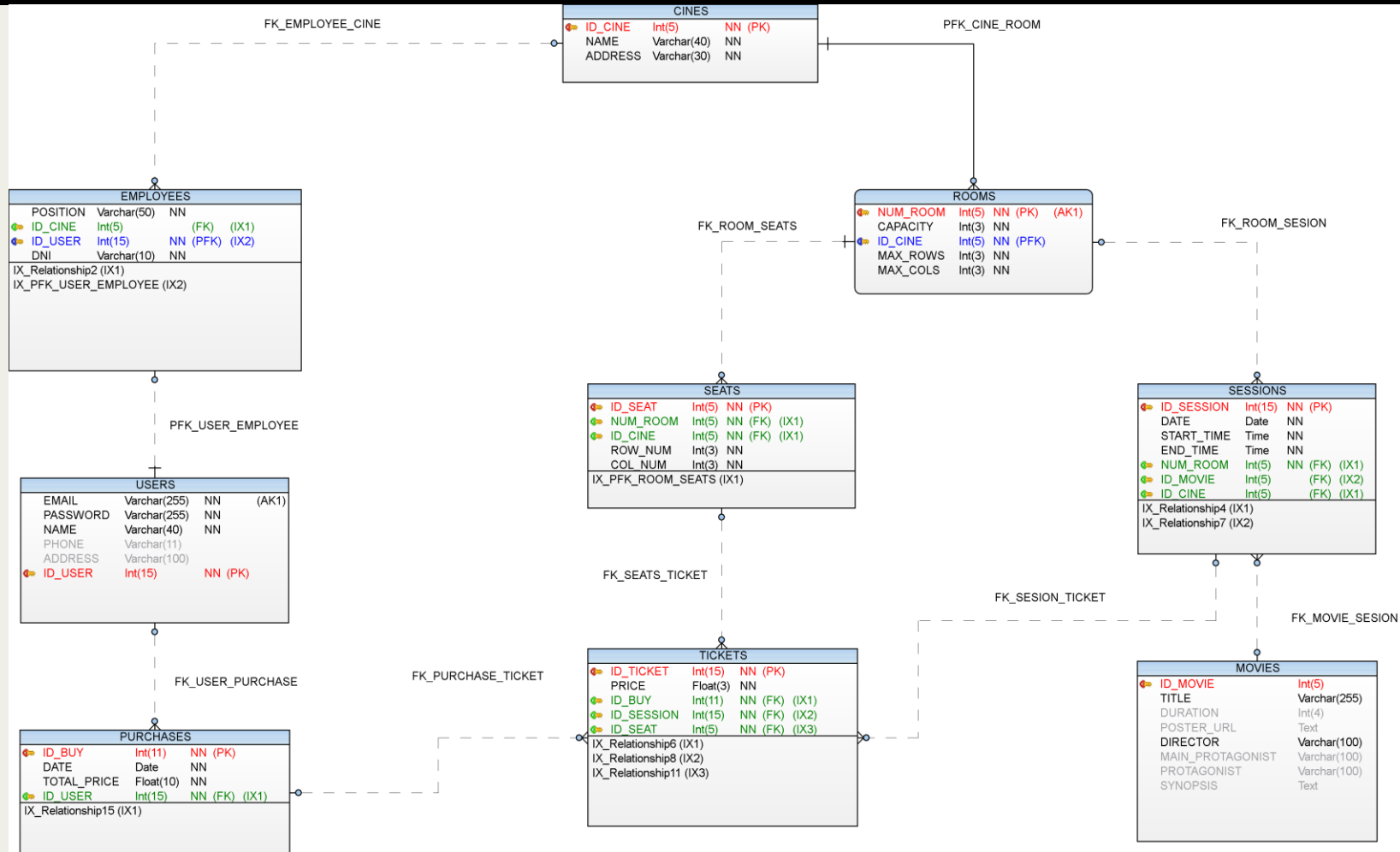
Operaciones CRUD

- **Update:** Se encargan de **actualizar los datos existentes**. Se implementa con la anotación de **@PutMapping**.
- **Delete:** permiten **eliminar registro** de la base de datos. Su anotación **@DeleteMapping**.



Gestión de información





UI/UX



TailwindCSS

AlpineJS

JavaScript

UI/UX



TailwindCSS

Facilita crear interfaces **modernas y consistentes** rápidamente con clases utilitarias, mejorando la **estética y usabilidad** sin escribir **CSS** desde cero.

AlpineJS

JavaScript

UI/UX



TailwindCSS

AlpineJS

JavaScript

UI/UX



TailwindCSS

AlpineJS

Añade interactividad ligera directamente en el HTML, ideal para comportamientos simples **sin cargar frameworks pesados**.

JavaScript

UI/UX



TailwindCSS

AlpineJS

JavaScript

UI/UX



TailwindCSS

AlpineJS

JavaScript

Permite controlar la lógica de interacción del usuario, validaciones, animaciones y **manipulación del DOM**, siendo la base de la experiencia dinámica en la web.

Infraestructura Digital



Servidor de dominio



Apache sobre AlmaLinux

Infraestructura Digital



Servidor de dominio

Hemos usado **Windows Server** porque facilita la gestión de usuarios, departamentos y permisos mediante **Active Directory**, ideal para un entorno corporativo como el de un cine.

Apache sobre AlmaLinux

Infraestructura Digital



Servidor de dominio



Apache sobre AlmaLinux

Infraestructura Digital



Servidor de dominio

Apache sobre AlmaLinux

Hemos utilizado **Apache** para montar el **servidor web** porque es confiable, fácil de configurar y ampliamente compatible con tecnologías **UI/UX**. Instalado en **AlmaLinux**, un sistema estable y gratuito con soporte empresarial, ideal para entornos de producción.

Gestión del proyecto



IntelliJ Code-With-Me

GitHub

Gestión del proyecto



IntelliJ Code-With-Me

Esta herramienta nos permitió trabajar de forma colaborativa en el mismo proyecto, compartiendo sesión entre **varios compañeros**. Así podíamos programar, revisar el código y resolver errores en **tiempo real**, como si estuviéramos todos en el mismo ordenador.

GitHub

Gestión del proyecto



IntelliJ Code-With-Me

GitHub

Gestión del proyecto

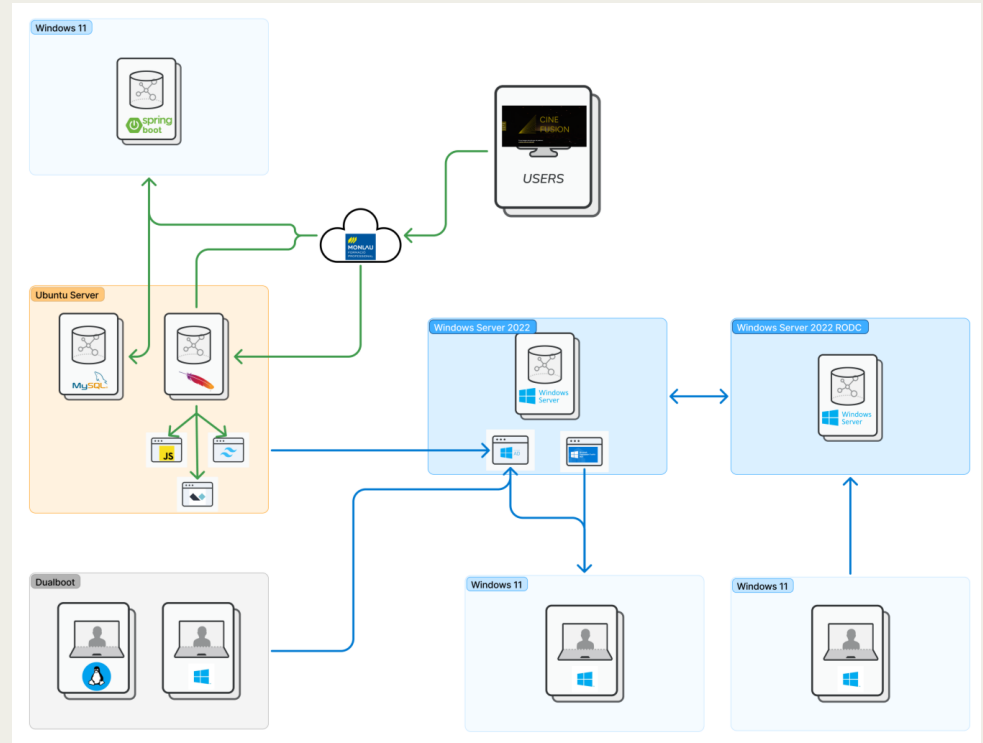


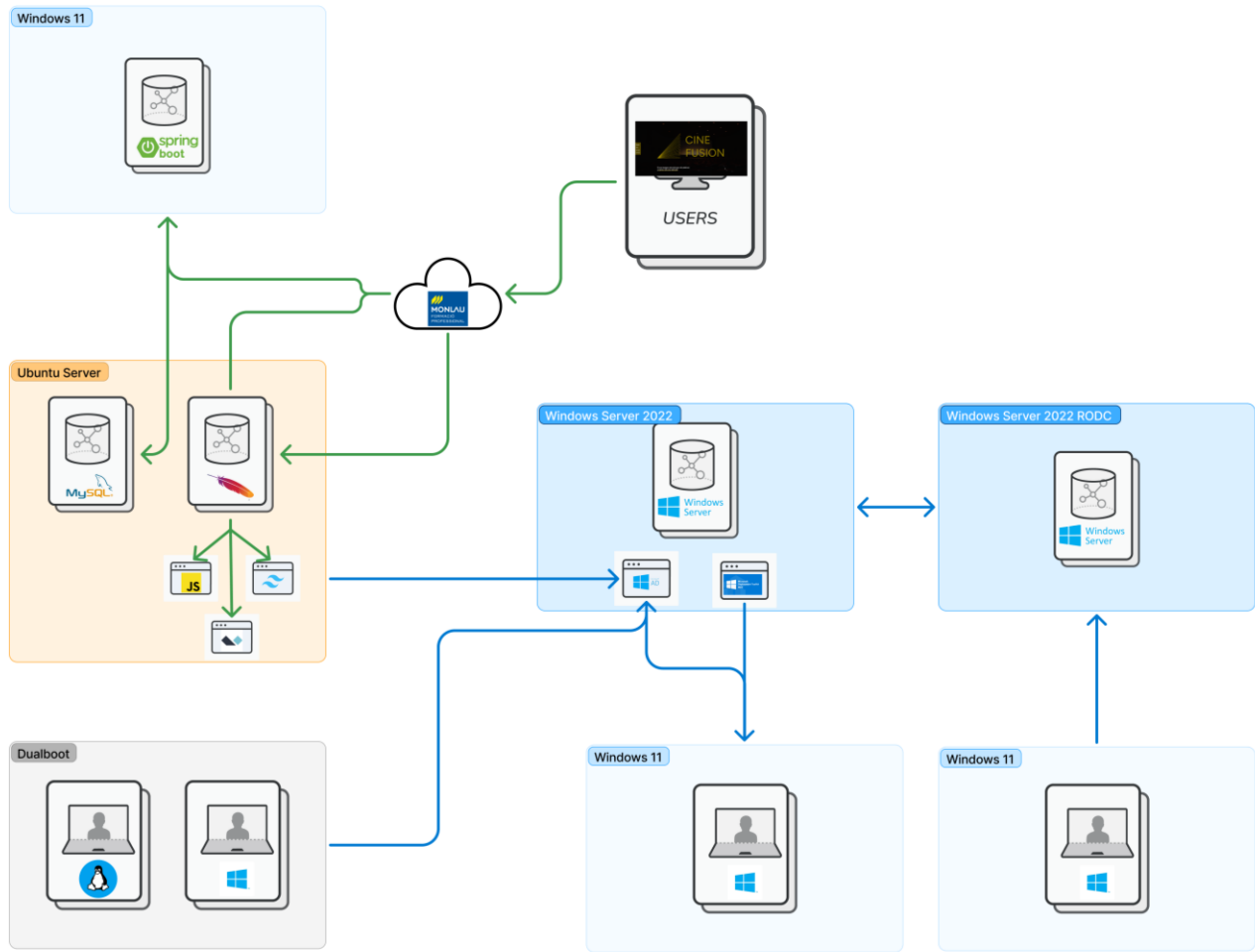
IntelliJ Code-With-Me

GitHub

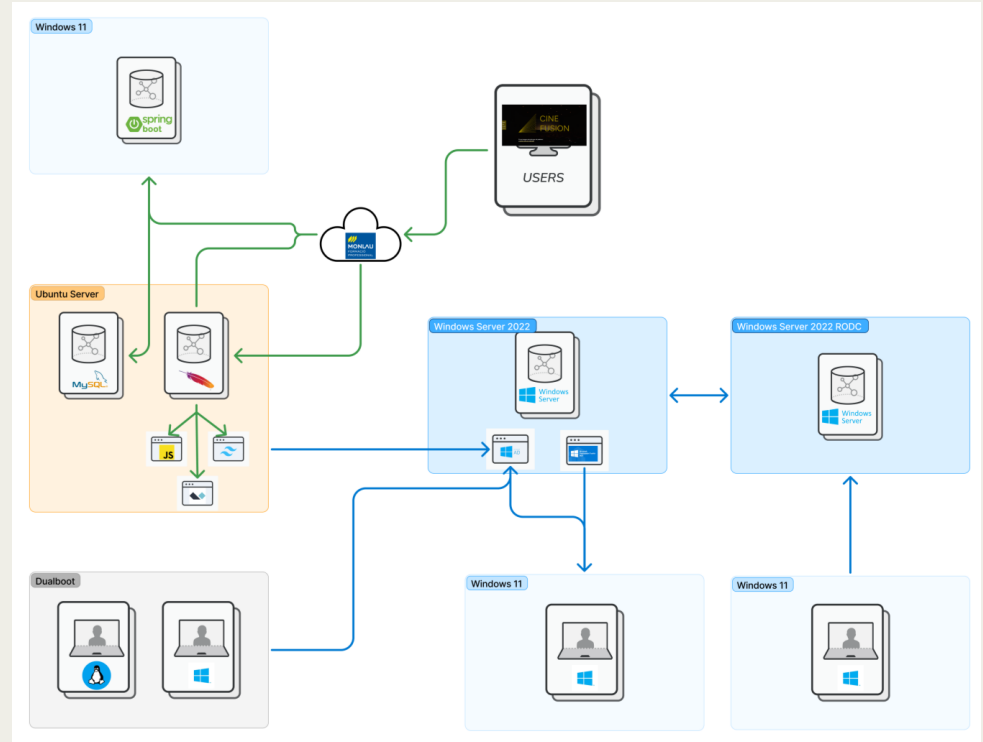
Usamos GitHub como plataforma central para alojar el **repositorio** del proyecto. Gracias al control de versiones, pudimos trabajar en **distintas ramas**, **fusionar cambios** sin pisarnos y mantener un historial claro y ordenado del desarrollo.

Esquema final del proyecto

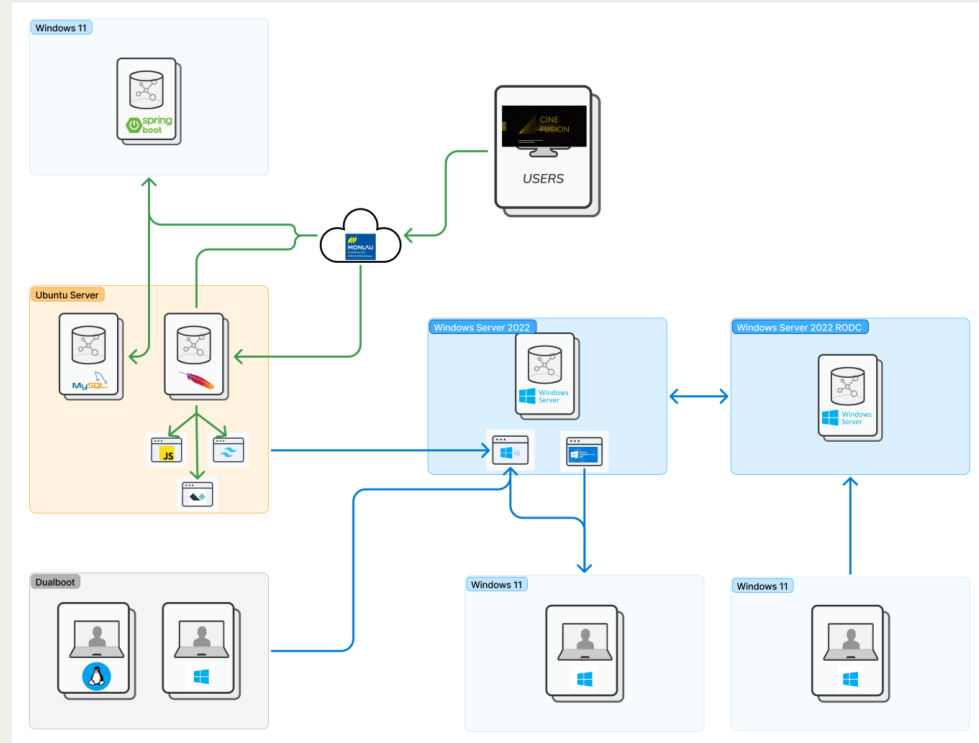




Esquema final del proyecto



Esquema ~~final~~ del proyecto



Esquema final del proyecto



<http://68.221.173.98/>

Dificultades y aprendizajes

Conclusión y valoración

i***?