

# Proyecto de desarrollo web: virtual**Class**

**Manuel Sánchez Cañadas**

Grado en Ingeniería Informática (2do semestre, curso 2019-2010)

Responsable de la asignatura: **Gregori Robles Martínez**

# Índice

- ▶ Justificación del trabajo y objetivos
- ▶ Tecnologías utilizadas
- ▶ Planificación temporal
- ▶ Requisitos del proyecto
- ▶ Diseño del proyecto
- ▶ Desarrollo del proyecto
- ▶ Fase de pruebas
- ▶ Conclusiones

# Justificación del trabajo y objetivos

- ▶ VirtualClass es una aplicación web de clases virtuales.
- ▶ Plataforma web que permita a profesores particulares conectarse mediante videoconferencia con sus alumnos, así como proporcionar una herramienta de dibujo (para explicaciones complementarias).

## OBJETIVOS DEL TRABAJO

Aplicación web con dos tipos de roles: profesor y alumno.

- ▶ El *profesor* puede crear sus clases, indicando las materias de las cuales puede ofrecer formación.
- ▶ El *alumno* estará suscrito a un profesor o varios de ellos.

En la plataforma existirá un chat entre el profesor y el alumno para comunicarse.

La clase dispondrá de los siguientes servicios:

- ▶ Videoconferencia
- ▶ Chat
- ▶ *Canvas* para dibujar

# Tecnologías utilizadas

Servidor	Servicios http	Base de datos	Cont versiones	Depliegue
Linux	Nginx	MongoDB Redis	Git	Docker

Backend	Frontend	Comunicación	Videoconferencia
Nodejs Express Socket.io WebSocket Mongoose	HTML CSS JavaScript Vue VueX Vue-router Axios	Rest WebSocket RabbitMQ	WebRTC

# Planificación temporal

Sprints (10 días aproximadamente cada uno):

- ▶ Planificar cada servicio o componente del backend (user, class, chat, wsservice ...): casos de uso, modelo de datos, modelado de las apis (rest, websocket, rabbitmq), etc.
- ▶ Programar librerías o microframework común a cada servicio, preparar contenedores docker, preparar una *api gateway* para llamar a los servicios y preparar la arquitectura del servidor.
- ▶ Programar los servicios iniciales (user, class) con sus respectivos *test*.
- ▶ Iniciar el desarrollo del front con vue, con sus llamadas a los servicios básicos
- ▶ Programar servicio de chat y el de websocket. Desarrollar las vistas frontend del nuevo servicio.
- ▶ Programar el servicio de clase con la videollamada (webrtc) y sincronizar con el servicio de websocket.
- ▶ Programar servicios que falten (canvas) o extras (calendario, documentos).

# Requisitos del proyecto

Antes de empezar a implementar la aplicación, se establecieron una serie de requisitos\* para especificar su alcance para cada una de las siguientes funcionalidades:

- ▶ Crear cuenta de usuario
- ▶ Conexión a la aplicación
- ▶ Perfil de usuario
- ▶ Buscar profesor/alumno
- ▶ Videollamada
- ▶ Chat
- ▶ Documentos

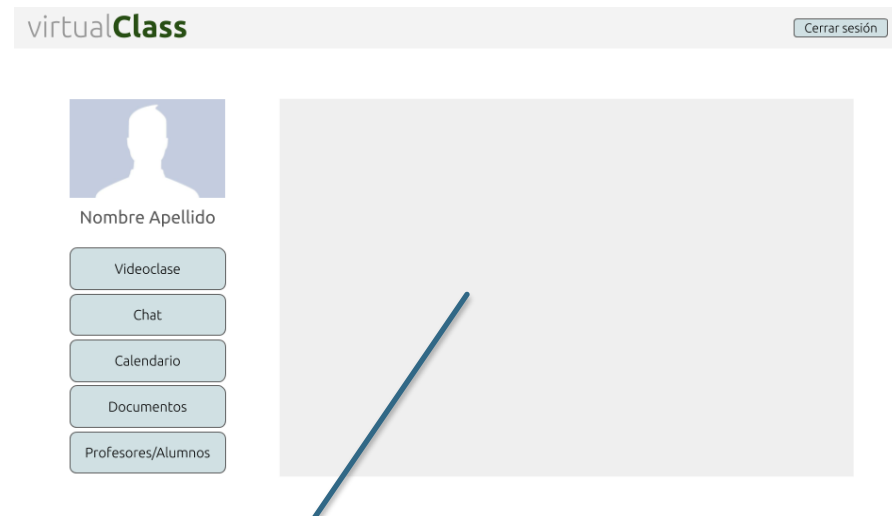
*\* Los requisitos quedan detallados en la memoria del Trabajo de Final de Grado*

# Diseño del proyecto

- ▶ Diseño del logo **virtualClass**
- ▶ Diseño de la web principal



- ▶ Diseño de la plataforma

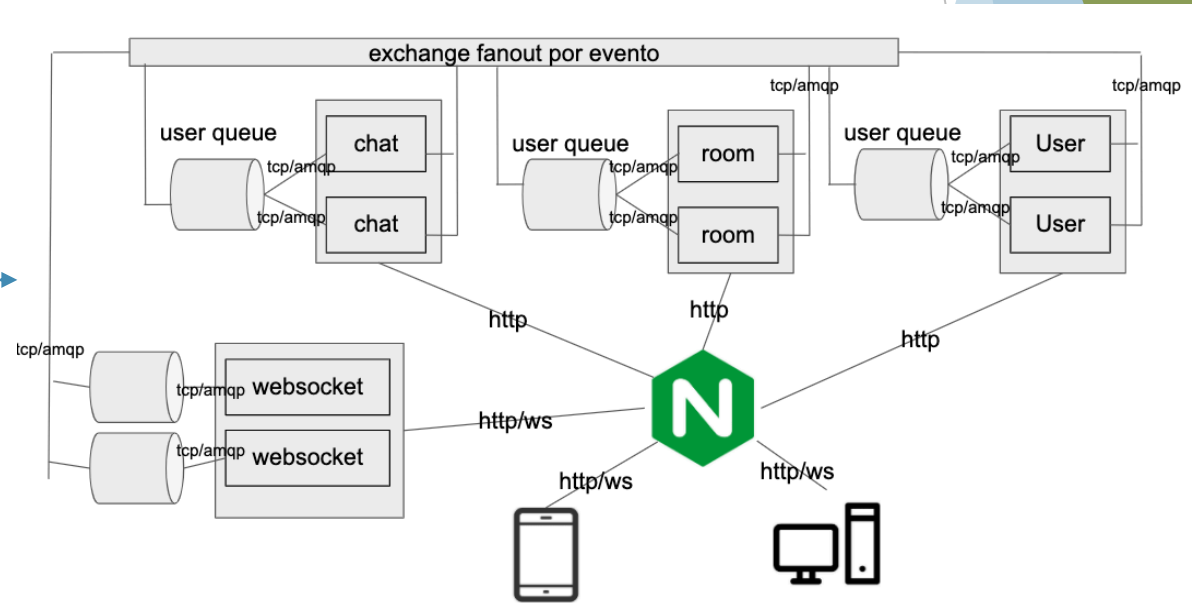
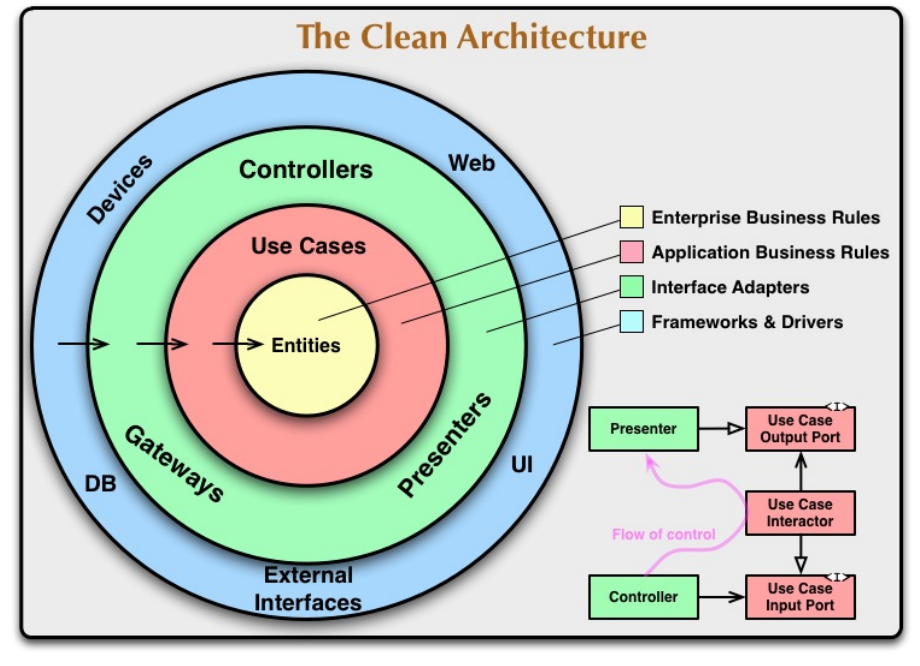


*Interfaz donde aparece contenido según aquella función que haya seleccionado el usuario en el menú de la izquierda*

# Desarrollo del proyecto

## ▶ Arquitectura del sistema

- ▶ Diseño de microservicio →
- ▶ Comunicación entre servicios
- ▶ Comunicación cliente-servidor
  - ▶ Sincronica
  - ▶ Asíncronica
- ▶ Diseño de la arquitectura →
- ▶ Despliegue de la arquitectura





# Desarrollo del proyecto

## ▶ Servicios

- ▶ Usuario
- ▶ Videollamada y Canvas
- ▶ Chat
- ▶ WebSocket
- ▶ Documentos
- ▶ Frontend

# Fase de pruebas

- ▶ *Tests* unitarios en las entidades de dominio y aplicación.
- ▶ Complejidad en al testear los eventos asíncronos.
- ▶ Aplicación *postman* para testear las *apis*.
  
- ▶ Análisis de diferentes navegadores y dispositivos (ordenadores, tabletas y teléfonos móviles).
  
- ▶ Pruebas realizadas por una tercera persona:
  - ▶ Registro ✓
  - ▶ Acceso ✓
  - ▶ Videollamada ✓
  - ▶ Chat ✓
  - ▶ Documentos ✗

# Conclusiones

- ▶ Se ha cumplido el objetivo principal: diseñar y desarrollar un sitio web que permitiese la realización de videoclases entre profesores y alumnos.
- ▶ Desarrollo de la plataforma complejo, debido a la complejidad de la infraestructura asíncrona y distribuida.
- ▶ No se han podido implementar las funcionalidades de Canvas, Calendario y Documentos.
- ▶ La dificultad que supone afrontar todas las fases de un proyecto un solo individuo.
- ▶ Diseño: experiencia óptima y sencilla, acorde al planteado en un principio.
- ▶ Gran potencial de desarrollo.