



Universidad  
Europea  
del Atlántico

# “Programación Web I”

---

Memoria práctica 1

Sergio Velarde Álvarez

17 - 12 - 2024

Febrero a Junio  
Santander, España

## MEMORIA DE PRACTICA

### 1. INTRODUCCIÓN

En esta práctica se nos ha solicitado desarrollar una aplicación web que implemente una interfaz de usuario (frontend) y un servidor (backend), utilizando tecnologías modernas como React y Node.js. El objetivo principal de la práctica es construir un sistema de gestión de usuarios para una residencia, en el que se puedan realizar varias acciones como el registro de nuevos usuarios, el inicio de sesión con JWT (JSON Web Tokens), la visualización de los usuarios registrados, la búsqueda de usuarios por nombre y la edición o eliminación de los mismos. Además, se requiere la creación de una base de datos donde se almacenen los datos de los usuarios y sus respectivas relaciones, como la asignación de habitaciones.

El backend debe ser responsable de gestionar las solicitudes relacionadas con los usuarios, incluyendo la autenticación, y debe estar protegido con rutas que solo sean accesibles si el usuario está autenticado. Por otro lado, el frontend debe ofrecer una interfaz sencilla pero funcional, donde se pueda interactuar con el sistema a través de formularios, tablas y filtros, mostrando la información de manera clara y permitiendo la búsqueda rápida de usuarios. La implementación de este sistema debe garantizar una experiencia de usuario intuitiva, con un diseño limpio y bien estructurado.

### Tecnologías Aplicadas:

#### Frontend:

1. **React:**
  - **Librería principal** para construir interfaces de usuario basadas en componentes. Usada para crear los formularios de login y registro, y manejar el estado del formulario.
  - Utilizado con **JSX** para crear estructuras HTML dentro del código JavaScript.
2. **TypeScript:**
  - **Superset de JavaScript** que añade tipado estático y otras funcionalidades para ayudar a mejorar la calidad del código, como el manejo de los estados en el formulario.
3. **React Router:**

- **Librería** que permite la navegación entre diferentes páginas en una aplicación de una sola página (SPA). En este caso, se usa para gestionar las rutas entre el login (`/login`) y el registro (`/sign-up`).
- 4. **Bootstrap** (a través de **React-Bootstrap** o clases CSS personalizadas):
  - **Framework de diseño** basado en CSS para crear interfaces de usuario responsivas y modernas. En este proyecto, se usa para estructurar y dar estilo a los formularios (por ejemplo, la clase `form-outline`).
  - **Sistema de rejilla** (Grid system) de Bootstrap se usa para organizar el contenido en columnas (`col-md-6`, `col-lg-12`, etc.).
- 5. **CSS/SCSS**:
  - **Estilos personalizados** para los componentes del formulario y el diseño general de la página. El archivo `Login.css` proporciona clases personalizadas para dar estilo a los formularios, los botones, y los colores de fondo (por ejemplo, la clase `.gradient-custom-2`).
  - **Gradientes y fondos personalizados** se utilizan para mejorar la apariencia visual.

## Backend:

1. **Node.js con Express.js**:
  - **Node.js** es el entorno de ejecución de JavaScript en el servidor, y **Express.js** es el framework que se usa para crear y manejar las rutas del servidor de manera sencilla.
2. **JWT (JSON Web Tokens)**:
  - Se utiliza para manejar la autenticación de usuarios. Cuando un usuario inicia sesión correctamente, el backend genera un token JWT que se devuelve al frontend. Este token es usado por el cliente para autenticar futuras solicitudes.
3. **bcrypt.js**:
  - Se utiliza para cifrar las contraseñas de los usuarios antes de guardarlas en la base de datos.

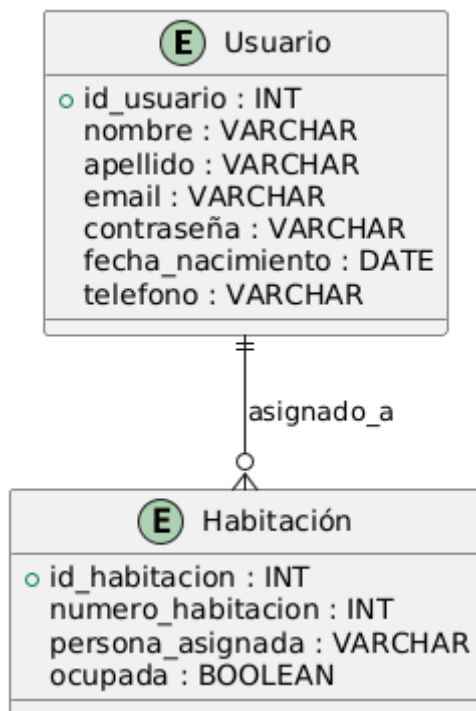
## Base de Datos:

1. **MongoDB:** Una base de datos NoSQL basada en documentos, puedes almacenar los datos de los usuarios en un formato flexible en lugar de tablas estrictas.

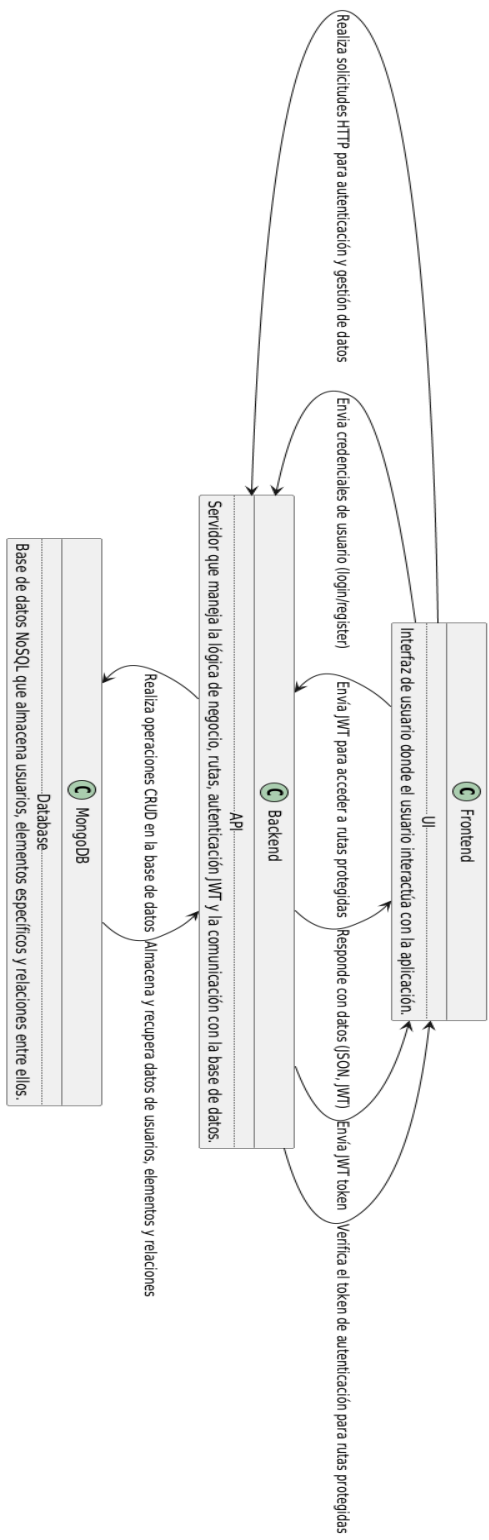
## 2. Diagramas

---

### Base de Datos:



### Diagrama de la arquitectura:



### 3. Dificultades encontradas

---

#### 1. Diseño y Layout del Formulario de Registro (Sign Up)

Uno de los problemas iniciales fue la disposición del formulario en la página de registro. El formulario no ocupaba todo el ancho de la pantalla, lo que resultaba en una interfaz visualmente poco atractiva y no optimizada para pantallas grandes. Al principio, el formulario se veía estrecho, con un espacio en blanco innecesario alrededor, lo que afectaba la experiencia de usuario.

Para resolver esto, se intentaron varios enfoques, como ajustar el contenedor, las clases de Bootstrap y las propiedades de CSS, para garantizar que el formulario se extendiera de manera adecuada. Sin embargo, los cambios en el código de diseño, como la manipulación de los contenedores o la modificación de las clases de diseño de Bootstrap, no tuvieron el efecto esperado en algunos casos, lo que requería pruebas adicionales y ajustes finos. Al final, fue necesario modificar el ancho y la altura del formulario para que se adaptara mejor a diferentes tamaños de pantalla, haciendo el formulario más amplio y centrado.

#### 2. Implementación de un Fondo de Imagen en el Formulario

Otro desafío fue la implementación de una imagen de fondo en la página de registro. Al intentar agregar una imagen de fondo que cubriera todo el cuerpo del formulario y no solo una parte del área de visualización, el código no funcionaba correctamente, ya que la imagen no se ajustaba al fondo de la página de manera adecuada, creando un efecto visual no deseado. En lugar de cubrir el área de la pantalla completa, la imagen se quedaba como un solo bloque detrás de los elementos, resultando en un diseño inconsistente.

Para solucionar esto, se necesitaron ajustes a las propiedades `background-image` y `background-size` de CSS, así como pruebas con diferentes valores para asegurar que la imagen se ajustara correctamente en diferentes dispositivos. Aunque se consiguió poner la imagen como fondo detrás de todos los elementos, la adaptación de la imagen a diferentes resoluciones de pantalla requirió un trabajo extra en la configuración de los estilos.

#### 3. Relación entre el Usuario y la Habitación

En el backend, uno de los principales problemas fue el diseño de la base de datos para gestionar la relación entre el usuario y la habitación. Se requería una estructura eficiente para almacenar la información del usuario y asignar habitaciones. Inicialmente, la relación entre el usuario y la habitación no estaba completamente

clara en el modelo, lo que llevó a una revisión y ajustes en el diseño del esquema de base de datos.

El modelo final, que implicaba que cada usuario tuviera una habitación asignada, requería una relación adecuada entre las tablas de usuarios y habitaciones, con atributos como `persona_asignada` y `ocupada` en la tabla de habitaciones. Sin embargo, los desafíos surgieron a la hora de definir correctamente las claves foráneas y las restricciones para mantener la integridad de los datos y asegurar que las habitaciones no pudieran ser asignadas a múltiples usuarios simultáneamente.

#### **4. Problemas de Responsividad y Experiencia de Usuario**

La responsividad fue otro desafío importante. A pesar de usar un framework como Bootstrap, hubo problemas al intentar que los formularios y otros elementos de la interfaz de usuario se adaptaran correctamente a dispositivos móviles. Algunos de los elementos no se ajustaban bien a pantallas más pequeñas, lo que causaba una mala experiencia de usuario.

Se intentaron ajustar las columnas y las filas del formulario para que se redistribuyeran adecuadamente según el tamaño de la pantalla, pero hubo problemas para evitar barras de desplazamiento horizontales en dispositivos más pequeños. Las pruebas de visualización en diferentes dispositivos y la implementación de media queries fueron necesarias para solucionar estos inconvenientes y asegurar que la página fuera completamente funcional en diferentes resoluciones.

#### **5. Problemas con la Autenticación y Validación de Datos**

Otro reto fue la implementación de una autenticación efectiva y la validación de datos en el formulario de registro. Aunque se utilizó React con un estado gestionado para almacenar la información del formulario, la validación de los campos del formulario no estaba completamente configurada al principio. Esto incluía la verificación de campos obligatorios, formatos de correo electrónico y contraseñas, y la confirmación de que los datos ingresados fueran válidos antes de enviar el formulario.

En resumen, los principales problemas que encontramos en el desarrollo de la página de registro y en la interacción con el backend fueron principalmente relacionados con el diseño responsivo, la integración de elementos visuales como el fondo de imagen, el establecimiento de relaciones entre las tablas de base de datos y la validación de datos en el frontend. Estos desafíos fueron solucionados mediante un enfoque iterativo, probando y ajustando continuamente el diseño y el flujo de trabajo del sistema.



Universidad  
Europea  
del Atlántico