

# PROYECTO

## Desarrollo Front-End

El **front-end** de una aplicación web es la parte que interactúa directamente con el usuario. Este es el diseño y la funcionalidad visible de la página web. En esta fase se debe crear la interfaz de usuario (UI) y garantizar que la experiencia de usuario (UX) sea atractiva y fácil de usar.

### Tecnologías comúnmente utilizadas:

- **HTML:** Para estructurar el contenido y las páginas web.
- **CSS:** Para diseñar la apariencia de la página (estilos, colores, disposición de elementos).
- **JavaScript:** Para añadir interactividad y dinamismo a las páginas (por ejemplo, formularios interactivos, validaciones, manipulaciones del DOM).
- **Frameworks/librerías recomendadas:**
  - React, Vue.js o Angular: Para crear interfaces dinámicas y reutilizables, si el proyecto lo requiere.
  - Bootstrap o Tailwind CSS: Para un diseño responsivo y predefinido.

### Actividades en el desarrollo Front-End:

- **Diseño de la interfaz de usuario (UI):** Crear una interfaz limpia, coherente con la identidad del proyecto y fácil de usar.
  - Estructura de navegación.
  - Páginas principales (inicio, contacto, perfil de usuario, etc.).
  - Formularios (registro, inicio de sesión, etc.).
- **Implementación de responsividad:** Asegurarse de que la aplicación sea accesible y se vea bien en dispositivos de diferentes tamaños (móviles, tabletas y escritorio).
- **Interactividad y validación de formularios:** Utilización de JavaScript para gestionar eventos, validaciones en tiempo real e interacciones dinámicas con el usuario.
- **Consumo de APIs:** Si el proyecto requiere, se debe hacer uso de APIs para interactuar con el back-end, ya sea para obtener datos o enviar información (por ejemplo, obtener productos de una base de datos, realizar una compra, etc.).

## Desarrollo Back-End

El **back-end** de una aplicación web es la parte que maneja la lógica de negocio y el almacenamiento de datos, y se ejecuta en el servidor. En esta fase hay que crear la arquitectura y la funcionalidad del servidor que soporta la aplicación web.

### Tecnologías comúnmente utilizadas:

- Lenguajes de programación:
  - Node.js (JavaScript), Python, PHP, Ruby, Java, C#, entre otros.
- Bases de datos:
  - Relacionales: MySQL, PostgreSQL (si el proyecto requiere datos estructurados y relaciones entre tablas).
  - No Relacionales: MongoDB (si se requiere flexibilidad en el modelo de datos).
- Frameworks/librerías recomendadas:
  - Express.js (Node.js): Para crear APIs y manejar rutas.
  - Django (Python): Para proyectos rápidos con funcionalidad completa (si se usa Python).
  - Spring Boot (Java): Para proyectos basados en Java que necesiten robustez y escalabilidad.
  - Laravel (PHP): Para un desarrollo rápido de aplicaciones web con PHP.

### Actividades en el desarrollo Back-End:

- Desarrollo de la lógica de negocio: Crear los procesos que gestionan la interacción con los datos. Por ejemplo:
  - Manejo de usuarios: autenticación, autorización, roles.
  - Gestión de productos, pedidos o cualquier otra funcionalidad central del sistema.
  - Implementación de la lógica de las acciones que el usuario realiza a través del front-end.
- Base de datos y modelo de datos:
  - Diseño del esquema de la base de datos (tablas, relaciones entre ellas, tipos de datos).

- Implementación de las operaciones CRUD (crear, leer, actualizar, eliminar) sobre la base de datos.
  - Conexión de la base de datos con el back-end y creación de consultas.
- Implementación de la API:
  - Crear rutas y controladores para gestionar las peticiones del front-end.
  - Asegurar que las rutas sean seguras y eficientes.
  - Validación de datos entrantes y salientes.
  - Implementación de autenticación (por ejemplo, usando JWT, OAuth, etc.).
- Pruebas del back-end: Realizar pruebas unitarias y de integración para asegurarse de que las funcionalidades del servidor funcionan correctamente. Por ejemplo:
  - Pruebas de las rutas de la API.
  - Validación de la interacción entre el servidor y la base de datos.

## Integración del Front-End y Back-End

Una vez que el front-end y el back-end estén desarrollados, hay que integrarlos de forma que el front-end pueda interactuar con las funcionalidades del servidor a través de peticiones HTTP (GET, POST, PUT, DELETE), enviando y recibiendo datos del servidor. Esta integración es esencial para hacer que la aplicación sea funcional.

### Pasos para la integración:

1. Conectar el front-end con el back-end: Asegurarse de que las solicitudes del front-end lleguen correctamente al back-end y que los datos sean procesados y enviados de vuelta.
2. Manejo de errores y respuestas: Implementar el manejo adecuado de errores (por ejemplo, respuestas de error si algo va mal en el servidor).
3. Asegurar que las interacciones sean fluidas: Comprobar que el front-end se actualice de manera dinámica en función de las respuestas del back-end, manteniendo una experiencia de usuario fluida.