

# INFORME DE BUENAS PRACTICAS PARA DESARROLLO FRONTEND

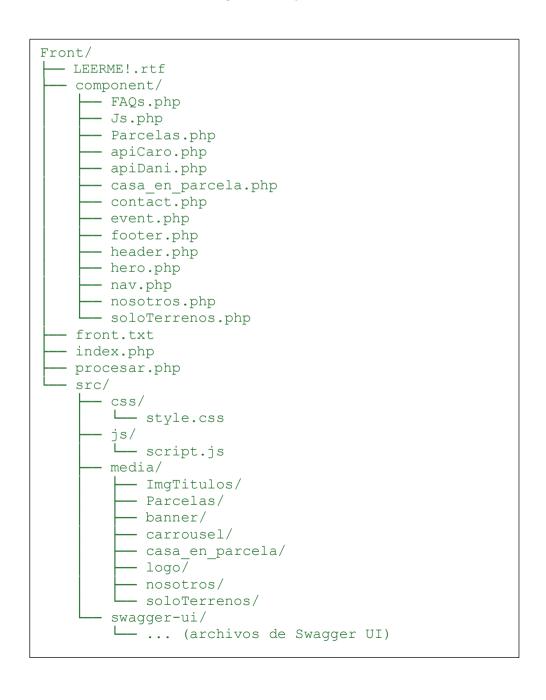
**DESARROLLO FRONTEND - IF201IINF** 

- Daniela Droguett
- Patricio Melo
- Carolina Bohle

## Tabla de contenido

DESARROLLO FRONTEND - IF201IINF	
Estructura del Proyecto (Basado en el archivo `front.txt`)	2
Informe Buenas Prácticas	3
HTML5	3
CSS3	
JavaScript con jQuery 3	3
Bootstrap 5	
Consumo de APIs en el Frontend	3
Desarrollo Frontend	4
Consumo de APIs	4

## Estructura del Proyecto (Basado en el archivo `front.txt`)



## Informe Buenas Prácticas

#### HTML5

- Semántica: Utiliza etiquetas HTML5 semánticas para mejorar la accesibilidad y el SEO. Por ejemplo, `<header>`, `<footer>`, `<article>`, `<section>`.
- Validación: Usa herramientas como el validador W3C para asegurarte de que tu HTML es válido.

## CSS3

- Modularidad: Divide tus estilos en varios archivos CSS si es necesario y organiza tus reglas CSS de manera modular.
- Preprocesadores CSS: Considera el uso de preprocesadores como Sass para mantener tu CSS limpio y fácil de mantener.
- Responsive Design: Utiliza media queries y las utilidades de Bootstrap para asegurar que tu diseño se adapte a diferentes tamaños de pantalla.
- Optimización: Minifica tus archivos CSS para reducir el tiempo de carga.

## JavaScript con jQuery 3

- Estructura de Código: Mantén tu código modular y organizado. Usa funciones y objetos para dividir tu código en partes reutilizables.
- Uso de jQuery: Aprovecha las facilidades que ofrece jQuery para la manipulación del DOM y manejo de eventos.
- Optimización: Minifica tus archivos JavaScript para mejorar el rendimiento.

## Bootstrap 5

- Componentes Reutilizables: Usa componentes de Bootstrap para acelerar el desarrollo y mantener la consistencia del diseño.
- Personalización: Personaliza Bootstrap a través de Sass para adaptar el framework a las necesidades específicas de tu proyecto.

## Consumo de APIs en el Frontend

- jQuery: Usa \$.get, \$.post para realizar solicitudes a las APIs.
- Manejo de Errores: Implementa un manejo adecuado de errores usando .fail() o .catch().
- Seguridad: Usa HTTPS para todas las solicitudes de API y maneja CORS adecuadamente en el servidor.
- Documentación de API: Documenta tus APIs usando Swagger UI u otra herramienta de

## **Desarrollo Frontend**

#### 1. HTML5:

- Usa etiquetas semánticas.
- Valida tu HTML regularmente.

## 2. CSS3:

- Organiza tus estilos en módulos.
- Usa preprocesadores como Sass.
- Implementa diseño responsive.
- Minifica tus archivos CSS.

## 3. JavaScript y jQuery:

- Mantén tu código modular.
- Usa jQuery para facilitar la manipulación del DOM.
- Minifica tus archivos JS.

## 4. Bootstrap 5:

- Usa componentes de Bootstrap.
- Personaliza Bootstrap según tus necesidades.

## Consumo de APIs

- 1. Solicitudes API:
  - Usa \$.get, \$.post.
  - Implementa manejo de errores.

## 2. Seguridad:

- Usa HTTPS.
- Configura CORS adecuadamente.

#### 3. Documentación:

- Documenta tus APIs con Swagger UI.