

DESARROLLO WEB

Estudiantes:
Luis Soto, B77551
Santiago Herra, C13721
Josué M. Jiménez, C13987



¿QUÉ ES EL DESARROLLO WEB?

Concepto

- Creación y mantenimiento de sitios y apps web.
- Incluye frontend (visible) y backend (oculto).

Frontend vs Backend

- Frontend (cliente): interfaz, diseño, UX, animaciones, eventos.
- Backend (servidor): lógica, datos, seguridad, APIs, autenticación.
- Razón de la división: separar responsabilidades → mejor optimización.

¿QUÉ ES EL DESARROLLO WEB?

Evolución del desarrollo web

- Web 1.0: páginas estáticas (HTML fijo).
- Web 2.0: páginas dinámicas (datos + servidor).
- Web moderna:
 - SPA: carga única, contenido dinámico.
 - PWA: apps web con funciones nativas (offline, push, instalación).

Roles principales

- Frontend: presenta → interfaz, interacción, validaciones rápidas.
- Backend: sostiene → procesos, lógica, datos, seguridad, APIs.



FRONTEND

FRONTEND

Tecnologías base

- HTML → Estructura del contenido
- CSS → Diseño y estilo visual
- JavaScript → Interactividad y dinamismo

FRONTEND

HTML

- Lenguaje de marcado estándar
- Define la estructura (títulos, párrafos, imágenes)
- Semántica: accesibilidad y SEO
- Permite multimedia y formularios

CSS

- Separa contenido y presentación
- Colores, tipografía, diseño visual
- Layouts adaptativos (Flexbox, Grid)
- Animaciones y transiciones

FRONTEND

JavaScript

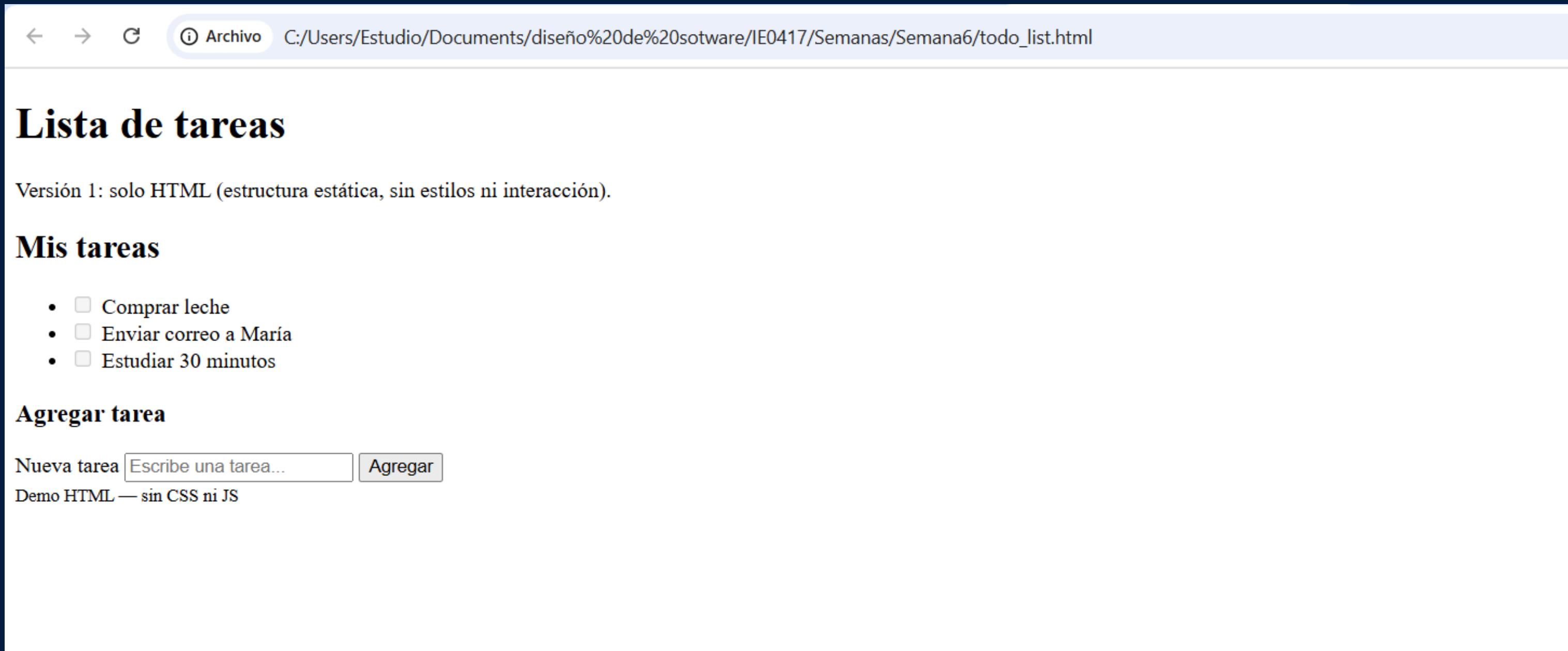
- Lógica e interactividad
- Modifica DOM en tiempo real
- Responde a eventos del usuario
- Validaciones, animaciones, AJAX

Ahora todos juntos

- Mejor experiencia de usuario
- Mantenimiento más fácil
- Adaptable a cualquier dispositivo
- Optimización de rendimiento

FRONTEND

Cómo se complementan



The screenshot shows a web browser window with the following details:

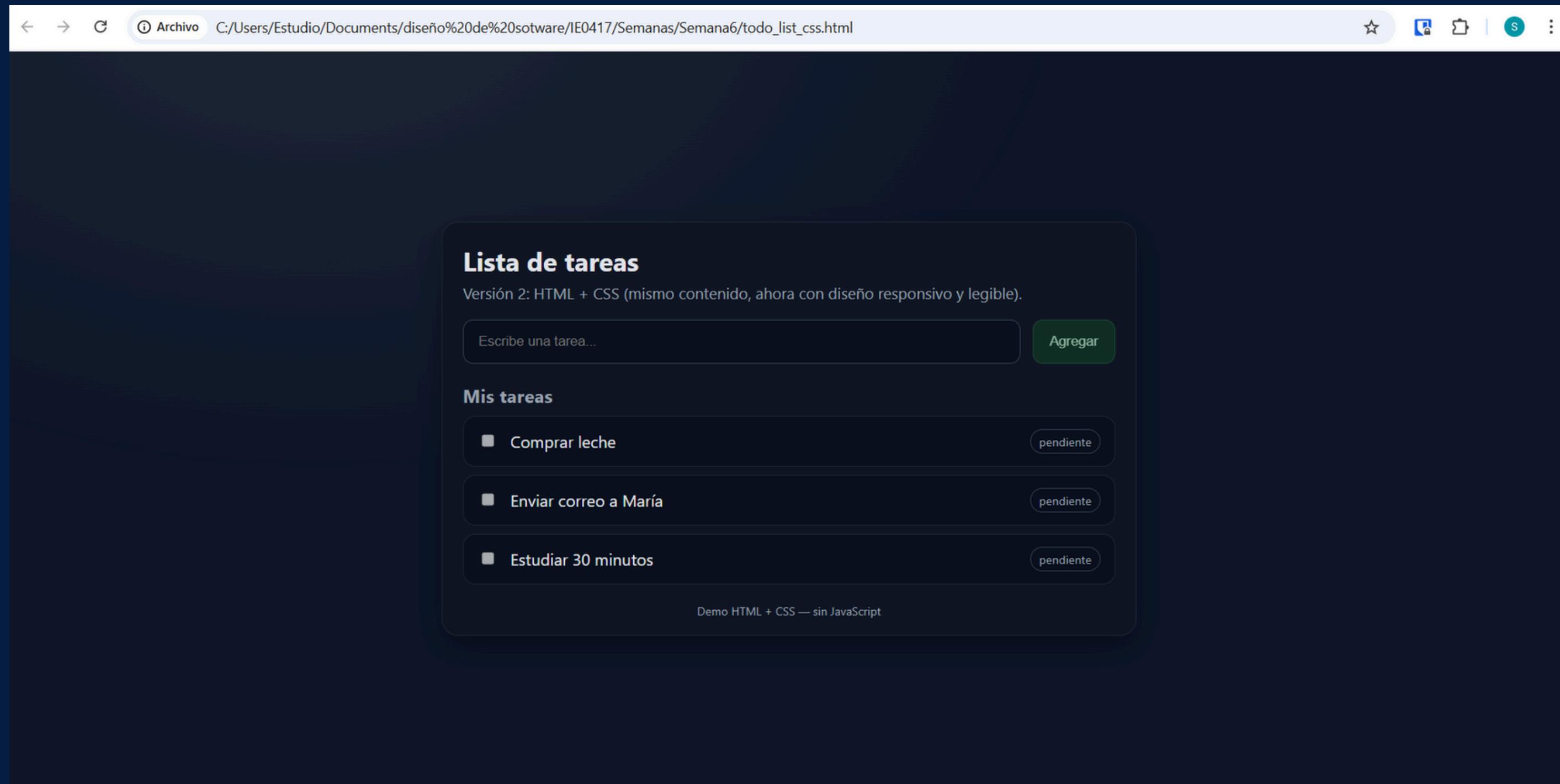
- Address Bar:** C:/Users/Estudio/Documents/diseño%20de%20sotware/IE0417/Semanas/Semana6/todo_list.html
- Title:** Archivo
- Content Area:**
 - ### Lista de tareas
 - Versión 1: solo HTML (estructura estática, sin estilos ni interacción).
 - ### Mis tareas

 - Comprar leche
 - Enviar correo a María
 - Estudiar 30 minutos
 - ### Agregar tarea

Nueva tarea
 - Demo HTML — sin CSS ni JS

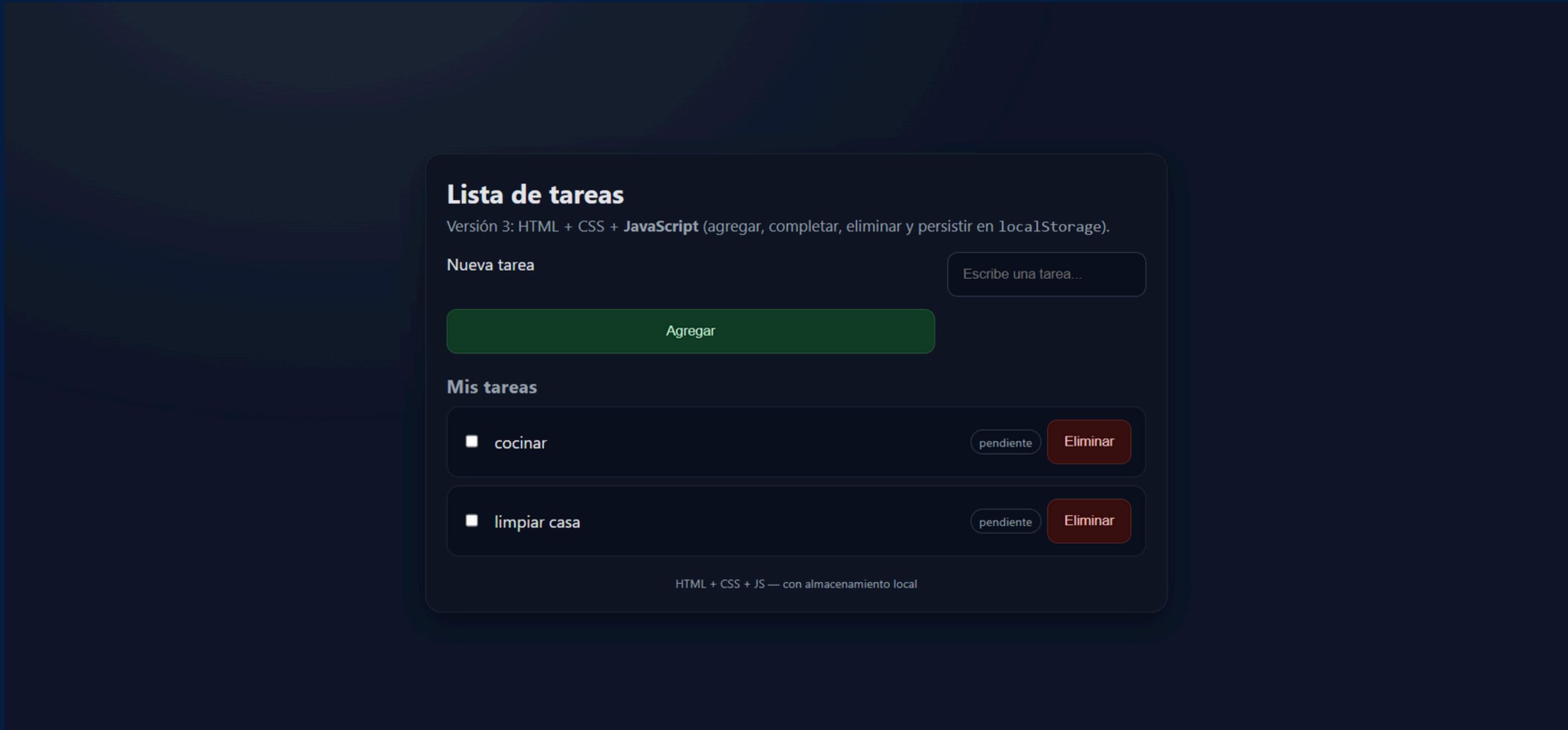
FRONTEND

CSS + HTML



FRONTEND

CSS + HTML + JS

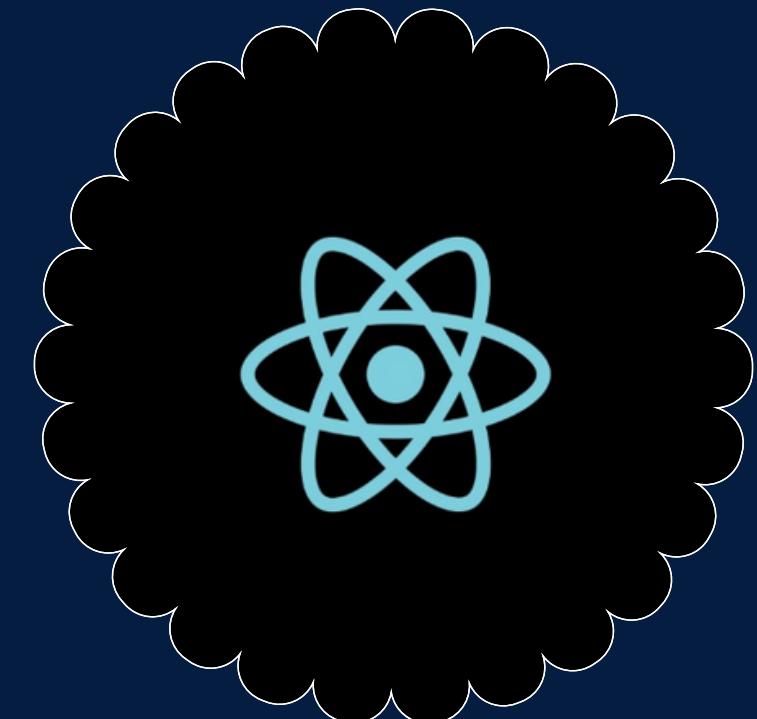


FRONTEND - FRAMEWORKS

- React, Angular, Vue
- Componentes reutilizables
- Virtual DOM → más rápido
- Organización y escalabilidad



- **Vue**: Alibaba, GitLab
- **Angular**: Netflix, Google
- **React**: Facebook, dashboards





BACKEND



BACK-END

Lenguajes Comunes

- Javascript (Node.js)
- Python
- Java
- C#
- PHP
- Rust
- Ruby
- Go (Golang)



BACK-END

Javascript (Node.js)

- Es asíncrono y no bloqueante.
- Buen lenguaje para desarrollar full-stack.

Frameworks

- Express.js
- NestJS
- Fastify
- Koa.js

FRAMEWORKS DE JAVASCRIPT

Express.js

- No fuerza una arquitectura definida.
- Relativamente sencillo y flexible.
- Utilizado en startups, para prototipar y desarrollo de microservicios.



FRAMEWORKS DE JAVASCRIPT

NestJS

- Construido sobre Express o Fastify
- Se debe implementar una arquitectura modular forzada.
- Resulta en codebases más uniformes y sostenibles para grandes equipos de trabajo.
- Arquitecturas de microservicios y empresas que desean escalar.



FRAMEWORKS DE JAVASCRIPT



Fastify

- Optimizado para realizar más solicitudes por segundo utilizando menos recursos.
- Enfocado en la capa de red para realizar transferencias de forma eficiente.
- Se complementa con otros frameworks para tener “lo mejor de ambos mundos”
- Bueno para APIs con tráfico alto y microservicios

FRAMEWORKS DE JAVASCRIPT



Koa.js

- Creado por el mismo equipo que Express.js
- Más liviano y sencillo.
- Posee menos funcionalidades.
- Utilizado en proyectos en donde se requiera una alternativa más “lightweight” que Express.

BACK-END

Python

- Útil cuando el back-end debe interactuar con procesamiento de datos masivos o complejos o interactuar con modelos de aprendizaje de máquina.

Frameworks

- Django
- Flask
- FastAPI

FRAMEWORKS DE PYTHON

Django

- Es de estilo “batteries included”.
- El usuario debe seguir una estructura definida para desarrollar proyectos.
- En la versión 3.1 en adelante se puede desarrollar código asincrónico.
- Los proyectos desarrollados con Django suelen ser escalables si se aplican las prácticas correctas.

The logo for Django, featuring the word "django" in a lowercase, sans-serif font. The letter "d" is positioned on the left side of the word, while the other letters "j", "a", "n", "g", and "o" are stacked vertically to its right, creating a stylized, dynamic effect.

FRAMEWORKS DE PYTHON

Flask

- Proporciona funcionalidades mínimas para el desarrollo web.
- Funcionalidades más complejas deben de ser agregadas manualmente.
- Es ligero, por lo que sirve en microservicios, APIs ligeras y aplicaciones IoT.



Flask
web development,
one drop at a time

FRAMEWORKS DE JAVASCRIPT



FastAPI

FastAPI

- Framework moderno para construir APIs.
- Genera documentación automática.
- Tiene soporte para operaciones asincrónicas.
- Es útil en la confección de REST APIs, microservicios y APIs relacionadas a aprendizaje de máquina.

BACK-END

Java

- Utilizado en sectores en donde la escalabilidad, estabilidad y el mantenimiento a largo plazo es importante.
- El lenguaje es “strong-typed”.
- Tiene un amplio soporte de librerías actualizadas.
- El código es compilado y ejecutado en JVM.

Frameworks

- El más utilizado es Spring / Spring Boot

FRAMEWORKS DE JAVA

Spring / Spring Boot

- Spring Boot es una versión simplificada de Spring.
- Son altamente integrables con servicios de nube.
- Spring existe desde el 2003 y ha demostrado ser confiable.
- Requieren de estructura determinada por lo que no es ideal para prototipar.
- Útil en escenarios en los que se necesita robustez en ejecución y desarrollo de microservicios complejos.
- Puede tener una curva de aprendizaje mayor que otros lenguajes y frameworks.



BACK-END

C#

- Se ejecuta en ambiente .NET
- Se considera la versión de Java de Microsoft.

Frameworks

- ASP.Net Core

FRAMEWORKS DE C#

ASP.Net Core

- Es de los más veloces para el desarrollo web.
- Utiliza operaciones asincrónicas en conjunto con optimizaciones en desarrollo web.
- Amplia integración con productos de Microsoft.
- Diseñado para aplicaciones web rápidas y escalables.
- Puede tener una curva de aprendizaje mayor que otros lenguajes y frameworks.



BACK-END

PHP

- Utilizado para generar páginas dinámicas de manera sencilla.
- Uno de los primeros lenguajes creados con el propósito de crear páginas web.
- Aproximadamente 80% de las páginas en internet utilizan PHP.

Frameworks

- Laravel (Simplifica el uso de PHP y proporciona funcionalidades de desarrollo web).



BACK-END

Go (Golang)

- Busca combinar simplicidad de Python, el rendimiento de C y un modelo de concurrencia eficiente.
- Los archivos binarios no necesitan de un entorno virtual.
- Ideal en creación de APIs de gran rendimiento, microservicios y ambientes de contenedores.

Frameworks

- Gin → REST Apis
- Echo → Más funcionalidades que Gin
- Fiber → Inspirado por Express.js para simplicidad y velocidad.



BACK-END

Rust

- Lenguaje enfocado en brindar el rendimiento de C utilizando un mejor manejo de memoria y concurrencia.
- No utiliza “garbage collector”.
- El compilador checkea reglas que deben cumplirse.
- Utilizado en APIs para la nube, contenedores y finanzas como blockchain.

Frameworks

- Actix Web → Velocidad, concurrencia
- Rocket → Batteries included



BACK-END

Ruby

- Lenguaje de alto nivel dinámico de propósito general.

Frameworks

- Ruby on Rails

FRAMEWORKS DE RUBY

Ruby on Rails

- Ampliamente utilizado por startups en años anteriores.
- Aplicaciones suelen ser más lentas que las desarrolladas en otros lenguajes / frameworks.
- Presenta muchas facilidades para el programador.
- Suele consumir bastante memoria.
- En los últimos años su uso ha disminuido por utilizar opciones más óptimas de desarrollo.





CASOS DE USO



Con el fin de comprender de manera más concreta cómo se aplica el desarrollo web en la práctica, a continuación se presentan algunos de sus principales casos de uso

E-COMMERCE

Definición: Compra y venta de bienes/servicios a través de Internet (Hayes y Downie)

Amazon.com

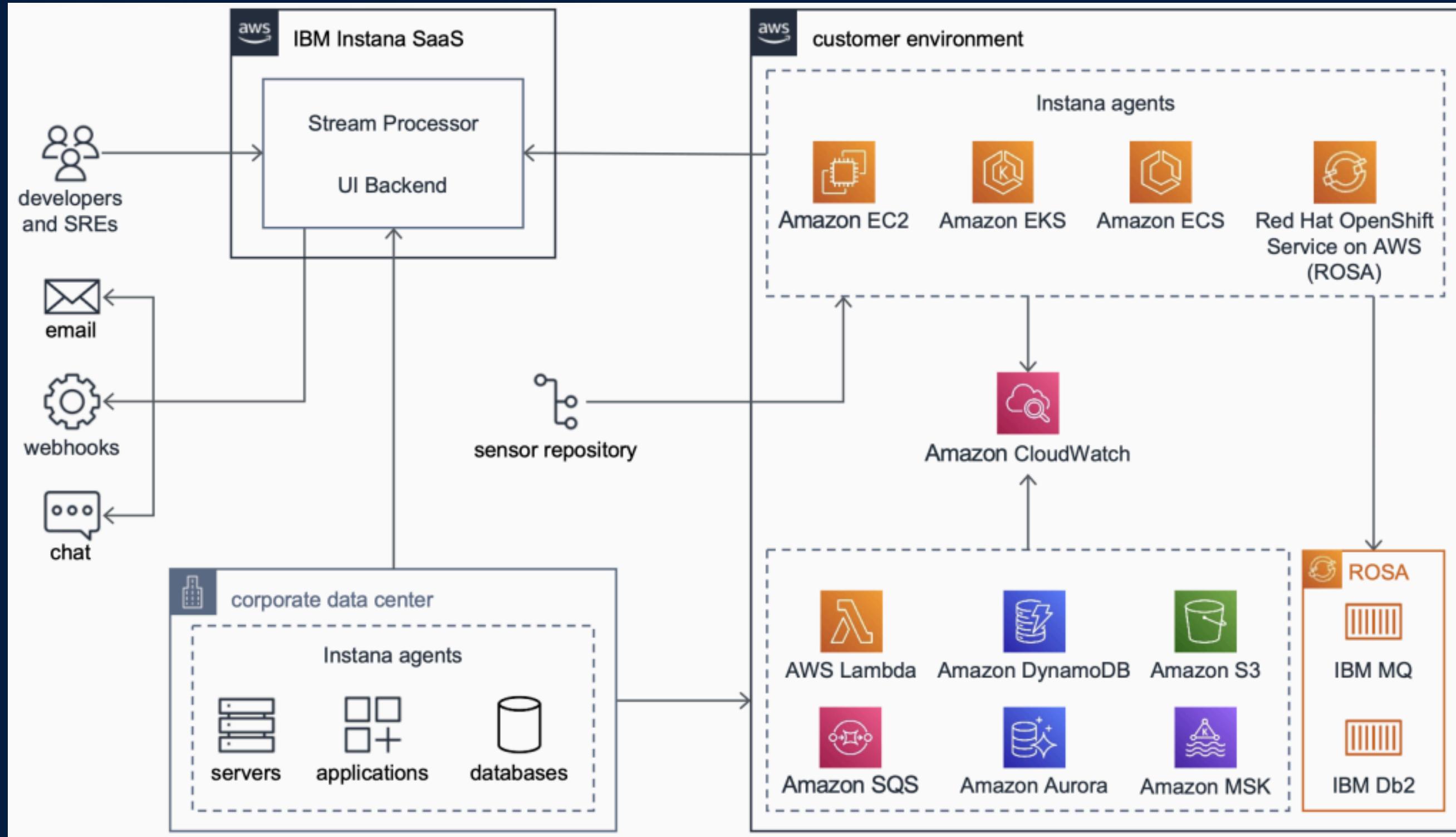
Front-end:

- Interfaz visual → catálogo, buscador, filtros, reseñas, carrito, pagos.
- Tecnologías → HTML, CSS, JavaScript, frameworks modernos.
- Objetivo → experiencia intuitiva, rápida y adaptable a distintos dispositivos.

Back-end:

- Lógica oculta → inventario, autenticación, pagos, seguridad, recomendaciones.
- Tecnologías → Java, Python, Node.js, arquitecturas distribuidas.
- Objetivo → escalabilidad y disponibilidad incluso en eventos masivos (Black Friday, Prime Day).

E-COMMERCE (Amazon.com)



Realtime monitoring of microservices and cloud-native applications with IBM Instana SaaS on AWS

REDES SOCIALES

Definición: Comunicación digital que permite formar redes y comunidades (Paljug).

Facebook

Front-end:

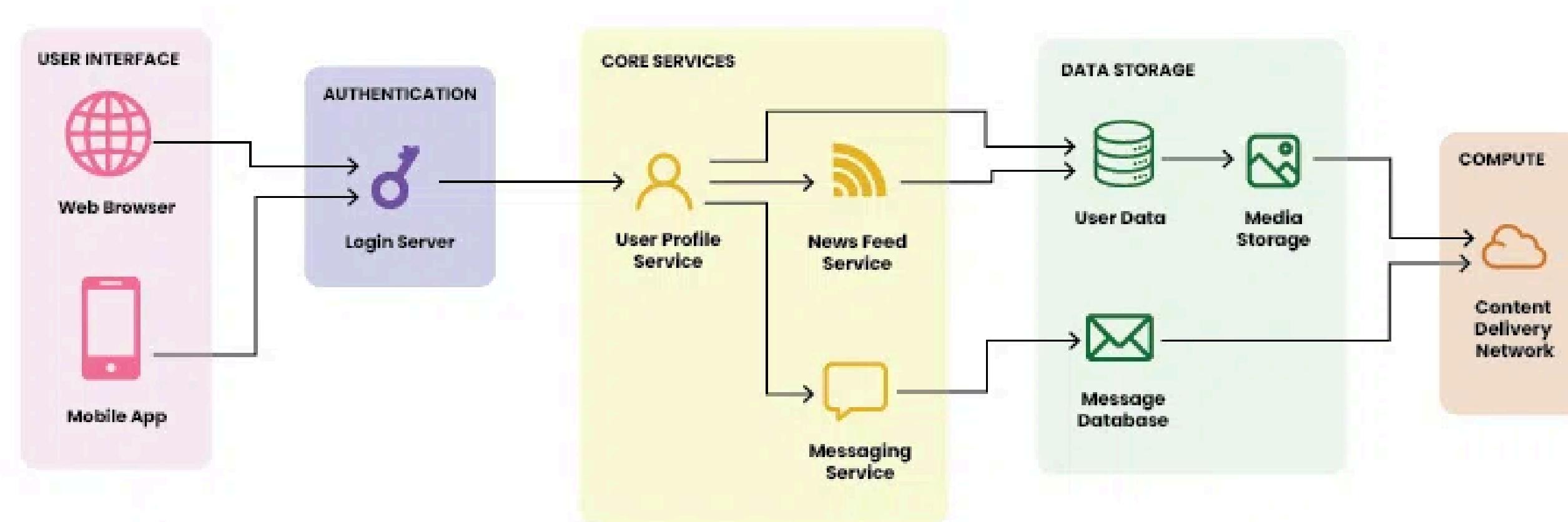
- Interfaz → muro, comentarios, mensajes, notificaciones, historias, transmisiones en vivo.
- Tecnologías → HTML, CSS, JavaScript, React.
- Objetivo → interacción en tiempo real, diseño atractivo y responsivo.

Back-end:

- Lógica y datos → publicaciones, relaciones de amigos, autenticación, recomendaciones IA, seguridad.
- Tecnologías → PHP, Python, C++, bases de datos distribuidas, sistemas de caché.
- Objetivo → respuesta rápida y experiencia global en tiempo real.

RED SOCIAL (Facebook)

High-Level Design





RETOS EN EL DESARROLLO WEB



RETOS EN EL DESARROLLO WEB

Escalabilidad

- Preparar la app para crecer de pocos a millones de usuarios.
- Estrategias: balanceadores de carga, bases de datos distribuidas, microservicios.
- Ejemplo: Amazon en Black Friday.

Seguridad

- Proteger datos de usuarios.
- Prevenir ataques: SQL Injection, XSS (código malicioso en frontend), Robo de sesiones
- Uso de HTTPS y cifrado de contraseñas.

Experiencia de Usuario (UX)

- Web rápida, intuitiva y accesible en cualquier dispositivo.
- Diseño responsivo y cargas veloces → evitar abandono.
- Ejemplos: Netflix, Spotify.

RETOS EN EL DESARROLLO WEB

Mantenimiento y evolución

- La app es un servicio en constante cambio.
- Actualizaciones sin afectar estabilidad.
- Buenas prácticas: versionado, documentación, pruebas automatizadas.

Integración con servicios externos

Uso de APIs de terceros:

- Pagos: Stripe, PayPal.
- Autenticación: Google, Facebook, GitHub.
- Mapas, mensajería.

Retos: compatibilidad y seguridad.

ÁREAS LABORALES EN EL DESARROLLO WEB



ÁREAS LABORALES EL DESARROLLO WEB

Front-End

- Se encargan de generar UI/UX que retenga a los usuarios.
- Normalmente contratados en compañías grandes en donde hay una distinción de roles entre back-end y front-end

Back-End

- Se encarga de desarrollar y mantener sistemas que integran la funcionalidad web.
- Se asegura de que la transmisión de información se ejecuta correctamente.
- Necesitados en industria en donde la escalabilidad web sea crítica.

Full-Stack

- Se encarga de realizar las labores tanto de front-end como de back-end.
- Se aseguran que cambios generados en la interfaz se integren correctamente con la lógica del servidor.
- Agencias de consultoría contratan desarrolladores full-stack para brindar apoyo a miembros de front-end y back-end.

ÁREAS LABORALES EL DESARROLLO WEB

Full-Stack

- El mercado laboral se tiende a postular más puestos laborales para desarrolladores full-stack.
- Es más cómodo para una empresa pequeña contratar a una persona que a dos.

ACTIVIDAD INTERACTIVA

KAHOOT