

Reporte Técnico de Actividades Práctico-Experimentales Nro. 006

1. Datos de Identificación del Estudiante y la Práctica

Nombre del estudiante(s)	Eberson Daniel Guayllas Poma
Asignatura	Desarrollo Basado en Plataformas
Ciclo	5 A
Unidad	2
Resultado de aprendizaje de la unidad	R1. Diseña e implementa aplicaciones Web básicas, bajo los principios de solidaridad, transparencia, responsabilidad y honestidad R2. Describe las diferencias entre software-comoservicio y los productos tradicionales de software, bajo los principios de solidaridad, transparencia, responsabilidad y honestida
Práctica Nro.	006
Título de la Práctica	Desarrollo de un prototipo de interfaz en Stitch y HTML/CSS con ocho pantallas documentadas en el repositorio del equipo.
Nombre del Docente	Edison Leonardo Coronel Romero
Fecha	Viernes 3 de octubre
Horario	07h30 – 10h30
Lugar	Laboratorio Computación aplicada Laboratorio Desarrollo de Software Laboratorio de redes y Sistemas Operativos Laboratorio Virtual EVA Aula
Tiempo planificado en el Sílabo	3 horas

2. Objetivo(s) de la Práctica

Diseñar y construir un prototipo sencillo de interfaz que represente al menos un flujo completo del proyecto (8 pantallas), utilizando Stitch como herramienta de apoyo al diseño y HTML/CSS para la implementación base.

Aplicar principios de diseño responsivo, maquetación semántica y criterios básicos de accesibilidad y usabilidad.

Documentar el prototipo mediante capturas de pantalla y una descripción técnica en el repositorio del equipo.

3. Materiales, Reactivos, Equipos y Herramientas

3.1 Materiales y reactivos:

- Computador con acceso a Internet.
- Navegador web actualizado.
- Editor de código (VS Code o similar).
- Herramienta de prototipado Stitch (o entorno definido por la asignatura).
- Repositorio del proyecto en GitHub / GitLab.
- Lineamientos de UI/UX del proyecto (si ya existen)

3.2 Equipos y herramientas

- Laboratorio de Desarrollo de Software o equipos personales.
- VS Code (o equivalente) con extensión Live Server (opcional).
- Stitch para diseño y organización de pantallas.
- Sistema de control de versiones Git + cliente (GitKraken, CLI, etc.).
- Navegador con DevTools para ver el diseño responsivo.

4. Procedimiento / Metodología Ejecutada

Paso 1: Definición rápida del flujo (en equipo)

- Elegir un flujo funcional del proyecto (por ejemplo: “registro y gestión de usuario”, “gestión de equipos”, “registro de resultados”, etc.).
- Definir las **8 pantallas** que formarán parte del flujo y sus relaciones (boceto rápido en pizarra o en Stitch).

Paso 2: Prototipado en Stitch

- Crear en Stitch las 8 pantallas con:
 - Distribución básica de elementos (header, menú, contenido, footer), ubicaciones aproximadas de botones, formularios, tablas/listas,
 - Jerarquía visual (títulos, subtítulos, texto).
- Ajustar la navegación entre pantallas (enlaces o interacciones de Stitch).

Paso 3: Implementación en HTML/CSS

- Crear una carpeta en el repositorio, por ejemplo:
`frontend/prototipos/ui-stitch/`
- Implementar las 8 pantallas en HTML5/CSS3 respetando:
 - Estructura semántica (`<header>`, `<nav>`, `<main>`, `<section>`, `<footer>`, etc.),
 - Diseño responsivo básico (uso de flex, grid o media queries), o legibilidad (tipografías, espaciados, tamaños de fuente).

Paso 4: Pruebas y capturas

- Verificar cada pantalla en al menos **dos tamaños de ventana** (ej. escritorio y móvil).
- Tomar **capturas de pantalla** de cada una de las 8 pantallas en su estado principal.

Paso 5: Documentación en el repositorio

En la carpeta `/docs/` o similar, crear un archivo, por ejemplo: `docs/ui_prototipo_stitch.md` con:

- Breve descripción del flujo que cubren las 8 pantallas.
- Una subsección para cada pantalla:
 - nombre de la pantalla,
 - propósito,
 - componentes principales (botones, formularios, listas, etc.), o captura de pantalla (enlazada o referenciada desde `/docs/img/`).

Paso 6: Commit y push

- Hacer commit con un mensaje descriptivo, por ejemplo: `feat(ui): prototipo HTML/CSS basado en Stitch (8 pantallas)`
- Hacer push a la rama correspondiente del equipo.

Cierre

- Breve socialización: cada equipo muestra 2–3 pantallas clave.
- Retroalimentación rápida del docente sobre:
 - coherencia visual,
 - uso de HTML semántico,
 - primeras evidencias de accesibilidad.

5. Resultados

Para poder cumplir con esta práctica en primer lugar elegimos nuestro flujo de trabajo que es el registro y gestión de usuario que se basa en:

- **Landing Page**
- **Login/Registro**
- **Dashboard Usuario**
- **Reto Activo**
- **Resultados del Reto**
- **Historial + Detalles del Reto**
- **Perfil del Usuario**
- **Notificaciones**

Las pantallas que elegimos para poder realizar los bocetos son las mismas que elegimos en nuestro flujo de trabajo.

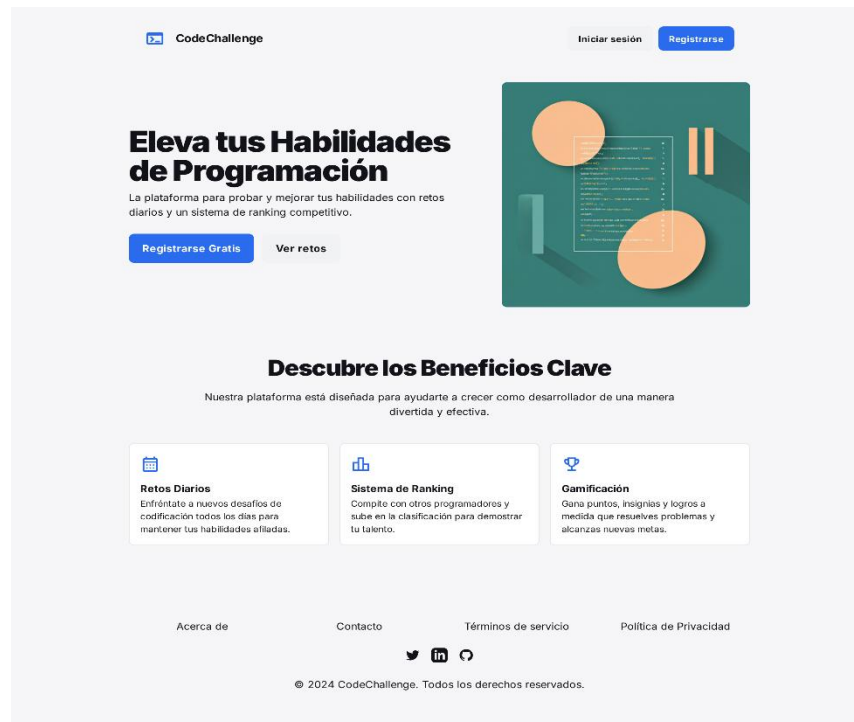
Una vez elegido nuestro flujo de trabajo empezamos con el prototipado en Stitch.

- **Lading page, es la primera vista que va tener el usuario el cual va a poder ingresar y crear una cuenta.**

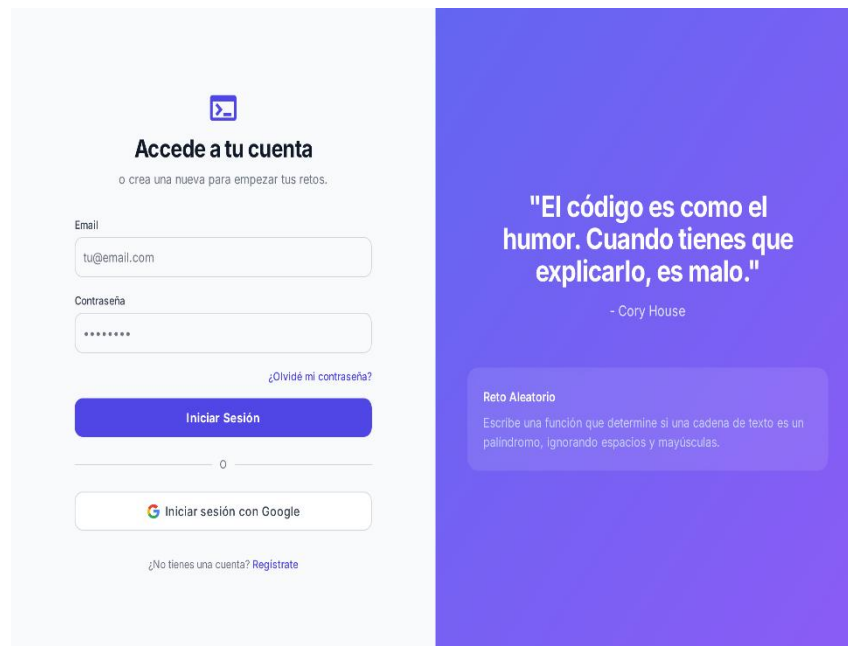


Universidad
Nacional
de Loja

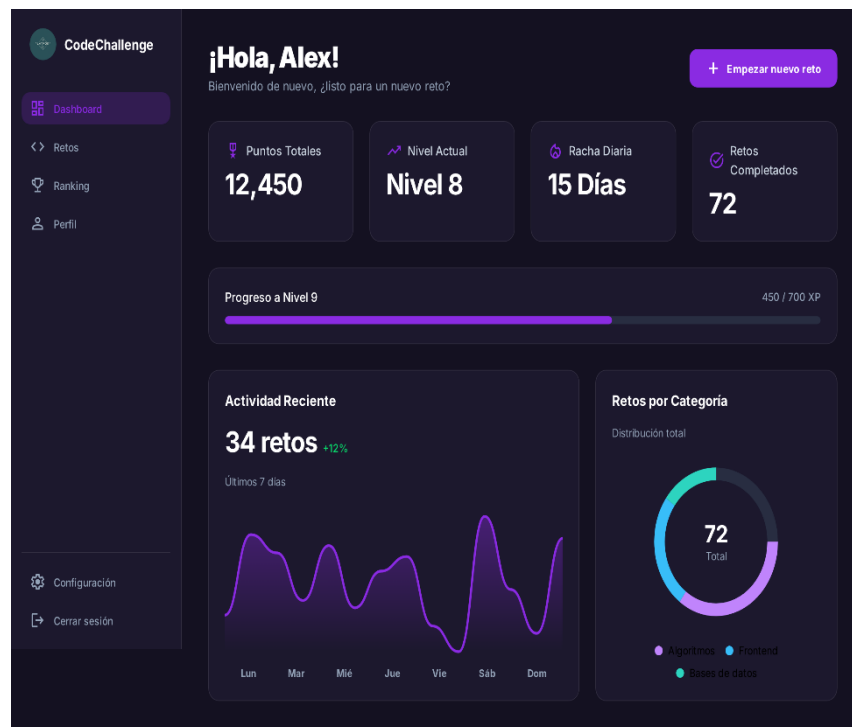
FEIRNNR - Carrera de Computación



- **Login/Registro, es la segunda vista del usuario el cual va acceder y recuperar la contraseña**



- **Dashboard usuario, aquí el usuario puede revisar sus puntos, el nivel que se encuentra, su racha diaria y el rendimiento.**



- **Reto activo, en esta pantalla se recibe los retos aleatorios y envía la solución**

The 'Reto Activo' page for 'Validación de Palíndromos' includes:

- Medio** difficulty level.
- Timer:** 01 Horas, 29 Minutos, 59 Segundos.
- Enunciado:** Descripción del Problema: Escribe una función que determine si una cadena de texto dada es un palíndromo. Un palíndromo es una palabra, frase, número u otra secuencia de caracteres que se lee igual hacia adelante que hacia atrás, ignorando espacios, puntuación y mayúsculas/minúsculas.
- Ejemplos:**
 - Input: "A man, a plan, a canal: Panama" Output: true
 - Input: "race a car" Output: false
- Restricciones:** La cadena de entrada consistirá solo en caracteres ASCII imprimibles. Considera que una cadena vacía es un palíndromo válido.
- Code Editor:** JavaScript environment with a function `isPalindrome(s)` to be implemented.
- Buttons:** 'Enviar Solución' and 'Volver a Retos'.

- **Resultados del reto, se puede revisar los test pasados y los puntos obtenidos**



CodeWarrior123

Level 15 - Grandmaster

Dashboard

Retos

Clasificación

Perfil

Discusiones

Ajustes

Cerrar Sesión

Alex Doe

CodeWarrior123

Rango Profesional

Editar Perfil

Compartir P...

Retos Completados

128

Puntos Totales

15,600

Racha Actual

42 Días

Insignias y Logros

10+

10+

10+

10+

10+

10+

Puntos Acumulados (Últimos 30 días)

Retos Completados

Reto	Dificultad	Lenguaje	Fecha	P.
Suma de Dos Números	Fácil	JavaScript	15 de Jul, 2024	
Invertir Cadena	Fácil	Python	14 de Jul, 2024	
Validar Palindromo	Medio	Java	12 de Jul, 2024	
Árbol Binario	Difícil	C++	10 de Jul, 2024	

- Historial, permite revisar todos los retos respondidos

CodeChallenges

Nuevos Retos

Mi Perfil

Clasificación

Historial de Retos

Retos Completados

28

Puntuación Media

185 pts

Tiempo Total

12h 45m

Buscar por título del reto...

Dificultad

Estado

Reto	Dificultad	Tiempo	Puntos	Estado	
Validación de Palindromos	Fácil	15:32 min	+150 pts	Completado	...
Algoritmo de Dijkstra	Difícil	45:10 min	-50 pts	Fallado	...
Inversión de Árbol Binario	Medio	22:05 min	+250 pts	Completado	...
FizzBuzz Extendido	Fácil	08:15 min	+100 pts	Completado	...
Ordenamiento de Burbuja	Medio	31:40 min	-20 pts	Fallado	...

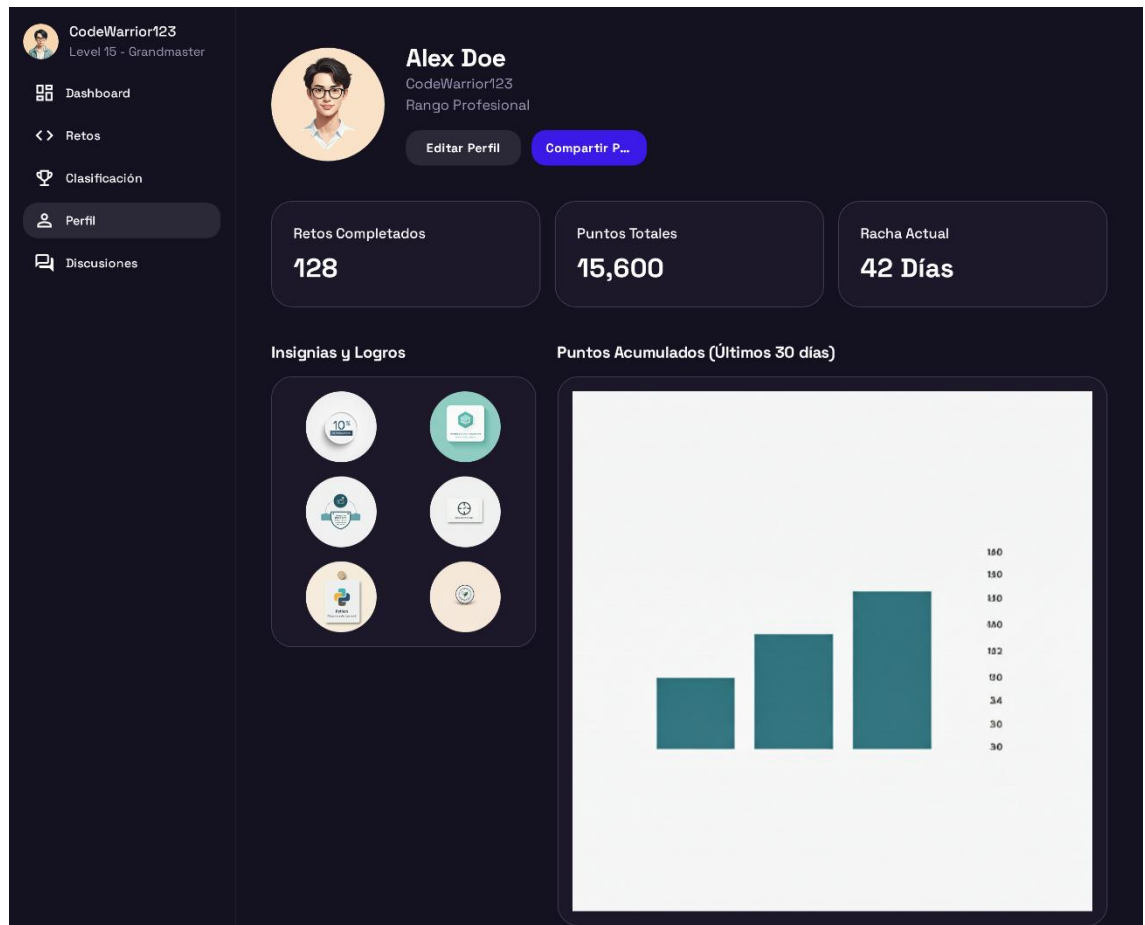
← Anterior

1 2 3 ... 10

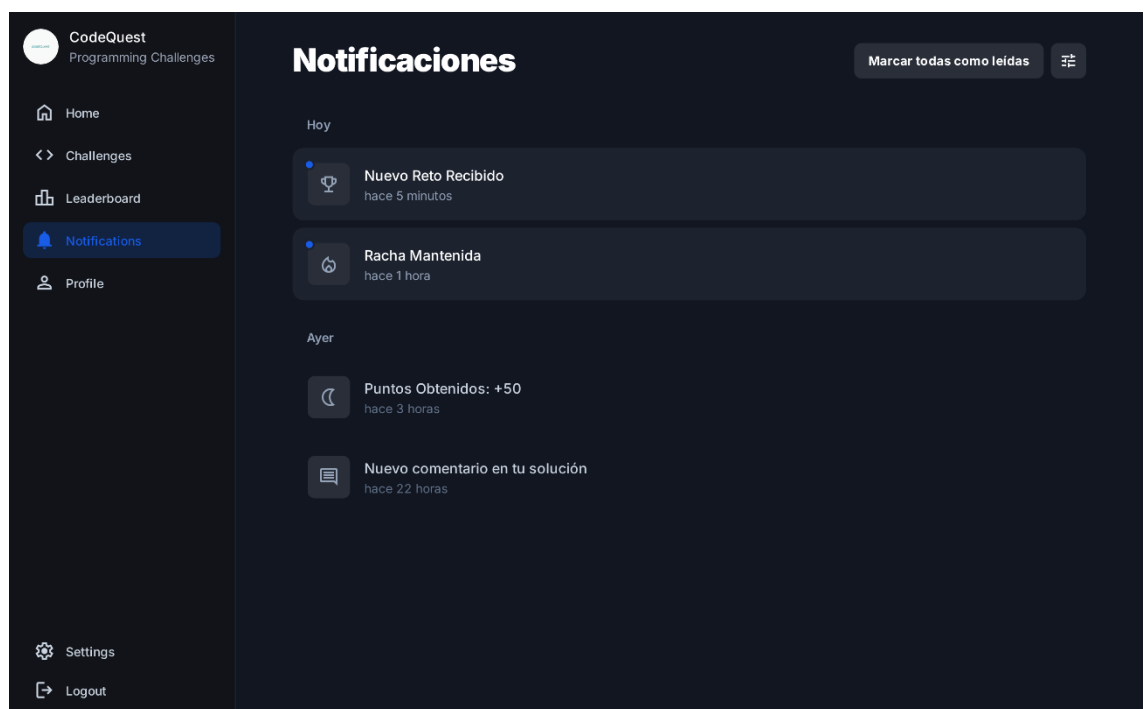
Siguiente →



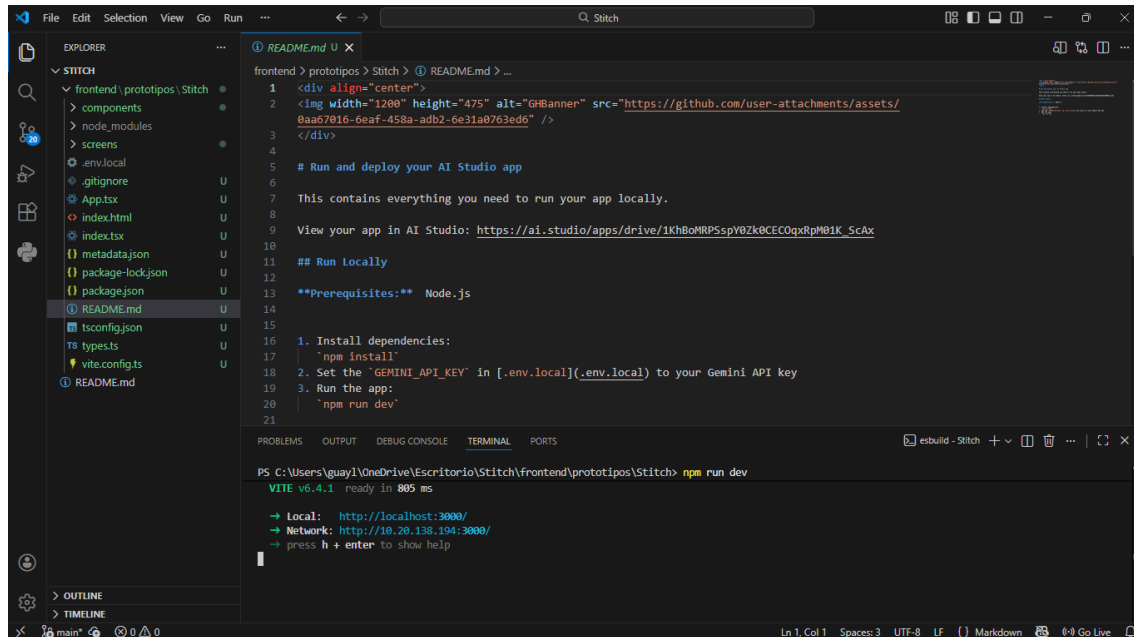
- **Perfil del usuario, se revisan las estadísticas y el progreso que obtiene a medida que responde los retos diarios**



- **Notificaciones, este módulo nos permite saber los nuevos retos diarios o recompensas.**



Después de implementar el prototipado en Stitch, realizamos la implementación en HTML/CSS, dividido en sus respectivas carpetas.

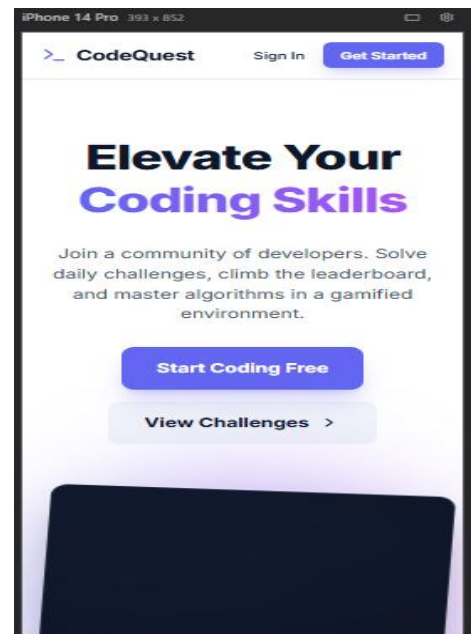
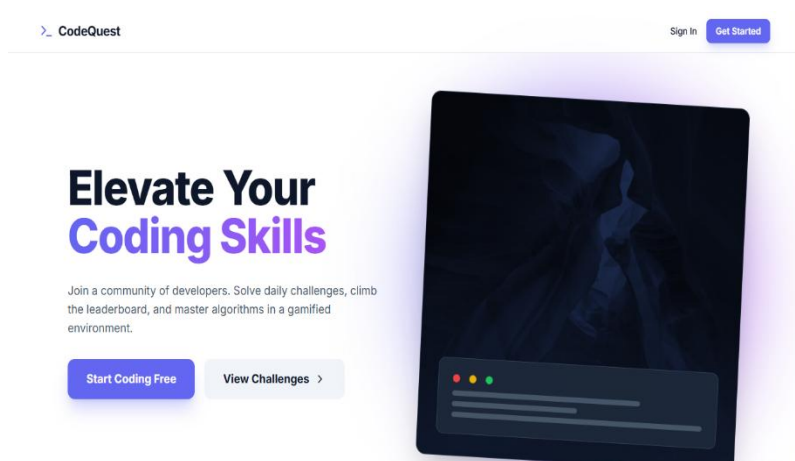


```
frontent > prototipos > Stitch > README.md > ...
1 <div align="center">
2 
3 </div>
4
5 # Run and deploy your AI Studio app
6
7 This contains everything you need to run your app locally.
8
9 View your app in AI Studio: https://ai.studio/apps/drive/1KhBoMRPSPy0Zk0CECOqxRpM01K_ScAx
10
11 ## Run Locally
12
13 **Prerequisites:** Node.js
14
15
16 1. Install dependencies:
17   'npm install'
18 2. Set the 'GEMINI_API_KEY' in [.env.local](.env.local) to your Gemini API key
19 3. Run the app:
20   'npm run dev'
21
```

```
PS C:\Users\guay1\OneDrive\Escritorio\Stitch\frontent\prototipos\Stitch> npm run dev
VITE v6.4.1 ready in 805 ms
  ➔ Local:   http://localhost:3000/
  ➔ Network: http://10.20.138.194:3000/
  ➔ press h + enter to show help
```

Ahora empezamos la verificación del responsive de las ocho pantallas en dos tamaños.

- Pantalla 1:



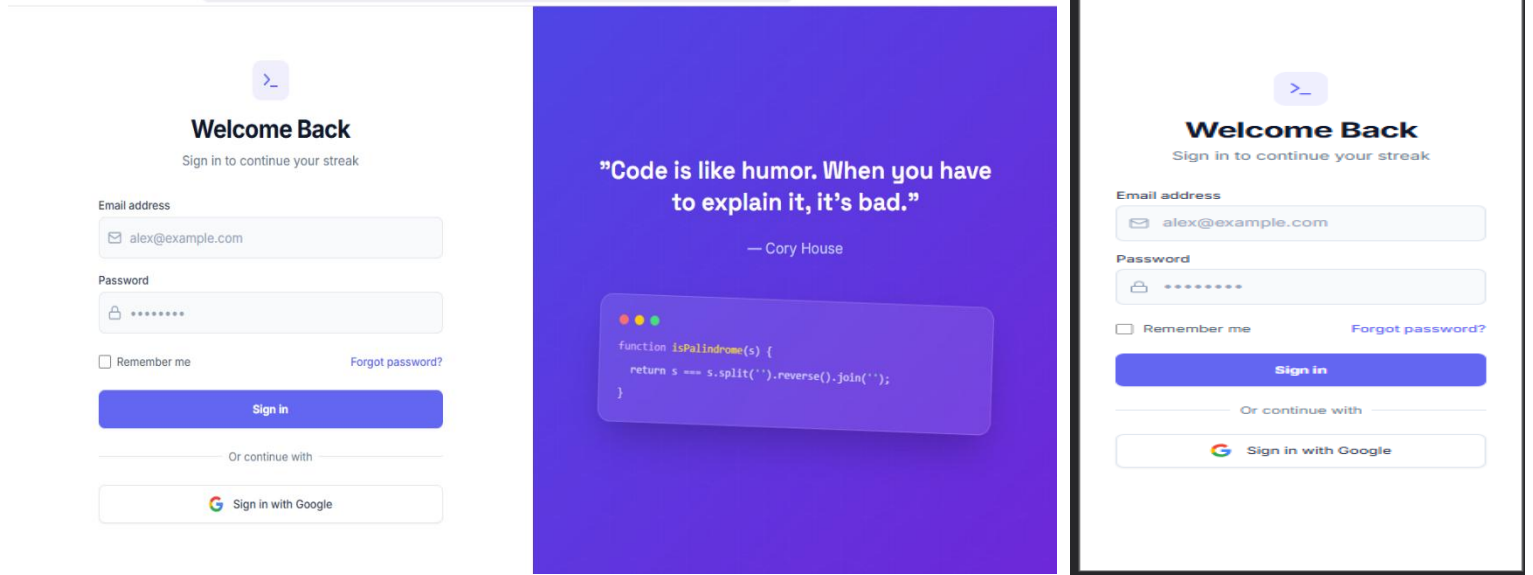


unl

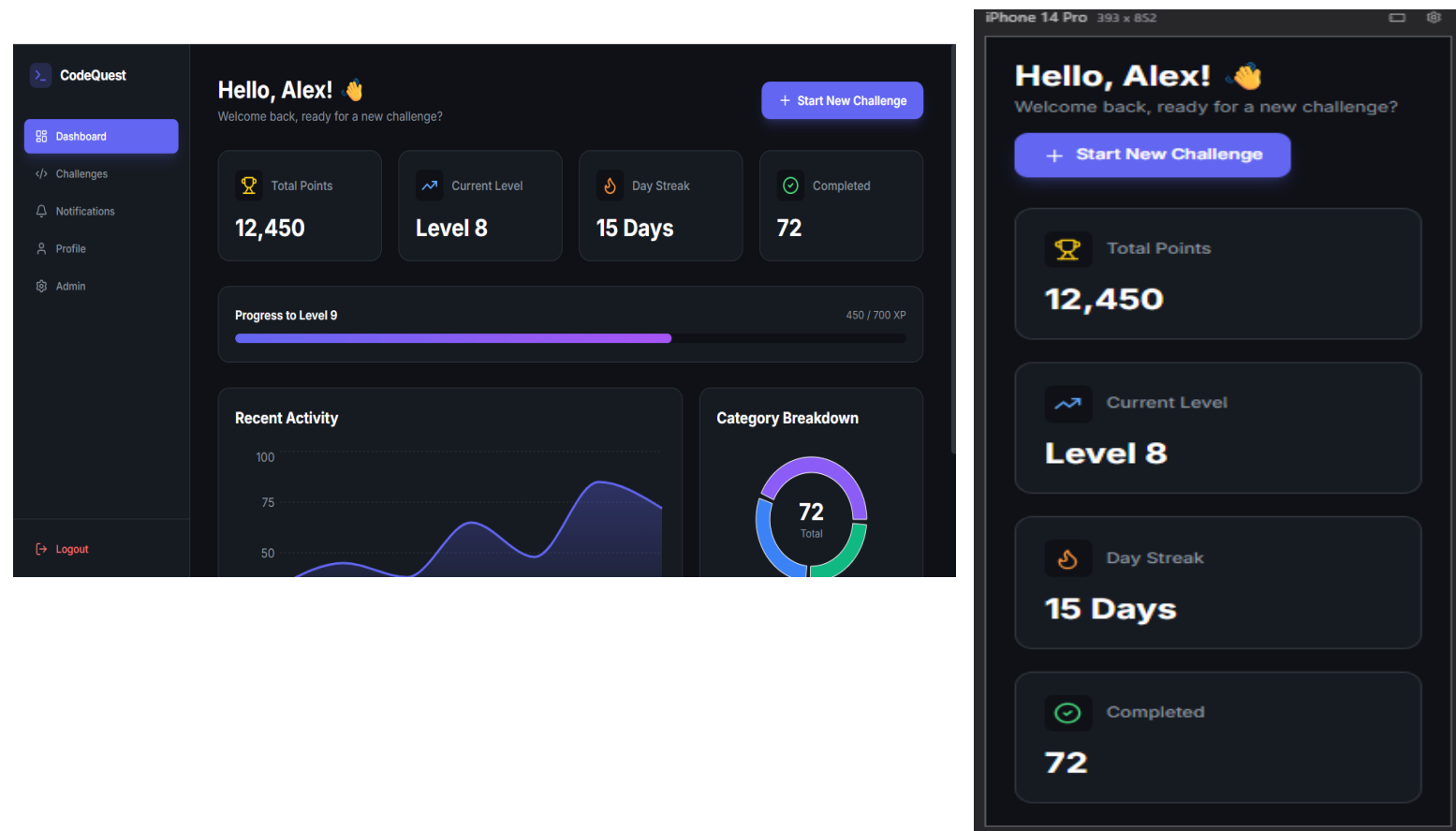
Universidad
Nacional
de Loja

FEIRNNR - Carrera de Computación

- Pantalla 2:



- Pantalla 3:





Universidad
Nacional
de Loja

FEIRNNR - Carrera de Computación

- Pantalla 4:

CodeQuest

Dashboard

Challenges

Notifications

Profile

Admin

Logout

Challenge History

Track your past submissions and performance.

Challenges Solved
28 Total

Average Score
185 Points

Time Spent
12h 45m Coding

CHALLENGE	DIFFICULTY	TIME	POINTS	STATUS	
Valid Palindrome JavaScript • 2 days ago	Easy	15:32 min	+150 pts	Solved	...
Dijkstra Algorithm Python • 5 days ago	Hard	45:10 min	50 pts	Failed	...
Invert Binary Tree TypeScript • 1 week ago	Medium	22:05 min	+250 pts	Solved	...
Two Sum Go • 1 week ago	Easy	08:15 min	+100 pts	Solved	...
Bubble Sort Java • 2 weeks ago	Medium	31:40 min	20 pts	Failed	...

Challenge History

Track your past submissions and performance.

Search challenges...

Filter

Challenges Solved
28 Total

Average Score
185 Points

Time Spent
12h 45m Coding

CHALLENGE	DIFFICULTY	TIME	POINTS	STATUS
Valid Palindrome JavaScript • 2 days ago	Easy	15:32 min	+150 pts	Solved
Dijkstra				

- Pantalla 5:

CodeQuest

Dashboard

Challenges

Notifications

Profile

Admin

Logout

Notifications

Mark all as read

TODAY

New Challenge Available
5 minutes ago

Streak Maintained
1 hour ago

YESTERDAY

Points Earned: +50
3 hours ago

New comment on your solution
22 hours ago

Notifications

Mark all as read

TODAY

New Challenge Available
5 minutes ago

Streak Maintained
1 hour ago

YESTERDAY

Points Earned: +50
3 hours ago

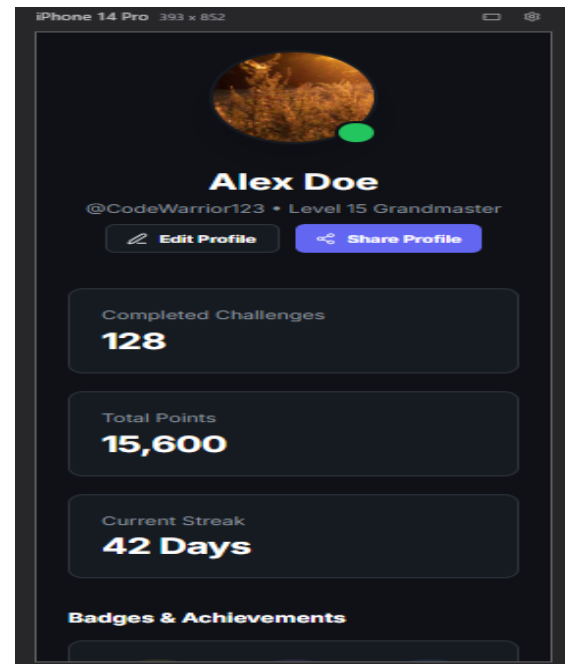
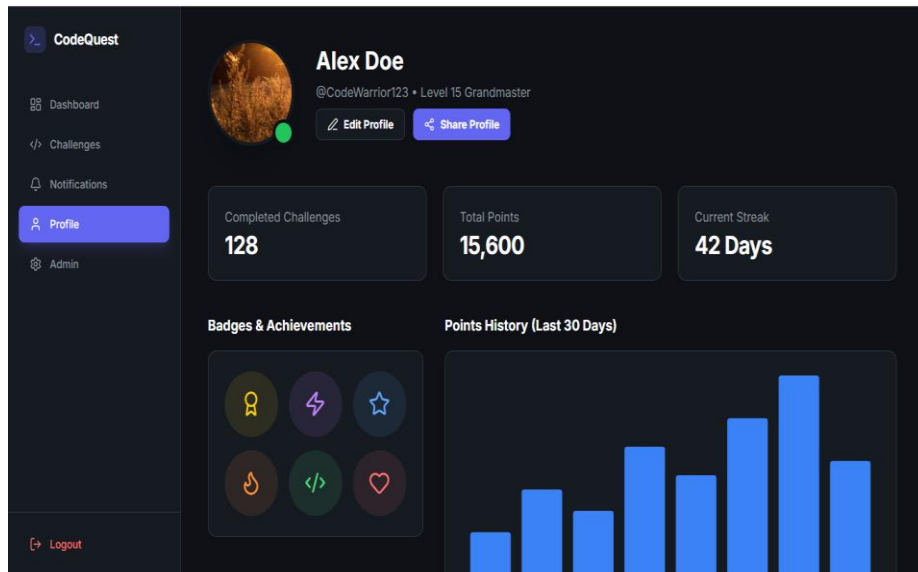
New comment on your solution
22 hours ago



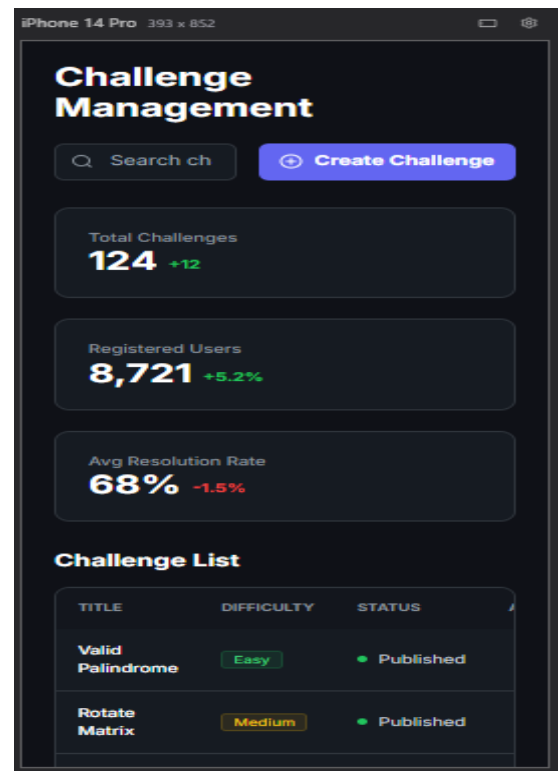
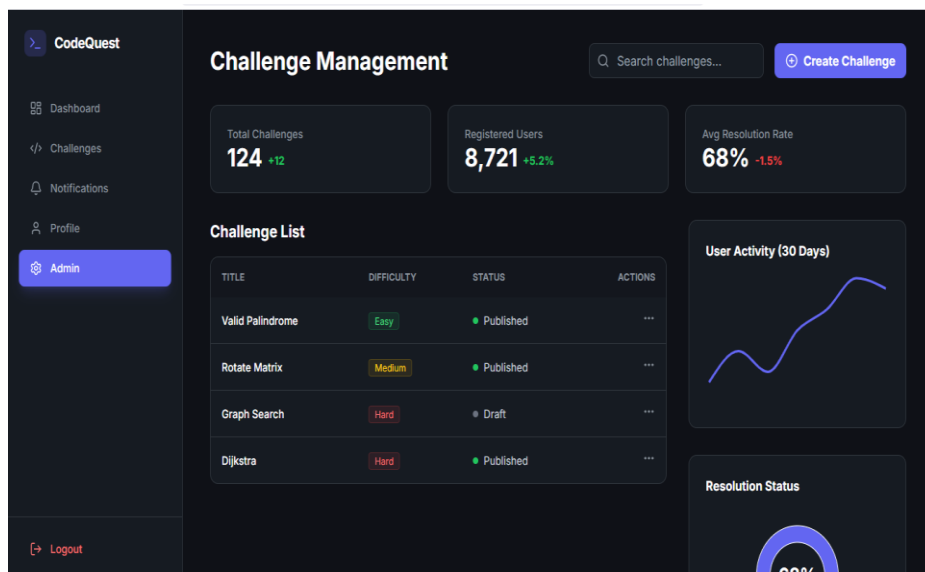
Universidad
Nacional
de Loja

FEIRNNR - Carrera de Computación

- Pantalla 6:

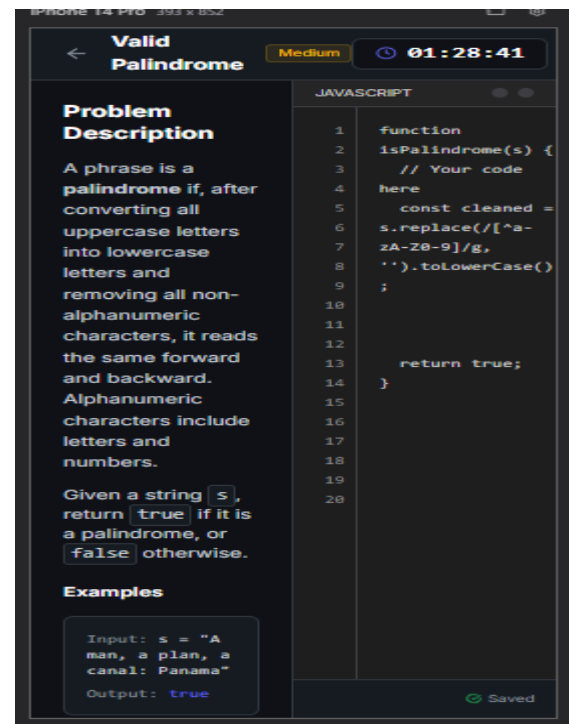
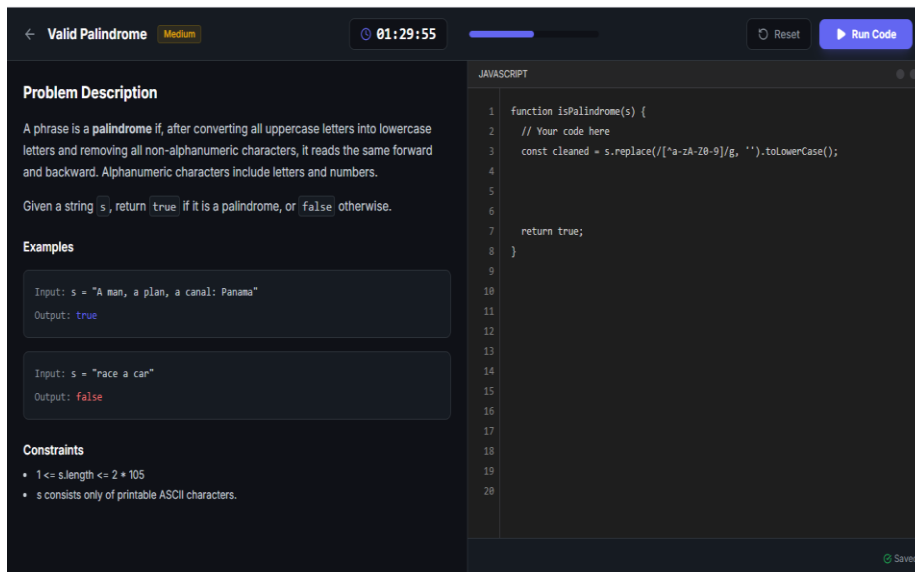


- Pantalla 7:

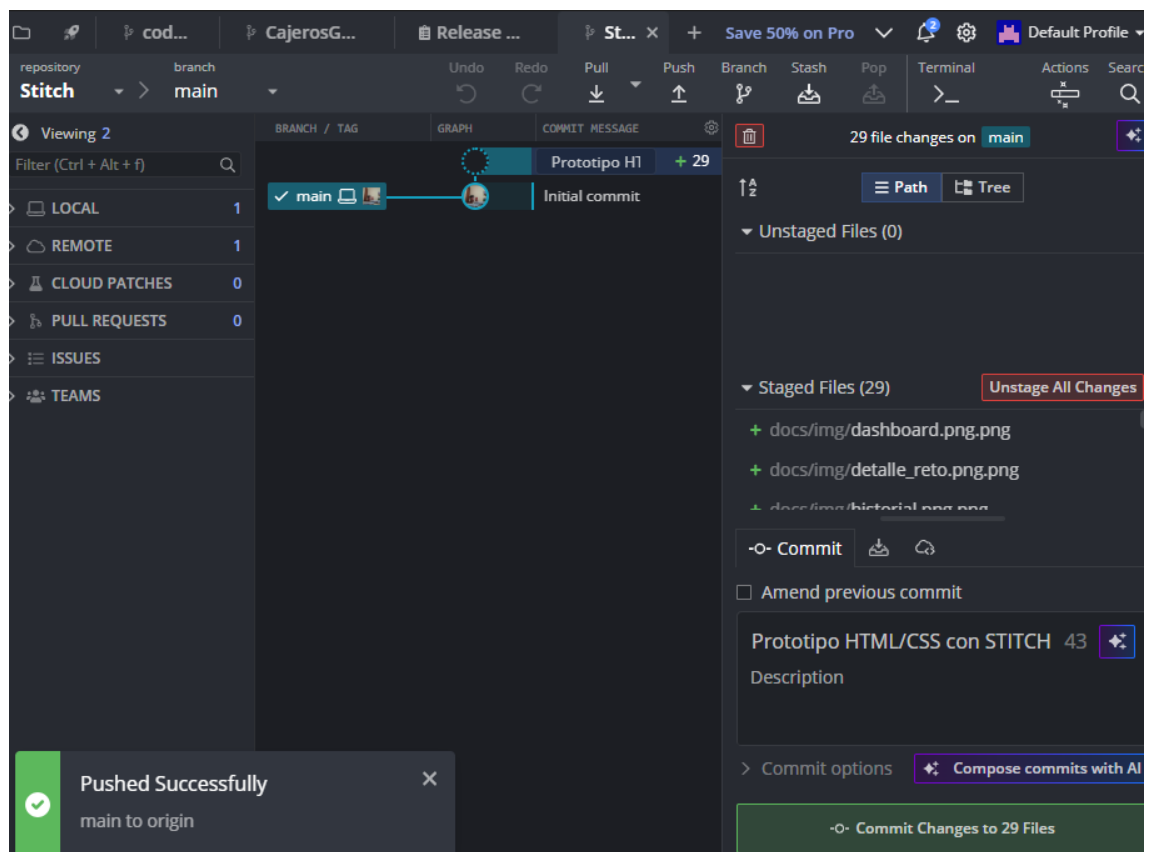




- Pantalla 8:



Una vez finalizado en GitHub realizamos un push con un mensaje descriptivo, así finalizamos la practica



6. Preguntas de Control

¿Qué ventajas encuentras al usar primero un prototipo en Stitch antes de codificar en HTML/CSS?

Utilizar Stitch para el prototipado ofrece una ventaja significativa en la velocidad de iteración y la validación temprana. Al ser una herramienta que facilita la transición de diseño a código, permite generar la estructura visual y exportar bases funcionales, lo que reduce drásticamente el tiempo de codificación manual y minimiza la "deuda técnica" inicial [1]. Además, permite visualizar la experiencia de usuario (UX) antes de invertir tiempo en la implementación técnica [5].

¿Por qué es importante que las pantallas estén construidas con etiquetas semánticas?

El uso de etiquetas semánticas (como `<header>`, `<main>`, `<footer>`) es fundamental porque dota de significado al contenido, más allá de su apariencia. Según la documentación de estándares web, esto es crucial para la accesibilidad, ya que permite a los lectores de pantalla interpretar correctamente la jerarquía de la información para usuarios con discapacidad visual [2]. Asimismo, los motores de búsqueda (SEO) priorizan el contenido estructurado semánticamente para indexar mejor las páginas [2].

¿Qué elementos de diseño responsivo aplicaste en, al menos, una de las pantallas?

Para garantizar la adaptabilidad en distintos dispositivos, se implementaron Media Queries (`@media`), las cuales permiten modificar los estilos CSS basándose en el ancho del `viewport` del dispositivo [4]. Adicionalmente, se utilizaron unidades relativas (como % o `rem`) y sistemas de diseño flexible como Flexbox, lo cual es esencial para asegurar que el contenido fluya correctamente sin romperse en pantallas móviles [4].

¿Qué mejoras podrías realizar para mejorar la accesibilidad (colores, tamaños, estructura)?

Siguiendo las Pautas de Accesibilidad para el Contenido Web (WCAG), una mejora prioritaria sería aumentar la relación de contraste entre el texto y el fondo para cumplir con el nivel AA o AAA [3]. También se podrían implementar atributos ARIA (`aria-label`) en elementos interactivos no textuales y asegurar indicadores de "foco" visibles para la navegación por teclado, garantizando una experiencia inclusiva para todos los usuarios [3].

¿Cómo ayuda esta práctica al desarrollo posterior del frontend definitivo del proyecto?

Esta práctica establece los cimientos del proyecto. Al validar el diseño y la estructura semántica desde el prototipo en Stitch, se reduce la incertidumbre y la reescritura de código en etapas avanzadas, esto permite que, en la fase de desarrollo definitivo, el equipo se concentre en la lógica de programación y la integración con el backend, teniendo ya resuelta la capa de presentación y accesibilidad básica.

7. Conclusiones

La integración de herramientas de prototipado como Stitch demuestra ser vital para optimizar el ciclo de desarrollo, permitiendo obtener código base utilizable y reduciendo los tiempos de entrega, lo cual se alinea con las metodologías ágiles actuales

Se concluye que un código HTML semántico no es opcional, sino un estándar de calidad indispensable. Garantizar la accesibilidad desde el inicio del proyecto (y no como un parche final) asegura el cumplimiento de estándares internacionales como WCAG

El diseño responsivo implementado mediante *media queries* y unidades relativas confirma que el desarrollo web moderno debe ser "Mobile First" para satisfacer la demanda de usuarios en múltiples plataformas

8. Recomendaciones

Validación Manual del Código Generado: Aunque herramientas como Stitch agilizan el proceso, se recomienda auditar manualmente el código exportado para asegurar que cumple estrictamente con los estándares semánticos y no contiene estilos redundantes

Pruebas con Lectores de Pantalla: Para garantizar la accesibilidad real, se sugiere no solo validar el contraste de color, sino realizar pruebas de navegación utilizando lectores de pantalla (como NVDA o VoiceOver) durante la etapa de desarrollo

Incorporar extensiones como Lighthouse o WAVE en el navegador para detectar fallos de rendimiento y accesibilidad en tiempo real mientras se codifica

9. Bibliografía / Referencias

[1] Y. Sharma, "Stitch by Google: From Prompt to Prototype and Code Generation," *Bootcamp on Medium*, May 21, 2024. [En línea]. Disponible: <https://medium.com>. [Accedido: 03-dic-2025].

[2] Mozilla Developer Network, "Semántica HTML y su importancia," *MDN Web Docs*, Jan. 2025. [En línea]. Disponible: <https://developer.mozilla.org/es/docs/Glossary/Semantics>. [Accedido: 03-dic-2025].

[3] W3C Web Accessibility Initiative (WAI), "Web Content Accessibility Guidelines (WCAG) 2.2," *W3C Recommendation*, Oct. 2023. [En línea]. Disponible: <https://www.w3.org/TR/WCAG22/>.



[4] Google Developers, "Responsive Web Design Basics," *web.dev*, Feb. 2024. [En línea]. Disponible: <https://web.dev/responsive-web-design-basics/>.

[5] J. Duckett, *HTML and CSS: Design and Build Websites*, 1st ed. Indianapolis, IN: Wiley, 2011, pp. 45-60.

10. Anexos

Link de git Hub: <https://github.com/ebertr1/Stitch>