

Reporte Técnico de Actividades Práctico-Experimentales Nro. 001

1. Datos de Identificación del Estudiante y la Práctica

Nombre del estudiante(s)	Manuel Santiago Tamayo
Asignatura	Desarrollo Basado en Plataformas
Ciclo	5 A
Unidad	1
Resultado de aprendizaje de la unidad	
Práctica Nro.	001
Título de la Práctica	Implementar un servicio REST con Node.js
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

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).

4. Procedimiento / Metodología Ejecutada

Inicio

1. El docente recuerda brevemente los conceptos de:
 - diseño responsivo,
 - maquetación semántica,
 - accesibilidad básica (contraste, tamaño de tipografía, estructura).
2. Explica el alcance de la práctica:
 - Construir **8 pantallas** relacionadas con un flujo del proyecto (por ejemplo: inicio, login, listado, detalle, formulario, resumen, perfil, ayuda).

Desarrollo

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),
 - 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:

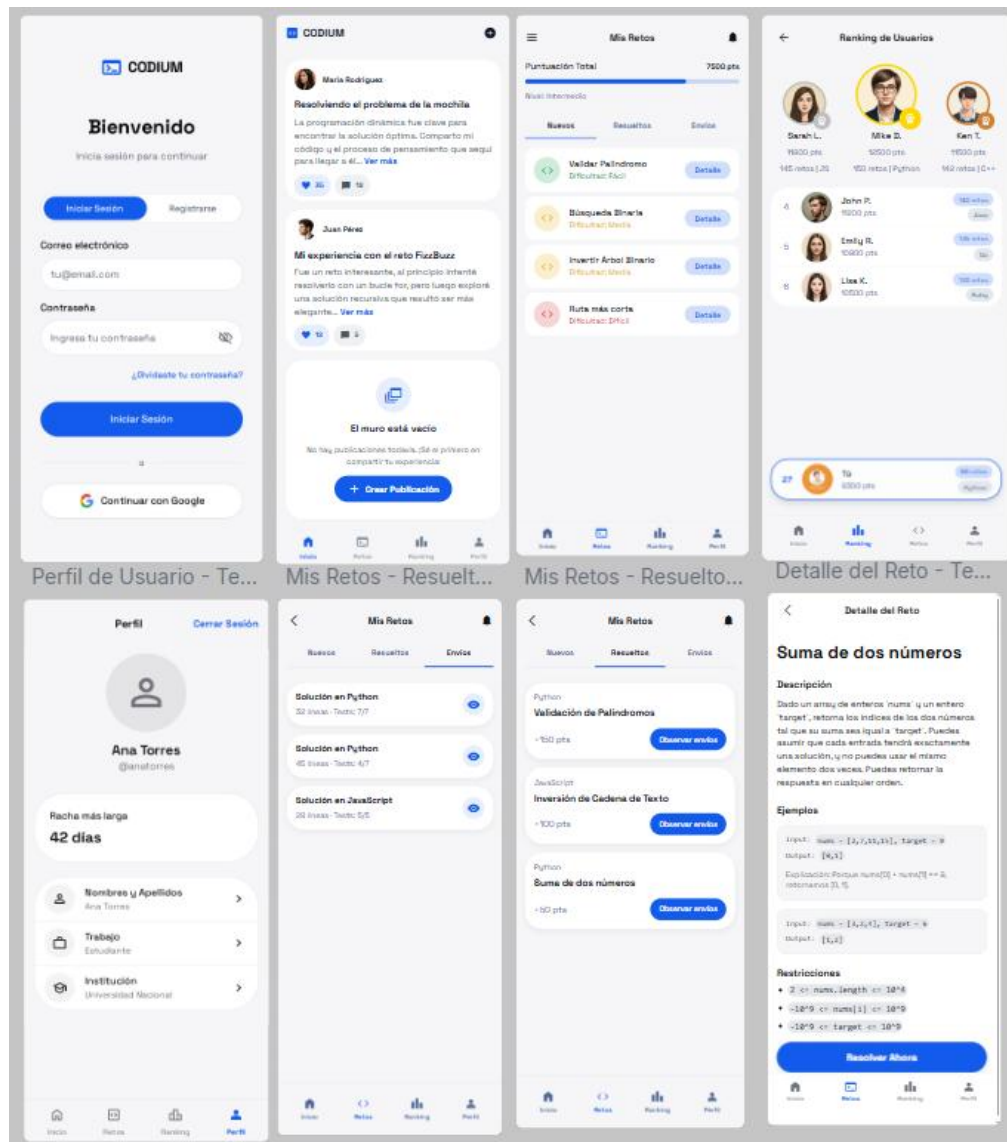
- Breve descripción del flujo que cubren las 8 pantallas.
- Una subsección para cada pantalla:
 - nombre de la pantalla,
 - propósito,
 - componentes principales (botones, formularios, listas, etc.), ◦ captura de pantalla (enlazada o referenciada desde `/docs/img/`).

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

- Prototipo de interfaz con **8 pantallas** implementadas en



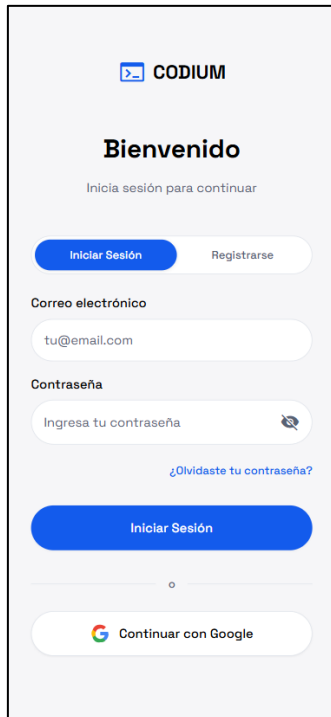
El flujo de trabajo que sigue el prototipo desarrollado corresponde a distintos casos de uso que puede tener el usuario el flujo completo permitirá al usuario conocer el proceso de iniciar sesión, observar el muro de interacción donde los usuarios pueden compartir ideas de solución o dificultades que han tenido al solucionar los retos.

Adicional podrá acceder al panel de los retos para conocer retos nuevos, soluciones enviadas y calificaciones obtenidas por cada uno de los retos que el usuario ha logrado solucionar

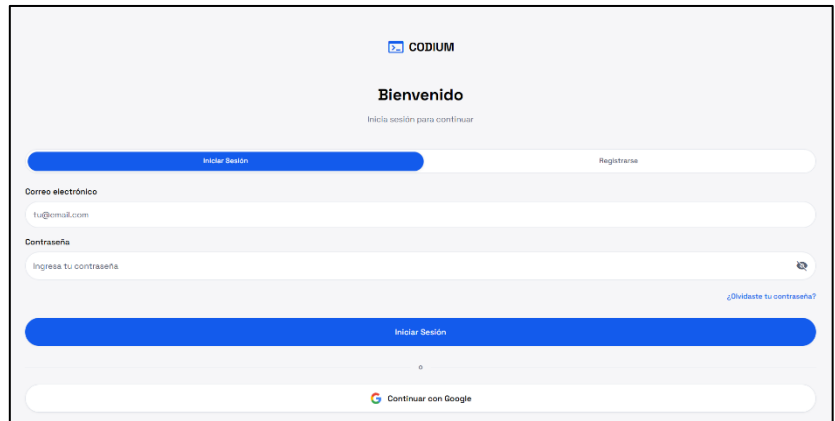
Por último, el cliente podrá acceder a los datos de su perfil si accede al botón de "perfil presente en el panel de navegación ubicado en la parte inferior.

1. Inicio de sesión:

Esta pantalla está diseñada para proporcionar al usuario un formulario de datos necesarios para identificar a cada uno de los usuarios dentro de la aplicación. Por ser prototipo no tiene interacción de entrada de los datos pero al presionar el botón iniciar sesión de color azul dará paso al panel de inicio.



Prototype of the login screen. It features the CODIUM logo at the top, followed by the heading "Bienvenido" and the subtitle "Inicia sesión para continuar". Below this are two buttons: "Iniciar Sesión" (highlighted in blue) and "Registrarse". The form includes a "Correo electrónico" field with the placeholder "tu@email.com" and a "Contraseña" field with the placeholder "Ingresa tu contraseña" and a toggle icon. A link "¿Olvidaste tu contraseña?" is positioned below the password field. At the bottom, there is a blue "Iniciar Sesión" button, a separator line, and a "Continuar con Google" button.



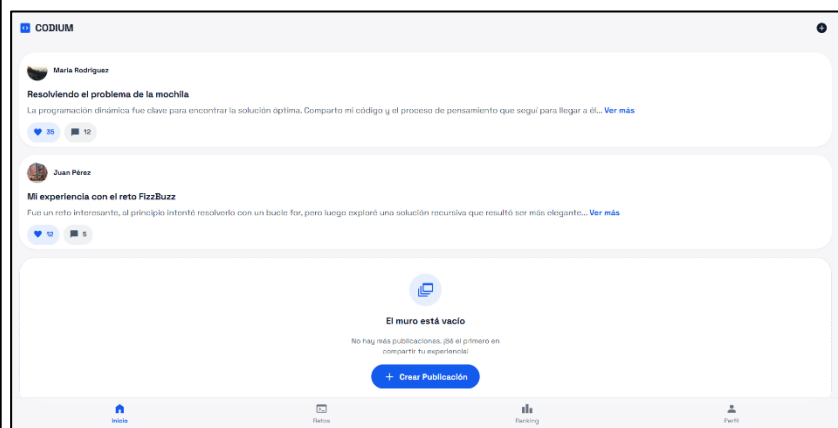
Wireframe of the login screen. It shows the layout with the CODIUM logo, "Bienvenido" heading, and "Inicia sesión para continuar" subtitle. The form fields for email and password are represented by rectangles, with a toggle icon for the password field. A link "¿Olvidaste tu contraseña?" is shown. The bottom section includes a blue "Iniciar Sesión" button, a separator line, and a "Continuar con Google" button.

2. Muro de interacción social:

Esta pantalla está construida para dar al usuario la funcionalidad de compartir sus ideas con el resto de los usuarios de la aplicación compartiendo ayudas para resolver un reto o comentando las dificultades que han tenido en otros determinados retos.



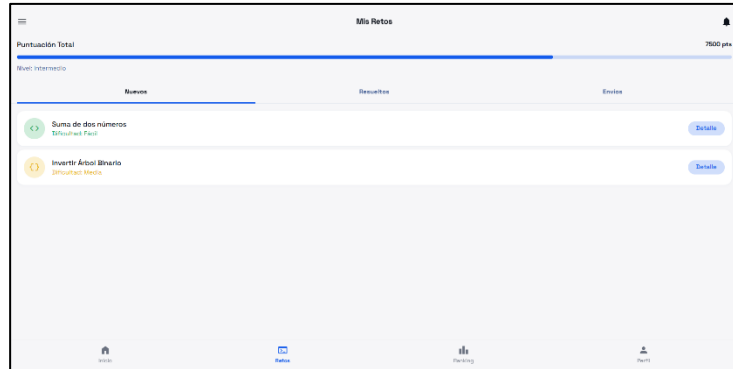
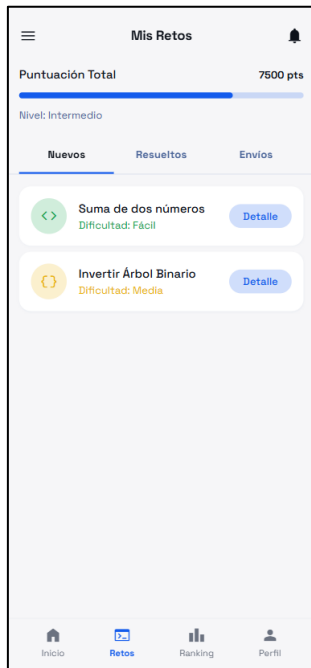
Prototype of the social wall. It shows a list of posts. The first post is by Maria Rodriguez, titled "Resolviendo el problema de la mochila", with 35 likes and 12 comments. The second post is by Juan Pérez, titled "Mi experiencia con el reto FizzBuzz", with 12 likes and 5 comments. Below the posts is a section titled "El muro está vacío" with the text "No hay más publicaciones. ¡Sé el primero en compartir tu experiencia!". At the bottom is a navigation bar with icons for Inicio, Retos, Ranking, and Perfil.



Wireframe of the social wall. It shows the layout of the posts, including the user profile, title, text, and interaction counts. Below the posts is a section titled "El muro está vacío" with the text "No hay más publicaciones. ¡Sé el primero en compartir tu experiencia!". At the bottom is a navigation bar with icons for Inicio, Retos, Ranking, and Perfil.

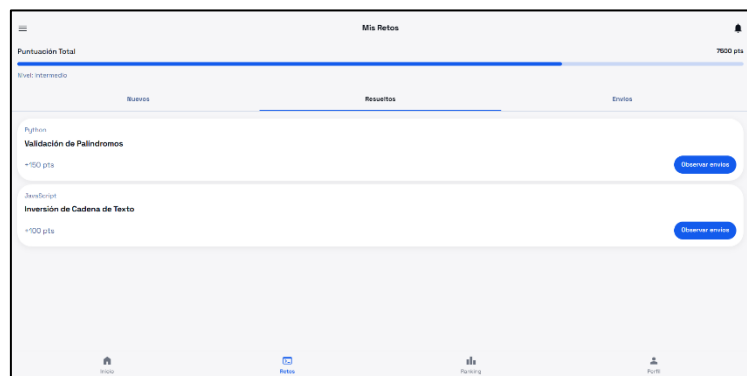
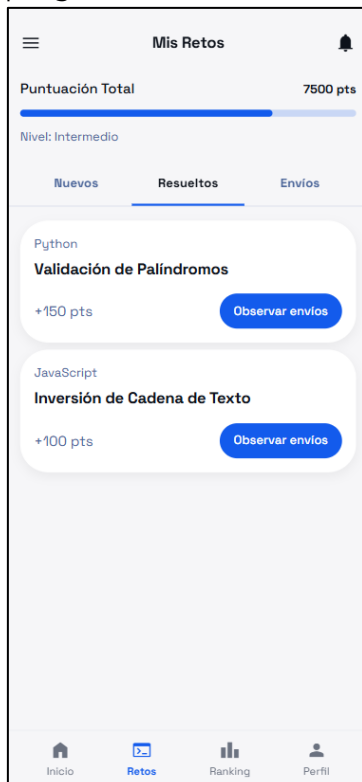
3. Panel de nuevos retos:

Esta pantalla permite a los usuarios entrar a una bandeja de mensajería donde podrá encontrar cada uno de los nuevos retos a los cuales puede dar solución.



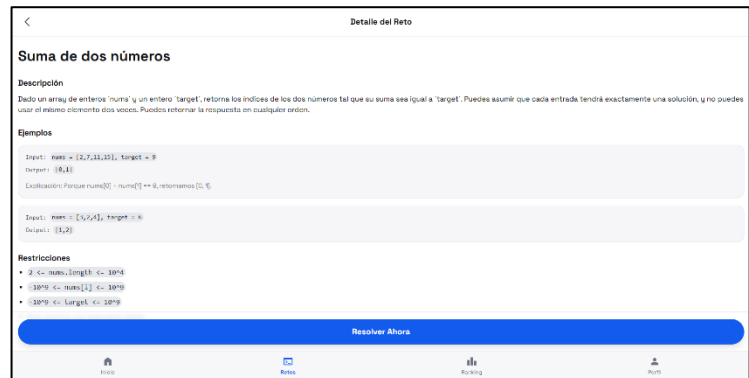
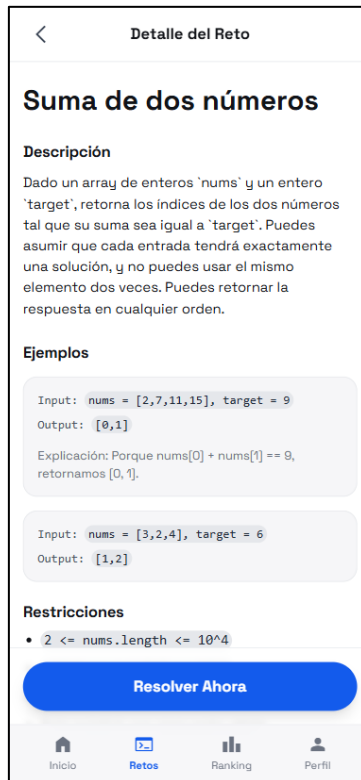
4. Panel de retos resueltos:

La ventana de retos resueltos está destinada a presentar una lista con el historial de todos los retos que ha podido responder nuestro usuario, detallando la información de número de puntos obtenidos y lenguaje de programación utilizado.



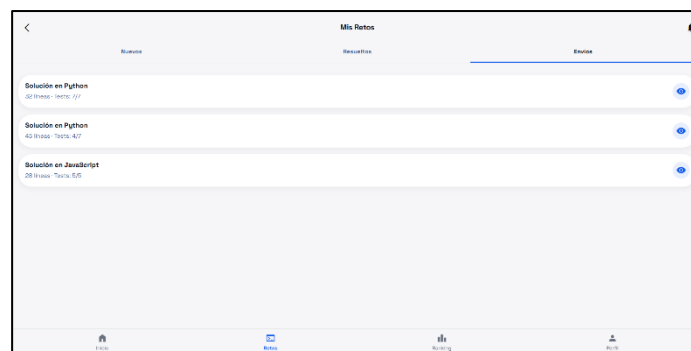
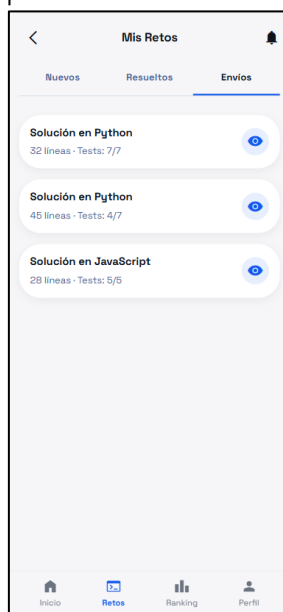
5. Descripción de los retos entrantes:

Después de entrar a la bandeja de entrada de los nuevos retos se puede acceder a una subpestaña que muestra la información referente al reto



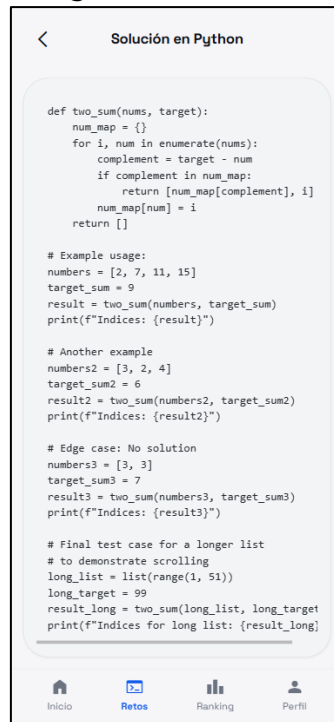
6. Panel de soluciones enviadas para cada reto:

Después de que el usuario elige un reto para dar solución, puede generar la solución de múltiples formas y usando múltiples lenguajes, esta pestaña le permite ver todas las respuestas que ha generado a un reto en específico.



7. Código en lenguaje de programación enviado:

Para cada una de las soluciones que puede generar el usuario se puede observar el código fuente, esta pestaña es la encargada de mostrar dichos códigos.



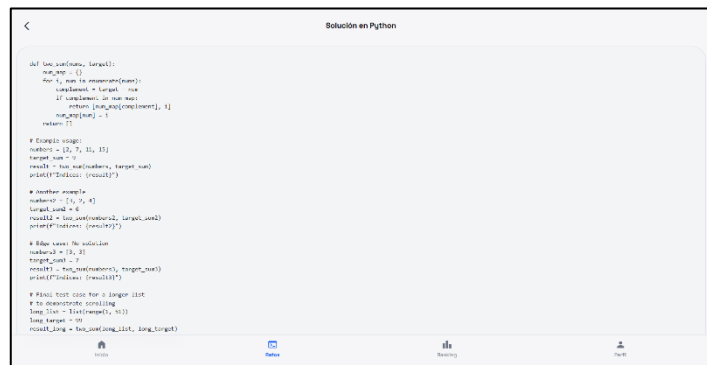
```
def two_sum(nums, target):
    num_map = {}
    for i, num in enumerate(nums):
        complement = target - num
        if complement in num_map:
            return [num_map[complement], i]
        num_map[num] = i
    return []

# Example usage:
numbers = [2, 7, 11, 15]
target_sum = 9
result = two_sum(numbers, target_sum)
print(f"Indices: {result}")

# Another example
numbers2 = [3, 2, 4]
target_sum2 = 6
result2 = two_sum(numbers2, target_sum2)
print(f"Indices: {result2}")

# Edge case: No solution
numbers3 = [3, 3]
target_sum3 = 7
result3 = two_sum(numbers3, target_sum3)
print(f"Indices: {result3}")

# Final test case for a longer list
# to demonstrate scrolling
long_list = list(range(1, 51))
long_target = 99
result_long = two_sum(long_list, long_target)
print(f"Indices for long list: {result_long}")
```



```
def two_sum(nums, target):
    num_map = {}
    for i, num in enumerate(nums):
        complement = target - num
        if complement in num_map:
            return [num_map[complement], i]
        num_map[num] = i
    return []

# Example usage:
numbers = [2, 7, 11, 15]
target_sum = 9
result = two_sum(numbers, target_sum)
print(f"Indices: {result}")

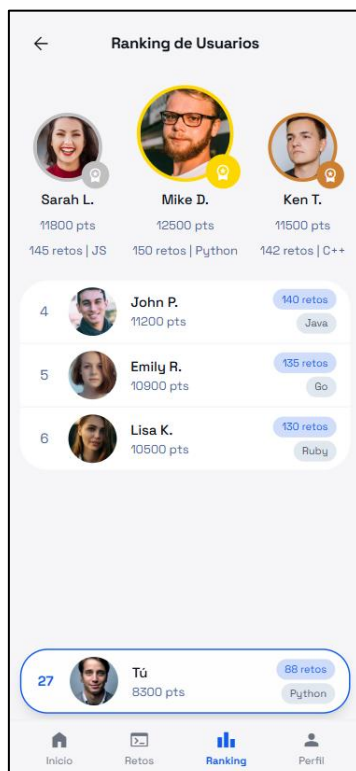
# Another example
numbers2 = [3, 2, 4]
target_sum2 = 6
result2 = two_sum(numbers2, target_sum2)
print(f"Indices: {result2}")

# Edge case: No solution
numbers3 = [3, 3]
target_sum3 = 7
result3 = two_sum(numbers3, target_sum3)
print(f"Indices: {result3}")

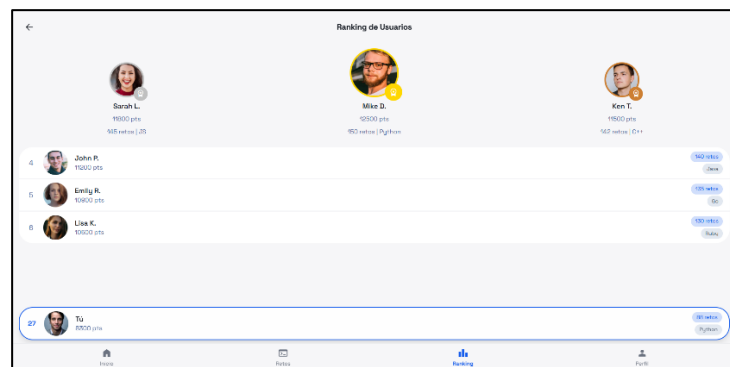
# Final test case for a longer list
# to demonstrate scrolling
long_list = list(range(1, 51))
long_target = 99
result_long = two_sum(long_list, long_target)
print(f"Indices for long list: {result_long}")
```

8. Tabla de ranking de los usuarios:

Como su nombre ya da una pista, está encargada de mostrar el ranking global de todos los usuarios de la aplicación y así mismo mostrar el puesto del usuario corriente.



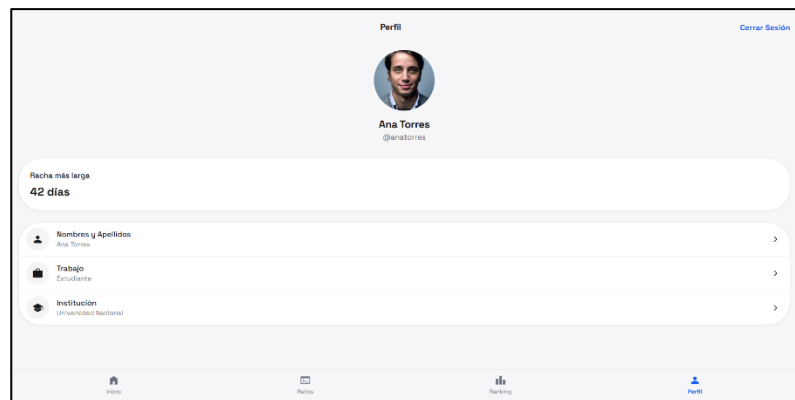
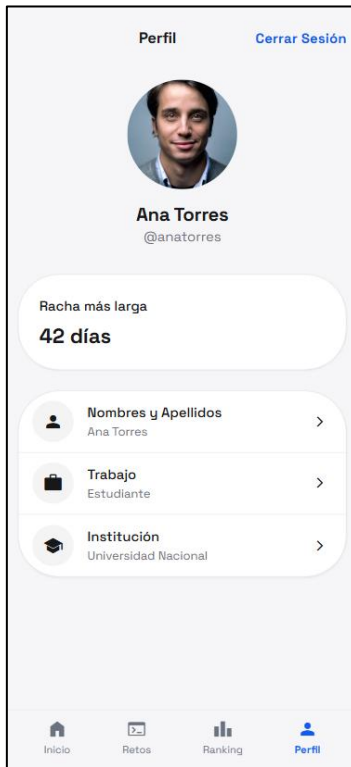
Rank	Username	Points	Retos	Language
1	Sarah L.	11800 pts	145 retos	JS
2	Mike D.	12500 pts	150 retos	Python
3	Ken T.	11500 pts	142 retos	C++
4	John P.	11200 pts	140 retos	Java
5	Emily R.	10900 pts	135 retos	Go
6	Lisa K.	10500 pts	130 retos	Ruby
27	Tú	8300 pts	88 retos	Python



Rank	Username	Points	Retos	Language
1	Sarah L.	11800 pts	145 retos	JS
2	Mike D.	12500 pts	150 retos	Python
3	Ken T.	11500 pts	142 retos	C++
4	John P.	11200 pts	140 retos	Java
5	Emily R.	10900 pts	135 retos	Go
6	Lisa K.	10500 pts	130 retos	Ruby
27	Tú	8300 pts	88 retos	Python

9. Perfil de usuario:

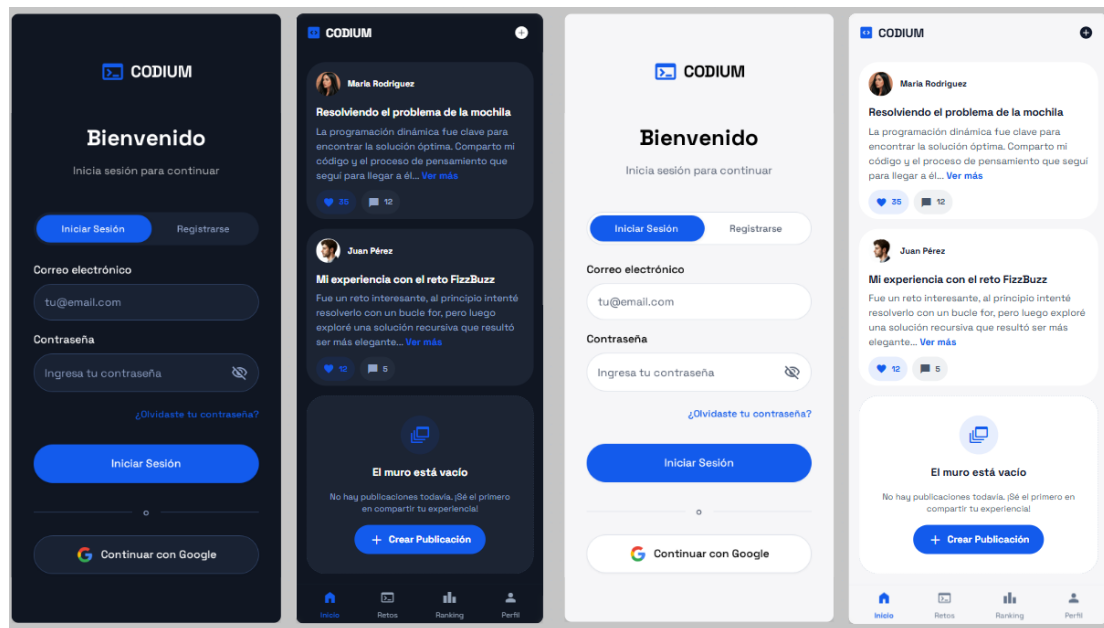
Esta última ventana está encargada de mostrar la información completa de los usuarios, esto contiene los nombres de usuario, los nombres completos del usuario su puntaje total.



➤ HTML/CSS, coherentes con el flujo del proyecto.

```
<!DOCTYPE html>
<html lang="es"><head>
<meta charset="utf-8"/>
<meta content="width=device-width, initial-scale=1.0" name="viewport"/>
<title>Iniciar Sesión / Registrarse - Tema Claro</title>
<link href="https://fonts.googleapis.com" rel="preconnect"/>
<link crossorigin="" href="https://fonts.gstatic.com" rel="preconnect"/>
<link href="https://fonts.googleapis.com/css2?family=Space+Grotesk:wght@400;500;700&display=swap" rel="stylesheet"/>
<link href="https://fonts.googleapis.com/css2?family=Material+Symbols+Outlined:wght@400" rel="stylesheet"/>
<script src="https://cdn.tailwindcss.com?plugins=forms,container-queries"></script>
<script>
  tailwind.config = {
    darkMode: "class",
    theme: {
      extends: {
        colors: {
          "primary": "#135bec",
          "background-light": "#f6f6f8",
          "background-dark": "#101622",
          "text-light": "#586274",
          "border-light": "#e0e4eb"
        },
        fontFamily: {
          "display": ["Space Grotesk", "sans-serif"]
        },
        borderRadius: {
          "DEFAULT": "1rem", "lg": "2rem", "xl": "3rem", "full": "9999px"
        }
      }
    }
  }
</script>
<style>
  .material-symbols-outlined {
    font-variation-settings: 'FILL' 0, 'wght' 400, 'GRAD' 0, 'opsz' 24;
    font-size: 24px;
  }
</style>
```

- Prototipo en Stitch que sirvió de base para la implementación.



6. Preguntas de Control

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

Usar un prototipo en Stitch permite visualizar la estructura y el flujo de las pantallas antes de escribir código. Esto evita retrabajos, facilita validar ideas con el equipo y acelera la toma de decisiones sobre diseño y funcionalidad. Además, permite detectar inconsistencias tempranas y definir mejor la jerarquía visual, reduciendo errores al momento de implementar en HTML/CSS.

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

Las etiquetas semánticas mejoran la accesibilidad, ya que ayudan a lectores de pantalla y otras herramientas a comprender el contenido. También favorecen el posicionamiento SEO, permiten que el código sea más fácil de mantener y facilitan su comprensión por parte de cualquier desarrollador que trabaje en el proyecto. Una estructura semántica también hace que los estilos CSS sean más coherentes y organizados.

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

Se aplicaron principios como el uso de **layouts flexibles (Flexbox/Grid)**, imágenes ajustables al contenedor, **tipografías fluidas**, y puntos de quiebre (breakpoints) para adaptar la distribución del contenido a pantallas pequeñas. También se ajustaron márgenes, tamaños y columnas para que la experiencia sea consistente en móviles, tablets y escritorio.

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

Algunas mejoras posibles incluyen:

- Aumentar el contraste entre texto y fondo para cumplir con WCAG.
- Incrementar tamaños mínimos de fuente y áreas clicables.
- Asegurar una estructura clara con encabezados ordenados (h1, h2, h3).
- Incluir textos alternativos en imágenes.
- Agregar etiquetas ARIA cuando sea necesario.
- Evitar colores como único medio para comunicar información.

Estas mejoras hacen que las pantallas sean más inclusivas para usuarios con discapacidades visuales, motoras o cognitivas.

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

Esta práctica agiliza el desarrollo del frontend porque define desde el inicio la estructura, comportamiento y flujo de las pantallas. Disminuye errores de implementación, permite que el equipo trabaje con una visión unificada y facilita la transición a código real. Además, ayuda a crear componentes reutilizables, definir reglas de diseño y asegurar coherencia en todo el proyecto.

7. Conclusiones

- Realizar prototipos de las vistas con las que contará el sistema nos ayuda a tener una vista previa sobre la funcionalidad que debe tener el mismo, con esto logramos obtener un producto final acorde a lo que nos ha pedido nuestro cliente.
- Tener prototipos con navegación entre las distintas pantallas ayuda al usuario a manera de guía para saber como funcionará el producto final. De esta manera puede solicitar al equipo de desarrollo cambios con lo que respecta a la UI y la UX.