



Tecnológico de Monterrey

Instituto Tecnológico y de Estudios Superiores de Monterrey

Planeación de sistemas de software

Gpo 105

Avance Documento de Requerimientos

Diego Dávila Hernández A01285584

Fernando Morán Fougerat A01284623

Imanol Armando González Solís A00835759

Ramiro Alejandro Garza Villarreal A01178167

Rogelio Garza Rendón A01571384

Campus Monterrey

3 de Mayo del 2025

Índice

1. Requerimientos	3
a. Requerimientos funcionales	3
b. Requerimientos no funcionales	4
2. Historias de usuario	4
a. HU-001 - Historia de usuario 1: Login	4
b. HU-002 - Historia de usuario 2: Resolver problemas de programación	5
c. HU-003 - Historia de usuario 3: Filtrar problemas de programación	5
d. HU-004 - Historia de usuario 4: Ver detalles de un problema	6
e. HU-005 - Historia de usuario 5: Escribir y ejecutar código	6
f. HU-006 - Historia de usuario 6: Validar código con estándares de calidad	6
g. HU-007 - Historia de usuario 7: Recibir recompensas por resolver problemas	7
3. MockUp	8
a. Página principal	8
b. Retos de programación	9
c. Tabla de posiciones	10
d. Recompensas	10
4. Especificación preliminar de la arquitectura de su solución y componentes.	11
a. Arquitectura General de la propuesta:	11
Monolítica o Microservicios, aún se está discutiendo.	11
b. Diagrama de arquitectura:	11

1. Requerimientos

a. Requerimientos funcionales

i. Gestión de Problemas:

1. Scrapping y Transformación:

- a. Uso de CF-Contest-Problem-Scraper para extraer problemas de CodeForces.
- b. Aplicar un script (basado en un LLM) que modifique el nombre y la descripción para adaptarlo a la temática de Tech Mahindra.
- c. Guardar el problema “nuevo” en la base de datos.

2. Tabla de Problemas:

- a. Incluir campos obligatorios: id, nombre, descripción, dificultad (rating) y link al problema original.
- b. Posibilidad de agregar tags, fecha de creación y actualización.

ii. Interfaz de Usuario – Sección Code Challenges:

1. Listado de Problemas:

- a. Mostrar los problemas sorteados o filtrados por dificultad y tags relevantes.

2. Detalle de Problema:

- a. Página específica con la descripción del problema en el panel izquierdo.
- b. Panel derecho con un editor de código integrado.
- c. Botón de “Enviar” para realizar la entrega.

iii. Validación y Evaluación de Envíos:

1. Editor de Código:

- a. Uso de un componente de editor (por ejemplo, Ace, CodeMirror o integración directa con Judge0).
- b. Permitir al usuario escribir, editar y enviar código.

2. Integración con Judge0:

- a. Al enviar el código, se invoca la API de Judge0 para compilar, ejecutar y comparar el resultado con el expected output.
- b. Mostrar mensajes de error o validación en tiempo real.

3. Validación Adicional con SonarQube:

- a. Enviar el código del usuario a SonarQube para análisis de calidad (syntaxis, estilo, seguridad, etc.).

- b. Si el código cumple con los estándares, se muestra un mensaje de “completado” y se marca el problema como resuelto.
 - 4. **Actualización de XP y Currency:**
 - a. Si la entrega es correcta, se actualizan las estadísticas del usuario (XP y currency).
- iv. **Gestión de Usuarios (a futuro o como base):**
 - 1. Registro, login y gestión de perfiles.
 - 2. Registro de progreso y problemas completados.

b. Requerimientos no funcionales

- i. **Usabilidad:**
 - 1. Interfaz intuitiva y responsiva, especialmente en el editor y en la presentación de estadísticas.
- ii. **Performance:**
 - 1. Respuesta rápida al cargar listados de problemas y al enviar evaluaciones de código.
- iii. **Seguridad:**
 - 1. Autenticación robusta y protección de datos de usuario.
 - 2. Validación y sanitización de entradas en el editor de código.
- iv. **Escalabilidad:**
 - 1. Arquitectura modular que permita la incorporación de futuras funcionalidades (Tech Trivia, rewards, etc.).
- v. **Mantenibilidad:**
 - 1. Código y documentación clara para facilitar futuras iteraciones y cambios.

2. Historias de usuario

a. HU-001 - Historia de usuario 1: Login

Título: Login de usuario

Descripción: Como usuario y trabajador de la empresa, quiero poder ingresar a la plataforma

Criterios de aceptación:

- El usuario debe ingresar su correo y contraseña de la empresa
- Se debe mostrar un mensaje de error si el correo o la contraseña son incorrectos o inválidos

b. HU-002 - Historia de usuario 2: Resolver problemas de programación

Título: Problemas de programación

Descripción: Como usuario, quiero poder acceder a los problemas de programación para resolverlos

Criterios de aceptación:

- El usuario debe haber ingresado correctamente
- El usuario debe entrar a la sección de “Problemas de programación”
- El usuario debe seleccionar el problema que quiere resolver

c. HU-003 - Historia de usuario 3: Filtrar problemas de programación

Título: Filtros en la lista de problemas

Descripción: Como usuario, quiero poder filtrar los problemas de programación por dificultad y etiquetas para encontrar los que más se adapten a mi nivel.

Criterios de aceptación:

- El usuario debe ver opciones de filtro por dificultad y etiquetas en la sección de problemas.
- Al seleccionar un filtro, la lista de problemas debe actualizarse automáticamente.
- Debe existir la opción de quitar los filtros para volver a ver todos los problemas.

d. HU-004 - Historia de usuario 4: Ver detalles de un problema

Título: Visualización del problema

Descripción: Como usuario, quiero ver la descripción completa de un problema antes de intentar resolverlo.

Criterios de aceptación:

- Al hacer clic en un problema, se debe abrir una página con su descripción.
- La página debe mostrar la dificultad, las etiquetas y el enlace al problema original en CodeForces.
- Debe haber un botón para regresar a la lista de problemas.

e. HU-005 - Historia de usuario 5: Escribir y ejecutar código

Título: Editor de código integrado

Descripción: Como usuario, quiero escribir mi solución en un editor de código integrado y ejecutarla para comprobar si funciona correctamente.

Criterios de aceptación:

- El usuario debe poder escribir código en un editor dentro de la página del problema.
- Al hacer clic en "Ejecutar", el código debe enviarse a Judge0 para su evaluación.
- Los resultados de la ejecución (errores o salida esperada) deben mostrarse en la interfaz.

f. HU-006 - Historia de usuario 6: Validar código con estándares de calidad

Título: Evaluación de calidad del código

Descripción: Como usuario, quiero que mi código sea analizado en cuanto a estilo y buenas prácticas para mejorar su calidad.

Criterios de aceptación:

- Al enviar el código, este debe ser evaluado por SonarQube.
- Si hay errores de calidad, el usuario debe recibir un mensaje con las recomendaciones.
- Si el código cumple con los estándares, se debe marcar como “completo”.

g. HU-007 - Historia de usuario 7: Recibir recompensas por resolver problemas

Título: Sistema de recompensas

Descripción: Como usuario, quiero recibir XP y currency cuando se resuelve correctamente un problema para motivarme a seguir aprendiendo.

Criterios de aceptación:

- Si el código es válido y pasa todas las pruebas, se debe actualizar la XP y currency del usuario.
- El usuario debe recibir una notificación de éxito tras resolver un problema.
- El progreso del usuario debe actualizarse en su perfil.

h. HU-008 - Historia de usuario 8: Ver avances

Título: Dashboard avances

Descripción: Como usuario quieres ver el avance de personas en tu organización, y tu propio.

Criterios de aceptación:

- Ver tu porcentaje de aceptación, problemas resueltos, y promedio de tu equipo.
- Poder ver información de usuarios compañeros.

3. MockUp

a. Página principal



Beneficios para Mahindras



Retos Técnicos

Desafíos diseñados por nuestros expertos, alineados con los proyectos y tecnologías de Tech Mahindra.



Desarrollo Profesional

Gana MCoinS y experiencia que se reflejarán en tu evaluación de desempeño y oportunidades de crecimiento.



Reconocimientos

Canjea tus MCoinS por beneficios exclusivos como días libres, cursos premium y merchandising de la empresa.

Top Mahindras

Los desarrolladores más destacados de esta semana en nuestra empresa.

1°



Developer_1

Nivel 26 • 45 desafíos

5000 XP

900 MC

2°



Developer_2

Nivel 25 • 38 desafíos

4000 XP

800 MC

3°



Developer_3

Nivel 24 • 31 desafíos

3000 XP

700 MC

4°



Developer_4

Nivel 22 • 27 desafíos

2000 XP

600 MC

5°



Developer_5

Nivel 21 • 25 desafíos

1000 XP

500 MC

b. Retos de programación

TM

Retos de Programación

Tabla de Posiciones

Recompensas

2,500 MC

Nivel 12

D

¡Duo ha desaparecido!

Duo ha desaparecido, pero ha dejado un rastro de su paso. Ayúdanos a encontrarlo reconstruyendo el camino que ha seguido.

★★★ 30 MC

Desafío Diario

MISSING

DUO

BY YOUR CLASH GAMES PROGRAMACIÓN

RENTA DUEÑO DE CLASH GAMES

¡¡¡¡¡

✓ Inicia sesión

📖 Resuelve un problema

📢 Comparte dos problemas

Misiones Diarias

✓ Inicia sesión 5 días

📖 Resuelve 5 problemas

🏆 Gana un torneo

Misiones Semanales

Todos los problemas

Estatus	Problema	Puntos	Tags	Dificultad
✓	Problema 1	20 MC	nice developer tag	★
✓	Problema 2	25 MC	nice developer tag	★★
✓	Problema 3	30 MC	nice developer tag	★★★
	Problema 4	35 MC	nice developer tag	★★★★
	Problema 5	40 MC	nice developer tag	★★★★★

c. Tabla de posiciones

TM

Retos de Programación

Tabla de Posiciones

Recompensas

2,500 MC

Nivel 12

D

Leaderboard

Rank	User	Score	Problems Solved
	A Alice	1500	120
	B Bob	1450	115
	C Charlie	1400	110
4	D David	1350	105
5	E Eve	1300	100

d. Recompensas

TM

Retos de Programación

Tabla de Posiciones

Recompensas

2,500 MC

Nivel 12

D

Rewards

Discover

My rewards

My mahindricks

Social

Ranking

About

Profile


Swag

Gift Cards

More


Earn cool stuff!

Top picks for you. Updated daily.



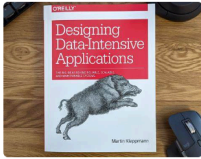
Mochila

4500 mahindricks



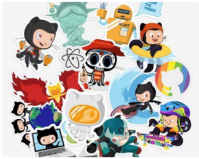
Taza café

1000 mahindricks



Libro Designing Data Intensive A...

10000 mahindricks



Stickers Github

100 mahindricks

Made for You

Restocked items

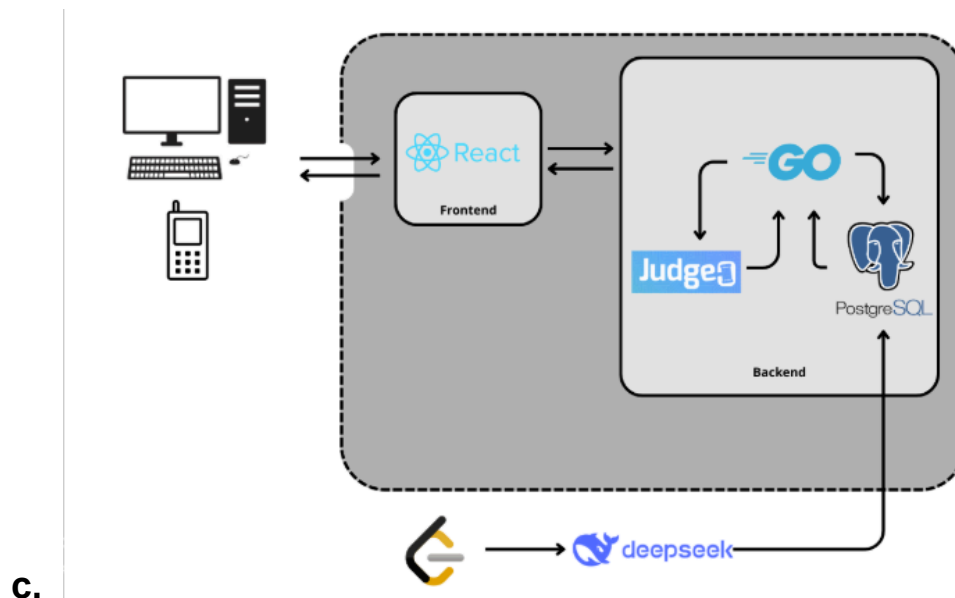
11

4. Especificación preliminar de la arquitectura de su solución y componentes.

a. Arquitectura General de la propuesta:

Nuestra solución tendrá una arquitectura monolítica, ya que será un proyecto de pequeña escala y no tenemos planeado mucha escalabilidad.

b. Diagrama de arquitectura:



d. Frontend: React

e. Backend: Go

f. Base de datos: PostgreSQL

g. Servicios externos: Judge0, LeetCode o aplicaciones similares y un modelo LLM para el proceso de problemas.