



Instituto Tecnológico y de Estudios Superiores de Monterrey
Campus Monterrey

TC3004B
Planeación de Sistemas de Software (Grupo 101)

Plan de proyecto

Profesores:

Ing. Alberto Ruiz García
Ing. Elvia Itzamná Rosas Herrera
Ing. Joel Fernando Sifuentes Ortíz
Ing. Luis Carlos Aguirre Mora
Ing. Omar Daniel Rodríguez Salinas

Equipo #2 | Integrantes:

INNOVATEC

Andrés Fuentes Alanís	A01177466
Jorge Eduardo de León Reyna	A00829759
Andrea Catalina Fernández Mena	A01197705
Mario Isaí Robles Lozano	A01197707
Gerardo Mora Beltrán	A00827128
Oscar Alejandro Reyna Monreal	A01570385

Junio 2023 | Monterrey, N.L.

Índice

Índice	1
0. Historial de versiones	2
1. Visión del proyecto	3
1.1 Problema a resolver	3
1.2 Propósito y propuesta de solución	3
1.3 Alcance del proyecto	3
1.4 Visión del proyecto	4
2. Work Breakdown Structure	5
3. Plan de comunicación	7
3.1 Descripción de stakeholders e involucrados en el proyecto	7
3.2 Plan de comunicación con stakeholders e involucrados en el proyecto	7
4. Estrategia de desarrollo	8
5. Cronograma del proyecto	11
6. Estrategia de requisitos	12
7. Plan de recursos	13
8. Plan de gestión de riesgos	15
9. Plan de calidad	19
9.1 Casos de prueba	19
9.2 Roles y responsabilidades en el plan de calidad	22
9.3 Prácticas de calidad y productos de trabajo	24
9.4 Métricas del proyecto	28
9.5 Estrategia de Pruebas	28
9.6 Entregables de Calidad	29
9.7 Tipos de severidades	29
9.8 Tipos de defectos	30
10. Plan de valor ganado	31

0. Historial de revisiones

Versión	Fecha	Responsable(s)	Descripción
V 1.0	2/Mar/23	Jorge Mario Andrés	Visión del proyecto Work Breakdown Structure Plan de comunicación Estrategia de desarrollo
V1.1	6/Mar/23	Andrea Gerardo Oscar	Cronograma del Proyecto Estrategia de requisitos
V2.0	9/Mar/23	Andrés Mario Andrea Gerardo	Plan de recursos Plan de gestión de riesgos Plan de valor ganado
V2.1	13/Mar/23	Andrés Mario Andrea Jorge Gerardo Oscar	Plan de calidad Casos de prueba
V2.2	16/Mar/23	Andrés Mario Jorge	Revisión de redacción y formato del documento
V3.0	22/May/23	Andrés Fuentes Jorge De León Catalina Fernández Mario Robles Gerardo Mora Oscar reyna	Edición de documento en base a nuevos requerimientos y necesidades.

1. Visión del proyecto

1.1 Problema a resolver

WizeLine es una empresa consultora de software que se distingue por su amplio conjunto de sistemas y procesos operativos. Entre estos se encuentra su plataforma de gestión de perfiles técnicos y habilidades de sus colaboradores. El propósito principal de esta herramienta es brindar un mayor control sobre el equipo de colaboradores de WizeLine, permitiendo conocer el proyecto en el que cada uno de ellos está involucrado. Además de simplificar su día a día, ya que la plataforma integra herramientas de inteligencia artificial, potenciando así su eficiencia.

Las problemáticas identificadas en este proceso son las siguientes:

- Los usuarios invierten mucho tiempo en la creación y actualización de sus perfiles en la plataforma, ya que deben llenar numerosos campos con información detallada acerca de su perfil técnico, habilidades, conocimientos y experiencia.
- Los usuarios carecen de una guía que les permita complementar sus habilidades técnicas y conocimientos, lo que limita su capacidad para avanzar en su carrera profesional.
- Los usuarios / project managers se toman mucho tiempo en desarrollar los documentos de requerimientos. Con la inteligencia artificial, se buscará crear una base de requerimientos para que puedan empezar a trabajar en ello.

1.2 Propósito y visión de solución

Se propone desarrollar una plataforma que utilice herramientas de IA para simplificar el proceso de creación y actualización de perfiles de usuario. Con el apoyo de la IA, se podrá mejorar y enriquecer la descripción del perfil de manera más eficiente. Además, se propone utilizar las mismas herramientas de IA para crear una funcionalidad que sugiera al usuario tecnologías, herramientas y temas complementarios basados en las habilidades técnicas y conocimientos que ha ingresado en la plataforma previamente. Esto permitiría a los usuarios mejorar continuamente sus habilidades y conocimientos, fomentando su crecimiento profesional. Finalmente, la creación de requerimientos de un sistema a partir de una descripción proporcionada por el administrador.

1.3 Alcance del proyecto

Para definir el alcance del proyecto de creación de perfiles con ayuda de *Generative AI*, se utilizará la metodología SMART (Specific, Mesurable, Achivable, Relevant, Time-Bound) para establecer objetivos claros y alcanzables. El alcance definido que se ha determinado para este proyecto es el siguiente:

- Crear una plataforma web con IA que eficiente y mejore la creación de perfiles dentro del sistema de asignación de personal a proyectos, proporcione un plan de aprendizaje personalizado basado en su experiencia en programación y cree requerimientos a partir de una descripción.

- Reducir el tiempo que los empleados tardan en completar sus perfiles entre un 50% y 90%. Generar rutas de aprendizaje personalizadas para el 100% de los empleados. Reducir el tiempo en el que los empleados desarrollan un documento de requerimientos por un 35%.
- Optimizar la forma en que se presentan los perfiles de los empleados y requerimientos, lo que se traducirá en un aumento en las ventas de Wizeline.
- Lanzar la plataforma de forma completa en un plazo de 10 semanas a partir de la fecha de inicio del proyecto, incluyendo pruebas y lanzamientos iterativos para garantizar su funcionalidad y efectividad.

1.4 Visión del proyecto

La visión del proyecto es crear una plataforma que aproveche las herramientas de IA para simplificar el proceso de creación y actualización de los perfiles de los usuarios. La plataforma también utilizará la IA para complementar y hacer más robusta la descripción del perfil del usuario. Además, la plataforma contará con una herramienta que, a partir de las habilidades técnicas y conocimientos del usuario, sugiere otras tecnologías, herramientas y temas que pueden complementar sus habilidades actuales. Así mismo, la creación de requerimientos con una descripción fácil y entendible.

2. Work Breakdown Structure

1. Planeación

- 1.1. Definición de necesidades del cliente.
- 1.2. Definición del alcance del proyecto.
- 1.3. Definición de requerimientos de software.
- 1.4. Definir tiempo límite de trabajo.
- 1.5. Definir los recursos necesarios.
- 1.6. Definir el costo del proyecto.
- 1.7. Definir la matriz de riesgos.

2. Diseño

- 2.1. Definir la arquitectura del software.
- 2.2. Definir los objetos requeridos para la base de datos.
- 2.3. Definir los parámetros a recolectar desde el front-end.
- 2.4. Definir los parámetros que se van a enviar a la base de datos.
- 2.5. Definir los parámetros a usar con la inteligencia artificial.
- 2.6. Definir las autorizaciones de los usuarios.
- 2.7. Definir las historias de usuario.
- 2.8. Desarrollar el diseño de la aplicación.
- 2.9. Definir las funcionalidades de aplicación.

3. Desarrollo

- 3.1. Creación de API para comunicación de aplicación web con la inteligencia artificial.
- 3.2. Creación de sistema para lectura de la información desde LinkedIn.
- 3.3. Creación de sistema para redacción del perfil y generación del Roadmap.
- 3.4. Creación de la base de datos para los usuarios.
- 3.5. Creación de API para comunicación de aplicación web con base de datos de usuarios.
- 3.6. Generar la lista de todos los usuarios en la aplicación web.
- 3.7. Creación de sistema para agregar usuarios.
- 3.8. Creación de sistema para editar la información de los usuarios.
- 3.9. Creación de sistema para borrar usuarios.
- 3.10. Creación de la base de datos para las órdenes.
- 3.11. Creación de API para comunicación de aplicación web con base de datos de órdenes.
- 3.12. Generar la lista de todos los proyectos en la aplicación web.
- 3.13. Creación de sistema para generar proyectos.
- 3.14. Creación de sistema para marcar proyectos como terminados.
- 3.15. Generar la lista de los proyectos de un cliente.
- 3.16. Generar la lista de los proyectos de un colaborador.
- 3.17. Creación de función para ver la información de un colaborador de una orden.
- 3.18. Generar la función de filtrado en la lista de proyectos y usuarios.
- 3.19. Generar sistema de inicio de sesión.
- 3.20. Generar sistema de cerrar sesión.
- 3.21. Generar sistema para editar información del perfil.
- 3.22. Generación de requerimientos de software sencillos en base a una descripción de usuario.
- 3.23. Realizar pruebas sprint final
- 3.24. Realizar pruebas de aceptación.

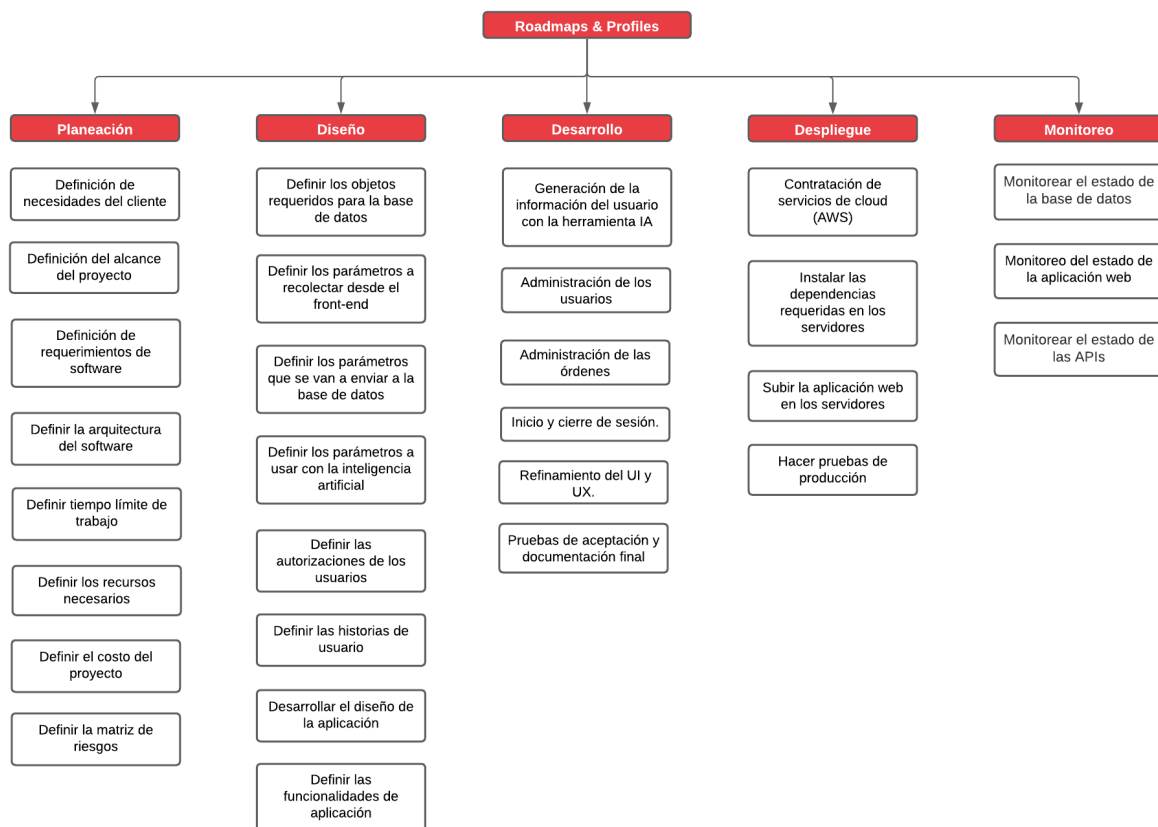
3.25. Realizar evaluación de calidad del sistema.

4. Despliegue

- 4.1. Contratación de servicios de cloud (AWS).
- 4.2. Instalar las dependencias requeridas en los servidores.
- 4.3. Subir la aplicación web en los servidores.
- 4.4. Hacer pruebas de producción.

5. Monitoreo

- 5.1. Monitoreo del estado de la base de datos.
- 5.2. Monitoreo del estado de la aplicación web.
- 5.3. Monitoreo del estado de las APIs.



3. Plan de Comunicación

3.1 Descripción de stakeholders e involucrados en el proyecto

Stakeholders e involucrados en el proyecto	
Stakeholder	Rol/Descripción
Socio Formador	Cliente principal. Su rol dentro del proyecto es exponer las necesidades principales del proyecto y dar revisión y seguimiento a las propuestas presentadas.
Equipo de desarrollo	Equipo principal de desarrollo cuyo rol principal es la planeación, diseño, desarrollo y evaluación del proyecto.
Stakeholders de apoyo	Su función consiste en brindar retroalimentación y apoyo al equipo de desarrollo.
Profesores	Su rol principal consiste en dar apoyo académico y técnico a los equipos de desarrollo, así como también facilitar la comunicación con los diferentes stakeholders del proyecto.

3.2 Plan de comunicación con stakeholders e involucrados en el proyecto

El plan de comunicación consiste en los siguientes elementos:

- **Comunicación con socio formador:** la comunicación con el socio formador se llevará de manera constante en lapsos de 2 semanas (duración del sprint) en donde se compartirán los avances, dudas y posibles problemas.
- **Comunicación entre el equipo de desarrollo:** la comunicación entre los miembros del equipo de desarrollo se llevará a cabo a través de 3 medios principales: Slack, WhatsApp y sesiones en clase.
- **Stakeholders de apoyo:** la comunicación con los stakeholders secundarios se llevará a cabo por medio de los canales de Slack designados de forma periódica, cuando se tengan dudas específicas o presentaciones.
- **Profesores:** La comunicación se llevará a cabo en las sesiones de clase presenciales y digitales, correo electrónico y WhatsApp.

4. Estrategia de desarrollo

La estrategia de desarrollo principal se basa en el seguimiento de metodologías ágiles, de manera que las 10 semanas que se disponen para esta tarea serán divididas en sprints con una duración de 2 semanas, dando como resultado 5 sprints destinados al desarrollo del software. A continuación se describen los puntos claves de la dinámica de desarrollo y gestión de trabajo:

- **Planificación de sprints:** El equipo debe reunirse al comienzo de cada sprint para planificar el trabajo que se realizará durante las próximas semanas.
- **Reuniones diarias de Scrum:** El equipo debe tener reuniones diarias de Scrum para discutir el progreso del trabajo y cualquier problema que surja.
- **Revisión del sprint:** Al final de cada sprint, el equipo de desarrollo debe reunirse para revisar el trabajo completado y demostrar cualquier nuevo software creado y el avance en el plan de valor ganado.
- **Retrospectiva del sprint:** Después de la revisión del sprint, el equipo debe tener una retrospectiva para discutir lo que funcionó bien durante el sprint, lo que no funcionó y cómo se pueden mejorar las cosas en el futuro.
- **Gestión del backlog:** El equipo de desarrollo debe mantener un backlog de todas las historias de usuario que se deben completar y cuando se debe hacerlo.

A continuación se muestra la distribución del desarrollo por sprints:

Sprint	HU	Requerimiento	Valor Ganado
Sprint 1 (27 mar - 14 abr) Generación de la información del usuario con la herramienta IA y se muestra en el UI y DB.	HU20101	RF201	20%
	HU20301	RF203	40%
	HU20401	RF204	40%
Sprint 2 (17 abr - 28 abr) Administración de los usuarios, se crean, modifican, eliminan de la BD y se muestra la lista en el UI.	HU30201	RF302	10%
	HU30202		35%
	HU30203		20%
	HU30204		35%
Sprint 3 (1 may - 19 may) Administración de las órdenes, se crean, modifican, eliminan de la BD y se muestra la lista en el UI. Los clientes y colaboradores pueden ver la lista de órdenes a las que fueron asignados.	HU30301	RF303	15%
	HU30302		17.5%
	HU30303		17.5%
	HU40101	RF401	10%
	HU40102		10%

	HU40103		15%
	HU20601	RF206	10%
	HU20602		5%
Sprint 4 (22 may - 2 jun) Inicio y cierre de sesión, se edita la información de los perfiles de usuario, creación de dashboard general para colaboradores y administradores. Refinamiento del UI y UX.	HU10101	RF101	8%
	HU10102		8%
	HU10103		4%
	HU10201	RF102	8%
	HU10202		8%
	HU10301	RF103	12%
	HU10302		12%
	HU10303		8%
	HU20501	RF205	16%
	HU30101	RF301	8%
	HU30102		8%
Sprint 5 (5 jun - 16 jun) Extracción de datos de un CV en PDF. Pruebas de aceptación y documentación final	HU20201	RF202	30%
	Pruebas, evaluaciones y presentaciones finales		70%

* **Descripción de historias de usuario:**  DocumentoRequerimientos

Además, se plantean las estrategias de desarrollo generales para el proyecto:

- **Planeación:** Definimos el alcance del proyecto, así como los objetivos y metas que queremos lograr. Identificamos los requerimientos de la plataforma y las necesidades de los usuarios finales. Además de definir un equipo de trabajo, definir roles y responsabilidades para cada uno de los miembros. Finalmente, establecemos un plan de trabajo con una calendarización (Gantt).
- **Diseño:** Diseñamos la arquitectura de la plataforma, incluyendo los flujos de trabajo y la interfaz de usuario, obviamente tomando en cuenta la inteligencia artificial para el completado rápido de tareas, y la definición de los algoritmos de recomendación para el roadmap personalizado de cada empleado. Crearemos un diseño que permita la escalabilidad de la plataforma, considerando un posible aumento de usuarios y datos.
- **Desarrollo:** Ya terminando el diseño, empezaremos a implementar el sistema. Nosotros deberemos de seguir prácticas de codificación estándar y buenas prácticas de seguridad trabajando de forma iterativa, implementando la metodología de desarrollo ágil, para asegurarnos de que la plataforma se construya según los diseños previamente aprobados.

- **Despliegue:** Ya que hayamos terminado de desarrollar la plataforma, esta se subirá a un servidor.
- **Monitoreo:** Por último, revisaremos el rendimiento y estabilidad de la plataforma. Cualquier detalle que llegará a surgir deberá de ser modificado.

5. Cronograma del proyecto

Liga al cronograma: [!\[\]\(21199eb166cc97331a0c54c649195dcc_img.jpg\) Diagrama de Gantt](#)

6. Estrategia de requisitos

El proceso de trabajo comenzó con la definición del proyecto que se va a desarrollar. Los proyectos fueron explicados por la maestra. Una vez explicados, se eligió uno de ellos y se empezaron a idear maneras en las que se puede desarrollar. Se hicieron varias preguntas sobre el mismo. Gracias a esta reunión, se definió con mayor precisión los objetivos y requerimientos del proyecto.

Después, se trabajó en el desarrollo de una propuesta detallada que incluía los objetivos del proyecto. Se presentó esta propuesta al CTO de Wizeline, quien evaluó la presentación y proporcionó retroalimentación valiosa. Con base en sus comentarios, se mejoró nuestra propuesta, incluyendo un diseño más detallado y explicativo. Una vez que tuvimos una propuesta final, se definió los requerimientos del sistema y sus criterios de aceptación.

7. Plan de recursos

Equipo de desarrollo:

Nombre	Andrés Fuentes Alanís
Rol	Product Manager
Responsabilidades	Gestión del equipo en los Sprints
Información de contacto	andres.fuentes@afared.com

Nombre	Jorge Eduardo de León Reyna
Rol	Product Owner
Responsabilidades	Contacto con los Stakeholders
Información de contacto	a00829759@tec.mx

Nombre	Mario Isaí Robles Lozano
Rol	Quality Manager
Responsabilidades	Asegurar la calidad de los productos
Información de contacto	a01197707@tec.mx

Nombre	Andrea Catalina Fernández Mena
Rol	Delivery Manager
Responsabilidades	Seguimiento con el cliente
Información de contacto	katy.fernandezmena@gmail.com

Nombre	Gerardo Mora Beltrán
Rol	Tech Leader
Responsabilidades	Gestión del desarrollo del software
Información de contacto	a00827128@tec.mx

Nombre	Oscar Alejandro Reyna Monreal
Rol	Data Manager
Responsabilidades	Gestión de los datos utilizados
Información de contacto	a01570385@tec.mx

Infraestructura Tecnológica:

- 6 ordenadores portátiles de alta gama.
- Servidor de aplicaciones en la nube (AWS).
- Herramientas de desarrollo de software (IDE, repositorio de código, herramientas de colaboración).

Tiempo:

- 10 semanas.

Plan de recursos:

- Adquisición de infraestructura tecnológica:
 - Compra de servidor de Amazon Web Services Lightsail para subir desarrollo web.
 - API de OpenAI, para hacer consultas a la inteligencia artificial y obtener respuestas de esta misma.

Producto	Rango de Costo
AWS	\$10 USD al mes
API Open AI	\$10 USD dependiendo del número de tokens utilizados. Casi creemos que 10 USD al año.

- Asignación de tareas y seguimiento del progreso:
 - Se asignan tareas específicas a cada miembro del equipo de desarrollo y se hará un seguimiento cada dos días del progreso para garantizar que el proyecto avance según lo planeado.
- Control de calidad:
 - Los cambios serán revisados por el equipo para asegurar que se hayan realizado con éxito. Si el equipo aprueba el cambio, se sube a la rama de desarrollo.

8. Plan de gestión de riesgos

Categorías del riesgo

Los riesgos serán clasificados dentro de una de las tres clasificaciones:

- **Entrega del proyecto:** Errores en la planificación, falta de recursos, falta de comunicación, entre otros.
- **Estratégico:** Riesgos causados por eventos que afectan el modelo de negocios y el alcance de las metas propuestas.
- **Operacional:** Riesgos causados por errores humanos, tecnológicos y de procesos internos que afectan al producto final.

Tabla de descripción y clasificación de riesgos

	Riesgo	Descripción del riesgo	Categoría del riesgo
RI001	Incumplimiento con la entrega final del proyecto	No se entrega a tiempo el proyecto o se entrega con una mala calidad debido a múltiples factores, como pérdida del tiempo, mal manejo del tiempo, entre otros.	Entrega del proyecto
RI002	Error en comunicación con los Stakeholders	Hubo un mal entendido con la petición o la necesidad del socio formador.	Entrega del proyecto
RI003	Alcance del proyecto muy ambicioso para el tiempo disponible	La descripción del proyecto promete muchas cosas, ya sea que son inalcanzables o realistas de desarrollar por el tiempo del desarrollo del proyecto.	Entrega del proyecto
RI004	Corrupción de los archivos que conforman el proyecto	Los archivos conteniendo el código del proyecto o parte del proyecto se corrompieron en algún momento del desarrollo y son inutilizables para el producto final.	Operacional
RI005	Retraso con el acceso a la API de OpenAI	El socio formador se retrasa con la entrega de la herramienta de inteligencia artificial a usar para la implementación del proyecto.	Estratégico
RI006	Retraso con el acceso a AWS	El socio formador se retrasa con la entrega del espacio donde se guardarán los archivos del proyecto y los datos a recolectar de los usuarios.	Estratégico
RI007	Dejar la Base de datos vulnerable para acceso de terceros no autorizados	No se implementan métodos de seguridad respectivos en la base de datos para restringir el acceso a los datos a usuarios no autorizados.	Operacional
RI008	Dejar las API vulnerables	Los datos que se transportan a través de las	Operacional

	para acceso de terceros no autorizados (Man-in-the-middle)	API son vulnerables a ataques, se pierden los datos durante el transporte o son vulnerables a lectura de estos por algún usuario no autorizado.	
RI009	Dejar el sistema vulnerable a ataques DoS	Algún usuario con intenciones maliciosas puede iniciar un ataque de DoS en el inicio de sesión para detener las operaciones del sistema y dejar vulnerable al sistema.	Operacional
RI010	Dejar el sistema vulnerable a ataques de inyección SQL	Algún usuario con intenciones maliciosas puede hacer queries SQL desde el UI del sistema, dejando vulnerable la base de datos para posibles modificaciones no autorizadas, robo y pérdida de datos.	Operacional
RI011	Pérdida o corrupción de los datos en la base de datos	Se pierden los datos de la base de datos o se corrompen al punto de ser inutilizables para el sistema.	Operacional
RI012	Suplantación de identidad para acceder	Algún usuario no autorizado utiliza la cuenta de un usuario autorizado para entrar al sistema.	Operacional
RI013	Error en integración de componentes del proyecto en Github	Se genera una incompatibilidad al momento de juntar las partes del sistema para hacerlo uno solo.	Entrega del proyecto
RI014	Errores dependiendo del navegador que se está usando	Los usuarios pueden experimentar errores de carga de datos o en el UI si entran al sistema desde algún navegador.	Operacional
RI015	Malinterpretación de los prompts para la AI	Generación de prompts no específicos a la IA para la generación de los datos necesarios.	Operacional

Definición de probabilidad e impacto de riesgo

Los riesgos serán calificados dependiendo de la probabilidad de que un riesgo ocurra y el nivel del impacto que tendrá sobre el proyecto, se usará la siguiente escala para la ponderación del riesgo en la matriz de riesgos:

Muy Bajo	1
Bajo	2
Medio	3
Alto	4
Muy Alto	5

Matriz de probabilidad e impacto de riesgos

Impacto		Muy Bajo	Bajo	Medio	Alto	Muy Alto
Probabilidad		1	2	3	4	5
Muy Bajo	1	1	2	3	4	5
Bajo	2	2	4	6	8	10
Medio	3	3	6	9	12	15
Alto	4	4	8	12	16	20
Muy Alto	5	5	10	15	20	25

Tabla de análisis de riesgos

	Riesgo	Prob	Impacto	Pond	Acción de prevención
RI001	Incumplimiento con la entrega final del proyecto	3	5	15	- Planeación de sprints y tiempo de desarrollo. - Sesiones de retroalimentación semanales/diarias.
RI002	Error en comunicación con los Stakeholders	3	5	15	- Sesiones recurrentes con los stakeholders. - Comunicación con los profesores por DM.
RI003	Alcance del proyecto muy ambicioso para el tiempo disponible	2	4	8	- Planeación de tareas previa. - Revisión de tareas y alcance con Stakeholders y profesores antes de empezar el desarrollo.
RI004	Corrupción de los archivos que conforman el proyecto	2	4	8	- Tener un historial de versiones en GitHub.
RI005	Retraso con el acceso a la API de OpenAI	2	2	4	- Comprar poquitos créditos de ChatGPT para realizar pruebas.
RI006	Retraso con el acceso a AWS	2	2	4	- Usar un espacio temporal para trabajar con las bases de datos y las API.
RI007	Dejar la Base de datos vulnerable para acceso de terceros no autorizados	3	3	9	- Autenticación de datos.
RI008	Dejar las API vulnerables para acceso de terceros no autorizados	3	3	9	- Usar autorizaciones para el acceso a las APIs. - Limitar el número de

					requests.
RI009	Dejar el sistema vulnerable a ataques DoS	2	3	6	- Uso de Auth0, el cual mitiga los ataques de DoS
RI010	Dejar el sistema vulnerable a ataques de inyección SQL	3	4	12	- Uso de Prisma, el cual ayuda a prevenir inyecciones de SQL.
RI011	Pérdida o corrupción de los datos en la base de datos	2	4	8	- Generar backups regularmente. - Implementar redundancia dentro de lo posible. - Validación de los datos. - Limitar el acceso a la BD.
RI012	Suplantación de identidad para acceder	3	3	9	- Usar autenticaciones fuertes. - Encriptar datos sensibles. - Educar a los usuarios para aplicar buenas prácticas de seguridad.
RI013	Error en integración de componentes del proyecto en Github	2	3	6	- Hacer un plan de desarrollo para el manejo de branches en Github.
RI014	Errores dependiendo del navegador que se está usando	1	2	2	- Asegurar que las funciones que vamos a utilizar sean compatibles con todos los navegadores deseados. - Hacer pruebas durante el desarrollo.
RI015	Malinterpretación de los prompts para la AI	3	5	15	- Hacer los prompts lo más específico posibles. - Hacer pruebas durante el desarrollo. - Especificar el formato de entrega de los datos (JSON)

9. Plan de calidad

9.1 Casos de prueba

Archivo de los casos de prueba y sus descripciones:  Caso de prueba .xlsx

En la siguiente prueba se presentan los casos de prueba, así como también la URL a el backlog donde se definieron.

HU	Descripción	Caso de prueba	Link a Backlog
HU10101	Ingresar a la cuenta	CP10101 CP10102 CP10103 CP10104	https://github.com/orgs/wizelineacademy/projects/6/views/1?pane=issue&itemId=22039576
HU10103	Intentar ingresar sin cuenta Wizeline	CP10106	https://github.com/orgs/wizelineacademy/projects/6/views/1?pane=issue&itemId=22039626
HU10201	Cerrar sesión desde el perfil	CP10201 CP10202	https://github.com/orgs/wizelineacademy/projects/6/views/1?pane=issue&itemId=22039683
HU10301	Entrar a la edición de info desde el perfil	CP10301 CP10302	https://github.com/orgs/wizelineacademy/projects/6/views/1?pane=issue&itemId=22039787
HU10302	Editar y guardar cada uno de los cambios	CP10303 CP10304 CP10305 CP10306	https://github.com/orgs/wizelineacademy/projects/6/views/1?pane=issue&itemId=22163317
HU10303	Editar y no guardar cada uno de los cambios	CP10307 CP10308 CP10309	https://github.com/orgs/wizelineacademy/projects/6/views/1?pane=issue&itemId=22716933
HU20101	Ingresar info de perfil manualmente	CP20101 CP20102 CP20103 CP20104	https://github.com/orgs/wizelineacademy/projects/6/views/1?pane=issue&itemId=22039805
HU20301	Ingresar info de perfil desde LinkedIn	CP20301 CP20302	https://github.com/orgs/wizelineacademy/projects/6/views/1?pane=issue&itemId=22039848
HU20401	Generar y ver Roadmap	CP20401 CP20402	https://github.com/orgs/wizelineacademy/projects/6/views/1?pane=issue&itemId=22039850

HU20501	Ver lista de colabs siendo colaborador	CP20501 CP20502 CP20503 CP20504	https://github.com/orgs/wizelineacademy/projects/6/views/1?pane=issue&itemId=22039855
HU20601	Ver los proyectos desde el perfil	CP20601 CP20602	https://github.com/orgs/wizelineacademy/projects/6/views/1?pane=issue&itemId=22040276
HU20602	Regresar al perfil después de ver los proyectos	CP20603 CP20604 CP20605	https://github.com/orgs/wizelineacademy/projects/6/views/1?pane=issue&itemId=22163617
HU30101	Ver lista de colabs siendo administrador.	CP30101 CP30102 CP30103 CP30104 CP30105	https://github.com/orgs/wizelineacademy/projects/6/views/1?pane=issue&itemId=22039909
HU30102	Ver lista de ordenes siendo administrador.	CP30106 CP30107 CP30108 CP30109 CP30110 CP30111	https://github.com/orgs/wizelineacademy/projects/6/views/1?pane=issue&itemId=22039912
HU30201	Ver a todos los colaboradores	CP30201 CP30202 CP30203 CP30204 CP30205 CP30206 CP30207 CP30208	https://github.com/orgs/wizelineacademy/projects/6/views/1?pane=issue&itemId=22039968
HU30202	Agregar colabs al sistema	CP30209 CP30210 CP30211 CP30212 CP30213	https://github.com/orgs/wizelineacademy/projects/6/views/1?pane=issue&itemId=22039970
HU30203	Editar info de colabs ya existentes	CP30214 CP30215 CP30216	https://github.com/orgs/wizelineacademy/projects/6/views/1?pane=issue&itemId=22039972
HU30204	Borrar (lógicamente) colabs del sistema	CP30217 CP30218	https://github.com/orgs/wizelineacademy/projects/6/views/1?pane=issue&itemId=22039973
HU30205	Borrar al empleado que ya no trabaja en Wizeline	CP30219	https://github.com/orgs/wizelineacademy/projects/6/views/1?pane=

			issue&itemId=23238542
HU30301	Ver todas las proyectos	CP30301 CP30302 CP30303 CP30304 CP30305 CP30306 CP30307	https://github.com/orgs/wizelineacademy/projects/6/views/1?pane=issue&itemId=22040035
HU30302	Generar proyectos	CP30308 CP30309	https://github.com/orgs/wizelineacademy/projects/6/views/1?pane=issue&itemId=22040037
HU30303	Marcar como terminada un proyecto	CP30310 CP30311	https://github.com/orgs/wizelineacademy/projects/6/views/1?pane=issue&itemId=22040040
HU40101	Ver lista de proyectos siendo cliente	CP30401 CP30402 CP30403 CP30404 CP30405 CP30406	https://github.com/orgs/wizelineacademy/projects/6/views/1?pane=issue&itemId=22040162
HU40102	Ver detalles de cada proyecto	CP30407 CP30408	https://github.com/orgs/wizelineacademy/projects/6/views/1?pane=issue&itemId=22040165
HU40103	Ver info de colabs en mi proyecto	CP30409	https://github.com/orgs/wizelineacademy/projects/6/views/1?pane=issue&itemId=22040166

9.2 Roles y responsabilidades en el plan de calidad

Los roles y responsabilidades relacionados al proceso de calidad dentro del desarrollo del proyecto se presentan en la siguiente tabla.

Roles	Responsabilidad	Colaborador
Desarrollador	Desarrollar requerimientos Back End, FE e infraestructura. Se debe cumplir con lo funcional del requerimiento y el proceso de calidad definido	Andres Fuentes Jorge De León Mario Robles Andres Fernandez Gerardo Mora Oscar Reyna
Code Review	Cada que alguien suba una modificación del desarrollo del proyecto, uno de los compañeros que impliquen una posible área afectada (calidad, datos, etc.) revisará en conjunto a la persona que realizó el cambio así como un grupo de test exitosos con el fin de contar con un parámetro de aseguramiento antes de subirlo a producción.	Andres Fuentes Jorge De León Mario Robles Andres Fernández Gerardo Mora Oscar Reyna
Auditor	En término aleatorios durante, al inicio o final del sprint se realizarán auditorías por parte de nuestro project manager o product owner, es decir, rendir cuentas de los avances de los departamentos dentro de cada sprint así como pendientes de sprints anteriores, esta actividad tendrá como objetivo principal brindar notacionaes tangibles para nuestro socio formador sobre nuestra gestión de proyecto dentro de las funcionalidades solicitadas y los tiempos acordados.	Andres Fuentes (Product Manager) Jorge De León (Product Owner)
Revisor	Además de las tareas propias de un auditor, contaremos con el apoyo de revisores, en este caso serán los instructores que durante el desarrollo nos apoyaran para ejemplificación e introducción de fundamentos del stack tecnológico a usar . Las sesiones de revisión sucederán principalmente cada que los instructores dictaminen un entregable parcial y sea necesario comenzar nuestro trabajo, y a la par solicitar revisiones o asesorías en caso de necesitarlas para que una vez que se llegue a la revisión se tenga una notación exitosa.	Ing. Alberto Ruiz García Ing. Elvia Itzamná Rosas Herrera Ing. Joel Fernando Sifuentes Ortíz Ing. Luis Carlos Aguirre Mora Ing. Omar Daniel Rodríguez Salinas
Gerente de calidad	Se encargará de que los estándares de calidad sean cumplidos por los revisores y auditores.	Mario Robles (Quality Manager)
Delivery	Ejecución de tareas ocurre hasta las revisiones con	Andrea Fernández

management	<p>instructores/instructoras, así como midterms con socio formador o presentaciones finales.</p> <p>Su mayor tarea como responsable es la revisión del producto junto con el cliente final, esto a su vez afecta al plan de calidad ya que las normativas para las rúbricas de aseguramiento de calidad no pueden ser estáticas, sino que debe ser adaptables a las necesidades del cliente, las interacciones son esenciales para poner claros todos los puntos sobre la mesa con el fin de asegurarse que aquello que se hace del lado de producción continua alineado con lo deseado por el cliente, y en caso de que no es necesario ajustar las normativas de calidad que generen cierta afectación.</p> <p>Dichas revisiones se dan cada que se ameriten dependiendo de las solicitudes respecto a las revisiones de instructores o presentaciones de revisión o avances de proyecto</p>	(delivery manager)
------------	--	--------------------

9.3 Prácticas de calidad y productos de trabajo

En el proceso de calidad se encuentran una serie de acciones, prácticas y tareas que se aplicarán a distintos productos de ingeniería de software correspondientes al proyecto, como se muestra a continuación.

Implementación de estándares de desarrollo.	<ul style="list-style-type: none">● Definir los estándares internacionales de calidad a utilizar durante la planeación del proyecto.● Utilizar los estándares internacionales de calidad durante el desarrollo del producto.● Revisar durante los code reviews que los estándares internacionales de calidad sean usados en el producto.
	Code Reviews <ul style="list-style-type: none">● El repositorio principal tendrá varias branches, las principales y las que se manejan para almacenar los proyectos completos serán main y development.● Cada nueva feature tendrá su propia branch, se seguirá la siguiente estructura: feature / nombre de feature. Cada una de las features va enlazada a una / unas tarea asignadas en Trello.<ul style="list-style-type: none">○ Las branches deberán ser creadas en base a la branch de desarrollo.○ Cada vez que se haga un Pull Request debe ser revisado por un code reviewer antes de hacer merge.○ El funcionamiento de la feature de usuario debe ser probada en la branch de desarrollo antes de dar como completada la historia y pasarse a main.○ Para pasar a main, se debe generar un Pull Request desde development<ul style="list-style-type: none">■ Cada merge a main puede incluir una o varias features
	Estándares de desarrollo <ul style="list-style-type: none">● Se utilizará la librería Prettier - Code Formatter para formalizar el código del proyecto.● Para el nombre de variables y funciones:<ul style="list-style-type: none">○ Deben tener un nombre descriptivo del propósito, no importa la longitud del nombre○ Deben estar en inglés○ Deben estar en formato camelCase● Para los Pull Requests<ul style="list-style-type: none">○ Nombre: feature + "/" + nombre del feature○ Descripción: detalle de los cambios

Definición de ambientes de trabajo

- **Ambientes de trabajo:** Desarrollo (dev) y Producción (main)
 - **Herramienta de desarrollo:** Se utiliza Visual Studio Code como editor de código para el desarrollo del software.
 - **Gestión de versiones:** A través de las branches que se llegarán a crear con su fecha ya establecida en la plataforma automáticamente.
 - **Ambientes existentes:** Existirán dos ambientes que alojarán los componentes mencionados previamente. El primero será el de Development (desarrollo) en el que se realizarán las pruebas relacionadas a las implementaciones añadidas o modificadas. El segundo será el de Main (main) en el que se subirán los cambios que funcionen correctamente en el ambiente de Development.
- **Code Reviews:** Los code reviews se harán después de subir la branch con la nueva feature
 - **Hacer merge a development:** Después de subir la branch con la nueva feature
 - **Hacer merge a main:** Cada final de sprint

Definición de pruebas

Cada caso de prueba debe de tener un criterio de aceptación de historia de usuario asociada, cada criterio de aceptación puede tener una o varios casos de prueba

Para cada una de las pruebas:

- Deben ser registradas en un excel de los casos de prueba
- Para las pruebas funcionales, deben ser basadas en alguna historia de usuario.
- Deben tener un ID único, por ejemplo, CP10203 + nombre
 - Tipo de requerimiento funcional
 - General (1)
 - De colaborador (2)
 - De administrador (3)
 - De cliente (4)
 - Número del requerimiento (02)
 - Número de la prueba (03)
 - Debe de haber 15 pruebas funcionales por desarrollador
 - Deben tener un nombre resumiendo la función a probar
- Para las pruebas End-to-end, deben ser basadas en un flujo dentro de la aplicación
- Deben tener un ID único, por ejemplo, E2E + número + nombre
 - Deben tener un número de prueba E2E único
 - Debe de haber una prueba E2E por desarrollador
 - Deben tener un nombre resumiendo la función a probar
- Deben tener una descripción específica de la función a probar

	<ul style="list-style-type: none"> • En caso de aplicar, deben tener una o varias precondiciones • Se debe registrar cada uno de los pasos que se deben hacer • Se debe registrar el estado del caso de prueba <ul style="list-style-type: none"> ○ Pendiente: no se ha comenzado. ○ No Aceptado: prueba negativa, debe ser corregida. ○ Aceptado: prueba positiva. • Deben tener un criterio de aceptación asociado con la prueba • Debe de tener el resultado esperado de la prueba • Debe de tener el resultado actual de la prueba <ul style="list-style-type: none"> ○ En caso de ser una prueba en proceso, debe incluir el resultado y la razón por la que falla ○ En caso de ser una prueba cumplida, debe describir el resultado
--	---

Ejecución de pruebas	<ul style="list-style-type: none"> • Al terminar el proyecto, se ejecutarán los casos de prueba establecidos para evitar errores a la hora de entregar el proyecto.
	Plan de Ejecución <ul style="list-style-type: none"> • Las pruebas unitarias se harán usando cypress. • Después de terminar el proyecto, los desarrolladores deberán hacer pruebas unitarias, manuales o automatizadas con cypress y utilizando CI (Github Actions), para revisar el funcionamiento de las funciones en el producto final • El desarrollador, deberá hacer 15 pruebas unitarias automatizadas. <ul style="list-style-type: none"> ○ En caso de que la prueba salga positiva, en la tabla de excel se deberá marcar el caso de prueba como Aceptado ○ En caso de que la prueba salga negativa, en la tabla de excel se deberá marcar el caso de prueba como No aceptado • Se nombrará cada prueba acorde al criterio de aceptación que se esté evaluando con esa prueba. • Se nombrará cada prueba acorde al tipo que corresponda (Integral, funcional, end to end, unitaria, etc.) • Se llevará un control del resultado de las pruebas en un archivo de excel donde se especificará el porcentaje de cumplimiento según el número de veces que se haya ejecutado.

Proceso de despliegue	<ul style="list-style-type: none"> • Los desarrolladores deberán hacer pruebas unitarias automatizadas para revisar el funcionamiento de las funciones en el producto final • Antes de desplegar la versión definitiva, se llevarán a cabo pruebas de aceptación E2E / casos de prueba para garantizar que el software funciona según lo planeado y cumple con los criterios de aceptación necesarios.
------------------------------	--

- | | |
|--|---|
| | <ul style="list-style-type: none">• Se implementará un sistema de control de versiones y ambientes adecuado para garantizar que solo se desplieguen versiones de software completamente probadas y aprobadas.• Se implementarán dos ambientes para el desarrollo y despliegue de la plataforma: desarrollo y producción, apoyándonos del sistema de versiones en GitHub para su manejo donde para el desarrollo y pruebas de las funcionalidades creadas para después pasar al ambiente de producción de manera segura,.• Se implementarán herramientas de integración continua automatizadas con el apoyo de las herramientas de GitHub Actions para llevarlo a cabo de manera confiable, consistente y segura. |
|--|---|

9.4 Métricas del proyecto

A continuación, en el siguiente enlace encontrarás algunas de las métricas del proyecto que te permitirán conocer el rango de error que puede presentarse: [📄 Métricas del proyecto](#)

9.5 Estrategia de Pruebas

Tipos de pruebas consideradas, su alcance y su aplicación:

- Prueba unitaria para cada requerimiento. La traza de cada caso de uso con los requerimientos debe ser explícita para diseñar las pruebas.
- Pruebas de integración (funcionalidad) consistentes con los criterios de aceptación en los casos de prueba funcional definidos por el cliente. Se enfocan en probar las interfaces y la comunicación entre los componentes ya probados unitariamente.
- Pruebas end-to-end. Estas pruebas son ejecutadas por el usuario, y prueban todo el flujo de punta a punta del desarrollo.

Tipo de prueba	Automatización	Periodicidad
Pruebas unitarias	Automatizada con las herramientas de infraestructura, devOps y calidad (cypress, amazon lightsail)	Al final de cada historia de usuario que se haya completado el desarrollo y que proceda a la etapa de revisión de departamento, previamente a su muestra frente a instructores o socios formadores. Lapsos de tiempo aproximados tres veces por semana o mínimo seis veces por sprint.
Pruebas de integración	Manual o Automatizada dependiendo de los requerimientos de la tarea.	Cada que una prueba unitaria sea completada y validada dentro de los parámetros deseados por el cliente, con el fin de revisar su compatibilidad con el servidor/ambiente que será integrado. Sus lapsos de ocurrencia tendría una agenda proporcional al progreso de las pruebas unitarias
Pruebas end-to-end	Manual o Automatizada dependiendo de los requerimientos de la tarea.	Una vez que se completen la validación de las pruebas de integración, probando con testers y métricas de datos similares con el fin de medir la latencia y rendimiento de las pruebas. Sus lapsos de ocurrencia tendrían una agenda proporcional al progreso de las pruebas unitarias e integrales.

9.6 Entregables de Calidad

Se realizarán pruebas al finalizar el proyecto, las estarán escritas en el github, separado por folders, distinguidos por nuestras matrículas.

9.7 Tipos de severidades

A través del proyecto existente declaramos dos tipos de severidades (Mayor y menor) una severidad es la mezcla entre la probabilidad del riesgo del proyecto y el impacto en el proyecto. Escribimos una descripción detallada ante cada una de nuestras severidades.

Severidad	Descripción
P2 - Baja	Los defectos de severidad baja son los que no tengan mucha importancia dentro del producto final, son los defectos que no afectan de manera negativa y que serán corregidos solo si hay tiempo de sobra durante el desarrollo del proyecto.
P1 - Media	Los defectos de severidad media son los que tienen importancia dentro del producto final, son defectos que afectan de alguna manera al producto y deben ser corregidas, pero no son cuestiones que necesitan parar la producción.
P0 - Alta	Los defectos de severidad alta con los que tienen mucha importancia dentro del producto final, son defectos que afectan negativamente al producto y la producción debe ser detenida para corregirse, de lo contrario el desarrollo no puede seguir en proceso.

9.8 Tipos de defectos

En la siguiente tabla se encuentran los tipos de defectos que son posibles a presentarse durante el desarrollo del producto. Los enlistamos de menor a mayor de acuerdo a su impacto en el proyecto. En el tipo 10, el menor, existen los defectos de entendimiento, los cuales fácilmente pueden ser explicados y corregidos. Por otro lado existe el tipo 50, donde sí existe una falla en el cumplimiento de los estándares de requerimientos definidos el defecto tiene un impacto mucho mayor en el proyecto.

Tipo de Defecto	Descripción
10	Defectos de entendimiento: Errores que pueden surgir en la interpretación de los requisitos o especificaciones del sistema. Con una aclaración ya se resuelve el problema.
20	Defectos de sintaxis: Errores en la manera en la que se escribe el código fuente que impiden que el programa compile o ejecute correctamente.
30	Defectos de diseño: Errores en la planificación del sistema que pueden causar problemas a lo largo del desarrollo del proyecto.
40	Defectos de funcionalidad: Errores que impiden que el sistema funcione como se espera. Pueden ser muy problemáticos y requerir una gran cantidad de tiempo y esfuerzo para corregirse.

50	Defectos de cumplimiento de estándares: Errores que ocurren cuando el sistema no cumple con los estándares de calidad y requisitos definidos por el cliente o la industria. Estos defectos pueden tener un gran impacto en el proyecto y pueden requerir una gran cantidad de trabajo para corregirse.
----	--

10. Plan de valor ganado

El Plan de Valor Ganado se desarrolló con base a los sprints del proyecto, esto con el fin de obtener un orden conforme a las etapas estipuladas para la gestión en las próximas 10 semanas del desarrollo. Con el fin de obtener el mayor balance posible se decidió dividir la cantidad de sprints en 5 períodos de dos semanas para el desarrollo de nuestro proyecto.

Se determinó que cada sprint ejecutaría determinada cantidad de tareas con base a las historias de usuario, dichas labores tendrían base y nivel de importancia o ponderación con relación a su funcionalidad y nivel de esencialidad para el funcionamiento básico de nuestro proyecto. No obstante debido a cambios dentro de nuestros casos de prueba así como posteriores solicitudes del socio formador existieron ligeramente cambios dentro del número de historias de usuario a cumplir, y así mismo algunas de las ponderaciones, logrando así generar la planeación y ponderación de un producto factible para sacar a producción dentro de los tiempo acordados a la par de las consideraciones y deseos del cliente.

Tal y como se definió inicialmente, cada una de las historias de usuario usó como valor de referencia el número de historias completadas, para lograr dicha meta, se realizó un parámetro de cuatro etapas dentro de la implementación de cada historia de usuario: Desarrollo del proceso, revisión con departamento, revisión y ajustes con el cliente, así como cierre y subida a producción.

Una vez completada cada historia de usuario dentro de la misma tabla se genera un porcentaje que representa el cálculo de valor ganado. Para lograr esto, se hizo un estudio de priorización sobre aquellas tareas cuya funcionalidad resultaba esencial para el proyecto, en nuestro caso dichas tareas eran mayormente orientadas al backend ya que la introducción y aprendizaje de Inteligencia Artificial será el pilar de nuestro desarrollo, además de tratarse de un tema nuevo dentro de nuestro stack de desarrollo, las cuáles hemos logrado completar dentro de las mayores facetas pendientes que teníamos de backend.

El progreso actual de nuestras tablas y algunos de los cambios fueron:

Sprint 1						
Desarrollo	Historias Completadas				Valor ganado	Total de progreso del sprint
Historia de Usuario	Desarrollo Dev	Revisión Dpto	Validación /cliente	Implementación en producción		Porcentaje total esperado
HU20101	Done	Done	Done	Done	20	100
HU20301	Done	Done	Done	Done	40	
HU20401	Done	Done	Done	Done	40	

Sprint 2						
Desarrollo	Historias Completadas				Valor ganado	Total de progreso del sprint
Historia de Usuario	Desarrollo Dev	Revisión Dpto	Validación /cliente	Implementación en producción		Porcentaje total esperado
HU30201	Done	Done	Done	Done	15	100
HU30202	Done	Done	Done	Done	45	
HU30203	Done	Done	Done	Done	40	

En el caso de sprint 2 se removió la historia de usuario HU30204 ya que algunas cuestiones para inhabilitar usuarios fueron removidas por cuestiones de seguridad así como posibles políticas dentro de recursos humanos para el socio formador.

HU30204	Como administrador quiero inhabilitar una cuenta para un colaborador que ya no esté en Wizeline.	<p>CA30204.1 Tener un botón dentro de la página en donde se podrá habilitar o inhabilitar a una cuenta.</p> <p>CA30204.2 El usuario inhabilitado no podrá acceder a su cuenta.</p>
---------	--	--

Sprint 3						
Desarrollo	Historias Completadas				Valor ganado	Total de progreso del sprint
Historia de Usuario	Desarrollo Dev	Revisión Dpto	Validación /cliente	Implementación en producción		Porcentaje total esperado
HU30301	Done	Done	Done	Done	15	95
HU30302	Done	Done	Done	Done	17,5	
HU30303	Done	Done	Done	Done	17,5	
HU40101	Done	Done	Done	Done	10	
HU40102	Done	Done	Done	Done	10	
HU40103	Done	Done	Done	Done	15	
HU20601	Done	Done	Done	Done	10	
HU20602	Not started	Not started	Not started	Not started	0	100

Sprint 4						
Desarrollo	Historias Completadas				Valor ganado	Total de progreso del sprint
Historia de Usuario	Desarrollo Dev	Revisión Dpto	Validación /cliente	Implementación en producción		Porcentaje total esperado
HU10101	Done	Done	Done	Not started	6	66
HU10103	Done	Done	Done	Done	4	
HU10201	Done	Done	Done	Done	12	
HU10301	Done	Not started	Not started	Not started	4	
HU10302	Done	Not started	Not started	Not started	5	
HU10303	Done	Not started	Not started	Not started	3	
HU20501	Done	Done	Done	Done	16	
HU30101	Done	Done	Done	Done	8	
HU30102	Done	Done	Done	Done	8	100

En el caso de sprint 4 algunas historias de usuario fueron resumidas dentro de una misma como el caso de las siguientes historias de usuario para inicio de sesión, las cual omitió la sección dentro de la aplicación de mantener una sesión abierta, dejando la especificación de ingreso correcto dentro de una única historia de usuario (HU10101), así mismo ocurrió en aquellas historias de usuario relacionadas a cerrar sesión, las cuáles se resumieron dentro de una sola (HU10201):

RF101 - Iniciar sesión			RF102 - Cerrar sesión		
HU	Descripción	Criterios de aceptación	HU	Descripción	Criterios de aceptación
HU10101	Como empleado de Wizeline, quiero ingresar a la plataforma utilizando mi usuario y contraseña de manera segura, y tener la opción de dejar mi sesión abierta para facilitar mi uso diario de la cuenta.	<p>CA10101.1 Como usuario registrado en la base de datos de Wizeline, podré iniciar sesión en la plataforma utilizando mi correo electrónico y la contraseña asociada a mi cuenta previamente registrada.</p> <p>CA10101.2 Se implementará un botón de "mantener sesión abierta" que al ser seleccionado permitirá a los usuarios mantener su sesión activa en la plataforma, incluso después de cerrar el navegador o salir de la plataforma temporalmente.</p> <p>CA10101.3 El sistema deberá mantener mi sesión abierta de forma segura y sin comprometer mi privacidad, evitando posibles accesos no autorizados a mi cuenta.</p>	HU10201	Como usuario quiero cerrar la plataforma, por lo que cierro sesión y me lleva a la página para iniciar sesión para asegurarse de que mi cuenta está cerrada y nadie más pueda acceder a ella, así como para poder volver a acceder a la plataforma en el futuro de manera segura.	<p>CA10201.1 Cuando el usuario selecciona el botón "logout" dentro de su perfil, la sesión se cierra y el usuario es redirigido a la página de inicio de sesión.</p> <p>CA10201.2 La sesión se cerrará sin importar si el usuario seleccionó o no la opción "dejar sesión abierta" al iniciar sesión, garantizando que la sesión esté completamente cerrada.</p>
HU10102	Como empleado de Wizeline, quiero ingresar a la plataforma utilizando mi correo electrónico y contraseña de manera segura, y tener la opción de no dejar mi	CA10102.1 Cuando el botón de "mantener sesión abierta" no sea seleccionado. Al cerrar sesión, la cuenta no deberá de ser guardada en el sistema.	HU10202	Como usuario quiero cerrar la plataforma y luego volver a abrirla después de 2 minutos para que me lleve a la página de inicio de sesión para asegurarse de que la sesión se cerró correctamente y nadie más pueda acceder a ella.	CA10202.1 Si el usuario cierra la pestaña de la plataforma y la vuelve a abrir en un lapso de 2 minutos, la plataforma debe llevarlo a la página de inicio de sesión.

Así mismo dentro de algunas historias de usuario tuvo que haber pequeños retrocesos debido a solicitudes de mejora posteriores a nuestra implementación. Así como nuevos avances dentro de otras historias de usuario.

Sprint 5						
Desarrollo	Historias Completadas				Valor ganado	Total de progreso del sprint
Historia de Usuario	Desarrollo Dev	Revisión Dpto	Validación /cliente	Implementación en producción		Porcentaje total esperado
Pruebas, evaluaciones y presentaciones finales	Not started	Not started	Not started	Not started	0	

De igual forma dentro del sprint 5 se removió una de las historias de usuario relacionadas a la carga del CV dentro de nuestra plataforma ya que tras posteriores evaluaciones con el cliente, fue sugerido por parte del mismo no implementarlo ya que cuentan actualmente con una librería que tiene dicha funcionalidad.

Dentro de esta etapa el PVG jugó un gran rol para la aportación de valor aquellas fortalezas del proyecto por medio de su impacto y reto intelectual como un peso a considerar, así como aquellas tareas que representaban pequeños lapsos de tiempo y esfuerzo en completarse.

En términos de cambios hasta este punto de inicio del quinto sprint, consideramos que obtuvimos como resultado un calendario muy alineado a nuestros objetivos del proyecto, no obstante existieron tareas que fueron teniendo ligeros retrasos dentro de sprint anteriores que no generan una afectación importante a los requerimientos mínimos del negocio, ya que es posible continuar con los estándares de calidad, seguridad, diseño ,y generación de resultados estipulados al inicio del proyecto ya que se cuenta con herramientas y fechas estipuladas dentro de nuestra organización como equipo de trabajo para retomar nuestro espacio dentro del proyecto .

Una vez obteniendo el total de nuestras actividades hasta mayo 27 del 2023, se ejecutó una tabla final en la cuál se hace un recuento considerando cada sprint y su resumen de progreso, así como el porcentaje de cada progreso respecto al colocado dentro del plan de valor ganado individual por cada sprint. Adicionalmente se colocó una ponderación (PVG) respectiva de cada sprint dependiendo de su relevancia total para el proyecto para generar un plan de valor ganado para la totalidad del proyecto, que a su vez en una sumatoria equivale al porcentaje total del proyecto, el cuál una vez completadas todas las tareas generará un porcentaje del cien por ciento.

Tomando en cuenta nuestro modelo predictivo podemos visualizar que contamos con un buen número de progreso dentro de nuestras actividades para finalizar, no obstante, dentro de nuestra organización estamos alineados a que habrá un mayor flujo de trabajo y horas dedicadas para este último sprint con el fin de que podamos completar las tareas de nuestro proyecto de acuerdo al valor ganado deseado así como el porcentaje ideal de finalización del proyecto:

Sprint	Descripción de actividades	Progreso del Sprint (%)	Valor Ganado (%)	Porcentaje total del proyecto	Porcentaje total esperado
Sprint 1 (27 mar - 14 abr)	Generación de la información del usuario con la herramienta IA y se muestra en el UI y DB.	100	30	78,9	100
Sprint 2 (17 abr - 28 abr)	Administración de los usuarios, se crean, modifican, eliminan de la BD y se muestra la lista en el UI.	100	20		
Sprint 3 (1 may - 12 may)	Administración de las órdenes, se crean, modifican, eliminan de la BD y se muestra la lista en el UI. Los clientes y colaboradores pueden ver la lista de órdenes a las que fueron asignados.	95	19		
Sprint 4 (15 may - 26 may)	Inicio y cierre de sesión, se edita la información de los perfiles de usuario, creación de dashboard general para colaboradores y administradores. Refinamiento del UI y UX.	66	9,9		
Sprint 5 (29 may - 9 jun)	Extracción de datos de un CV en PDF. Pruebas de aceptación y documentación final	0	0		

Sprint	Porcentaje
1	30%
2	20%
3	20%
4	15%
5	15%

Dentro del siguiente enlace podrán encontrar e interactuar con las hojas para la visualización del plan de valor ganado (PVG): [Plan de Valor Ganado PVG](#)