



Tecnológico de Monterrey

Planeación de sistemas de software (Gpo 103)

Documentación:

Administración

Alan De Loa Larios

Gilberto Ángel Camacho Lara

Eugenio Elizondo Jaime

Gianni André Freire Vásquez

Aldo Leonardo López Ruiz

Carlos Fernando Ramos Mena

Misión y Valores

Misión

Desarrollar una plataforma basada en inteligencia artificial que optimice la gestión del talento en la empresa, facilitando la asignación eficiente de personal, la automatización de resúmenes de currículums y la generación de recomendaciones inteligentes para managers y empleados. Nuestro objetivo es reducir el tiempo de asignación en un 30% y automatizar el 80% de los resúmenes de CVs, mejorando la precisión en la asignación de personal y la satisfacción de los usuarios.

Valores

1. **Innovación** – Aplicamos inteligencia artificial para transformar la gestión del talento y mejorar la toma de decisiones en la empresa.
2. **Eficiencia** – Reducimos el tiempo de asignación del personal y automatizamos procesos clave para optimizar recursos.
3. **Transparencia** – Brindamos a gerentes, empleados y especialistas en talento una visión clara del estado y evolución del talento en la organización.
4. **Seguridad** – Garantizamos la privacidad y protección de la información sensible de empleados y la empresa mediante medidas de cifrado y control de acceso.
5. **Crecimiento Profesional** – Facilitamos el desarrollo de habilidades y la planificación de carrera de los empleados mediante recomendaciones personalizadas y análisis de trayectorias.
6. **Colaboración** – Fomentamos la comunicación entre empleados, managers y especialistas en talento para asegurar una gestión eficiente y alineada con los objetivos organizacionales.

Equipo y Roles

Integrantes	Rol	Responsabilidades
Alan De Loa Larios	Ingeniero en base de datos y arquitecto de software	Definir la estructura interna de los datos, así como diseñar una arquitectura adecuada para el desarrollo del proyecto y la comunicación entre sus componentes
Gilberto Angel Camacho	Desarrollador Front End	Diseño de componentes de UI con enfoque en UX.
Eugenio Elizondo Jaime	Scrum Master y Product Owner	Dirigir y llevar el control del proyecto con metodología SCRUM, además de funcionar como un vínculo entre el cliente y los desarrolladores
Gianni Andre Freire	QA & Engineer & tester	Asegurar la calidad del software mediante el desarrollo de pruebas unitarias y el testing de la app
Aldo Leonardo Lopez Ruiz	Desarrollador Front End	Diseño de componentes de UI con enfoque en UX
Carlos Fernando Ramos Mena	Desarrollador Back end	Desarrollo de endpoints para la obtención, modificación, creación y eliminación de datos. Manejo de datos seguro

Roadmap

Road Map Bloque 2

Sprint	24 Marzo - 28 Marzo	31 Marzo - 4 Abril	7 Abril - 11 Abril	21 Abril- 25 Abril	28 Abril - 2 Mayo
Sprint 1: Configuración Inicial y Preparación	<ul style="list-style-type: none">Configuración del entorno de trabajoAjuste y Revisión del diseño				
Sprint 2: Autenticación y Usuarios		<ul style="list-style-type: none">Gestión de perfilesImplementación de Roles BásicosDesarrollo del sistema de login			
Sprint 3: Perfil y Edición de Usuario			<ul style="list-style-type: none">Conexión con backend para edición de datosValidaciones de formularios		
Sprint 4: Módulo de Recomendaciones (IA)				<ul style="list-style-type: none">Desarrollo del algoritmo basado de IAIntegración con perfil del usuario	
Sprint 5: Generador Automático de CV					<ul style="list-style-type: none">Creación del módulo para generar curriculumsExportación y pruebas del CV

Road Map Bloque 3

Sprint	12 Mayo - 16 Mayo	19 Mayo - 23 Mayo	26 Mayo - 30 Mayo	2 Junio - 6 Junio	9 Junio - 13 Junio
Sprint 6: Asignación Inteligente de Personal	<ul style="list-style-type: none">Desarrollo del comparador de habilidadesSugerencias de asignación a proyectos				
Sprint 7: Seguridad y Privacidad		<ul style="list-style-type: none">Definición de permisos de rol			
Sprint 8: Integración de Módulos y Pruebas			<ul style="list-style-type: none">Pruebas de integraciónUnión completa de backend y frontend		
Sprint 9: Optimización y Ajustes Finales				<ul style="list-style-type: none">Mejoras de rendimientoAjustes	
Sprint 10: Despliegue y Cierre del Proyecto					<ul style="list-style-type: none">Despliegue finalDocumentación CompletaReunión de cierre

Distribución de Roles por Sprints

Sprint 1 – Configuración y Preparación

- Scrum Master:** Organización inicial y revisión del diseño en Figma.
- Backend Developer:** Configuración del entorno y base de datos.

- **Frontend Developers:** Validación del diseño para desarrollo.

Sprint 2 – Autenticación y Usuarios

- **Backend:** APIs de login y gestión de perfiles.
- **Frontend:** Pantallas de registro e inicio de sesión.
- **QA:** Pruebas básicas de autenticación.

Sprint 3 – Perfil y Edición de Usuario

- **Backend:** Actualización de perfiles.
- **Frontend:** Formularios y validaciones.
- **QA:** Pruebas de edición.

Sprint 4 – Recomendaciones IA

- **Arquitecto:** Diseño del modelo de IA.
- **Backend:** Desarrollo e integración.
- **QA:** Validación del sistema de recomendaciones.

Sprint 5 – Generador Automático de CV

- **Backend:** Procesamiento de datos para CV.
- **Frontend:** Interfaz de visualización del CV.
- **QA:** Pruebas de generación y exportación

Sprint 6 – Asignación Inteligente de Personal

- **Arquitecto:** Lógica de comparación de habilidades.
- **Backend:** Desarrollo del algoritmo de asignación.
- **QA:** Validación de sugerencias.

Sprint 7 – Seguridad y Privacidad

- **Backend:** Roles avanzados y protección de datos.
- **QA:** Auditoría de seguridad.

Sprint 8 – Integración y Pruebas

- **Backend & Frontend:** Integración total del sistema.
- **QA:** Pruebas integrales de módulos.

Sprint 9 – Optimización Final

- **Backend & Frontend:** Mejoras de rendimiento y ajustes finales.
- **QA:** Validación antes del despliegue.

Sprint 10 – Despliegue y Cierre

- **Scrum Master:** Coordinación final y entrega.
- **Backend & Frontend:** Últimos detalles técnicos.
- **QA:** Revisión final.
- **Todo el equipo:** Capacitación al cliente y cierre del proyecto.

Work Breakdown Structure (WBS)

El siguiente diagrama de descomposición del trabajo (WBS) representa la estructura del proyecto "Desarrollo de PathFinder", segmentando sus fases en distintos niveles para facilitar la organización y gestión del trabajo.

Niveles de Descomposición del Trabajo

1. **Nivel 1:** Proyecto principal – Desarrollo de PathFinder.
2. **Nivel 2:** Principales fases del proyecto, incluyendo:
 - Planificación y extracción de requerimientos.
 - Diseño del sistema.
 - Desarrollo del sistema.
 - Pruebas y validación.
 - Implementación y despliegue.
3. **Nivel 3:** Tareas específicas dentro de cada fase, como:
 - Análisis de requerimientos y reunión con el cliente.

- Diseño de arquitectura, base de datos e inteligencia artificial.
- Implementación del backend y frontend, conexión con base de datos.
- Creación y ejecución de pruebas, integración de IA.
- Despliegue del producto final.

Esta estructura facilita la gestión del alcance del proyecto, asegurando que cada tarea está claramente definida y asignada a la fase correspondiente.



Estrategia de Comunicación

Para garantizar una comunicación efectiva entre todos los involucrados en el proyecto, se utilizarán diferentes herramientas y canales de comunicación según el tipo de interacción necesaria.

Medios de Comunicación

- **Microsoft Teams:** Se utilizará para reuniones formales con el cliente y sesiones de seguimiento semanales.
- **WhatsApp:** Será el canal de comunicación rápida para resolver dudas operativas y coordinar tareas internas.
- **Correo Electrónico:** Para el envío de reportes, documentos y actualizaciones clave del proyecto.
- **Jira/Trello:** Servirá para el seguimiento de tareas, asignación de responsabilidades y estado de avance en cada sprint.

Periodicidad de la Comunicación con Stakeholders

Stakeholder	Frecuencia de Comunicación	Medio de Comunicación	Propósito
Gerentes	Reunión semanal	Microsoft teams	Revisión del progreso del proyecto, ajustes estratégicos y validación de requerimientos
Empleados	Comunicación asíncrona según necesidades	Microsoft teams	Resolución de dudas sobre el sistema y retroalimentación sobre usabilidad
Directores y Altos Ejecutivos	Revisión cada 2 semanas	Correo electrónico, Microsoft Teams	Presentación de avances. KPIs y toma de decisiones estratégicas
Talent Fulfillment Specialists (TFS)	Comunicación continua cada sprint	Jira, Microsoft teams	Coordinación de asignación de personal, validación de datos y retroalimentación

Plan de Riesgos

El siguiente plan identifica y evalúa los principales riesgos del proyecto, estableciendo estrategias para su mitigación.

1. Identificación de Riesgos y Evaluación

Riesgo	Impacto	Probabilidad	Mitigación
Cambios en los requerimientos del cliente	Alto	Alta	Mantener comunicación constante con el cliente y realizar revisiones frecuentes. Aplicar metodología ágil para adaptarse rápidamente.
Problemas en la integración del backend con la IA	Alto	Media	Implementar pruebas unitarias y de integración. Realizar pruebas piloto antes de la implementación completa.
Fallos en la seguridad de datos	Alto	Media	Implementar encriptación de datos y autenticación robusta. Realizar auditorías de seguridad periódicas.
Baja adopción del sistema por parte de los usuarios	Medio	Media	Realizar sesiones de capacitación y brindar documentación clara. Recoger feedback de usuarios clave para optimización.
Retrasos en el desarrollo por carga de trabajo	Alto	Alta	Definir prioridades en Trello y asignar tareas de manera eficiente. Ajustar el alcance si es

			necesario.
Fallos en la comunicación interna	Media	Media	Establecer reuniones /semanales y mantener canales abiertos en Teams
Dependencia de herramientas externas (APIs, modelos IA)	Alto	Media	Evaluar alternativas viables y contar con planes de contingencia en caso de fallos en terceros.

Plan de Mitigación de Riesgos

Para minimizar los riesgos identificados, el equipo implementará las siguientes estrategias de mitigación:

- **Revisiones constantes con el cliente:** Se realizarán reuniones semanales para identificar cambios tempranos en los requerimientos y ajustar la planificación de manera ágil.
- **Pruebas continuas de integración:** Se aplicarán pruebas unitarias y de integración desde etapas tempranas del desarrollo para evitar fallos críticos en la IA y el backend.
- **Auditorías de seguridad periódicas:** Se implementarán mecanismos de cifrado y control de acceso, con pruebas regulares para detectar vulnerabilidades.
- **Capacitación a usuarios finales:** Se impartirán sesiones de formación y se habilitará documentación accesible para garantizar la adopción del sistema.
- **Gestión eficiente del backlog:** Se definirán prioridades claras en Jira/Trello y se redistribuirán tareas si hay sobrecarga de trabajo.
- **Uso de múltiples canales de comunicación:** Se establecerán reuniones semanales en Microsoft Teams.
- **Planes de contingencia para herramientas externas:** Se identificarán alternativas en caso de fallos en servicios de terceros (APIs o modelos de IA) para evitar bloqueos en el desarrollo.

Descripción de las Ceremonias

- **Daily Stand-up**

Se realiza cada mañana con una duración de 15 minutos. Cada integrante responde tres preguntas clave:

- ¿Qué hice ayer?
- ¿Qué haré hoy?
- ¿Tengo algún impedimento?

Esto permite detectar bloqueos y coordinar mejor el trabajo diario.

- **Sprint Planning**

- Se lleva a cabo el martes al inicio de cada sprint.
- Se definen las tareas que se trabajarán en la semana y se priorizan.
- Se asignan responsables y se verifica que las historias de usuario estén listas.

- **Sprint Review**

- Se realiza el martes al finalizar el sprint.
- Se presentan los avances al cliente y a los stakeholders.
- Se recopila feedback para ajustes en las siguientes iteraciones.

- **Sprint Retrospective**

- Se realiza después del Sprint Review.
- El equipo analiza qué salió bien, qué falló y cómo mejorar.
- Se documentan aprendizajes para optimizar el trabajo en próximos sprints.

- **Sprint Demo**

- Se muestra la funcionalidad desarrollada durante el sprint.
- Los desarrolladores y testers presentan las nuevas características listas para producción.
- El Product Owner valida los entregables.

Métricas SCRUM

Para evaluar el rendimiento del equipo, optimizar la planificación de tareas y garantizar el cumplimiento de los objetivos del proyecto, se implementarán las siguientes métricas en cada sprint.

Velocidad del Equipo

Definición:

La velocidad del equipo se mide en función del número total de puntos de historia completados en un sprint. Esta métrica permite evaluar la productividad y capacidad del equipo de desarrollo a lo largo del tiempo, brindando una referencia para la planificación futura.

Objetivo:

- Determinar la cantidad de trabajo que el equipo puede manejar en cada sprint.
- Establecer un ritmo de trabajo sostenible para los desarrolladores.
- Facilitar la estimación del esfuerzo requerido para los sprints siguientes.
- Identificar tendencias en la eficiencia del equipo y posibles bloqueos en el proceso de desarrollo.

Cómo se mide:

- Se asignan puntos de historia a cada tarea dentro del backlog.
- Al finalizar el sprint, se suman los puntos de historia de todas las tareas completadas que cumplen con la **Definición de Hecho (Definition of Done - DoD)**.
- Se compara la velocidad con la de sprints anteriores para identificar variaciones en la productividad.
- Se analiza si hay discrepancias significativas que puedan indicar problemas en la estimación de tareas o en la ejecución de los desarrollos.

El seguimiento de la velocidad se hará mediante herramientas de gestión de proyectos como **Jira o Trello**, donde se registrarán las historias completadas en cada sprint.

Trabajo Pendiente

Definición:

La métrica de trabajo pendiente permite evaluar la cantidad de tareas abiertas frente a las cerradas en cada sprint. Su análisis facilita la identificación de riesgos asociados al incumplimiento de plazos y la acumulación de trabajo no finalizado.

Objetivo:

- Evaluar si el equipo está cumpliendo con el volumen de trabajo planificado.
- Detectar tareas que han quedado rezagadas o que podrían comprometer la entrega de funcionalidades clave.
- Mejorar la asignación de recursos y la distribución de trabajo dentro del equipo.
- Anticipar posibles cuellos de botella en el desarrollo de software.

Cómo se mide:

- Se contabilizan las tareas pendientes al inicio y final del sprint.
- Se registra la cantidad de tareas en cada estado: **"To Do"**, **"In Progress"**, **"Review/Testing"** y **"Done"**.
- Se genera un **Burnup Chart** o **Burndown Chart**, que reflejan gráficamente la evolución del sprint en función del trabajo completado.
- Se comparan los resultados con los sprints anteriores para detectar patrones en el desempeño del equipo.

Esta métrica se analizará en la **Sprint Review**, donde el equipo revisará si la cantidad de trabajo pendiente es aceptable o si es necesario ajustar la planificación en futuros sprints.

Uso de Tableros para Seguimiento

Para visualizar y gestionar estas métricas, se utilizarán **tableros Kanban en Jira o Trello**, los cuales estarán estructurados con los siguientes estados:

- **Backlog:** Contiene todas las tareas aún no asignadas a un sprint.
- **To Do:** Tareas planificadas para el sprint actual pero aún no iniciadas.
- **In Progress:** Tareas que están siendo trabajadas activamente por el equipo de desarrollo.

- **Review/Testing:** Tareas que han sido implementadas y están en revisión o en proceso de pruebas.
- **Done:** Tareas completadas que cumplen con la definición de hecho (DoD).

Cada semana, durante la **Sprint Review**, se analizarán los datos obtenidos a partir de estas métricas para evaluar el desempeño del equipo y optimizar la estrategia de desarrollo. Si se detectan inconsistencias o bloqueos en el flujo de trabajo, se realizarán ajustes en la planificación para garantizar la entrega de valor en cada sprint.