**Documento de requisitos funcionales y no funcionales**

**Juana Jaramillo Montoya**

**Mariana Urrego Montoya**

**Alfonso Jiménez Rivas**

**Sistemas empresariales – UPB 2025-01**

**Requisitos funcionales**

* La plataforma debe permitir al cliente crear nuevas vacantes, que contengan toda la información de la vacante, como experiencia necesaria, títulos necesarios, palabras claves, descripción del perfil, etc.
* La plataforma debe soportar que el cliente ingrese aplicantes a estas ofertas de trabajo
* La plataforma debe permitir al usuario cambiar el estado de los aplicantes durante el proceso de selección
* La plataforma debe permitir al usuario filtrar las hojas de vida de los candidatos, y seleccionar solo las que cumplen con todos los requisitos mínimos
* La plataforma debe permitir al usuario enviar notificaciones al correo del aplicante.

**Requisitos no funcionales**

* El modelo de Inteligencia artificial no se debe demorar más de 10 segundos por hoja de vida

**Historias de usuario**

**Como reclutador del equipo de recursos humanos,**

**Quiero crear nuevas vacantes** que publiquen trabajos,

**Para que pueda** definir los requisitos específicos, técnicos y en habilidades blandas como experiencia necesaria, títulos requeridos, palabras clave y descripción del perfil del candidato.

Ejemplo:

* Vacante: "MLops developer."
* Experiencia necesaria: "1-2 Años como MLops developer."
* Títulos requeridos: "Ingeniero de sistemas e informática / Ingeniero en ciencia de datos"
* Habilidades blandas: "Trabajo en equipo, habilidades de comunicación efectiva."
* Palabras clave: "Devops, Machine Learning, Conteinerización, CI/CD, Monitoreo de modelos de Inteligencia Artificial."
* Descripción del perfil: "El candidato ideal tendrá experiencia en despliegue de modelos de Inteligencia Artificial y deberá trabajar en equipo con los ingenieros de IA y otros desarrolladores."

**Como reclutador del equipo de recursos humanos,**

**Quiero ingresar fácilmente aplicantes a una vacante**,

**Para que pueda** añadir a los candidatos que se han postulado para una oferta de trabajo de forma sencilla, es decir, poder subir los postulados en grupo.

*Ejemplo:*

El reclutador llena una carpeta con PDFs obtenidos de la oferta de trabajo de LinkedIn, los selecciona todos y los arrastra a la pantalla de cargado del ATS para subirlos al sistema.

**Como gerente de contratación,**

**Quiero ver a los postulados aceptados por el equipo de recursos humanos,**

**Para analizar sus hojas de vida** y decidir si planear una primera entrevista.

**Como reclutador del equipo de recursos humanos,**

**Quiero hacer un filtrado automático de las hojas de vida de los aplicantes,**

**para ver solo aquellos que cumplen con los requisitos que especifiqué** en la vacante laboral.

*Ejemplo:*

* Mostrar sólo los candidatos que aplican a la vacante ya creada de “MLops developer”.
* El sistema debería mostrar solo los candidatos que cumplan con los requisitos que el reclutador creó para la vacante de MLops developer.

**Como reclutador del equipo de recursos humanos,**

**Quiero aceptar o rechazar aplicantes** durante el proceso de selección,

**Para un filtrado final de sus hojas de vida** antes de pasarlas a gerencia de contratación.

**Como reclutador del equipo de recursos humanos,**

**Quiero mandar correos electrónicos** durante el proceso de selección,

**Para mantener a los aplicantes informados de su estado** en el proceso.

**Flujo del proceso de reclutamiento:**

1. Al subir las hojas de vida, el sistema la va a categorizar automáticamente, para determinar cuáles cumplen con los requisitos, cuales no cumplen con los requisitos, y cuales deben ir a un proceso de revisión más detallado.
2. Luego de que una hoja de vida es aceptada, pasa a un proceso de pruebas y entrevista.
3. Luego se pasa a realizar un listado de los mejores candidatos.
4. Por último, se escoge a uno de los finalistas y se cierra la oferta.

Estados que deberíamos manejar:

* Aceptada, rechazada, por revisar, fase de pruebas, fase de entrevista, finalista

Arquitectura:

* **Framework de vue.js:** por simplicidad en el desarrollo, por tener componentes gráficos modernos, que ayudan en el desarrollo del frontend
* **Backend:** C#, por el conocimiento que tenemos de la herramienta y por su versatilidad en el desarrollo backend.
* **Base de datos:** Postgres ya que es una base de datos ligera. Se integrará con un almacenamiento en nube por objetos en azure.