

Evidencia 6 PROYECTO FORMATIVO

Diseño de Especificaciones Técnicas para su proyecto formativo

1. Introducción

1.1 Descripción general del proyecto:

El presente proyecto tiene como objetivo el desarrollo de una plataforma web denominada RHinnovate, diseñada específicamente para gestionar de manera eficiente y centralizada el proceso de contratación de personal para la empresa Miguel Caballero S.A.S.

La plataforma permitirá a los distintos actores del proceso —administradores del sistema, supervisores y candidatos— interactuar de acuerdo con sus respectivos roles. A través de una interfaz intuitiva, los supervisores podrán crear y publicar ofertas laborales, así como gestionar las postulaciones recibidas. Por su parte, los candidatos tendrán la posibilidad de registrarse, completar su perfil, subir su hoja de vida y postularse a las vacantes disponibles. Finalmente, el administrador del sistema tendrá acceso total para gestionar usuarios, supervisar la actividad del sistema y garantizar su correcto funcionamiento.

RHinnovate será desarrollada utilizando tecnologías modernas como React para el frontend y Node.js con PostgreSQL para el backend, y se implementará bajo la metodología ágil SCRUM, permitiendo una construcción progresiva, validación constante con el cliente y rápida adaptación a cambios en los requerimientos.

1.2 Alcance del sistema:

Usuarios principales del sistema:

- **Administrador del sistema:** Responsable de la gestión global del sistema, incluyendo la administración de cuentas de supervisores y control de actividad general.
- **Supervisor:** Encargado de crear, editar y gestionar las ofertas laborales, así como de revisar y calificar las postulaciones recibidas.
- **Candidato (usuario externo):** Persona interesada en postularse a las vacantes de la empresa. Puede registrarse, cargar su hoja de vida y aplicar a las ofertas laborales disponibles.

Funcionalidades clave:

- Registro y autenticación de usuarios según su rol.
- Gestión de perfil para candidatos, incluyendo carga de hoja de vida.
- Creación, edición y publicación de ofertas laborales por parte de los supervisores.
- Visualización de ofertas disponibles por parte de los candidatos.
- Postulación de candidatos a las ofertas laborales.
- Gestión de postulaciones: revisión, comentarios y actualización de estado por parte del supervisor.

- Envío de notificaciones automáticas a los usuarios sobre cambios relevantes (estado de postulación, nueva oferta, etc.).
- Panel administrativo para control de usuarios y monitoreo general del sistema.

Limitaciones o restricciones:

- Solo se permitirá la postulación de usuarios con rol Candidato.
- Un candidato no podrá postularse más de una vez a la misma oferta.
- El sistema será accesible únicamente en idioma español en su primera versión.
- Los documentos cargados (hoja de vida) deberán tener formato .pdf, .doc o .docx y un tamaño máximo de 5 MB.
- El acceso al sistema requerirá conexión a internet y un navegador actualizado.

2. REQUERIMIENTOS TECNICOS

2.1 Hardware Requerido

A continuación se detalla el hardware mínimo recomendado para el correcto funcionamiento y acceso al sistema RHinnovate por parte de los diferentes actores del sistema.

1. Servidor de Producción (en la nube o físico)

Opción sugerida: Servidor en la nube (ej. AWS EC2, DigitalOcean, etc.)

- CPU: 4 vCPUs
- RAM: 8 GB
- Almacenamiento SSD: 100 GB
- Sistema Operativo: Ubuntu Server 22.04 LTS
- Costo estimado mensual (DigitalOcean): **\$150.000 COP – \$200.000 COP**

2. Estaciones de Trabajo (Supervisores y Administrador)

Uso en oficina o remoto, mediante navegador web

- CPU: Intel Core i5 o equivalente
- RAM: 8 GB
- Almacenamiento: 256 GB SSD
- Pantalla: 13" o superior
- Navegador: Chrome, Firefox o Edge actualizado
- Costo aproximado por equipo: **\$2.000.000 – \$2.800.000 COP**

3. Dispositivos Móviles (opcional, para candidatos o revisión remota)

- *Acceso vía navegador responsive (no se requiere aplicación)*
- Sistema operativo: Android 10+ o iOS 13+
- RAM: 3 GB o superior
- Navegador actualizado
- Costo aproximado por dispositivo: **\$700.000 – \$1.200.000 COP**

4. Servidor de Desarrollo / Pruebas (local o nube)

- CPU: 2 vCPUs
- RAM: 4 GB
- Almacenamiento: 50 GB SSD
- Sistema Operativo: Ubuntu Server / Windows Server
- Costo estimado (nube): **\$80.000 – \$120.000 COP / mes**

2.2 Software Requerido

→ Lenguaje de programación

- Node.js **v20.x**

- **Licencia: MIT License**
- **Usado para el desarrollo del backend (API REST).**
- **JavaScript / TypeScript ES6+**
 - **Licencia: MIT License**
 - **Lenguaje principal del frontend.**
- **Frameworks y bibliotecas**
- **React V18.X**
 - **Licencia: MIT License**
 - **Biblioteca de JavaScript para la construcción de interfaces de usuario (frontend).**
- **Express.js v4.x**
 - **Licencia: MIT License**
 - **Framework minimalista para Node.js utilizado en el backend.**
- **PostgreSQL v15.x**
 - **Licencia: PostgreSQL License (similar a MIT)**

- **Sistema de gestión de base de datos relacional.**
 - **Sequelize v6.x**
 - **Licencia: MIT License**
 - **ORM para Node.js que facilita la interacción con PostgreSQL.**
 - **JWT (jsonwebtoken)**
 - **Licencia: MIT License**
 - **Librería para autenticación basada en tokens.**
 - **bcrypt.js**
 - **Licencia: MIT License**
 - **Librería para el cifrado de contraseñas.**
- **Sistema Operativo**
- **Ubuntu Server 22.04 LTS (Linux)**
 - **Licencia: GNU General Public License (GPL)**
 - **Sistema operativo recomendado para los servidores de desarrollo y producción.**

- **Windows 10/11 Pro (para estaciones de trabajo)**

- **Licencia: Propietaria (Microsoft)**
- **Sistema operativo opcional en equipos de supervisores o administradores.**

→ **Herramientas de desarrollo y soporte**

- **Visual Studio Code**

- **Licencia: MIT License**
- **Editor de código multiplataforma.**

- **Git**

- **Licencia: GPLv2**
- **Control de versiones.**

- **Postman**

- **Licencia: Gratuito (plan básico)**
- **Herramienta para pruebas de APIs.**

3. Arquitectura del Software

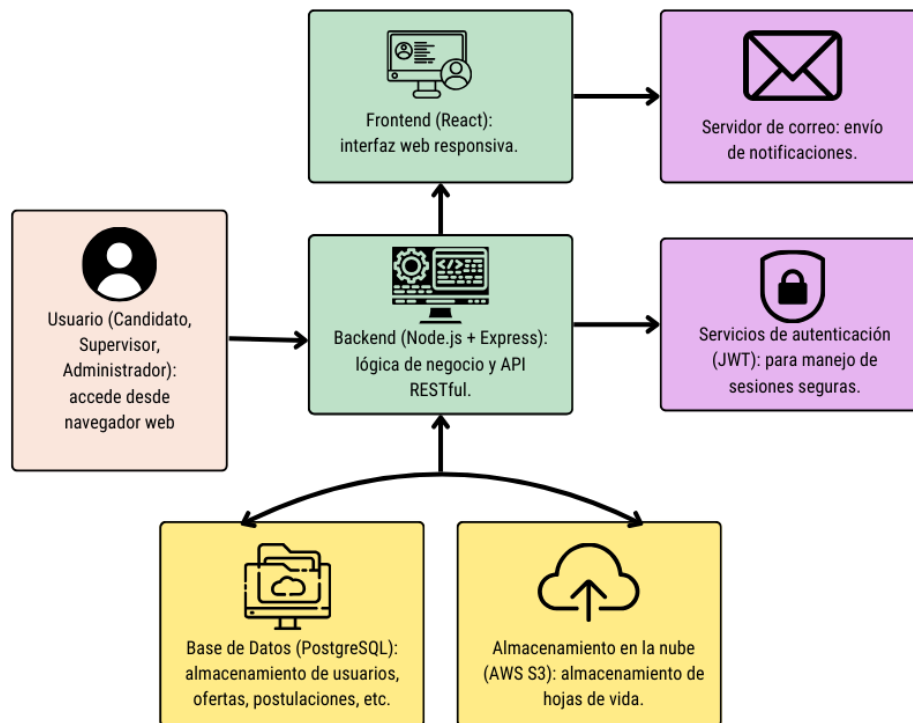
3.1 Modelo de Arquitectura

Monolítico/Microservicios

Justificación:

RHinnovate está en una etapa inicial y será desarrollado por un equipo reducido. Una arquitectura monolítica es más simple de implementar, mantener y desplegar. Permite tener todos los módulos (login, registro, ofertas, postulaciones, notificaciones) en un solo proyecto centralizado, lo que facilita el control y acelera el desarrollo.

3.2 Diagrama de Arquitectura



3.3 Mockup

RH INNOVATE

Crea tu cuenta

Ingresa tus datos

REGISTRATE

o continuar con



Google

Al hacer clic en continuar, acepta nuestros [Términos de servicio](#) y [Política de privacidad](#).



Pepito Perez
Candidato

Tus datos

[EDITAR](#)

Tu hoja de vida

HDV-PEPITO.pdf

[Cargar nueva HDV](#)

Tus postulaciones

#	VACANTE	FECHA - POSTULACION	ESTADO
P - 1	Operario de Maquinaria Plana	10/04/2025	En evaluación
P - 2	Supervisor de Producción	05/03/2025	Rechazado
P - 3	Inspector de Calidad Textil	15/04/2025	En evaluación

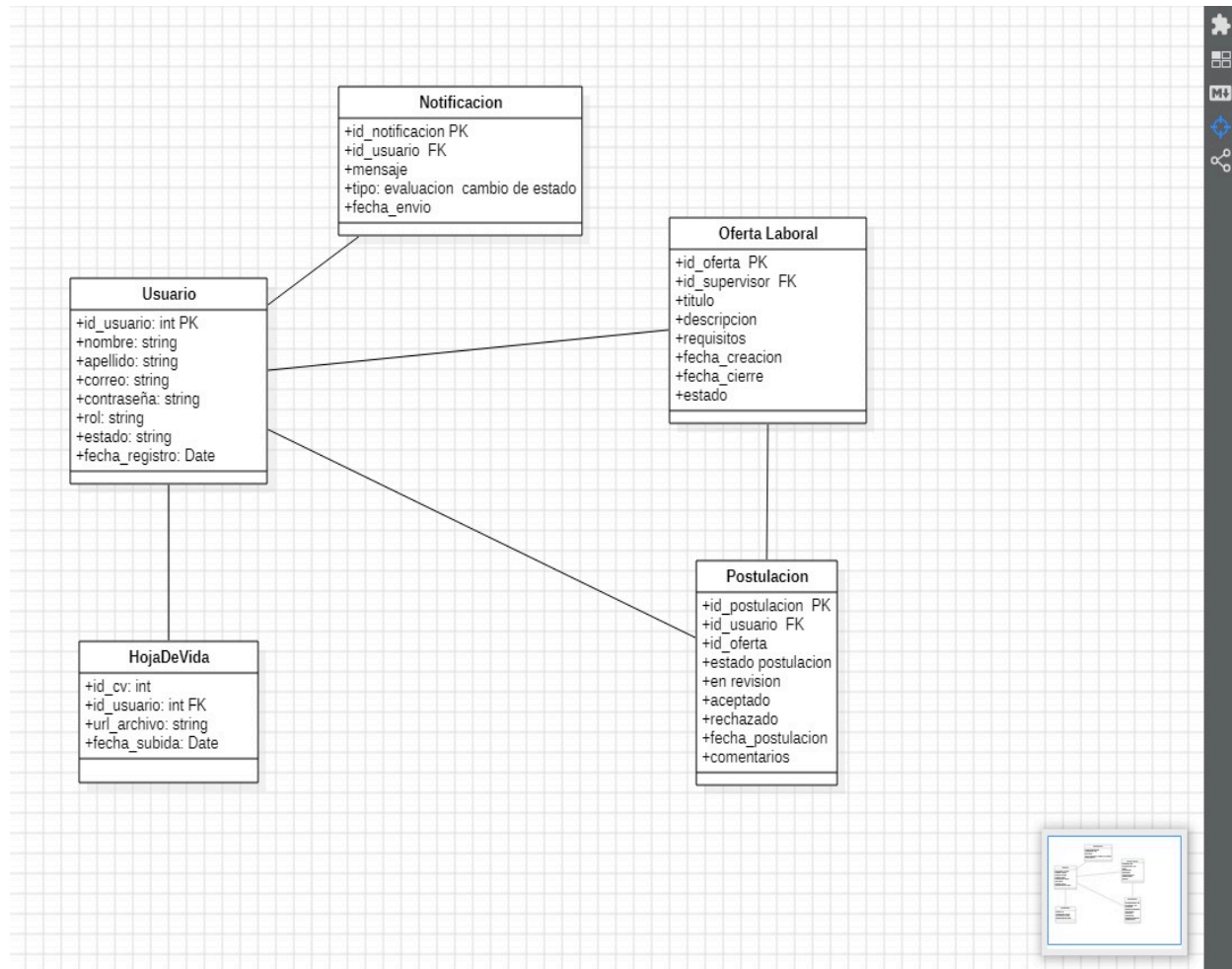
4.

4.1 Entidades y Relación

RELACIONES PRINCIPALES

Relación	Descripción
<ul style="list-style-type: none">• Un usuario puede recibir muchas notificaciones	1 a N
<ul style="list-style-type: none">• Un Supervisor puede crear muchas ofertas Laborales	1 a N
<ul style="list-style-type: none">• Un candidato puede hacer muchas Postulaciones	1 a N
<ul style="list-style-type: none">• Una oferta laboral puede tener muchas Postulaciones	1 a N
<ul style="list-style-type: none">• Una postulación pertenece a un único Candidato y una única Oferta Laboral	N a N
<ul style="list-style-type: none">• Un administrador puede gestionar muchos usuarios del sistema	1 a N
<ul style="list-style-type: none">• Un usuario tiene un único rol(Administrador, supervisor, Candidato)	1 a 1
<ul style="list-style-type: none">• Un supervisor puede revisar muchas postulaciones	1 a N

4.2 DIAGRAMA ENTIDAD RELACIÓN



4.3 ESPECIFICACIÓN DE TABLAS

TABLA CANDIDATO

Campo	Tipo de Dato	Restricciones
<u>id_candidato</u>	INT	PRIMARY KEY, AUTO INCREMENT
nombre	VARCHAR(50)	NOT NULL
correo	VARCHAR(50)	NOT NULL UNIQUE
contraseña	VARCHAR(100)	NULL(puede contener ruta o archivo)
<u>perfil_profesional</u>	VARCHAR(255)	NULL(descripción del candidato)

TABLA HOJA DE VIDA


Campo	Tipo de Dato	Restricciones
<u>id_cv</u>	INT	PRIMARY KEY, AUTO_INCREMENT
id_usuario	INT	FOREIGN KEY NOT NULL
<u>url_archivo</u>	VARCHAR(255)	NOT NULL(ruta o nombre del archivo almacenado)
<u>fecha_subida</u>	DATE	NOT NULL

TABLA SUPERVISOR

Campo	Tipo de Dato	Restricciones
<u>id_supervisor</u>	INT	PRIMARY KEY, AUTO_INCREMENT
nombre	VARCHAR(50)	NOT NULL
correo	VARCHAR(100)	NOT NULL, UNIQUE
contraseña	VARCHAR(50)	NOT NULL
cargo	VARCHAR(50)	NULL(nombre de cargo o rol)

TABLA ADMINISTRADOR

Campo	Tipo de Dato	Restricciones
<u>id_administrador</u>	INT	PRIMARY KEY, AUTO_INCREMENT
nombre	VARCHAR(50)	NOT NULL

correo	VARCHAR(100)	NOT NULL, UNIQUE 
contraseña	VARCHAR(50)	NOT NULL
<u>nivel_acceso</u>	VARCHAR(30)	DEFAULT

5. Restricciones del Proyecto

- Técnicas:
 - El sistema debe desarrollarse con las tecnologías seleccionadas: React.js para frontend y Node.js con Express para backend.
 - La base de datos principal será PostgreSQL.

- Se debe implementar bajo una arquitectura cliente-servidor monolítica en su primera versión.
- **Económicas:**
 - El proyecto tiene un presupuesto limitado, por lo tanto, se utilizarán herramientas y tecnologías de código abierto o gratuitas.
 - El alojamiento del sistema estará sujeto a recursos disponibles, inicialmente en servidores gratuitos o económicos.
- **De Tiempo:**
 - El sistema debe estar completamente funcional en un plazo máximo de 4 meses.
 - La metodología Scrum será utilizada para planificar entregas parciales mediante sprints quincenales.

6. Recomendaciones para el Futuro

- Migrar a una arquitectura de microservicios si el sistema crece en complejidad o volumen de usuarios, para mejorar el rendimiento y escalabilidad.
- Implementar una aplicación móvil en caso de alta demanda desde dispositivos móviles por parte de candidatos o supervisores.
- Incorporar analítica de datos para obtener métricas sobre postulaciones, ofertas más visitadas, tasas de conversión, entre otros.
- Automatizar procesos con IA, como la preselección de candidatos o recomendaciones de ofertas.

- Integrar con plataformas externas, como LinkedIn o servicios de correo para mejorar la difusión de las ofertas laborales.
- Asegurar el cumplimiento normativo, como la Ley de Protección de Datos Personales en Colombia (Habeas Data).

7. Referencias Técnicas

- Documentación oficial de React.js: <https://react.dev>
- Documentación oficial de Node.js y Express: <https://expressjs.com>
- Documentación de PostgreSQL: <https://www.postgresql.org/docs/>
- Guía Scrum – Scrum.org: <https://scrumguides.org>
- Ley 1581 de 2012 – Protección de Datos Personales (Colombia)

8. Licencias y Servicios Utilizados

Herramienta / Servicio	Tipo de Licencia	Observaciones
React.js	MIT	Uso libre para desarrollo web
Node.js + Express	MIT	Backend con API REST

PostgreSQL	Open Source (PostgreSQL License)	Motor de base de datos
GitHub	Gratuito (plan básico)	Repositorio de código
Vercel / Netlify	Gratuito (free tier)	Despliegue del frontend
Render / Railway	Gratuito (free tier)	Despliegue del backend

9. Software de Terceros Utilizado

Software / Librería	Función Principal	Licencia
React Router DOM	Navegación en frontend	MIT
Axios	Cliente HTTP para consumo de API	MIT
Bcrypt	Encriptación de contraseñas	MIT
JSON Web Token (JWT)	Autenticación	MIT
Dotenv	Manejo de variables de entorno	BSD-2

Express Validator	Validaciones en backend	MIT
Multer	Carga de archivos (hojas de vida)	MIT

10. Leyes y Normativas Aplicables

- Ley 1581 de 2012 (Colombia): Protección de datos personales. Regula la recolección, almacenamiento, uso y circulación de datos personales.
- Ley 1266 de 2008 (Colombia): Habeas Data financiero, aplicable si se gestionan antecedentes laborales.
- Ley 527 de 1999 (Colombia): Define y reglamenta el acceso y uso del comercio electrónico.
- Normativas de la empresa Miguel Caballero SAS: Aplicación de políticas internas de contratación y privacidad.
- Reglamento General de Protección de Datos (GDPR - Unión Europea): A considerar si la plataforma escala a nivel internacional.

11. Tiempos Estimados del Proyecto

Fase / Actividad	Duración Estimada	Fecha Aproximada
1. Planificación del proyecto	1 semana	Semana 1
2. Diseño de la base de datos	1 semana	Semana 2
3. Diseño UI/UX (Mockups)	1 semana	Semana 3
4. Desarrollo del frontend (React)	4 semanas	Semanas 4 a 7
5. Desarrollo del backend (Node.js)	4 semanas	Semanas 4 a 7
6. Integración frontend + backend	1 semana	Semana 8
7. Pruebas funcionales y corrección	2 semanas	Semanas 9 y 10
8. Documentación técnica y del usuario	1 semana	Semana 11

9. Despliegue y entrega final	1 semana	Semana 12
--------------------------------------	-----------------	------------------

12. Costos Asociados al Proyecto

Recurso / Servicio	Costo Estimado mensual	Total 3 meses
Hosting backend (Railway / Render)	\$50.000	\$150.000
Hosting frontend (Vercel / Netlify)	\$0	\$0
Base de datos (PostgreSQL)	\$0	\$0
Dominio web (.com)	\$45.000	\$45.000
Certificado SSL (Let's Encrypt)	\$0	\$0