



Regional Antioquia
Centro de Servicios Y Gestión Empresarial

INFORME TÉCNICO:

**Sistema Gestor de Turnos para la
Agencia Pública de Empleo, Basado
en Códigos QR e Inteligencia Artificial**

Autores: Alvaro Pérez Niño, Angel Stiven García Luna
y Samuel Leonardo Arias Rebolledo

Informe Técnico: Sistema Gestor de Turnos para la Agencia Pública de Empleo, basado en Códigos QR e Inteligencia Artificial

© Centro de Servicios y Gestión Empresarial – SENA Regional Antioquia

© Agencia Pública de Empleo – SENA Regional Antioquia

© Alvaro Perez Niño, Ángel Stiven Garcia Luna y Samuel Leonardo Arias Rebolledo

Centro de Servicios y Gestión Empresarial – SENA Regional Antioquia

Programa de Tecnólogo en Análisis y Desarrollo de Software

Calle 51 #57 - 70, La Candelaria

Medellín, Antioquia

Teléfono: +57 (601) 546 1500

www.sena.edu.co

Este manual técnico, no puede reproducirse sin la autorización expresa y por escrito del Centro de Servicios y Gestión Empresarial – SENA Regional Antioquia

Esta obra se encuentra bajo una Licencia [Creative Commons 4.0: Atribución/Reconocimiento – NoComercial – CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/)





INFORME TÉCNICO:

**Sistema Gestor de Turnos para la
Agencia Pública de Empleo, Basado
en Códigos QR e Inteligencia Artificial**

Autores: Alvaro Pérez Niño, Angel Stiven García Luna
y Samuel Leonardo Arias Rebolledo

Tabla de Contenido

| | |
|---|----|
| Introducción..... | 7 |
| 1. Identificación y descripción del problema..... | 8 |
| 2. Justificación..... | 9 |
| 3. Marco conceptual | 10 |
| 4. Objetivos | 12 |
| 5. Descripción metodología | 13 |
| 6. Desarrollo de la Aplicación..... | 14 |
| 7. Conclusiones | 24 |
| 8. Los Autores..... | 25 |
| 9. Bibliografía | 26 |
| 10. Anexos..... | 28 |

Lista de Figuras

| | |
|--|----|
| Figura 1: Diseño de la Arquitectura de la Aplicación Digiturno | 14 |
| Figura 2: Diagrama de Contexto de la Aplicación Digiturno | 15 |
| Figura 3: Diagrama de Contenedores de la Aplicación Digiturno | 16 |
| Figura 4: Diagrama de Componentes de la Aplicación Digiturno | 18 |
| Figura 5: Diseño de la Base de Datos Relacional - Digiturno..... | 20 |
| Figura 6: Pruebas de Calidad de la aplicación Digiturno | 20 |
| Figura 7: Diagrama de Despliegue de la Aplicación | 23 |

Introducción

El presente proyecto desarrolló un sistema de gestión de turnos sin contacto, utilizando códigos QR e inteligencia artificial, con el objetivo de optimizar los tiempos de espera y mejorar la experiencia del usuario. La Agencia Pública de Empleo del SENA Regional Antioquia atiende a un alto volumen de usuarios diariamente, lo que generaba cuellos de botella y largos tiempos de espera.

A través de la digitalización de los turnos y la implementación de un sistema inteligente, se mejoró la eficiencia del servicio, reduciendo la congestión y garantizando una atención más ágil e inclusiva. Este enfoque también contribuyó a la sostenibilidad ambiental mediante la eliminación del uso de tickets físicos.

La aplicación de metodologías ágiles permitió un desarrollo iterativo y flexible, asegurando que el sistema respondiera a las necesidades de los usuarios y de la entidad. En este contexto, se realizaron pruebas de usabilidad y capacitaciones para garantizar una correcta adopción del sistema.

1. Identificación y descripción del problema

En los sistemas tradicionales de gestión de turnos, la asignación de los mismos suele depender de procesos manuales o semi-automatizados que no optimizan el flujo de atención ni consideran variables clave, como la carga de trabajo del personal o las necesidades de los usuarios. Esto genera tiempos de espera prolongados y una experiencia de usuario deficiente. Además, los sistemas actuales a menudo se basan en la emisión de tickets físicos, lo que no solo aumenta la generación de residuos de papel, sino que también crea barreras de accesibilidad para personas con discapacidades motoras, visuales o cognitivas, dificultando su interacción con los servicios.

La Agencia Pública de Empleo del SENA Regional Antioquia enfrentaba retos significativos en la gestión de turnos debido a los métodos tradicionales utilizados. Anteriormente, los usuarios debían presentarse físicamente y obtener un turno de manera manual o a través de sistemas que no consideraban factores clave como la carga de trabajo del personal o las necesidades específicas de los ciudadanos.

Estos problemas generaban:

- Largos tiempos de espera y congestión en las instalaciones.
- Ineficiencia en la asignación de turnos debido a la falta de automatización.
- Barreras de accesibilidad para personas con discapacidad.
- Uso innecesario de papel, aumentando el impacto ambiental.

El problema radicaba en la falta de un sistema inteligente y automatizado que optimizara la asignación de turnos y brindara una experiencia de usuario eficiente e inclusiva.

2. Justificación

Este proyecto tuvo como objetivo desarrollar un sistema de gestión de turnos sin contacto, basado en códigos QR y con integración de inteligencia artificial para la optimización del tiempo de espera y la mejora de la experiencia de usuario. Este enfoque abordó varias problemáticas críticas:

Optimización del flujo de atención: La integración de algoritmos de IA permitió que los turnos se asignaran de manera dinámica y basada en la demanda, lo que redujo significativamente los tiempos de espera y mejoró la eficiencia del servicio. Estudios recientes han mostrado que el uso de IA en sistemas de asignación puede reducir los tiempos de espera hasta en un 30%.

Accesibilidad e inclusión: Al utilizar códigos QR y tecnologías móviles, se eliminaron las barreras físicas para la interacción con los sistemas de turnos, facilitando el acceso de personas con discapacidades. De acuerdo con la Resolución 01726 de 2014, se buscó garantizar la inclusión de personas con discapacidad en todos los sistemas institucionales, y este proyecto se alineó con esos principios.

Impacto ambiental positivo: La eliminación de los tickets físicos redujo el consumo de papel y la generación de residuos, contribuyendo a la sostenibilidad ambiental, un factor importante en la agenda global de desarrollo sostenible.

En resumen, este proyecto no solo mejoró la eficiencia operativa de los servicios, sino que también garantizó un entorno accesible y ecológicamente responsable, beneficiando tanto a la institución como a los usuarios.

3. Marco conceptual

3.1. Gestión de Turnos

La gestión de turnos se refiere al proceso de organizar y controlar el flujo de usuarios para que reciban atención en un servicio o institución de manera ordenada. Tradicionalmente, este proceso ha dependido de la emisión de tickets físicos, pero la transformación digital ha abierto nuevas posibilidades para su mejora, especialmente mediante el uso de tecnologías móviles y algoritmos de optimización.

3.2. Accesibilidad Digital

La accesibilidad digital implica diseñar productos y servicios que puedan ser utilizados por todas las personas, incluidas aquellas con discapacidades. En el contexto de sistemas de gestión de turnos, la accesibilidad debe garantizar que las personas con discapacidades físicas, sensoriales o cognitivas puedan interactuar con el sistema de manera autónoma y efectiva. Esto se puede lograr mediante el diseño de interfaces amigables y la integración de tecnologías asistidas, como lectores de pantalla y opciones de alto contraste.

3.3. Inteligencia Artificial en la Optimización de Turnos

La inteligencia artificial (IA) aplicada a la gestión de turnos permite analizar grandes volúmenes de datos y tomar decisiones en tiempo real para asignar los turnos de manera eficiente. La IA puede adaptarse a la demanda de los usuarios, optimizar los recursos disponibles y mejorar la experiencia general del usuario. Se ha demostrado que el uso de IA en la gestión de servicios reduce tiempos de espera y mejora la equidad en la distribución de recursos.

3.4. Experiencia de Usuario (UX)

La experiencia de usuario (UX) se refiere a cómo una persona percibe y experimenta la interacción con un producto o sistema. En el diseño de sistemas de gestión de turnos, la UX debe enfocarse en proporcionar una navegación intuitiva y accesible, garantizando que el sistema sea fácil de usar para personas de todas las habilidades y conocimientos tecnológicos.

3.5. Antecedentes

En la última década, la digitalización de servicios ha ganado un papel crucial en la modernización de las instituciones, especialmente en la gestión de turnos. La necesidad de mejorar la eficiencia operativa y la satisfacción del cliente ha llevado a muchas organizaciones a adoptar soluciones tecnológicas avanzadas. La implementación de tecnologías móviles y sistemas basados en la nube se ha convertido en una práctica común, permitiendo una gestión más ágil y efectiva de los servicios (Zhang et al., 2020). Este cambio ha sido especialmente relevante en sectores como la salud, donde las colas y los tiempos de espera prolongados pueden afectar directamente la atención al paciente (Lee et al., 2021).

El uso de sistemas de gestión de turnos sin contacto, como la implementación de códigos QR, ha emergido como una solución efectiva para minimizar el contacto físico y mejorar la experiencia del usuario. Esta tendencia se ha intensificado en respuesta a la pandemia de COVID-19, que aceleró la adopción de tecnologías sin contacto en diversos entornos (Wang & Kim, 2021). Sin embargo, a pesar de los avances en la digitalización, las personas con discapacidades siguen enfrentando barreras significativas en el acceso a servicios eficientes y amigables (Smith, 2019). Esta situación subraya la importancia de desarrollar sistemas inclusivos que consideren las necesidades de todos los usuarios.

Además, la creciente preocupación por el medio ambiente ha llevado a muchas instituciones a replantear su enfoque hacia la sostenibilidad. La reducción del uso de papel y la implementación de soluciones digitales son pasos fundamentales hacia prácticas más sostenibles en la gestión de turnos (Jones et al., 2020). De esta manera, el presente proyecto busca no solo innovar en la gestión de turnos, sino también contribuir a un entorno más accesible y sostenible.

Por lo tanto, el desarrollo de un sistema de gestión de turnos sin contacto, basado en códigos QR e inteligencia artificial, se alinea con las tendencias actuales en digitalización, accesibilidad e sostenibilidad. Este proyecto aspira a abordar las limitaciones existentes, proporcionando una solución que no solo mejore la eficiencia operativa, sino que también promueva la inclusión y el respeto al medio ambiente.

4. Objetivos

Implementar un sistema de información de gestión de turnos sin contacto, basado en códigos QR e inteligencia artificial, para optimizar el tiempo de espera y mejorar la experiencia de usuario.

- Diseñar un sistema de información de gestión de turnos sin contacto, basado en códigos QR y orientado a la optimización del tiempo de espera, garantizando accesibilidad y facilidad de uso.
- Desarrollar el sistema de información con integración de inteligencia artificial para la asignación inteligente de turnos, aplicando tecnologías que optimicen el flujo de atención y reduzcan los tiempos de espera.
- Desplegar el sistema de información de gestión de turnos en un entorno real, asegurando su funcionamiento eficiente, accesibilidad para personas con discapacidades y la eliminación total del uso de tickets físicos.

5. Descripción metodología

El desarrollo del sistema siguió la metodología ágil Scrum, que permitió iteraciones cortas y entregas parciales del producto para su validación continua. Se llevaron a cabo las siguientes fases:

- Planificación del proyecto: Definición de requerimientos y funcionalidades.
- Diseño y desarrollo: Implementación del sistema utilizando React para la interfaz, NodeJS y Flask para la lógica de negocio y PostgreSQL/ para la gestión de datos.
- Pruebas y validación: Evaluación del rendimiento, accesibilidad y usabilidad del sistema.
- Despliegue y capacitación: Implementación en un entorno real y capacitación de usuarios y administradores.

Con este enfoque, se aseguró un desarrollo ágil, adaptable a las necesidades de la Agencia Pública de Empleo del SENA Regional Antioquia, garantizando la eficacia del sistema y su alineación con los objetivos institucionales.

6. Desarrollo de la Aplicación

6.1. Requerimientos del Sistema

Las funcionalidades principales del sistema se definieron mediante historias de usuario; cada historia detalla las necesidades específicas de los interesados, traducidas en funcionalidades clave. Los criterios de aceptación asociados aseguran que cada funcionalidad cumpla con los requerimientos establecidos, priorizando la experiencia del usuario y la calidad del sistema, desde la creación de una landing page (página de inicio o aterrizaje) hasta la gestión y seguimiento de los turnos.

Para estructurar el desarrollo del sistema, se identificaron y clasificaron las historias de usuario en las siguientes categorías:

- Configuración – 6 historias de usuario
- Empleados – 8 historias de usuario
- Servicios – 11 historias de usuario
- Turnos – 19 historias de usuario
- Dashboard – 3 historias de usuario

Cada una de estas historias permitió definir con precisión los requerimientos de la plataforma, asegurando que las funcionalidades implementadas respondieran a las necesidades operativas del sistema y a la optimización de la gestión de turnos *Ver anexo - Historias de Usuario y Criterios de Aceptación.*

6.2. Diseño de la Arquitectura

La aplicación para la gestión de turnos de la APE se diseñó bajo una arquitectura RESTful, ideal para sistemas distribuidos como aplicaciones web. Este enfoque permitió una interacción eficiente y flexible entre el cliente y el servidor mediante protocolos HTTP estandarizados, ver la *Figura 1*:

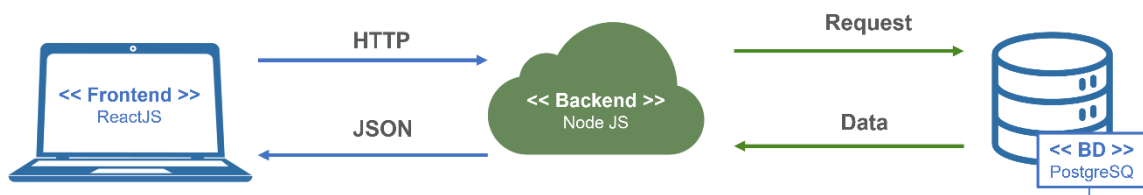


Figura 1: Diseño de la Arquitectura de la Aplicación Digiturno

6.3. Componentes de la Aplicación

Los componentes de la aplicación fueron gestionados con las siguientes Tecnologías:

- La *Administración* del Proyecto, se utilizó *JIRA* para gestionar tareas y realizar un seguimiento del progreso, asegurando una organización efectiva dentro del equipo de desarrollo.

- El *Repositorio y Control de Versiones*, se emplearon *Git* y *GitHub*, lo que permitió una colaboración eficiente entre los miembros del equipo y un control coherente de versiones durante todo el proceso de desarrollo.
- El *Editor de Codificación*, se adoptó *Visual Studio Code* debido a su versatilidad y soporte para múltiples lenguajes de programación, lo que facilitó la edición y el manejo del código fuente.
- El Desarrollo del *Frontend* se utilizó JavaScript con el framework React.JS y el Backend, se utilizó Node.JS, ofreciendo una lógica de negocio robusta y adaptable a las necesidades del sistema.
- En la Gestión de *Base de Datos*, se implementó PostgreSQL como gestor de bases de datos, destacándose por su capacidad para manejar volúmenes de datos estructurados y garantizar alta disponibilidad y rendimiento en el almacenamiento y recuperación de información.
- En la Seguridad, se implementaron métodos de verificación, autenticación y encriptación para proteger datos sensibles, garantizando la privacidad e integridad de la información.
- En la Comunicación entre Componentes, se establecieron protocolos REST y se utilizó el formato JSON para la transmisión de datos, garantizando una interacción fluida, eficiente y estandarizada entre los diferentes módulos del sistema.
- Los Diagramas de la Arquitectura de la aplicación fueron desarrollados bajo el enfoque de C4 Models, los cuales proporcionan una visión progresiva y detallada de la estructura del sistema a través de varios niveles de abstracción. Este método permitió visualizar claramente las relaciones y dependencias entre los distintos componentes del sistema desde una perspectiva amplia hasta un nivel más detallado.
 - El diagrama de contexto como se ilustra en la *Figura 2*, muestra la interacción del sistema con los usuarios finales y otros sistemas externos:



Figura 2: Diagrama de Contexto de la Aplicación Digiturno

- Los usuarios, incluyendo el administrador, los orientadores y los usuarios finales, gestionan las diversas acciones disponibles para administrar los turnos de los servicios de la Agencia Pública de Empleo (APE).
- La Aplicación Web, sirve como el punto central para el manejo de solicitudes y la entrega de funcionalidades a los usuarios.
- La BD SIPSE-APP, funciona como el almacenamiento principal de datos, asegurando que toda la información de suelos y estabilizantes esté centralizada y accesible.

- El diagrama de contenedores, como se muestra en la *Figura 3*, describe la arquitectura de la aplicación Digiturno en términos de los principales contenedores tecnológicos que lo conforman. Cada contenedor representa un conjunto lógico de responsabilidades que trabajan en conjunto para proporcionar las funcionalidades del sistema:

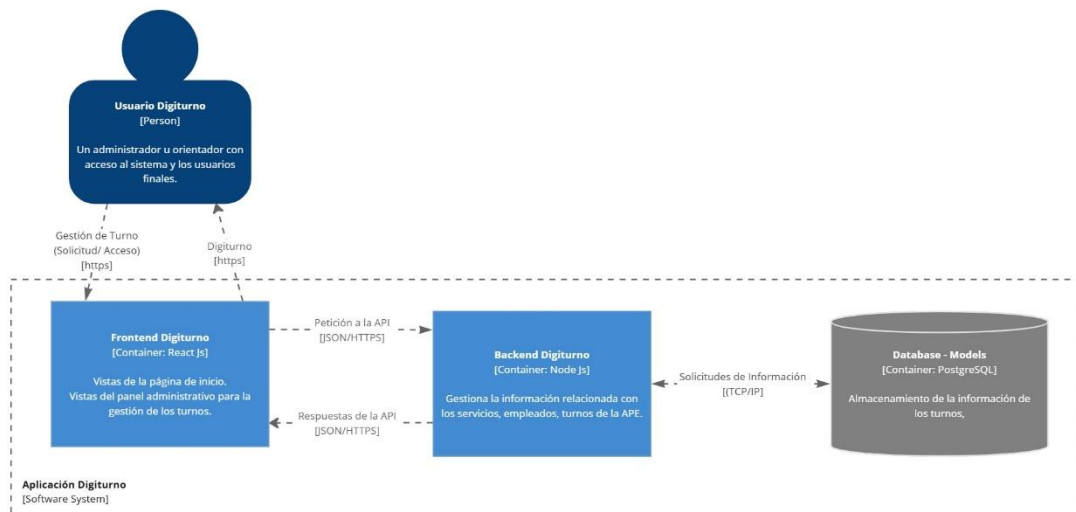


Diagrama de contenedor de la aplicación Digiturno
Aplicación para la gestión de turnos en la Agencia Pública de Empleo (APE).

Figura 3: Diagrama de Contenedores de la Aplicación Digiturno

- El Contenedor del Frontend, incluye los diferentes componentes desarrollados por el framework React.JS relacionados por la Landing Page y el Dashboard para la gestión de los turnos en la APE. Estos contenedores ofrecen las interfaces gráficas del sistema, permitiendo a los usuarios interactuar con las diferentes funcionalidades de la aplicación de manera intuitiva.

- El Contenedor del Backend implementado con Node.JS. Este contenedor gestiona la lógica de negocio, procesa las solicitudes de los usuarios, se comunica con la base de datos y proporciona las respuestas correspondientes.
 - El Contenedor de Base de Datos PostgreSQL, este contenedor almacena y gestiona toda la información relacionada con la gestión de los turnos en la APE. Es el núcleo de almacenamiento de datos del sistema, y trabaja en estrecha colaboración con el backend para garantizar el acceso eficiente y seguro a la información.
 - La comunicación entre los contenedores se realiza mediante el uso de protocolos HTTP y JSON, asegurando una interacción eficiente entre el frontend, el backend y la base de datos. Este diseño modular y distribuido asegura que el sistema sea escalable, flexible y fácil de mantener.
- El diagrama de componentes, como se muestra en la *Figura 4*, detalla la organización interna y las interacciones entre los principales módulos de la aplicación Digiturno. Cada componente cumple una función específica, facilitando la integración y comunicación en el sistema:
- *Landing Page Component*: Presenta información general sobre la aplicación Digiturno, brindando a los usuarios una visión clara de sus funcionalidades y beneficios.
 - *Solicitud de Turnos Component*: Permite a los usuarios solicitar turnos para acceder a los servicios de la Agencia Pública de Empleo (APE) de manera digital y sin contacto, optimizando el proceso de atención.
 - *Autenticación Component*: Administra los procesos de inicio de sesión y restablecimiento de contraseña, garantizando la seguridad en el acceso a la plataforma.
 - *Dashboard Component*: Funciona como el panel administrativo para la gestión de turnos, empleados y servicios, facilitando la supervisión y control de la plataforma por parte de los administradores y orientadores.
 - *Perfil Component*: Gestiona la información personal de los usuarios, permitiéndoles actualizar datos como nombres, apellidos, correo electrónico y contraseña.
 - *Empleados Component*: Permite administrar el CRUD de los orientadores adscritos a la APE, asegurando un registro actualizado de los responsables de la atención.

- **Servicios Component:** Gestiona el CRUD de los servicios disponibles en la aplicación, permitiendo su actualización y asignación a los turnos según la demanda de los usuarios.
- **Turnos Component:** Administra el historial de turnos de los usuarios, proporcionando herramientas para la organización, seguimiento y priorización de turnos, con el objetivo de mejorar la eficiencia del servicio.
- **Database – Models:** Utiliza PostgreSQL para almacenar la información del sistema, incluyendo datos de empleados, servicios y turnos, asegurando su disponibilidad y seguridad.



Diagrama de componentes de la aplicación Digiturno
Aplicación para la gestión de turnos en la Agencia Pública de Empleo (APE).

Figura 4: Diagrama de Componentes de la Aplicación Digiturno

Estos componentes se comunican a través de una arquitectura RESTful, garantizando la interacción eficiente entre el frontend y el backend de la aplicación. Además, el sistema emplea métodos de seguridad para la autenticación y el manejo seguro de datos, brindando una experiencia confiable y optimizada para los usuarios y administradores.

6.4. Diseño de la Base de Datos

La base de datos de la aplicación Digiturno ha sido diseñada utilizando PostgreSQL, garantizando un almacenamiento estructurado y eficiente de la información. Su arquitectura permite gestionar los datos de los empleados, servicios y turnos de manera segura, optimizando el acceso y la integridad de la información.

La base de datos está compuesta por diversas tablas que almacenan información clave del sistema, incluyendo:

- *Usuarios y autenticación:* Contiene los datos de los administradores, orientadores y usuarios, asegurando un acceso seguro mediante credenciales encriptadas.
- *Empleados:* Registra a los orientadores y su relación con los servicios que pueden gestionar.
- *Servicios:* Almacena los diferentes servicios que ofrece la Agencia Pública de Empleo (APE), permitiendo su actualización y asignación.
- **Turnos:** Gestiona los turnos asignados a los usuarios, su historial y estado en tiempo real, facilitando el seguimiento y optimización del flujo de atención.
- El diseño de la base de datos permite una comunicación fluida con el backend a través de consultas SQL optimizadas, asegurando tiempos de respuesta rápidos y una administración eficiente del sistema.

A continuación, en la *Figura 5*, se presenta el Diagrama de la Base de Datos Relacional - Digiturno, donde se visualizan las tablas principales y sus relaciones dentro del sistema.

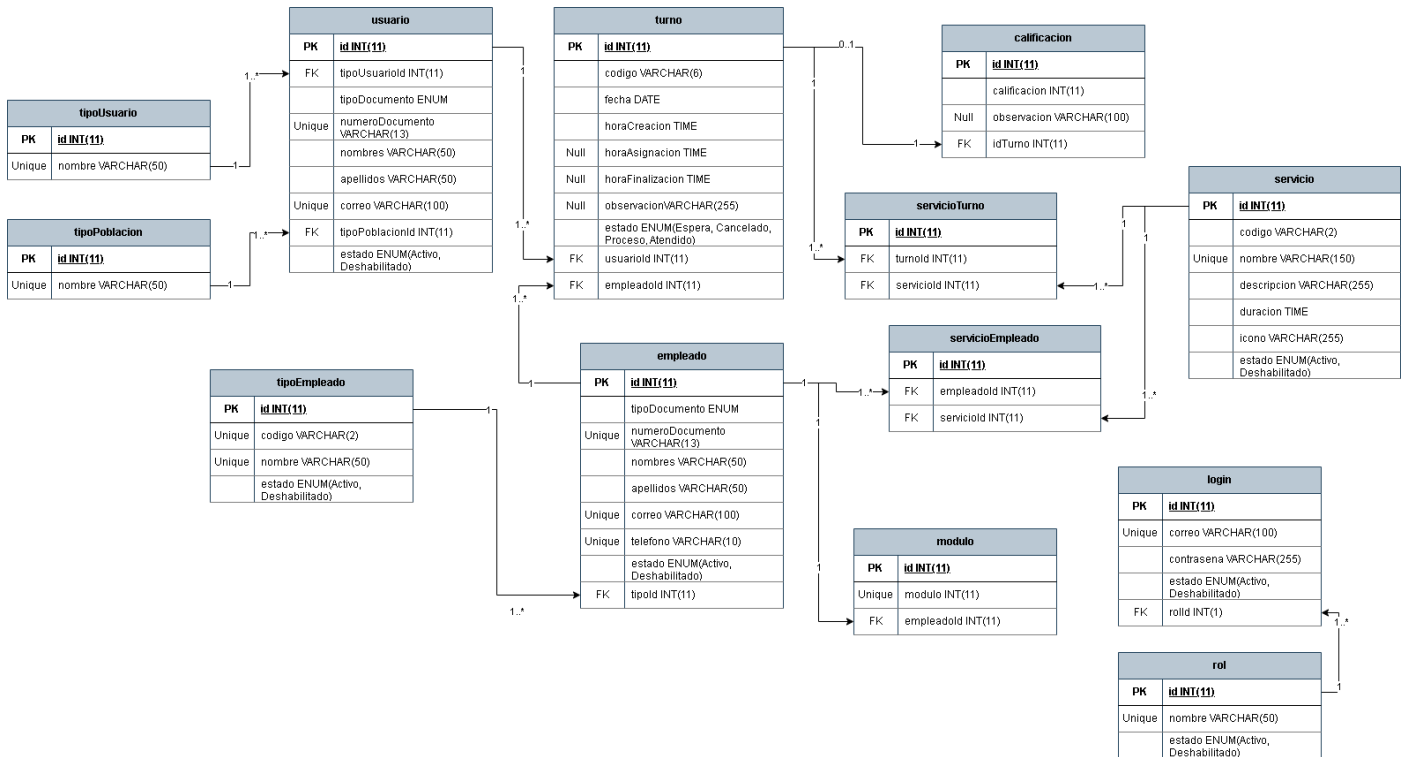


Figura 5: Diseño de la Base de Datos Relacional - Digiturno

6.5. Pruebas de Calidad

Para garantizar el correcto funcionamiento y la optimización de la aplicación Digiturno, se realizaron pruebas de calidad enfocadas en rendimiento, accesibilidad, prácticas recomendadas y SEO, como se muestra en la *Figura 6*. Estas pruebas permitieron identificar oportunidades de mejora en la eficiencia del sistema y su adaptabilidad a diferentes dispositivos.

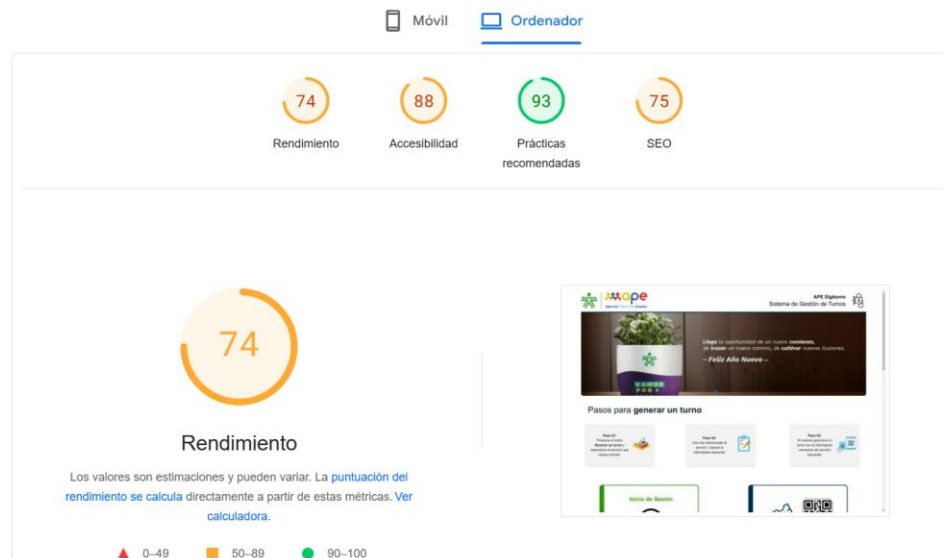


Figura 6: Pruebas de Calidad de la aplicación Digiturno

6.5.1. Evaluación de Rendimiento

Se llevó a cabo un análisis de rendimiento utilizando Google Lighthouse, obteniendo una puntuación de 74 en la evaluación para ordenadores. Este resultado indica que el sistema presenta un desempeño aceptable, pero con margen de mejora en términos de tiempos de carga y optimización de recursos. Se identificaron áreas clave a optimizar, como la reducción del tamaño de las imágenes, la mejora en la carga de scripts y la minimización del tiempo de respuesta del servidor.

6.5.2. Accesibilidad

La accesibilidad de la aplicación alcanzó una puntuación de 88, lo que demuestra que el sistema está diseñado para ser utilizado por una amplia variedad de usuarios, incluyendo aquellos con discapacidades. Sin embargo, se recomienda la implementación de mejoras en contraste de colores, etiquetas de accesibilidad y compatibilidad con lectores de pantalla para alcanzar una calificación óptima.

6.5.3. Prácticas Recomendadas

En términos de prácticas recomendadas, la aplicación obtuvo una calificación de 93, lo que indica un cumplimiento alto de estándares de desarrollo web, buenas prácticas de seguridad y estabilidad en la ejecución del código. Se recomienda mantener actualizadas las dependencias y revisar periódicamente los estándares de seguridad aplicados.

6.5.4. Evaluación de SEO

El análisis SEO arrojó una puntuación de 75, lo que indica que la aplicación está optimizada para motores de búsqueda, pero aún presenta oportunidades de mejora en indexación, metadatos y estructura semántica del contenido. Se sugiere la optimización de etiquetas meta, la mejora de tiempos de carga y la implementación de estrategias para aumentar la visibilidad del sistema en buscadores.

Las pruebas de calidad permitieron validar el desempeño de la aplicación y definir estrategias para optimizar su rendimiento. Se recomienda la implementación de ajustes en la carga de recursos, accesibilidad y SEO para mejorar la experiencia del usuario y garantizar una mayor eficiencia operativa.

6.6. Despliegue

El despliegue de la aplicación Digiturno se ha diseñado bajo una arquitectura RESTful, garantizando una comunicación eficiente entre los diferentes componentes del sistema. La infraestructura del despliegue permite la escalabilidad, seguridad y disponibilidad del servicio, optimizando la asignación de turnos mediante códigos QR e inteligencia artificial.

6.6.1. Arquitectura de Despliegue

La aplicación sigue un modelo cliente-servidor en el cual los usuarios acceden a la plataforma a través de un navegador web o dispositivo móvil. La comunicación entre el frontend y el backend se realiza mediante solicitudes HTTP y respuestas en formato JSON, asegurando tiempos de respuesta rápidos y compatibilidad con múltiples dispositivos.

Los principales componentes en el proceso de despliegue incluyen:

- Frontend (Digiturno Web App): Desarrollado en **React.js**, este componente ofrece una interfaz intuitiva y accesible para la gestión de turnos. Se aloja en un servidor web con capacidad de escalado automático, lo que permite atender a múltiples usuarios simultáneamente.
- Backend (API REST): Implementado en **Node.js**, este componente gestiona la lógica de negocio y se encarga del procesamiento de solicitudes de los usuarios. La API permite la autenticación, gestión de turnos, asignación de servicios y consulta de información en tiempo real.
- Inteligencia Artificial (IA para Optimización de Turnos): Se ha integrado un módulo de **IA** que analiza la demanda de turnos y ajusta dinámicamente la distribución de citas, priorizando a los usuarios según criterios predefinidos.
- Base de Datos (PostgreSQL): El sistema utiliza **PostgreSQL** como gestor de base de datos, garantizando la integridad y disponibilidad de la información de turnos, usuarios y empleados. La base de datos se encuentra alojada en un servidor con replicación y copias de seguridad automáticas para evitar pérdida de datos.
- Servicios en la Nube: La aplicación está conectada con un servicio en la nube que permite el almacenamiento de logs, monitoreo de métricas y escalado automático de los servidores en función de la demanda.

6.6.2. Flujo de Despliegue

- El usuario accede a la aplicación web a través de su navegador o dispositivo móvil.
- Se establece la comunicación con la API REST, enviando y recibiendo datos en formato JSON.
- La API interactúa con la base de datos y el módulo de IA, procesando la información y optimizando la asignación de turnos.
- La información es enviada de vuelta al frontend, actualizando la interfaz en tiempo real.

Este modelo de despliegue garantiza un servicio confiable y accesible, permitiendo a la Agencia Pública de Empleo (APE) optimizar la atención a los usuarios y mejorar la eficiencia operativa.

A continuación, en la *Figura 7*, se presenta el Diagrama de Despliegue, ilustrando la interacción entre los diferentes componentes del sistema.

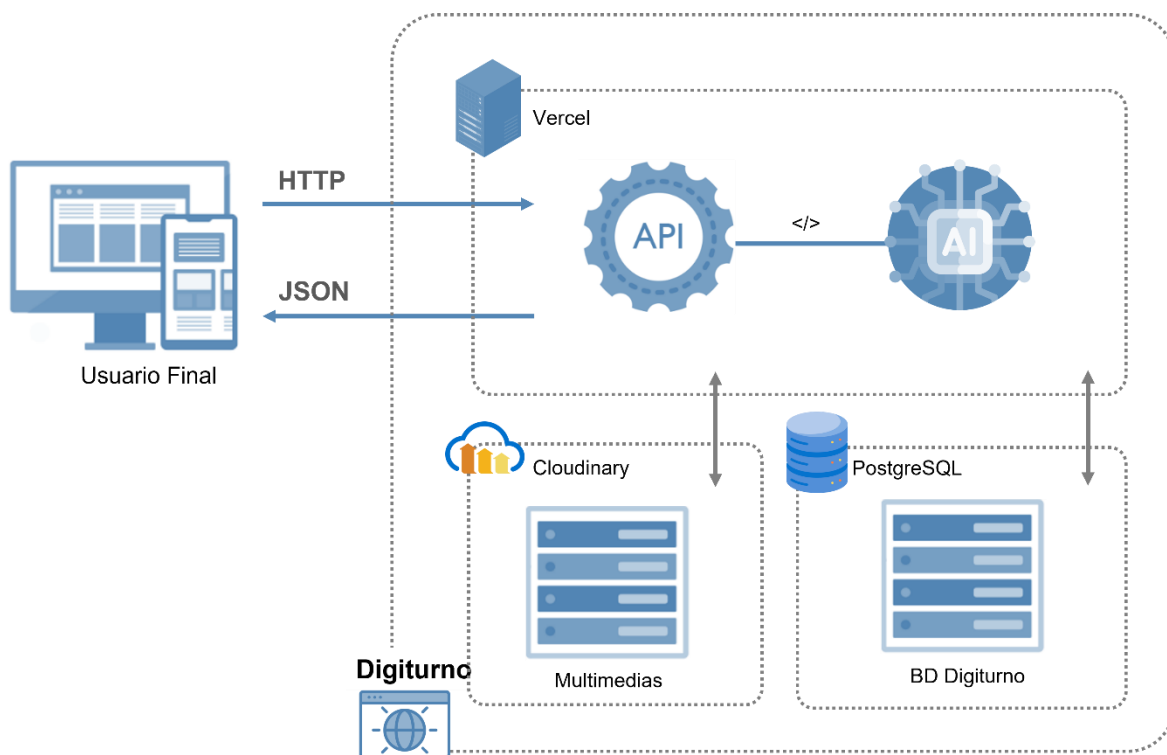


Figura 7: Diagrama de Despliegue de la Aplicación

7. Conclusiones

El desarrollo del Sistema Gestor de Turnos para la Agencia Pública de Empleo (APE) del SENA Regional Antioquia ha permitido la implementación de una solución tecnológica innovadora basada en códigos QR e inteligencia artificial, optimizando los tiempos de espera y mejorando la experiencia de los usuarios.

La digitalización del proceso de gestión de turnos ha representado un avance significativo en términos de eficiencia operativa, accesibilidad y sostenibilidad ambiental. La eliminación de tickets físicos ha reducido el impacto ambiental, alineándose con las políticas de sostenibilidad y modernización digital de la institución.

Desde el punto de vista técnico, la aplicación se diseñó con una arquitectura RESTful, utilizando React.js para el frontend, Node.js para el backend y PostgreSQL como gestor de base de datos, lo que garantiza una solución escalable, flexible y segura. Además, la integración de inteligencia artificial ha permitido una distribución dinámica y equitativa de los turnos, optimizando la atención a los usuarios en función de la demanda y las necesidades de cada servicio.

Las pruebas de calidad realizadas evidenciaron un desempeño positivo en términos de accesibilidad, prácticas recomendadas y SEO, aunque se identificaron oportunidades de mejora en la optimización del rendimiento. Se recomienda la implementación de ajustes en la carga de recursos, la mejora en tiempos de respuesta del servidor y la optimización de imágenes para incrementar la velocidad de la aplicación.

El despliegue del sistema en entornos cloud ha facilitado su disponibilidad y mantenimiento, garantizando su funcionamiento en tiempo real y permitiendo futuras escalabilidades. La adopción de metodologías ágiles en el desarrollo permitió iteraciones constantes, asegurando una solución adaptada a las necesidades de la APE y sus usuarios.

En conclusión, la implementación de Digiturno ha logrado una transformación digital efectiva en la gestión de turnos dentro de la Agencia Pública de Empleo, proporcionando una experiencia más eficiente, accesible e innovadora para los ciudadanos. Se recomienda continuar con mejoras en la optimización del rendimiento y explorar futuras integraciones con otros sistemas para potenciar aún más su impacto y escalabilidad.

8. Los Autores

Ángel Stiven Garcia Luna. Técnico en Programación de Software y Estudiante de Análisis y Desarrollo de Software del Servicio Nacional de Aprendizaje SENA. Actualmente Aprendiz Investigador del grupo de investigación GIGAT y del Semillero de Investigación MERLIN. Sus áreas de interés se enmarcan en la Ingeniería del software y sistemas de información e innovación tecnológica.

Correo electrónico: angelstivengarcialuna@gmail.com

Samuel Leonardo Arias Rebolledo. Técnico en Programación de Software y Estudiante de Análisis y Desarrollo de Software del Servicio Nacional de Aprendizaje SENA. Actualmente Aprendiz Investigador del grupo de investigación GIGAT y del Semillero de Investigación MERLIN. Sus áreas de interés se enmarcan en la Ingeniería del software y sistemas de información e innovación tecnológica.

Correo electrónico: samueleleonardo.escuela@gmail.com

Alvaro Perez Niño. Ingeniero de Sistemas de la Universidad Cooperativa de Colombia y Magister en Ciencia de Datos de la Pontificia Universidad Javeriana, Actualmente Instructor Investigador del grupo de investigación GIGAT y del Semillero de Investigación MERLIN en el SENA Regional Antioquia, del centro de Servicios y Gestión Empresarial, Sus áreas de interés se enmarcan en la Ingeniería del software y sistemas de información, ciencia de datos e innovación tecnológica.

Correo electrónico: aperezn@sena.edu.co

9. Bibliografía

R. Kuo, "Optimization of Queue Management System with AI," IEEE Transactions on Service Computing, vol. 12, no. 3, pp. 407-418, 2020.

Resolución 01726 de 2014, "Política Institucional para Atención de las Personas con Discapacidad," Ministerio de Educación, 2014.

J. Smith, "The Role of Technology in Sustainable Practices," IEEE Access, vol. 7, pp. 123456-123469, 2019.

A. Brown, "Digital Queue Management Systems: A Review," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 50, no. 4, pp. 2232-2241, 2020.

Y. Chen, "Artificial Intelligence in Queue Management: A Case Study," IEEE International Conference on Big Data (Big Data), 2019, pp. 1021-1026.

S. Williams, "User Experience and Accessibility in Digital Systems," IEEE Transactions on Human-Machine Systems, vol. 51, no. 2, pp. 315-325, 2021.

L. Zhang, "Designing Accessible User Interfaces," IEEE Software, vol. 38, no. 6, pp. 29-37, 2021.

Jones, M., Taylor, R., & Smith, P. (2020). Sustainability in service management: The role of digital solutions. *International Journal of Service Industry Management*, 31(2), 245-267.

Lee, T., Park, S., & Choi, J. (2021). Improving patient flow in healthcare settings: The role of digital technology. *Health Informatics Journal*, 27(1), 3-15.

Smith, J. (2019). Access to services: Challenges for people with disabilities. *Journal of Social Inclusion*, 12(3), 45-58.

Wang, S., & Kim, H. (2021). The rise of contactless technology: Transforming customer experiences in retail. *Journal of Retailing and Consumer Services*, 58, 102234.

Zhang, Y., Lee, T., & Johnson, R. (2020). Digital transformation in public services: Improving efficiency through technology. *Government Information Quarterly*, 37(4), 101-114.

10. Anexos

10.1. Historias de Usuario y Criterios de Aceptación

| Código Historia Usuario | Redacción Historia de Usuario | Código Criterio de Aceptación | Criterios de Aceptación |
|-------------------------|--|-------------------------------|---|
| HU_01 | Yo como usuario necesito especificar el servicio que necesito de la APE para poder recibir un turno dependiendo del motivo de mi visita. | CA_HU01_01 | Solo se listarán los servicios Activos y asignados como mínimo a un empleado. |
| | | CA_HU01_02 | El servicio seleccionado no se guardará hasta que se cree el turno. |
| | | CA_HU01_03 | Se mostrará un modal de éxito o error dependiendo del resultado del proceso. |
| HU_02 | Yo como usuario quiero proporcionar mi correo para poder recibir un seguimiento del turno que voy a pedir. | CA_HU02_01 | El usuario debe ingresar el email para poder recibir un seguimiento de su turno. |
| | | CA_HU02_02 | El valor no tiene un formato de correo no se enviará el formulario. |
| | | CA_HU02_03 | Se mostrará un modal de éxito o error dependiendo del resultado del proceso. |
| HU_03 | Yo como usuario necesito solicitar un turno para poder recibir el servicio por parte de la APE. | CA_HU03_01 | El turno no tendrá 2 id iguales. |
| | | CA_HU03_00 | Se creará un cadena de 5-6 caracteres para identificar el turno; concatenará el código del servicio junto con el código del tipo de empleado que creó el turno; adicionalmente un numero de 3 digitos (ej. 001) |
| | | CA_HU03_01 | El usuario deberá proporcionar su email para enviar el seguimiento del turno al correo. |
| | | CA_HU03_02 | El usuario puede indicar mediante un checkbox si tiene prioridad. |
| | | CA_HU03_03 | Solo puede crearse un turno por usuario a la vez. |
| | | CA_HU03_04 | Al crearse el turno, los empleados podrán tomar el turno. |
| | | CA_HU03_05 | Si la validación del registro no se cumple, no se creará el turno. |
| | | CA_HU03_06 | Se guardarán los datos del usuario en el aplicativo. |
| | | CA_HU03_07 | Si cancela el proceso o sale de la pestaña no se guardarán los datos del registro. |
| | | CA_HU03_08 | Se mostrará un modal de éxito o error dependiendo del resultado del proceso. |
| HU_03.1 | Yo como usuario necesito validar el registro del usuario en la página de la APE para poder solicitar un turno. | CA_HU03.2_01 | Se hará uso del número de identidad y correo previamente ingresado. |
| | | CA_HU03.2_02 | Se hará uso del Web Scraping para validar si el usuario se ha registrado previamente en la APE. |

| | | | |
|---------|---|--------------|--|
| | | CA_HU03.2_03 | Se hará un registro con el ID del usuario en la página de la APE para validar si su usuario ya existe. |
| | | CA_HU03.2_04 | No se guardará ningún dato obtenido desde la APE. |
| | | CA_HU03.2_05 | Si el usuario no se ha registrado en la APE, se le agregará como servicio extra. |
| | | CA_HU03.2_06 | Se mostrará un modal de éxito o error dependiendo del resultado del proceso. |
| HU_04 | Yo como usuario necesito consultar el turno vigente para poder obtener información sobre mi turno. | CA_HU04_01 | Al crear el turno se mostrará la información del turno del usuario. |
| | | CA_HU04_02 | Se creará un QR para obtener los datos del turno recién creado. |
| | | CA_HU04_03 | Se mostrará un modal de éxito o error dependiendo del resultado del proceso. |
| HU_04.1 | Yo como usuario quiero escanear el código QR de mi turno para poder realizar el seguimiento de mi turno. | CA_HU04.1_01 | El usuario podrá escanear el código con su celular. |
| | | CA_HU04.1_02 | El usuario será redirigido a una vista donde podrá consultar la información de su turno. |
| | | CA_HU04.1_03 | El qr se vencerá después de un día desde su creación. |
| | | CA_HU04.1_04 | El qr no será obligatorio para el proceso. |
| HU_05 | Yo como usuario quiero tener un seguimiento de mi turno para poder tener un conocimiento de aproximadamente cuándo y dónde será atendido por un empleado. | CA_HU05_01 | El usuario recibirá un correo al email proporcionado acceso al seguimiento del turno. |
| | | CA_HU05_02 | El email se enviará al crearse el turno. |
| HU_06 | Yo como usuario quiero recibir información de atención recibida en el turno para poder contar con un historial de la atención del servicio prestado. | CA_HU06_01 | El usuario recibirá un correo al email proporcionado mostrando la información de su turno terminado. |
| | | CA_HU06_02 | El email solo se enviará cuando el empleado encargado termine el turno recién atendido. |
| | | CA_HU06_03 | Se le hará saber al usuario que es su turno con el televisor ubicado en la sala. |
| HU_07 | Yo como usuario quiero calificar el servicio recibido para poder ofrecer un feedback que busque mejorar el servicio. | CA_HU07_01 | El formulario será enviado al usuario cada vez que se atienda un turno. |
| | | CA_HU07_02 | Solo puede hacer una calificación por turno. |
| | | CA_HU07_03 | Deberá ingresar un valor entre 1 y 5 representado en estrellas. |
| | | CA_HU07_04 | La observación es opcional. |
| | | CA_HU07_05 | Se mostrará un modal de éxito o error dependiendo del resultado del proceso. |

| | | | |
|--------------|---|-------------------|---|
| HU_08 | Yo como usuario quiero ver los turnos tomados para poder saber cuando es mi turno y recibir el servicio. | CA_HU08_01 | La vista mostrará todos los turnos que han sido tomados por los empleados en el día actual. |
| | | CA_HU08_02 | Al atender un turno se retirará de la lista. |
| | | CA_HU08_03 | Los turnos tomados estarán organizados del más reciente al más antiguo |
| | | CA_HU08_04 | Cada turno tendrá el logo del servicio escogido, el código del turno, el nombre del usuario, el estado del turno representado con un color y módulo en el que el usuario recibirá el asesoramiento. |
| | | CA_HU08_05 | Cuando un empleado toma un turno, se mostrará en la vista un modal indicando el turno recién tomado, para que el usuario sepa donde recibirá el servicio. |
| HU_09 | Yo como administrador/orientador necesito iniciar sesión para poder acceder a las funciones del aplicativo. | CA_HU09_01 | El administrador estará creado previamente. |
| | | CA_HU09_02 | El administrador tendrá que ingresar el correo y contraseña para ingresar. |
| | | CA_HU09_03 | Si el empleado está desactivado, no podrá iniciar sesión. |
| | | CA_HU09_04 | Al iniciar sesión, se reiniciará el módulo en caso de que tenga uno asignado. |
| | | CA_HU09_05 | Se mostrará un modal de éxito o error dependiendo del resultado del proceso. |
| HU_10 | Yo como administrador/orientador necesito visualizar mis datos personales para poder corroborar la veracidad de los mismos. | CA_HU10_01 | Se mostrará el nombre, el correo, el n° de documento y el celular del usuario. |
| | | CA_HU10_02 | No se mostrará la contraseña ni el estado del usuario. |
| HU_11 | Yo como administrador/orientador necesito editar mis datos para poder actualizar los datos de mi contacto. | CA_HU11_01 | El usuario podrá editar únicamente el correo y el celular. |
| | | CA_HU11_02 | Si los datos del registro no cumplen con la validación no se cambiarán los datos. |
| | | CA_HU11_03 | El registro tendrá validaciones en tiempo real. |
| | | CA_HU11_04 | Se mostrará un modal de éxito o error dependiendo del resultado del proceso. |
| HU_12 | Yo como administrador/orientador necesito cambiar mi contraseña para poder garantizar la seguridad de mi cuenta. | CA_HU12_01 | El usuario tendrá que ingresar su contraseña actual para actualizar la contraseña. |
| | | CA_HU12_02 | EL usuario tendrá que confirmar la nueva contraseña. |
| | | CA_HU12_03 | Si los datos del registro no cumplen con la validación no se guardarán los cambios. |
| | | CA_HU12_04 | Se mostrará un modal de éxito o error dependiendo del resultado del proceso. |
| HU_13 | Yo como administrador/orientador quiero cerrar la sesión para poder restringir el acceso a mi cuenta. | CA_HU13_01 | Al cerrar la sesión se borrará el token que mantiene la sesión abierta y permite el acceso al aplicativo. |
| | | CA_HU13_02 | Si la sesión está cerrada ningún posterior podrá ingresar a la cuenta. |
| | | CA_HU13_03 | El administrador debe haber iniciado sesión para poder cerrar sesión. |

| | | | |
|-------|--|------------|---|
| | | CA_HU13_04 | Al cerrar la sesión, el usuario será redirigido a la página de inicio. |
| HU_14 | Yo como administrador/orientador necesito recuperar mi contraseña en caso de que la olvide para poder re-ingresar al aplicativo. | CA_HU14_01 | EL usuario ingresará el correo con el que está registrado en el aplicativo. |
| | | CA_HU14_02 | Si el correo no esta registrado no se enviará ningún código de confirmación. |
| | | CA_HU14_03 | El usuario deberá ingresar un código (enviado al correo) para validar que el usuario es propietario del correo. |
| | | CA_HU14_04 | Si el código es válido, el usuario deberá ingresar la nueva contraseña. |
| | | CA_HU14_05 | Se mostrará un modal de éxito o error dependiendo del resultado del proceso. |
| HU_15 | Yo como administrador quiero registrar orientadores al aplicativo para poder ofrecerles acceso al aplicativo y monitorear sus aportes. | CA_HU15_01 | No habrá 2 empleados con un mismo id. |
| | | CA_HU15_02 | Si no le da al botón "Crear Empleado" no se guardará el registro del empleado. |
| | | CA_HU15_03 | Si el registro no cumple con las validaciones no se creará el empleado. |
| | | CA_HU15_04 | El empleado se creará con el tipo de empleado "Orientador" por defecto. |
| | | CA_HU15_05 | El registro tendrá validaciones en tiempo real. |
| | | CA_HU15_06 | Si cancela el proceso o sale de la pestaña no se guardarán los datos del registro. |
| | | CA_HU15_07 | Se mostrará un modal de éxito o error dependiendo del resultado del proceso. |
| HU_16 | Yo como administrador quiero listar los empleados vigentes en el aplicativo para poder tener conocimiento de cuáles empleados tienen acceso al aplicativo. | CA_HU16_01 | Se mostrarán todos los empleados registrados en el aplicativo. |
| | | CA_HU16_02 | El empleado "En espera" estará quemado en el aplicativo. |
| | | CA_HU16_03 | Solo el administrador podrá listar los empleados. |
| | | CA_HU16_04 | El administrador no podrá ver la contraseña de los empleados. |
| | | CA_HU16_05 | La tabla de empleados se paginará al llegar a un tope determinado. |
| | | CA_HU16_06 | Se mostrará un modal de éxito o error dependiendo del resultado del proceso. |
| HU_17 | Yo como administrador quiero buscar entre los empleados vigentes en el aplicativo para poder encontrarlo con agilidad. | CA_HU17_01 | Se filtrarán los empleados por todos los campos disponibles. |
| | | CA_HU17_02 | Si el buscador está vacío no se aplicará ningún filtro. |
| | | CA_HU17_03 | Si ningún registro concuerda con el buscador, no se mostrará ningún registro. |
| HU_18 | Yo como administrador necesito asignar los servicios de cada empleado para poder determinar los turnos que puede tomar un empleado segun sus capacidades. | CA_HU18_01 | Por defecto cada empleado tendrá el servicio "Registro en la APE" asociado. |
| | | CA_HU18_02 | Al asociar un servicio al empleado, este podrá tomar turnos que tengan dicho servicio. |
| | | CA_HU18_03 | El empleado debe tener mínimo 1 servicio asociado. |
| | | CA_HU18_04 | Al desasociar un servicio del empleado, este ya no podrá ver ni tomar los turnos con este servicio. |

| | | | |
|-------|---|------------|--|
| | | CA_HU18_05 | Si cancela el proceso o sale de la pestaña no se guardarán los datos del registro. |
| | | CA_HU18_06 | Se mostrará un modal de éxito o error dependiendo del resultado del proceso. |
| HU_19 | Yo como administrador quiero actualizar la información de los empleados para que su información sea verídica y este al día. | CA_HU19_01 | No se podrá editar el id del empleado. |
| | | CA_HU19_02 | No se podrá editar la contraseña del empleado. |
| | | CA_HU19_03 | El registro tendrá validaciones en tiempo real. |
| | | CA_HU19_04 | Si el registro no cumple con las validaciones no se actualizará el empleado. |
| | | CA_HU19_05 | Si no le da al botón "Editar Empleado" no se actualizará la información del empleado. |
| | | CA_HU19_06 | Si cancela el proceso o sale de la pestaña no se guardarán los datos del registro. |
| | | CA_HU19_07 | Se mostrará un modal de éxito o error dependiendo del resultado del proceso. |
| HU_20 | Yo como administrador necesito ver los detalles del empleado para poder obtener información relevante del mismo. | CA_HU20_01 | El administrador podrá ver el detalle de los empleados. |
| | | CA_HU20_02 | Se mostrará la lista de los servicios a los que está asociado. |
| | | CA_HU20_03 | Se mostrará un modal de éxito o error dependiendo del resultado del proceso. |
| HU_21 | Yo como administrador quiero eliminar un empleado para poder retirar su acceso del sistema. | CA_HU21_01 | Si el empleado tiene una relación previa con un turno realizado, no se podrá eliminar al empleado. |
| | | CA_HU21_02 | El empleado "En espera" no podrá ser eliminado. |
| | | CA_HU21_03 | Se mostrará un modal de confirmación antes de eliminar el empleado. |
| | | CA_HU21_04 | Se mostrará un modal de éxito o error dependiendo del resultado del proceso. |
| HU_22 | Yo como administrador quiero cambiar el estado del usuario para poder limitar su acceso al sistema. | CA_HU22_01 | Se mostrará un modal de confirmación antes de cambiar el estado del empleado. |
| | | CA_HU22_02 | El empleado "En espera" no podrá ser modificado. |
| | | CA_HU22_03 | Solo el administrador podrá cambiar el estado de un empleado. |
| | | CA_HU22_04 | Se mostrará un modal de éxito o error dependiendo del resultado del proceso. |
| HU_23 | Yo como administrador quiero registrar un servicio para poder ofrecerlo al usuario al solicitar el turno. | CA_HU23_01 | No habrá 2 servicios con un mismo id. |
| | | CA_HU23_02 | El código del servicio será de un máximo de 2 letras, y será único. |
| | | CA_HU23_03 | Si no le da al botón "Crear Servicio" no se guardará el registro del servicio. |
| | | CA_HU23_04 | Si el registro no cumple con las validaciones no se creará el servicio. |
| | | CA_HU23_05 | El registro tendrá validaciones en tiempo real. |
| | | CA_HU23_06 | Se mostrará un modal de éxito o error dependiendo del resultado del proceso. |
| HU_24 | | CA_HU24_01 | Se mostrarán todos los servicios registrados en el aplicativo. |

| | | | |
|-------|--|------------|---|
| | Yo como administrador quiero listar los servicios para poder tener conocimiento de cuáles son las opciones ofrecidas a los usuarios. | CA_HU24_02 | El servicio "Registro en la APE" se listará de forma predeterminada. |
| | | CA_HU24_03 | La tabla de servicios se paginará al llegar a un tope determinado. |
| | | CA_HU24_04 | Se mostrará un modal de éxito o error dependiendo del resultado del proceso. |
| HU_25 | Yo como administrador quiero actualizar la información de los servicios para poder mantener su información al día. | CA_HU25_01 | No se podrá editar el id del servicio. |
| | | CA_HU25_02 | El registro tendrá validaciones en tiempo real. |
| | | CA_HU25_03 | El servicio "Registro en la APE" no podrá ser editado. |
| | | CA_HU25_04 | Si el registro no cumple con las validaciones no se actualizará el servicio. |
| | | CA_HU25_05 | Si no le da al botón "Editar Servicio" no se actualizará la información del servicio. |
| | | CA_HU25_06 | Se mostrará un modal de éxito o error dependiendo del resultado del proceso. |
| HU_26 | Yo como administrador quiero ver el detalle de los servicios para poder obtener información sobre el servicio. | CA_HU26_01 | El administrador podrá ver el detalle de los servicios. |
| | | CA_HU26_02 | Se mostrará un modal de éxito o error dependiendo del resultado del proceso. |
| HU_27 | Yo como administrador quiero buscar entre los servicios para poder tener encontrar el que necesito con facilidad. | CA_HU27_01 | Se filtrarán los servicios por todos los campos disponibles. |
| | | CA_HU27_02 | Si el buscador está vacío no se aplicará ningún filtro. |
| | | CA_HU27_03 | Si ningún registro concuerda con el buscador, no se mostrará ningún registro. |
| HU_28 | Yo como administrador quiero eliminar un servicio para poder retirarlo de las opciones posibles y así evitando confusiones. | CA_HU28_01 | Si el servicio tiene una relación previa con un turno, no se podrá eliminar al servicio. |
| | | CA_HU28_02 | El servicio "Registro en la APE" no podrá ser eliminado. |
| | | CA_HU28_03 | Se mostrará un modal de confirmación antes de eliminar el servicio. |
| | | CA_HU28_04 | Se mostrará un modal de éxito o error dependiendo del resultado del proceso. |
| HU_29 | Yo como administrador quiero cambiar el estado del servicio para poder limitar temporalmente los servicios ofrecidos al usuario. | CA_HU29_01 | Solo el administrador podrá cambiar el estado de un servicio. |
| | | CA_HU29_02 | El servicio "Registro en la APE" no podrá cambiar de estado. |
| | | CA_HU29_03 | Se mostrará un modal de confirmación antes de cambiar el estado del servicio. |
| | | CA_HU29_04 | Se mostrará un modal de éxito o error dependiendo del resultado del proceso. |
| HU_30 | Yo como administrador/orientador quiero listar los turnos realizados para poder evaluar el historial de los turnos. | CA_HU30_01 | Te mostrarán todos los turnos finalizados registrados en el aplicativo. |
| | | CA_HU30_02 | La tabla de turnos se paginará al llegar a un tope determinado. |
| | | CA_HU30_03 | Los turnos se listarán por fecha de finalización de forma descendiente (de la más reciente a la más antigua). |
| | | CA_HU30_04 | Se mostrará un modal de error dependiendo del resultado del proceso. |

| | | | |
|--------------|---|-------------------|--|
| HU_31 | Yo como administrador/orientador necesito ver los detalles de los turnos finalizados para poder tener un conocimiento amplio de cada turno. | CA_HU31_01 | El empleado podrá ver los detalles del turno finalizado. |
| | | CA_HU31_02 | Se mostrarán los servicios realizados al usuario. |
| | | CA_HU31_03 | Se mostrará un modal de éxito o error dependiendo del resultado del proceso. |
| HU_32 | Yo como administrador/orientador quiero buscar entre los turnos realizados para poder filtrar los registros necesarios. | CA_HU32_01 | Se filtraran los turnos por todos los campos disponibles. |
| | | CA_HU32_02 | Si el buscador esta vacio no se aplicará ningún filtro. |
| | | CA_HU32_03 | Si ningun registro concuerda con el buscador, no se mostrará ningun registro. |
| HU_33 | Yo como orientador necesito un dashboard para poder examinar mi productividad. | CA_HU33_01 | Se mostrará el número de turnos atendidos en el día. |
| | | CA_HU33_02 | Se seleccionará un rango de fechas para actualizar los datos del dashboard. |
| | | CA_HU33_03 | Se mostrará el tiempo promedio de espera de los turnos asignados al empleado en el rango de fechas. |
| | | CA_HU33_04 | Se mostrará un gráfica del numero de turnos hechos en cada día por el empleado en el rango de fechas seleccionadas. |
| | | CA_HU33_05 | Se mostrará un gráfica de los servicios solicitados en los turnos asignados al empleado en el rango de fechas seleccionadas. |
| | | CA_HU33_06 | Se mostrará una gráfica de los tipos de población de los turnos asignados al empleado en el rango de fechas seleccionadas. |
| | | CA_HU33_07 | Se mostrará el numero de turnos pendientes en el dia. |
| | | CA_HU33_08 | Se mostrará el tiempo promedio de atención (eficiencia) de los turnos asignados al empleado en el rango de fechas. |
| | | CA_HU33_09 | Se mostrará un modal de exito o error dependiendo del resultado del proceso. |
| HU_34 | Yo como administrador quiero un dashboard para poder tener acceso a información relevante acerca del sistema y sus procesos. | CA_HU34_01 | Se mostrará el número de turnos atendidos en el día. |
| | | CA_HU34_02 | Se seleccionará un rango de fechas para actualizar los datos del dashboard. |
| | | CA_HU34_03 | Se mostrará el tiempo promedio de espera de cada empleado en el rango de fechas. |
| | | CA_HU34_04 | Se mostrará un gráfica del numero de turnos hechos en cada día en el rango de fechas seleccionadas. |
| | | CA_HU34_05 | Se mostrará un gráfica de servicios solicitados en el rango de fechas seleccionadas. |
| | | CA_HU34_06 | Se mostrará una gráfica de los tipos de población de los turnos en el rango de fechas seleccionadas. |
| | | CA_HU34_07 | Se mostrará el numero de turnos pendientes en el dia. |

| | | | |
|-------|--|------------|---|
| | | CA_HU34_08 | Se mostrará el tiempo promedio de atención de cada empleado en el rango de fechas. |
| | | CA_HU34_09 | Se mostrará un modal de éxito o error dependiendo del resultado del proceso. |
| HU_35 | Yo como administrador/orientador quiero crear reportes para poder recibir información adicional sobre los procesos del aplicativo. | CA_HU35_01 | Se utilizará el rango de fechas determinados en el dashboard para crear el reporte. |
| | | CA_HU35_02 | El reporte mostrará el rango de fechas que uso el reporte para obtener los datos. |
| | | CA_HU35_03 | El reporte tendrá un formato acorde a la identidad visual del SENA. |
| | | CA_HU35_04 | Se mostrarán los turnos hechos en el rango de fechas seleccionado. |
| | | CA_HU35_05 | Se mostrará un modal de éxito o error dependiendo del resultado del proceso. |
| | | | |
| HU_36 | Yo como administrador/orientador necesito listar los turnos en espera para poder tomarlos y gestionarlos. | CA_HU36_01 | Se mostrarán todos los turnos en espera (sin empleado asignado). |
| | | CA_HU36_02 | Se actualizará la lista de los turnos en espera cada vez que se tome/agregue un turno. |
| | | CA_HU36_03 | La lista de turnos pendientes estará organizada por prioridad. |
| | | CA_HU36_04 | Esta lista se mostrará a todos los empleados. |
| | | CA_HU36_05 | Los turnos visibles varían de los servicios asociados al empleado. |
| | | CA_HU36_06 | Se mostrará un modal de éxito o error dependiendo del resultado del proceso. |
| HU_37 | Yo como administrador/orientador necesito ingresar el módulo que voy a cubrir para poder empezar a tomar turnos. | CA_HU37_01 | El modal se mostrará al ingresar a la vista de Gestionar Turnos. |
| | | CA_HU37_02 | El módulo es obligatorio, por tanto el modal no se podrá cerrar sin haber especificado el módulo. |
| | | CA_HU37_03 | El módulo tendrá un máximo de 2 números, no puede ser nulo. |
| | | CA_HU37_04 | Al cerrar la sesión se reiniciará el número del módulo. |
| | | CA_HU37_05 | Se mostrará un modal de éxito o error dependiendo del resultado del proceso. |
| HU_38 | Yo como administrador/orientador necesito editar el módulo que voy a cubrir para poder cambiar de puesto sin confundir a los usuarios. | CA_HU38_01 | El módulo tendrá un máximo de 2 números, no puede ser nulo. |
| | | CA_HU38_02 | Se mostrará un modal de éxito o error dependiendo del resultado del proceso. |
| HU_39 | Yo como administrador/orientador necesito tomar un turno para poder brindarle la atención requerida. | CA_HU39_01 | Se validará que el turno no haya sido tomado por otro empleado. |
| | | CA_HU39_02 | El empleado debe terminar el turno actual para poder tomar otro turno. |
| | | CA_HU39_03 | Al tomar el turno se actualizará la lista de turnos actuales del televisor, agregándolo. |
| | | CA_HU39_04 | El turno tomado tendrá por defecto el estado de "Espera". |
| | | CA_HU39_05 | Se mostrará un modal de éxito o error dependiendo del resultado del proceso. |

| | | | |
|--------------|---|-------------------|---|
| HU_40 | Yo como administrador/orientador necesito cambiar el estado del turno recién tomado para poder informarle al usuario la fase en la que se encuentra su turno. | CA_HU40_01 | Los estados disponibles serán Espera, Proceso, Cancelado, Atendido. |
| | | CA_HU40_02 | El estado del turno por defecto será Espera. |
| HU_41 | Yo como administrador/orientador necesito ver los detalles del turno para poder informarme acerca de la naturaleza del turno. | CA_HU41_01 | El empleado podrá ver el detalle de los turnos en espera. |
| | | CA_HU41_02 | Se mostrará un modal de éxito o error dependiendo del resultado del proceso. |
| HU_42 | Yo como administrador/orientador necesito priorizar los turnos para poder atender primero a los usuarios más relevantes. | CA_HU42_01 | Cada turno tendrá 2 flechas (una hacia arriba, otra hacia abajo) con las cuales podrá mover el turno en la lista, modificando el orden en el que recibirá los turnos. |
| | | CA_HU42_02 | El orden solo tendrá efecto en la vista del empleado. |
| | | CA_HU42_03 | Se mostrará un modal error dependiendo del resultado del proceso. |
| HU_43 | Yo como administrador/orientador necesito visualizar los datos del turno actual para poder conocer al usuario y su necesidad. | CA_HU43_01 | El empleado al tomar el turno, mostrará un recuadro con información sobre el turno. |
| | | CA_HU43_02 | Si el empleado no está atendiendo a un turno, no se mostrará ninguna información, en cambio se mostrará el mensaje "Tome un turno para empezar." |
| | | CA_HU43_03 | Se mostrará un modal error dependiendo del resultado del proceso. |
| HU_44 | Yo como administrador/orientador necesito finalizar el turno para poder terminar el proceso con el usuario. | CA_HU44_01 | Al finalizar el turno, se le hará llegar al correo del usuario un formulario para calificar el servicio. |
| | | CA_HU44_02 | Al finalizar el turno se abrirá un modal donde el empleado ingresará una observación acerca del servicio y marcará los servicios que se le realizaron al usuario. |
| | | CA_HU44_03 | Al finalizar el turno se actualizará los turnos actuales de la pantalla del televisor, retirándolo. |
| | | CA_HU44_04 | Si el turno es finalizado con el estado "Cancelado", el turno se quedará en la vista del televisor por el resto del día. |
| | | CA_HU44_05 | Se mostrará un modal de éxito o error dependiendo del resultado del proceso. |
| HU_45 | Yo administrador/orientador necesito determinar los servicios realizados al usuario para poder tener un registro completo del turno. | CA_HU45_01 | Por defecto estarán seleccionados los servicios con el que se creó el turno. |
| | | CA_HU45_02 | El turno debe finalizarse con mínimo un servicio. |
| | | CA_HU45_03 | El empleado al darle "Finalizar turno" desplegará el modal donde estará los checkbox de los servicios. |
| | | CA_HU45_04 | Se mostrará un modal de éxito o error dependiendo del resultado del proceso. |

| | | | |
|--------------|--|-------------------|---|
| HU_46 | Yo como administrador/orientador necesito agregar una observacion del servicio al finalizarlo para poder tener un registro del turno y su resultado. | CA_HU46_01 | El empleado al darle "Finalizar turno" desplegará el modal donde estará el campo de Observaciones. |
| | | CA_HU46_02 | El empleado tendrá un select de observaciones predeterminadas para agilizar el diligenciamiento de la observación. |
| | | CA_HU46_03 | El campo es obligatorio, significando así que el turno no se finalizará si el empleado no ingresa una observación. |
| | | CA_HU46_04 | Se mostrará un modal de éxito o error dependiendo del resultado del proceso. |
| HU_47 | Como sistema de inteligencia artificial necesito analizar y priorizar los turnos según el tipo de servicio solicitado para poder asegurar una atención efectiva a la población preferencial. | CA_HU47_01 | Se tendrá en cuenta los servicios disponibles en el aplicativo. |
| | | CA_HU47_02 | El cálculo se hará periódicamente. |
| | | CA_HU47_03 | Se calculará el tiempo promedio que se demoraría cada servicio en realizarse, esto mediante un modelo entrenado con los turnos ya realizados. |

10.2. Historias de Usuario y Criterios de Aceptación

A continuación, se describe el proceso que debe seguir el usuario para desplegar la aplicación Digiturno en un servidor en la nube (Vercel):

1. *Despliegue de la base de datos*

1. Crear una cuenta de supabase

-Ve a <https://supabase.com/>.

-Regístrate o inicia sesión con tu cuenta de GitHub (es el único método soportado actualmente).

2. Crear un nuevo proyecto

En el panel de control de Supabase, haz clic en el botón **"New Project"**.

Completa los siguientes detalles:

- **Project name:** Escribe un nombre para tu proyecto.
- **Database password:** Introduce una contraseña segura para la base de datos.
- **Region:** Elige una región cercana para menor latencia.

Haz clic en **"Create new project"**.

3. Configurar la base de datos PostgreSQL

Una vez que el proyecto esté creado, Supabase configurará automáticamente una instancia de PostgreSQL para ti.

En la pestaña **Database**, encontrarás:

- El **URL** de conexión.
- El **Usuario, Contraseña, Nombre de la base de datos** y otros detalles.
- Presionar el botón **Connect** y copiar URL

Esta URL se utiliza para conectar a una base de datos de Supabase.

Despliegue de Back-end en Render

1. Crear una cuenta de Render

-Ve a [Cloud Application Platform | Render](#).

-Regístrate o inicia sesión con tu cuenta de GitHub (es el único método soportado actualmente).

2. Crear un nuevo proyecto

-ir a Projects y presionar un Create new project

4. Prepara tu proyecto

- **Asegúrate de que tu proyecto está listo para producción:**
 - Incluye un archivo `package.json` con scripts necesarios como `start`.

Asegúrate de que el servidor escuche el puerto proporcionado por Render a través de la variable de entorno `PORT`. Por ejemplo:

```
const PORT = process.env.PORT || 3000;

app.listen(PORT, () => {

  console.log(`Servidor corriendo en el puerto ${PORT}`);

});
```

Instala dependencias necesarias:

```
npm install
```

- **Crea un archivo `.env.example`** con las variables de entorno utilizadas en tu proyecto para referencias futuras.

5. Sube tu proyecto a GitHub

Render utiliza repositorios de GitHub para el despliegue. Asegúrate de:

- **Ignorar carpetas innecesarias** como `node_modules` en `.gitignore`.
- Subir tu código a un repositorio remoto en GitHub.

6. Crea un nuevo servicio en Render

Ve a **Dashboard** en Render y selecciona **New + > Web Service**.

Elige tu repositorio de GitHub que contiene tu proyecto Node.js.

Configura el servicio:

- **Nombre:** Dale un nombre a tu aplicación.
- **Branch:** Selecciona la rama que deseas desplegar (generalmente `main` o `master`).
- **Build Command:** Usa `npm install` (Render instalará automáticamente las dependencias).

Start Command: Asegúrate de que sea el comando para iniciar tu servidor, como:

`npm start`

-
- **Runtime:** Selecciona "Node.js".

6. Configura las variables de entorno

En la sección de configuración del servicio:

- Agrega las variables de entorno necesarias para tu aplicación desde el archivo `.env` (por ejemplo, claves API, URIs de base de datos, etc.).

7. Configura el despliegue continuo

Cualquier cambio en la rama configurada (por ejemplo, `main`) se reconstruirá y desplegará automáticamente.

Despliegue de front-end en Vercel

1. Crear una cuenta de Vercel

-Ve a [Vercel](#)

-Regístrate o inicia sesión con tu cuenta de GitHub (es el único método soportado actualmente).

2. Preparar tu Proyecto de Vite

Antes de subirlo, asegúrate de que el proyecto esté listo para producción:

Instala las dependencias:

`npm install`

Construye el proyecto:

`npm run build`

-Esto generará una carpeta `dist` con los archivos listos para producción.

3. Subir el Proyecto a Vercel

-Accede al Dashboard de Vercel: Inicia sesión en tu cuenta de Vercel y ve al panel de control.

-Crear un Nuevo Proyecto:

- Haz clic en el botón **"Add New..."** y selecciona **"Project"**.

- En lugar de conectar un repositorio, selecciona **"Import..."** y luego elige **"Manual Upload"**.

-Configurar el Proyecto:

- Sube el contenido de tu carpeta **dist** directamente (puedes arrastrar y soltar los archivos).
- Configura los ajustes básicos:
 - **Framework Preset:** Selecciona **Vite**.
 - **Output Directory:** Asegúrate de que esté configurado como **dist**.

4. Configurar vite.config.js

En tu archivo **vite.config.js**, asegúrate de configurar correctamente la base y otras opciones para que los recursos se sirvan adecuadamente:

```
import { defineConfig } from 'vite';

export default defineConfig({

  base: './', // Esto asegura que las rutas sean relativas

  build: {

    outDir: 'dist', // Asegúrate de que la carpeta de salida sea "dist"

  },

});
```

-Instalar Dependencias

Ejecuta el siguiente comando en la raíz del proyecto para asegurarte de que las dependencias estén instaladas:

```
npm install -g vercel
```

-Inicia Sesión en Vercel desde la CLI:

```
vercel login
```

Sube el Proyecto a Vercel: Desde la raíz de tu proyecto, ejecuta:

```
vercel
```

Durante el proceso:

- Si Vercel detecta que el proyecto usa Vite.js, aplicará automáticamente las configuraciones necesarias.

- Especifica la carpeta de salida (**dist**) como el directorio de despliegue cuando te lo pida.