

**Manual Técnico de la Herramienta para la Gestión de Proyectos de Formación  
(OPTIMA)**

**Proyecto:**  
**Optimización de la Gestión de Proyectos de Formación en la Universidad Visión  
de las Américas (OPTIMA)**

**Autores:**  
**Santiago Sepúlveda Montoya**  
**Santiago Adolfo Yepes Zuleta**  
**Julián Herrera Leal**

**Universidad Visión de las Américas**

**Universidad de Medellín**  
**Facultad de Ingeniería**  
**Programa de Ingeniería de Sistemas**

**Medellín**

**2024**

Este manual tiene como objetivo proporcionar una guía técnica completa del proyecto, detallando su arquitectura, despliegue, herramientas utilizadas y los procedimientos para generar reportes estadísticos. A continuación, se describen los aspectos clave del sistema.

## Arquitectura del Proyecto

El proyecto sigue el patrón **MVC (Modelo-Vista-Controlador)**, lo que permite una separación clara de responsabilidades y facilita el mantenimiento y escalabilidad.

### 1. Frontend:

- Desarrollado utilizando **NEXT**, un framework basado en componentes que facilita la creación de interfaces de usuario dinámicas y reutilizables. La arquitectura de componentes garantiza una organización modular, permitiendo que cada funcionalidad esté encapsulada y sea independiente.
- Se utilizó la librería **Tailwind CSS** para el diseño y estilización del frontend, permitiendo una creación rápida y eficiente de interfaces personalizadas mediante clases predefinidas.
- Las dependencias fueron manejadas mediante **npm**, herramienta estándar para la gestión de paquetes en el ecosistema de Node.js.
- El entorno frontend está desplegado en **VERCEL**, una plataforma que permite un despliegue continuo y sin complicaciones.

### 2. Backend:

- Implementado en **Node.js**, con una arquitectura **hexagonal**, que promueve la separación entre la lógica de negocio y los elementos externos como bases de datos, APIs o frameworks. Esta arquitectura permite que los casos de uso principales no dependan directamente de tecnologías específicas, facilitando cambios futuros y el mantenimiento.
- Al igual que el frontend, las dependencias del backend fueron gestionadas con **npm**.
- Desplegado en **RENDER**, una plataforma que asegura una gestión eficiente del servidor y servicios backend.

## Funcionamiento e Integración

- El enlace principal del proyecto está alojado en **VERCEL**. Cuando un usuario accede al sistema, el backend, desplegado en **RENDER**, se activa automáticamente. Este proceso inicial de carga y conexión a la base de datos puede tomar entre **3 a 5 minutos**, dependiendo de la cantidad de datos almacenados y la carga del servidor.
- Una vez completada la carga, el sistema está listo para operar y permite gestionar nueva información, almacenar datos, y realizar consultas en tiempo real.

## Gestión de Reportes Estadísticos

El sistema incorpora la generación de reportes estadísticos a través de **Power BI Service**, que se conecta a la base de datos alojada en **MongoDB Atlas** mediante un **ODBC**.

- Para garantizar la correcta actualización de los datos en Power BI, es necesario instalar una **puerta de enlace** en un equipo local. Este componente permite la comunicación entre Power BI Service y la base de datos.
- La configuración e instalación de la puerta de enlace está detallada en el manual de usuario.
- Los datos pueden actualizarse hasta **8 veces al día**, con un intervalo aproximado de **12 a 15 minutos** por actualización.

## Tecnologías y Herramientas Utilizadas

### Frontend

- **Next**: Permite desarrollar componentes dinámicos y mejorar la experiencia del usuario.
- **Tailwind CSS**: Usado para estilizar las interfaces de usuario con un enfoque en la personalización y la productividad.
- **npm**: Herramienta empleada para gestionar librerías y dependencias.
- **VERCEL**: Proporciona un entorno de despliegue continuo para el frontend.

### Backend

- **Node.js(V18)**: Utilizado para la lógica de negocio y la creación de servicios API.

- **npm:** Facilitó la instalación y manejo de librerías como Express, Mongoose, entre otras.
- **RENDER:** Asegura el despliegue confiable del backend y su integración con el frontend.

### Base de Datos

- **MongoDB Atlas:** Base de datos NoSQL en la nube que permite un acceso rápido y escalable a los datos. En su versión M0 (versión que se usa en el proyecto) sólo permite un almacenamiento máximo de 512MB.

### Reportes

- **Power BI Service:** Herramienta empleada para la creación y visualización de reportes dinámicos.

### Despliegue y Configuración

- Todo el sistema está diseñado para operar en capas gratuitas de las herramientas utilizadas, optimizando costos sin comprometer funcionalidad.
- Es fundamental que la **puerta de enlace** esté correctamente configurada para permitir la sincronización y actualización de datos entre Power BI y MongoDB Atlas.
- La integración entre el frontend y backend está diseñada para ser fluida y permite una experiencia de usuario eficiente y robusta.

### Consideraciones Finales

Este manual proporciona una visión integral de los aspectos técnicos del sistema, asegurando que cualquier miembro del equipo técnico pueda comprender su diseño, funcionamiento e integración. La utilización de herramientas modernas como **React**, **Tailwind CSS**, y **Node.js**, junto con plataformas de despliegue eficientes como **VERCEL** y **RENDER**, garantiza una solución robusta, escalable y fácil de mantener. El acceso al código fuente se encuentra en dos repositorios de los cuales se les hará entrega.