Documentación Técnica — Proyecto GestPass

1. Introducción y Propósito

GestPass es una aplicación web desarrollada con tecnologías modernas que permite a los usuarios **gestionar sus contraseñas de forma segura, intuitiva y accesible**. El proyecto surge para responder a la necesidad de proteger credenciales digitales sin requerir conocimientos técnicos avanzados, enfocándose en la **modularidad, la seguridad y la experiencia de usuario**.

El objetivo general es desarrollar una aplicación web que permita almacenar, gestionar y cifrar contraseñas de forma **segura, accesible y eficiente**.

2. Objetivos y Características Implementadas

2.1 Objetivos Específicos Cumplidos

- Implementar un sistema de autenticación con **NextAuth y JWT**.
- Integrar **cifrado de contraseñas con bcrypt**.
- Diseñar una interfaz responsiva y accesible con **Radix UI**.
- Incorporar recuperación de contraseñas vía correo electrónico con **Resend**.
- Aplicar buenas prácticas de **modularización y arquitectura MVC**.

2.2 Características Principales

1. **Gestión de Contraseñas:** Almacenamiento y administración segura de credenciales.
2. **Cifrado:** Encriptación de contraseñas con **bcrypt**.
3. **Generador de Claves:** Creación de contraseñas robustas con caracteres especiales.
4. **Autenticación:** Inicio de sesión con **NextAuth** y proveedor de credenciales.
5. **Recuperación de Clave:** Envío de correos de recuperación mediante **Resend**.

3. Arquitectura del Sistema y Despliegue

3.1 Tecnologías Clave

Tecnología	Propósito
Next.js	Framework para aplicaciones web modernas.
React	Biblioteca para interfaces interactivas.
Prisma	ORM para gestión de base de datos.
MongoDB	Base de datos NoSQL para almacenamiento.
bcrypt	Encriptación y comparación de contraseñas.
NextAuth / JWT	Autenticación y manejo de sesiones.
Resend	Envío de correos de recuperación.
Radix UI / Sonner	Componentes accesibles y estilizados / Notificaciones visuales.

3.2 Arquitectura y Estructura

El proyecto sigue el patrón **MVC**, asegurando una separación clara entre lógica, presentación y datos.

- **Estructura de Carpetas:** Organizada para facilitar la mantenibilidad.
  - app/: Rutas y componentes principales.
  - components/: Elementos reutilizables de la interfaz.
  - prisma/: Configuración del cliente Prisma.
  - lib/: Funciones auxiliares (ej. conexión a MongoDB).
  - next-auth/: Configuración de autenticación.
- **Ejecución:** Para iniciar el entorno de desarrollo local, se utiliza npm run dev (o yarn dev/pnpm dev).

3.3 Entorno de Despliegue (Logística de Entrada y Salida)

El entorno de despliegue y aprovisionamiento está basado en servicios **gratuitos** y escalables, reflejando un modelo de **bajo costo operacional** (Aprovisionamiento y Logística de Entrada).

• **Despliegue y DNS:** Se utiliza **Vercel** para el despliegue de la aplicación y **Cloudflare** para la gestión de DNS. La **Logística de Salida** se concreta con el despliegue del producto en el dominio personalizado (https://gestpass.mtsprz.org/).

• **Base de Datos:** Se utiliza **MongoDB Atlas** como base de datos NoSQL.

• **Control de Versiones: GitHub** se utiliza para el control de versiones y como repositorio del código fuente.

• **Servicios Externos:** El envío de correos se gestiona mediante **Resend**.

4. Diseño de Base de Datos (DB Schema)

El diseño de la base de datos, gestionada con Prisma sobre MongoDB, se compone de tres modelos principales que establecen las relaciones de la información crítica.

4.1 Modelo User (Usuario)

Representa a los usuarios del sistema:

Campo	Tipo	Notas Clave
id	String (ObjectId)	Identificador único (@id @default(auto())).
username	String?	Debe ser único (@unique).
email	String?	Debe ser único (@unique).
hashedPassword	String?	Almacena la clave cifrada.
elements	Element[]	Relación con las credenciales almacenadas.
passwordResetTokens	PasswordResetToken[]	Relación con los tokens de recuperación.

4.2 Modelo Element (Elemento/Credencial)

Representa cada credencial o elemento seguro almacenado por el usuario:

Campo	Tipo	Notas Clave
id	String (ObjectId)	Identificador único.
typeElement	String	Clasifica el elemento (e.g., inicio de sesión, Tarjeta, Identidad).
userId	String (ObjectId)	Clave foránea que lo relaciona con el User.

user	User?	Relación donde los elementos son dependientes del usuario (onDelete: Cascade).
password	String?	Almacena la contraseña cifrada.

4.3 Modelo PasswordResetToken (Token de Recuperación)

Se utiliza para el proceso de recuperación de clave segura:

Campo	Tipo	Notas Clave
id	String (ObjectId)	Identificador único.
token	String	El token de un solo uso para la recuperación (@unique).
userId	String (ObjectId)	Clave foránea que lo relaciona con el User.
expiresAt	DateTime	Indica cuándo expira el token.

5. Metodología de Desarrollo y Gestión (Operaciones)

La gestión del proyecto se realiza bajo la metodología **Scrum** con **iteraciones semanales**.

• **Planificación y Seguimiento:** Se utiliza **GitHub Projects** para gestionar las tareas y realizar la planificación y seguimiento. Las tareas se organizan en las columnas: “TODO”, “En proceso” y “DONE”.

• **Mejora Continua:** Se prioriza la **modularización, la seguridad y la trazabilidad** en las **Operaciones**. La capacidad para investigar e incorporar modificaciones que permitan la mejora continua es una actividad clave del **Desarrollo Tecnológico**.

• **Recursos Humanos:** Al ser un proyecto individual, la **Gestión de R.R.H.H.** se centra en que el desarrollador organiza sus tiempos dedicando 1 hora cada 2 días en etapa de producción.

6. Seguridad y Servicios (Posventa)

La seguridad es gestionada en múltiples capas (según la Cadena de Valor y el Informe Técnico).

• **Encriptación de Claves:** Se utiliza **bcrypt** para la encriptación y manejo de contraseñas.

• **Autenticación y Sesiones:** Manejo de sesiones con **JWT** y validación de credenciales con **NextAuth**.

• **Servicios de Posventa:** Para aumentar la confianza del consumidor:

◦ **Recuperación de Contraseña:** El sistema incluye recuperación de contraseña vía **Resend**, lo cual está operativo.

◦ **Reporte de Errores:** Se incluye una sección para reportar errores, cuyos mensajes llegan directamente al correo del desarrollador.

7. Estado Actual y Próximos Pasos

El proyecto cuenta con las funcionalidades principales implementadas, autenticación y cifrado activos, y una interfaz responsiva.

7.1 Estado de los Pasos Anteriores (Actualización de Avance)

Tarea (Originalmente pendiente)	Estado Actual (Según Avance)
Redacción de documentación para el usuario final	**Completado.** El Manual de Usuario/Soporte Técnico ha sido creado y es parte de las evidencias de servicio.
Optimización SEO y estrategia de difusión	**En Planificación.** Se contempla la difusión futura en redes sociales (Facebook e Instagram) y la optimización SEO, la cual es una actividad estratégica de **MKT y Ventas**.

7.2 Próximos Pasos Prioritarios

- **Incorporación de agente de ayuda para *onboarding***: Esta funcionalidad se contempla en futuras iteraciones.
- Refactorización visual y mejoras en accesibilidad.