# 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. Propuesta de valor

GestPass se centra en ofrecer a los usuarios una solución segura, accesible y gratuita para la gestión de contraseñas, diferenciándose de otros gestores tradicionales por su enfoque modular y su integración con servicios modernos de autenticación y recuperación. El proyecto busca responder a una problemática transversal: la dificultad que enfrentan usuarios sin conocimientos técnicos para proteger sus credenciales digitales en un entorno cada vez más expuesto a riesgos de ciberseguridad.

El valor diferencial de GestPass se expresa en los siguientes aspectos:

- Seguridad avanzada: Las contraseñas se almacenan cifradas mediante bcrypt, y las sesiones se gestionan con JWT y NextAuth, garantizando un control seguro y confiable de los accesos.

- Accesibilidad y simplicidad: La interfaz, desarrollada con Radix UI, es intuitiva y responsiva, permitiendo que cualquier usuario pueda gestionar sus credenciales sin necesidad de conocimientos técnicos especializados.

- Recuperación confiable: El sistema integra Resend para el envío de correos electrónicos de recuperación, asegurando que los usuarios puedan restablecer sus contraseñas de manera rápida y segura.

- Generación de contraseñas robustas: Incluye un generador de claves que fomenta buenas prácticas de seguridad, creando contraseñas complejas y resistentes a ataques comunes.

- Escalabilidad y sostenibilidad: Al estar desplegado en Vercel con gestión de DNS en Cloudflare, el proyecto asegura disponibilidad, bajo costo operativo y capacidad de crecimiento futuro.

En síntesis, la propuesta de valor de GestPass radica en combinar seguridad, usabilidad y confianza en una herramienta que democratiza el acceso a la protección digital. Su diseño modular y su enfoque en la mejora continua lo convierten en una solución innovadora que aporta beneficios directos a estudiantes, profesionales independientes y usuarios cotidianos, fortaleciendo su seguridad en el entorno digital.

## 4. Resultados Esperados

GestPass busca generar un impacto tangible en la seguridad digital de los usuarios, así como en la validación académica y técnica del proyecto. Los resultados esperados se alinean con los objetivos planteados y con la propuesta de valor, asegurando que la aplicación cumpla con su propósito de ser una herramienta segura, accesible y confiable para la gestión de contraseñas.

Entre los principales resultados se destacan:

- Difusión efectiva del proyecto: Se espera que GestPass sea presentado en entornos académicos y difundido en plataformas digitales, logrando visibilidad inicial y posicionamiento como una solución innovadora en el ámbito de la seguridad digital.

- Adopción por parte de usuarios reales: La aplicación debe ser utilizada por estudiantes, profesionales independientes y usuarios cotidianos, quienes validarán su utilidad en la gestión de credenciales.

- Retroalimentación significativa: A través de formularios integrados y correos electrónicos de soporte, se espera obtener comentarios de los usuarios que permitan identificar fortalezas y áreas de mejora, fomentando la evolución continua del sistema.

- Validación de la propuesta de valor: El proyecto debe demostrar que combina seguridad, simplicidad y accesibilidad, diferenciándose de gestores tradicionales y aportando beneficios concretos al público objetivo.

- Mejora continua del sistema: Los resultados esperados incluyen la incorporación de ajustes derivados del feedback, optimización de la interfaz y refactorización visual para aumentar la accesibilidad y la confianza del usuario.

- Impacto académico y profesional: Se espera que GestPass sea reconocido como un proyecto de especialidad que integra buenas prácticas de desarrollo, gestión del cambio y socialización, aportando evidencia técnica y metodológica para futuras iniciativas.

En síntesis, los resultados esperados buscan no solo validar la funcionalidad técnica de GestPass, sino también consolidar su relevancia como herramienta práctica y académica, capaz de contribuir a la seguridad digital y a la formación profesional en el área de desarrollo web.

## 5. Arquitectura del Sistema y Despliegue

### 5.1 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. |

## 5.2 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).

## 5.3 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**.

## 6. 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.

### 6.1 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. |

### 6.2 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. |

### 6.3 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. |

## 7. 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.

## 8. 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.

## 9. Estado Actual y Próximos Pasos

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

### 9.1 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**. |

### 9.2 Próximos Pasos Prioritarios

• **Incorporación de agente de ayuda para *onboarding***: Esta funcionalidad se contempla en futuras iteraciones.

• Refactorización visual y mejoras en accesibilidad.

## 10. Conclusión

**GestPass** representa una propuesta innovadora en el ámbito de la seguridad digital, concebida para responder a la creciente necesidad de proteger credenciales en un entorno donde los riesgos cibernéticos son cada vez más frecuentes y sofisticados. A lo largo del desarrollo se ha demostrado que es posible integrar tecnologías modernas como **Next.js**, **React**, **Prisma** y **MongoDB**, junto con servicios especializados como **NextAuth**, **JWT**, **bcrypt** y **Resend**, para construir una solución modular, segura y accesible. Esta combinación de herramientas garantiza que los usuarios puedan

gestionar sus contraseñas de manera confiable, sin requerir conocimientos técnicos avanzados, lo que amplía el alcance y la utilidad del sistema.

La propuesta de valor de GestPass se centra en ofrecer seguridad, simplicidad y confianza, diferenciándose de gestores tradicionales por su enfoque académico y su despliegue en entornos escalables de bajo costo. El proyecto no solo cumple con los objetivos técnicos planteados, sino que también aporta evidencia metodológica en la aplicación de buenas prácticas de desarrollo, modularización y gestión del cambio. Asimismo, la incorporación de un plan de socialización asegura que el producto pueda ser difundido, validado y mejorado a partir de la retroalimentación de los usuarios, consolidando su relevancia tanto en el ámbito académico como en el profesional.

Los resultados esperados incluyen la adopción inicial por parte de usuarios reales, la validación de la propuesta de valor y la mejora continua del sistema. Estos hitos permitirán que GestPass evolucione hacia una herramienta más robusta y confiable, capaz de contribuir de manera significativa a la seguridad digital de estudiantes, profesionales independientes y comunidades en general. En conclusión, GestPass no solo es un proyecto técnico exitoso, sino también una iniciativa con impacto real, que refleja el compromiso con la innovación, la excelencia y la mejora continua en el desarrollo de soluciones digitales.

## 11. Bibliografía


- Vercel. (2025). Next.js Documentation. Recuperado de https://nextjs.org/docs
- Prisma. (2025). Prisma Documentation. Prisma.io. Recuperado de https://www.prisma.io/docs
- MongoDB Inc. (2025). MongoDB Documentation. Recuperado de https://www.mongodb.com/docs
- NextAuth.js. (2025). NextAuth.js Documentation. Recuperado de https://next-auth.js.org/
- npm. (2025). bcrypt — Node.js library for password hashing. Recuperado de https://www.npmjs.com/package/bcrypt
- Resend. (2025). Resend Email API Documentation. Recuperado de https://resend.com/features/email-api