Gestor de contraseñas

**Datos institucionales**

Localidad: Rojas, provincia de Buenos Aires

Escuela: EESTN°1

Año: 7° 2°

**Participantes**

Rossini, Matias

**Profesores asignados**

Ravagnan, Claudio

De angelis, Andrea

*Ricci, Sandra*

**FECHA: 26/11/2023**

**Indice**

[Descripción 3](#__RefHeading___Toc1814_3107868080)

[Fundamento 3](#__RefHeading___Toc204_3155032530)

[Fundamentos curriculares 4](#__RefHeading___Toc206_3155032530)

[Introducción 5](#__RefHeading___Toc210_3155032530)

[Tecnologías utilizadas 6](#__RefHeading___Toc208_3155032530)

[Funcionamiento 8](#__RefHeading___Toc418_3155032530)

[Procesos generales 8](#__RefHeading___Toc420_3155032530)

[Procesos específicos 8](#__RefHeading___Toc336_251533259)

[Almacenamiento 8](#__RefHeading___Toc338_251533259)

[Navegación 9](#__RefHeading___Toc340_251533259)

[Flujo de trabajo 12](#__RefHeading___Toc1816_3107868080)

[Herramientas 12](#__RefHeading___Toc1818_3107868080)

[Plantilla 12](#__RefHeading___Toc1820_3107868080)

[Nomenclaturas 12](#__RefHeading___Toc1822_3107868080)

[Tamaños BD 12](#__RefHeading___Toc1822_31078680801)

[Diagramas: 13](#__RefHeading___Toc1826_3107868080)

# **Descripción**

Aplicación web de fácil acceso para el almacenamiento seguro de contraseñas; una vez que el usuario se registre podrá comenzar a almacenar sus contraseñas.

El usuario podrá cambiar el aspecto de su UI (Mediante un sistema de Temas), crear grupos para almacenar las contraseñas, configurar para que tipo de servicio es la contraseña (Con un nombre personalizable y un icono a elegir desde la base de datos del sistema), editar, copiar y/o borrar las contraseñas guardadas, así como su nombre de usuario, correo y contraseña.

Las contraseñas de cada usuario serán almacenadas en la base de datos del sistema siendo antes pasadas por la función de encriptación del framework para que se almacene de forma segura en la base de datos.

## Fundamento

La idea nace de la experiencia propia, la forma anterior que tenia de utilizar las contraseñas era muy insegura ya que repetía contraseñas ya existentes en otros sistemas con tal de no tener que recordar nuevas.

Con este sistema ese problema se resuelve ya que estas quedan protegidas solo a las manos del usuario y cuando el las quiera recuperar.

# Fundamentos curriculares

* Gracias al proyecto se lograron desarrollar los contenidos y temas:
  + Uso y modificación de Frameworks de trabajo web:
    - Utilización de Laravel, Alpine Js y Tailwind Css, para la construcción, el funcionamiento y el estilo visual del proyecto.
  + Seguridad informática:
    - Almacenaje de datos de forma encriptada para que su recuperación no sea posible sin cumplir los requisitos necesarios.
  + Gestión segura de datos.
    - Encriptación y Desencriptación de datos, oculto del usuario para que no pueda alterar el proceso.
  + Diseño y aplicación de diagramas:
    - Creación y corrección de diagramas utilizados para explicar el funcionamiento del sistema.
  + Diseño de paginas web:
    - Edición de estilos de la pagina web, mejorando su apariencia mediante la estructuración de elementos y colores utilizados.
  + Investigación sobre tecnologías:
    - Aprendizaje sobre tecnologías aplicadas en el sistema así como sistemas similares al creado, para un mejor desarrollo del mismo, teniendo en cuenta lo que posee uno profesional.

# Introducción

Tras medio año de aprendizaje sobre el funcionamiento y creación de sistemas web para uso cotidiano y profesional, se busco desarrollar un proyecto que englobara todo lo aprendido y más, para llevar a cabo esto se comenzó a aprender y utilizar los Frameworks de libre uso “***Laravel***” y “***Tailwindcss***”

Cada proceso del proyecto se desarrollo de forma individual y independiente, con la ayuda de los profesores a la hora de idear funciones técnicas y visuales del sistema.

El gestor de contraseñas creado, ***“KeyLock”***, cuenta, en su estado actual, con las funcionalidades esenciales para el uso del mismo de forma personal sin ningún impedimento para el usuario, permitiéndole crear un usuario y comenzar a almacenar sus contraseñas de forma segura.

Se espera completar en cuestión de semanas tras la presentación inicial, implementando las funciones secundarias que fueron demoradas para la presentación actual.

A continuación se presentaran y explicaran las tecnologías utilizadas para el funcionamiento del mismo, así como la representación grafica del funcionamiento del sistema:

# Tecnologías utilizadas

* **Laravel 10:**
  + Framework back-end para la creación, funcionamiento y configuración del proyecto, utiliza el lenguaje de programación PHP.
  + Gracias a su funcionamiento, no es necesario crear la estructura de los directorios, ya que el propio Framework genera la suya al crearse.
  + Este facilita el manejo de las rutas en el sistema, la comprobación de usuario, la implementación de elementos externos, la creación de funciones y el almacenamiento en la base de datos.
* **Tailwind 3:**
  + Framework Css para la implementación de estilos y tema oscuro.
  + Con el se implemento la plantilla utilizada en todo el sistema, la cual le dio el estilo actual.
* **Alpine Js 2:**
  + Framework front-end JS para la implementación de estilos y funcionamiento de efectos visuales.
  + Con el se pueden modificar los aspectos visuales de los elementos sin mucho codigo, para que desaparezcan tras cierto tiempo, cambien sus valores o devuelvan un valor indicado.
  + En el proyecto se utiliza para la presentación de mensajes de éxito, error o devolver la contraseña que el usuario intenta recuperar.
* **Cipboard.js:**
  + Librería para el copiado de datos al portapapeles del dispositivo.
  + Con el se logro el devolverle la contraseña desencriptada al usuario guardandola en el portapapeles para que la pegue donde la necesite.
* **Heroicons:**
  + Lista iconos para la decoración visual del sistema.
  + Con el se implementaron los iconos que posee el sistema, en los botones, menús y formularios.
* **Visual Studio Code 1.84:**
  + IDE y editor de código fuente de libre uso.
  + Se utilizo para la escritura del código del proyecto, así como la gestión visual de los archivos.
* **Git**
  + Software de controlador de versiones.
  + Permite crear ramificaciones del proyecto, se utilizo para que no haya conflictos de versiones en el código.
* **GitHub**
  + Plataforma web para almacenar códigos y sus ramificaciones de git.
  + Se utilizo para almacenar de forma web el proyecto y su publicación.
* **MySQL**
  + Gestor de bases de datos.
  + Se utilizo para almacenar y controlar la base de datos del sistema.
* **Plantilla**
  + windmill-dashboard – (<https://github.com/estevanmaito/windmill-dashboard>).

# Funcionamiento

El sistema funciona realizando procesos ya definidos por código que van desde la creación de un usuario hasta el almacenado de una contraseña y su desencriptación para uso cotidiano.

## Procesos generales

El sistema trabaja con procesos que recolectan información que luego manejan internamente para posteriormente mostrarle en pantalla el resultado al usuario

**Img 1.1 – Diagrama de procesos**

En el diagrama (**Img 1.1**) se puede ver de forma simple los procesos principales y los datos que toman.

## Procesos específicos

Los procesos específicos son aquellos mediante los cuales los generales funcionan, en el proyecto son las funciones representadas por el controlador de paginas de Laravel (**Img 1.2**)

Aqui se pueden observar las funciones que ejecuta el sistema en la sección de contraseñas, siendo las predominantes en el sistema las funciones:

* Create = Crear registro
* Store = Almacenar registro
* Edit = Editar registro
* Update = Actualizar registro
* Destroy = Destruir/Borrar registro

Y en el caso de contraseñas tiene uno propio

* Decrypt = Desencriptar

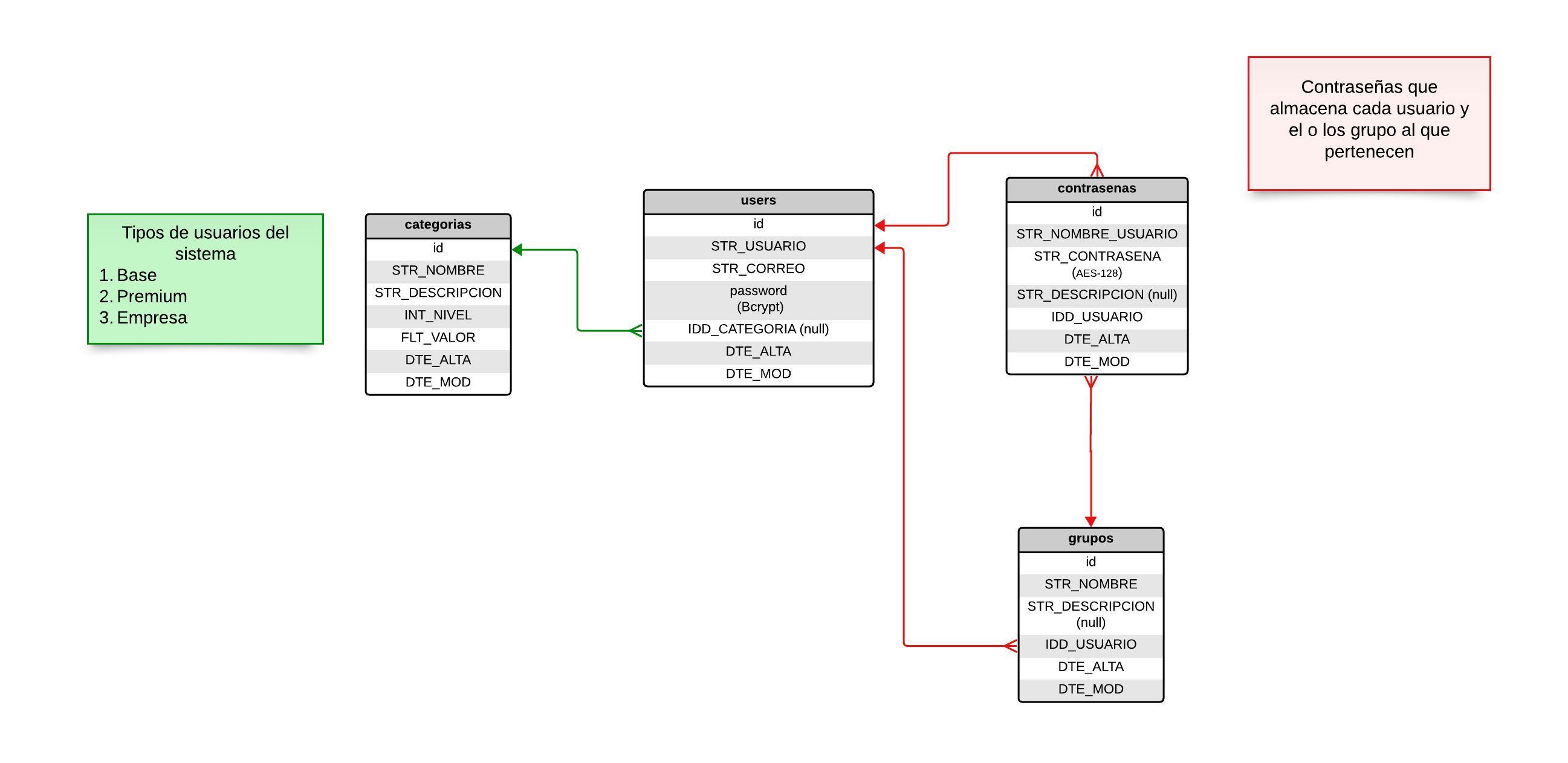
**Img 1.2 – Código controlador vistas**

## Almacenamiento

Todos estos procesos que realiza el sistema son almacenados en la base de datos designada (**Img 1.3**), representada mediante el diagrama de entidad relación

Podemos resaltar datos importantes como, el nombre de cada campo que se almacena, los nombres de las tablas utilizadas y sus conexiones.

La tabla “users” es la principal y es la que almacena los datos del usuario, como se demuestra por sus campos, a esta misma están conectadas el resto de tablas principales, ya que cada una pertenece de alguna manera al usuario; que contraseña creo, que grupo, a que categoría pertenece al usuario.



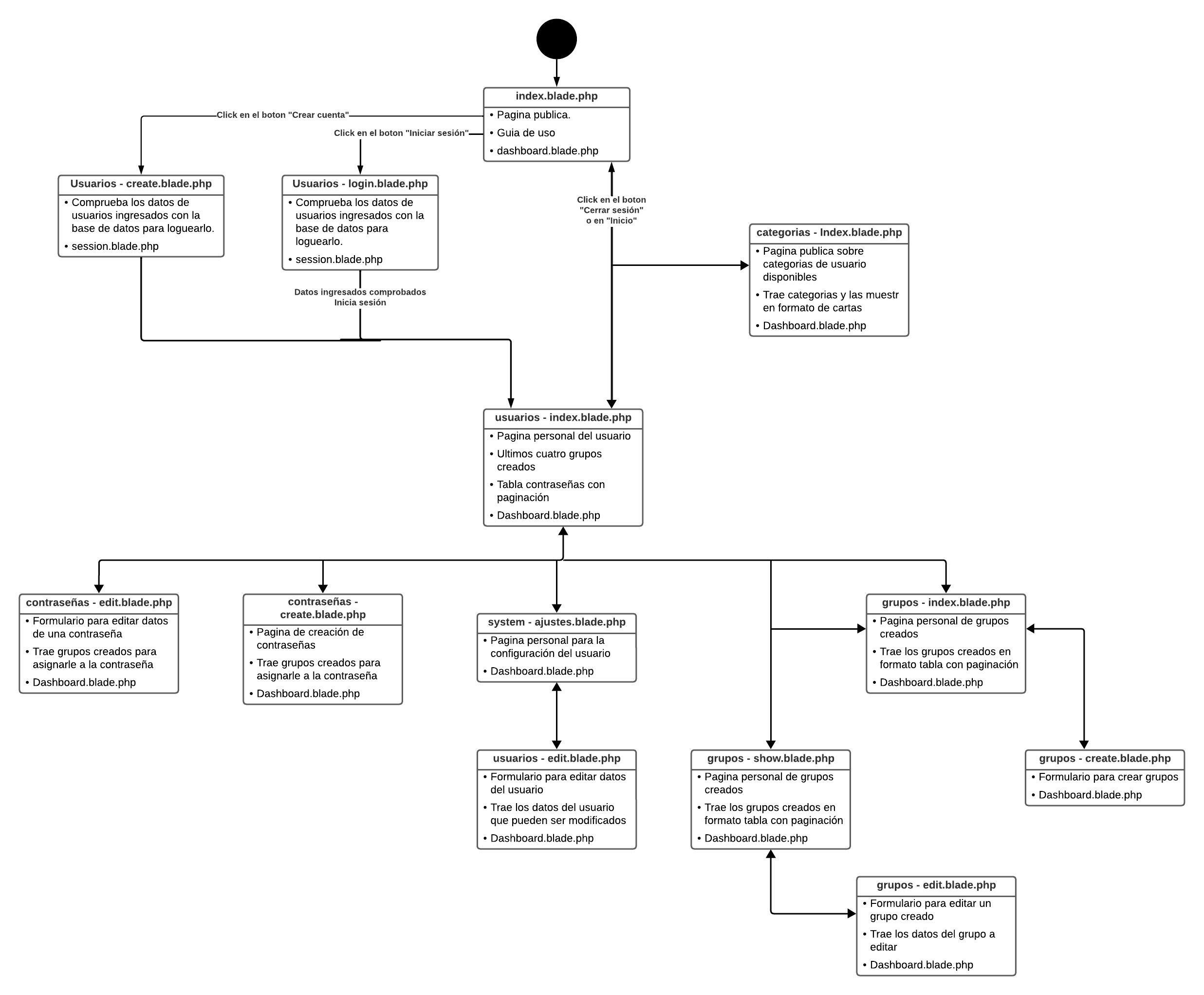
**Img 1.3 – Diagrama entidad relación**

Por su parte las tablas “contrasenas” y “grupos” llevan una conexión entre si, ya que una contraseña puede pertenecer a un grupo y un grupo puede tener varias contraseñas.

### Navegación

Para moverse por la pagina el usuario primero debera tener una sesión iniciada, sea mediante una cuenta ya existente o al crear una, como se muestra (**Img 1.4**).

A excepción de la pantalla inicial y la que contiene la información sobre los “Premiums”, el resto de paginas serán inaccesibles a menos que el usuario tenga una cuenta creada y la este utilizando.



**Img 1.4 – Mapa de navegación**

# Flujo de trabajo

## Herramientas

* **Frameworks:**
  + *Laravel 9.52.15 - (Framework back-end)*
  + *Tailwind 3.3.3 - (Framework front-end)*
  + *Alpine Js 2.x.x – (Framework JavaScript)*
* **Programas:**
  + *Visual Studio Code 1.8.1 - (Programa para la escritura de código)*
  + *Composer 2.5.8 - (Sistema de gestión de paquetes PHP)*
* **Lenguajes:**
  + *Html 5 - (Lenguaje de etiquetas)*
  + *Php 8.2.4 - (Lenguaje de programación conector con BD)*
  + Css 3 - (Lenguaje de diseño grafico)
  + *JavaScript ES13 - (Lenguaje de programación)*
* **Librerias:**
  + clipboard.js – *(Librería JavaScript)*

## Plantilla

* ***Mosaic Lite Laravel***
  + *Repositorio: (*[*https://github.com/cruip/laravel-tailwindcss-admin-dashboard-template*](https://github.com/cruip/laravel-tailwindcss-admin-dashboard-template)*)*

## Nomenclaturas

* General (Todo mayúsculas):
  + “\_” - *Separador.*
  + IDD – *IDs (id).*
  + INT – *Entero (int).*
  + FLT – *Decimales (float).*
  + STR – *Texto (varchar – text).*
  + TME – *Tiempo (time).*
  + DTE – *Fechas (date).*
  + DTC – *Fecha y hora completa ().*
  + BOO – *Buleano/Lógico (tinyint).*
  + LNK – *Link externo.*
  + IDJ – *Variable JavaScript.*
  + Ejemplo: IDD\_LOREM\_IPSUM
* Funciones (\_ ; LowerCamel):
  + xY – *Funciones.*

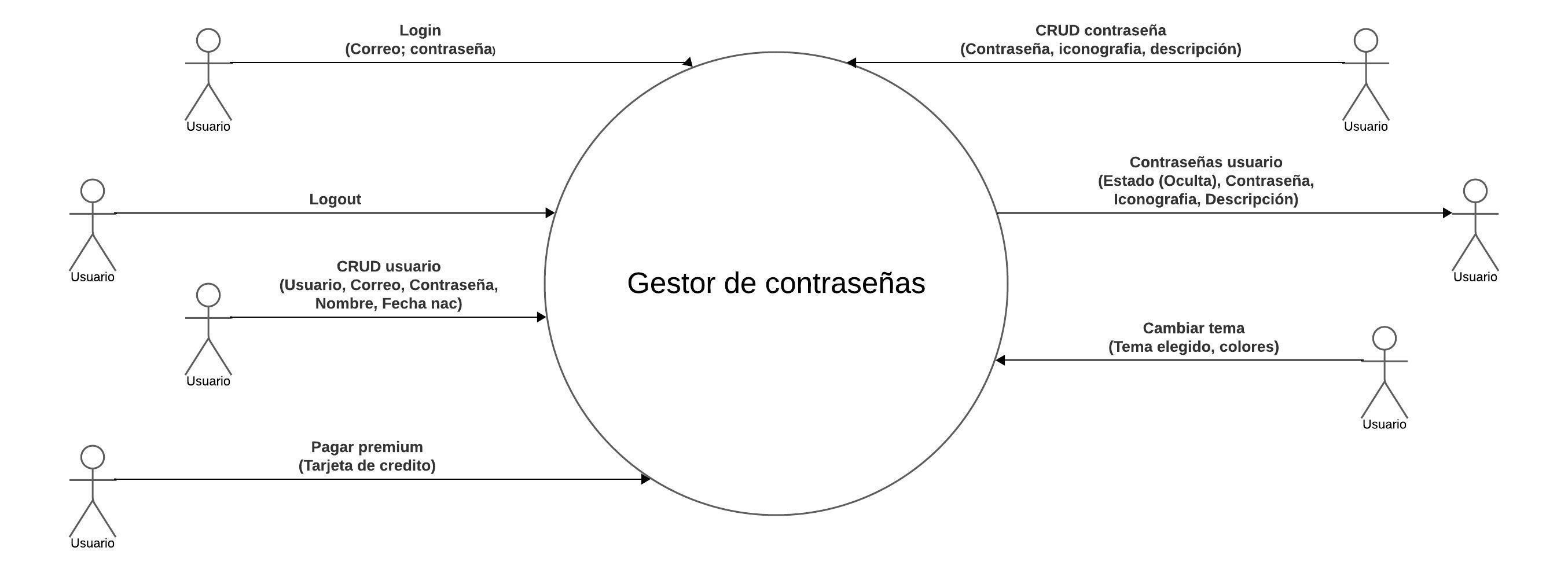
## Tamaños BD

* Tamaños varchar
  + 100 – *Pequeño*
  + 255 – *Mediano*
  + 512 – *Grande*

## Diagramas:

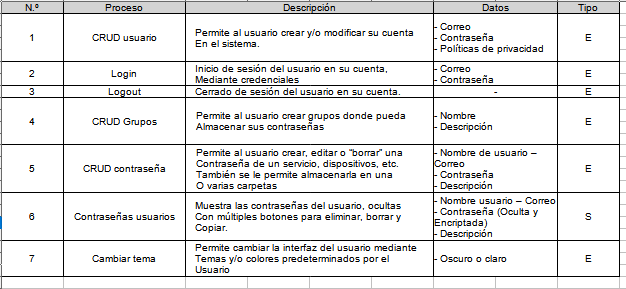
**Mapa de contexto**:

Diagrama con el cual de forma prematura se definen las entradas y salidas del sistema.

****

**Diagrama E/S:**

Continuación del diagrama anterior, explaya las E/S y los datos que utiliza y las respuestas que entrega el sistema

****

**Mapa preliminar de datos:**

También llamado “entidad relación”, es el esqueleto de lo que sera la base de datos y como funcionara.

