

# Informe del proyecto



**Nombre del Proyecto:** "SmartLock" Cerradura Electrónica.

**Apellidos y Nombres:** Ainiceder Francisco,

Lopez Zaupaiara,

Toledo Barrientos Alan.

**Profesor:** Alberto Gusbeth.

**Curso:** 7°1°

**Año:** 2023.

## **Índice**

Introducción.....	
1 Planteamiento del Problema.	
..... 1 Formulación del problema	
.....	1

Objetivos.....	
2                      justificación                      de                      la                      investigación	
.....	2
Limitaciones.....	
2	Marco
Teórico.....	3
Antecedentes de la Investigación .....	3
Base Teórica.....	5
Definición de Términos Básicos .....	7
Sistema de Variables.....	10
Lógica .....	14
Antecedentes.....	
	20



## **Introducción**

Este informe tratará de la investigación de cómo se realizó el “SmartLock”, el problema base que motivó nuestra inquietud es las inseguridades, perdidas de las llaves entre otras cosas. Llegando así al proyecto; para el mismo utilizaremos una mejor explicación, un entorno ficticio de empresa, esta organización tomará el nombre de “seguridad doméstica”.

A través de los datos extraídos de esta investigación nos enfocamos en una de sus problemáticas e imperfecciones a la hora de desempeñarse en su área. Que a la vez se la puede tomar como una actualización en este entorno y una mejora didáctica hacia la tecnología.

## **Planteamiento del Problema.**

La empresa de la ciudad de Corrientes Capital brinda seguridad a ciertos barrios de la localidad. Según estadísticas realizadas por esta empresa el 54% involucra que la puerta de entrada es el punto de acceso principal utilizado en la mayoría de los robos a casas en la ciudad. Aunque estas tengan una buena cerradura, por lo cual tenía muchos clientes insatisfechos por el servicio brindado. Esto llevó a que la empresa pierda muchos clientes.

Haciendo una vista general del panorama de hoy en día, son mayores los domicilios y establecimientos con cerraduras tradicionales, impidiendo el avance a nuevas tecnologías que puedan ir surgiendo. La mayoría de casas siguen teniendo esa forma de acceder tan clásica, una forma que tiene aspectos de vulnerabilidad por técnicas como el bumping o la ganzúa. Por eso, suele ser recomendable optar por cerraduras más avanzadas y seguras.

Acorde a las investigaciones que vimos, tomamos referencias de que hay más casos de robos, pérdidas y negligencia referidos a la seguridad en el ingreso a las casas debido a perder tiempo abriendo la puerta de manera tradicional.

## **Formulación del problema**

Para mejorar algo ya existente se necesitaron una serie de aspectos negativos para tener en cuenta, además, se requiere una mejora en la seguridad y el servicio que esta empresa ofrece, por lo que es importante identificar algunas características que requieren atención para lograr una mayor eficacia en estos.

- La hora del ingreso a la casa es un momento de gran vulnerabilidad para las personas, ya que, al estar sacando las llaves de su cartera, se distraen y se vuelven más propensas a sufrir robos o hurtos.
- En el caso de ser utilizado en un edificio, ayuda al control de cada inquilino o uso domiciliario que también ayudaría a la hora de tener un control de estas situaciones para mayor seguridad.



- Problemas con la cerradura, rotura de la misma, estancamiento de la llave, ya que, al momento de tener varios métodos de apertura, la única falla que puede tener es la de rotura de su mecanismo.

## **Objetivos**

### **Objetivos Generales:**

- Desarrollar un sistema de control de acceso seguro y confiable.
- Mejorar la comodidad en el acceso a nuestro hogar.
- Mejorar las medidas de seguridad y protección.
- Promover métodos de entrada eficientes y fáciles de usar para el usuario. • Asegurar la compatibilidad del sistema con la tecnología y dispositivos modernos.

### **Objetivos Específicos:**

- Diseñar un mecanismo de autenticación sólido para una identificación confiable del usuario.
- Implementar una interfaz fácil de usar para facilitar la interacción con el sistema.
- Desarrollar un algoritmo de cifrado eficiente para proteger los datos sensibles del usuario.
- Crear un canal de comunicación seguro entre la cerradura y el usuario de manera amigable.
- Integrar una función para situaciones de emergencia y asistencia inmediata. • Realizar pruebas rigurosas y garantía de calidad para asegurar la confiabilidad del sistema.
- Optimizar el consumo de energía para prolongar la vida útil de la batería de la cerradura.
- Proporcionar documentación completa, manuales de usuario para una mejor utilización y mantenimiento sin problemas.

## **justificación de la investigación**

La investigación se justifica por la necesidad de mejorar la seguridad, satisfacción del cliente y adoptar tecnologías más avanzadas en el ámbito de la seguridad de acceso en la ciudad de Corrientes Capital. Se busca abordar las vulnerabilidades existentes, mejorar la eficiencia y comodidad en el acceso, y brindar soluciones más seguras y confiables para la protección de los hogares y establecimientos.

## **Limitaciones**

Existen varias limitaciones internas a considerar antes de empezar un proyecto de una cerradura inteligente, entre ellas podemos mencionar:

- Alcance del proyecto: El proyecto se enfoca en la implementación física y desarrollo de una cerradura electrónica inteligente, abarcando tanto el hardware como el software. Sin embargo, no se incluye la responsabilidad de establecer conexiones o integraciones específicas con servicios o sectores externos.



- Integración con sistemas de seguridad existentes: Puede haber limitaciones en cuanto a la integración de la cerradura electrónica con sistemas de seguridad existentes en el hogar. Esto puede depender de la compatibilidad y los recursos disponibles, pero no será una responsabilidad directa del proyecto.
- Actualizaciones de seguridad: Se recomienda realizar actualizaciones periódicas del software y el hardware de la cerradura electrónica para abordar posibles vulnerabilidades y amenazas de seguridad. Sin embargo, no se brindará un servicio continuo de actualizaciones o monitoreo de seguridad.

## **Marco Teórico**

### **Antecedentes de la Investigación**

El picaporte fue inventado por el Arquitecto Alemán Adolf Richter en 1878, se interesó en mejorar los mecanismos de apertura de puertas con un diseño que pudiera ser usado con una mano. Después de varios años de pruebas y ensayos, Richter creó una varilla de metal con una forma curva que se ajustaba a la palma de la mano. Esta varilla se conectaba a una cerradura y se abría cuando se presionaba; durante la Edad Media, los picaportes o aldabas consistían en martillitos suspendidos en la parte exterior de las puertas. Estos martillitos, generalmente en forma de argolla o aro de hierro unido a una cabeza de bronce, se golpeaban sobre una cabeza de clavo para abrir la puerta.



Sin embargo, el picaporte moderno, tal como lo conocemos hoy en día, fue inventado en 1878 por Osbourn Dorsey. Presentó una patente para un mecanismo interno que cerraba las puertas. Este mecanismo consistía en una varilla de metal que se enganchaba en un agujero en el marco de la puerta, desatándola al accionar el pomo del picaporte. Antes de la invención del picaporte, las puertas solían asegurarse mediante cuerdas o barras que se levantaban o bajaban para mantenerlas cerradas.

3



#### Prácticas Profesionalizantes

En la actualidad, existen diferentes tipos y formas de picaportes, también conocidos como manijas, pomos, perillas o tiradores. Algunos son giratorios, otros fijos, y también hay aquellos que se deben empujar hacia abajo para abrir. Los materiales comunes utilizados en la fabricación de picaportes incluyen el latón, bronce, aluminio, zamak, hierro y otras aleaciones de metales.

Así, el picaporte ha experimentado una evolución a lo largo del tiempo, desde las aldabas medievales hasta los picaportes modernos, adaptándose a las necesidades y avances tecnológicos.

Tipo de Cerradura	Riesgo a ser alterada	Descripción de la vulnerabilidad
Cerraduras de llave	Moderada	Las cerraduras de llave pueden ser vulnerables al bumping, impresioning o técnicas de ganzuado, donde se manipula la cerradura para abrir sin llave.
Cerraduras de combinación	Moderada	Las cerraduras de combinación pueden ser vulnerables a ataques de fuerza bruta o adivinando la combinación, especialmente si se utilizan combinaciones débiles o se pueden deducir.

Cerraduras electrónicas	Baja	Las cerraduras electrónicas suelen ser más seguras ya que requieren una clave o tarjeta electrónica para abrir, aunque pueden ser vulnerables a ataques informáticos o hackeo si no están bien protegidas.
Cerraduras inteligentes	Baja	Las cerraduras inteligentes generalmente cuentan con medidas de seguridad avanzadas, como autenticación de múltiples factores y encriptación, lo que las hace menos propensas a ser violadas.
Cerraduras de cilindro	Alta	Las cerraduras de cilindro son propensas al bumping, ganzuado y ataques de extracción, donde se extrae el cilindro para abrir la cerradura sin dañarla.

4



Prácticas Profesionalizantes

## Base Teórica

### Base de Datos

	Plataforma (5)	Seguridad (30)	Código Abierto (15)	Permite valores nulos (5)	Permite PHP (10)	Funcionalidad (5)	Backup (30)	Nivel de Aprendizaje (30)	Claves primarias y foráneas (5)	Optimización de Consultas (5)	Herramientas (10)	compatible con ref. remota (30)	Puntuación
MySQL	Multiplataforma (5)	Seguridad baja (escrita los datos) (5)	Código abierto (15)	si permite (5)	si permite (10)	Proyectos pequeños y grandes (5)	Tiene copia de seguridad (30)	No requiere mucho conocimiento porque es fácil de usar (30)	si permiten (5)	Optimización media-alta (5)	Buena (5)	si posee (30)	55
Oracle	Multiplataforma (5)	Seguridad intermedia (seguridad de múltiples capas) (15)	No es de código abierto (15)	No permite (10)	si permite (10)	Proyectos pequeños y grandes (5)	Tiene copia de seguridad (30)	Requiere un nivel medio de conocimiento para utilizarla (5)	si permiten (5)	Optimización alta (5)	Muy buena (10)	si posee (30)	72
Microsoft SQL Server	Multiplataforma (5)	Seguridad alta (no permite acceso a los archivos. Monitoreo de bases de datos) (30)	No es de código abierto (15)	Si permite (5)	Si permite (10)	Proyectos pequeños y grandes (5)	Tiene copia de seguridad (30)	Requiere un nivel medio de conocimiento para utilizarla (5)	si permiten (5)	Optimización media-baja (11)	Buena (5)	si posee (30)	71

<https://www.javatpoint.com/mysql-vs-oracle>

<https://www.dnsstuff.com/mysql-vs-mssql-performance#differences-between-mssql-and-mysql-server>

<https://www.hostinger.com.ar/tutoriales/diferencia-mysql-sql-server#Diferencia entre MySQL y SQL Server>

<https://blog.hubspot.es/website/que-es-mysql>

### Hosting

	Panel de Control (30)	Valor (20)	Almacenamiento (30)	Transferencia (5)	Compatibilidad con PHP (10)	Soporte Técnico (30)	Copia de seguridad (30)	Certificado SSL (10)	Escalabilidad (5)	Publicidad invasiva (5)	Cuenta de correo electrónico (5)	Puntuación
000WebHost	0Panel (30)	Si gratis (20)	100MB (5)	5GB por mes (5)	Es compatible (10)	Si 24/7 (30)	Tanto la base de datos (30)	Tiene (10)	Muy buena escalabilidad (5)	no posee (5)	posee (5)	30
u00Hosting	0Panel (30)	Si gratis (20)	100MB (5)	5GB por mes (5)	Es compatible (10)	no tiene (30)	Tanto la base de datos (30)	Tiene (10)	Buena escalabilidad (5)	no posee (5)	posee (5)	40
Bluehosting	Contiene un panel y 0Panel de uso (30)	Si gratis (20)	1GB (5)	5GB por mes (5)	Es compatible (10)	no tiene (30)	Tanto la base de datos (30)	Tiene (10)	Buena escalabilidad (5)	no posee (5)	no posee (5)	70

<https://www.crearpaginaweb.gratis/mejor-hosting/gratis/>

<https://saasscout.com/hosting/x10host-vs-000webhost/>

## Servidor Local

	Precio (10)	Tiempo de respuesta (20)	Almacenamiento (10)	IPv6 (5)	Instalación (10)	Migración a otro servidor local (10)	Manera de uso (10)	Código abierto (10)	Estabilidad (5)	Rendimiento (10)	Flexibilidad (5)	Puntuación
Apache	Es gratis (5)	Respuesta inmediata (10)	Tiene bajo consumo de memoria (10)	Es compatible (5)	Fácil y fácil (10)	Si permite (10)	Fácil de usar para todos los usuarios (10)	Es de código abierto (10)	Muy eficiente (5)	Satisfactorio bueno (10)	Muy flexible (5)	95
LiteSpeed	Es gratis (5)	Respuesta inmediata (10)	Tiene bajo consumo de memoria (10)	No es compatible (5)	Fácil y fácil (10)	Si permite (10)	Fácil de usar para todos los usuarios (10)	Es de código abierto (10)	Muy eficiente (5)	Aún más rápido (10)	Muy flexible (5)	90
Nginx	Es gratis (5)	Respuesta inmediata (10)	Tiene bajo consumo de memoria (10)	Es compatible (5)	Fácil y fácil (10)	Si permite (10)	Requiere configuración previa (10)	Es de código abierto (10)	Muy eficiente (5)	Muy rápido (10)	Muy flexible (5)	90

<https://www.stackscale.com/es/blog/top-servidores-web/>

<https://www.nginx.com/resources/wiki/>

<https://www.litespeedtech.com/open-source/litespeed-sapi/php>

<https://chemicloud.com/blog/litespeed-vs-nginx-vs-apache/>

5



Practicas Profesionalizantes

## Framework PHP

	Registro (10)	Nivel de Aprendizaje (5)	Rendimiento con PHP (10)	Seguridad (10)	Instalación (5)	Migraciones (10)	Integrado de datos (5)	Código abierto (10)	Comunidad (5)	Estrellas GitHub (5)	Puntuación
Laravel	Tiene la mayor cantidad de documentación y es fácil de encontrar (15)	Se necesita un nivel mediano a alto de conocimiento para utilizarla (10)	Tiene un alto rendimiento (10)	Tiene la mayor seguridad para php (10)	Fácil de instalar (5)	Manuales (10)	No se requiere mucho conocimiento de SQL (10)	Si (10)	Extensiva Comunidad (5)	46,46 (5)	98
CodeIgniter	No tiene mucha documentación (5)	Es fácil de utilizar por lo que no requiere mucho conocimiento (10)	Tiene un alto rendimiento (10)	Seguridad alta para php (10)	Fácil de instalar (5)	Automáticas (10)	Requiere bastante conocimiento de SQL (10)	Si (10)	Buena Comunidad (5)	18,18 (5)	70
Symfony	Tiene gran cantidad de documentación (10)	Se necesita un nivel mediano a alto de conocimiento para utilizarla (10)	Tiene un rendimiento medio (5)	Seguridad media para php (10)	Fácil de instalar (5)	Automáticas (10)	Requiere bastante conocimiento de SQL (10)	No (10)	Comunidad Buena (5)	21,86 (5)	72

<https://www.hostinger.com.ar/tutoriales/mejores-frameworks-php>

<https://infostride.com/laravel-vs-codeigniter/>

## Framework CSS

	Adaptación (5)	Utilización (10)	Documentación (10)	Nivel de Aprendizaje (10)	Instalación (5)	Código abierto (5)	Responsive Design (10)	Stufo complejo (5)	Ejemplo en Carga Rápida (10)	Personalización (10)	Compatibilidad con navegadores (10)	Plataforma (5)	Puntuación
Tailwind CSS	Utiliza base de CSS (5)	Fácil de modificar los estilos (10)	Tiene como un manual (10)	Requiere un nivel medio de conocimiento (10)	No se instala, se utiliza mediante código (5)	Si (10)	Si (10)	Si (10)	Compatible (10)	Alta (10)	Buena (10)	Multiplataforma (5)	98
Bootstrap	Utiliza HTML (5)	Complejo y le toma de modificar los estilos (10)	Gran cantidad de documentación (10)	No requiere mucho conocimiento (10)	No se instala, se utiliza mediante código (5)	Si (10)	Si (10)	No (10)	Carga lenta (10)	Resp. info en detalles pequeños (10)	Buena (10)	Multiplataforma (5)	75
Bulma	Utiliza base de CSS (5)	Fácil de modificar los estilos (10)	No tiene documentación (10)	Requiere un nivel medio de conocimiento (10)	Fácil de instalar (5)	Si (10)	No (10)	No (10)	Compatible (10)	Medio alta (10)	Medio alta (10)	Multiplataforma (5)	97

<https://desarrolloweb.com/coleccion/es/frameworks-css>

<https://blog.logrocket.com/bulma-vs-tailwind-css-better-bootstrap-alternative/>

## Editor de Texto

	Plataforma (5)	Costo (5)	Atajos de teclado (10)	Herramientas (20)	Interfaz (10)	Super código (10)	Instalación (8)	Extensiones (10)	Resultado de errores y advertencias (10)	Auto- guardado y recuperación (12)	Puntuación
Visual Studio Code	multiplataforma (5)	servicio gratis (5)	si posee (10)	si posee (20)	sencilla e intuitiva (10)	si posee (10)	facil de instalar (8)	Si posee (10)	Viene incluido (10)	Viene incluido (12)	100
Notepad++	multiplataforma (5)	servicio gratis (5)	si posee (10)	si posee (20)	sencilla e intuitiva (10)	si posee (10)	facil de instalar (8)	Si posee (10)	Es personalizable (5)	Es personalizable (16)	89
Sublime Text 3	multiplataforma (5)	servicio paga (0)	si posee (10)	si posee (20)	sencilla e intuitiva (10)	no tiene (0)	facil de instalar (8)	Si posee (10)	Es personalizable (5)	Es personalizable (16)	74

<https://www.versuante.com/visual-studio-code-o-sublime-text-que-editor-es-mejor/>

<https://www.muycomputer.com/2023/03/07/notepad-8-5-nueva-version-del-mejor-bloc-de-notas/>

6



Practicas Profesionalizantes

Microcontroladores



<https://robocraze.com/blogs/post/arduino-vs-nodemcu>

<https://www.xataka.com/basics/arduino-raspberry-pi-que-cuales-sus-diferencias>

<https://picockpit.com/raspberry-pi/es/todo-sobre-la-frambuesa-pi-pico/>

<https://www.espressif.com/en/products/devkits>

<https://www.raspberrypi.com/documentation/microcontrollers/raspberry-pi-pico.html>

### **Prototipado a Implementación**



<https://es.wikipedia.org/wiki/Contactor>

<https://es.wikipedia.org/wiki/Rel%C3%A9>





<https://hetpro-store.com/TUTORIALES/modulo-lector-rfid-rc522-rf-con-arduino/>  
<https://programarfacil.com/blog/arduino-blog/lector-rfid-rc522-con-arduino/>  
<https://ja-bots.com/producto/modulo-lector-rfid-rdm6300-de-125khz/>



<https://www.digikey.com/es/product-highlight/s/stmicroelectronics/stm32wb-32-bit-mcu-wireless-module>  
<https://programarfacil.com/podcast/esp8266-wifi-coste-arduino/>  
<https://accedacris.ulpgc.es/bitstream/10553/76842/2/TFG%20Irene%20Merino%20Fern%C3%A1ndez.pdf>

### **Definición de Términos Básicos**

**Visual Studio** es un editor potente y personalizable que puede ser utilizado por desarrolladores de todo tipo. Soporta una amplia gama de lenguajes de programación y tiene una gran comunidad de usuarios y desarrolladores.

### **Prácticas Profesionalizantes**

**Framework** es una estructura o conjunto de herramientas que proporcionan una base para el desarrollo de aplicaciones. Estos pueden ayudar a los desarrolladores a acelerar el desarrollo, mejorar la calidad y el rendimiento de sus aplicaciones.

**Bootstrap** es un framework que facilita el desarrollo web al proporcionar componentes, estilos y utilidades listas para usar, lo que permite crear sitios web de manera más rápida y sencilla, sin necesidad de escribir todo el código CSS y JavaScript desde cero.

**Tailwind CSS** es un framework CSS que se utiliza para crear interfaces de usuario (UI) de forma rápida y sencilla. A diferencia de los frameworks CSS tradicionales, Tailwind CSS no proporciona clases predefinidas para los elementos de la interfaz de usuario. En su lugar, proporciona una biblioteca de clases de utilidad que se pueden combinar para crear diseños personalizados.

**MySQL** es un sistema de gestión de bases de datos relacionales ampliamente utilizado en el desarrollo de aplicaciones web y empresariales. Se basa en el modelo de base de datos relacional, donde los datos se organizan en tablas, filas y columnas; Permitiendo almacenar, gestionar y manipular grandes cantidades de datos de manera eficiente y segura.

**PHP** es un lenguaje de programación interpretado, lo que significa que no necesita

ser compilado antes de ser ejecutado. Esto lo hace ideal para el desarrollo web, ya que permite que las páginas web se carguen más rápidamente. PHP también es un lenguaje muy versátil, y se puede utilizar para crear una amplia gama de aplicaciones web, incluyendo sitios web dinámicos, aplicaciones web, y servicios web.

**Laravel** es un marco de desarrollo de aplicaciones web que proporciona un conjunto estructurado de herramientas y recursos predefinidos. Se basa en patrones de diseño y prácticas recomendadas para agilizar el proceso de desarrollo, permitiendo a los programadores construir aplicaciones web de manera eficiente y organizada. Ofrece características como enrutamiento, manejo de bases de datos, autenticación y muchas otras que facilitan la creación de aplicaciones robustas y escalables

**Arduino** es una placa de prototipado electrónico de código abierto que permite a los usuarios crear proyectos interactivos; es una placa de hardware con un microcontrolador y un entorno de desarrollo integrado (IDE) fácil de usar. Arduino simplifica la programación y la conexión de componentes electrónicos, como sensores o mecanismos de accionamiento, permitiendo a los usuarios construir proyectos interactivos sin requerir un conocimiento profundo de electrónica. Es utilizado en proyectos de robótica, domótica, arte interactivo y muchas otras áreas.

**Blade** en Laravel es como un conjunto de herramientas que te ayuda a crear páginas web de manera más fácil. Piensa en ello como un conjunto de piezas que puedes

#### Practicas Profesionalizantes

combinar para hacer un diseño web, donde algunas piezas pueden mostrar información que cambia y otras pueden ayudarte a hacer decisiones simples, como "si esto, muestra aquello". Te ayuda a hacer páginas web bonitas y funcionales sin tener que escribir demasiado código complejo

**Migraciones** las migraciones son un conjunto de instrucciones que definen cómo se deben crear o modificar tablas en una base de datos. Estas instrucciones permiten mantener la estructura de la base de datos de manera controlada y fácil de manejar, sin necesidad de intervenir manualmente en la configuración de la tabla cada vez que cambia.

**HTML** es un lenguaje de marcado que se utiliza para crear páginas web, se basa en etiquetas, que son palabras clave que encierran el texto y le dan formato; es un lenguaje muy simple y fácil de aprender. Se puede utilizar para crear páginas web básicas, así como páginas web más complejas. Además, es un lenguaje estándar, lo que significa que es compatible con todos los navegadores web.


**Apache** es un servidor web de código abierto ampliamente utilizado para alojar sitios web y servir contenido en línea. Es conocido por su estabilidad, seguridad y flexibilidad.

**JavaScript** es un lenguaje de programación interpretado que se utiliza principalmente para crear interactividad en sitios web; es un lenguaje de scripting versátil y ampliamente utilizado que permite agregar funcionalidades dinámicas a las páginas web, como manipulación del DOM (Document Object Model), interacciones del usuario, animaciones, validaciones de formularios y comunicación con servidores. Se ejecuta en el lado del cliente, es decir, en el navegador web del usuario, lo que le brinda la capacidad de interactuar con la página web de forma dinámica y responder a eventos en tiempo real.

**Back-end** es la parte de una aplicación web que no es visible para el usuario. Es la parte de la aplicación que se ejecuta en el servidor y se encarga de procesar las solicitudes del usuario y generar la respuesta.

**Base de datos** es un lugar donde puedes guardar mucha información. Esta información puede ser sobre personas, lugares, cosas o eventos. Las bases de datos se utilizan para hacer que sea fácil encontrar la información que necesitas, y para mantener la información organizada.

**GitHub** es un sitio web donde los desarrolladores pueden almacenar y compartir código. Es una herramienta popular para el desarrollo de software colaborativo, ya que permite a los desarrolladores trabajar en el mismo proyecto al mismo tiempo, incluso si están en diferentes lugares, también ofrece una variedad de características que facilitan la colaboración, como foros de discusión, notificaciones y integración con otras herramientas de desarrollo.

**Prácticas Profesionalizantes**  Una **migración** es el proceso de mover un repositorio de un lugar a otro. Esto se puede hacer por varias razones, como cambiar de proveedor de alojamiento de repositorios, combinar varios repositorios en uno o mover un repositorio a una nueva organización.

El **código abierto** es un modelo de desarrollo de software que permite a cualquier persona acceder, modificar y distribuir el código fuente de un programa.

La **normalización** de bases de datos es un proceso en el diseño de bases de datos que busca organizar los datos de manera eficiente y evitar la redundancia e inconsistencia. Se compone de varias formas normales (1NF, 2NF, 3NF, etc.) que establecen reglas específicas para garantizar la integridad y eficiencia de los datos.

## Sistema de Variables

#### users

id INT, PK (Belongs to primary key), NN (Not Null) name  
VARCHAR (45)  
gmail VARCHAR (45)  
password VARCHAR (45)  
rol VARCHAR (45)

#### rfid

id INT, PK (Belongs to primary key), NN (Not Null)  
code VARCHAR (45)  
id\_user INT, NN (Not Null)

#### state

id INT, PK (Belongs to primary key), NN (Not Null) state  
VARCHAR (10)  
id\_rfid INT, NN (Not Null)

#### system information

id INT, PK (Belongs to primary key), NN (Not Null) time  
VARCHAR (10)  
state INT, NN (Not Null)

#### input or output

id INT, PK (Belongs to primary key), NN (Not Null) id\_rfid INT,  
NN (Not Null)  
time INT, NN (Not Null)

10

Practicas Profesionalizantes

#### role

id bigint(20) unsigned  
name varchar (255)  
guard\_name varchar (255)  
created\_at Timestamp  
update\_at Timestamp

#### permissions

```
id bigint(20) unsigned  
name varchar (255)  
guard_name varchar (255)  
created_at Timestamp  
update_at Timestamp
```

#### role\_has\_permissions

```
id_permission bigint(20) unsigned  
id_role bigint(20) unsigned
```

### Relaciones

#### 1. users -> rfid:

- Un usuario puede tener varios registros RFID, pero cada registro RFID pertenece a un solo usuario. Esto crea una relación de uno a muchos (1:N) entre "users" y "rfid".
- Se crea una clave foránea `id\_user` en la tabla "rfid" que hace referencia al campo `id` en la tabla "users".

#### 2. rfid -> state:

- Un registro RFID puede estar asociado con varios estados, pero cada estado está vinculado a un solo registro RFID. Esto crea una relación de uno a muchos (1:N) entre "rfid" y "state".
- Se crea una clave foránea `id\_rfid` en la tabla "state" que hace referencia al campo `id` en la tabla "rfid".

#### 3. state -> system information:

- Un estado puede estar relacionado con varios registros de información del sistema, pero cada registro de información del sistema corresponde a un solo estado. Esto crea una relación de uno a muchos (1:N) entre "state" y "system information".
- Se crea una clave foránea `id\_rfid` en la tabla "system information" que hace referencia al campo `id` en la tabla "state".

#### 4. system information -> input or output:

- Un registro de información del sistema puede estar asociado con varios registros de entrada o salida, pero cada registro de entrada o salida está relacionado con un solo registro de información del sistema. Esto crea una relación de uno a muchos (1:N) entre "system

information" y "input or output".

- Se crea una clave foránea `id` en la tabla "input or output" que hace referencia al campo `id` en la tabla "system information".

5. users -> role:

- Un usuario puede tener un rol, y cada rol puede asignarse a varios usuarios. Esto crea una relación de uno a muchos (1:N) entre "users" y "role".
- No se menciona explícitamente una clave foránea, pero la relación se establece a través del campo "rol" en la tabla "users" y el campo "name" en la tabla "role".

6. role -> role\_has\_permissions:

- Un rol puede tener varios permisos, y cada permiso puede estar asociado con varios roles. Esto crea una relación de muchos a muchos (N:M) entre "role" y "permissions" a través de la tabla "role\_has\_permissions".

- Se crean dos claves foráneas, `id\_permission` e `id\_role`, en la tabla "role\_has\_permissions" que hacen referencia a las respectivas claves primarias en las tablas "role" y "permissions".

### Normalizacion

1NF (Primera Forma Normal):

- Cada tabla cumple con la 1NF al tener una clave primaria única.
- La clave primaria no contiene valores nulos.
- Los campos que no son clave deben depender por completo de la clave primaria.

2NF (Segunda Forma Normal):

- Cumple con los requisitos de la 1NF.
- Los atributos que no son clave deben depender de la clave primaria completa.
- Evita dependencias parciales, lo que significa que los atributos que no son clave no deben depender solo de una parte de la clave primaria.

3NF (Tercera Forma Normal):

- Cumple con los requisitos de la 2NF.

### **Practicas Profesionalizantes**

- No existen dependencias transitivas entre los atributos que no son clave, lo que implica que los atributos que no son clave no deben depender de otros atributos que no sean clave.





13

**Practicas Profesionalizantes**

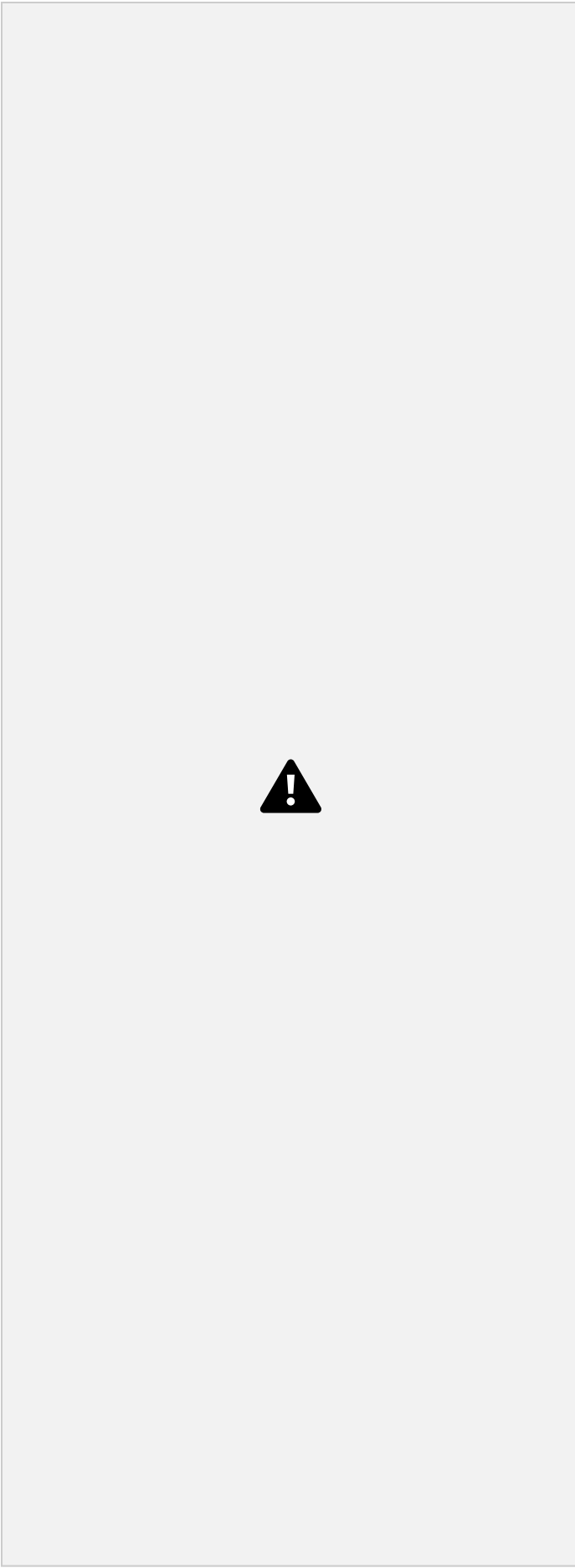








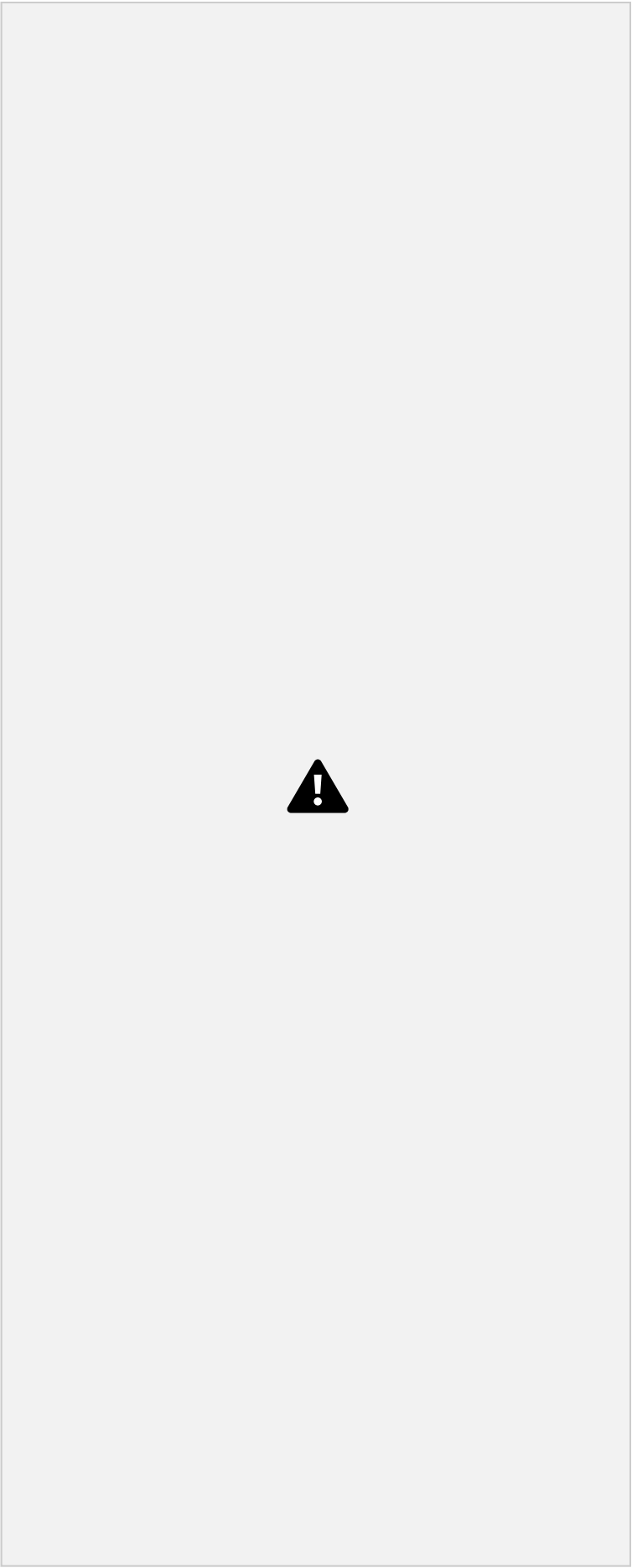




**Practicas Profesionalizantes**

**Practicas Profesionalizantes**







## **Antecedentes**

<https://conceptodefinicion.net/descubre-quien-invento-el-picaporte-historia-de-un-invento-unico/>

<https://www.lavanguardia.com/participacion/las-fotos-de-los-lectores/20211106/7839124/puertas-grandes.html>

<https://es.411answers.com/a/quien-invento-el-picaporte.html>

