

## Documentación del Esquema de la Base de Datos

Realice la siguiente base de datos

La startup Pixel-Security-360 requiere del diseño de la base de datos, para su plataforma one-click one -secutiry. Donde fueron contratados como los Administradores de base de datos .

El cual la compañía desea implementar una plataforma para la visualización de Pruebas de penetración de seguridad en el cual será una plataforma web, a sus clientes. El sistema debe contar con lo siguiente:

- Módulo de usuario
- Módulo de reporte
- Módulo de pago
- Módulo de tipos de pruebas
- Módulo de autenticación

I PARTE - Estructuras y registros - 60 puntos

El diseño de base de datos debe tener como mínimo:

- En módulo de autenticación: se necesita nombre del usuario, contraseña, agente de usuario (nombre del dispositivo) y un token.
  - En módulo de usuario: Los usuarios tienen: nombre, apellido, nombre usuario, token, contraseña, ciudad, sexo, estado civil, tipo de empresa (pública o privada) y dirección.
  - Los tipos usuarios del sistema son: Cliente, Administradores, Vendedores, Ejecutivos y otros.
  - En el módulo de tipos de pruebas: Se debe poder agregar información al catálogo de pruebas, en cuál tiene los siguientes campos: referencia, nombre, descripción, fecha de ingreso y estado (activo o desactivado)
- Con base en la imagen anterior, se debe tener insertada en dicho módulo, registros iniciales con excepción de “informe”.
- Módulo de pago: Este módulo consiste en poder almacenar la información qué

tendrá el formulario de pago, qué debe estar relación con un usuario de la plataforma.

Base de Datos: pixel\_security\_360

La base de datos pixel\_security\_360 se utiliza para gestionar la autenticación de usuarios, tipos de usuarios, pruebas de seguridad, pagos realizados y reportes generados en el sistema de seguridad Pixel Security 360.

Tablas:

#### **Tabla autenticacion**

Almacena la información de autenticación de los usuarios.

- Campos:

autenticacion\_id: Identificador único de autenticación (clave primaria).

usuario\_id: Identificador del usuario (clave externa referenciando la tabla usuario).

nombre\_usuario: Nombre de usuario único.

contrasena: Contraseña del usuario.

agente\_usuario: Agente de usuario utilizado.

token: Token de autenticación.

- Restricciones:

La columna usuario\_id referencia la columna usuario\_id de la tabla usuario.

#### **Tabla tipo\_usuario**

Almacena los tipos de usuarios disponibles en el sistema.

- Campos:

tipo\_usuario\_id: Identificador único del tipo de usuario (clave primaria).

nombre\_tipo: Nombre del tipo de usuario.

### **Tabla usuario**

Almacena la información de los usuarios del sistema.

- Campos:

usuario\_id: Identificador único del usuario (clave primaria).

nombre: Nombre del usuario.

apellido: Apellido del usuario.

nombre\_usuario: Nombre de usuario único.

contrasena: Contraseña del usuario.

ciudad: Ciudad del usuario.

sexo: Género del usuario (M, F u O).

estado\_civil: Estado civil del usuario.

tipo\_usuario\_id: Identificador del tipo de usuario (clave externa referenciando la tabla tipo\_usuario).

direccion: Dirección del usuario.

- Restricciones:

La columna tipo\_usuario\_id referencia la columna tipo\_usuario\_id de la tabla tipo\_usuario.

### **Tabla pruebas**

Almacena información sobre las pruebas de seguridad realizadas.

- Campos:

prueba\_id: Identificador único de la prueba (clave primaria).

referencia: Referencia única de la prueba.

nombre: Nombre de la prueba.

descripcion: Descripción de la prueba.

fecha\_ingreso: Fecha de ingreso de la prueba (valor por defecto: fecha y hora actual).

estado: Estado de la prueba (activo o desactivado).

usuario\_id: Identificador del usuario asociado a la prueba (clave externa referenciando la tabla usuario).

- Restricciones:

La columna usuario\_id referencia la columna usuario\_id de la tabla usuario.

## Tabla pago

Almacena información sobre los pagos realizados por los usuarios.

- Campos:

pago\_id: Identificador único del pago (clave primaria).

usuario\_id: Identificador del usuario que realizó el pago (clave externa referenciando la tabla usuario).

monto: Monto del pago.

fecha\_pago: Fecha de realización del pago (valor por defecto: fecha y hora actual).

- Restricciones:

La columna usuario\_id referencia la columna usuario\_id de la tabla usuario.

## Tabla reporte

Almacena los reportes generados a partir de las pruebas de seguridad.

- Campos:

reporte\_id: Identificador único del reporte (clave primaria).

usuario\_id: Identificador del usuario asociado al reporte (clave externa referenciando la tabla usuario).

prueba\_id: Identificador de la prueba asociada al reporte (clave externa referenciando la tabla pruebas).

fecha\_reporte: Fecha de generación del reporte (valor por defecto: fecha y hora actual).

resultado: Resultado del reporte.

- Restricciones:

La columna usuario\_id referencia la columna usuario\_id de la tabla usuario.

La columna prueba\_id referencia la columna prueba\_id de la tabla pruebas.

- Observaciones:

La base de datos incluye un conjunto de datos de ejemplo para las tablas usuario, pruebas, pago y reporte.

Las relaciones entre las tablas están establecidas mediante claves primarias y claves externas para garantizar la integridad referencial de los datos.

Se utilizan convenciones de nomenclatura significativas para los nombres de las tablas y los campos.

Se han establecido valores predeterminados apropiados para ciertos campos, como fechas y estados.

## Prueba:

```
/* Filas afectadas: 1 Filas encontradas: 0 Advertencias: 0 Duración para 2 de 13 consultas: 0,015 seg. */
CREATE DATABASE IF NOT EXISTS pixel_security_360;
/* Filas afectadas: 1 Filas encontradas: 0 Advertencias: 1 Duración para 1 consulta: 0,000 seg. */
SHOW WARNINGS LIMIT 5;
SHOW DATABASES;
/* Entrando a la sesión "Unamed" */
SELECT `DEFAULT_COLLATION_NAME` FROM `information_schema`.`SCHEMATA` WHERE `SCHEMA_NAME`='pixel_security_360';
SHOW TABLE STATUS FROM `pixel_security_360`;
SHOW FUNCTION STATUS WHERE `Db`='pixel_security_360';
SHOW PROCEDURE STATUS WHERE `Db`='pixel_security_360';
SHOW TRIGGERS FROM `pixel_security_360`;
SELECT *, EVENT_SCHEMA AS `Db`, EVENT_NAME AS `Name` FROM information_schema.`EVENTS` WHERE `EVENT_SCHEMA`='pixel_security_360';
-- Tabla de tipo de usuario CREATE TABLE IF NOT EXISTS tipo_usuario ( tipo_usuario_id INT AUTO_INCREMENT, nombre_tipo VARCHAR(100) NOT NULL, PRIMARY KEY(tipo_usuario_id) );
-- Insertar tipos de usuarios iniciales INSERT INTO tipo_usuario (nombre,tipo) VALUES ('Cliente'), ('Administrador'), ('Vendedor'), ('Ejecutivo'), ('Otro');
/* Filas afectadas: 5 Filas encontradas: 0 Advertencias: 0 Duración para 2 consultas: 0,047 seg. */
CREATE TABLE IF NOT EXISTS usuario ( usuario_id INT AUTO_INCREMENT, nombre VARCHAR(100) NOT NULL, apellido VARCHAR(100) NOT NULL, nombre_usuario VARCHAR(100) UNIQUE NOT NULL, contrasena VARCHAR(255) NOT NULL, agente_usuario VARCHAR(100) );
/* Filas afectadas: 0 Filas encontradas: 0 Advertencias: 0 Duración para 1 consulta: 0,047 seg. */
USE pixel_security_360;
CREATE TABLE IF NOT EXISTS autenticacion ( autenticacion_id INT AUTO_INCREMENT, usuario_id INT UNIQUE, nombre_usuario VARCHAR(100) UNIQUE NOT NULL, contrasena VARCHAR(255) NOT NULL, agente_usuario VARCHAR(100) );
/* Filas afectadas: 0 Filas encontradas: 0 Advertencias: 0 Duración para 2 consultas: 0,032 seg. */
SHOW DATABASES;
/* Entrando a la sesión "Unamed" */
SELECT `DEFAULT_COLLATION_NAME` FROM `information_schema`.`SCHEMATA` WHERE `SCHEMA_NAME`='pixel_security_360';
SHOW TABLE STATUS FROM `pixel_security_360`;
SHOW FUNCTION STATUS WHERE `Db`='pixel_security_360';
SHOW PROCEDURE STATUS WHERE `Db`='pixel_security_360';
SHOW TRIGGERS FROM `pixel_security_360`;
SELECT *, EVENT_SCHEMA AS `Db`, EVENT_NAME AS `Name` FROM information_schema.`EVENTS` WHERE `EVENT_SCHEMA`='pixel_security_360';
-- Crear la base de datos CREATE DATABASE IF NOT EXISTS pixel_security_360;
-- Usar la base de datos creada USE pixel_security_360;
CREATE TABLE IF NOT EXISTS autenticacion ( autenticacion_id INT AUTO_INCREMENT, usuario_id INT UNIQUE, nombre_usuario VARCHAR(100) UNIQUE NOT NULL, contrasena VARCHAR(255) NOT NULL, agente_usuario VARCHAR(100) );
-- Tabla de tipo de usuario CREATE TABLE IF NOT EXISTS tipo_usuario ( tipo_usuario_id INT AUTO_INCREMENT, nombre_tipo VARCHAR(100) NOT NULL, PRIMARY KEY(tipo_usuario_id) );
-- Insertar tipos de usuarios iniciales INSERT INTO tipo_usuario (nombre,tipo) VALUES ('Cliente'), ('Administrador'), ('Vendedor'), ('Ejecutivo'), ('Otro');
-- Tabla de usuario con relación a tipo de usuario CREATE TABLE IF NOT EXISTS usuario ( usuario_id INT AUTO_INCREMENT, nombre VARCHAR(100) NOT NULL, apellido VARCHAR(100) NOT NULL, nombre_usuario VARCHAR(100) UNIQUE NOT NULL, contrasena VARCHAR(255) NOT NULL, agente_usuario VARCHAR(100) );
-- Insertar datos de usuarios INSERT INTO usuario (nombre, apellido, nombre_usuario, contrasena, ciudad, sexo, estado_civil, tipo_usuario_id, direccion) VALUES ('Juan', 'Pérez', 'Juan123', 'password1', 'Ciudad A', 'M', 'casado', 1, 'Calle 123');
/* Error de SQL (1265): Data truncated for column 'estado_civil' at row 2 */
/* Filas afectadas: 6 Filas encontradas: 0 Advertencias: 4 Duración para 6 de 13 consultas: 0,047 seg. */
SHOW WARNINGS LIMIT 5;
CREATE TABLE IF NOT EXISTS prueba ( prueba_id INT PRIMARY KEY AUTO_INCREMENT, referencia VARCHAR(100) UNIQUE NOT NULL, nombre VARCHAR(100) NOT NULL, descripcion TEXT, fecha_ingreso TIMESTAMP DEFAULT CURRENT_TIMESTAMP );
/* Filas afectadas: 0 Filas encontradas: 0 Advertencias: 0 Duración para 1 consulta: 0,031 seg. */
CREATE TABLE IF NOT EXISTS pago ( pago_id INT AUTO_INCREMENT, usuario_id INT, monto DECIMAL(10, 2) NOT NULL, fecha_pago TIMESTAMP DEFAULT CURRENT_TIMESTAMP, FOREIGN KEY (usuario_id) REFERENCES usuario(usuario_id) );
/* Filas afectadas: 0 Filas encontradas: 0 Advertencias: 0 Duración para 1 consulta: 0,031 seg. */
CREATE TABLE IF NOT EXISTS reporte ( reporte_id INT AUTO_INCREMENT, usuario_id INT, prueba_id INT, fecha_reporte TIMESTAMP DEFAULT CURRENT_TIMESTAMP, resultado TEXT, FOREIGN KEY (usuario_id) REFERENCES usuario(usuario_id) );
/* Filas afectadas: 0 Filas encontradas: 0 Advertencias: 0 Duración para 1 consulta: 0,063 seg. */
INSERT INTO tipo_usuario (nombre,tipo) VALUES ('Cliente'), ('Administrador'), ('Vendedor'), ('Ejecutivo'), ('Otro');
/* Filas afectadas: 5 Filas encontradas: 0 Advertencias: 0 Duración para 1 consulta: 0,031 seg. */

330 -- Insertar datos en la tabla usuario con los tipos de usuario correspondientes INSERT INTO usuario (nombre, apellido, nombre_usuario, contrasena, ciudad, sexo, estado_civil, tipo_usuario_id, direccion) VALUES ('Juan', 'Pérez',
331 /* Error de SQL (1265): Data truncated for column 'estado_civil' at row 2 */
332 /* Filas afectadas: 0 Filas encontradas: 0 Advertencias: 0 Duración para 0 de 2 consultas: 0,000 seg. */
333 INSERT INTO usuario (nombre, apellido, nombre_usuario, contrasena, ciudad, sexo, estado_civil, tipo_usuario_id, direccion) VALUES ('Juan', 'Pérez', 'Juan123', 'password1', 'Ciudad A', 'M', 'casado', 1, 'Calle 123'); -- Cliente
334 /* Error de SQL (1265): Data truncated for column 'estado_civil' at row 2 */
335 /* Filas afectadas: 0 Filas encontradas: 0 Advertencias: 0 Duración para 0 de 2 consultas: 0,000 seg. */
336 INSERT INTO usuario (nombre, apellido, nombre_usuario, contrasena, ciudad, sexo, estado_civil, tipo_usuario_id, direccion) VALUES ('Juan', 'Pérez', 'Juan123', 'password1', 'Ciudad A', 'M', 'casado', 1, 'Calle 123'); -- Cliente
337 -- Otro
338 /* Filas afectadas: 10 Filas encontradas: 0 Advertencias: 0 Duración para 2 consultas: 0,000 seg. */
339 INSERT INTO autenticacion (usuario_id, nombre_usuario, agente_usuario, token) VALUES (1, 'Juan123', 'password1', 'Equipo1', MD5(RAND())); (2, 'maria456', 'password2', 'Equipo2', MD5(RAND())); (3, 'pedro789', 'password3', 'Equipo3', MD5(RAND()));
340 /* Error de SQL (1452): Cannot add or update a child row: a foreign key constraint fails ('pixel_security_360`.`autenticacion`, CONSTRAINT `autenticacion_ibfk_1` FOREIGN KEY (`usuario_id`) REFERENCES `usuario` (`usuario_id`)) */
341 /* Filas afectadas: 0 Filas encontradas: 0 Advertencias: 0 Duración para 0 de 1 consulta: 0,000 seg. */
342 SELECT * FROM usuario;
343 /* Filas afectadas: 0 Filas encontradas: 10 Advertencias: 0 Duración para 1 consulta: 0,016 seg. */
344 INSERT INTO autenticacion (usuario_id, nombre_usuario, contrasena, agente_usuario, token) VALUES (24, 'Juan123', 'password1', 'Equipo1', MD5(RAND())); (25, 'maria456', 'password2', 'Equipo2', MD5(RAND())); (26, 'pedro789', 'password3', 'Equipo3', MD5(RAND()));
345 /* Filas afectadas: 10 Filas encontradas: 0 Advertencias: 0 Duración para 1 consulta: 0,015 seg. */
346 SELECT * FROM usuario;
347 /* Filas afectadas: 0 Filas encontradas: 10 Advertencias: 0 Duración para 1 consulta: 0,000 seg. */
348 INSERT INTO pago (usuario_id, monto) VALUES (24, 50.00), (25, 75.00), (26, 100.00), (27, 125.00), (28, 150.00), (29, 175.00), (30, 200.00), (31, 225.00), (32, 250.00), (33, 275.00);
349 /* Filas afectadas: 10 Filas encontradas: 0 Advertencias: 0 Duración para 1 consulta: 0,016 seg. */
350 SELECT * FROM prueba;
351 /* Error de SQL (1146): Table 'pixel_security_360.prueba' doesn't exist */
352 /* Filas afectadas: 0 Filas encontradas: 0 Advertencias: 0 Duración para 0 de 1 consulta: 0,000 seg. */
353 SHOW DATABASES;
354 /* Entrando a la sesión "Unamed" */
355 SELECT `DEFAULT_COLLATION_NAME` FROM `information_schema`.`SCHEMATA` WHERE `SCHEMA_NAME`='pixel_security_360';
356 SHOW TABLE STATUS FROM `pixel_security_360`;
357 SHOW FUNCTION STATUS WHERE `Db`='pixel_security_360';
358 SHOW PROCEDURE STATUS WHERE `Db`='pixel_security_360';
359 SHOW TRIGGERS FROM `pixel_security_360`;
360 SELECT *, EVENT_SCHEMA AS `Db`, EVENT_NAME AS `Name` FROM information_schema.`EVENTS` WHERE `EVENT_SCHEMA`='pixel_security_360';
361 SELECT * FROM prueba;
362 /* Filas afectadas: 0 Filas encontradas: 0 Advertencias: 0 Duración para 1 consulta: 0,000 seg. */
363 INSERT INTO pruebas (referencia, nombre, descripcion, usuario_id) VALUES ('REF001', 'Prueba de vulnerabilidad de red', 'Escaneo de puertos y detección de vulnerabilidades en la red.', 1), ('REF002', 'Prueba de seguridad web', 'Análisis de seguridad de una aplicación web.', 2);
364 /* Error de SQL (1452): Cannot add or update a child row: a foreign key constraint fails ('pixel_security_360`.`pruebas`, CONSTRAINT `pruebas_ibfk_1` FOREIGN KEY (`usuario_id`) REFERENCES `usuario` (`usuario_id`)) */
365 /* Filas afectadas: 0 Filas encontradas: 0 Advertencias: 0 Duración para 0 de 1 consulta: 0,000 seg. */
366 INSERT INTO pruebas (referencia, nombre, descripcion, estado, usuario_id) VALUES ('REF001', 'Prueba de vulnerabilidad de red', 'Escaneo de puertos y detección de vulnerabilidades en la red.', 'activo', 24), ('REF002', 'Prueba de seguridad web', 'Análisis de seguridad de una aplicación web.', 'activo', 25);
367 /* Filas afectadas: 10 Filas encontradas: 0 Advertencias: 0 Duración para 1 consulta: 0,016 seg. */
368 INSERT INTO reporte (usuario_id, prueba_id, resultado) VALUES (26, 1, 'Se encontraron múltiples vulnerabilidades en la red corporativa.'), (25, 2, 'La aplicación web presenta vulnerabilidades de inyección de SQL.'), (26, 3, 'La aplicación web presenta vulnerabilidades de inyección de SQL.'), (26, 4, 'Se encontraron vulnerabilidades de seguridad en la red corporativa.'), (26, 5, 'Se encontraron vulnerabilidades de seguridad en la red corporativa.'), (26, 6, 'Se encontraron vulnerabilidades de seguridad en la red corporativa.'), (26, 7, 'Se encontraron vulnerabilidades de seguridad en la red corporativa.'), (26, 8, 'Se encontraron vulnerabilidades de seguridad en la red corporativa.'), (26, 9, 'Se encontraron vulnerabilidades de seguridad en la red corporativa.'), (26, 10, 'Se encontraron vulnerabilidades de seguridad en la red corporativa.').
369 /* Error de SQL (1452): Cannot add or update a child row: a foreign key constraint fails ('pixel_security_360`.`reporte`, CONSTRAINT `reporte_ibfk_2` FOREIGN KEY (`prueba_id`) REFERENCES `pruebas` (`prueba_id`)) */
370 /* Filas afectadas: 0 Filas encontradas: 0 Advertencias: 0 Duración para 0 de 1 consulta: 0,000 seg. */
371 SELECT * FROM pruebas;
372 /* Filas afectadas: 0 Filas encontradas: 10 Advertencias: 0 Duración para 1 consulta: 0,016 seg. */
373 INSERT INTO reporte (usuario_id, prueba_id, resultado) VALUES (24, 11, 'Se encontraron múltiples vulnerabilidades en la red corporativa.'), (25, 12, 'La aplicación web presenta vulnerabilidades de inyección de SQL.'), (26, 13, 'La aplicación web presenta vulnerabilidades de inyección de SQL.'), (26, 14, 'Se encontraron vulnerabilidades de seguridad en la red corporativa.'), (26, 15, 'Se encontraron vulnerabilidades de seguridad en la red corporativa.'), (26, 16, 'Se encontraron vulnerabilidades de seguridad en la red corporativa.'), (26, 17, 'Se encontraron vulnerabilidades de seguridad en la red corporativa.'), (26, 18, 'Se encontraron vulnerabilidades de seguridad en la red corporativa.'), (26, 19, 'Se encontraron vulnerabilidades de seguridad en la red corporativa.'), (26, 20, 'Se encontraron vulnerabilidades de seguridad en la red corporativa.').
374 /* Filas afectadas: 10 Filas encontradas: 0 Advertencias: 0 Duración para 1 consulta: 0,015 seg. */
375 SELECT * FROM usuario;

/* Filas afectadas: 0 Filas encontradas: 10 Advertencias: 0 Duración para 1 consulta: 0,000 seg. */
SELECT * from autenticacion;
/* Error de SQL (1146): Table 'pixel_security_360.autenticacion' doesn't exist */
/* Filas afectadas: 0 Filas encontradas: 0 Advertencias: 0 Duración para 0 de 1 consulta: 0,000 seg. */
SELECT * from autenticacion;
/* Filas afectadas: 0 Filas encontradas: 10 Advertencias: 0 Duración para 1 consulta: 0,000 seg. */
SELECT * from pago;
/* Filas afectadas: 0 Filas encontradas: 10 Advertencias: 0 Duración para 1 consulta: 0,000 seg. */
```

## Lo aprendido:

Aprendimos a que debemos analizar mas antes de ir de una vez a programar ya que con el análisis o realizar el diagrama podemos ahorrarnos mucho mas tiempo.