Índice General

Índice de figuras

[Figura 1 10](#_Toc209796619)

[Figura 2 10](#_Toc209796620)

[Figura 3 11](#_Toc209796621)

[Figura 4 13](#_Toc209796622)

[Figura 5 14](#_Toc209796623)

[Figura 6 15](#_Toc209796624)

[Figura 7 16](#_Toc209796625)

[Figura 8 16](#_Toc209796626)

[Figura 9 17](#_Toc209796627)

[Figura 10 18](#_Toc209796628)

[Figura 11 19](#_Toc209796629)

[Figura 12 20](#_Toc209796630)

[Figura 14 21](#_Toc209796631)

Desarrollo

# POPCODE

## Método propuesto

## CASCADA

La metodología cascada cuenta con 5 faces que son fundamentales para el desarrollo del sistema web dividiendo el trabajo en fases secuenciales, en la que cada fase debe de completarse en su totalidad antes de que se pueda avanzar a lo siguiente.

## Fase 1

## Análisis

* Herramienta de recolección de información.
* Generar la propuesta.
* Definir los requisitos funcionales del sistema.
* Analizar el lector de códigos como protocolos soportados, configuración, documentación técnica.
* Establecer el flujo de trabajo en la línea de producción como el escaneo, Validación, Registro.
* Determinar roles y permisos (administrador, operador, supervisor).
* Analizar la infraestructura de red y servidor.
* Identificar riesgos y limitaciones.

## Fase 2

## Diseño

1. Crear diagramas de caso de uso
2. Diagramas secuenciales
3. Arquitectura cliente-servidor
4. Backend en PHP (Diagrama E-R)
5. Diagrama relacional
6. Frontend de la ventana de registro de usuarios
7. Frontend para el login
8. Frontend para el panel principal del administrador
9. Frontend para la ventana escanear del administrador
10. Frontend para seleccionar el producto que se escaneara e indicar el número de Lote del administrador
11. Frontend para el botón escanear del administrador
12. *Frontend para el botón registros del administrador*
13. *Frontend para descargar el reporte de los productos*
14. *Frontend para el registro de productos del administrador*
15. Frontend para los productos registrados del administrador
16. Frontend para la ventana inventarios del administrador
17. Frontend para el submenú de la ventana de inventarios del administrador
18. Frontend para el botón de registrar productos activos de la ventana inventarios del administrador
19. Frontend para el botón de registrar productos equipos de cómputo de la ventana de inventarios del administrador
20. Frontend para el botón de registrar productos CISCO de la ventana de inventarios del administrador.
21. Frontend para la ventana de usuarios registrados del administrador
22. Frontend para la ventana de generar reportes de cada inventario
23. Frontend para el panel principal del supervisor
24. Frontend para la ventana escanear del supervisor
25. Frontend para el botón escanear del supervisor
26. Frontend para el submenú del botón escanear del supervisor
27. Frontend para escanear productos del botón escanear del supervisor
28. Frontend para la lista de productos en la ventana escanear y generar reporte del supervisor
29. frontend para la ventana de productos registrados del supervisor
30. frontend para el botón de activos de material TI y generar reportes para el supervisor
31. frontend para el botón de equipo de cómputo y generar reportes para el supervisor
32. Frontend para el botón de CISCO y generar reportes para el supervisor
33. Arquitectura del sistema

## Fase 3

## Implementación

* Elegir el SGBD a utilizar
* Codificación de la base de datos
* Configurar el lector Keyence para enviar datos al servidor.
* Programar el backend para recibir, validar y almacenar los datos.
* Programar el frontend para mostrar información al administrador.
* Configurar la base de datos en XAMPP.
* Integrar la lógica de activación de alarmas desde el lector.

## Fase 4

## Verificación

* Pruebas de la base de datos
* Pruebas unitarias de cada módulo (escaneo, validación, almacenamiento).
* Pruebas de integración del sistema completo
* Pruebas de rendimiento de la velocidad de respuesta, carga de datos.
* Pruebas de error
* Validación con usuarios

## Fase 5

## Mantenimiento

* Corrección de errores
* Hacer actualizaciones
* Manual de usuario
* Manual técnico

**Desarrollo**

## Fase 1

## Análisis

### Herramienta de recolección de información.

* Levantamiento de requerimientos

Para poder llevar a cabo el desarrollo del sistema web se necesitó hacer un levantamiento de requerimientos con el operador de la línea de producción de las palomitas SLIMPOP y con los gerentes de cada área para recabar la información fundamental de sus inventarios, revisando a si los manuales técnicos del lector de códigos de barras SICAR Checko2B que se consideran una opción a usar, verificando también su compatibilidad y las especificaciones de los mismos.

### Generar la propuesta

Se propone el desarrollo de un sistema web que automatice el proceso de escaneo de productos en la línea de producción y a si mismo integrar este módulo de escaneo en el inventariado de cada área, el sistema recibirá los datos del lector de códigos de barras SICAR Checko2B, validará los códigos escaneados y registrará la información en una base de datos, el administrador podrá acceder a un panel de control donde visualizará y editar los campos del inventario en general, el registro de los usuarios, las actualizaciones y eliminación de los productos, así como también podrá hacer escaneos, reportes detallados por lote, fecha, hora, código del producto, descripción y estado, por otra parte el Supervisor solo tendrá accesos limitados a los inventarios, como el registro de producto, actualización, escaneo y a generar reportes, finalmente el operador solo escaneara productos, esta solución busca reducir errores, mejorar y optimizar el tiempo.

### Definir los requisitos funcionales del sistema.

El sistema estará contando con requisitos específicos que son fundamentales para el funcionamiento.

* Escaneo de códigos de barras desde el lector.
* Validación automática del código escaneado.
* Registro de productos válidos y errores en la base de datos.
* Gestión de inventarios por cada área.
* Generación de reportes por fecha, usuario y tipo de producto.
* Control de acceso por roles definidos.

### Analizar el lector de códigos como protocolos soportados, configuración.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Nombre del lector** | **Lenguaje** | **Función** | **Compatibilidad** | **Componentes** | **Librerías** | **Interfaces compatibles** |
| SICAR Checko2B | JavaScript + Node.js | El lector actúa como si fuera un teclado cada vez que escanea un código, lo “escribe” en el campo activo del navegador. | Si es adecuado para el sistema en JavaScript solo se necesita configurar en modo teclado para que funcione sin complicaciones | Frontend (JavaScript)  Backend (PHP con XAMPP)  Lector Checko2B  Entrada de datos vía teclado USB o Bluetooth | Vanilla JS o React.js para la interfaz  SweetAlert2 para alertas visuales de éxito/error  Axios o Fetch API para enviar datos al backend. | Interfaz serie RS-232 – Teclado USB – Interfaz serie virtual USB – Bluetooth. |

### Establecer el flujo de trabajo en la línea de producción como el escaneo, Validación, Registro, Visualización.

El flujo de trabajo del sistema será el siguiente:

* El producto pasa frente al lector de códigos de barras SICAR Checko2B.
* El lector escanea el código de barras del producto.
* El lector envía el código escaneado al sistema web, actuando como teclado (USB/Bluetooth).
* JavaScript en el frontend captura el código y lo envía al servidor mediante una solicitud HTTP.
* El backend (PHP) recibe el código y lo valida contra la base de datos MySQL.
* Si el código es válido lo registra como exitoso y continua su proceso, pero si el código es inválido emite una alerta sonora y visual, la cual el programa marca como erróneo.
* El sistema guarda el registro del escaneo.
* El administrador accede al panel web para consultar registros históricos filtrar por fechas, productos o estados y generar reportes.

### Determinar roles y permisos (administrador, operador).

El sistema contará con un Login que permitirá el acceso a tres tipos de usuarios uno será el administrados, Supervisor y el otro el administrador.

Administrador: Usuario con acceso completo al sistema, podrá consultar todos los registros, actualizar y eliminar, generando reportes por fecha y producto, configurar parámetros del sistema, y gestionar usuarios.

Supervisor: Usuario con acceso limitado al sistema, podrá consultar registros de sus áreas y actualizarlas, generar reportes por fecha y producto, escanear el producto.

Operador: Usuario encargado de supervisar el escaneo en la línea de producción, solo tendrá acceso a la interfaz web, donde podrá visualizar el estado de los escaneos en tiempo real, pero no podrá modificar ni consultar reportes históricos.

La autenticación se realizará mediante credenciales únicas, y se implementará control de acceso basado en roles para garantizar la seguridad de la información.

1. **Infraestructura de Red y Servidor**

El sistema funcionará en una red local (LAN) donde el lector SICAR Checko2B se conectará por USB o Bluetooth a una computadora o directamente al lector de escaneo, el servidor local estará configurado con XAMPP, el cual alojará tanto el sistema web como la base de datos, los dispositivos dentro de la red podrán acceder al sistema mediante la IP local asignada al servidor para garantizar mayor estabilidad y velocidad en la transmisión de datos, permitiendo el acceso multiusuario con roles definidos, lo que facilitará la gestión de permisos y tareas dentro del sistema, la infraestructura está preparada para escalar a un entorno en la nube sin afectar la operatividad del sistema.

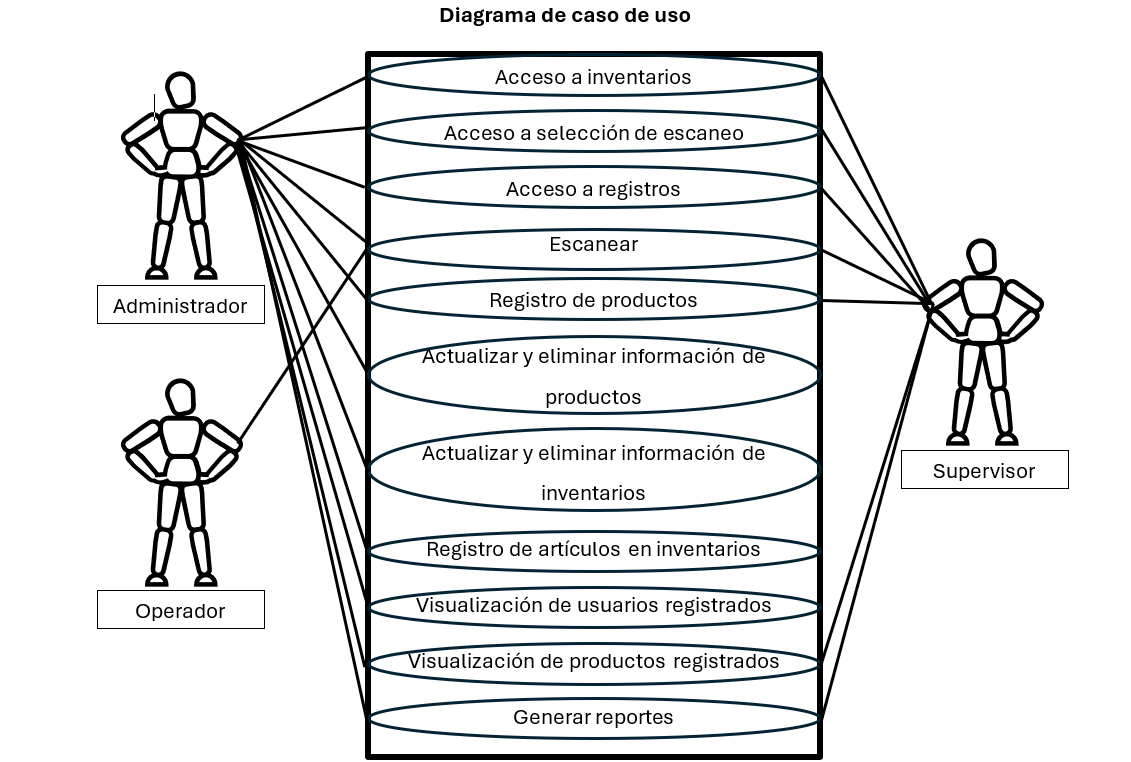
## Fase 2

## Diseño

### Crear diagramas de caso de uso

****Figura**** 1

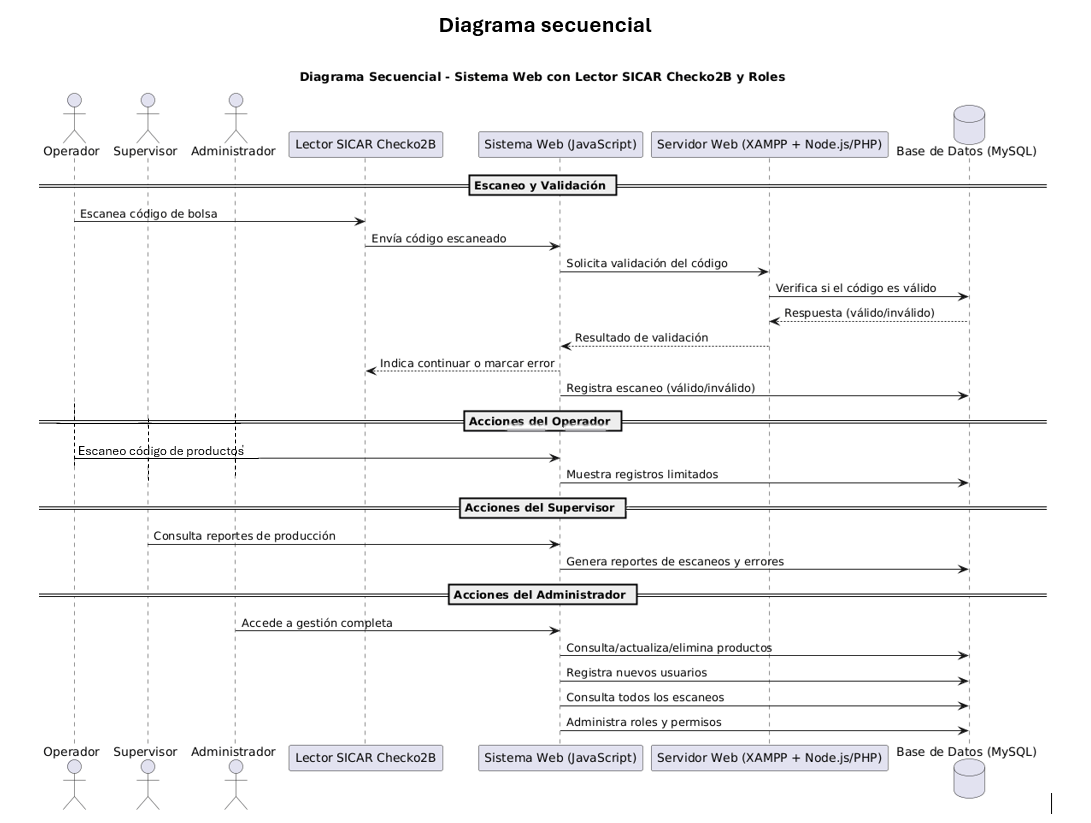
*Diagrama del caso de uso del sistema web POPCODE*



*Nota:* El diagrama representa una secuencia de lo que es la funcionalidad principal del sistema desde la perspectiva del usuario por cada rol, capturando los requisitos del sistema, y facilitar la comunicación.

### Diagramas secuenciales

****Figura 2****

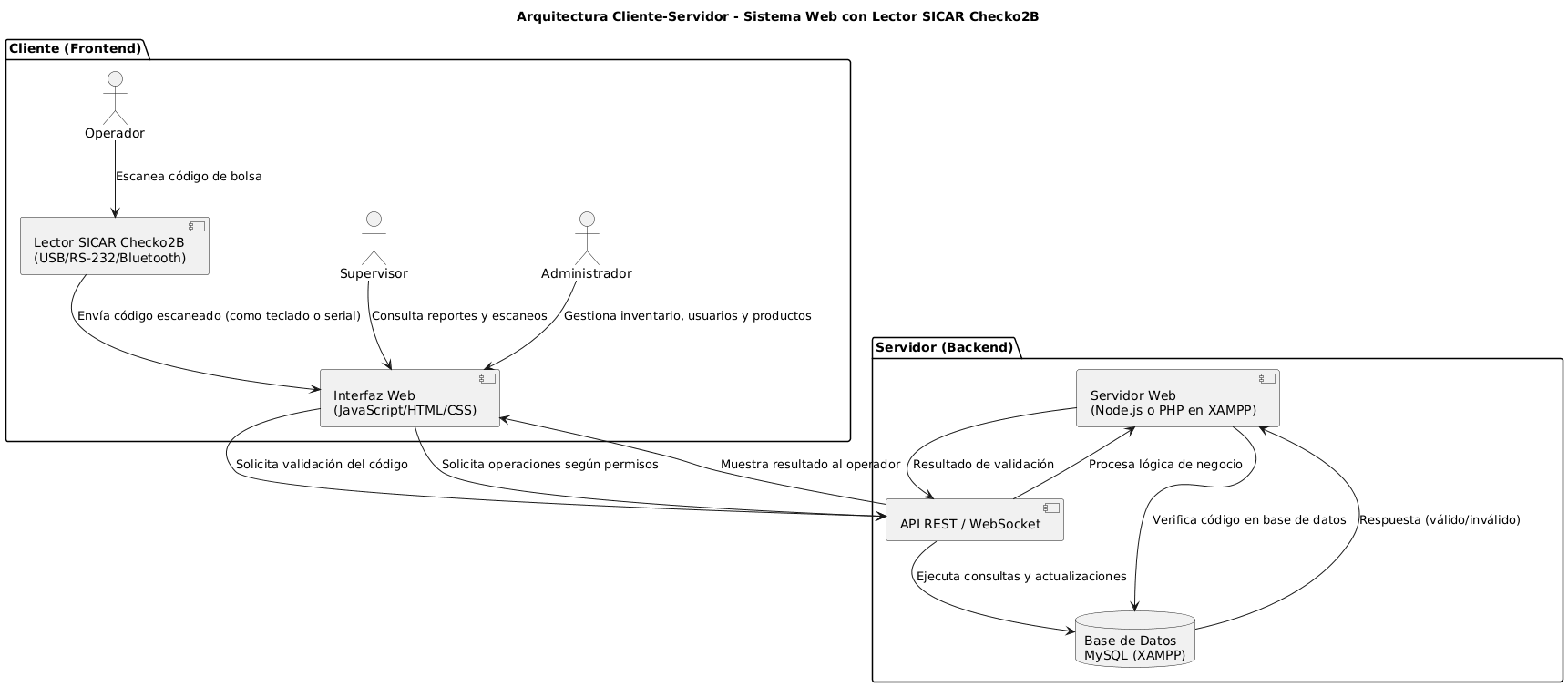
*Diagrama secuencial del sistema web POPCODE*

*Nota:* El diagrama secuencial hace una representación de como es que se lleva el proceso del sistema y como interactúa, visualiza y documenta la interacción ordenada de objetos a lo largo del tiempo, mostrando cómo los mensajes son intercambiados entre ellos para cumplir con un caso de uso específico.

### Arquitectura cliente-servidor

****Figura 3****

*Arquitectura de cliente-servidor del sistema web POPCODE*

*Nota:* La arquitectura del cliente-servidor se divide y organiza el sistema en responsabilidades que le corresponden como al cliente que solicita información y los servidores la proporcionan y gestionan los datos de manera centralizada.

### Backend en PHP (Diagrama E-R)

Figura 4

Diseño del diagrama entidad relación correspondiente a los campos de la Base de Datos.

*Nota:* El diagrama de entidad relación permite ver de manera organizada y estructurada la información y los campos a utilizar en la base de datos como las relaciones, los atributos y las claves primarias como foránea.

### Diagrama relacional

Figura 5

*Diagrama relacional correspondiente las tablas a utilizar.*

*Nota:* El diagrama relacional permite visualizar las tablas de manera organizada y conectada entre sus claves correspondientes.

### Frontend de la ventana de registro de usuarios

Figura 6

*Interfaz grafica de la ventana de registro de usuarios por roles.*

### Componentes de la ventana:

* Campo de texto “Numero del colaborador”: Campo para introducir el número de colaborador y almacenarse en la BD como numero\_colaborador.
* Campo de texto “Nombre del colaborador”: Campo para introducir el nombre del colaborador almacenándose en BD como nombre\_colaborador.
* ComboBox de “Rol”: Para la selección del rol del colaborador entre Administrador, Supervisor y Operador.
* ComboBox de “Área”: Para la selección de área del colaborador.
* Campo de texto “Contraseña”: Campo para introducir una contraseña que se asignara para el usuario.
* Botón “Registrar”: Registra al usuario.

Imagen que contiene Logotipo

El contenido generado por IA puede ser incorrecto.

**Registrarse**

**Número del colaborador**

**Nombre del colaborador**

**Contraseña**

**Registrar**

**Rol**

**Área**

*Nota:* Se visualiza la interfaz que tendría el sistema para registrarse por cada área.

### Frontend para el login

Figura 7

*Interfaz gráfica de la ventana del login para acceder al sistema.*

### Componentes de la ventana:

* Campo texto “Usuario”: Campo para introducir el nombre del colaborador.
* Campo de texto “Contraseña”: Campo para introducir la contraseña del usuario y que en la base de datos la almacene de manera [hasheada](https://www.bing.com/ck/a?!&&p=3f6755b9a0d7698690a66309176c5ad695fe8202fe3c120254f6f43831d8bf99JmltdHM9MTc1ODY3MjAwMA&ptn=3&ver=2&hsh=4&fclid=172ace2f-8723-6f0d-348f-d84a86e26e67&psq=contrase%c3%b1a+hasheada&u=a1aHR0cHM6Ly93d3cuZGFzaGxhbmUuY29tL2VzL2Jsb2cvcXVlLWVzLWVsLWhhc2gtZGUtY29udHJhc2VuYXM).
* Botón “Iniciar Sesión”: Botón que te va a permitir ingresar al sistema.



**Registrar nuevo usuario**

**Iniciar Sesión**

**Usuario**

**Contraseña**

**Iniciar Sesión**

*Nota:* Se visualiza la interfaz que tendrá el sistema a la hora de iniciar sesión.

### Frontend para el panel principal del administrador

Figura 8

*La interfaz grafica del panel principal del administrador.*

### Componentes de la ventana:

* Botón “Escanear”: Acceder a módulo de escaneo.
* Botón “Inventarios”: Acceso a los inventarios con los que cuenta el administrador de esa área.
* Botón “Reportes”: Acceso a panel de selección del reporte que desea descargar ya sea de sus inventarios o de sus escaneos.
* Botón “Cerrar sesión”: Cierra sesión
* Botón “Usuarios”: Visualización de los usuarios que están registrados en el sistema.



*Nota:* Se visualiza la interfaz principal que tendrá el sistema a la hora de iniciar sesión.

### Frontend para la ventana escanear del administrador

Figura 9

*La interfaz gráfica de la ventana escanear del administrador.*

### Componentes de la ventana:

* Botón “Escaneo”: Botón para ingresar a el apartado de escaneo.
* Botón “Registrar”: Botón para registrar nuevos productos para su próximo escaneo.
* Botón “Registrados”: Botón para acceder al apartado de productos registrados.
* Botón “Cerrar sesión”: Botón para cerrar la sesión.



**¿Qué deseas realizar?**

**¡Bienvenido, \_\_\_\_\_\_\_\_\_!**

**Cerrar Sesión**

Registrados

Escaneo

Registrar

*Nota:* En esta interfaz se estará mostrando una siguiente ventana que permite desplazarse por las siguientes interfaces de escaneo, registro y productos registrados.

### Frontend para seleccionar el producto que se escaneara e indicar el número de Lote del administrador

Figura 10

*La interfaz gráfica de la ventana que permite seleccionar el producto a escanear y el número de lote.*

### Componentes de la ventana:

* ComboBox “Producto a escanear”: Selección de producto que se escaneara.
* Campo de texto “No. Lote”: Ingreso del número del lote que se escaneara
* Botón “Cancelar”: Botón para cancelar la operación y volver a la sección anterior.
* Botón “Continuar”: Botón para continuar el proceso.

**Forma, Rectángulo

El contenido generado por IA puede ser incorrecto.**

**¿Qué producto escanearas hoy?**

Producto a escanear

Cancelar

Continuar

No. Lote

Palomitas XXXX 16G

Palomitas XXXX 16G

*Nota:* En esta interfaz permite que el administrador seleccione el producto que desea escanear indicando el numero de lote que tendrán los productos escaneados.

### Frontend para el botón escanear del administrador

Figura 11

*La interfaz gráfica del botón escanear donde se muestras los datos del producto escaneado.*

### Componentes de la ventana:

* Campo de texto “Código de barras”:
* Campo de texto “Descripción”:
* Botón “Regresar”: Botón para regresar al panel anterior.
* Botón “Registros”: Botón para acceder a los registros de escaneos.

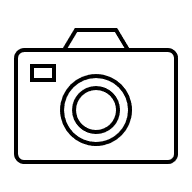
**Imagen que contiene Patrón de fondo

El contenido generado por IA puede ser incorrecto.**

Código de barras:

Descripción:

Imagen:

****

Regresar

Registros

*Nota:* En esta interfaz se mostrará los datos ya escaneados según su correspondencia.

### Frontend para el botón registros del administrador

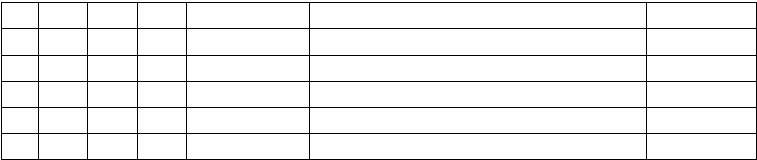
Figura 12

*La interfaz gráfica del botón que muestra la anterior ventana de registros*

### Componentes de la ventana:

* Tabla de “Registro de escaneos”: Tabla donde se almacenan todos los escaneos conforme a ID, Lote, Fecha, Hora, Código de barras, Descripción y Estado.
* Botón “Generar reporte”: Acceso a el apartado de preferencias del reporte.
* Botón “Regresar”: Retorno al panel anterior.

Regresar



Generar Reporte

**Registro de Escaneos**

**ID**

**Lote**

**Fecha**

**Hora**

**Código barras**

**Descripción**

**Estado**

*Nota:* La interfaz que va a permitir generar los reportes de los productos que ya están escaneados.

### Frontend para descargar el reporte de los productos

Figura 13

*La interfaz gráfica muestra la selección de descarga del reporte por fecha.*

### Componentes de la ventana:

* Campo de texto “Fecha inicio”: Fecha inicial para conteo de fechas.
* Campo de texto “Fecha final”: Fecha final para conteo de fechas.
* Checkbox “Con Lote”: Para incluir los lotes en el reporte.
* Checkbox “Con Horas”: Para incluir la hora en el reporte.
* Botón “Generar”: Para la generación del reporte

**Imagen que contiene Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.**

Cancelar

Generar

**Orden**

Fecha: Desde Hasta

Con Lote Con Hora

**Preferencias de reporte**

### Frontend para la ventana inventarios del administrador

Figura 14

### Frontend para el submenú de la ventana de inventarios del administrador

### Frontend para el botón de registrar productos activos de la ventana inventarios del administrador

### Frontend para el botón de registrar productos equipos de cómputo de la ventana de inventarios del administrador

### Frontend para el botón de registrar productos CISCO de la ventana de inventarios del administrador.

### Frontend para la ventana de usuarios registrados del administrador

### Frontend para la ventana de generar reportes de cada inventario

### Frontend para el panel principal del supervisor

### Frontend para la ventana escanear del supervisor

### Frontend para el botón escanear del supervisor

### Frontend para el submenú del botón escanear del supervisor

### Frontend para escanear productos del botón escanear del supervisor

### Frontend para la lista de productos en la ventana escanear y generar reporte del supervisor

### frontend para la ventana de productos registrados del supervisor

### frontend para el botón de activos de material TI y generar reportes para el supervisor

### frontend para el botón de equipo de cómputo y generar reportes para el supervisor

### Frontend para el botón de CISCO y generar reportes para el supervisor

### Arquitectura del sistema

## Fase 3

## Implementación

### Elegir el SGBD a utilizar

Para el desarrollo del sistema se utilizará XAMPP, este siendo una distribución de código abierto que permite instalar fácilmente un servidor web, este paquete incluye componentes esenciales como Apache, MySQL y PHP, lo que lo convierte en una herramienta gratuita ideal para el desarrollo y prueba de aplicaciones web en sistemas operativos como Windows, Linux y macOS, además, incorpora phpMyAdmin, una interfaz gráfica que facilita la gestión de bases de datos de manera sencilla y eficiente.

### Codificación de la base de datos

Create database PopCode;

Create table usuarios (

id\_usuario INT Primary key not null,

nombre\_usuario varchar(45),

contraseña varchar(16),

rol varchar(20)

);

Create table productos (

codigo\_barras INT Primary key not null,

descripcion varchar(45) unique,

imagen varchar(255)

);

Create table productos\_escaneados (

id\_escaneo int primary key not null,

fecha\_escaneo date,

hora\_escaneo time,

codigo\_barras int not null,

descripcion\_producto varchar(45),

estado varchar(45),

constraint fk\_codigo\_barras

foreign key (codigo\_barras)

references productos(codigo\_barras),

constraint fk\_descripcion\_producto

foreign key (descripcion\_producto)

references productos(descripcion)

);

Insert into usuarios (id\_usuario, nombre\_usuario, contraseña, rol) values (1, ‘Admin1’, ‘admin1p’, ‘Administrador’), (2, ‘Oper1’, ‘Oper1p’, ‘Operador’);

### Configurar el lector X para enviar datos al servidor.

### Programar el backend para recibir, validar y almacenar los datos.

### Programar el frontend para mostrar información al administrador.

### Configurar la base de datos en XAMPP.

### Integrar la lógica de activación de alarmas desde el lector.

## Fase 4

## Verificación

### Pruebas de la base de datos

### Pruebas unitarias de cada módulo (escaneo, validación, almacenamiento).

### Pruebas de integración del sistema completo

### Pruebas de rendimiento de la velocidad de respuesta, carga de datos.

### Pruebas de error

### Validación con usuarios

## Fase 5

## Mantenimiento

### Corrección de errores

### Hacer actualizaciones

### Manual de usuario

### Manual técnico

Cronograma de actividades