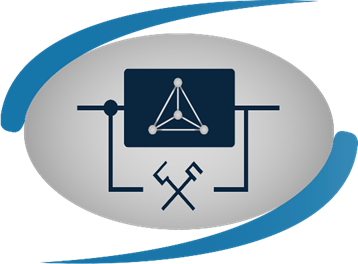
UNIVERSIDAD TECNICA DE ORURO

FACULTAD NACIONAL DE INGENIERÍA

INGENIERÍA INFORMÁTICA



PROYECTO SIS – 2420 “ACTUALIZACIÓN TECNOLÓGICA”



**SISTEMA DE INFORMACIÓN PARA**

**VENTAS DE COMPUTADORAS Y PORTÁTILES**



**Docente:** Ing. Saul Mamani Mamani

**Auxiliar:** Egr. Achabal Villalpando William Mucio

**Estudiante:** Univ**.** Lucano cayo Daniel

Univ**.** Rios Zegarra Axel Santiago

Univ**.** Copa Siles Brayan Ruben

Univ**.** Ahern Quiñones Kevin Willams

Univ**.** Beltrán Canaza Alan Joseph

Oruro – Bolivia

2023

**INDICE**

1. INTRODUCCIÓN 1
   1. Presentación 1
   2. Planteamiento del problema 1
   3. Objetivos 2
      1. Objetivo general 2
      2. Objetivos específicos 2
   4. Alcances 2
   5. Ingeniería del proyecto 3
2. MARCO TEÓRICO 4
   1. Algoritmo 4
   2. Software 4
   3. Sistema de información 4
   4. Scrum 4
   5. Modelado 4
   6. UML: Lenguaje de Modelado Unificado 4
      1. Diagrama de casos de uso 4
      2. Diagrama de clases 4
      3. Diagrama de secuencias 4
   7. .Net Framework 4
   8. C# 4
   9. SQL Server 4
   10. Enterprise Architect 4

2.12 Balsamiq 4

1. MARCO PROPOSITIVO 5
   1. Identificación del sistema 5
   2. Equipo Scrum y los Stakeholders 5
   3. Historias de Usuario 5
   4. Producto Backlog 7
   5. Release Planning 7
   6. Diagrama de casos de uso del sistema 8
   7. Diagrama de clases persistentes 9
   8. Modelo Relacional de la Base de Datos 9
   9. Diagrama de secuencias 10
   10. Primer Sprint 11
       1. Sprint backlog 11
       2. Diseño de interfaces 11
       3. Sprint review 12
       4. Sprint retrospective 13
   11. Segundo Sprint 14
       1. Sprint backlog 14
       2. Diseño de interfaces 14
       3. Sprint review 14
       4. Sprint retrospective 14
   12. Tercer Sprint 15
       1. Sprint backlog 15
       2. Diseño de interfaces 15
       3. Sprint review 15
       4. Sprint retrospective 15
2. CONCLUSIONES Y RECOMENDACIONES 16

BIBLIOGRAFÍA 17

ANEXOS 18

# 1 INTRODUCCIÓN

## 1.1 Presentación

**//Describir acerca de su sistema, en las condiciones actuales en la que está…¿por qué están haciendo el proyecto y en donde se está realizando?**

## El proyecto consiste en desarrollar un sistema de información web para Laptomanía Oruro, una empresa dedicada a la venta de laptops de marcas reconocidas como Asus, Dell y HP. La necesidad surge de optimizar la administración de equipos, el registro de ventas y la emisión de recibos. Laptomanía Oruro, con más de 8 años en el mercado de Oruro, busca modernizar sus operaciones y mejorar la experiencia tanto para el equipo administrativo como para sus clientes.

## El sistema se está desarrollando para abordar las siguientes necesidades identificadas en una reunión entre el Product Owner y el Gerente General de la empresa:

## Inventario detallado de equipos disponibles para la venta.

## Registro de ventas con información del cliente y fecha de la transacción.

## Generación de recibos al concluir una venta, con un diseño que incluye un código QR.

## Registro y administración de equipos a cargo de un usuario administrador.

## Generación de informes mensuales detallados de todas las ventas y montos recaudados.

## Reportes que enumeren los equipos vendidos por marca.

## Acceso público para que los clientes vean los productos disponibles en la aplicación web.

## 1.2 Planteamiento del problema

**//Se requiere describir los problemas actuales que enfrenta el negocio y la necesidad de implementar un sistema de información**

El negocio actualmente enfrenta desafíos en la gestión manual de inventario y ventas, lo que lleva a posibles errores y a una falta de eficiencia. La emisión de recibos también se realiza de manera manual, lo que puede resultar en demoras y posibles confusiones. La falta de un sistema centralizado dificulta la generación de informes y análisis efectivos. Por lo tanto, se busca implementar un sistema de información que automatice y mejore estos procesos.

## 1.3 Objetivos

### 1.3.1 Objetivo general

**//Modificar de acuerdo a su proyecto**

Implementar un sistema de información web para Laptomanía Oruro que optimice la gestión de inventario, el registro de ventas y la emisión de recibos, mejorando así la eficiencia y la experiencia del cliente.

### 1.3.2 Objetivos específicos

**//Pueden mejorar los objetivos específicos depende de ustedes**

* Desarrollar un módulo de inventario que detalle todas las características importantes de los equipos disponibles para la venta.
* Crear un sistema de registro de ventas que capture los datos del cliente y la fecha de la transacción.
* Implementar una funcionalidad para la generación automática de recibos al concretar una venta, con un diseño que incluya un código QR.
* Facilitar el proceso de registro de ventas, asignando esta responsabilidad a la secretaria de la empresa.
* Establecer un usuario administrador con acceso total al sistema para gestionar tanto los equipos como las funcionalidades del sistema.
* Permitir que la secretaria genere reportes mensuales detallados de todas las ventas y el monto recaudado.
* Generar un informe que liste las laptops disponibles y vendidas, permitiendo filtrar por marca.
* Desarrollar una interfaz web para que los clientes puedan visualizar los equipos disponibles para la venta.

## 1.4 Alcances

**//Hacer una lista resumida de todo lo que se espera que haga el sistema**

* Desarrollo de un inventario detallado de equipos disponibles para la venta.
* Registro automatizado de ventas con información de clientes y fechas.
* Generación automática de recibos con diseño que incluye un código QR.
* Creación de un usuario administrador para la gestión eficiente de equipos.
* Capacidad para que la secretaria genere informes mensuales detallados.
* Generación de informes que enumeren los equipos vendidos por marca.
* Acceso público a través de la aplicación web para que los clientes vean los productos disponibles.

## 1.5 Ingeniería del proyecto

**//Pueden mejorarlo depende de ustedes**

Para el desarrollo del proyecto se toman en cuenta los tres pilares de la ingeniería de software.

(Triángulo de éxito de la ingeniería de software)

1. **Metodología o proceso de desarrollo**

o Marco de trabajo ágil **SCRUM**, como proceso de desarrollocomo metodología principal. SCRUM permite una gestión adaptativa, iterativa e incremental del proyecto, facilitando la entrega continua y la adaptación a cambios en los requisitos

1. **Notación de Modelado**

o Lenguaje de Modelado Unificado (UML), proporciona una representación gráfica estandarizada, permitiendo una comprensión clara y visual del diseño del sistema mediante diagramas de casos de uso, clases, secuencias y despliegue.

1. **Herramientas**

La elección de herramientas es crucial para la eficacia del proyecto, y se seleccionan cuidadosamente:

o Framework de desarrollo: .NET Framework

o Lenguaje de programación: C#

o Entorno de Desarrollo Integrado (IDE): Visual Studio

o Manejador de Base de Datos: SQL Server

o Enterprise Architect, para los diagramas UML

o Balsamiq, para el diseño de interfaces de usuario

o Trello para la administración de las historias de usuario

**2 MARCO TEÓRICO**

**// No debe pasar de las 5 hojas**

## 2.1 Algoritmo

Un algoritmo es una secuencia de instrucciones que se siguen para resolver un problema. Los algoritmos se pueden representar de diversas formas, como en texto, diagramas de flujo, pseudocódigo o lenguaje de programación.

Los algoritmos se pueden clasificar en función de su complejidad, su eficiencia o su ámbito de aplicación.

* Por complejidad: Los algoritmos se pueden clasificar como deterministas o probabilísticos. Los algoritmos deterministas siempre producen el mismo resultado para los mismos datos de entrada. Los algoritmos probabilísticos, por otro lado, pueden producir resultados diferentes para los mismos datos de entrada.
* Por eficiencia: Los algoritmos se pueden clasificar como eficientes o ineficientes. Los algoritmos eficientes son aquellos que requieren una cantidad mínima de recursos para ejecutarse. Los algoritmos ineficientes, por otro lado, requieren una cantidad excesiva de recursos para ejecutarse.
* Por ámbito de aplicación: Los algoritmos se pueden clasificar como generales o específicos. Los algoritmos generales son aquellos que se pueden aplicar a una amplia gama de problemas. Los algoritmos específicos, por otro lado, están diseñados para resolver un problema concreto.

## 2.2 Software

El software es el conjunto de instrucciones que controlan el funcionamiento de un sistema informático. El software se puede clasificar en función de su propósito, su entorno de ejecución o su licencia.

* Por propósito: El software se puede clasificar como software de sistemas o software de aplicación. El software de sistemas se encarga de gestionar los recursos del sistema informático, como la memoria, el procesador y el almacenamiento. El software de aplicación se utiliza para realizar tareas específicas, como escribir documentos, crear presentaciones o jugar a videojuegos.
* Por entorno de ejecución: El software se puede clasificar como software de escritorio, software de servidor o software móvil. El software de escritorio se ejecuta en ordenadores personales. El software de servidor se ejecuta en servidores. El software móvil se ejecuta en dispositivos móviles, como teléfonos inteligentes y tabletas.
* Por licencia: El software se puede clasificar como software propietario o software libre. El software propietario es aquel que es propiedad de una empresa o persona individual. El software libre es aquel que está disponible para su uso, modificación y redistribución sin restricciones.

## 2.3 Sistema de información

Un sistema de información es un conjunto de componentes que interactúan entre sí para recopilar, procesar, almacenar y distribuir información. Los sistemas de información se utilizan en una amplia gama de aplicaciones, como la gestión empresarial, la educación y la salud.

Los sistemas de información se pueden clasificar en función de su tamaño, su propósito o su ámbito de aplicación.

* Por tamaño: Los sistemas de información se pueden clasificar como grandes, medianos o pequeños. Los grandes sistemas de información se utilizan en organizaciones complejas, como las empresas multinacionales. Los medianos sistemas de información se utilizan en organizaciones de tamaño medio. Los pequeños sistemas de información se utilizan en organizaciones pequeñas.
* Por propósito: Los sistemas de información se pueden clasificar como transaccionales o analíticos. Los sistemas transaccionales se utilizan para realizar tareas repetitivas, como procesar pedidos o realizar pagos. Los sistemas analíticos se utilizan para analizar datos y tomar decisiones.
* Por ámbito de aplicación: Los sistemas de información se pueden clasificar como sistemas de gestión empresarial, sistemas de información geográfica o sistemas de apoyo a la toma de decisiones. Los sistemas de gestión empresarial (ERP) se utilizan para gestionar las operaciones de una empresa. Los sistemas de información geográfica (GIS) se utilizan para recopilar, almacenar y analizar datos geográficos. Los sistemas de apoyo a la toma de decisiones (DSS) se utilizan para ayudar a los usuarios a tomar decisiones informadas.

## 2.4 Scrum

Scrum es un marco de trabajo ágil para el desarrollo de software. Se basa en la iteración y la retroalimentación, y utiliza un proceso de desarrollo iterativo y incremental.

Scrum se divide en tres roles principales:

* El Product Owner es responsable de definir el alcance del producto y priorizar las historias de usuario.
* El Scrum Master es responsable de facilitar el proceso Scrum y garantizar que el equipo se adhiera a los principios y valores de Scrum.
* El equipo Scrum es responsable de desarrollar el producto.

Scrum se divide en dos fases principales:

* La fase de planificación se centra en definir el alcance del sprint y crear el backlog del sprint.
* La fase de ejecución se centra en completar las tareas del backlog del sprint.

Scrum es un marco de trabajo flexible que puede adaptarse a una variedad de proyectos.

## 2.5 Modelado

El modelado es el proceso de representar un sistema de forma abstracta. Los modelos se utilizan para comprender, analizar y diseñar sistemas.

Los modelos se pueden representar de diversas maneras, incluyendo diagramas, ecuaciones y lenguajes formales.

En el desarrollo de software, los modelos se utilizan para representar el sistema de información, el sistema operativo y las aplicaciones.

Los modelos ayudan a los desarrolladores a comprender el sistema, identificar problemas potenciales y comunicar sus ideas a los demás.

## 2.6 UML: Lenguaje de Modelado Unificado

UML es un lenguaje de modelado visual para el desarrollo de software. Se utiliza para representar los sistemas de información, el sistema operativo y las aplicaciones.

UML proporciona una serie de diagramas para representar diferentes aspectos de un sistema. Los diagramas más comunes de UML son:

* Diagrama de casos de uso muestra las interacciones entre los usuarios y el sistema.
* Diagrama de clases muestra las clases y sus relaciones en un sistema.
* Diagrama de secuencias muestra la secuencia de eventos en un sistema.
* Diagrama de despliegue muestra la distribución de los componentes de un sistema.

### 2.6.1 Diagrama de casos de uso

Un diagrama de casos de uso es un diagrama de UML que representa las interacciones entre los usuarios de un sistema y el sistema. Los diagramas de casos de uso se utilizan para comprender los requisitos funcionales de un sistema.

Los elementos de un diagrama de casos de uso son:

* Casos de uso: Representan una tarea que un usuario puede realizar con el sistema.
* Actores: Representan a los usuarios o sistemas externos que interactúan con el sistema.
* Relaciones: Representan las interacciones entre los casos de uso y los actores.

### 2.6.2 Diagrama de clases

Un diagrama de clases es un diagrama de UML que representa las clases de un sistema y las relaciones entre ellas. Los diagramas de clases se utilizan para comprender la estructura estática de un sistema.

Los elementos de un diagrama de clases son:

* Clases: Representan un conjunto de objetos que comparten atributos, métodos y relaciones comunes.
* Atributos: Representan las características de una clase.
* Métodos: Representan las acciones que puede realizar una clase.
* Relaciones: Representan las conexiones entre las clases.

### 2.6.3 Diagrama de secuencias

Un diagrama de secuencias es un diagrama de UML que representa la interacción entre objetos durante un cierto período de tiempo. Los diagramas de secuencias se utilizan para comprender el comportamiento dinámico de un sistema.

Los elementos de un diagrama de secuencias son:

* Objetos: Representan los objetos que participan en la interacción.
* Líneas de vida: Representan la existencia de un objeto durante un período de tiempo.
* Mensajes: Representan las comunicaciones entre objetos.

**2.6.4 Diagrama de Despliegue**

Un diagrama de despliegue es un diagrama de UML que representa la distribución física de los componentes de un sistema. Los diagramas de despliegue se utilizan para comprender la arquitectura de un sistema.

Los elementos de un diagrama de despliegue son:

* Nodos: Representan los componentes físicos de un sistema.
* Conexiones: Representan las conexiones entre los nodos.
* Recursos: Representan los recursos que se utilizan en un sistema.

## 2.7 .Net Framework

.Net Framework es un marco de software de Microsoft que proporciona una plataforma para el desarrollo, la ejecución y el despliegue de aplicaciones .NET. .Net Framework consta de una biblioteca de clases, un conjunto de herramientas y un modelo de ejecución.

Biblioteca de clases

La biblioteca de clases de .Net Framework proporciona un conjunto de clases que se pueden utilizar para crear aplicaciones .NET. Estas clases proporcionan funciones para una amplia gama de tareas, incluyendo la entrada/salida, el procesamiento de datos y la comunicación.

La biblioteca de clases de .Net Framework está organizada en un conjunto de namespaces. Cada namespace contiene un conjunto de clases relacionadas.

Algunos de los namespaces más importantes de la biblioteca de clases de .Net Framework son:

* System: Este namespace proporciona clases fundamentales para el desarrollo de aplicaciones .NET, incluyendo tipos de datos, flujos de entrada/salida y excepciones.
* System.Collections: Este namespace proporciona clases para el almacenamiento y la manipulación de colecciones de datos.
* System.IO: Este namespace proporciona clases para la entrada/salida de datos.
* System.Net: Este namespace proporciona clases para la comunicación a través de redes.
* System.Drawing: Este namespace proporciona clases para la creación de gráficos y elementos visuales.

## 2.8 C#

C# (pronunciado "C sharp") es un lenguaje de programación desarrollado por Microsoft. Es un lenguaje orientado a objetos diseñado para la plataforma .NET y es ampliamente utilizado en el desarrollo de aplicaciones Windows y web.

## 2.9 SQL Server

SQL Server es un sistema de gestión de bases de datos relacional desarrollado por Microsoft. Se utiliza para almacenar y recuperar datos de manera eficiente y segura.

## 2.10 Enterprise Architect

Enterprise Architect es una herramienta de modelado UML que permite a los equipos de desarrollo crear, visualizar y compartir modelos de sistemas complejos.

**2.11** **Trello**

Trello es una herramienta de gestión de proyectos basada en tableros que utiliza tarjetas y listas para organizar tareas y colaborar en proyectos de manera eficiente.

## 2.12 Balsamiq

Balsamiq es una herramienta de diseño de interfaz de usuario que permite a los equipos crear prototipos rápidos y efectivos para aplicaciones y sitios web. Facilita la comunicación visual de ideas de diseño.

**3 MARCOPROPOSITIVO**

## 3.1 Identificación del sistema

**// Modificar de acuerdo a su proyecto**

Se identifican las entradas, el proceso y las salidas del sistema de la biblioteca

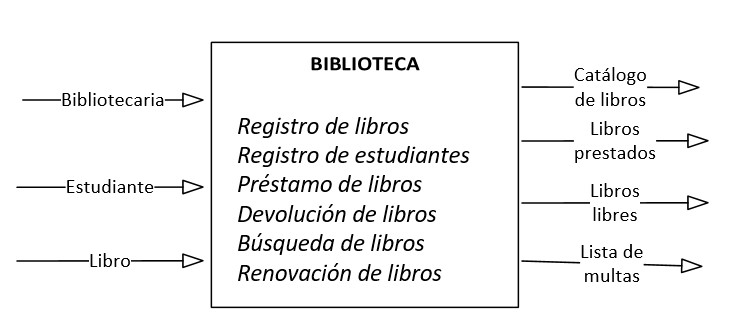


Figura 1: Sistema biblioteca

## 3.2 Equipo Scrum y los Stakeholders

**// Modificar de acuerdo a su proyecto**

Con fines académicos, los roles de Scrum están representados por una sola persona.

|  |  |
| --- | --- |
| **Product Owner** | Rios Zegarra Axel Santiago |
| **Scrum Master** | Lucano cayo Daniel |
| **Development Team** | Copa Siles Brayan Ruben  Ahern Quiñones Kevin Willams  Beltrán Canaza Alan Joseph |
| **Client** | Laptomanía Oruro es una empresa que se dedica a la venta de laptop de las siguientes marcas Asus, Dell, y HP |

Tabla 1: Equipo Scrum

## 3.3 Historias de Usuario

**// Modificar de acuerdo a su proyecto**

Para la determinación de requerimientos del sistema se recolectan historias de usuario.

|  |  |
| --- | --- |
| **HU1: Ingresar al sistema con inicio de sesión** | |
| **Como** | Usuario del sistema |
| **Quiero** | Ingresar al sistema con una cuenta y contraseña |
| **Para** | Tener acceso a las opciones de la aplicación web |

|  |  |
| --- | --- |
| **HU2: Gestionar (CRUD) Secretarias** | |
| **Como** | Administrador |
| **Quiero** | Crear, leer, editar y eliminar CRUD secretarios |
| **Para** | Tener acceso a todas las funcionalidades del sistema |

|  |  |
| --- | --- |
| **HU3: Gestionar (CRUD) Clientes** | |
| **Como** | Secretaria |
| **Quiero** | Crear, Leer, Editar y Eliminar (CRUD) nuevos clientes |
| **Para** | Que sean miembros de la tienda y puedan ver productos nuevos |

|  |  |
| --- | --- |
| **HU4: Gestionar (CRUD) Equipos** | |
| **Como** | Administrador |
| **Quiero** | Quiero tener todos los equipos inventariados en el sistema |
| **Para** | Para tener un mayor control sobre la mercadería |

|  |  |
| --- | --- |
| **HU5: Generar Reportes de Productos Disponibles** | |
| **Como** | Administrador |
| **Quiero** | Generar reportes en el sistema |
| **Para** | Saber mediante un listado de marcas las laptops vendidas y disponibles |

|  |  |
| --- | --- |
| **HU6: Generar Recibos** | |
| **Como** | Secretaria |
| **Quiero** | Agregar registros de venta |
| **Para** | Tener control del stock de los equipos |

|  |  |
| --- | --- |
| **HU7: Generar registros de venta** | |
| **Como** | Secretaria |
| **Quiero** | Agregar registros de venta |
| **Para** | Tener control del stock de los equipos |

|  |  |
| --- | --- |
| **HU8: Mostrar Ventana Productos** | |
| **Como** | Secretaria |
| **Quiero** | Ver los productos disponibles |
| **Para** | Poder dar una mejor información a los clientes |

|  |  |
| --- | --- |
| **HU9: Generar Reporte mensual** | |
| **Como** | Secretaria |
| **Quiero** | Generar un reporte mensual |
| **Para** | Tener un control del monto recaudado en el mes |

## 3.4 Producto Backlog

**// Modificar de acuerdo a su proyecto**

La pila del producto de pendientes a desarrollar está constituida por las historias de usuario (Requerimientos funcionales), y ordenada según prioridad de implementación.

|  |  |  |
| --- | --- | --- |
| **Historia de**  **usuario** | **Descripción** | **Prioridad** |
| **HU1 (Spike)** | **Captura de requisitos** | 0 |
| **HU2 (Spike)** | **Modelado del sistema** | 0 |
| **HU3 (Spike)** | **Configuración del proyecto Software** | 0 |
| **HU1** | **Ingresar al sistema con inicio de sesión** | 1 |
| **HU2** | **Gestionar (CRUD) secretarias** | 1 |
| **HU3** | **Gestionar (CRUD) Clientes** | 1 |
| **HU4** | **Gestionar (CRUD) Equipos** | 1 |
| **HU5** | **Generar reportes de Productos Disponibles** | 2 |
| **HU6** | **Generar Recibos** | 3 |
| **HU7** | **Generar Reportes Mensuales** | 2 |
| **HU8** | **Mostrar Ventana de productos** | 3 |
| **HU9** | **Generar Reportes Mensuales** | 2 |

Tabla 12: Product Backlog

## 3.5 Release Planning

**// Modificar de acuerdo a su proyecto**

En el plan de despliegue se identifican las iteraciones (**Sprints Backlog**) y los entregables que se van a realizar durante el transcurso del proyecto.

|  |  |  |
| --- | --- | --- |
| **Sprint / Entregable** | **HU** | **Descripción** |
| **Sprint cero** | **HU1 (spike)** | **Captura de Requisitos** |
| **HU2 (spike)** | **Modelado del Sistema** |
| **HU3 (spike)** | **Configuración del Proyecto Software** |
| **Primer Sprint** | **HU1** | **Ingresar al sistema con inicio de sesión** |
| **HU2** | **Gestionar (CRUD) Secretarias** |
| **HU3** | **Gestionar (CRUD) Clientes** |
| **HU4** | **Gestionar (CRUD) Equipos** |
| **Segundo Sprint** | **HU5** | **Generar Reportes de Productos Disponibles** |
| **HU7** | **Generar Registros de Venta** |
| **HU9** | **Generar Reportes Mensuales** |
| **Tercer Sprint** | **HU4** | **Mostrar Ventana de productos** |
| **HU10** | **Generar Recibos** |

Tabla 13: Sprint Backlog

## 3.6 Diagrama de casos de uso del sistema

**// Modificar de acuerdo a su proyecto**

El diagrama de casos de uso del sistema muestra la funcionalidad global del sistema.

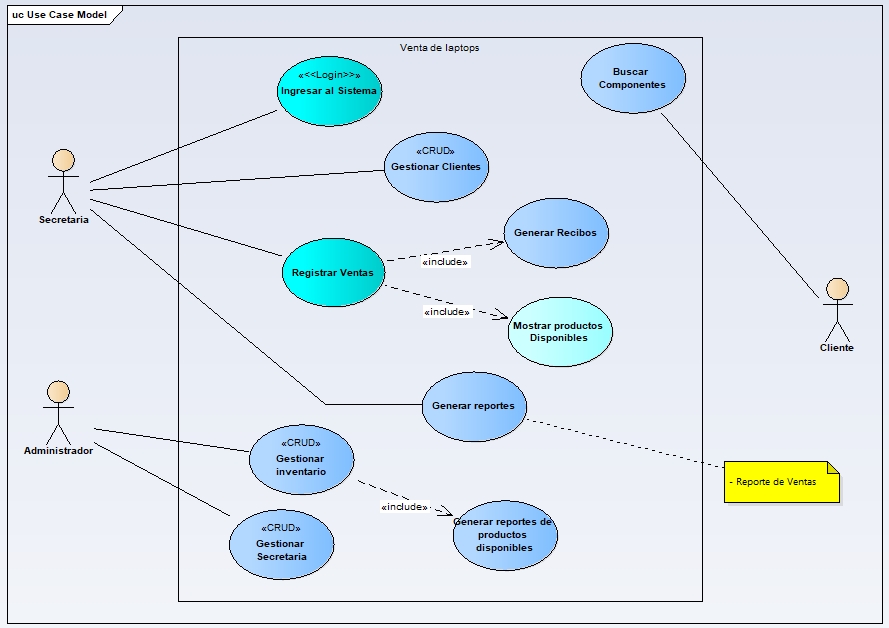


Figura 2: Diagrama de casos de uso

## 3.7 Diagrama de clases persistentes

**// Modificar de acuerdo a su proyecto**

El diagrama de clases persistentes muestra las clases que servirán para el diseño de la base de datos

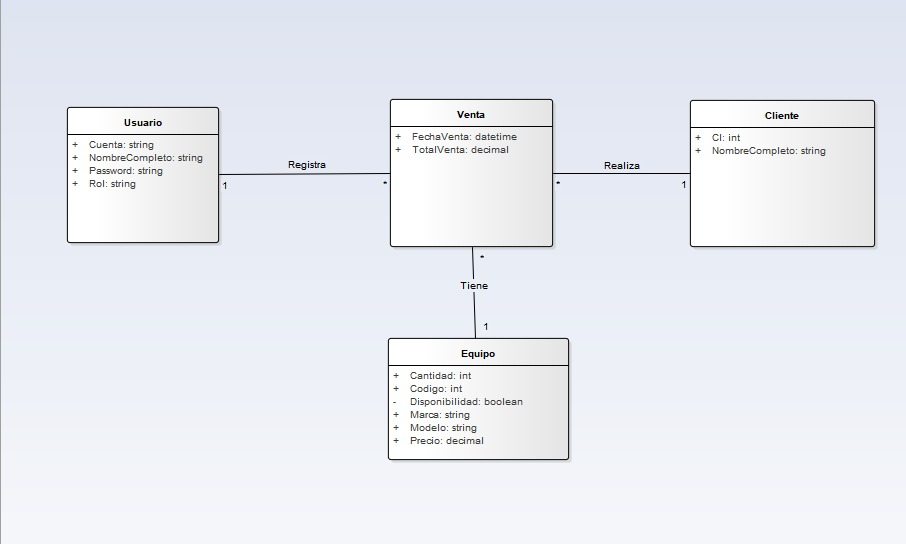


Figura 3: Diagrama de clases

## 3.8 Modelo Relacional de la Base de Datos

**// Modificar de acuerdo a su proyecto**

A partir del diagrama de clases persistentes diseñamos el modelo relacional de la base de datos

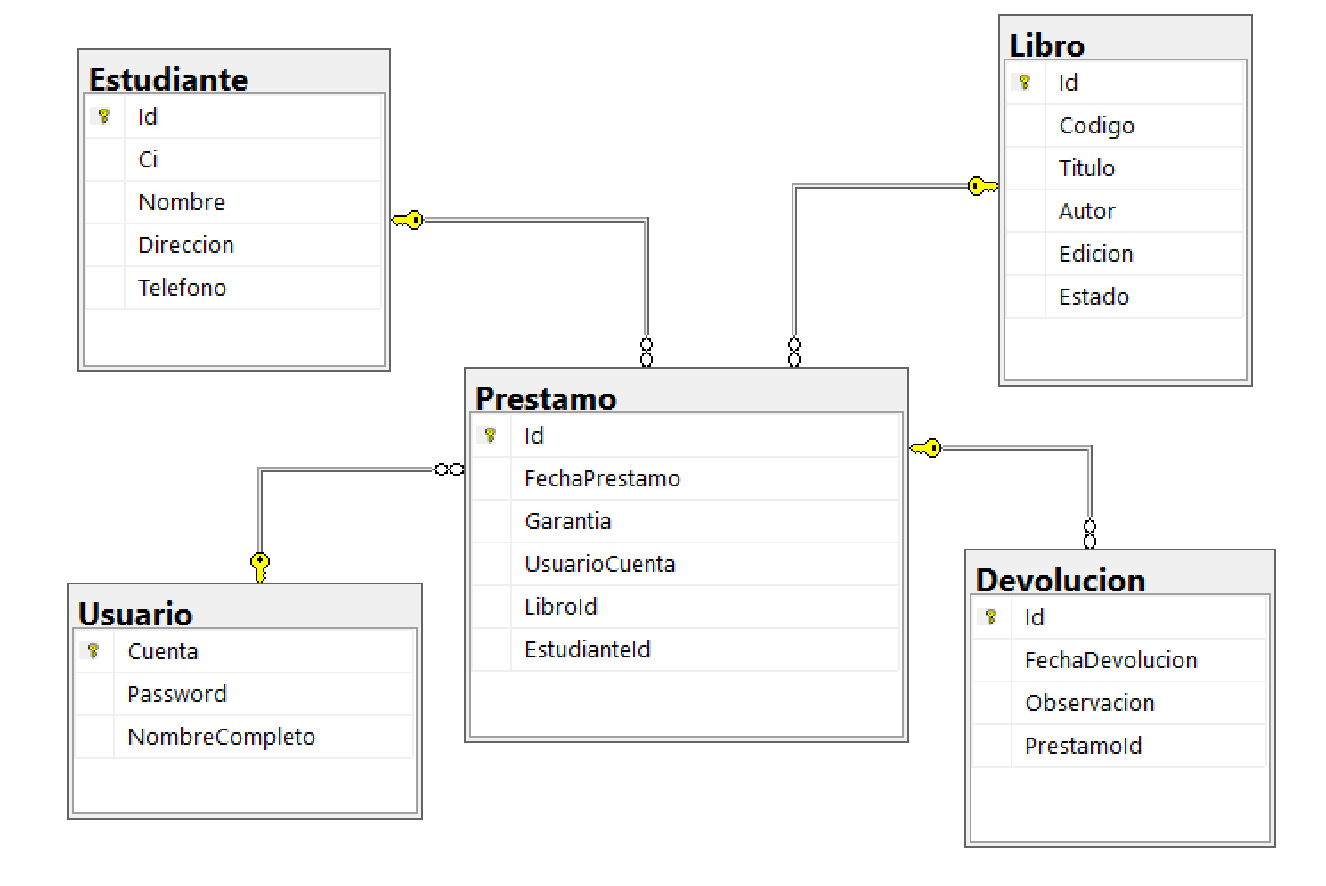


Figura 4: Modelo Relacional de la Base de Datos

## 3.9 Diagrama de secuencias

//por lo menos 2, se recomienda el caso de uso inicio de sesión y un CRUD

El diagrama de secuencias muestra el flujo de acciones que realiza un caso de uso.

## 3.10 Primer Sprint

### 3.10.1 Sprint backlog

**// Modificar de acuerdo a su proyecto**

El sprint backlog muestra **qué[[1]](#footnote-1)** es lo que se tiene que desarrollar en esta iteración.

|  |  |  |  |
| --- | --- | --- | --- |
| **Historia de**  **usuario** | **Descripción** | | **Duración** |
| **Primer Sprint** | **HU1** | **Ingresar al sistema con inicio de sesión** | **2 semanas** |
| **HU2** | **Gestionar (CRUD) Secretarias** |
| **HU3** | **Gestionar (CRUD) Clientes** |
| **HU4** | **Gestionar (CRUD) Equipos** |

Tabla 14: Primer Sprint Backlog

### 3.10.2 Diseño de interfaces

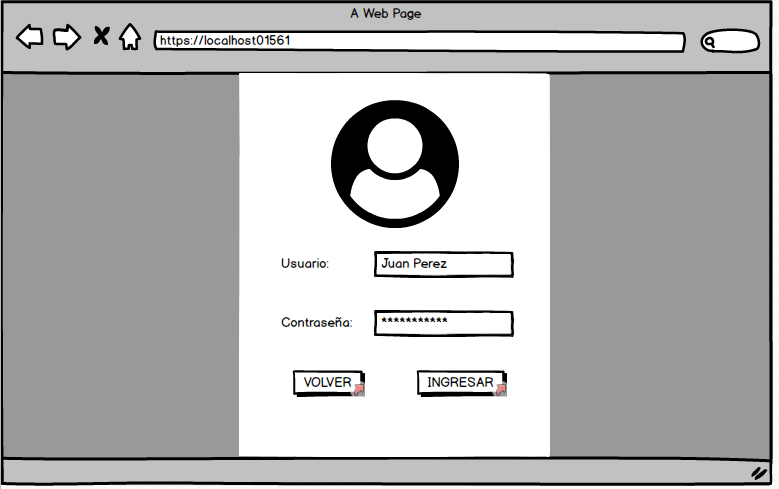


Figura 5: Interfaz login

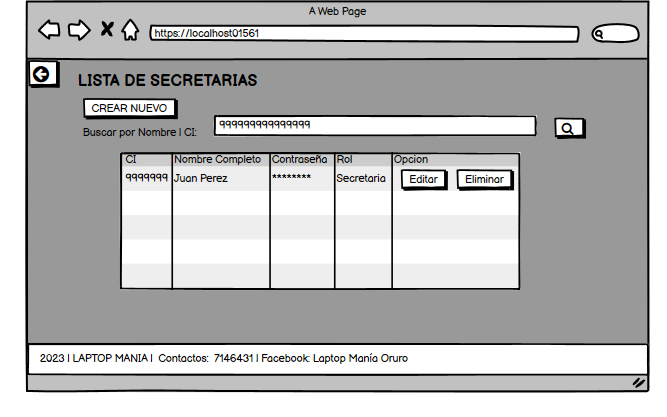


Figura 6: Interfaz CRUD de Secretarias

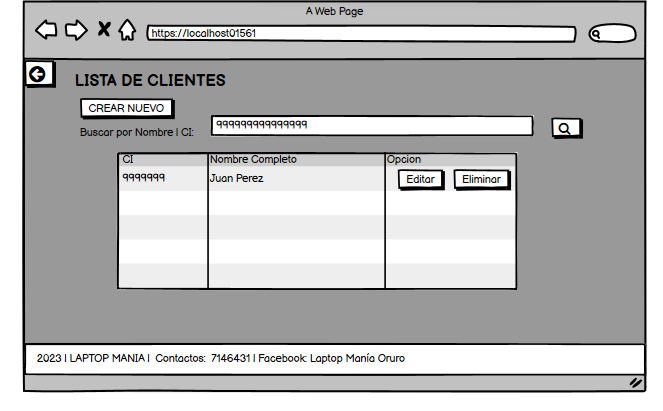


Figura 7: Interfaz CRUD de Clientes

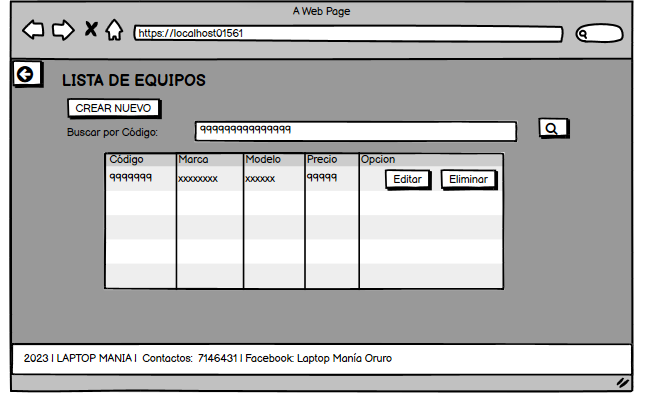


Figura 8: Interfaz CRUD de Productos

### 3.10.3 Sprint review

En la reunión del sprint review se muestra el product increment, que representa los entregables realizados al cliente.

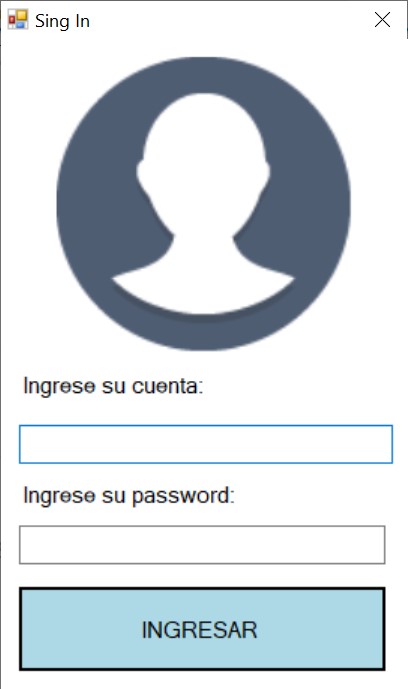


Figura 8: Pantalla LogIn

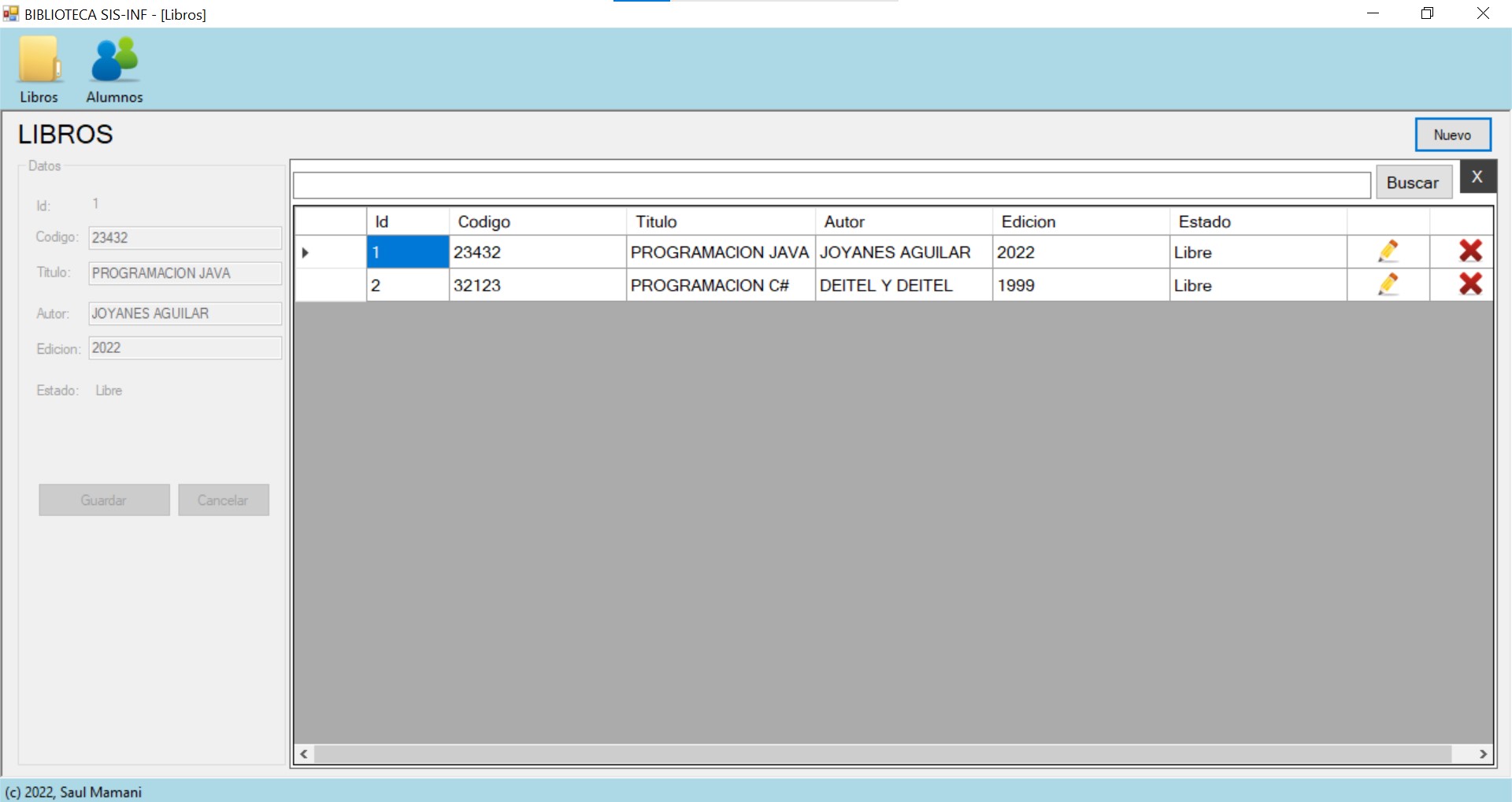


Figura 9: Pantalla CRUD de libros

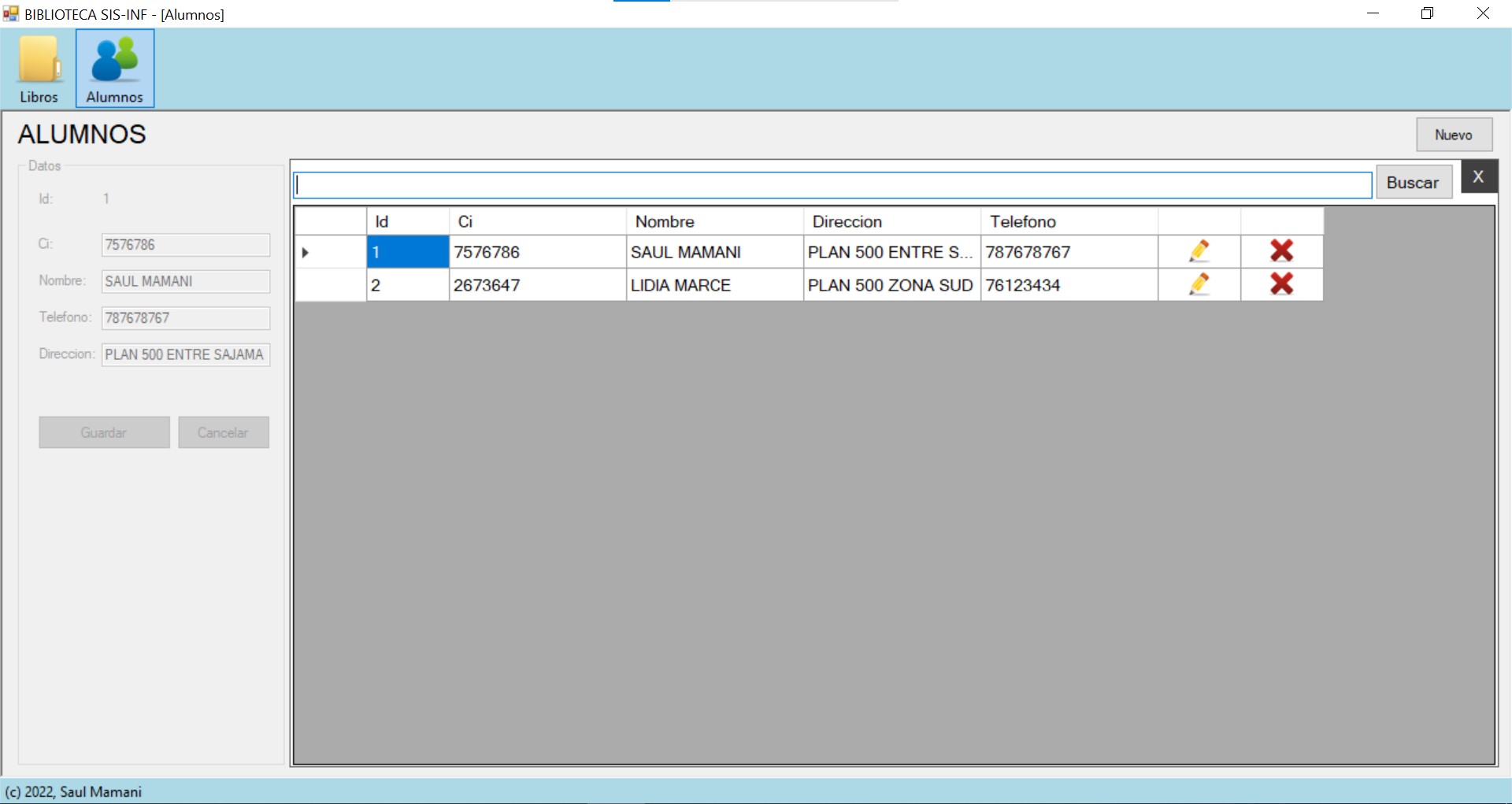


Figura 10: Pantalla CRUD de libros

### 3.10.4 Sprint retrospective

**//Esto es igual para todos y para todos los sprints,**

**//colocar los commit de su GIT y GITHUB**

El desarrollo del sprint (iteración) se ha cumplido a cabalidad, entregando el sistema en las una semana planificadas según el sprint backlog.

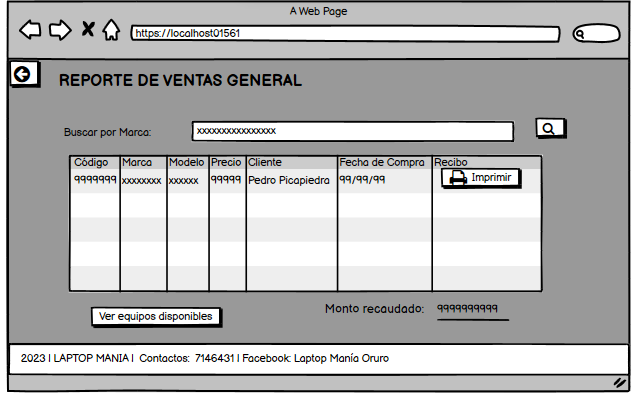
## 3.11 Segundo Sprint

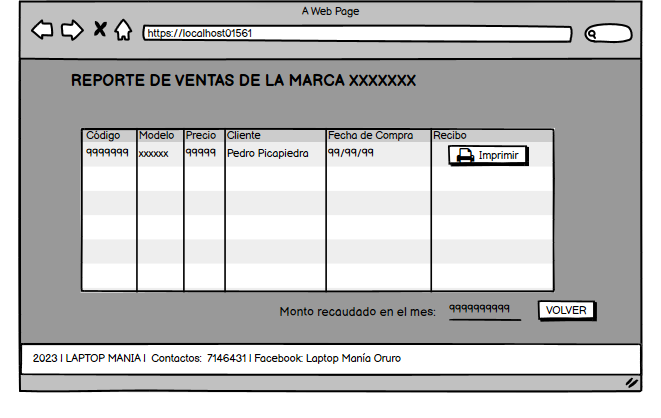
### 3.11.1 Sprint backlog

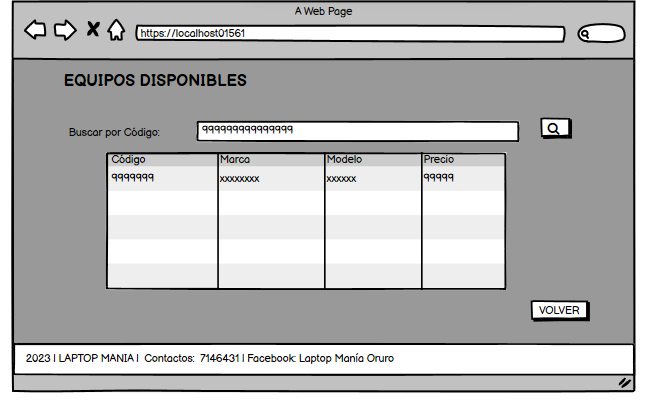
|  |  |  |  |
| --- | --- | --- | --- |
| **Historia de**  **usuario** | **Descripción** | | **Duración** |
| **Segundo Sprint** | **HU5** | **Generar Reportes de Productos Disponibles** | **1 semana** |
| **HU7** | **Generar Registros de Ventas** |
| **HU9** | **Generar Reportes Mensuales** |

Tabla 15: Segundo Sprint Backlog

### 3.11.2 Diseño de interfaces







### 3.11.3 Sprint review

En la reunión del sprint review se muestra el product increment, que representa los entregables realizados al cliente.

### 3.11.4 Sprint retrospective

El desarrollo del sprint (iteración) se ha cumplido a cabalidad, entregando el sistema en las una semana planificadas según el sprint backlog.

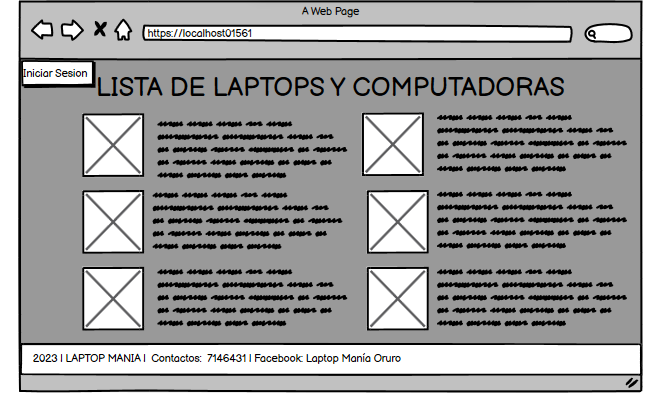
## 3.12 Tercer Sprint

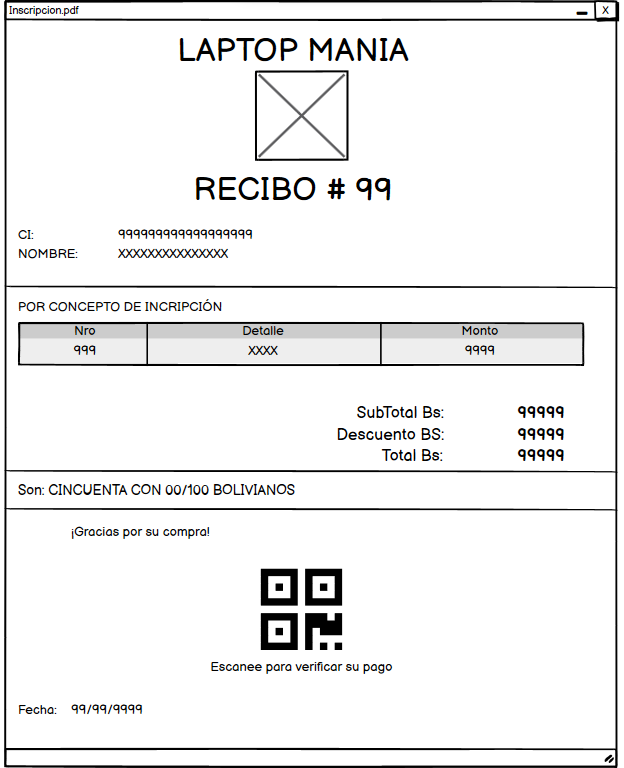
### 3.12.1 Sprint backlog

|  |  |  |  |
| --- | --- | --- | --- |
| **Historia de**  **usuario** | **Descripción** | | **Duración** |
| **Segundo Sprint** | **HU6** | **Generar Recibos** | **1 semana** |
| **HU8** | **Mostrar Ventana de Productos** |

Tabla 16: Tercer Sprint Backlog

### 3.12.2 Diseño de interfaces





### 3.12.3 Sprint review

En la reunión del sprint review se muestra el product increment, que representa los entregables realizados al cliente.

### 3.12.4 Sprint retrospective

El desarrollo del sprint (iteración) se ha cumplido a cabalidad, entregando el sistema en las dos semanas planificadas según el sprint backlog.

**4 CONCLUSIONES Y RECOMENDACIONES**

# BIBLIOGRAFÍA

**// Colocar la bibliografía en formato APA 7**

1. "Scrum: A Pocket Guide" - Gunther Verheyen
2. "Scrum: The Art of Doing Twice the Work in Half the Time" - Jeff Sutherland
3. "Scrum Mastery: From Good to Great Servant Leadership" - Geoff Watts
4. "The Scrum Field Guide: Practical Advice for Your First Year" - Mitch Lacey
5. "Essential Scrum: A Practical Guide to the Most Popular Agile Process" - Kenneth S. Rubin
6. "Clean Code: A Handbook of Agile Software Craftsmanship" - Robert C. Martin
7. "The Pragmatic Programmer: Your Journey to Mastery" - Andrew Hunt and David Thomas
8. "Refactoring: Improving the Design of Existing Code" - Martin Fowler
9. "Agile Web Development with Rails" - Sam Ruby, Dave Thomas, and David Heinemeier Hansson
10. "Programming Ruby: The Pragmatic Programmers' Guide" - Dave Thomas, Chad Fowler, and Andy Hunt
11. "Coaching Agile Teams: A Companion for ScrumMasters, Agile Coaches, and Project Managers in Transition" - Lyssa Adkins
12. "Agile Metrics in Action: Measuring and Enhancing the Performance of Agile Teams" - Christopher G. Worley and Dan McDonough
13. "Agile Project Management: Scrum Master Certification: PSM I Exam Preparation Guide and Handbook" - Paul Vii

# ANEXOS

/**/Todos del tablero de kanban, realizado en trello y fotos del tablero kanban realizado en clases**

ANEXO 1

Tablero kanban realizado en Trello

ANEXO 2

Fotos del tablero kanban hecho en clases presenciales

Anexo 3

Diagrama de Clases Control

Anexo 4

Diagrama de despliegue

1. La palabra **qué** es lo que se tiene que hacer, hace referencia al **análisis** de sistemas dentro el ciclo de vida de un proyecto de desarrollo de software. [↑](#footnote-ref-1)