

Software de Gestión y manejo de productos

Descripción de la Arquitectura – Diseño del Software.

IEEE 1471-2000 *Recommended Practice for Architectural Description of Software.* Descripción de la Arquitectura - Diseño de Software.

Versión [1.0]

Nombre: Jorge Daray Padilla Perez

Código: 216584703

Historia de revisiones

Fecha	Versión	Descripción	Autor
15/03/2023	1.0	Descripción de la arquitectura del software	Jorge Daray padilla perez

Contenido

1.	Error! Bookmark not defined.3	
<u>1.1.</u>	Error! Bookmark not defined.3	
<u>1.2.</u>	Error! Bookmark not defined.3	
<u>1.3.</u>	Error! Bookmark not defined.3	
<u>1.4.</u>	Error! Bookmark not defined.3	
<u>1.5.</u>	Error! Bookmark not defined.3	
2.	Error! Bookmark not defined.3	
<u>2.1.</u>	Error! Bookmark not defined.3	
<u>2.2.</u>	Error! Bookmark not defined.3	
<u>2.2.1.</u>	Error! Bookmark not defined.3	
<u>2.2.2.</u>	Error! Bookmark not defined.3	
3.	Error! Bookmark not defined.4	
<u>3.1.1.</u>	Error! Bookmark not defined.4	
<u>3.1.2.</u>	Error! Bookmark not defined.4	
4.	Error! Bookmark not defined.4	
<u>4.1.</u>	Error! Bookmark not defined.4	
<u>4.1.1.</u>	Error! Bookmark not defined.4	
<u>4.2.</u>	Error! Bookmark not defined.4	
<u>4.2.1.</u>	Error! Bookmark not defined.4	
<u>4.3.</u>	Error! Bookmark not defined.4	
<u>4.3.1.</u>	Error! Bookmark not defined.4	
5.	Error! Bookmark not defined.4	
6.	Error! Bookmark not defined.4	
<u>6.1.</u>	Error! Bookmark not defined.4	
<u>6.1.1.</u>	Error! Bookmark not defined.4	
<u>6.2.</u>	Error! Bookmark not defined.4	
<u>6.2.1.</u>	Error! Bookmark not defined.4	
<u>6.3.</u>	Error! Bookmark not defined.5	
<u>6.3.1.</u>	Error! Bookmark not defined.5	
7.	Error! Bookmark not defined.5	
<u>7.1.</u>	Error! Bookmark not defined.5	
<u>7.2.</u>	Error! Bookmark not defined.5	
<u>7.2.1.</u>	Error! Bookmark not defined.5	
<u>7.2.2.</u>	Error! Bookmark not defined.5	
<u>7.3.</u>	Error! Bookmark not defined.5	
<u>7.3.1.</u>	Error! Bookmark not defined.5	
<u>7.3.2.</u>	Error! Bookmark not defined.5	
8.	DESCRIPCIÓN DE DISEÑO DETALLADO.....	6
8.1	Resumen del diseño detallado.....	6
8.2	Diseño detallado del elemento 1.....	6
8.2.1	Vistas estructurales	6
8.2.2	Vistas de comportamiento	6
8.2.3	Otras vistas relevantes	6
8.2.4	Razón fundamental	6
8.3	Diseño detallado del elemento 2.....	6
8.3.1	6
...	6
8.x	Información de diseño detallado relevante a múltiples elementos.....	6

1. Introducción

Este documento describe los procesos de la estructura global del sistema, deja ver el orden jerarquizado de dicha estructura y su composición en módulos, expone la interacción entre los componentes y estructuras de datos del software, explica los estilos arquitectónicos que se usan para construir el software, este documento abstrae el sistema de nuestro software y nos permite ver su estructura y relaciones.

Los elementos de la Arquitectura de Software típico son: descripción de procesos, actividades, tareas, roles y productos, plantillas, lista de verificación, ejemplo y referencia a estándares y modelos, y herramientas.

El propósito de este documento, es proveer con guías y materiales personalizables y fáciles de usar para poder implementar un buen *Diseño de Software*.

Invertir esfuerzo en las **actividades** de diseño asegura que para la solución propuesta (por ejemplo software a construir) se habrá meditado alguna idea previa a su implementación (por ejemplo, codificación). Construir algo sin diseñarlo típicamente produce una solución que no cumple con los requerimientos, se entrega tarde, excede el presupuesto o es de baja calidad.

Invertir esfuerzos en **documentación** explícita habilita la comunicación y negociación entre los stakeholders del proyecto, más específicamente a aquellos que tienen interés en el diseño. Capturando un diseño en alguna forma (documento electrónico, documento en papel, modelos...) no es solo útil mientras el proyecto de software está activo, sino también para el futuro mantenimiento y mejoras.

La actividad de Arquitectura de Software y Diseño Detallado produce un documento denominado el *Diseño de Software* que permite a los stakeholders entender las interacciones en el software y la trazabilidad de los elementos diseñados hacia los requerimientos. Esto permite una forma de verificar que cada requerimiento ha sido atendido (por ejemplo, completitud de diseño). El Diseño de Software también se usa para el mantenimiento del software porque describe los componentes y sus interfaces.

1.1. Propósito

Este documento proporciona una apreciación global y comprensible de la Arquitectura - Diseño del Software usando diferentes puntos de vista para mostrar distintos aspectos del sistema. Intenta capturar y llegar a las decisiones de arquitectura críticas que han sido hechas en el sistema.

1.2. Alcance

Este documento da ideas del desarrollo del software para una mayor claridad a la hora de crear el diseño del software, así como llevar un orden de este mismo y tener una visión más clara del proyecto.

1.3. Definiciones, siglas y abreviaturas.

Hardware: En informática, para designar el conjunto de los componentes que integran la parte material de una computadora u ordenador.

Software: Conjunto de programas y rutinas que permiten a la computadora realizar determinadas tareas.

Web: Se usa normalmente en la expresión página web, que significa 'documento de la Red, al que se accede mediante enlaces de hipertexto'.

1.4. Referencias

Referencias	Título	Ruta	Fecha
REAL ACADEMIA ESPAÑOLA. (2021, 17 de enero). Definición de Hardware. Real Academia Española. https://dle.rae.es	Definición de Hardware	https://dle.rae.es	08/03/23
REAL ACADEMIA ESPAÑOLA. (2021, 17 de enero). Definición de Hardware. Real Academia Española. https://dle.rae.es	Definición de Software	https://dle.rae.es	08/03/23
REAL ACADEMIA ESPAÑOLA. (2021, 17 de enero). Definición de Web. Real Academia Española.	Definición de web	https://dle.rae.es	08/03/23

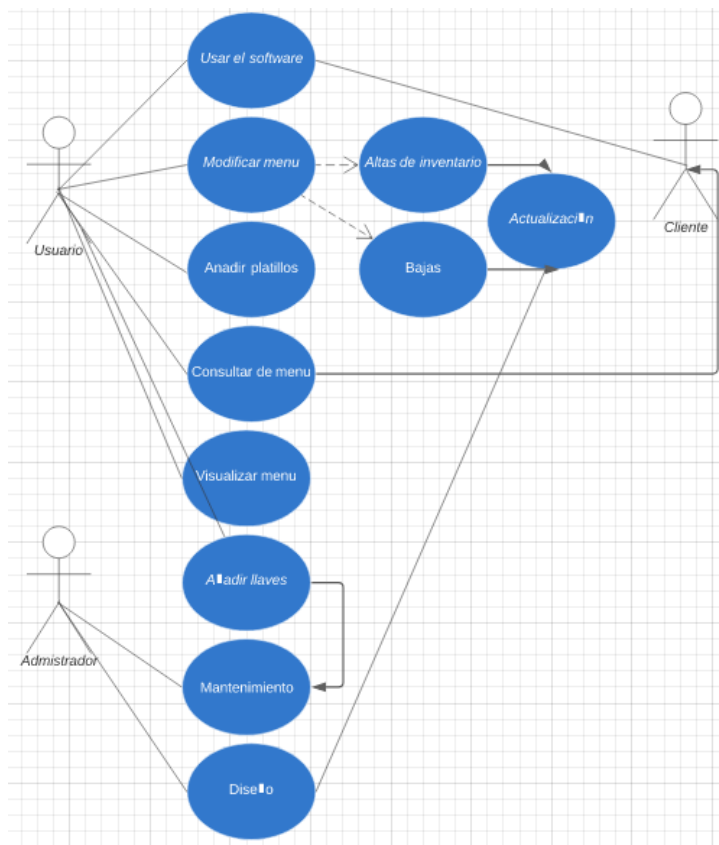
1.5. Visión general

Se mostrarán los diagramas que explican la manera en que estará detallado el software.

casos de uso, modelos de diseño y descripción de diseño.

2. Vista del Modelo de Casos de Uso

1.6. Diagrama de Casos de Uso relevantes a la Arquitectura



1.7. Casos de Uso relevantes a la Arquitectura

Casos de Uso y actores más significativos para definir la Arquitectura del Software. Estos casos de uso son:

- los que ayudan a mitigar los riesgos más serios
- los más importantes para los usuarios y clientes
- los que ayudan a cubrir toda la funcionalidad importante para que nada quede en sombras.

1.1.1. Actualizacion de menu

Dependiendo del usuario tendrá diferentes permisos.

1.1.2. Permisos de usuarios

dependiendo del usuario se le otorgarán diferentes permisos.

1.1.3. Ver inventario.

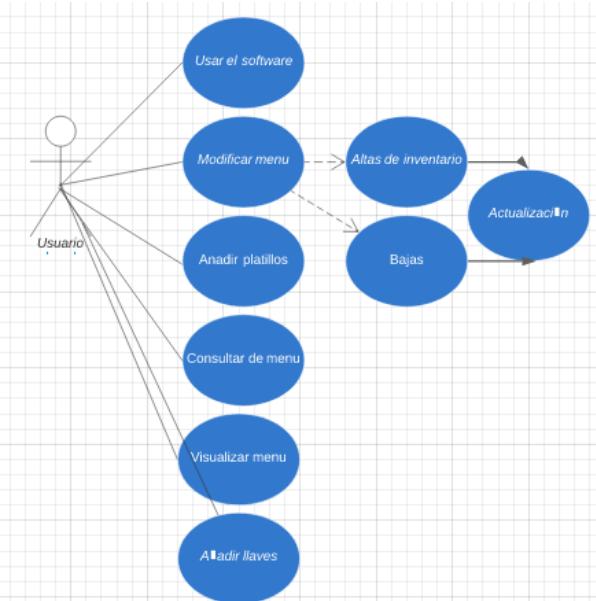
Apartado donde se podrá ver el menu y los diferentes platillos que ofrecen.

3. Trazabilidad desde el Modelo de Casos de Uso al Modelo de Diseño

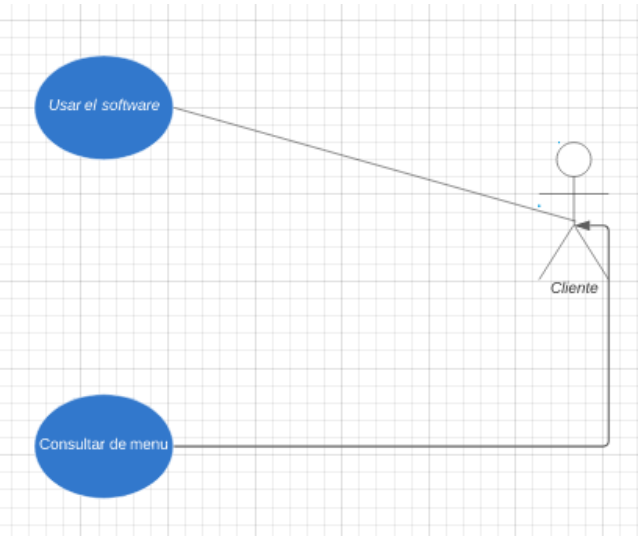
Describir la trazabilidad, la cual es la capacidad de rastrear todos los procesos, desde la adquisición de materias primas hasta la producción, consumo y eliminación. Esto aplicado en el Modelo de Casos de Uso nos es muy útil pues nos permite ver de manera clara el usuario de nuestra aplicación, en conjunto con sus características y funciones que puede llegar a realizar. Este además permite documentar las acciones que nuestro programa realiza de manera clara y concisa. El modelo de diseño se basa sobre el modelo de análisis describiendo, en mayor detalle, la estructura del sistema

y cómo será implementado el sistema. Aquí podemos ver a detalle cómo se relacionan las funciones con el usuario.

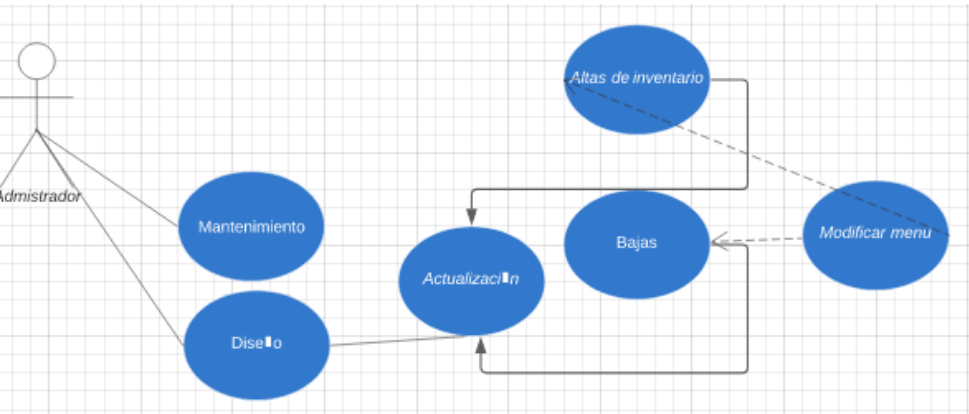
1.1.4. Caso de Uso relevante a la Arquitectura 1



1.1.5. Caso de Uso relevante a la Arquitectura 2

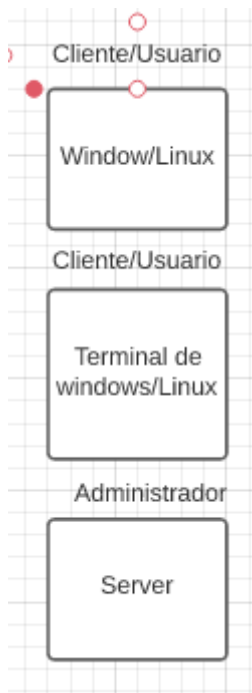


1.1.6. Caso de Uso relevante a la Arquitectura 3



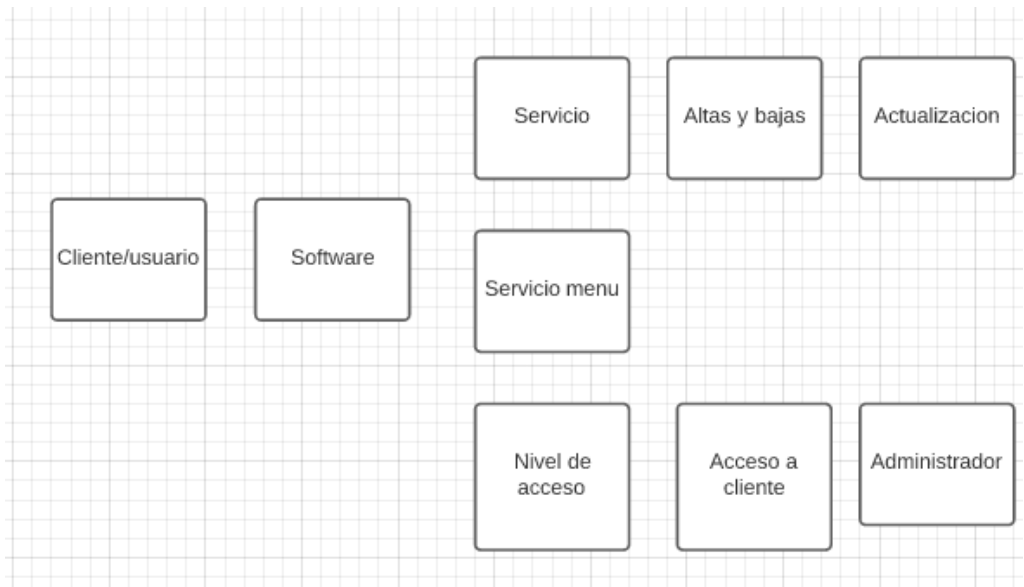
4. Vista del Modelo de Diseño

1.8. Descomposición en Subsistemas

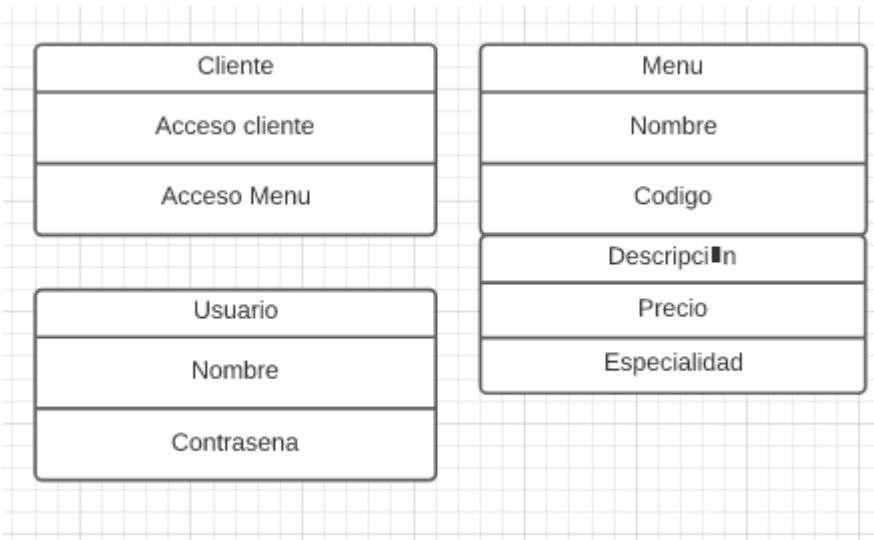


1.1.7. Subsistema 1

Se tendra diferentes niveles de acceso en los cuales primero tenemos el de cliente/usuario el cual dependera del acceso permitido, luego otro donde la interfaz sera en la terminal de windows o linux para al final el nivel de datos el cual el administrador sera el encargado de eso.



1.9. Diseño de Clases



1.1.8. Clase cliente

Se encargará de la administración y gestión de clientes.

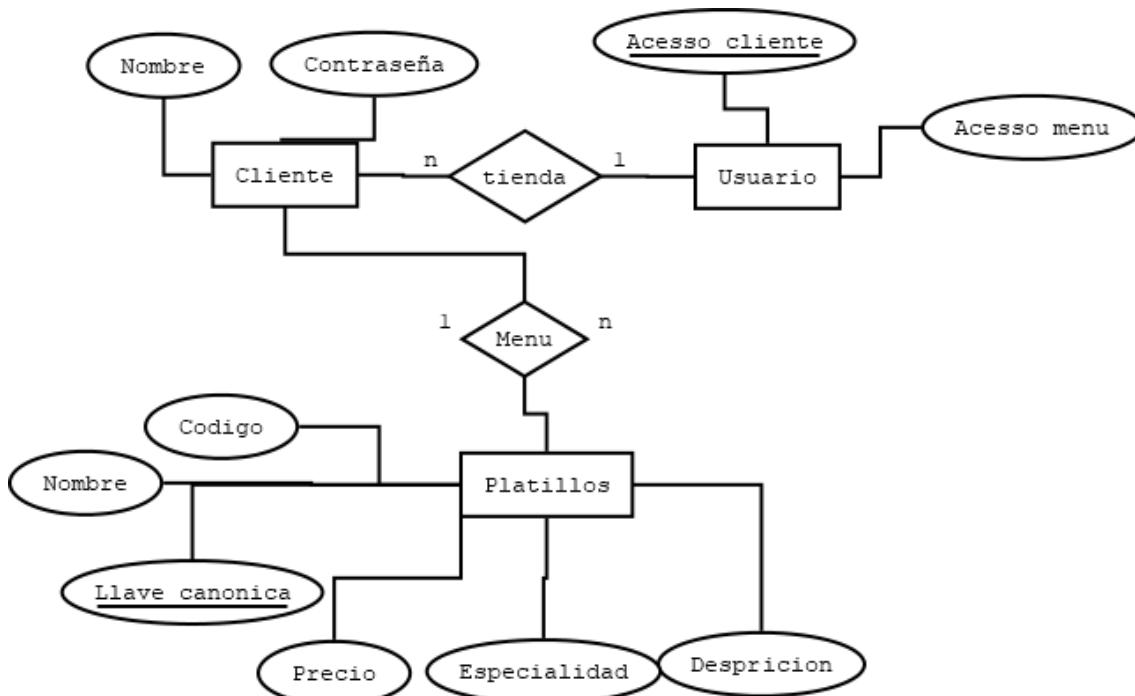
1.1.9. Clase menu

Se encargará de gestionar el apartado de cada producto para que después esté en el menú.

1.1.10. Clase usuario

Controlara la entrada y salidas de los usuarios que estarán en el software.

1.10. Diseño de Casos de Uso



1.1.9. Diseño del Caso de Uso 1

Usuario: Persona que estará usando el software como por ejemplo el administrador.

Cliente: Persona que puede ver el menú empleado o cliente.

Platillos: platillos que están en el menú actual.

Menu: lugar donde se almacenan el producto con su llave canónica.

Tienda: donde el administrador modifica el cliente o menu.

5. Trazabilidad desde el Modelo de Diseño al Modelo de Implementación

Actualizaciones de menú: lo implementa el administrador, añadir plato, eliminar plato.

Acceso al software: el usuario ya sea cliente o empleado entra al módulo del sistema.

6. Vista del Modelo de Implementación

1.11. Subsistemas

1.1.10. Modificación y actualización de menu

Realización de cambios en el menú por medio de altas y bajas las cuales crean una llave canónica en dado caso de ser altas y este lo añadirá al archivo indice.

1.12. Componentes

1.1.11. Componente platillos

El componente platillo se asigna al menú y a la vez al indice para que estén entrelazados

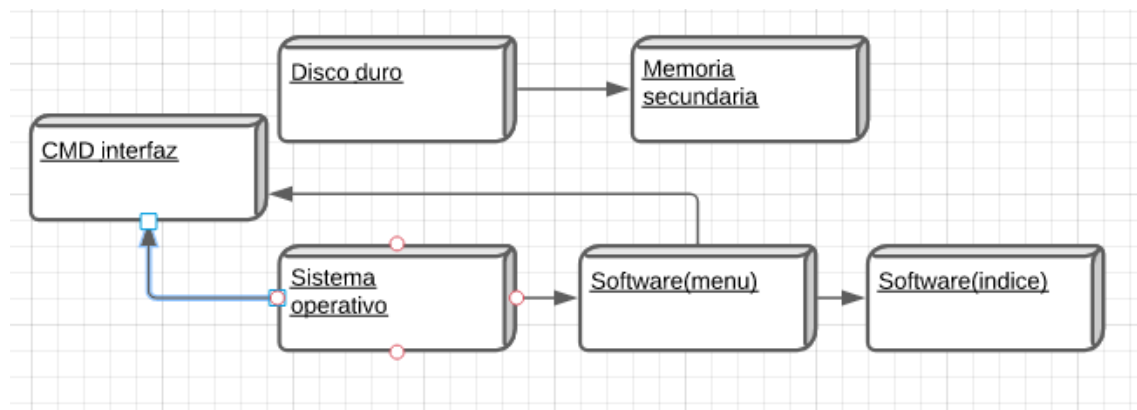
1.13. Interfases

1.1.12. Interface 1

En este caso no se usará una interfaz como tal ya que se usara la interfaz de la consola de Windows

7. Vista del Modelo de Distribución

1.14. Diagrama de Distribución



1.15. Nodos

1.1.13. Nodo CMD

Este estará conectado a el software para ver el menú de los platillos los cuales estarán almacenados en el índice.

1.1.14. Nodo Archivo

Se creará un archivo al final en el cual este tendrá los índices guardados con anterioridad.

1.16. Conexiones

1.1.15. Conexión CMD

Siempre estará disponible cuando el programa se esté ejecutando.

1.1.16. [Conexión archivo de texto

Se conectará cuando se guarde un nuevo archivo o un nuevo platillo al menú

8. DESCRIPCIÓN DE DISEÑO DETALLADO

8.1 Resumen del diseño detallado

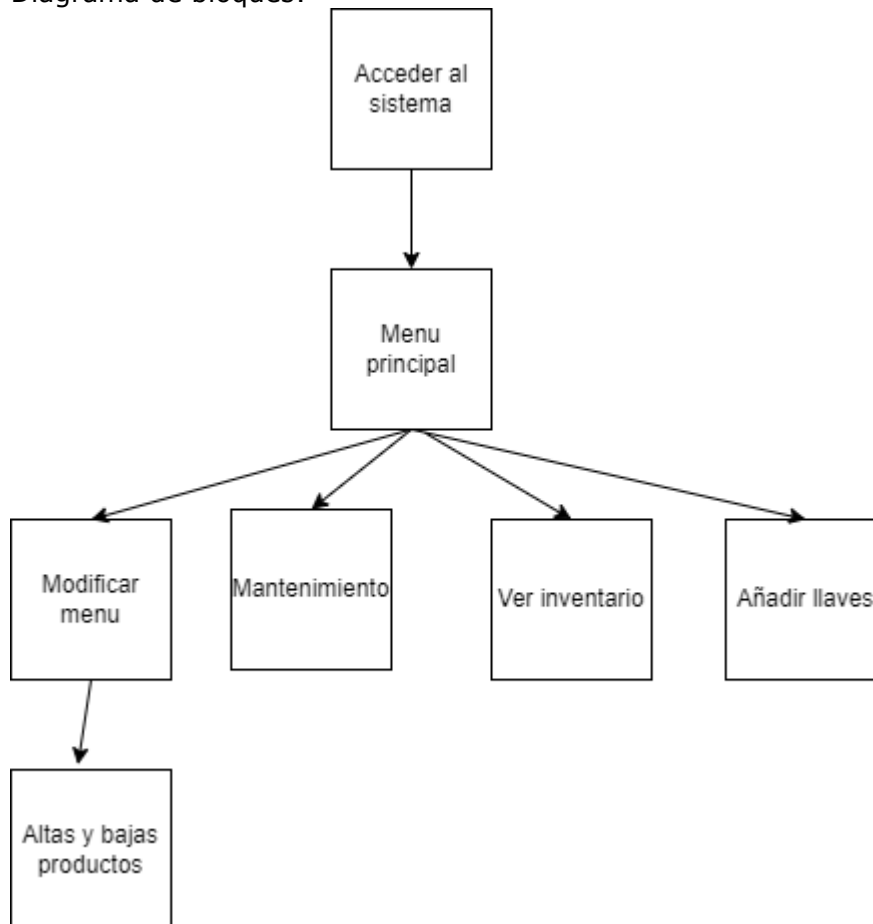
En resumen, este software fue desarrollado meticulosamente para poder cumplir con las necesidades de una tienda de zapatos donde se podrá administrar el inventario en el almacén donde se tendrá un control de manera virtual en una base de datos donde podremos organizar a comodidad, con un sistema de facturación que nos permitirá llevar el registro de las ventas y devoluciones para llevar la verificación de las compras.

Este software también cuenta con un catálogo en el cual nuestros usuarios podrán navegar y buscar por ID o nombre así como proporcionarle información como su costo, modelo y colores, con un sistema de devoluciones donde uno tendrá que validar su identidad para continuar con el proceso que gracias a la base de datos se podrá ver un historial de compras para el usuario y así pedir la devolución llevando un reporte de ventas para generar reportes que nos permitirá saber de la pérdida o ganancia de nuestra tienda.

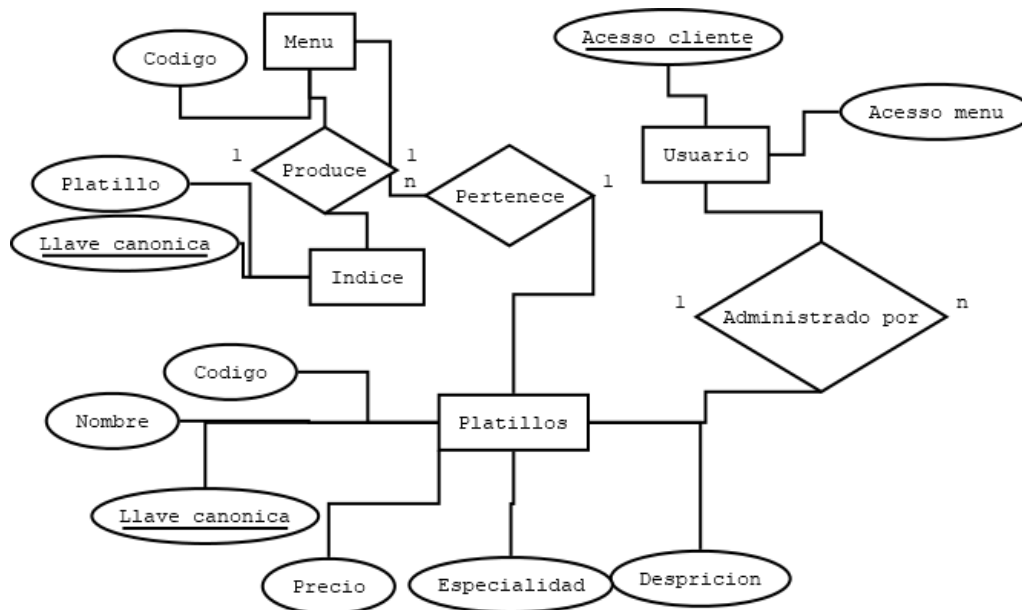
8.2 Diseño detallado del elemento 1

8.2.1 Vistas estructurales

Diagrama de bloques:



Diseño semántico de dato:



Diccionario de clases:

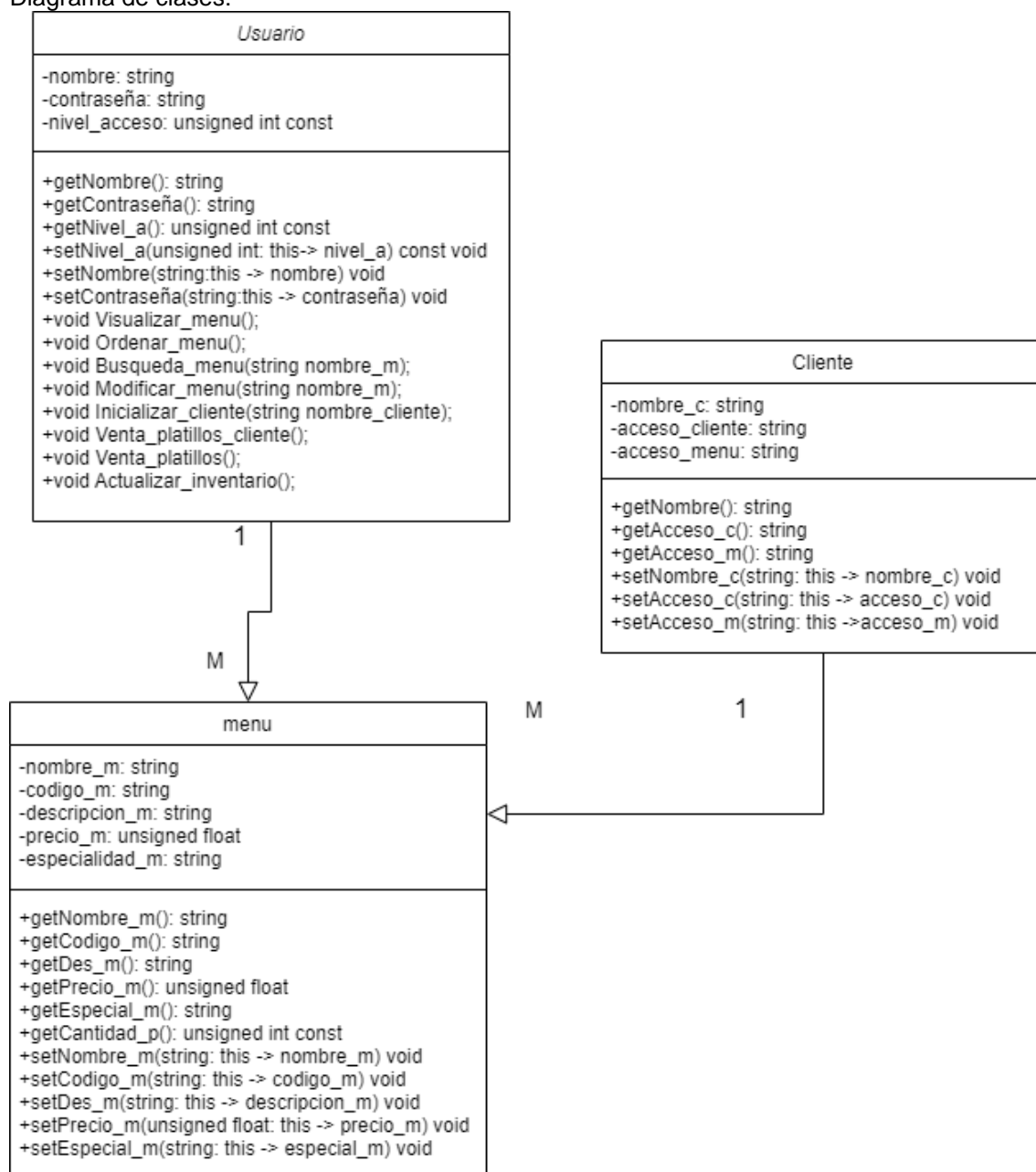
NOMBRE:	Usuario		
Descripción:	Contendrá las características del producto		
Atributos			Operaciones
Nombre	Tipo de dato	Visibilidad	
Nombre	String	Privado	Setter y getter
Tipo	String	Privado	Setter y getter
Contraseña	String	Privado	Setter y getter
Nivel de acceso	String	Privado	Setter y getter

NOMBRE:	Cliente		
----------------	---------	--	--

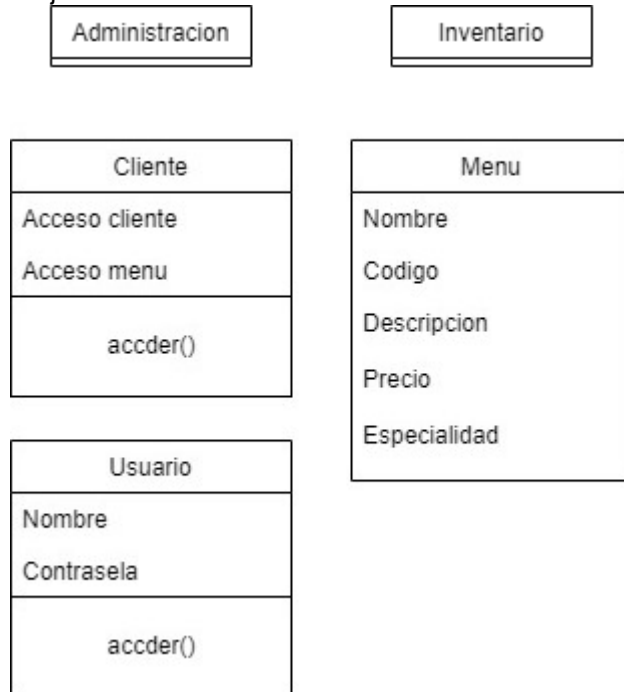
Descripción:	Contendrá la llave primaria del producto a identificar		
Atributos			Operaciones
Nombre	Tipo de dato	Visibilidad	
tipo	String	Privado	Setter y getter
nombre	String	Privado	Setter y getter
Acceso a menú	String	Privado	Setter y getter

NOMBRE:	Menú		
Descripción:	Contendrá la forma canónica del producto		
Atributos			Operaciones
Nombre	Tipo de dato	Visibilidad	
nombre	String	Privado	Getter y aumento
Código	String	Privado	Setter y aumento
descripción	String	Privado	Setter y getter
precio	INT	Privado	Setter y getter

Diagrama de clases:



Tarjetas CRC:



8.2.2 Vistas de comportamiento

Diagrama de comunicaci3n:

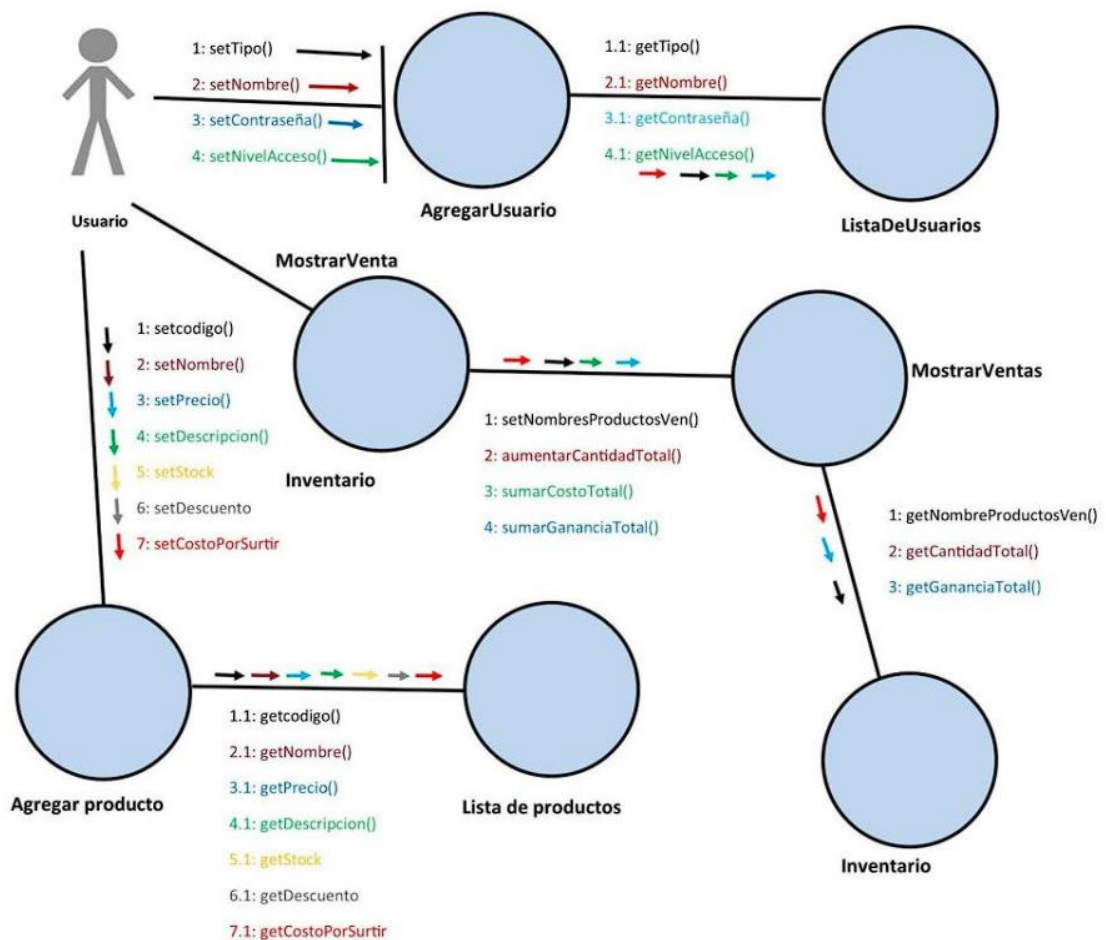


Diagrama de estados:

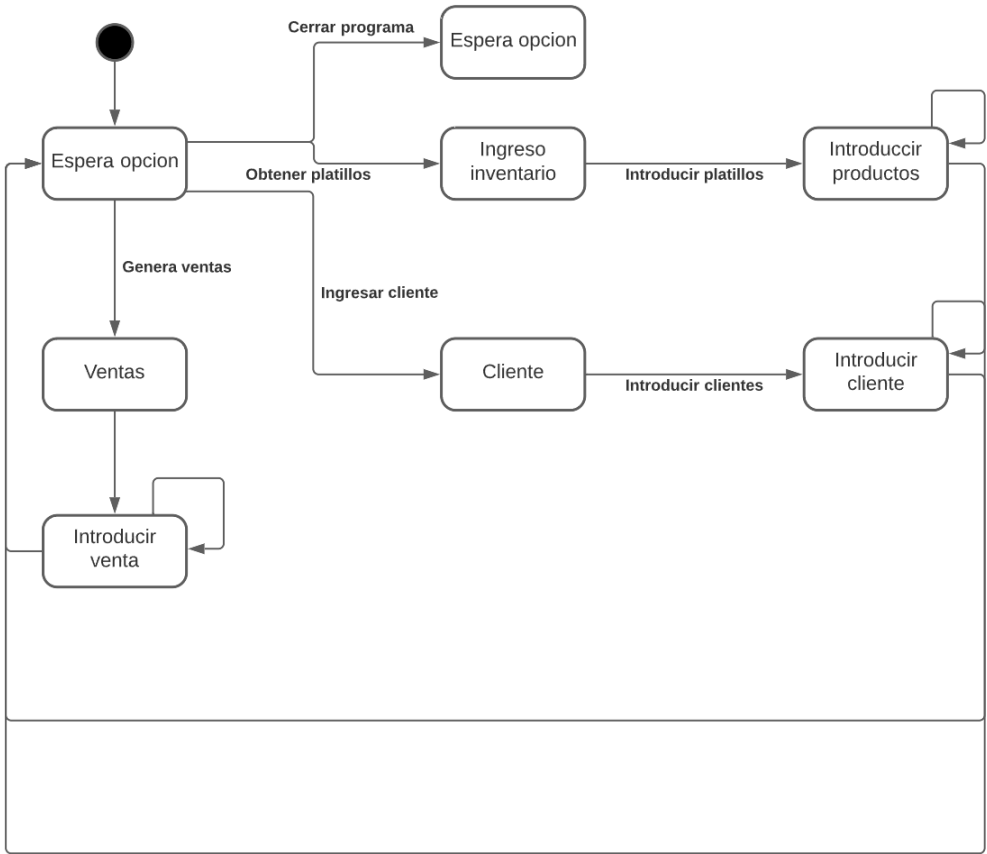


Diagrama de actividades:

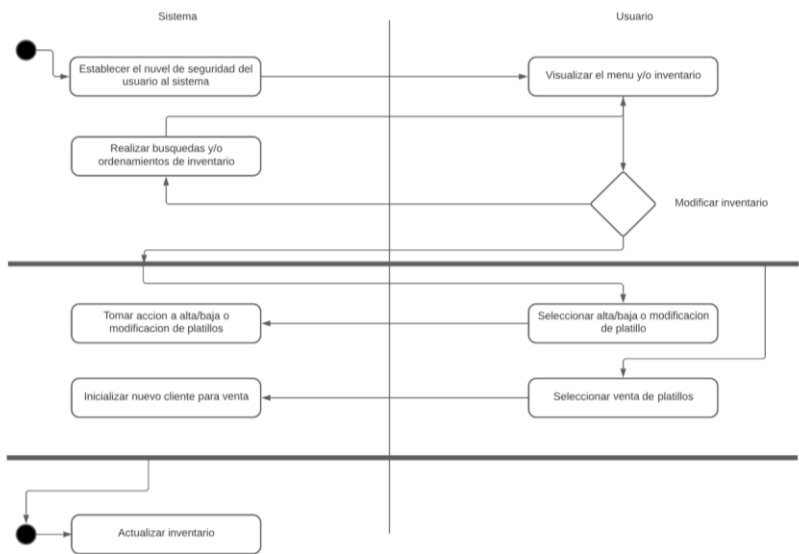


Diagrama de componentes:

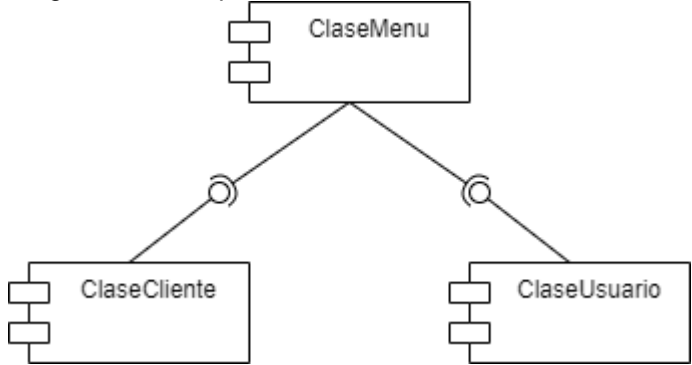
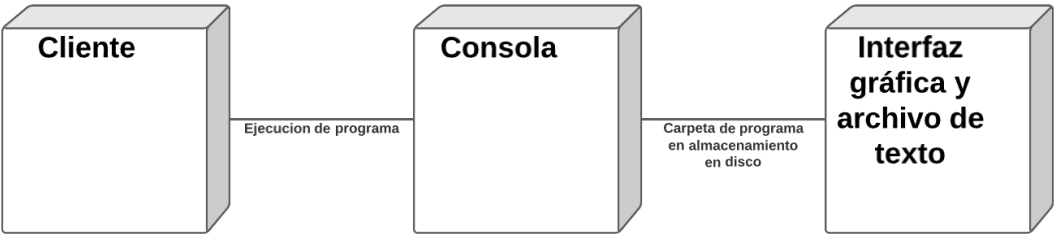


Diagrama de despliegue:



8.2.3 Otras vistas relevantes

Diagrama de objetos:

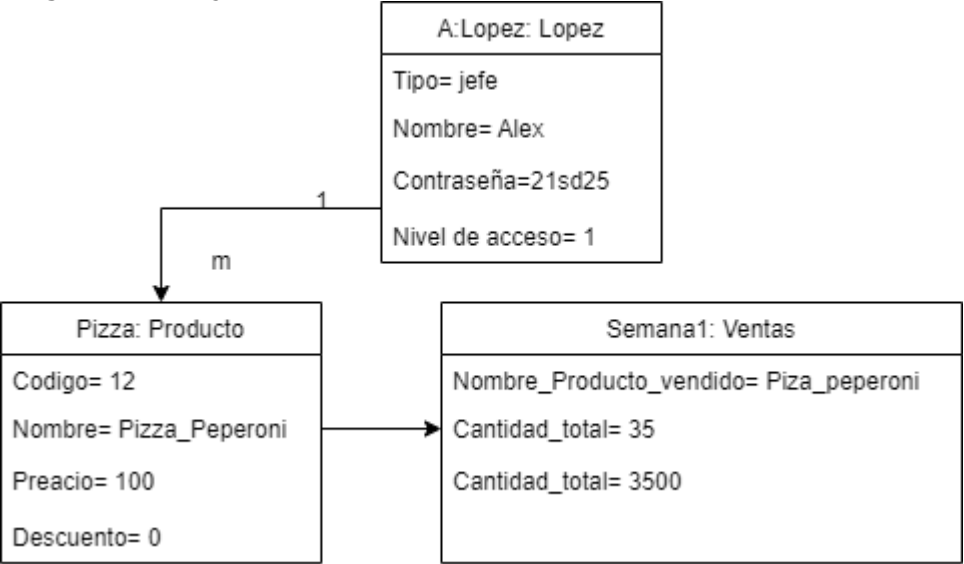
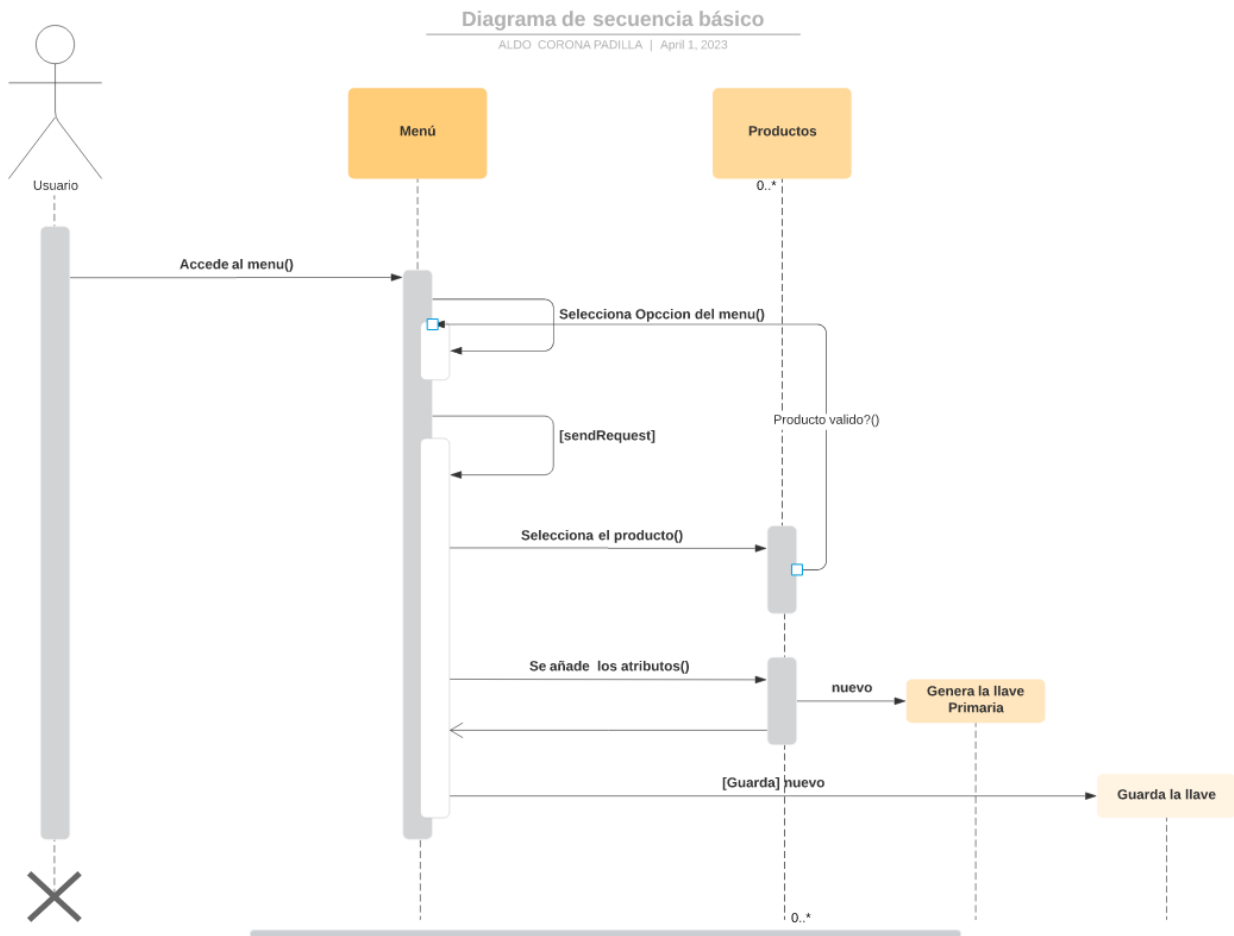


Diagrama de secuencia:



8.2.4 Razón fundamental

N/A

8.3 Diseño detallado del elemento 2

8.3.1 ...

...

8.x Información de diseño detallado relevante a múltiples elementos

