“Año de la unidad, la paz y el desarrollo”

Imagen que contiene botiquín de primeros auxilios, objeto, dibujo, señal

Descripción generada automáticamente

SISTEMA DE REGISTRO DE ORDENES PARA EL RESTAURANTE

“VICIUS CHICKEN”

CURSO: Programación Orientada a Objetos

**INTEGRANTES:**

Cristian Eloy Casahuaman Ramírez U19217444

Cunurana Quispe Oscar Tony Ronaldo U21211917

Trujillo Fierro Bill Uriel U21223870

Vera Espino Marcela Felicitas U21302469

**DOCENTE: HERNAN FRANCISCO PEÑA CARNERO**

**SECCIÓN: 39001**

**AÑO 2023**

**ÍNDICE**

CAPÍTULO 1

[1 ASPECTOS GENERALES 4](#_Toc140886000)

[1.1 Definición del Problema 4](#_Toc140886001)

[1.1.1 Descripción del Problema 4](#_Toc140886002)

[1.2 Definición de Objetivos 4](#_Toc140886003)

[1.2.1 Objetivo general 4](#_Toc140886004)

[1.2.2 Objetivos específicos 5](#_Toc140886005)

[1.2.3 Alcances y limitaciones 5](#_Toc140886006)

[2 MARCO TEÓRICO 6](#_Toc140886007)

[2.1 Antecedentes 6](#_Toc140886008)

[2.2 Fundamento Teórico 7](#_Toc140886009)

[2.2.1 JAVA 7](#_Toc140886010)

[2.2.2 BASE DE DATOS 8](#_Toc140886011)

[2.2.3 NETBEANS 8](#_Toc140886012)

[**3** **Lista de requerimientos funcionales** 9](#_Toc140886013)

[**4** **Lista de requerimientos no funcionales** 10](#_Toc140886014)

[**5** **Diagrama de casos de uso** 11](#_Toc140886015)

[**6** **Diagrama de Actividades** 12](#_Toc140886016)

[7 **Especificaciones de los casos de uso** 13](#_Toc140886017)

[**8** **Diagrama de clases** 15](#_Toc140886018)

[**9** **Prototipo de Referencia** 16](#_Toc140886019)

[**10** **Project Charter** 19](#_Toc140886020)

[**11** **Prototipo en desarrollo** 20](#_Toc140886021)

[**12** **Código del proyecto** 22](#_Toc140886022)

[**12.1** **Principal** 22](#_Toc140886023)

[**12.2** **Login** 22](#_Toc140886024)

[**12.3** **Registrar Cliente** 23](#_Toc140886025)

[**12.4** **Selección Menú** 24](#_Toc140886026)

CAPÍTULO 1

# ASPECTOS GENERALES

## Definición del Problema

En San Juan de Miraflores, como en otras partes del mundo las personas se encuentran en la búsqueda de satisfacer sus necesidades, socializarse y relajarse de los problemas que los aquejan o carga del trabajo semanal, y además de compartir tiempo con la familia, en tal sentido los restaurantes que son un reflejo de las tendencias que nos define como consumidores se han convertido en un medio frecuentemente usado por la población. Una buena alternativa es la Pollería “Vicius Chicken”, el cual se encuentra en la ciudad de SJM. Además, desea seguir escalando hacia lo más alto, obteniendo la preferencia del consumidor. Motivo por el cual realizó la siguiente investigación, ya que en los últimos meses se ha generado una alta demanda de clientes, la cual hacen que halla una desorganización en las órdenes.

### Descripción del Problema

Uno de los elementos más importantes de la baja rentabilidad del restaurante es que no tiene un sistema de orden, la cual ayudaría a que las ordenes o pedidos estén organizados y los empleados sean más eficientes. Ya dicho lo expuesto, la pregunta principal que guía esta investigación es; ¿Cómo el sistema de pedido de ordenes mejoraría los pedidos de los clientes?

## Definición de Objetivos

### Objetivo general

El objetivo general en si es poder arreglar el problema de órdenes o pedidos. Determinar que el cliente pueda realizar su orden más rápido y que además no tenga la necesidad de hacer una larga espera. A la vez, una mayor facilidad mayor satisfacción en el cliente con un código el cual sea de forma muy sencilla de manejar, en el cual se puedan realizar pedidos de una forma más rápida y ordenada. También que pueda registrar datos personales de la persona que realice el pedido. , con este registro queremos asegurarnos que el pedido es 100% legítimo y no será una persona la cual por accidente ordene algo del menú, con todo lo anterior dicho este servirá de tal forma que puedan darnos su recomendación con otras personas y dar una mejora al restaurante.

### Objetivos específicos

* Implementar el sistema de forma que sea intuitivo y cómodo para el uso del personal.
* Reducir los problemas de saturación de pedidos con nuestro sistema
* Desarrollar un sistema de registro de órdenes que permita a los clientes realizar pedidos de forma rápida y sencilla.
* Determinar los sistemas necesarios para el correcto funcionamiento del programa.

### Alcances y limitaciones

#### **Alcances**

El proyecto tiene como finalidad en implementar un sistema de registro y gestión eficiente de las órdenes de los clientes con el objetivo de mejorar la experiencia del cliente y agilizar la entrega de pedidos de manera efectiva.

El propósito de realizar esta investigación como grupo es aplicar de manera efectiva los conocimientos adquiridos durante el ciclo académico, con el fin de demostrar las habilidades y competencias adquiridas. A través de la ejecución exitosa de este proyecto de investigación.

#### **Limitaciones**

* El sistema podría presentar limitaciones para personas denominadas "principiantes digitales", es decir, aquellas personas que no están familiarizadas con la tecnología o tienen poca experiencia en su uso. Estos usuarios pueden encontrar dificultades al interactuar con el sistema debido a su falta de conocimiento y habilidades tecnológicas.
* Ese proyecto de sistema de registro de ordenes puede requerir tiempo, habilidades técnicas y recursos financieros adecuados. Esto puede limitar el alcance del proyecto y la capacidad de la empresa para llevar a cabo

CAPÍTULO 2

# MARCO TEÓRICO

## Antecedentes

* Toast, un POS compacto y sencillo:

Toast es un Software POS para restaurantes que se adapta a todo tipo de negocios y de cualquier tamaño, ya que ofrece diferentes soluciones. Permite gestionar ventas, pagos o pedidos en un aparato el cual sea facil de manejar para los clientes. (Muchas cadenas de comida rápida usan este software por ejemplo Poki´s Kyng o FASFU).

* Lightspeed, muy rápido e intuitivo:

Lightspeed es un POS con un sistema de comandos para restaurantes de servicio completo para mesas, servicio rápido, bares, cafeterías o restaurantes de hoteles con todas las funciones. Tiene funciones muy sencillas para implementarse en una máquina, la cual será manipulada por los clientes (Este software está basado en la nube y te permitirá servir a tus clientes de manera precisa y sin esperas. Cadenas de comida rápida como Micky´s burguer´s o Toño Búrguer dependen de este software para sus pedidos).

## Fundamento Teórico

### JAVA

Java es un lenguaje de programación orientado a objetos que permite crear sistemas más complejos y fomenta la reutilización y ampliación del código. La programación orientada a objetos se asemeja al mundo real y agiliza el desarrollo de software. Además, facilita el trabajo en equipo.

Los aportes más significativos que nos proporciona java son:

* Clases y objetos: Java nos permite la creación de clases y objetos de manera rápida y eficaz. La sintaxis de Java para la creación de objetos es muy clara y entendible, lo que hace sencilla su comprensión y mantenimiento.
* Herencia: Java nos permite definir las clases que heredan los miembros y comportamientos de otras clases. La herencia en Java permite reutilizar el código existente y facilita la creación de jerarquías de clases
* Polimorfismo: Java implementa el concepto de polimorfismo mediante la posibilidad de que los objetos de una clase puedan ser tratados como objetos de otra clase. Esto permite escribir código más genérico y flexible.
* Encapsulamiento: Java implementa el encapsulamiento mediante el uso de clases y modificadores de acceso. Las clases en Java permiten definir la estructura y el comportamiento de un objeto, mientras que los modificadores de acceso nos permiten controlar el acceso a los miembros de una clase.
* Interfaces: Las interfaces en Java permiten la creación de una jerarquía de tipos que pueden ser utilizados de manera polimórfica.

### BASE DE DATOS

La base de datos es una herramienta fundamental para la programación orientada a objetos ya que nos permite gestionar y almacenar los datos de una aplicación de manera más rápida, eficiente y organizada. Algunos de los beneficios que aporta la base de datos para nuestro proyecto son:

* Persistencia de datos: Los datos almacenados permanecen intactos incluso después de que el programa se haya cerrado. Esto significa que la información almacenada en la base de datos puede ser recuperada y utilizada en futuras ejecuciones.
* Escalabilidad: Nos permite manejar grandes volúmenes de datos y escalar a medida que la aplicación crece y se agregan nuevas funcionalidades.
* Facilidad de acceso: Se proporciona una interfaz para acceder y manipular los datos almacenados. Esto facilita la gestión y manipulación de los datos, simplificando el diseño y la implementación de la lógica de negocio en el programa.
* Seguridad de los datos: Ofrecen mecanismos que garantizan la integridad y seguridad de los datos, como la encriptación o las copias de seguridad.
* Flexibilidad: Son fácilmente adaptables a cualquier tipo de necesidad y aplicación.

### NETBEANS

Ambiente de desarrollo integrado o IDE, que ofrece una serie de herramientas útiles para programar en lenguajes de programación orientada a objetos, tales como Java, C++ y Ruby.

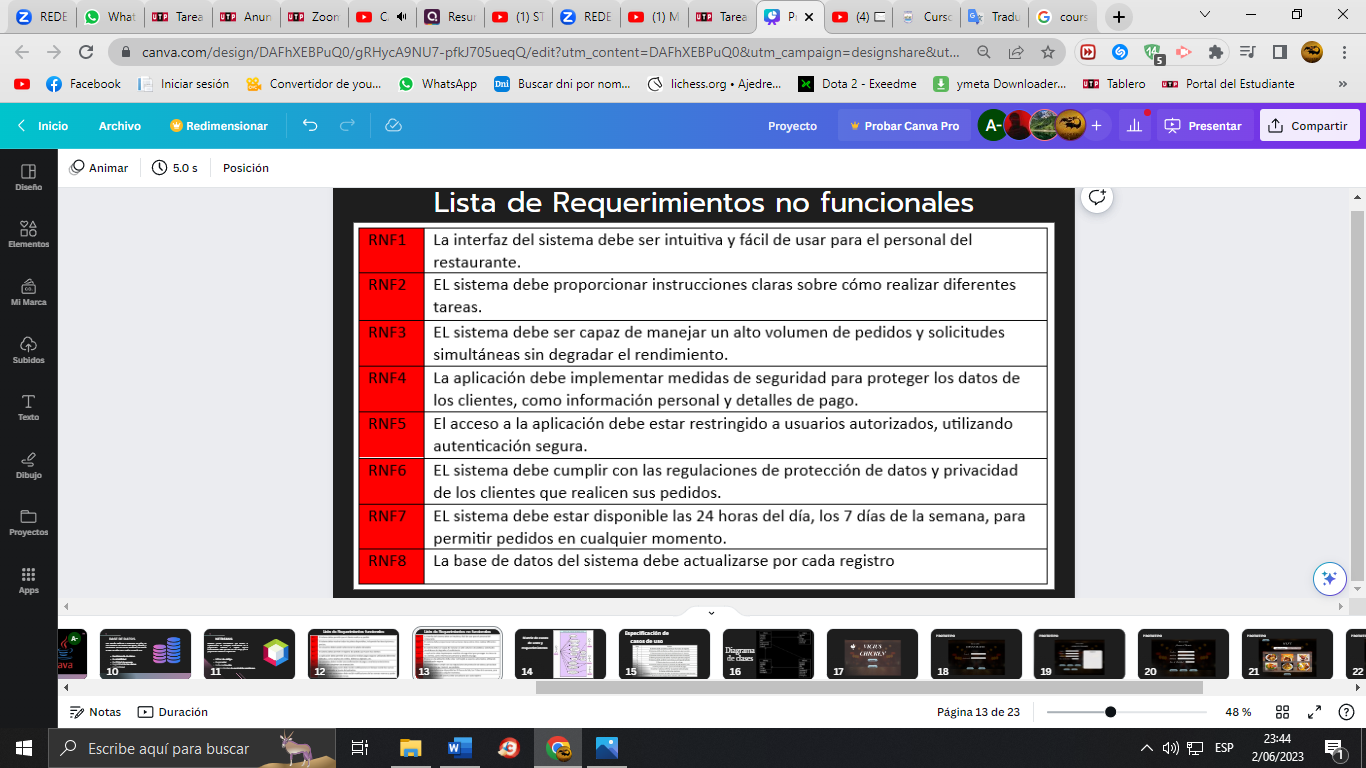
NetBeans provee características que facilitan la implementación de conceptos fundamentales de la programación orientada a objetos. Algunas funcionalidades útiles que esta IDE nos otorga para la programación orientada a objetos son:

* Editor de código: Facilita la escritura de código en lenguajes de POO, ofreciendo resaltado de sintaxis, autocompletado de código y sugerencia de correcciones, entre otros.
* Depurador: NetBeans incluye un depurador que ayuda a identificar y corregir errores en el código, hacer seguimiento de la ejecución del programa, inspeccionar el valor de las variables y establecer puntos de integración en el código.
* Refactorización: Proceso importante que permite mejorar la calidad del código, facilitan la reorganización y modificaciones del código sin afectar su funcionalidad.
* Herramientas de control de versiones: Tales como Git o Subversion, permiten trabajar en proyectos de manera colaborativa y coordinada.

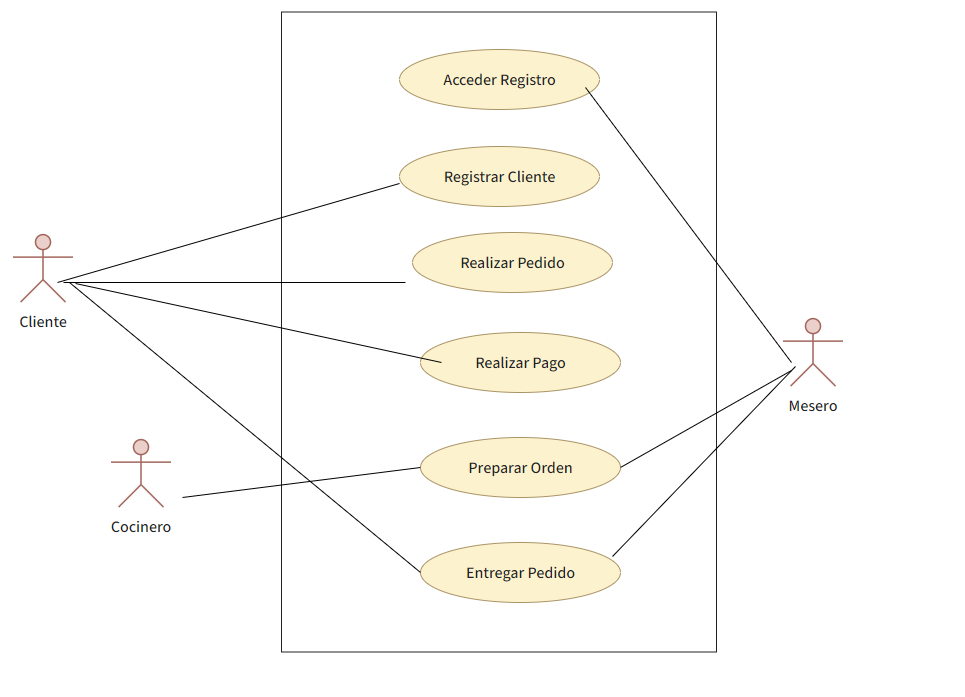
CAPÍTULO 3

# **Lista de requerimientos funcionales**

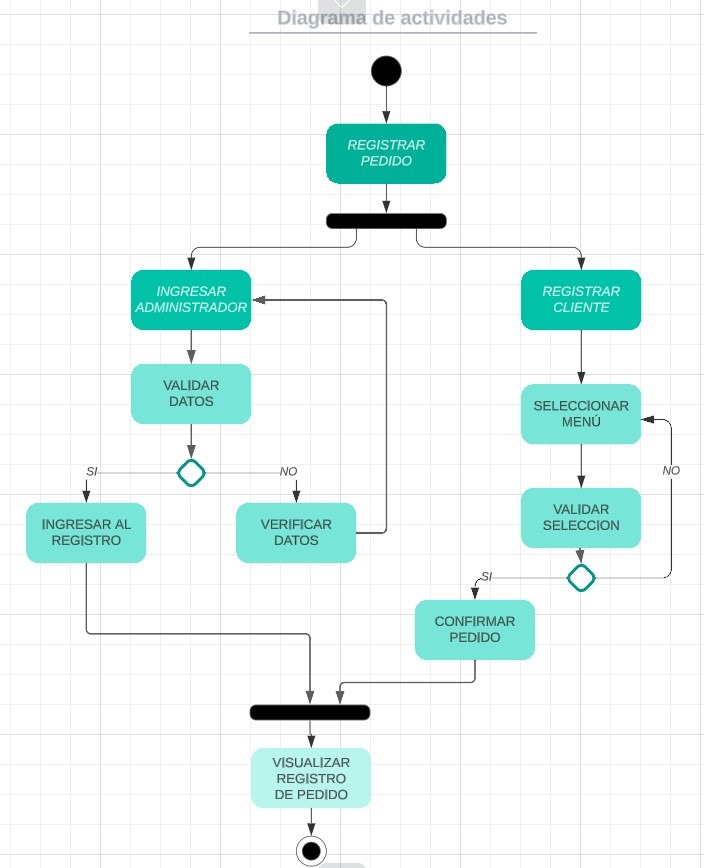
# **Lista de requerimientos no funcionales**



# **Diagrama de casos de uso**



# **Diagrama de Actividades**



# **Especificaciones de los casos de uso**

|  |  |
| --- | --- |
| Identificador | DCU-1 |
| Nombre | Acceder registro |
| Descripción | Permite al mesero acceder a los registros de clientes |
| Actores | Mesero |
| Precondiciones | El mesero cuenta con un usuario registrado. |
| Postcondiciones | Se muestra todos los clientes registrados del día. |
| Flujo normal de eventos | |
| El mesero inicia con el proceso de ingreso.  El sistema le solicita su usuario y contraseña.  El sistema valida los datos.  El sistema muestra la ventana de registro de clientes. | |
| Flujos alternos y excepciones | |
| El mesero no tiene usuario.   * Solicitar al administrador un usuario registrado.   El mesero ingresa sus datos incorrectos.   * El sistema le indica un mensaje de error. | |

|  |  |
| --- | --- |
| Identificador | DCU-2 |
| Nombre | Registrar Cliente |
| Descripción | Permite el registro de los clientes. |
| Actores | Cliente |
| Precondiciones | El cliente tiene acceso al sistema de pedido. |
| Postcondiciones | Se ha realizado el registro del cliente exitosamente. |
| Flujo normal de eventos | |
| El cliente le solicita al sistema comenzar el proceso de registro.  El sistema solicita al cliente sus datos necesarios para acceder.  El cliente proporciona los datos solicitados por el sistema.  El sistema continua con el proceso mostrándole el menú a seleccionar. | |
| Flujos alternos y excepciones | |
| El cliente ingresa términos inválidos en el nombre. | |

|  |  |
| --- | --- |
| Identificador | DCU-3 |
| Nombre | Realizar pedido |
| Descripción | Permite realizar la orden del menú a los clientes. |
| Actores | Cliente |
| Precondiciones | El cliente ya se ha registrado en el sistema exitosamente. |
| Postcondiciones | Se ha registrado la orden del cliente y muestra el monto. |
| Flujo normal de eventos | |
| El sistema solicita al cliente que seleccione el menú.  El cliente escoge la opción a adquirir.  El sistema solicita confirmar la opción seleccionada.  El cliente confirma el pedido.  El sistema muestra el monto a pagar. | |
| Flujos alternos y excepciones | |
|  | |

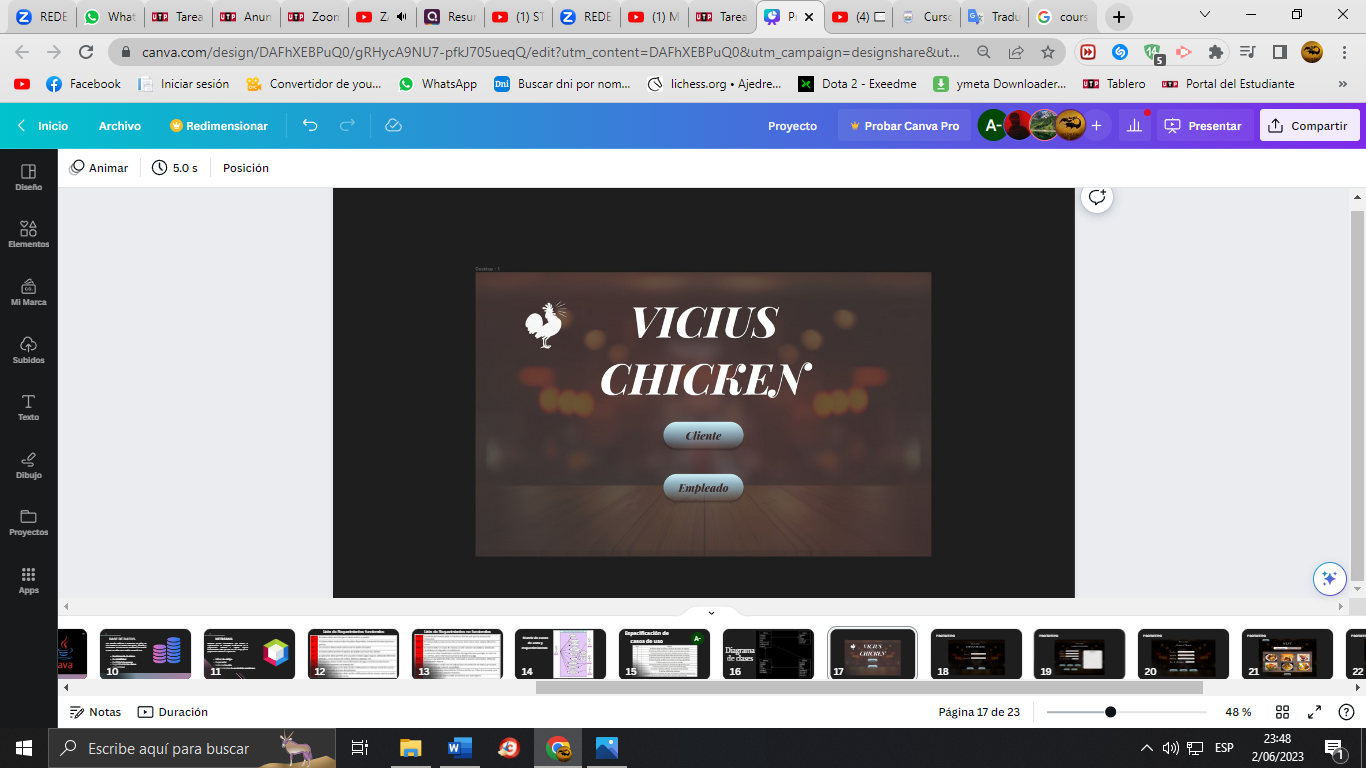
|  |  |
| --- | --- |
| Identificador | DCU-4 |
| Nombre | Realizar pago |
| Descripción | Permite el pago |
| Actores | Cliente |
| Precondiciones | El cliente ya confirmó su pedido. |
| Postcondiciones | El pedido es cancelado y se genera boleta. |
| Flujo normal de eventos | |
| El sistema muestra los métodos de pago.  El cliente escoge el método de pago.  El sistema solicita la confirmación.  El cliente confirma y realiza el pago.  El sistema confirma el pago.  El sistema entrega boleta. | |
| Flujos alternos y excepciones | |
| El cliente desea anular el pedido luego del pago.   * Debe solicitar ayuda del personal. | |

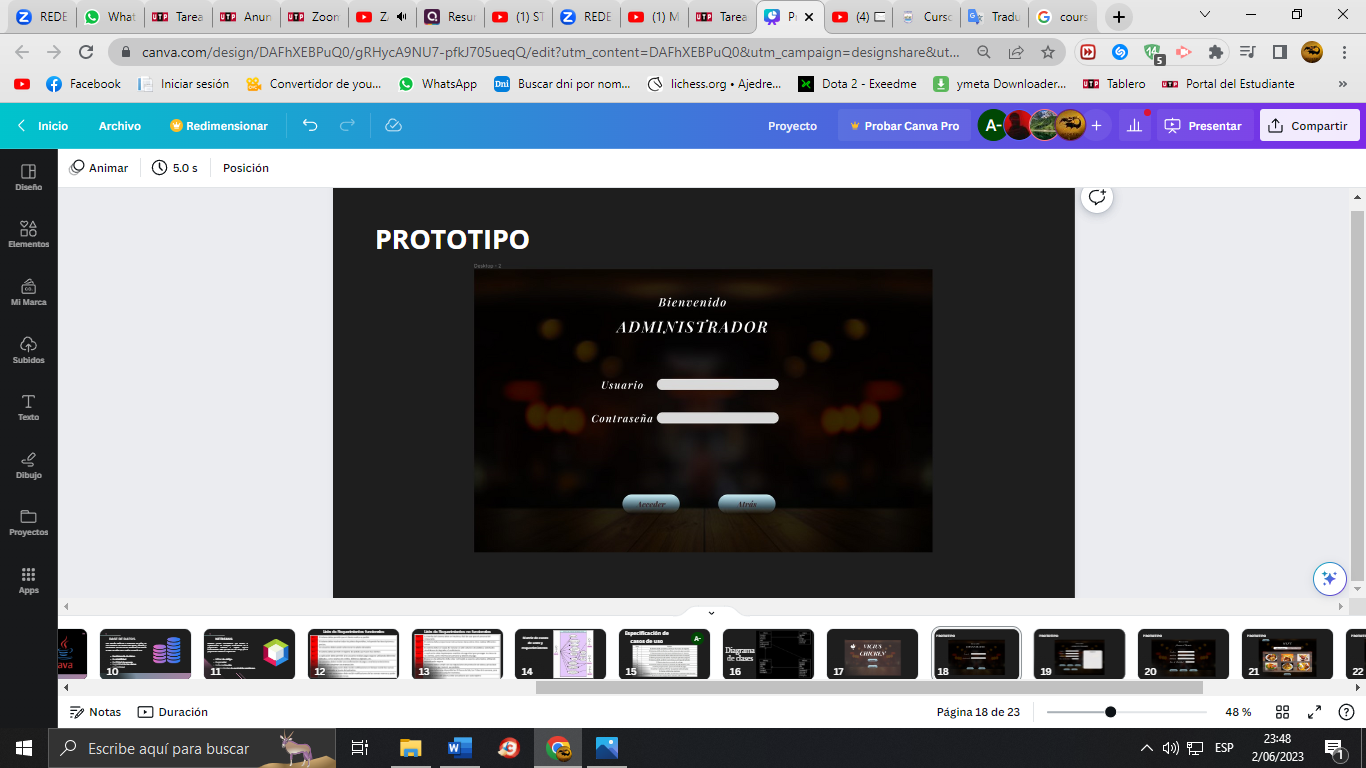
|  |  |
| --- | --- |
| Identificador | DCU-5 |
| Nombre | Preparar orden. |
| Descripción | El cocinero se informa del orden y procede a prepararlo. |
| Actores | Mesero, Cocinero |
| Precondiciones | El mesero debe obtener la boleta del cliente. |
| Postcondiciones | El mesero recibe la orden lista. |
| Flujo normal de eventos | |
| El mesero otorga la boleta con la orden.  El cocinero recibe y prepara la orden.  El cocinero entrega la orden lista. | |
| Flujos alternos y excepciones | |
|  | |

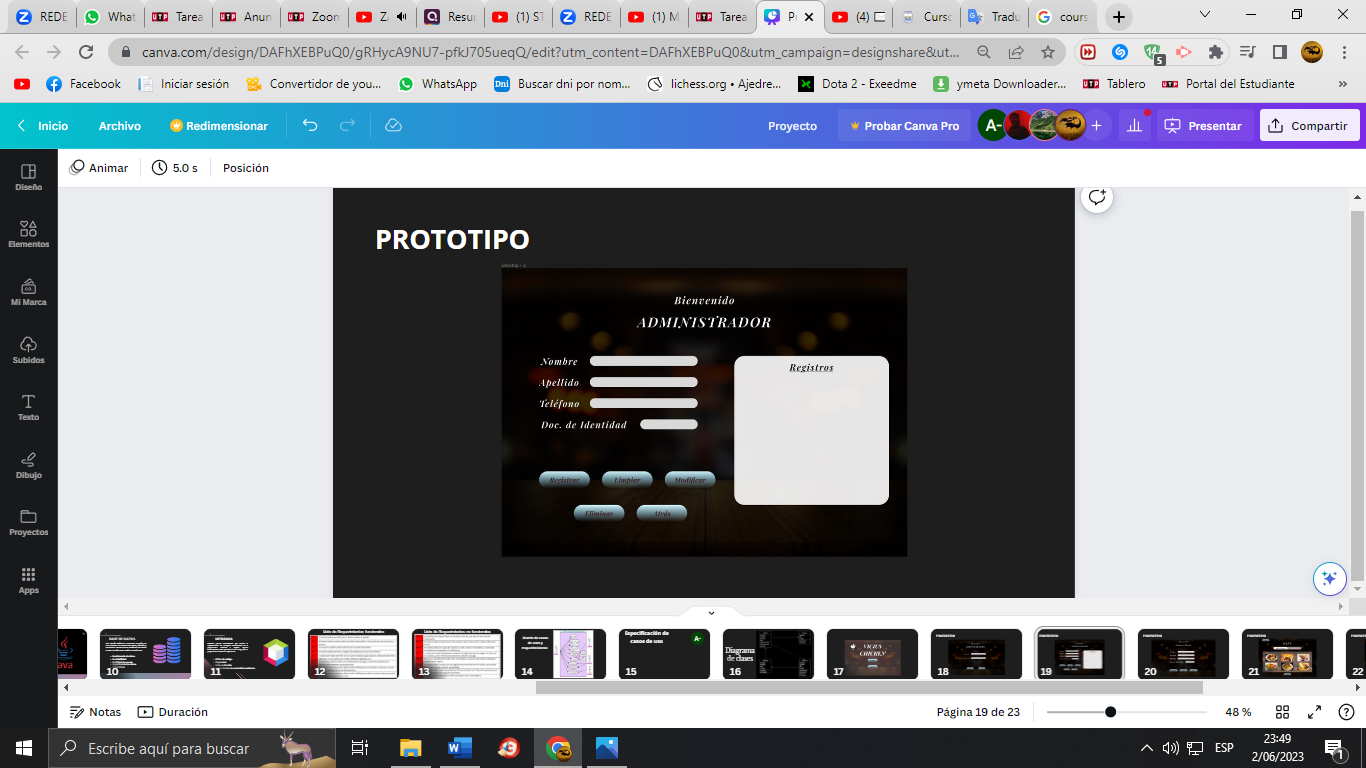
|  |  |
| --- | --- |
| Identificador | DCU-6 |
| Nombre | Entregar pedido |
| Descripción | El pedido es finalmente entregado al cliente. |
| Actores | Cliente, Mesero |
| Precondiciones | El mesero tiene la orden lista. |
| Postcondiciones | El cliente recibe satisfactoriamente la orden. |
| Flujo normal de eventos | |
| El mesero recibe la orden lista.  El mesero se dirige a la mesa correspondiente.  El cliente recibe finalmente la orden. | |
| Flujos alternos y excepciones | |
|  | |

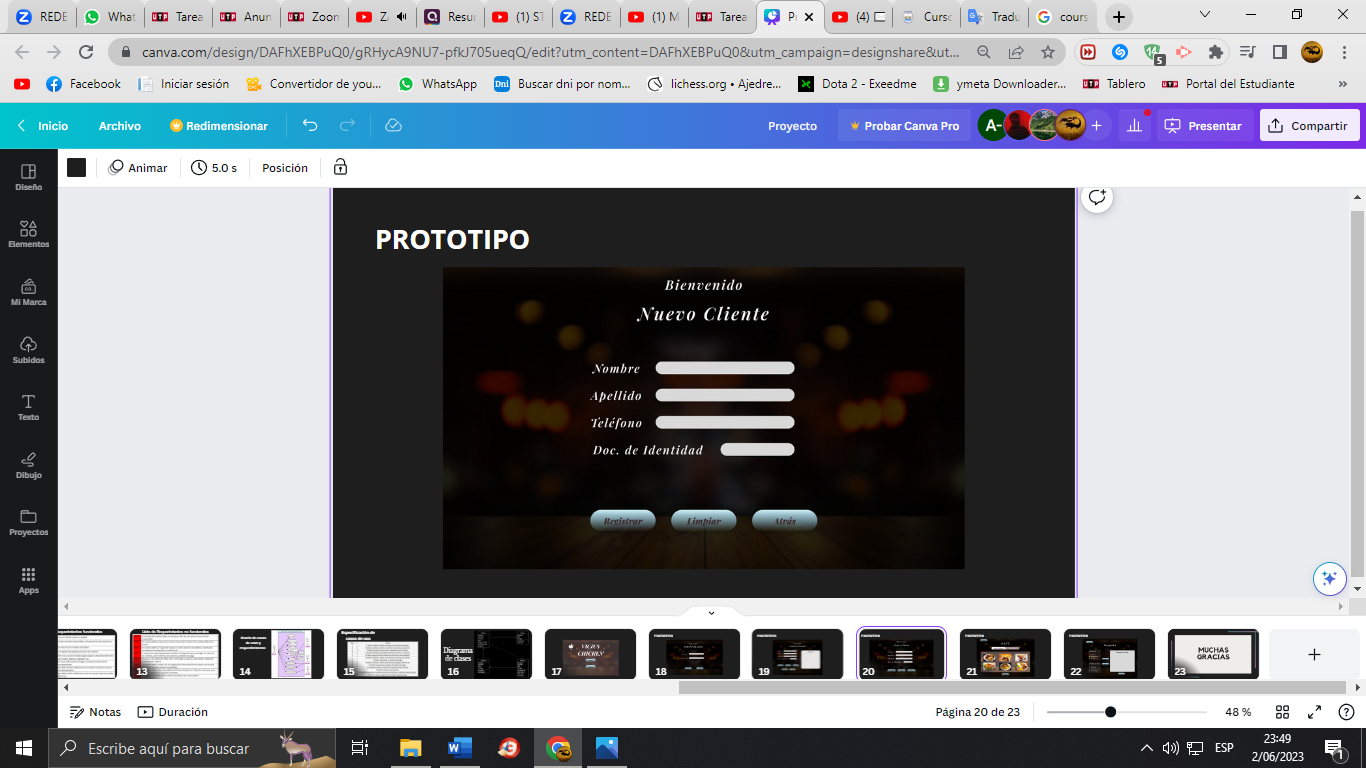
# **Diagrama de clases**

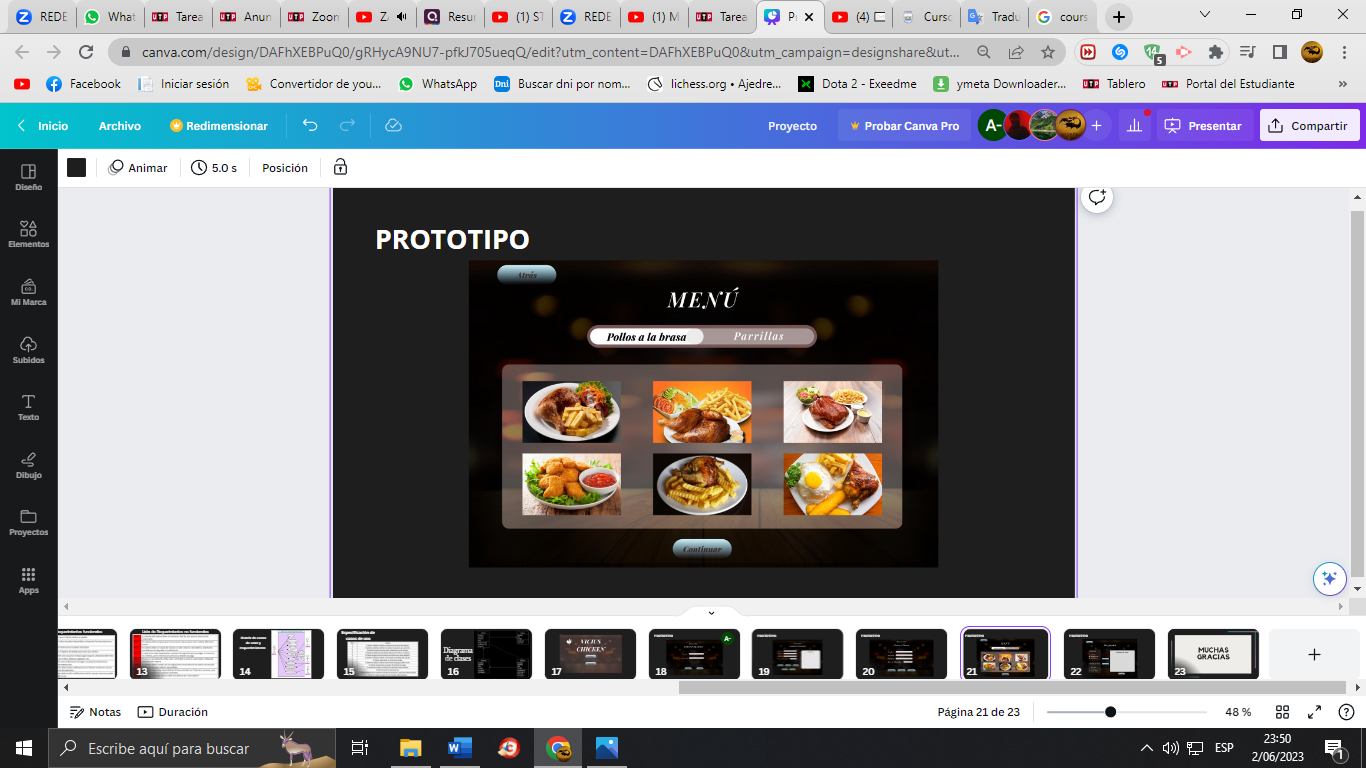
# **Prototipo de Referencia**









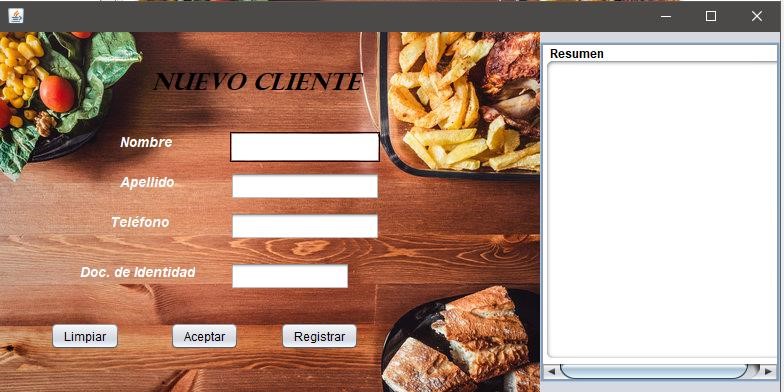


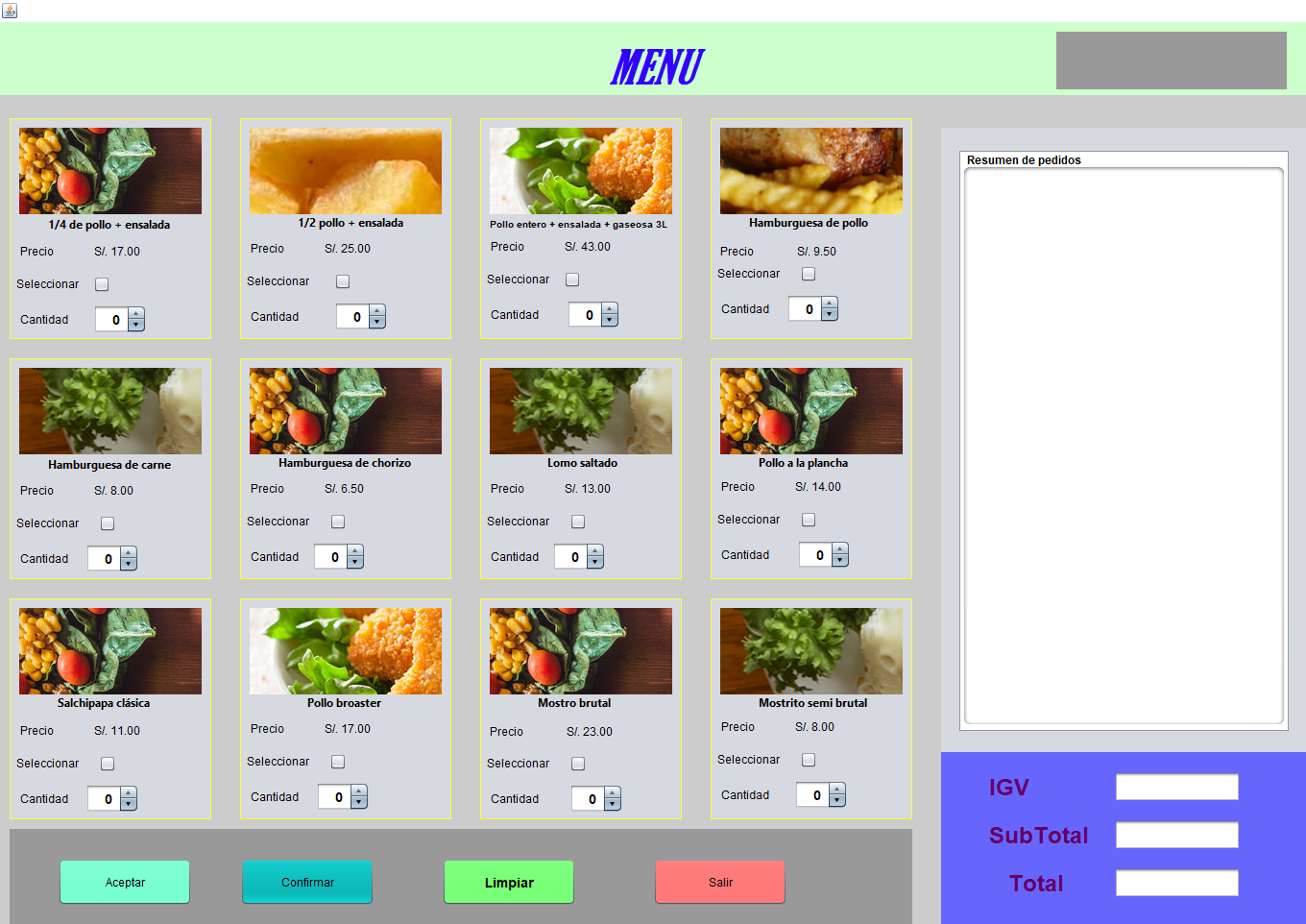
# **Project Charter**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Project Charter | | | | | | | |
| Titulo del Proyecto | Sistema de registros de ordenes  Para un restaurante | | | | Lider del Proyecto | | Cristian Casahuaman |
| Inicio del proyecto | 20 de Abril 2023 | Fin del proyecto | | | 22 de julio de 2023 | | |
| Descripcion del Proyecto | | | | | | | |
| Desarrollar un programa la cual determina que el cliente pueda realizar su orden más rapido y que ademas no tenga la necesidad de hacer una larga espera. Además, el sistema pueda ser usada de forma muy sencilla de manejar, en el cual se puedan realizar pedidos en menor tiempo. | | | | | | | |
| Alcances | | | Limitaciones | | | | |
| Implementar un sistema de registros. | | | Complicaciones para personas de la tercera edad | | | | |
| Finanzas | | | | | | | |
| Presupuesto del proyecto completado: 1000 soles. | | | | | | | |
| Equipo del proyecto | | | | | | | |
| Lider del proyecto | Cristian Casahuaman | | | Anlista de sistemas | | Marcela Vera | |
| Analista de prog. | Tony Cunurana | | | Programador | | Cristian Casahuaman | |
| Tester | Bill Trujillo | | |  | |  | |

# **Prototipo en desarrollo**

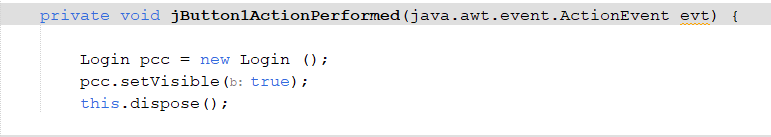




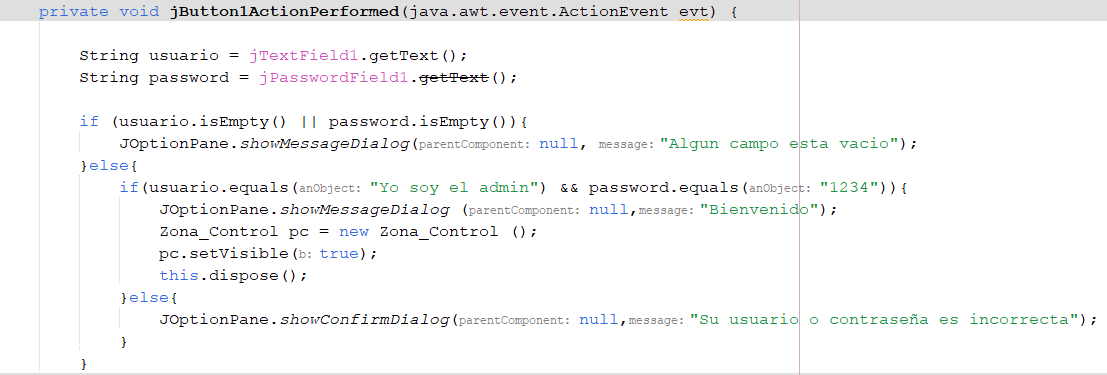


# **Código del proyecto**

## **Principal**



## **Login**



## **Registrar Cliente**

## **Selección Menú**

Programación de los jCheckbox (Seleccion1-12) en función al estado del JSpinner.

Si no hay ningún dato ingresado en el Spinner, el programa alertará la ausencia del dato.

Método propina para el cálculo del mismo.

Método detalles para mostrar los resultados en los jTexField inferiores.

Método resumen para mostrar los mismo resultados en el jTextArea.



El botón Aceptar mostrara el cálculo de los datos en el jTextArea, simulando la impresión del ticket.



Método reset, será llamado por el botón Limpiar y borrará todos los datos seleccionados actualmente.

Canva:

<https://www.canva.com/design/DAFoBXoiOXY/I8xRwieWIqy-TR0lzHAUzw/edit?utm_content=DAFoBXoiOXY&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton>