



ALUMNO: ÁNGEL ALAN GONZÁLEZ ESCALONA

MATERIA: ARQUITECTURA DE SOFTWARE

PROFESORA: RODRÍGUEZ FLORES JAZMIN

ESCUELA: UNIVERSIDAD POLITÉCNICA DE PACHUCA

INGENIERA EN: SOFTWARE

GRUPO: 02

TITULO DEL TRABAJO: SOFTWARE PARA RESTAURANTE

FECHA DE ENTREGA: 24/05/2022

Contenido

Introducción	1
Objetivo general	1
Objetivos específicos.....	1
Metodología de desarrollo	2
Justificación.....	3
Comunicación	4
Listado de requisitos:	4
Planeación	7
Diagrama de Gantt:.....	7
Presupuesto.....	9
Riesgos	9
Planes de contingencia.....	14
Modelado	20
Representación de Casos de uso	20
Herramientas y lenguajes por usar	21
Diccionario de datos	22
Patrones de diseño	24
Principio de diseño.....	27
Arquitectura utilizada	28
Construcción.....	31
Código	31
Programación de Cliente Servidor	39
Prueba.....	39
Despliegue.....	40
Entrega.....	40
Asistencia.....	40
Evaluación.....	40
Conclusión	40
Lecciones aprendidas.....	41

Introducción

En el presente proyecto se desea desarrollar un sistema para gestionar un establecimiento de comida el cual permita administrar, controlar, operar de una manera más ágil y segura, el sistema podrá administrar y registrar usuarios los cuales se almacenarán en una base datos así mismo brindará una mayor agilidad en los procesos de cobro además que permitirá un mejor control sobre la información que se maneje en el establecimiento; así mismo sistema está enfocado específicamente para la empresa que en este caso sería la fonda

El proyecto será documentado a través de la metodología cascada basada en la adaptación de Pressman, el cual contendrá los procesos que se llevarán a cabo para el desarrollo del software así mismo se anexará los casos de uso, diagramas, descripción detallada de requisitos y funcionamiento del programa todo esto para una mejor comprensión de proyecto

Objetivo general

Hacer más óptimo los diversos procesos realizan en la fonda de manera local, proporcionando un mejor desempeño de sus meseros y el establecimiento en general, en cuestión de las ventas y cobros, así poder agilizar el proceso de atención al cliente y menores pérdidas económicas para los diferentes ámbitos a desarrollar.

Objetivos específicos

- Crear una interfaz atractiva y sencilla de usar para la comodidad del usuario.
- Analizar las necesidades y peticiones del cliente.
- Comprender lo que se necesita para la creación del sistema deseado.
- Generar una factura en PDF
- Incluir una calculadora con funciones básicas
- Registrar usuarios y meseros

Metodología de desarrollo

La metodología que se usará en el proyecto fonda será la de cascada basada en Pressman (2005), ya que nuestro plan para este proyecto es dividir las etapas del proceso para el desarrollo del software, para que de dicha forma tengamos que terminar el primer paso y que sea funcional para acceder o trabajar con la otra etapa. Esta metodología nos sugiere un enfoque sistemático y secuencial hacia el desarrollo del programa que nos indica los procesos y etapas, estas son:

- Comunicación
- Planeación
- Modelado
- Construcción
- Despliegue

A pesar de estas etapas existe una variedad de propuestas de proceso de software que nos ayuda en nuestro proyecto, existe un conjunto de actividades fundamentales que se encuentran presentes en todos ellos.

- Especificación de software: Se debe definir la funcionalidad y restricciones operacionales que debe cumplir el software.
- Diseño e Implementación: Se diseña y construye el software de acuerdo a la especificación.
- Validación: El software debe validarse, para asegurar que cumpla con lo que quiere el cliente.
- Evolución: El software debe evolucionar, para adaptarse a las necesidades del cliente.

Además de estas actividades fundamentales, Pressman menciona un conjunto de “actividades protectoras”, que se aplican a lo largo de todo el proceso del software:

- Seguimiento y control de proyecto de software.
- Revisiones técnicas formales.
- Garantía de calidad del software.
- Gestión de configuración del software.

- Preparación y producción de documentos.
- Gestión de reutilización.
- Mediciones.
- Gestión de riesgos.

Y para finalizar el proceso del proyecto, Pressman caracteriza un proceso de desarrollo de software con los siguientes elementos:

Un marco común del proceso, definiendo un pequeño número de actividades del marco de trabajo que son aplicables a todos los proyectos de software, con independencia del tamaño o complejidad.

Un conjunto de tareas, cada uno es una colección de tareas de ingeniería en software, hitos de proyectos, entregas y productos de trabajo del software, y puntos de garantía de calidad, que permiten que las actividades del marco de trabajo se adapten a las características del proyecto de software y los requisitos del equipo del proyecto.

Las actividades de protección, tales como garantía de calidad del software, gestión de configuración del software y medición, abarcan el modelo del proceso. Las actividades de protección son independientes de cualquier actividad del marco de trabajo y aparecen durante todo el proceso.

Justificación

El modelo de cascada nos permite estructurar la organización de forma clara en este proyecto que se va a desarrollar con las diversas fases del proyecto, se diferencian claramente entre sí. Como cada una de las fases concluye con un hito, el proceso de desarrollo es muy fácil de comprender para todo el equipo. El punto clave del modelo reside en la documentación de todos y cada uno de los pasos del proceso. Los conocimientos adquiridos se registran en pliegos de requisitos o borradores preliminares. El modelo en cascada nos prevé la realización de ajustes a lo largo del proyecto. Sin embargo, el proyecto de software en el que vamos a trabajar nos detalla que el desarrollo se define al comienzo, solo podría concluir con éxito si desde el principio se invirtiera una gran cantidad de tiempo, análisis de la documentación y el diseño.

Comunicación

Listado de requisitos:

Número de requisito	RF01
Nombre de requisito	Inicio de Sesión
Tipo	Requisito Restricción
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/Opcional
Descripción detallada	El sistema será capaz de controlar el acceso a las funciones del mismo a través de la verificación de una cuenta con la utilización de diferentes tipos de usuarios

Número de requisito	RF01_1
Nombre de requisito	Usuario tipo Cliente
Tipo	Requisito Restricción
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/Opcional
Descripción detallada	El sistema tendrá un tipo de usuario llamado cliente quien realiza la solicitud del pedido además de tener la opción de generar e imprimir la factura del pedido solicitado

Número de requisito	RF01_2
Nombre de requisito	Usuario tipo Mesero
Tipo	Requisito Restricción
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/Opcional
Descripción detallada	El sistema tendrá un tipo especial de usuario llamado mesero que será capaz de utilizar las funciones de la calculadora

Número de requisito	RF02
Nombre de requisito	CRUD de usuarios
Tipo	Requisito Restricción
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/Opcional
Descripción detallada	El administrador tendrá la capacidad de controlar la alta, la modificación y la eliminación de los usuarios dentro del sistema

Número de requisito	RF03
Nombre de requisito	Menú de productos
Tipo	Requisito Restricción
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/Opcional
Descripción detallada	El cliente hará un pedido seleccionando de los alimentos disponibles para que posteriormente el mesero genere dicho pedido a través del sistema así mismo se menciona que si se cuenta con permisos de administrador se podrá hacer un CRUD con los alimentos y bebidas

Número de requisito	RF04
Nombre de requisito	Generar factura
Tipo	Requisito Restricción
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/Opcional
Descripción detallada	El cliente tendrá la opción de solicitar la factura del pedido realizado con los siguientes datos: nombre del establecimiento, fecha, cantidad de productos, nombre del cliente, descripción, precio, RFC y monto total

Número de requisito	RF05
Nombre de requisito	CRUD de pedidos
Tipo	Requisito Restricción
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/Opcional
Descripción detallada	El sistema tendrá las funcionalidades de agregar, modificar, actualizar y eliminar pedidos

Número de requisito	RNF01
Nombre de requisito	Fluidez continua
Tipo	Requisito Restricción
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/Opcional
Descripción detallada	El sistema deberá funcionar de una manera fluida en el tiempo en que el usuario utilice el sistema

Número de requisito	RNF02
Nombre de requisito	Entrada restringida
Tipo	Requisito Restricción
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/Opcional
Descripción detallada	El sistema deberá asegurar que los datos esten protegidos del acceso no autorizado

Número de requisito	RNF03
Nombre de requisito	Protección de datos
Tipo	Requisito Restricción
Prioridad del requisito	Alta/Esencial Media/Deseado Baja/Opcional
Descripción detallada	El sistema no revelara a sus operadores no autorizados otros datos de los clientes distintos a nombres y números de referencia

Planeación

Diagrama de Gantt:

N. Actividad	Actividad a Elaborar	Responsable de la Actividad	Fecha de inicio	Fecha de fin
1	Introducción	Angel Alan González Escalona	22/03/2020	23/03/2020
2	Objetivo general	Jezer Alfredo Pérez cruz	22/03/2020	24/03/2020
3	Objetivos específicos	Citlaly Alejandra Hernández Cuevas	22/03/2020	24/03/2020
4	Comunicación			
	Listado de requisitos	Vicente Ledezma Rojo	23/03/2020	24/01/1900
5	Planeación			
	Diagrama de actividades	Angel Alan González Escalona Jezer Alfredo Pérez cruz	24/03/2021	25/03/2021
6	Presupuestos			
	Riesgos	Todo el equipo	01/04/2021	05/04/2021
	Planes de contingencia	Todo el equipo	02/04/2021	05/04/2021
7	Modelado			
	Representación de caso de uso administrador	José De Jesús Damián Hernández Ángel Alan González Escalona Jezer Alfredo Pérez cruz	24/03/2021	14/04/2021
	Representación de caso de uso mesero	Vicente Ledezma Rojo Jezer Alfredo Pérez cruz	24/03/2021	14/04/2021
	Representación de caso de uso cliente	Citlaly Alejandra Hernández Cuevas Vicente Ledezma Rojo	24/03/2021	14/04/2021
	Herramientas a usar	Angel Alan González Escalona		30/03/2021
	Modelo base de datos	Angel Alan González Escalona	28/03/2021	30/03/2021
	Diccionario de Datos	José De Jesús Damián Hernández	29/03/2021	30/03/2021
	Patrones de diseño	José De Jesús Damián Hernández	30/03/2021	30/03/2021
	Principios de diseño	Citlaly Alejandra Hernández Cuevas	31/03/2021	31/03/2021
	Arquitectura utilizada	Jezer Alfredo Pérez cruz	31/03/2021	31/03/2021
8	Construcción			
	código	Todo el equipo	05/04/2021	12/04/2021
	prueba	Angel Alan González Escalona	12/04/2021	13/04/2021
9	Despliegue			
	Entrega	Vicente Ledezma Rojo	12/04/2021	12/04/2021
	Asistencia	Citlaly Alejandra Hernández Cuevas	14/04/2021	14/04/2021
	Evaluación	Jezer Alfredo Pérez cruz	14/04/2021	14/04/2021
10	Conclusión			
	Lecciones aprendidas	Todo el equipo	14/04/2021	14/04/2021

TEMA	22/03/2021	23/03/2021	24/03/2021	25/03/2021	26/03/2021	27/03/2021	28/03/2021	29/03/2021	30/03/2021	31/03/2021	01/04/2021
INTRODUCCION											
OBJETIVO GENERAL											
OBJETIVO ESPECIFICO											
COMUNICACIÓN											
PLANEACION											
PRESUPUESTO											
MODELADO											
CONSTRUCCION											
DESPLIEGUE											
CONCLUSION											

Presupuesto

Riesgos

1. Mala estimación de tiempos

Atributo	Valor
Impacto	Crítico
Probabilidad	Muy probable.
Marco de tiempo	Mediano Plazo
Tipo de Riesgo	Técnico

2. Diseño Inadecuado

Atributo	Valor
Impacto	Catastrófico
Probabilidad	Poco probable.
Marco de tiempo	Largo plazo
Tipo de Riesgo	Técnico

3. Falta de organización

Atributo	Valor
Impacto	Crítico
Probabilidad	Poco probable.
Marco de tiempo	Corto plazo
Tipo de Riesgo	Del Proyecto

4. Falta de informes al cliente

Atributo	Valor
Impacto	Catastrófico.
Probabilidad	Improbable
Marco de tiempo	Corto Plazo
Tipo de riesgo	Técnico

5. Escasez del hardware necesario

Atributo	Valor
Impacto	Catastrófico.
Probabilidad	Poco probable
Marco de tiempo	Corto Plazo
Tipo de riesgo	Del proyecto

6. Desarrolladores inexpertos para la construcción del programa

Atributo	Valor
Impacto	Catastrófico.
Probabilidad	Poco probable
Marco de tiempo	Mediano Plazo
Tipo de riesgo	Del proyecto

7. Falta de manejo de la metodología a emplear

Atributo	Valor
Impacto	Catastrófico.
Probabilidad	Poco probable
Marco de tiempo	Corto Plazo
Tipo de error	Del proyecto

8. Conflictos entre los miembros del equipo.

Atributo	Valor
Impacto	Catastrófico.
Probabilidad	Probable.
Marco de tiempo	Mediano Plazo
Tipo de riesgo	Del Proyecto

9. Requerimientos mal definidos.

Atributo	Valor
Impacto	Catastrófico.
Probabilidad	Probable.
Marco de tiempo	Mediano Plazo
Tipo de Riesgo	Técnico

10. Integrante abandona el proyecto.

Atributo	Valor
Impacto	Crítico.
Probabilidad	Improbable
Marco de tiempo	Largo Plazo.
Tipo de Riesgo	Del Proyecto

11. Mala calidad de software.

Atributo	Valor
Impacto	Catastrófico.
Probabilidad	Improbable
Marco de tiempo	Largo Plazo.
Tipo de Riesgo	Técnico

12. Integrantes con problemas de motivación

Atributo	Valor
Impacto	Critico
Probabilidad	Probable
Marco de tiempo	Corto Plazo.
Tipo de riesgo	Impredecible

13. Cambio de lenguaje con el que se implementa el software

Atributo	Valor
Impacto	Catastrófico.
Probabilidad	Improbable
Marco de tiempo	Largo Plazo.
Tipo de riesgo	Técnico

14. Integrante se infecta de COVID-19

Atributo	Valor
Impacto	Catastrófico.
Probabilidad	Muy probable
Marco de tiempo	Medio Plazo.
Tipo de riesgo	Impredecible

Planes de contingencia

Descripción: Mala estimación de tiempos

Impacto: Crítico.

Probabilidad: Muy probable

Marco de tiempo: Mediano plazo

Plan de contingencia:

- Apresurar el trabajo y entregar lo antes posible
- Evitar distracciones

Descripción: Diseño Inadecuado

Impacto: Catastrófico

Probabilidad: Poco probable

Marco de tiempo: Largo plazo

Plan de contingencia:

- Consulta con el cliente
- Modificación del diseño

Descripción: Falta de organización

Impacto: Crítico

Probabilidad: Poco probable

Marco de tiempo: Corto plazo

Plan de contingencia:

- Reunión del equipo de trabajo
- Realizar un cronograma de actividades
- Aclarar dudas respecto a las actividades a realizar

Descripción: Falta de informes al cliente

Impacto: Catastrófico.

Probabilidad: Improbable.

Marco: Corto Plazo.

Plan de Contingencia:

- Corroborar el conocimiento del cliente acerca del proyecto
- Entregar la información faltante al cliente
- Vigilar que se le informe correctamente al cliente

Descripción: Escasez del hardware requerido

Impacto: Catastrófico.

Probabilidad: Poco probable.

Marco: Corto Plazo.

Plan de Contingencia:

- Delimitar la cantidad de hardware faltante
- Conseguir los recursos faltantes
- Asegurarse de satisfacer la necesidad en su totalidad

Descripción: Desarrolladores inexpertos para la construcción del programa

Impacto: Catastrófico.

Probabilidad: Poco probable.

Marco: Mediano Plazo.

Plan de Contingencia:

- Evaluar la inexperiencia de los desarrolladores
- Decidir sobre contratar a otros trabajadores o capacitarlos.
- Ajustar la planeación del proyecto

Descripción: Falta de manejo de la metodología a emplear

Impacto: Catastrófico.

Probabilidad: Poco probable.

Marco: Corto Plazo.

Plan de Contingencia:

- Examinar el nivel aproximado de dominio que se tiene de la metodología
- Capacitar a los trabajadores en el uso de la metodología

Descripción: Conflictos entre los miembros del equipo.

Impacto: Catastrófico.

Probabilidad: Probable.

Marco: Mediano Plazo.

Plan de Contingencia:

- Aceptar el problema y animar a encontrar la solución.
- Buscar una solución inmediata y eficaz.
- Orientar de manera asertiva las reuniones o discusiones.

Descripción: Requerimientos mal definidos.

Impacto: Catastrófico.

Probabilidad: Probable.

Marco: Mediano Plazo.

Plan de Contingencia:

- Realizar una entrevista de nuevo con el cliente.
- Aclarar los requisitos mal definidos.

Descripción: Integrante abandona el proyecto.

Impacto: Crítico.

Probabilidad: Improbable.

Marco: Corto Plazo.

Plan de Contingencia:

- Conseguir a otra persona para el puesto.
- Entrevistarle y analizar si es apto para el puesto.
- Hacerle una propuesta de contrato.

Descripción: Mala calidad de software.

Impacto: Catastrófico.

Probabilidad: Improbable.

Marco: Largo Plazo.

Plan de Contingencia:

- Seguir el proceso correcto.
- Involucrar a las personas adecuadas.
- Exigir los resultados correctos.

Descripción: Integrantes con problemas de motivación

Impacto: Catastrófico

Probabilidad: Probable

Marco: Corto Plazo

Plan de Contingencia:

- Darle 1 o 2 días de descanso a la persona para reflexionar.
- Hablar con él/ella para poder entender la desmotivación.
- Hacer que hable con alguien sobre su problema.
- Intentar motivar al integrante de alguna forma diferente a la que tenía.

Descripción: Cambio de lenguaje con el que se implementará el software

Impacto: Catastrófico

Probabilidad: Improbable

Marco: Largo Plazo

Plan de Contingencia:

- Aumentar las horas de trabajo para el proyecto.
- Dar prioridad a las funciones principales y abandonar un poco el **diseño**.
- Ocupar código de algunas otras fuentes para agilizar el proyecto.
- Más horas de estudio sobre el nuevo lenguaje.
- Intentar pedir un poco más de tiempo al cliente.

Descripción: Integrante de infecta de COVID-19

Impacto: Catastrófico

Probabilidad: Muy probable

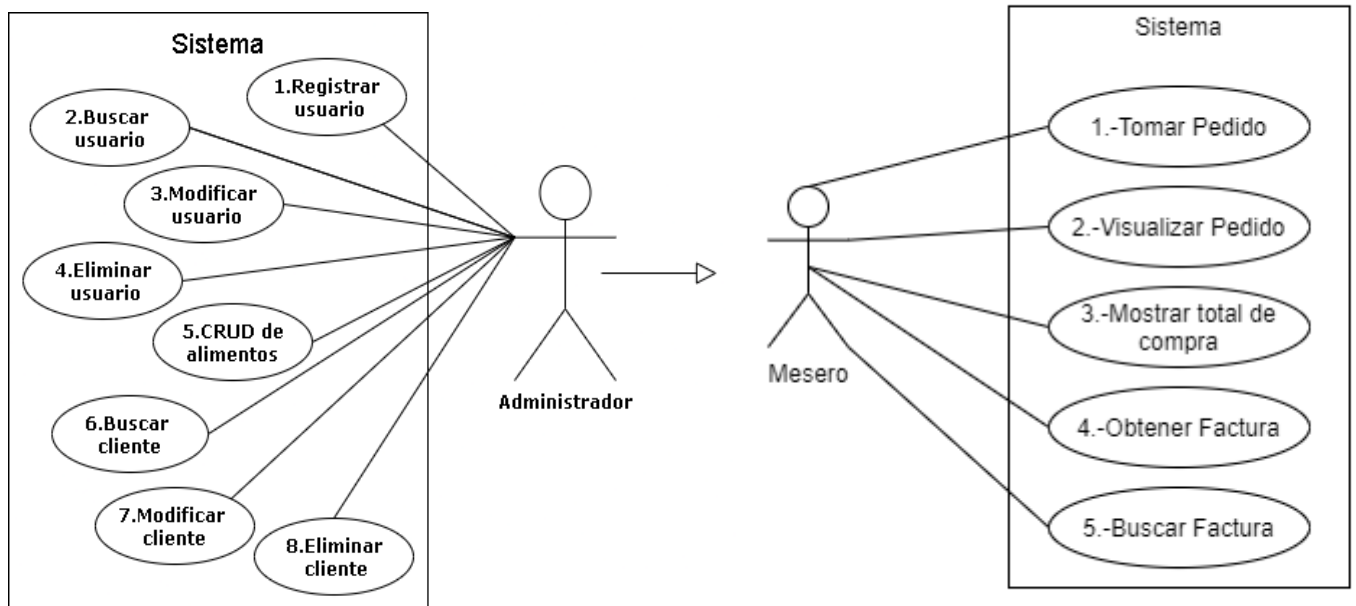
Marco: Medio Plazo

Plan de Contingencia:

- Dependiendo del individuo saber si puede trabajar y darle menos horas de jornada.
- Su trabajo se dividirá entre todo el equipo.
- El proyecto se entregará con tiempo tardío.
- Se le dará a conocer la situación al cliente del problema.
- Intentar pedir tiempo al cliente por dicha situación.
- Tener una conversación del equipo si es factible hacer el proyecto con esta situación.
- Dar a conocer esta situación a cualquiera que esté involucrado con el proyecto.

Modelado

Representación de Casos de uso



Herramientas y lenguajes por usar

En el presente proyecto se usará el entorno de desarrollo NetBeans versión 8.2 y se implementará el plugin iReport el cual es un diseñador de informes visual, que permite a los usuarios generar visualmente informes, cartas y documentos, en este caso se ocupará para generar las facturas del establecimiento; de igual manera se ocuparán paqueterías las cuales permiten agregar imágenes, calendarios e iconos para dar un atractivo visual al programa.

Cabe mencionar que NetBeans utiliza el lenguaje de programación Java el cual se destaca por ser rápido, seguro, fiable y pasa por una JVM (Java Virtual Machine) que permite ser interpretado para posteriormente ser traducido al lenguaje de la «máquina». En el apartado de base de datos se ocupará el software XAMPP en su versión 8.0.3 que es una paquetería la cual contiene gestor de datos MySQL en su versión 10.4.14, servidor web e intérpretes para lenguaje script PHP y Perl; El cual nos permitirá a través del entorno phpMyAdmin en su versión 5.0.3, hacer uso de la interfaz y diseñar las tablas que complementarán las bases de datos que ocuparemos en nuestro programa las cuales almacenarán los datos de administradores, usuarios, clientes y facturas.

Modelo de base de datos

En la siguiente imagen se muestra lo que es el modelo de base de datos el cual contiene las tablas a ocupar en el programa, así como el tipo de dato de cada tabla. Cabe mencionar que dicha base de datos nos permitirá almacenar la información dada de alta en el programa para hacer las validaciones las cuales nos permitirán verificar el tipo de acceso que tendrá el usuario, así como el control de los datos almacenados de los clientes. De igual manera se contará con la tabla de platos y bebida las cuales permitirán dar de alta los alimentos para mostrarlos en el menú.

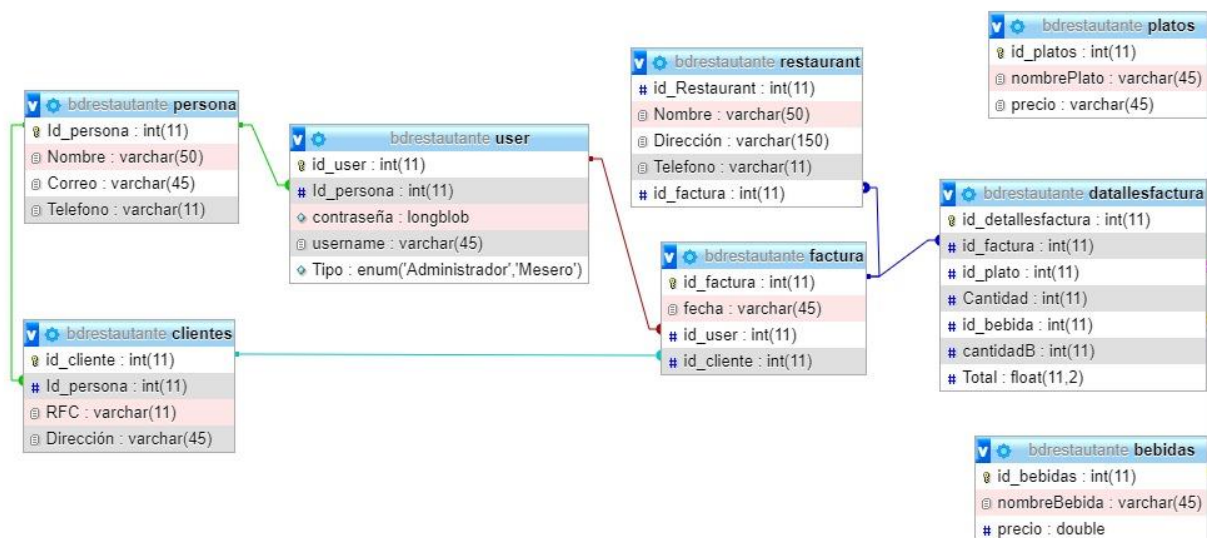


Imagen del modelo de base de datos

Diccionario de datos

BD_Restaurante							
CAMPO	TIPO DE DATO	TAMAÑO/VALORES	PK	FK	NN	AI	Descripcion
Persona							
idPersona	int	11	SI		SI	SI	Número unico identificador de Persona
nombre	varchar	50					Nombre unico de Persona
correo	varchar	45					Correo electronico de persona
telefono	varchar	11					Telefono de la persona
Clientes							
idCliente	int	11	SI		SI	SI	Numero unico identificador de Cliente
idPersona	int	11		SI			Número unico identificador de Persona
RFC	varchar	11					Registro Federal de Contribuyentes
telefono	varchar	11					Número telefonico del cliente
User							
idUser	int	11	SI		SI	SI	Numero identificador del User
idPersona	int	11		SI			Numero identificador de la Persona
contraseña	longblob						Contraseña del user
username	varchar	45					Usuario unico
tipo	enum	('Administrador','Mesero')					Tipo de usuario a seleccionar
Restaurant							
idRestaurant	int	11	SI		SI	SI	Número unico identificador de Restaurant
Nombre	varchar	50					Nombre unico del restaurant
Direccion	varchar	150					Dirección unica del restaurant
Telefono	varchar	11					Telefono unico del restaurant
idFactura	int	11		SI			Número unico identificador de la Factura
Factura							
idFactura	int	11	SI		SI	SI	Número unico identificador de la Factura
fecha	varchar	45					Fecha de cuando es generada la factura
idUser	int	11		SI			Número unico identificador del User
idCliente	int	11		SI			Número unico identificador de Cliente
Platos							
idPlatos	int	11	SI		SI	SI	Número unico identificador de Platos
nombrePlato	varchar	45					Nombre unico del platillo
precio	double	45					Costo del platillo

Bebidas							
idBebida	int	11	SI		SI	SI	Número unico identificador de Bebidas
nombreBebida	varchar	45					Nombre unico de la Bebida
precio	double						Costo de la bebida
DetallesFactura							
idDetallesFactura	int	11	SI		SI	SI	Número unico identificador de los Detalles de la Factura
idFactura	int	11		SI			Número unico identificador de la Factura
idPlato	int	11		SI			Número unico identificador de Platos
cantidad	int	11					Total de consumo de platos
idBebida	int	11		SI			Número unico identificador de Bebidas
cantidadB	int	11					Cantidades de bebidas consumidas
total	float	11,2					Total de consumo

Patrones de diseño

Composite

El patrón decorador permite al usuario agregar las funciones y responsabilidades del objeto de manera dinámica y transparente. Esto se hace a través de la relación con otras clases, extendiendo su funcionalidad al incorporar las de las clases asociadas, de esta forma el patrón no es dependiente de la Herencia, en este caso se utilizará para administración usuarios como los son administradores, meseros, cliente ya que estos suelen ocupar los mismos campos y a través de esta se pueden añadir funcionalidades ya que dichas clases implementará campos de la principal.

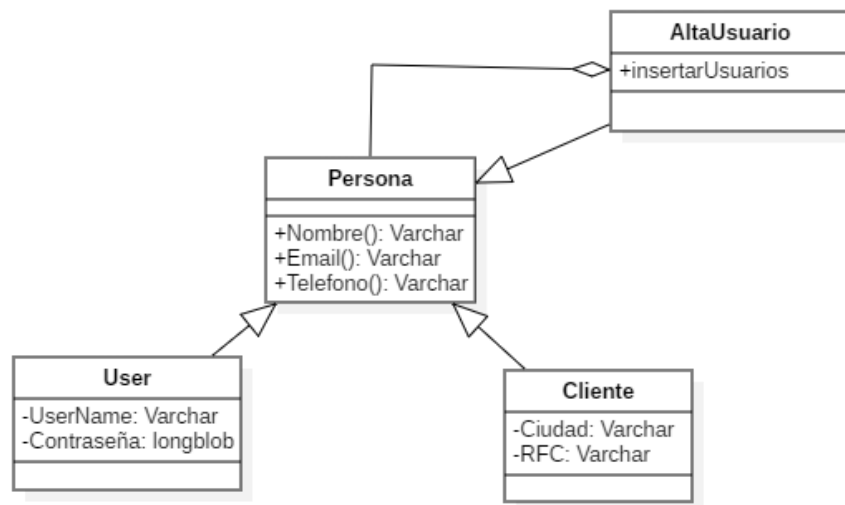


Diagrama composite

Primero se creará una clase persona que tiene los mismos atributos o campos que ocupan administradores, mesero, clientes dichos campos son (nombre, email, teléfono) una vez que se tenga estos elementos se procederán a crear las demás clases que heredaran los atributos de la clase persona y así puedan ser implementados

```
package Clases;

public class Persona {
    private String nombre;
    private String telefono;
    private String email;
    //constructores
    public Persona() {
    }
    public Persona(String nombre,String telefono) {
        this.nombre = nombre;
        this.telefono = telefono;
    }
    // get and set
    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public String getTelefono() {
        return telefono;
    }
}
```

Se proceden a crear las demás clases que será User y Cliente cabe mencionar que la clase User engloba a los usuarios administrador y cliente, Estas clases van a heredar los atributos de la clase principal como lo muestra el diagrama

```
package Clases;
```

```
public class User extends Persona{
    private String usernameUsuario;
    private String contrasenaUsuario;
    private String tipoNivelUsuario;

    public User () {

    }

    public String getUsernameUsuario() {
        return usernameUsuario;
    }

    public void setUsernameUsuario(String usernameUsuario) {
        this.usernameUsuario = usernameUsuario;
    }

    public String getContrasenaUsuario() {
        return contrasenaUsuario;
    }

    public void setContrasenaUsuario(String contrasenaUsuario) {
        this.contrasenaUsuario = contrasenaUsuario;
    }

    public String getTipoNivelUsuario() {
        return tipoNivelUsuario;
    }
}
```

```
public class Cliente extends Persona{
    String rfc;
    String ciudad;

    public String getRuc() {
        return rfc;
    }

    public void setRuc(String ruc) {
        this.rfc = rfc;
    }

    public String getCiudad() {
        return ciudad;
    }

    public void setCiudad(String ciudad) {
        this.ciudad = ciudad;
    }

    @Override
    public void setTelefono(String telefono) {
        super.setTelefono(telefono); //To change body of generated methods, choose Tools | Templates.
    }

    @Override
    public String getTelefono() {
        return super.getTelefono(); //To change body of generated methods, choose Tools | Templates.
    }
}
```

Una vez que se crearon dicha clase se procederá a hacer la función insertar cabe mencionar que este patrón nos va permitir modificar las clases externas ya que como tal nunca se afectan la función principal ya que dichas clases están heredando los atributos de la clase principal persona

```
public boolean insertarUsuarios(User usua) {
    cadena = "insert into usuario(userName,contraseña,tipoNivel,mail,telefono,nombre) "
        + "values(?,?,?,?,?,?)";
    try {
        PreparedStatement pst = co.prepareStatement(cadena);
        pst.setString(1, usua.getUsernameUsuario());
        pst.setString(2, usua.getContrasenaUsuario());
        pst.setString(3, usua.getTipoNivelUsuario());
        pst.setString(4, usua.getMailUsuario());
        pst.setString(5, usua.getTelefono());
        pst.setString(6, usua.getNombre());

        int n = pst.executeUpdate();//de esto depende si e completo completamente
        if (n != 0) {
            return true;
        } else {
            return false;
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "CONTACTE CON EL ADMINISTRADOR" + e);
        return false;
    }
}
```

```
public boolean insertarCliente(Cliente cli) {
    cadena = "insert into clientes(nombre,ruc,telefono,correo,ciudad) "
        + "values(?,?,?,?,?)";

    try {
        PreparedStatement pst = co.prepareStatement(cadena);
        pst.setString(1, cli.getNombre());
        pst.setString(2, cli.getRuc());///aquí va el ruc
        pst.setString(3, cli.getTelefono());
        pst.setString(4, cli.getCorreo());///aquí va el correo
        pst.setString(5, cli.getCiudad());///aquí va ciudad

        int n = pst.executeUpdate();//de esto depende si e completo completamente
        if (n != 0) {
            return true;
        } else {
            return false;
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "CONTACTE CON EL ADMINISTRADOR" + e);
        return false;
    }
}
```

Principio de diseño

En este caso ocuparemos el principio de sustitución de Liskov que menciona que si usamos una clase en alguna parte del código y ampliamos la clase, debemos poder usar cualquier subclase y el programa seguirá siendo válido. Esto nos obliga a asegurarnos de que el comportamiento del padre no cambie cuando ampliamos la clase. Este principio es contrario al pensamiento preconcebido de que una clase es una forma directa de modelar. en esta ocasión lo implementaremos en los usuarios ya que son las clases que implementaran los métodos sin alterar el comportamiento esperado en el programa. ahorrados tiempo al momento de programar

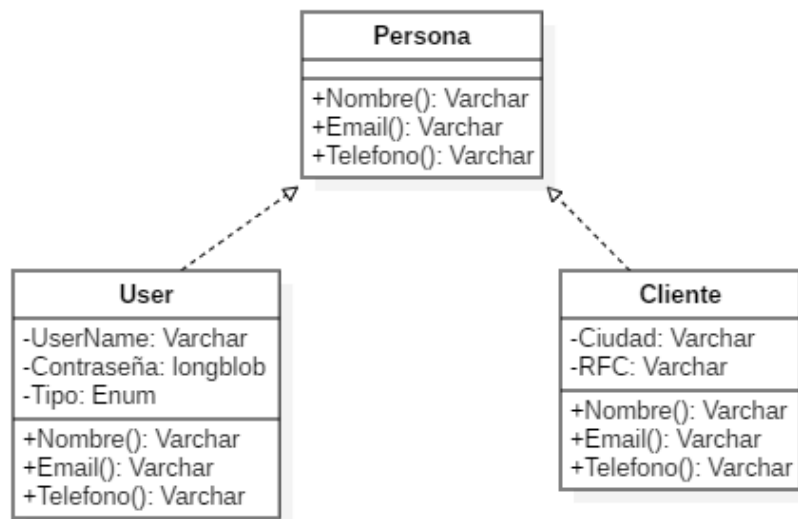


Diagrama de principio de sustitución

Arquitectura utilizada

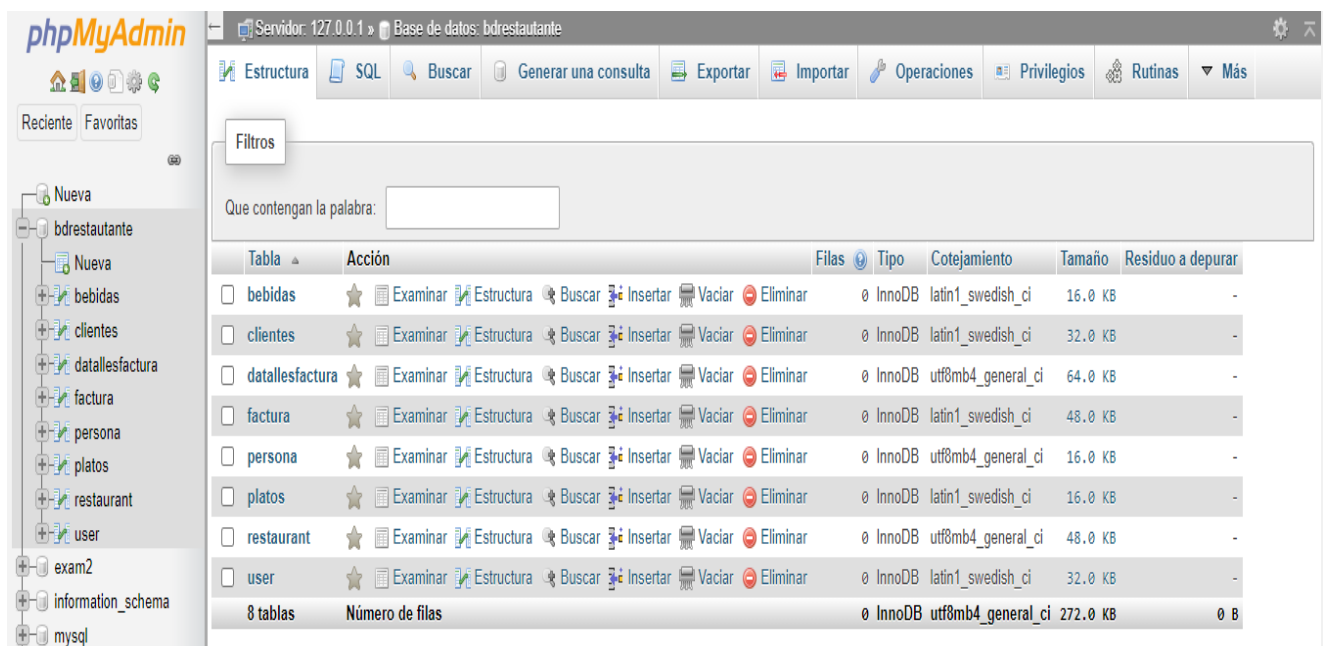
La arquitectura a implementar es la de Modelo de Vista Controlador mejor conocida como

El patrón de MVC: Es para el diseño de aplicaciones con sofisticadas interfaces es muy recomendable usar el patrón de diseño Modelo-Vista-Controlador. La lógica de una interfaz de usuario cambia con más frecuencia que los almacenes de datos y la lógica de negocio. Elementos del patrón, son:

- **Modelo:** Encapsula los datos y las funcionalidades. El modelo es independiente de cualquier representación de salida y/o comportamiento del controlador o la vista
- **Vista:** Muestra la información a través de una interfaz de usuario. Pueden existir múltiples vistas del modelo. Cada vista tiene asociado un componente controlador con el que interactúa.
- **Controlador:** Reciben las entradas de las vistas, usualmente como eventos que codifican los movimientos o pulsación de botones del ratón, pulsaciones de teclas, etc. Los eventos son traducidos a solicitudes de servicio ("service requests ") para el modelo o la vista

Modelo ejemplo:

El modelo implementado será MYSQL en PHPMYADMIN ya que a través de las tablas hechas en la base de datos se almacenará la información introducida por los 2 tipos de usuarios, en la siguiente imagen se muestra la base de datos desarrollada en el entorno mencionado



The screenshot shows the phpMyAdmin interface for a database named 'bdrestaurante'. The left sidebar lists the database and its tables: bebidas, clientes, datallesfactura, factura, persona, platos, restaurant, user, exam2, information_schema, and mysql. The main panel displays the 'Estructura' (Structure) tab for the 'bdrestaurante' database. It shows a list of 8 tables with their respective actions (Examinar, Estructura, Buscar, Insertar, Vaciar, Eliminar) and details like engine, collation, and size.

Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar
bebidas	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	16.0 KB	-
clientes	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	32.0 KB	-
datallesfactura	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8mb4_general_ci	64.0 KB	-
factura	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	48.0 KB	-
persona	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8mb4_general_ci	16.0 KB	-
platos	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	16.0 KB	-
restaurant	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8mb4_general_ci	48.0 KB	-
user	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	32.0 KB	-
8 tablas	Número de filas	0	InnoDB	utf8mb4_general_ci	272.0 KB	0 B

Imagen del entorno BD

Vista ejemplo:

En este caso la vista seria la interfaz gráfica diseñado por nosotros ya que a través de ella se realizarán los procedimientos requeridos como el CRUD de alimentos y usuarios en donde a través de los formularios se modificará la base de datos



Ejemplo de uno de los formularios que sirve como vista

Ejemplo controlador:

serán los eventos programados que se realizarán al presionar los botones de los formularios ya que a través de ellos se generar los procesos como la alta edición de usuarios y alimentos, en las siguientes imágenes se mostrarán como a través del código se almacena la información

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    //variables de recuperacion de datos  
    int validacion = 0; //variables  
    String nombre = "", mail = "", telefono = "", username = "", pass = "", tipoNivel = "Administrador";  
  
    //recuperar valores dentro de los texfile  
    mail = txt_email.getText().trim(); //metodo trim eliminamos los espacios que estan al inicio y al final de la ca  
    username = txt_username.getText().trim();  
    pass = txt_password.getText().trim();  
    nombre = txt_nombre.getText().trim();  
    telefono = txt_telefono.getText().trim();  
  
    //se cambian de color a rojo  
    //validamos los campos antes de enviar los datos  
    if (mail.equals("")) {  
        txt_email.setBackground(Color.red);  
        validacion++;  
    } else {  
        txt_email.setBackground(Color.white);  
    }  
    if (nombre.equals("")) {  
        txt_nombre.setBackground(Color.red);  
        validacion++;  
    } else {  
        txt_nombre.setBackground(Color.white);  
    }  
    if (telefono.equals("")) {  
        txt_telefono.setBackground(Color.red);  
        validacion++;  
    } else {  
        validacion++;  
    }  
}
```

Código del botón guardar de Administrador

```

//Enviamos los datos obtenidos a las clases que hemos creado
Usuario usuario = new Usuario();
Administrador administrador = new Administrador();
bd_Conexion con = new bd_Conexion();
administrador.setNombre(nombre);
administrador.setMailUsuario(mail);
administrador.setTelefono(telefono);
administrador.setUsernameUsuario(username);
administrador.setContraseñaUsuario(pass);
//administrador.setTipoNivelUsuario(tipoNivel);
//Usuario
usuario.setNombre(nombre);
usuario.setMailUsuario(mail);
usuario.setTelefono(telefono);
usuario.setUsernameUsuario(username);
usuario.setContraseñaUsuario(pass);
usuario.setTipoNivelUsuario(tipoNivel);

if (con.insertarAdministrador(administrador) && con.insertarUsuarios(usuario)) {
    JOptionPane.showMessageDialog(null, "Administrador Registrado corecctamente");
    Limpiar();
} else {
    JOptionPane.showMessageDialog(null, "Error Contacte con el Administrador");
}

```

Código del botón guardar de Administrador

De igual manera se usó el patrón Cliente-Servidor: Es una arquitectura de suma importancia en la que se consigue un procesamiento cooperativo de la información por medio de un conjunto de procesadores, de tal forma que uno o varios clientes, distribuidos geográficamente o no, solicitan servicios de computación a uno o más servidores.

```

public class conexion {

    Connection co;

    public static Connection Conectar(){
        try{
            Class.forName("com.mysql.jdbc.Driver");
            Connection co=DriverManager.getConnection ("jdbc:mysql://127.0.0.1/bdrestaurante","root","");
            System.out.println("CONEXION EXITOSA");
            return co;
        } catch(SQLException ex){
            System.out.println("ERROR CON MYSQL" + ex);
        } catch(ClassNotFoundException exc){
            exc.printStackTrace();
        }
        return null;
    }

    public static ResultSet getTabla(String Consulta){
        Connection cn=Conectar();
        Statement st;
        ResultSet datos = null;
        try{
            st= cn.createStatement();
            datos = st.executeQuery(Consulta);
        }catch(Exception e){
            System.out.print(e.toString());
        }return datos;
    }
}

```

Código de conexión con el servidor

De esta manera, y gracias a esta arquitectura, la totalidad de los procesadores, clientes y servidores, trabajaran de forma cooperativa para realizar el determinado tratamiento de la información ya que el servidor será la base de datos del programa y el cliente los usuarios los cuales realizar los procedimientos.

Construcción

En esta tapa se anexarán las capturas en donde se presentarán gran parte del código y el correcto funcionamiento del programa así mismo se anexará un video en donde se mostrará a detalle los formularios y las clases que se ocuparon para el diseño del programa

Código

esta captura muestra el código de la calculadora que se ocupara en programa cabe mencionar que estará las operaciones básicas y se utilizara GUI que es aquello que nos permitirá la interacción entre el usuario y la aplicación

```
package calculadora;

import java.text.DecimalFormat;

public class InterpretadorExpresionesMatemáticas_Básicas {

    public static void main(String[] args) {
        String expresión = "-3+20/2+1";
        String Resultado = Interpretar(expresión);
        System.out.println("Interpretación: " + Resultado + " de: " + expresión);
        double d = Double.parseDouble(Resultado);
        System.out.println("Número: " + d);
    }

    public static String Interpretar(String cadena) {
        if (cadena == null || cadena.isEmpty()) {
            return "0";
        }
        String[] agrupadores = {
            "(", ")", "[", "]", "{", "}"
        };
        for (String agrupador : agrupadores) {
            if (cadena.contains(agrupador)) {
                throw new RuntimeException(
                    "El interpretador no reconoce expresiones con agrupaciones, es muy básico"
                );
            }
        }
        cadena = cadena
            .replace("+-", "-")
            .replace("-+", "-")
            .replace("...", "...");

        protected static double ObterRestas(String cadena) {
            boolean restar = cadena.startsWith("-");
            if (restar) {
                cadena = cadena.substring(1);
            }
            String[] términos = cadena.split("-");
            double[] op = new double[términos.length];
            for (int i = 0; i < op.length; i++) {
                try {
                    double d = Double.parseDouble(términos[i]);
                    op[i] = d;
                } catch (Exception e) {
                    op[i] = ObterMultiplicaciones(términos[i]);
                }
            }
            double resultado = 0;
            for (double d : op) {
                if (restar) {
                    resultado -= d;
                } else {
                    resultado += d;
                    restar = true;
                }
            }
            return resultado;
        }

        protected static double ObterMultiplicaciones(String cadena) {
            String[] términos = cadena.split("x");
            double[] factores = new double[términos.length];
            for (int i = 0; i < factores.length; i++) {
```

código general del formulario principal en donde su función es enlazar a los demás formularios, funciona como un menú de opciones en donde el usuario escoge la operación a realizar

```
package ventanas;
import calculadora.CalculadoraGUI;
import java.awt.Color;
import java.awt.Image;
import java.awt.Toolkit;
import javax.swing.WindowConstants;
/**
 *
 * @author Frank
 */
public class frm_Admin extends javax.swing.JFrame {
    String user, nombre_usuario;
    public static int session_usuario;//enviar datos entre interfaces
    public frm_Admin() {
        initComponents();
        user = frm_Login.user;
        session_usuario = 1;//bandera
        setSize(650, 430);
        setResizable(false);
        setTitle("Administrador - Sesión " + user);
        setLocationRelativeTo(null);
        setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);//al ser cerrada el programa dejara de ejecutarla(ev.
    }
    //cambiar la imagen del logo de la taza
    //Inicio
    @Override
    public Image getIconImage() {
        Image retValue = Toolkit.getDefaultToolkit().getImage(ClassLoader.getResource("imagenes/icono.png"));
        return retValue;
    }
}
```

```
private void jButton_RegistrarUsuarioActionPerformed(java.awt.event.ActionEvent evt) {
    //Cuando el usuario da click en el boton abra esta ventana
    frm_RegistrationOption ro=new frm_RegistrationOption();
    ro.setVisible(true);
}

private void jButton_MenuActionPerformed(java.awt.event.ActionEvent evt) {
    //Cuando el usuario da click en el boton abra esta ventana
    frm_Menu menu = new frm_Menu();
    menu.setVisible(true);
}

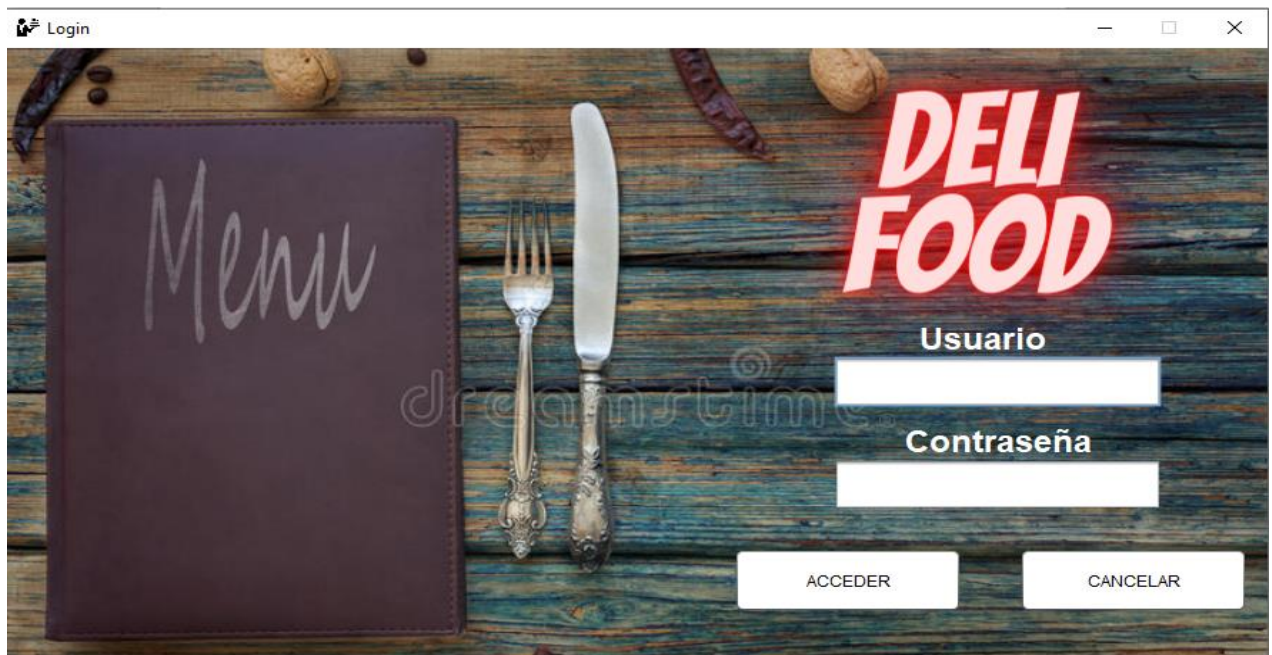
private void jButton_FacturaActionPerformed(java.awt.event.ActionEvent evt) {
    //Cuando el usuario da click en el boton abra esta ventana
    frm_ChooseVoucher CV = new frm_ChooseVoucher();
    CV.setVisible(true);
}

private void jButton_CalculadoraActionPerformed(java.awt.event.ActionEvent evt) {
    //Cuando el usuario da click en el boton abra esta ventana
    CalculadoraGUI calc = new CalculadoraGUI();
    calc.setVisible(true);
}

private void jButton_ReportesActionPerformed(java.awt.event.ActionEvent evt) {
    //Cuando el usuario da click en el boton abra esta ventana
    frm_Reports rp = new frm_Reports();
    rp.setVisible(true);
}
```

Login de usuario

Este login permite el ingreso a las cuentas de usuario mostrando 2 diferentes tipos de menú uno para administradores y otro para meseros así mismo permite la autenticación al acceder, delimitando las opciones según el tipo de



usuario que utilice el programa

Menú de opciones de administradores

Es un formulario en donde se observarán las opciones las cuales permitirán al usuario realizar una determinada tarea, cabe destacar que dependiendo el usuario estas opciones se limitaran o cambiaran.



Menú de opciones de meseros



Menú de Alimentos

Formulario en donde se mostrarán los alimentos y bebidas con sus respectivos precios así mismo se seleccionará de acuerdo al consumo de los clientes



Factura:

Formulario de la factura en donde se ingresarán los datos del usuario y se indicarán los alimentos consumidos, así como la cantidad y el precio por cada alimento de igual manera se indicará el total a pagar por el consumo de los alimentos

Cantidad	Descripcion	Precio Unitario	Monto
----------	-------------	-----------------	-------

TOTAL : 0.0

Generar Factura

Factura PDF

ejemplo del pdf el cual se genera automáticamente y muestra los datos introducidos en la factura así mismo se observa que en el encabezado proporciona la información del restaurante

FACTURA
Deli Food
TELEFONO:7791002648
Av. Juárez Nte. s/n, Cuxtitla, 43800 Tizayuca, Hgo.

Nombre: Fernando Hernandez
RFC: FERHDS7041256
Telefono: 7791003678

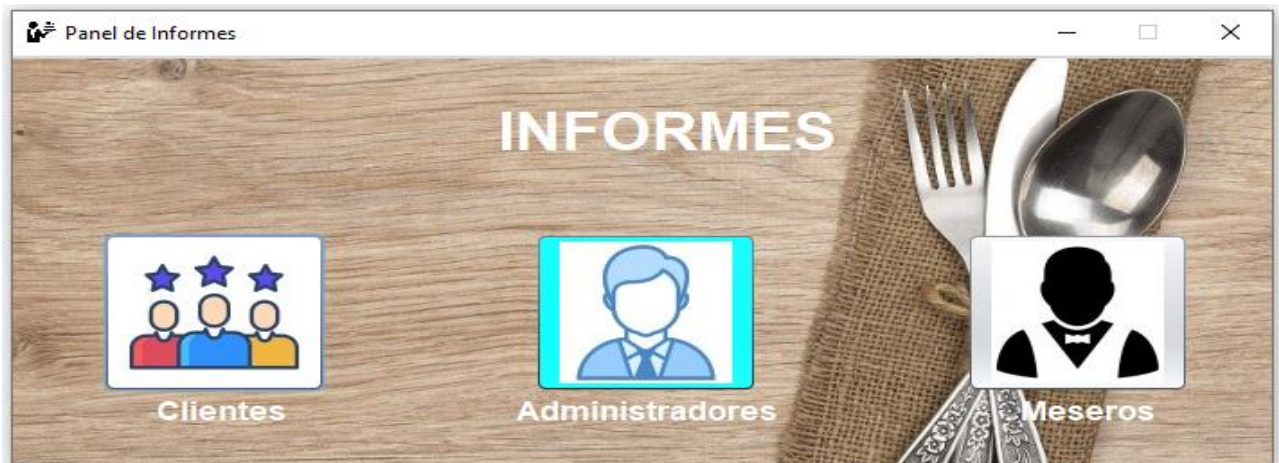
Direccion: av 16 de Enero Tizayuca Hgo
Correo: fernando55f@gmail.com
Fecha: 12-04-2021

Descripcion	Cantidad	Precio Unitario	Total
ARROZ CON POLLO	1	35	35
BISTEC	1	40	40
INDIO 355ml	1	30	30

Total: 105.0

panel de Informes

Formulario en donde se mostrarán las tablas las cuales contienen los datos de los clientes y usuarios que se dieron de alta en el programa así mismo estos se podrán editar o actualizar si el administrador lo desea



Panel de datos de administradores

Buscar Administradores

Ingrese nombre para buscar o eliminar

Nombre	Telefono	Mail	Contraseña	UserName
Jezer Pérez	976747953	jezer25@gmail.com	admin	jezeradmi
Vicente Ledezma	98562359	viceledk@gmail.com	admin	vicadmi
Jesús Hernández	982365895	jesu58@gmail.com	admin	jesusadmi
Alan Gonzalez	5583497291	alan530@gmail.com	admin	alanadmi

Panel de datos de clientes

Buscar Clientes

Ingrese nombre para buscar o eliminar

Nombre	Ruc	Telefono	Correo	Ciudad
Maria Benitez	10754578975	976749582	Maria@gmail.com	Sullana
Jenifer Carrasco	00000000000	978562356	jenifer126@gmail.com	Mexico
Andre Risco	00000000011	975862358	andreris@gmail.com	Monterrey
Gustavo Canova	00000000000	989856685	Sin especificar	Mexico
Sebastian Alvarado	00000000000	978523654	sebas123@gmail.com	Monterrey

Panel de datos de mesero

Mesero

Buscar Mesero

Ingrese nombre para buscar o eliminar

Nombre	Telefono	Mail	Contraseña	UserName	N°
Jezzer	985623658	jezzer@gmail.com	mesero	jezermr	1
Oscar	978568953	oscar58@gmail.com	mesero	oscarmsr	2
Marco	975856325	marko@gmail.com	nesero	marcomsr	3

Panel de datos de factura

Factura

Buscar Factura

Ingrese nombre para buscar o eliminar




N°	Fecha	Nombre	Ciudad	Telefono	Total
1	17-07-2019	Juana Avila	Sullana	975865235	102
2	17-07-2019	Pepe Ojeda	Piura	976568652	83
3	17-07-2019	Rocio Quedena	Piura	974845645	26
4	17-07-2019	Teresa Ortiz	Chidlayo	978653265	74
5	17-07-2019	Anthony Reyes	catacaos	978562365	41
6	14-07-2019	Ismael Lopez	Sechura	975864582	37
7	17-07-2019	Hector Yesang	Piura	976856235	77

Calculadora

Formulario de calculadora para realizar operaciones básicas en caso de que se ofrezca algún tipo de descuento se muestra una interfaz gráfica sencilla con botones, números y símbolos de las operaciones para que se pueda manipular de una manera rápida

A screenshot of the Windows 10 calculator application. The title bar at the top shows the application icon, the text "Calculadora", and standard window controls (minimize, maximize, close). Below the title bar is a toggle switch labeled "Siempre visible" (Always visible), which is currently turned off. The main display area shows a large "0". Below the display is a grid of buttons. The first row contains buttons for square root (√), square (x²), cube (x³), and percentage (%). The second row contains buttons for "CE", "C", a left arrow (←), and division (÷). The third row contains buttons for "7", "8", "9", and multiplication (×). The fourth row contains buttons for "4", "5", "6", and addition (+). The fifth row contains buttons for "1", "2", "3", and subtraction (-). The sixth row contains buttons for positive/negative (±), "0", a decimal point (.), and equals (=).

Panel de alta Administradores

Los formularios de alta permiten guarda la información de los usuarios y alimentos en las tablas de base de datos para que posteriormente se haga uso de ella ya que a través de esta información se valida los usuarios y se muestran los alimentos en el menú



Registrar nuevo Administrador

Registro de Administrador

Nombre:

Email:

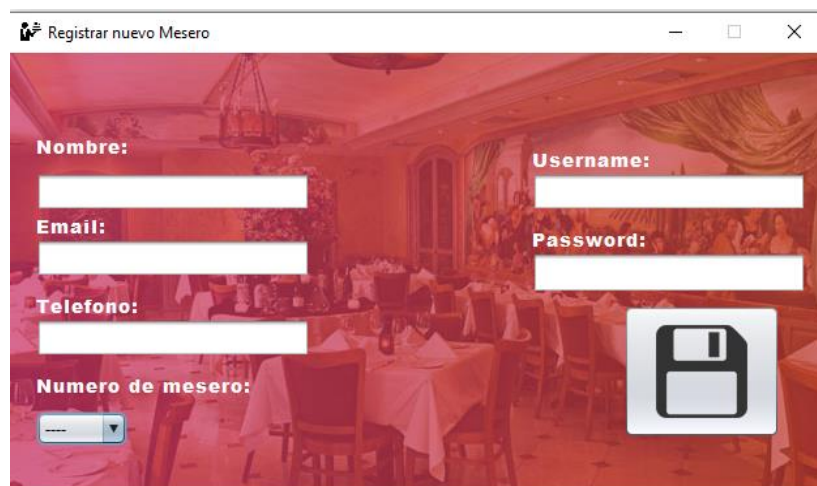
Telefono:

Username:

Password:



Panel de alta Meseros



Registrar nuevo Mesero

Nombre:


Email:

Telefono:


Numero de mesero:

Username:

Password:



Panel de alta Alimentos

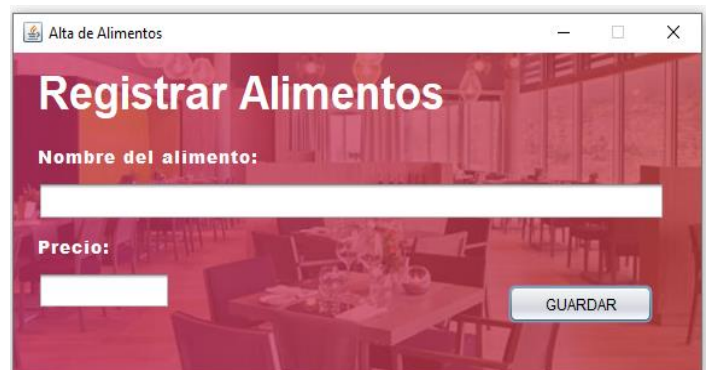


Alta de Bebidas

Registrar Bebida

Nombre de la bebida:

Precio:



Alta de Alimentos

Registrar Alimentos

Nombre del alimento:

Precio:

Programación de Cliente Servidor

En este caso la arquitectura a ocupar es la siguiente Cliente Pasivo, Servidor Activo ya que el servidor que en este caso es MySQL_server se encargará de realizar toda la actividad de procesamiento como lo es los procedimientos de búsqueda los cuales permitirá encontrar un dato específico dentro de la información de las tablas de igual manera este será el encargado de gestionar la información, mientras que el cliente es pasivo ya que simplemente presenta los datos que serán introducidos a través del programa

Programación del servidor

En este caso la programación del servidor corresponde a la base de datos elaborada en MYSQL ya que es aquella que administrará y almacenará las transacciones de dato que se ejecutaran en el software, así mismo este tendrá una interacción y comunicación con las diferentes solicitudes del usuario que en este caso serán los administradores y meseros ya que ellos aplicaran las herramientas para ejecutar los comandos de intercambio de datos haciendo que la base obtenga una actualización de la información de los registros y tablas al instante, Cabe señalar que al registrar usuarios en la base de datos este podrá acceder a la información almacenada que le permitirá realizar diferentes tareas de acuerdo a sus niveles de privilegio.

La función principal de este servidor es la gestión de registros del personal y un mejor manejo de los datos almacenados

Programación del cliente

Como se mencionó anteriormente la programación de cliente estarán principalmente basada en las peticiones de los administradores y meseros donde a través de un usuario, contraseña y nombre de host, se le permiten obtener una serie de beneficios los cuales darán control a la información guardada y solicitada en el sistema, En este caso la principal función de la programación del cliente es mantener comunicación con los procesos para establecer una conexión con el la base de datos para presentar la solicitudes de y al mismo tiempo gestionar las fallas que puedan ocurrir en el proceso

Prueba

En el siguiente link se presenta un video en el cual se muestra el funcionamiento del programa elaborado y se explica a detalle la arquitectura ocupada, así como los patrones de diseño

<https://drive.google.com/file/d/1FMvstnDFfhJabVdTPzPQIJkUOCz0zeN6/view?usp=sharing>

Despliegue

Entrega

Al tener acabado el producto con sus respectivas pruebas además de la validación del cliente sobre este, a continuación se procede a comunicarse con el cliente para solicitarle la indicación sobre en qué computadora o computadoras se realizara la instalación del programa adema de brindarle al cliente una guía de un tamaño un poco pequeño para que los usuarios que vayan a ocupar el software se acostumbren de una manera más fácil y sencilla al manejo del mismo.

Asistencia

En caso de que haya algun detalle o problema con respecto al desarrollo del programa, nuestra obligacion es asistir y apoyar, resolviendo todas las dudas y/o problemas que se presenten

Evaluación

El equipo determina y se evalúa en tanto a la eficiencia con la que trabajó el equipo y si se logra los propósitos esperados del proyecto.

Pérez Cruz Jezer Alfredo **10**

González Escalona Ángel Alan **10**

Ledezma Rojo Vicente **9.5**

Damián Hernández José De Jesús **10**

Citlaly Alejandra Hernández Cuevas **9**

Conclusión

Luego de haber puesto en práctica y presentado el proyecto podemos concluir los siguientes puntos:

- Con base a los pasos de la metodología de cascada esta se reduce de manera significativa, todo el esfuerzo, tiempo y recursos invertidos en el diseño e implementación del software, optimizando además tiempo de programación.
- Es muy importante el definir claramente los requerimientos del software al principio para evitar cambios al final del mismo (evitar el uso de Riesgos).
- Se debe tener en cuenta el diseño del software, ubicar nombres a los objetos con la función que desempeñan.

- El sistema permite manejar la información de distintas sucursales (que sean de la misma empresa). Es así que, al terminar y entregar el proyecto, aparte de estar implantado para la fonda, puede usarse para otros lugares si el cliente lo desea.

Después de todo lo mencionado, podemos decir que nuestro proyecto tendrá grandes oportunidades en la empresa de ser aceptado y usado dentro de la fonda, lo cual se refleja un buen resultado hacia la empresa, la reducción de tiempo en tomar una orden o cancelar, eliminar el uso de hojas para escribir cada vez que se da de alta un nuevo platillo.

Al ser este un proyecto pequeño o de clase, no es un borrador, este podría tener grandes expectativas (como se menciona anteriormente) hacia alguna fonda chica que le interesa actualizarse en el mundo tecnológico y poder ofrecer más ideas a este proyecto y poder hacer que el proyecto crezca.

Lecciones aprendidas

Calendarización

En esta lección aprendida tomamos en cuenta las habilidades de tu equipo para poder tener un proyecto bueno, y poder adaptar las actividades a cada integrante de equipo para que esto se haga en tiempo y forma.

Trabajo en equipo

Trabajar en equipo es fundamental para poder tener éxito en el proyecto, ya que, al dividir las actividades y el poder ayudarnos el desarrollo es más eficiente y rápido.

Implementación de una Metodología

La metodología que aprendimos a usar fue la de Cascada basada en pressman, esta nos ayudó a comenzar el software con bastante rapidez.

Implementación del Plugin iReport

Con esta implementación de este plugin nos facilitó el diseño de las plantillas de jasperReport.

Implementación de la Arquitectura Vista controlador MVC

Esta implementación de la Arquitectura Vista Controlador MVC nos ayudó a separar los datos de la aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos.