

**Facultad de Ingeniería**

Carrera Profesional de Ingeniería de Sistemas e Informática

**Curso**

1° Integrador - Software

**Título del proyecto**

Sistema de reserva de mesas y cálculo de ingresos de un restaurante

**Alumno**

Garces Alva Renzo

Ponce Fernández Melissa

Paredes Rivas Yonatan

**Lima – Perú**

**ÍNDICE**

**CAPÍTULO 1 - ASPECTOS GENERALES**.....................................................

1.1 Descripción del Problema................................................................

1.2 Definición de Objetivos....................................................................

1.2.1 Objetivo General..........................................................................

1.2.2 Objetivos Específicos...................................................................

1.3 Alcances y Limitaciones................................................................

1.3.1 Alcances......................................................................................

1.3.2 Limitaciones.................................................................................

1.4 Justificación.................................................................................

1.5 Estado del Arte............................................................................

**CAPÍTULO 2 - MARCO TEÓRICO**................................................................

2.1 Marco teórico de programación orientada a objetos……………………

2.1.1 Java Netbeans…………………………………………………………..

2.2.2 Breve Reseña Histórica de la Empresa…………………………….

2.2.3 Definición del Sistema………………………………………………

2.2.4 Descripción de la secuencia de pasos del proceso actual del sistema…..

2.2.5 Gráfico de la secuencia de pasos del sistema manual………………..

**CAPÍTULO 3 - DESARROLLO DE LA SOLUCIÓN**..........

3.1 Diagrama de clases.........................................................................

3.2 Modelamiento de base de datos ......................................................

3.3 Entorno visual de los formularios......................................................

3.4 Código Fuente................................................................................

**CAPÍTULO 4 - RESULTADOS**........................

4.1 Resultados de la encuesta................. **¡Error! Marcador no definido.**

4.2 Presupuesto..................................... **¡Error! Marcador no definido.**

**CONCLUSIONES**

**BIBLIOGRAFÍAS**

**ANEXOS**

· Modelo CANVAS

· Modelo de encuesta

CAPÍTULO 1

ASPECTOS GENERALES

1. **Aspectos Generales**
   1. **Descripción del Problema**

En la actualidad en los restaurantes suele suceder que para hacer alguna reservación es todo un trámite, ya que deben apersonarse y realizar algunas cosas para la confirmación. Ante ello, se va crear un sistema en el cual podremos reservar y tener registrado tan solo llamando y dar los datos necesarios que serán guardados en una base de datos para una mayor organización y tener todo detallado. Además, este sistema calcula los ingresos diarios, genera comprobantes de pago del restaurante mediante reportes de los pedidos, cálculo y visualización del consumo total.

* 1. **Definición de Objetivos** 
     1. **Objetivo General**
* Desarrollar un sistema para el registro y reserva de mesas.
  + 1. **Objetivos Específicos**
* Calcular ingresos diarios
* Crear una base de datos
* Generar un comprobante de pago por consumo
  1. **Alcances y Limitaciones**
     1. **Alcances**
* Diseñada para el entorno escritorio
* Se podrá descargar el sistema desde la web.
* Servirá para tener el control y organización de las reservaciones
* Se utilizará una base de datos
  + 1. **Limitaciones**
* No está disponible para plataformas Android.
  1. **Justificación**

Debido a que en los restaurantes para reservar es toda una pérdida de tiempo, este sistema nos permitirá tener de forma más ordenada los datos, lo cual facilita el servicio brindado al usuario reduciendo los tiempos de reserva de mesa y atención considerablemente, para que así tenga una buena experiencia al visitar dicho restaurante.

* 1. **Estado del Arte**

**Allset**

Allset busca optimizar la experiencia completa, desde la reserva hasta elegir tu plato e incluso pagar. Tras descargarla e instalarla, puedes hacer tu reserva en el restaurante que elijas. Allset fue creada por Stas Matviyenko y Anna Polishchuk en 2015 y tiene su sede en Los Ángeles, California.

Imagen 1

**TPV 123**.

Nuestro principal objetivo es ayudar a nuestros clientes a ahorrar tiempo y dinero en las gestiones diarias de su negocio, proporcionando sencillez y fiabilidad en sus gestiones, permitiéndoles concentrarse en la lógica de su negocio. Esta app registra los datos de los clientes y también permite la recepción de pedidos mediante llamadas. Fue creado por TPV 123 en España en el 2015.

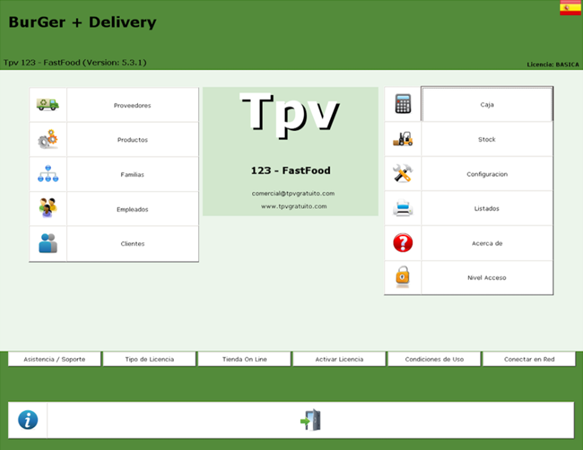


Imagen 2

CAPÍTULO 2

MARCO TEÓRICO

2.1 Marco teórico de programación orientada a objetos

2.1.1 Java Netbeans

**Clases en java:**

Las clases son la base fundamental de java, ya que, en base a estas se construye todo el lenguaje ya que nos permiten definir cuál es la naturaleza del objeto, siendo también el pilar de la programación orientada a objetos.

**Pilares de la programación orientada a objetos**

**Abstracción**: La abstracción te permite identificar una característica y comportamiento necesarias o importantes de un objeto, en otras palabras, no ayuda a identificar la información relevante para el contexto del problema

**Encapsulamiento:** Esta propiedad no permite ocultar la estructura del código que se encuentra en la parte privada de la clase, a la cual no se puede acceder por otro código.

**Herencia**: La herencia te permite la reutilización del código ya que está, como dice su nombre, te permite heredar los atributos de una clase superior o también llamada clase padre a sus clases hijas

**Polimorfismo:** El polimorfismo es la que nos permite insertar métodos en un objeto con el mismo nombre, logrando modificar los parámetros de los métodos, básicamente el polimorfismo nos permite definir clases diferentes con métodos con el mismo nombre pero que tienen funciones distintas y estas se distinguen por los atributos.

2.2.2 Breve reseña histórica de la empresa

Somos un Restaurante dedicado a la elaboración de platos de comida china a base de las distintas carnes, verduras y otros añadidos propios de esta especialidad de comida, en el cual usted podrá pasar de un momento placentero entre familia y/o amigos degustando una amplia variedad de platos atractivos al paladar, en un ambiente bastante agradable con un buen servicio.

2.2.3 Definición del Sistema

* Sistema de control de Ventas

un sistema de [control de ventas](https://www.zendesk.com.mx/blog/estrategia-de-ventas-coronavirus/) controla y también gestiona todos los datos de lo que se vende en una empresa, desde para quien se vendió, cuánto, cómo y quién lo vendió.(Douglas da Silva, 2020)

Ese control de compras y ventas existe para que el proceso se haga de forma transparente y los ejecutivos puedan tener números exactos de cómo el negocio se va desarrollando cada mes.(Douglas da Silva, 2020)

Este sistema ayuda en la disposición de que todos los precios de los productos o servicios que la empresa ofrece estén registrados sin ningún tipo de error. El resultado es que se tenga un control de las ventas transparente, con los valores correctos, tanto para la empresa como para el [cliente](https://www.zendesk.com.mx/blog/gartner-5-key-emerging-technologies-customer-experience/).(Douglas da Silva, 2020)

2.2.4 Descripción de la secuencia de pasos del proceso actual del sistema

* Sistema de control de Ventas

Paso 1: Llamada del cliente para reserva (5 minuto)

Paso 2: Solicita recepción (3 minutos)

Paso 3: Disponibilidad de mesa (2 minutos)

Paso 4: Realizar recepción (20 segundos)

Paso 5: Ingreso del cliente al restaurante (20 segundos)

Paso 6: Solicita el menú (20 segundos)

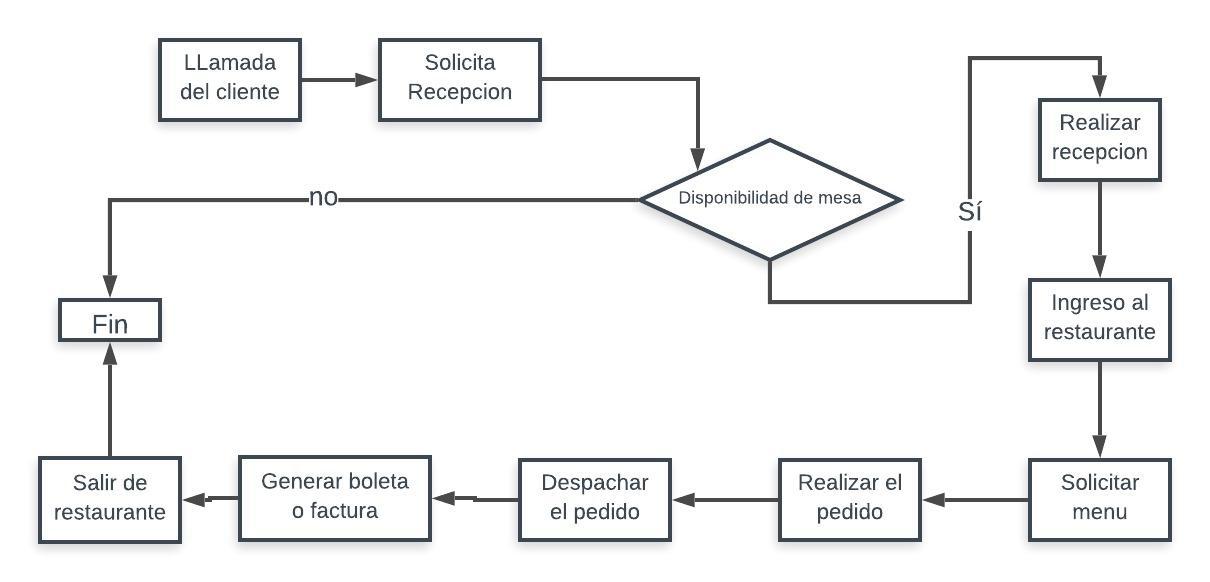
Paso 7: Realiza el pedido (20 segundos)

Paso 8: Despachar el pedido (8 minutos)

Paso 9: Generar boleta o factura (10 segundos)

Paso 10: Salir del restaurante

2.2.5

Imagen 3

CAPÍTULO 3

DESARROLLO DE LA SOLUCIÓN

3.1 Diagrama de clases

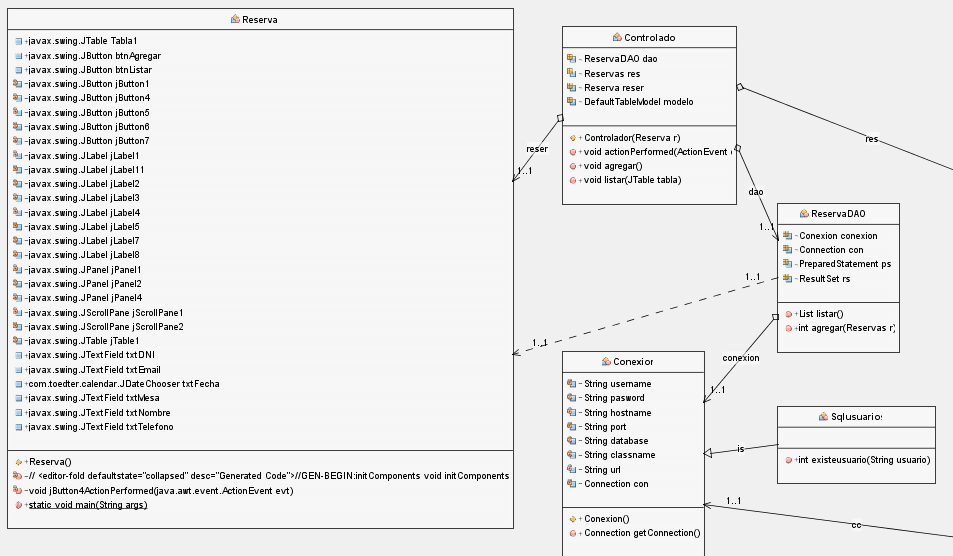


Imagen 4

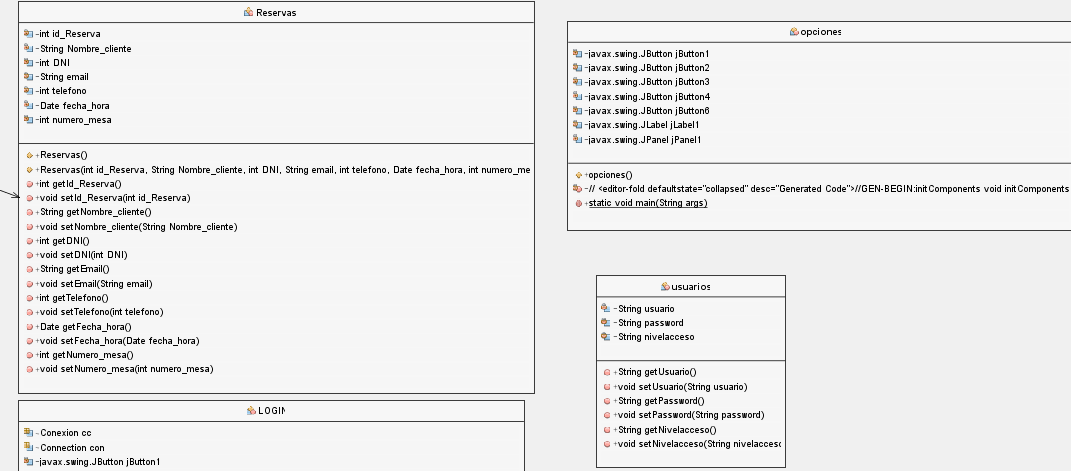


Imagen 5

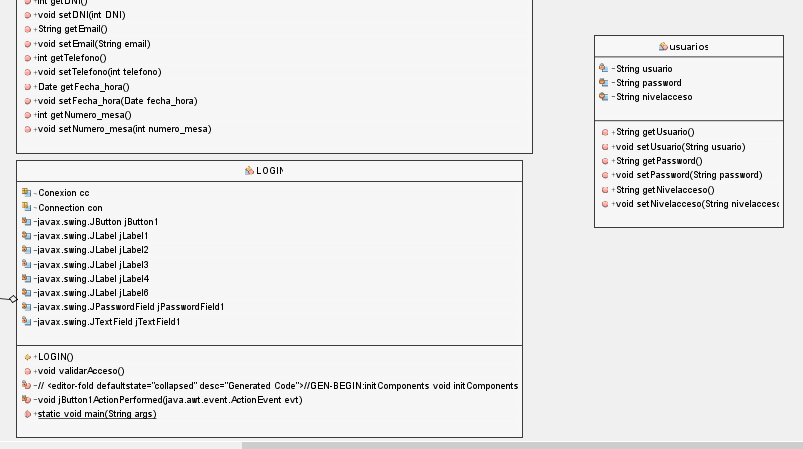
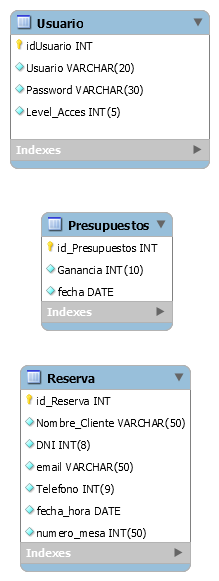


Imagen 6

3.2 Modelamiento de base de datos



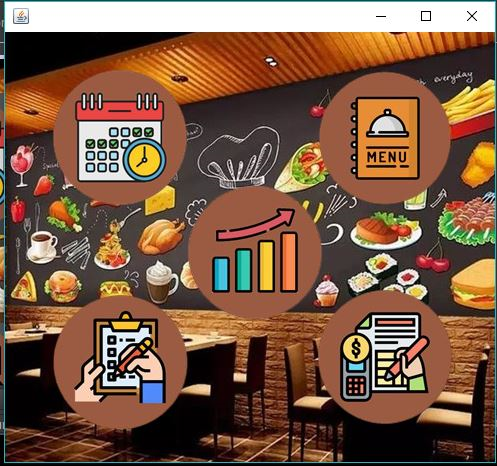


Imagen 7



Imagen 8



Imagen 9

3.4 Código fuente

LOGIN

public class LOGIN extends javax.swing.JFrame {

Conexion cc=new Conexion();

Connection con=cc.getConnection();

public LOGIN() {

initComponents();

}

public void validarAcceso(){

int resultado=0;

try {

String usuario=jTextField1.getText();

String password=String.valueOf(jPasswordField1.getPassword());

String sql="Select \* from usuarios where Usuario='" +usuario+ "' and Password='"+password+"' ";

Statement st=con.createStatement();

ResultSet rs=st.executeQuery(sql);

if(rs.next()){

resultado=1;

if (resultado==1){

Sistema\_Restaurante form=new Sistema\_Restaurante();

form.setVisible(true);

this.dispose();

}else{

JOptionPane.showMessageDialog(null, "Datos Incorrectos, Vuelve a Intentarlo");

}

}

}catch (Exception e){

JOptionPane.showMessageDialog(null, "Error en el Acceso, Vuelve a Intentarlo" + e.getMessage());

}

}

USUARIOS

public class usuarios {

private String usuario;

private String password;

private String nivelacceso;

public String getUsuario() {

return usuario;

}

public void setUsuario(String usuario) {

this.usuario = usuario;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

public String getNivelacceso() {

return nivelacceso;

}

public void setNivelacceso(String nivelacceso) {

this.nivelacceso = nivelacceso;

}

}

Sqlusuarios

public class Sqlusuarios extends Conexion {

public int existeusuario (String usuario) throws SQLException {

PreparedStatement ps = null;

ResultSet rs = null;

Connection con = getConnection();

String sql = "SELECT count(id) FROM usuarios WHERE usuario = ? ";

try {

ps = con.prepareStatement(sql);

ps.setString(1, usuario);

rs = ps.executeQuery();

if (rs.next())

{

return rs.getInt(1);

}

return 1;

}catch (SQLException ex){

Logger.getLogger(Sqlusuarios.class.getName()) .log(Level.SEVERE, null, ex);

return 1;

}

}

Anexos

