

1. Documentación (portada y enunciado)

Documentación del proyecto “BarberShop”

Curso 2021/2022
IES Pio Baroja, Madrid

Autores: Freddy Andrade y Gabriel Mamani

2. Plan inicial

	Nombre de la tarea	Duración	Inicio	Finalizar	Predecesor...	Asignado a
1	Plan inicial	1d	09/05/22	09/05/22		
2	Modelo de datos	1d	09/05/22	09/05/22		Gabriel
3	Tablas y pantallas(botones) para IBMV	1d	09/05/22	09/05/22		Freddy
4	Diagrama de clases	1d	09/05/22	09/05/22		Gabriel
5	Diagrama de pantalla	1d	09/05/22	09/05/22		Freddy
6	diagrama de pantalla	1d	09/05/22	09/05/22		Freddy
7	Comportar Github y documentacion de los profes					
8	Código java de validación de datos de entrada	1d	13/05/22	13/05/22		Freddy
9	Creación de las tablas	1d	15/05/22	15/05/22		Gabriel
10	Conexión de las tablas	1d	15/05/22	15/05/22		Gabriel
11	Control de excepciones java	1d	16/05/22	16/05/22		Freddy
12	Implementación de clases y métodos	1d	18/05/22	18/05/22		Freddy
13	Plan de pruebas	1d	19/05/22	19/05/22		Gabriel
14	Diagrama de Pantallas	1d	20/05/22	20/05/22		Freddy
15	Diagrama de clases	1d	23/05/22	23/05/22		Gabriel
16	Documentación/** */	27d	29/05/22	04/07/22		
17	Requisitos de instalación					
18						








3. Bases de datos:

3.1. Modelo de datos

3.2. Tablas y pantallas (y botones) para IBMV

	I user	I user	I botón	B user	B pantalla	B botón	M user	M pantalla	M botón	V user	I pantalla	V botón
CLIENTES												
CodigoCliente	Usuario	Registro	"Registrarse"	Cliente	Mi perfil	Borrar	Cliente	Mi perfil	Editar	Cliente	Mi perfil	Perfil
DNI	Usuario	Registro	"Registrarse"	Cliente	Mi perfil	Borrar	Cliente	Mi perfil	Editar	Cliente	Mi perfil	Perfil
Nombre	Usuario	Registro	"Registrarse"	Cliente	Mi perfil	Borrar	Cliente	Mi perfil	Editar	Cliente	Mi perfil	Perfil
Telefono	Usuario	Registro	"Registrarse"	Cliente	Mi perfil	Borrar	Cliente	Mi perfil	Editar	Cliente	Mi perfil	Perfil
Direccion	Usuario	Registro	"Registrarse"	Cliente	Mi perfil	Borrar	Cliente	Mi perfil	Editar	Cliente	Mi perfil	Perfil
Email	Usuario	Registro	"Registrarse"	Cliente	Mi perfil	Borrar	Cliente	Mi perfil	Editar	Cliente	Mi perfil	Perfil
Password	Usuario	Registro	"Registrarse"	Cliente	Mi perfil	Borrar	Cliente	Mi perfil	Editar	Cliente	Mi perfil	Perfil
EMPLEADOS												
CodigoEmpleado	empleados	Registro	"Registrarse"	Empleado	Mi perfil	Borrar	Empleado	Mi perfil	Editar	Empleado	Mi perfil	Mi perfil
DNI	empleados	Registro	"Registrarse"	Empleado	Mi perfil	Borrar	Empleado	Mi perfil	Editar	Empleado	Mi perfil	Mi perfil
Nombre	empleados	Registro	"Registrarse"	Empleado	Mi perfil	Borrar	Empleado	Mi perfil	Editar	Empleado	Mi perfil	Mi perfil
Teléfono	empleados	Registro	"Registrarse"	Empleado	Mi perfil	Borrar	Empleado	Mi perfil	Editar	Empleado	Mi perfil	Mi perfil
Dirección	empleados	Registro	"Registrarse"	Empleado	Mi perfil	Borrar	Empleado	Mi perfil	Editar	Empleado	Mi perfil	Mi perfil
Especialidad	empleados	Registro	"Registrarse"	Empleado	Mi perfil	Borrar	Empleado	Mi perfil	Editar	Empleado	Mi perfil	Mi perfil
PAGOS												
CodigoPagos	Programa	Programa	Programa	Programa	Programa	Programa	Programa	Programa	Programa	Programa	Programa	Programa
CodigoEmpleado	Programa	Programa	Programa	Programa	Programa	Programa	Programa	Programa	Programa	Programa	Programa	Programa
FormaPago	Programa	Programa	Programa	Programa	Programa	Programa	Programa	Programa	Programa	Programa	Programa	Programa
FechaPago	Programa	Programa	Programa	Programa	Programa	Programa	Programa	Programa	Programa	Programa	Programa	Programa
Cantidad	Programa	Programa	Programa	Programa	Programa	Programa	Programa	Programa	Programa	Programa	Programa	Programa
COSMETICOS												
CodigoCosmeticos	CodigoCosmeticos	Proovedor	cosmeticos	"Añadir"	Proovedor	Cosmeticos	Borrar	Proovedor	Cosmeticos	Editar	Proovedor	Cosmeticos
CodigoEmpleado	CodigoCosmeticos	Proovedor	cosmeticos	"Añadir"	Proovedor	Cosmeticos	Borrar	Proovedor	Cosmeticos	Editar	Proovedor	Cosmeticos
NombreCosmetico	CodigoCosmeticos	Proovedor	cosmeticos	"Añadir"	Proovedor	Cosmeticos	Borrar	Proovedor	Cosmeticos	Editar	Proovedor	Cosmeticos
CodigoCliente	CodigoCosmeticos	Proovedor	cosmeticos	"Añadir"	Proovedor	Cosmeticos	Borrar	Proovedor	Cosmeticos	Editar	Proovedor	Cosmeticos
PrecioProducto	CodigoCosmeticos	Proovedor	cosmeticos	"Añadir"	Proovedor	Cosmeticos	Borrar	Proovedor	Cosmeticos	Editar	Proovedor	Cosmeticos
Citas												
CodigoCitas	Pograma	Programa	Programa	Administrador	Administrador	Administrador	Administrador	Administrador	Administrador	Administrador	Administrador	Administrador
NombreEmpleado	Programa	Programa	Programa	Administrador	Administrador	Administrador	Administrador	Administrador	Administrador	Administrador	Administrador	Administrador
NombreCliente	Programa	Programa	Programa	Administrador	Administrador	Administrador	Administrador	Administrador	Administrador	Administrador	Administrador	Administrador
Fecha	Programa	Programa	Programa	Administrador	Administrador	Administrador	Administrador	Administrador	Administrador	Administrador	Administrador	Administrador
Hora	Programa	Programa	Programa	Administrador	Administrador	Administrador	Administrador	Administrador	Administrador	Administrador	Administrador	Administrador
SERVICIOS												
CodigoServicio	Usuario	Registro	"Registrarse"	Usuario	Registro	"Registrarse"	Usuario	Registro	"Registrarse"	Usuario	Registro	"Registrarse"
CodigoEmpleado	Usuario	Registro	"Registrarse"	Usuario	Registro	"Registrarse"	Usuario	Registro	"Registrarse"	Usuario	Registro	"Registrarse"
NombreEmpleado	Usuario	Registro	"Registrarse"	Usuario	Registro	"Registrarse"	Usuario	Registro	"Registrarse"	Usuario	Registro	"Registrarse"
CodigoCliente	Usuario	Registro	"Registrarse"	Usuario	Registro	"Registrarse"	Usuario	Registro	"Registrarse"	Usuario	Registro	"Registrarse"
NombreCliente	Usuario	Registro	"Registrarse"	Usuario	Registro	"Registrarse"	Usuario	Registro	"Registrarse"	Usuario	Registro	"Registrarse"
TipoServicio	Usuario	Registro	"Registrarse"	Usuario	Registro	"Registrarse"	Usuario	Registro	"Registrarse"	Usuario	Registro	"Registrarse"
ComisionServicio	Usuario	Registro	"Registrarse"	Usuario	Registro	"Registrarse"	Usuario	Registro	"Registrarse"	Usuario	Registro	"Registrarse"

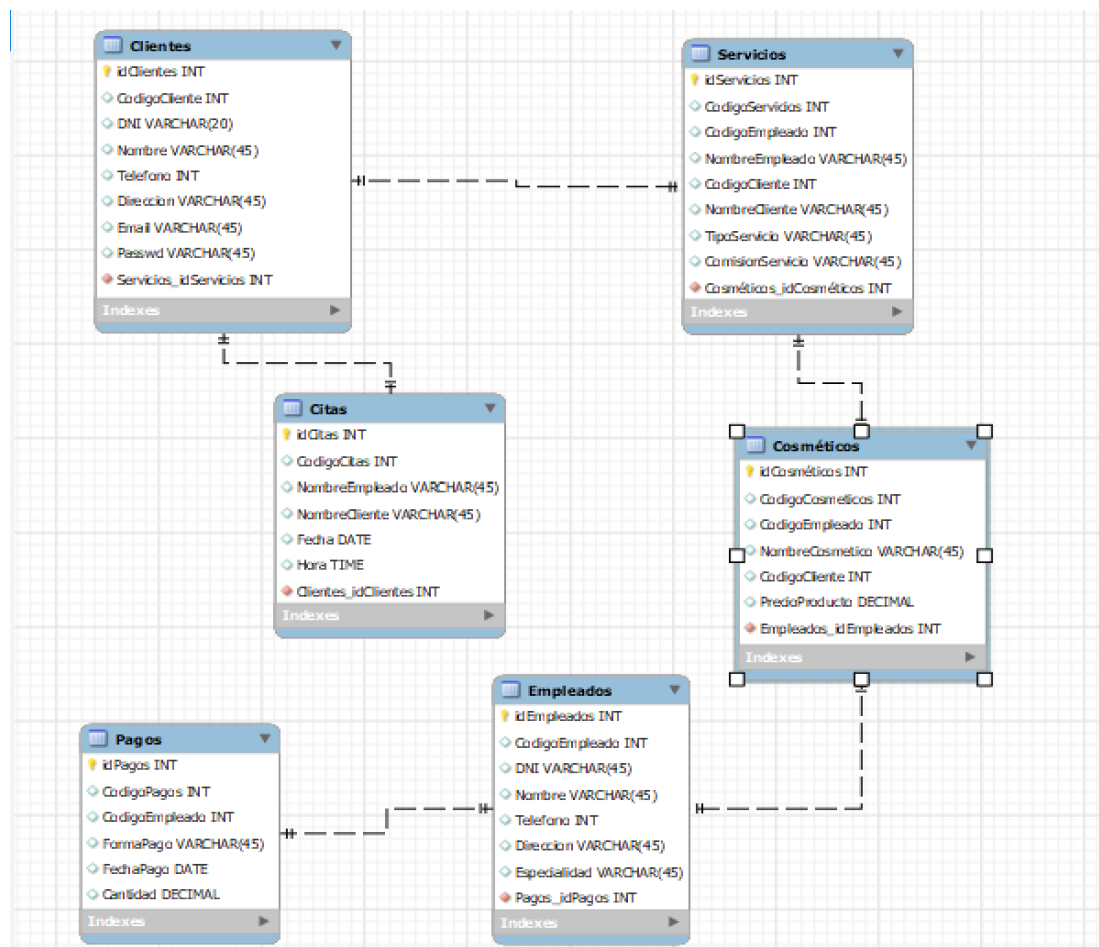
3.3. Creación de las tablas

▼ 	peluqueria	96,0 KiB
	citas	16,0 KiB
	clientes	16,0 KiB
	cosmeticos	16,0 KiB
	empleados	16,0 KiB
	pagos	16,0 KiB
	servicios	16,0 KiB

3.4. Conexión de la tablas

4. Programación

4.1. Diagrama de clases



4.2. Control de excepciones java

```
public static boolean InsertarServicios(Servicios Servicios) {
    boolean bRes = true;

    ConexionBBDD miConexion = new ConexionBBDD();
    try {
        miConexion.conectar();
        int iRes = miConexion.InsertarServicios(Servicios);
        System.out.println("Resultado de insertar el Servicio " + iRes);
    } catch (SQLException sqllex) {
        System.out.println("Error: " + sqllex.getMessage());
        sqllex.printStackTrace();
        bRes = false;
    } finally {
        miConexion.desconectar();
    }
    return bRes;
}
```

4.3. Implementación de clases y métodos

//////////////////////////////////CLASE CLIENTES//////////////////////////////////

```
package peluqueria;  
import java.util.Scanner;  
import javax.swing.JOptionPane;
```

```
public class Clientes {  
    int CodigoCliente;  
    String DNI;  
    String Nombre;  
    int Telefono;  
    String Direccion;  
    String Email;  
    String Passwd;
```

```
        public Clientes(int CodigoCliente, String DNI,String Nombre, int Telefono ,String  
Direccion , String Email, String Passwd){  
//ATRIBUTOS DE LA SUPERCLASE  
this.CodigoCliente=CodigoCliente;  
this.DNI = DNI;  
this.Nombre = Nombre;  
this.Telefono=Telefono;  
this.Direccion=Direccion;  
this.Email = Email;  
this.Passwd = Passwd;  
  
}
```

```
        public int getCodigoCliente() {  
            return CodigoCliente;  
        }
```

```
        public void setCodigoCliente(int CodigoCliente) {  
            Scanner entrada = new Scanner(System.in);  
            int chiv=0;
```

```

        String id = String.valueOf(CodigoCliente);
        boolean p = id.matches("{0-10}");

do{

if(id.length()<=10 && p){
    this.CodigoCliente = CodigoCliente;
    chiv=0;

    }

else {

    try {

        System.out.print("Introduce correctamente el código del cliente");

        CodigoCliente=entrada.nextInt();
        chiv=1;

    }

    catch(Exception ex) {
        ex.printStackTrace();
    }
}

while(chiv==1);
entrada.close();

}

public String getDNI() {
    return DNI;
}

public void setDNI(String DNI) {
    Scanner entrada = new Scanner(System.in);
    int chiv=0;
    String id = String.valueOf(DNI);
    boolean p = id.matches("[0-9]{8}[A-Za-z]{1}");

```

```

do{

if(id.length()<=9 && p){
    this.DNI = DNI;
    chiv=0;

    }

else {

    try {

        System.out.print("Introduce correctamente el DNI del empleado");

        DNI=entrada.toString();
        chiv=1;

    }

    catch(Exception ex) {
        ex.printStackTrace();
    }
}

}

while(chiv==1);
entrada.close();

}

public String getNombre() {
    return Nombre;
}

public void setNombre(String Nombre) {
    Scanner entrada = new Scanner(System.in);
    int chiv=0;
    String id = String.valueOf(Nombre);
    boolean p = id.matches("a-zA-ZñÑáéíóúÁÉÍÓÚüÜ]{2,25}");

do{

if(id.length()<=25 && p){
    this.Nombre = Nombre;

```

```

        chiv=0;

    }

else {

    try {

        System.out.print("Introduce correctamente el nombre del cliente");

        Nombre=entrada.next();
        chiv=1;

    }

    catch(Exception ex) {
        ex.printStackTrace();
    }
}

while(chiv==1);
entrada.close();

}

public int getTelefono() {
    return Telefono;
}

public void setTelefono(int telefono) {
    Scanner entrada = new Scanner(System.in);
    int chiv=0;
    String id = String.valueOf(telefono);
    boolean p = id.matches("[0-9]{9}");

do{

if(id.length()<=9 && p){
    this.Telefono = Telefono;
    chiv=0;

    }

else {

```



```

try {

    System.out.print("Introduce correctamente el teléfono del cliente");

    telefono=entrada.nextInt();
    chiv=1;

}

catch(Exception ex) {
    ex.printStackTrace();
}
}

while(chiv==1);
entrada.close();

}

public String getDireccion() {
    return Direccion;
}

public void setdireccion(String Direccion) {
    Scanner entrada = new Scanner(System.in);
    int chiv=0;
    String id = String.valueOf(Direccion);
    boolean p = id.matches("[a-zA-ZÀ-ÿ\\u00f1\\u00d1\\s]{2,250}");

do{

if(id.length()<=250 && p){
    this.Direccion = Direccion;
    chiv=0;

}

else {

try {

    System.out.print("Introduce correctamente la dirección del cliente");

```

```

        Direccion=entrada.toString();
        chiv=1;

    }

    catch(Exception ex) {
        ex.printStackTrace();
    }
}

while(chiv==1);
entrada.close();

}

public String getEmail() {
    return Email;
}

public void setEmail(String Email) {
    Scanner entrada = new Scanner(System.in);
    int chiv=0;
    String id = String.valueOf(Email);
    boolean p = id.matches("[A-Z0-9._%-]+@[A-Z0-9.-]+\\.[A-Z]{2,4}");

do{

if(id.length()<=9 && p){
    this.Email = Email;
    chiv=0;

    }

else {

    try {

        System.out.print("Introduce correctamente el email del cliente");

        Email=entrada.toString();
        chiv=1;

```

```

    }

    catch(Exception ex) {
        ex.printStackTrace();
    }
}

}

while(chiv==1);
entrada.close();

}

public String getPasswd() {
    return Email;
}

public void setPasswd(String Passwd) {
    Scanner entrada = new Scanner(System.in);
    int chiv=0;
    String id = String.valueOf(Passwd);
    boolean p = id.matches("[A-Z0-9._%-]+@[A-Z0-9.-]+\\.[A-Z]{2,4}");

do{

if(id.length()<=50 && p){
    this.Passwd = Passwd;
    chiv=0;

    }

else {

    try {

        System.out.print("Introduce correctamente la contraseña del cliente");

        Passwd=entrada.toString();
        chiv=1;

    }

    catch(Exception ex) {

```

```

        ex.printStackTrace();
    }
}

}

while(chiv==1);
entrada.close();

}

}

```

////////////////////////////////// CLASE CITAS//////////////////////////////////

```

package peluqueria;
import java.sql.Date;
import java.sql.Time;
import java.util.Scanner;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

import javax.swing.JOptionPane;

public class Citas {
    intCodigoCitas;
    String NombreEmpleado;
    String NombreCliente;
    Date Fecha;
    Time Hora;

    public Citas(intCodigoCitas,String NombreEmpleado, String NombreCliente , Date Fecha,
        Time Hora){
//ATRIBUTOS DE LA SUPERCLASE
this.CodigoCitas = CodigoCitas;
this.NombreEmpleado = NombreEmpleado;
this.NombreCliente = NombreCliente;
this.Fecha = Fecha;
this.Hora=Hora;
}

```

```
public int getCodigoCitas() {  
    return CodigoCitas;  
}
```

```
public void setCodigoCitas(int CodigoCitas) {  
    Scanner entrada = new Scanner(System.in);  
    int chiv=0;  
    String id = String.valueOf(CodigoCitas);  
    boolean p = id.matches("{0-10}");
```

```
do{
```

```
    if(id.length()<=10 && p){  
        this.CodigoCitas = CodigoCitas;  
        chiv=0;  
    }
```

```
else {
```

```
    try {
```

```
        System.out.print("Introduce correctamente el código de la cita");
```

```
        CodigoCitas=entrada.nextInt();
```

```
        chiv=1;
```

```
    }
```

```
    catch(Exception ex) {  
        ex.printStackTrace();
```

```
    }
```

```
}
```

```
}
```

```
while(chiv==1);  
entrada.close();
```

```
}
```

```
public String getNombreEmpleado() {
    return NombreEmpleado;
}

public void setNombreEmpleado(String NombreEmpleado) {
    Scanner entrada = new Scanner(System.in);
    int chiv=0;
    String id = String.valueOf(NombreEmpleado);
    boolean p = id.matches("{0-10}");

do{

if(id.length()<=10 && p){
    this.NombreEmpleado = NombreEmpleado;
    chiv=0;

    }

else {

    try {

        System.out.print("Introduce correctamente el nombre del empleado");

        NombreEmpleado=entrada.toString();
        chiv=1;

    }

    catch(Exception ex) {
        ex.printStackTrace();
    }
}

while(chiv==1);
entrada.close();

}
```

```

    public String getNombreCliente() {
        return NombreCliente;
    }

    public void setnombre(int CodigoCliente) {
        Scanner entrada = new Scanner(System.in);
        int chiv=0;
        String id = String.valueOf(CodigoCliente);
        boolean p = id.matches("{0-10}");

do{

if(id.length()<=3 && p){
    this.NombreCliente = NombreCliente;
    chiv=0;

    }

else {

    try {

        System.out.print("Introduce correctamente el nombre del cliente");

        CodigoCliente=entrada.nextInt();
        chiv=1;

    }

    catch(Exception ex) {
        ex.printStackTrace();
    }
}

while(chiv==1);
entrada.close();

}

    public Date getFecha() {
        return Fecha;
    }

    public void settelefono(Date Fecha) {
        Scanner entrada = new Scanner(System.in);

```

```

        int chiv=0;
        String id = String.valueOf(Fecha);
        boolean p = id.matches("[1-9][12][0-9]3[01]");
do{

if(id.length()<=20 && p){
    this.Fecha = Fecha;
    chiv=0;

    }

else {

    try {

        System.out.print("Introduce correctamente el la fecha");

        chiv=1;

    }

    catch(Exception ex) {
        ex.printStackTrace();
    }
}

}

while(chiv==1);
entrada.close();

}

public Time getHora() {
    return Hora;
}

public void setdireccion(Time Hora) {
    Scanner entrada = new Scanner(System.in);
    int chiv=0;
    String id = String.valueOf(Hora);
    boolean p = id.matches("[01]?[0-9]2[0-3]:[0-5][0-9]");

```



```

do{

if(id.length()<=10 && p){
    this.Hora = Hora;
    chiv=0;

    }

else {

    try {

        System.out.print("Introduce correctamente la hora");

        chiv=1;

    }

    catch(Exception ex) {
        ex.printStackTrace();
    }
}

while(chiv==1);
entrada.close();

}
}

```

////////////////////CLASE EMPLEADOS////////////////////

```

package peluqueria;
import java.util.Scanner;
import javax.swing.JOptionPane;

public class Empleados {
    int CodigoEmpleado;
    String dni;
    String nombre;
    int telefono;

```

```
String direccion;  
String especialidad;
```

```
    public Empleados(int CodigoEmpleado, String dni,String nombre, int telefono ,String  
direccion , String especialidad){  
//ATRIBUTOS DE LA SUPERCLASE  
this.CodigoEmpleado=CodigoEmpleado;  
this.dni = dni;  
this.nombre = nombre;  
this.telefono=telefono;  
this.direccion=direccion;  
this.especialidad = especialidad;  
}
```

```
    public int getCodigoEmpleado() {  
        return CodigoEmpleado;  
    }
```

```
    public void setCodigoEmpleado(int CodigoEmpleado) {  
        Scanner entrada = new Scanner(System.in);  
        int chiv=0;  
        String id = String.valueOf(CodigoEmpleado);  
        boolean p = id.matches("{0-10}");  
  
        do{  
  
            if(id.length()<=10 && p){  
                this.CodigoEmpleado = CodigoEmpleado;  
                chiv=0;  
  
            }  
  

```

```
        else {
```

```
            try {
```

```
                System.out.print("Introduce correctamente el código del empleado");
```

```
            }  
  
            CodigoEmpleado=entrada.nextInt();  
            chiv=1;
```

```

    }

    catch(Exception ex) {
        ex.printStackTrace();
    }
}

}

while(chiv==1);
entrada.close();

}

public String dni() {
    return dni;
}

public void setdni(String dni) {
    Scanner entrada = new Scanner(System.in);
    int chiv=0;
    String id = String.valueOf(dni);
    boolean p = id.matches("[0-9]{8}[A-Za-z]{1}");

do{

if(id.length()<=9 && p){
    this.dni = dni;
    chiv=0;

    }

else {

    try {

        System.out.print("Introduce correctamente el DNI del empleado");

        dni=entrada.next();
        chiv=1;

    }

    catch(Exception ex) {

```

```

        ex.printStackTrace();
    }
}

while(chiv==1);
entrada.close();

}

public String getnombre() {
    return nombre;
}

public void setnombre(String nombre) {
    Scanner entrada = new Scanner(System.in);
    int chiv=0;
    String id = String.valueOf(nombre);
    boolean p = id.matches("a-zA-ZñÑáéíóúÁÉÍÓÚüÜ]{2,25}");

do{

if(id.length()<=25 && p){
    this.nombre = nombre;
    chiv=0;

    }

else {

    try {

        System.out.print("Introduce correctamente el nombre del empleado");

        nombre=entrada.next();
        chiv=1;

    }

    catch(Exception ex) {
        ex.printStackTrace();
    }
}

}

```

```

while(chiv==1);
entrada.close();

    }
    public int gettelefono() {
        return telefono;
    }

    public void settelefono(int telefono) {
        Scanner entrada = new Scanner(System.in);
        int chiv=0;
        String id = String.valueOf(telefono);
        boolean p = id.matches("[0-9]{9}");

do{

if(id.length()<=9 && p){
    this.telefono = telefono;
    chiv=0;

    }

else {

    try {

        System.out.print("Introduce correctamente el teléfono del cliente");

        telefono=entrada.nextInt();
        chiv=1;

    }

    catch(Exception ex) {
        ex.printStackTrace();
    }
}

}

while(chiv==1);
entrada.close();

    }

```

```

        public String getdireccion() {
            return direccion;
        }

        public void setdireccion(String direccion) {
            Scanner entrada = new Scanner(System.in);
            int chiv=0;
            String id = String.valueOf(direccion);
            boolean p = id.matches("[a-zA-ZÀ-ÿ\\u00f1\\u00d1\\s]{2,250}");

do{

if(id.length()<=250 && p){
    this.direccion = direccion;
    chiv=0;

    }

else {

    try {

        System.out.print("Introduce correctamente la dirección del cliente");

        direccion=entrada.next();
        chiv=1;

    }

    catch(Exception ex) {
        ex.printStackTrace();
    }
}

}

while(chiv==1);
entrada.close();

}

        public String getespecialidad() {
            return especialidad;

```

```

    }

    public void setespecialidad(String especialidad) {
        Scanner entrada = new Scanner(System.in);
        int chiv=0;
        String id = String.valueOf(especialidad);
        boolean p = id.matches("a-zA-ZñÑáéíóúÁÉÍÓÚüÜ]{2,25}");

        do{

            if(id.length()<=25 && p){
                this.especialidad = especialidad;
                chiv=0;

            }

            else {

                try {

                    System.out.print("Introduce correctamente la especialidad del empleado");

                    especialidad=entrada.next();
                    chiv=1;

                }

                catch(Exception ex) {
                    ex.printStackTrace();
                }
            }

        }

        while(chiv==1);
        entrada.close();

    }

}

```

////////////////////////////////CLASE COSMÉTICOS////////////////////////////////

```

package peluqueria;
import java.sql.Date;
import java.sql.Time;
import java.text.DecimalFormat;
import java.util.Scanner;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

import javax.swing.JOptionPane;

public class Cosmeticos {
    int CodigoCosmeticos;
    int CodigoEmpleado;
    String NombreCosmetico;
    int CodigoCliente;
    float PrecioProducto;

    public Cosmeticos(int CodigoCosmeticos, int CodigoEmpleado, String
NombreCosmetico, int CodigoCliente, float PrecioProducto){

//ATRIBUTOS DE LA SUPERCLASE

this.CodigoCosmeticos = CodigoCosmeticos;
this.CodigoEmpleado = CodigoEmpleado;
this.NombreCosmetico = NombreCosmetico;
this.CodigoCliente = CodigoCliente;
this.PrecioProducto=PrecioProducto;

}

    public int CodigoCosmeticos() {
        return CodigoCosmeticos;
    }

    public void setCodigoCitas(int CodigoCosmeticos) {
        Scanner entrada = new Scanner(System.in);
        int chiv=0;
        String id = String.valueOf(CodigoCosmeticos);
        boolean p = id.matches("{0-10}");

```



```

do{

if(id.length()<=10 && p){
    this.CodigoCosmeticos = CodigoCosmeticos;
    chiv=0;

    }

else {

    try {

        System.out.print("Introduce correctamente el código del cosmético");

        CodigoCosmeticos=entrada.nextInt();

        chiv=1;

    }

    catch(Exception ex) {
        ex.printStackTrace();
    }
}

}

while(chiv==1);
entrada.close();

}

public int CodigoEmpleado() {
    return CodigoEmpleado;
}

public void setCodigoEmpleado(int CodigoEmpleado) {
    Scanner entrada = new Scanner(System.in);
    int chiv=0;
    String id = String.valueOf(CodigoEmpleado);
    boolean p = id.matches("{0-10}");

```

```

do{

if(id.length()<=10 && p){
    this.CodigoEmpleado = CodigoEmpleado;
    chiv=0;

    }

else {

    try {

        System.out.print("Introduce correctamente el código del empleado");

        CodigoEmpleado=entrada.nextInt();
        chiv=1;

    }

    catch(Exception ex) {
        ex.printStackTrace();
    }
}

}

while(chiv==1);
entrada.close();

}

public String getNombreCosmetico() {
    return NombreCosmetico;
}

public void setNombreCosmetico(String NombreCosmetico) {
    Scanner entrada = new Scanner(System.in);
    int chiv=0;
    String id = String.valueOf(NombreCosmetico);
    boolean p = id.matches("{0-10}");

do{

```

```

if(id.length()<=10 && p){
    this.NombreCosmetico = NombreCosmetico;
    chiv=0;

    }

else {

    try {

        System.out.print("Introduce correctamente el nombre del cosmético");

        chiv=1;

    }

    catch(Exception ex) {
        ex.printStackTrace();
    }
}

while(chiv==1);
entrada.close();

}

public int getCodigoCliente() {
    return CodigoCliente;
}

public void setnombre(int CodigoCliente) {
    Scanner entrada = new Scanner(System.in);
    int chiv=0;
    String id = String.valueOf(CodigoCliente);
    boolean p = id.matches("{0-10}");

do{

if(id.length()<=3 && p){
    this.CodigoCliente = CodigoCliente;
    chiv=0;

    }

```

```

else {

    try {

        System.out.print("Introduce correctamente el código del cliente");

        CodigoCliente=entrada.nextInt();
        chiv=1;

    }

    catch(Exception ex) {
        ex.printStackTrace();
    }
}

while(chiv==1);
entrada.close();

}

public float getPrecioProducto() {
    return PrecioProducto;
}

public void setdireccion(float PrecioProducto) {
    Scanner entrada = new Scanner(System.in);
    int chiv=0;
    String id = Float.toString(PrecioProducto);
    boolean p = id.matches("[0-9]*");

do{

if(id.length()<=5 && p){
    this.PrecioProducto = PrecioProducto;
    chiv=0;

    }

else {

```

```

        try {

            System.out.print("Introduce correctamente el precio");

            chiv=1;

        }

        catch(Exception ex) {
            ex.printStackTrace();
        }
    }

}

while(chiv==1);
entrada.close();

}

}

```

////////////////////////////////// CLASE PAGOS//////////////////////////////////

```

package peluqueria;

import java.util.Scanner;
import javax.swing.JOptionPane;

public class Pagos {
    int CodigoPagos;
    int CodigoEmpleado;
    String FormaPago;
    String FechaPago;
    float cantidad;

    public Pagos(int CodigoPagos,int CodigoEmpleado ,String FormaPago,String
FechaPago,float  cantidad){
//ATRIBUTOS DE LA SUPERCLASE
this.CodigoPagos = CodigoPagos;
this.CodigoEmpleado = CodigoEmpleado;

```

```
this.FormaPago = FormaPago;  
this.FechaPago = FechaPago;  
this.cantidad = cantidad;
```

```
}
```

```
public int CodigoPagos() {  
    return CodigoPagos;  
}
```

```
public void setCodigoPagos(int CodigoPagos) {  
    Scanner entrada = new Scanner(System.in);  
    int chiv=0;  
    String id = String.valueOf(CodigoPagos);  
    boolean p = id.matches("{0-10}");
```

```
do{
```

```
if(id.length()<=10 && p){  
    this.CodigoPagos = CodigoPagos;  
    chiv=0;
```

```
}
```

```
else {
```

```
try {
```

```
    System.out.print("Introduce correctamente el codigo del pago");
```

```
    CodigoPagos=entrada.nextInt();  
    chiv=1;
```

```
}
```

```
catch(Exception ex) {  
    ex.printStackTrace();
```

```
}
```

```
}
```

```
}
```

```
while(chiv==1);  
entrada.close();
```

```
}
```

```

    public int CodigoEmpleado() {
        return CodigoEmpleado;
    }

    public void setCodigoEmpleado(int CodigoEmpleado) {
        Scanner entrada = new Scanner(System.in);
        int chiv=0;
        String id = String.valueOf(CodigoEmpleado);
        boolean p = id.matches("{0-10}");

        do{

            if(id.length()<=10 && p){
                this.CodigoEmpleado = CodigoEmpleado;
                chiv=0;

            }

            else {

                try {

                    System.out.print("Introduce correctamente el código del empleado");

                    CodigoEmpleado=entrada.nextInt();
                    chiv=1;

                }

                catch(Exception ex) {
                    ex.printStackTrace();
                }
            }

        }

        while(chiv==1);
        entrada.close();

    }

```

```

    public String getFormaPago() {
        return FormaPago;
    }

    public void setFormaPago(String FormaPago) {
        Scanner entrada = new Scanner(System.in);
        int chiv=0;
        String id = String.valueOf(FormaPago);
        boolean p = id.matches("a-zA-ZñÑáéíóúÁÉÍÓÚüÜ]{2,20}");

do{

if(id.length()<=20 && p){
    this.FormaPago = FormaPago;
    chiv=0;

    }

else {

    try {

        System.out.print("Introduce correctamente la forma de pago ");

        FormaPago=entrada.next();
        chiv=1;

    }

    catch(Exception ex) {
        ex.printStackTrace();
    }
}

}

while(chiv==1);
entrada.close();

}

    public String getFechaPago() {

```



```

        return FormaPago;
    }

    public void setFechaPago(String FechaPago) {
        Scanner entrada = new Scanner(System.in);
        int chiv=0;
        String id = String.valueOf(FechaPago);
        boolean p = id.matches("[01]?[0-9]|2[0-3]:[0-5][0-9]");

        do{

            if(id.length()<=5 && p){
                this.FechaPago = FechaPago;
                chiv=0;

            }

            else {

                try {

                    System.out.print("Introduce correctamente la fecha del pago ");

                    FechaPago=entrada.next();
                    chiv=1;

                }

                catch(Exception ex) {
                    ex.printStackTrace();
                }
            }

        }

        while(chiv==1);
        entrada.close();
    }

```

```

    public float getcantidad() {
        return cantidad;
    }

```

```

    public void setcantidad(float cantidad) {
        Scanner entrada = new Scanner(System.in);

```

```

        int chiv=0;
        String id = Float.toString(cantidad);
        boolean p = id.matches("[0-9]*");

        do{

            if(id.length()<=5 && p){
                this.cantidad = cantidad;
                chiv=0;

            }

            else {

                try {

                    System.out.print("Introduce correctamente la cantidad de dinero ");

                    cantidad=entrada.nextInt();
                    chiv=1;

                }

                catch(Exception ex) {
                    ex.printStackTrace();
                }
            }

        }

        while(chiv==1);
        entrada.close();
    }

}

```

//////////////////////////////////CLASE SERVICIOS//////////////////////////////////

```

package peluqueria;
import java.sql.Date;
import java.sql.Time;
import java.text.DecimalFormat;
import java.util.Scanner;

```

```
import java.util.regex.Matcher;
import java.util.regex.Pattern;
import javax.swing.JOptionPane;
```

```
public class Servicios {
    int CodigoServicio;
    int CodigoEmpleado;
    String NombreEmpleado;
    int CodigoCliente;
    String NombreCliente;
    String TipoServicio;
    float ComisionServicio;
    public Servicios(int CodigoServicio,int CodigoEmpleado ,String NombreEmpleado, int
CodigoCliente,String TipoServicio, float ComisionServicio){
//ATRIBUTOS DE LA SUPERCLASE
this.CodigoServicio = CodigoServicio;
this.CodigoEmpleado = CodigoEmpleado;
this.NombreEmpleado=NombreEmpleado;
this.CodigoCliente = CodigoCliente;
this.NombreCliente=NombreCliente;
this.TipoServicio = TipoServicio;
this.ComisionServicio=ComisionServicio;
}
}
```

```
    public int getCodigoServicio() {
        return CodigoServicio;
    }
```

```
    public void setCodigoServicio(int CodigoServicio) {
        Scanner entrada = new Scanner(System.in);
        int chiv=0;
        String id = String.valueOf(CodigoServicio);
        boolean p = id.matches("[0-10]");
```

```
do{

    if(id.length()<=10 && p){
        this.CodigoServicio = CodigoServicio;
        chiv=0;

    }
```

```
else {
```

```

try {

    System.out.print("Introduce correctamente el código del servicio");

    CodigoServicio=entrada.nextInt();

    chiv=1;

}

catch(Exception ex) {
    ex.printStackTrace();
}
}

while(chiv==1);
entrada.close();
}

public int getCodigoEmpleado() {
    return CodigoEmpleado;
}

public void setCodigoEmpleado(int CodigoEmpleado) {
    Scanner entrada = new Scanner(System.in);
    int chiv=0;
    String id = String.valueOf(CodigoEmpleado);
    boolean p = id.matches("[0-10]");

do{

if(id.length()<=10 && p){
    this.CodigoEmpleado = CodigoEmpleado;
    chiv=0;

    }

else {

    try {

        System.out.print("Introduce correctamente el código del empleado");

```

```

        CodigoEmpleado=entrada.nextInt();
        chiv=1;

    }

    catch(Exception ex) {
        ex.printStackTrace();
    }
}

}

while(chiv==1);
entrada.close();

}

public String getNombreEmpleado() {
    return NombreEmpleado;
}

public void setNombreEmpleado(String NombreEmpleado) {
    Scanner entrada = new Scanner(System.in);
    int chiv=0;
    String id = String.valueOf(NombreEmpleado);
    boolean p = id.matches("[0-10]");

do{

if(id.length()<=10 && p){
    this.NombreEmpleado = NombreEmpleado;
    chiv=0;

    }

else {

    try {

        System.out.print("Introduce correctamente el nombre del empleado");

```

```

        NombreEmpleado=entrada.toString();
        chiv=1;

    }

    catch(Exception ex) {
        ex.printStackTrace();
    }
}

while(chiv==1);
entrada.close();

}

public int getCodigoCliente() {
    return CodigoCliente;
}

public void setnombre(int CodigoCliente) {
    Scanner entrada = new Scanner(System.in);
    int chiv=0;
    String id = String.valueOf(CodigoCliente);
    boolean p = id.matches("[0-10]");

do{

if(id.length()<=10 && p){
    this.CodigoCliente = CodigoCliente;
    chiv=0;

    }

else {

    try {

```

```

        System.out.print("Introduce correctamente el código del cliente");

       CodigoCliente=entrada.nextInt();
        chiv=1;

    }

    catch(Exception ex) {
        ex.printStackTrace();
    }
}

while(chiv==1);
entrada.close();

}

public String getNombreCliente() {
    return NombreCliente;
}

public void setNombreCliente(String NombreCliente) {
    Scanner entrada = new Scanner(System.in);
    int chiv=0;
    String id = String.valueOf(NombreCliente);
    boolean p = id.matches("[0-10]");

do{

    if(id.length()<=10 && p){
        this.NombreCliente = NombreCliente;
        chiv=0;

    }

else {

    try {

        System.out.print("Introduce correctamente el nombre del cliente");

```

```

        NombreEmpleado=entrada.toString();
        chiv=1;

    }

    catch(Exception ex) {
        ex.printStackTrace();
    }
}

}

while(chiv==1);
entrada.close();

}

public String getTipoServicio() {
    return TipoServicio;
}

public void setNombreCosmetico(String TipoServicio) {
    Scanner entrada = new Scanner(System.in);
    int chiv=0;
    String id = String.valueOf(TipoServicio);
    boolean p = id.matches("a-zA-ZñÑáéíóúÁÉÍÓÜüÛ]{2,25}");

do{

if(id.length()<=25 && p){
    this.TipoServicio = TipoServicio;
    chiv=0;

    }

else {

    try {

        System.out.print("Introduce correctamente el tipo de servicio");

```



```

        chiv=1;

    }

    catch(Exception ex) {
        ex.printStackTrace();
    }
}

while(chiv==1);
entrada.close();

}

public float getComisionServicio() {
    return ComisionServicio;
}

public void setComisionServicio(float ComisionServicio) {
    Scanner entrada = new Scanner(System.in);
    int chiv=0;
    String id = Float.toString(ComisionServicio);
    boolean p = id.matches("[0-9]*");

do{

if(id.length()<=9 && p){
    this.ComisionServicio = ComisionServicio;
    chiv=0;

    }

else {

    try {

        System.out.print("Introduce correctamente la comisi3n del servicio");
        ComisionServicio=entrada.nextInt();

        chiv=1;

    }

```

```

        catch(Exception ex) {
            ex.printStackTrace();
        }
    }

    while(chiv==1);
    entrada.close();

}

```

4.4. Plan de pruebas

4.4.1. De pantalla

Cientes	Null	Range	Tamaño	Tipo de Datos	otros	codigo java	Pattern
CodigoCliente	Not Null	1-10 caracteres	10	INT	Public		boolean p = id.matches("{0-10}");
DNI	Not Null	9 caracteres	9	VARCHAR	Public		boolean p = id.matches("[0-9]{8}[A-Za-z]{1}");
Nombre	Not Null	2-25 caracteres	25	VARCHAR	Public		boolean p = id.matches("a-zA-ZñÑ áéíóúÁÉÍÓÚüÛ]{2,25} ");
Teléfono	Not Null	9 caracteres	9	INT	Public		boolean p = id.matches("[0-9]{9}");
Dirección	Not Null	2-60 caracteres	60	VARCHAR	Public		boolean p = id.matches("([a-zA-ZÀ- ÿ\\u00f1\\u00d1\\s]{2, 60})");
Email	Not Null	8-30 caracteres	30	VARCHAR	Public		boolean p = id.matches("[A-Z0-9._ %-]+@[A-Z0-9.-]+\\.[A- Z]{2,4}");
Passwd	Not Null	8-30 caracteres	30	VARCHAR	Public		boolean p = id.matches("a-zA-ZñÑ áéíóúÁÉÍÓÚüÛ]{1,30} ");

Empleados	Null	Range	Tamaño	Tipo de Datos	otros	codigo java	Pattern
CodigoEmpleado	Not Null	1-10 caracteres	10	INT	Public		boolean p = id.matches("{0-10}");
DNI	Not Null	9 Caract	9	VARCHAR	Public		boolean p = id.matches("[0-9]{8}[A-Za-z]{1}");
Nombre	Null	2-25 caracteres	25	VARCHAR	Public		boolean p = id.matches("a-zA-ZñÑáéíóúÁÉÍÓÚüÛ){2,25}");
Telefono	Null	9 Caract	9	VARCHAR	Public		boolean p = id.matches("[0-9]{9}");
Direccion	Null	2-60 Caract	60	VARCHAR	Public		boolean p = id.matches("[a-zA-ZÀ-ÿ\\u00f1\\u00d1\\s]{2,60}");
Especialidad	Null	2-60 Caract	60	VARCHAR	Public		boolean p = id.matches("a-zA-ZñÑáéíóúÁÉÍÓÚüÛ){2,60}");

Pagos	Null	Range	Tamaño	Tipo de Datos	otros	codigo java	Pattern
CodigoPagos	Not Null	1-10 caracteres	10	INT	Public		boolean p = id.matches("{0-10}");
CodigoEmpleado	Not Null	1-10 caracteres	10	INT	Public		boolean p = id.matches("{0-10}");
FormaPago	Not Null	2-40 caracteres	40	VARCHAR	Public		boolean p = id.matches("a-zA-ZñÑáéíóúÁÉÍÓÚüÛ){2,40}");
fechaPago	Not Null			DATE	Public		boolean p = id.matches("[01]?[0-9][2[0-31]:[0-5][0-9]");
Cantidad	Not Null	0,0f	4 bites	FLOAT	Public		boolean p = id.matches("[0-9]*");

Citas	Null	Range	Tamaño	Tipo de Datos	otros	codigo java	Pattern
CodigoCitas	Not Null	1-10 caracteres	10	INT	Public		boolean p = id.matches("{0-10}");
NombreEmpleado	Null	2-25 caracteres	25	VARCHAR	Public		boolean p = id.matches("a-zA-ZñÑ áéíóúÁÉÍÓÚüÛ}{2,25}");
NombreCliente	Null	2-25 caracteres	25	VARCHAR	Public		boolean p = id.matches("a-zA-ZñÑ áéíóúÁÉÍÓÚüÛ}{2,25}");
Fecha	Not Null			DATE	Public		boolean p = id.matches("[1-9][12][0-9][3[01]]");
Hora	Not Null			TIME	Public		boolean p = id.matches("[01]?[0-9][2[0-3]]:[0-5][0-9]");

Servicios	Null	Range	Tamaño	Tipo de Datos	otros	codigo java	Pattern
CodigoServicio	Not Null	1-10 caracteres	10	INT	Public		boolean p = id.matches("{0-10}");
CodigoEmpleado	Not Null	1-10 caracteres	10	VARCHAR	Public		boolean p = id.matches("{0-10}");
NombreEmpleado	Not Null	2-25 caracteres	25	VARCHAR	Public		boolean p = id.matches("a-zA-ZñÑ áéíóúÁÉÍÓÚüÛ){2,25}");
CodigoCliente	Not Null	1-10 caracteres	10	INT	Public		boolean p = id.matches("{0-10}");
NombreCliente	Not Null	2-25 caracteres	25	VARCHAR	Public		boolean p = id.matches("a-zA-ZñÑ áéíóúÁÉÍÓÚüÛ){2,25}");
TipoServicio	Not Null	2-20 Caract	20	VARCHAR	Public		boolean p = id.matches("a-zA-ZñÑ áéíóúÁÉÍÓÚüÛ){2,20}");
ComisionServicio	Not Null	0,0f	4 bites	FLOAT	Public		boolean p = id.matches("[0-9]*");

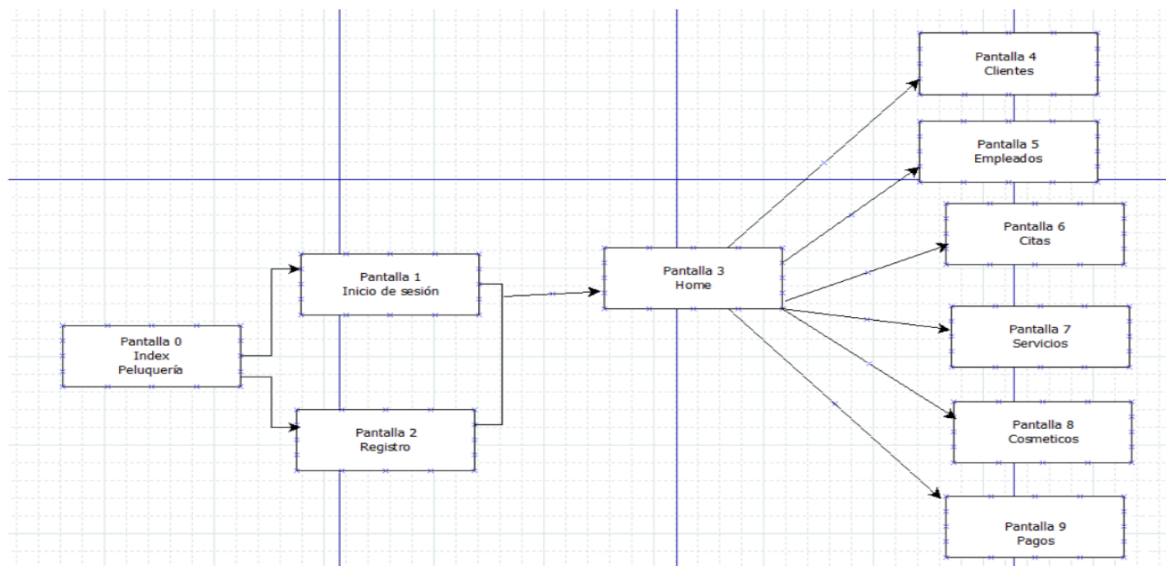
Cosmeticos	Null	Range	Tamaño	Tipo de Datos	otros	codigo java	Pattern
CodigCosmetico	Not Null	1-10 caracteres	10	INT	Public		boolean p = id.matches("{0-10}");
CodigoEmpleado	Not Null	1-10 caracteres	10	INT	Public		boolean p = id.matches("{0-10}");
NombreCosmetico	Not Null	2-25 Caract	25	VARCHAR	Public		boolean p = id.matches("a-zA-ZñÑ áéíóúÁÉÍÓÚüÿ]{2,25} ");
CodigoCliente	Not Null	1-10 caracteres	10	INT	Public		boolean p = id.matches("{0-10}");
PrecioProducto	Null	0.0f	4 bites	FLOAT	Public		boolean p = id.matches("[0-9]*");

4.5. Documentación

4.5.1. Documentación generada con java doc /** ... **/

5. Lenguaje de marcas

5.1 Diagrama de pantallas



4.6. Descripción de cada pantalla

Pantalla 0: index

En la pantalla index visualiza todos los botones que hay como:

inicio, acerca de , servicios, cosméticos , iniciar sesión y entrar como administrador.



Pantalla 1: inicio de sesión

En esta pantalla se crea una cuenta para aquellas personas que quieran comprar algún producto , pedir una cita.....

Pantalla 2: Registrarse

Esta pantalla permite crearte una cuenta si todavía no tienes cuenta y poder acceder a comprar productos , pedir cita....

Pantalla 3: home

La pantalla home permite volver al estado a la página de inicio.



Pantalla 4: Clientes

En esta pantalla permite visualizar los clientes registrados

Listado de Clientes

DNI	Nombre	Telefono	Direccion	Correo	Contraseña
52498564R	maria	625945857	C/pintor velasco	maria62@gmail.com	maria62
64955495P	luis	685985547	C/mariestela	luis68@gmail.com	luis68
69545628J	juan	669236482	C/mulatos	juan66@gmail.com	juan66
95648545F	antonio	696439546	C/escuelas	antonio69@gmail.com	antonio69
56954898T	rosa	644365248	C/trujillo	rosa64@gmail.com	rosaa64
62354895I	ana	673915689	C/valle hermoso	ana67@gmail.com	ana67
95235486S	hector	634856455	C/san victor	hector63@gmail.com	hector63

Pantalla 5: Empleados

En esta pantalla empleados permite visualizar los empleados registrados.

Listado de Empleados

DNI	Nombre	Telefono	Direccion	Especialidad
98654258F	lorena	648925494	C/ marcela	estilista
65235894H	rosario	649135645	C/ arboleo	peluquero
58994258X	franco	642815455	C/ bailen	peluquero
65204895B	manuela	645891256	C/ requena	estilista
64554587K	jorge	646564255	C/ juanilo	estilista
98564205J	nicole	654824599	C/ puerto	barbero

Pantalla 6: Citas


En esta pantalla me permite ver las citas creadas hasta el momento

Pantalla 7, 8 y 9 lo mismo pero con Servicios , Cosméticos y Pagos pero por lo alargado no estarán disponibles en la página.

5. Usuarios

5.1. Descripción de los usuarios y programación

Los usuarios pueden pedir tantas citas como comprar alguno de los cosméticos que aparecen en la página

5.2. Compartir GitHub y documentación con los profes 

5.3. Código java de validación de datos de entrada

Muestra que se introduzca correctamente el DNI del empleado

```
try {  
    System.out.print("Introduce correctamente el DNI del empleado");  
    DNI=entrada.toString();  
    chiv=1;  
}  
  
catch(Exception ex) {  
    ex.printStackTrace();  
}  
}  
  
while(chiv==1);  
entrada.close();  
}
```

Muestra que se introduzca correctamente el código del empleado

```
try {  
    System.out.print("Introduce correctamente el código del empleado");  
    CodigoEmpleado=entrada.nextInt();  
    chiv=1;  
}  
  
catch(Exception ex) {  
    ex.printStackTrace();  
}  
}  
  
while(chiv==1);  
entrada.close();  
}
```

Muestra que se introduzca correctamente el código de la cita.

```

        try {

            System.out.print("Introduce correctamente el código de la cita");

            CodigoCitas=entrada.nextInt();

            chiv=1;

        }

        catch(Exception ex) {
            ex.printStackTrace();
        }

    }

    while(chiv==1);
    entrada.close();

}

```

6. Seguimiento del plan inicial

Plan inicial	1d	09/05/22	09/05/22		
Modelo de datos	1d	09/05/22	09/05/22		Gabriel
Tablas y pantallas(botones) para IBMV	1d	09/05/22	09/05/22		Freddy
Diagrama de clases	1d	09/05/22	09/05/22		Gabriel
Diagrama de pantalla	1d	09/05/22	09/05/22		Freddy
diagrama de pantalla	1d	09/05/22	09/05/22		Freddy
Comportar Github y documentacion de los profes					
Código java de validación de datos de entrada	1d	13/05/22	13/05/22		Freddy
Creación de las tablas	1d	15/05/22	15/05/22		Gabriel
Conexión de las tablas	1d	15/05/22	15/05/22		Gabriel
Control de excepciones java	1d	16/05/22	16/05/22		Freddy
Implementación de clases y métodos	1d	18/05/22	18/05/22		Freddy
Plan de pruebas	1d	19/05/22	19/05/22		Gabriel
Diagrama de Pantallas	1d	20/05/22	20/05/22		Freddy
Diagrama de clases	1d	23/05/22	23/05/22		Gabriel
Documentación/** */	27d	29/05/22	04/07/22		
Requisitos de instalación					

7. Herramientas utilizadas

Eclipse para insertar código.

Heidi para crear la base de datos

Dia para crear los diagramas y el workbench para poner los atributos.

Smartsheet para hacer el plan inicial

Excel para las clases con sus características.

8. Webs consultadas

<https://webdesign.tutsplus.com/es/articles/20-amazing-bootstrap-template-to-try-in-2019--cms-32742>

