


App Agencias de viajes



usuario

Ingresar

Salir

App Agencias de viajes

Principal

AGENCIAS DE VIAJES MIAHUATLÁN

destino:

Dirección:

codigo postal:

Telefono:

Agregar

Actualizar

Eliminar


Item One

Item Two

Item Three

salir

App Agencias de viajes

 Transporte

AGENCIAS DE VIAJES MIAHUATLÁ


ingrese el nombre del transporte:


Ingrese el numero de acientos:


Agregar

Actualizar

Eliminar







One

Two

Three

Four

Salir

App Agencias de viajes

Reservación

AGENCIAS DE VIAJES MIAHUATLÁ

Eliga el Id del destino que desea solicitar

Ingrese el Id del transporte en el que quiere viajar

One

Two

Three

Four

Salir

Destino
- idlugar: long -nombre:String -direccion: String -telefono: String -codigoPost: int
+getIdlugar():long +setIdlugar(long Idlugar): void +getNombre():String +setNombre(String Nombre): void +getTelefono():String +setTelefono(String Telefono): void +getCodigoPost(): int +setCodigoPost(int CodigoPost): void

Viaje
- codigo: long -destino:String -salida: String -regreso: String -preciot: int
+getCodigo():long +setCodigo(long Codigo): void +getDestino():String +setDestino(String Destino): void +getSalida():String +setSalida(String Salida): void +getPrecio(): int +setPrecio(int Precio): void

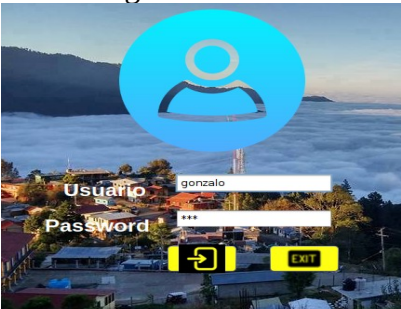
Transporte
- idTransporte: long -numAsiento: int -nombreTransport: String -precio: int
+getIdTransporte():long +setIdTransporte(long idTransporte): void +getNumAsiento():int +setNumAsiento(int NumAsiento): void +getNombreTransport():String +setNombreTransport(String NombreTransport): void +getPrecio(): int +setPrecio(int Precio): void

Cliente
- nombre: String -apellidos:String -telefono: String -viaje String
+getNombre(): String +setNombre(String Nombre): void +getApellidos():String +setApellidos(String Apellidos): void +getTelefono():String +setTelefono(String Telefono): void +getViaje(): String +setViaje(String Viej): void

Paradigma Orientado a objetos

El objetivo de mi proyecto de la agencia de viajes es conocer los lugares que contiene el distrito de Miahuatlán de Porfirio Díaz para que a las personas que son de fuera o lo que les gusta conocer de los lugares, además para las personas turísticas les sea mas fácil llegar a su destino.

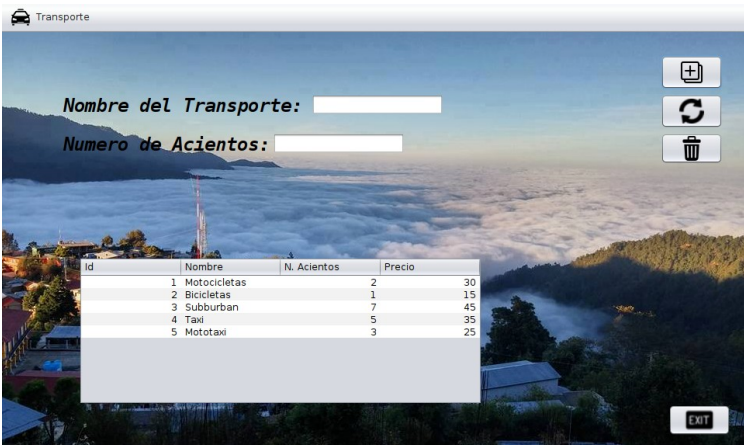
ALTA FIDELIDAD
Login de usuario



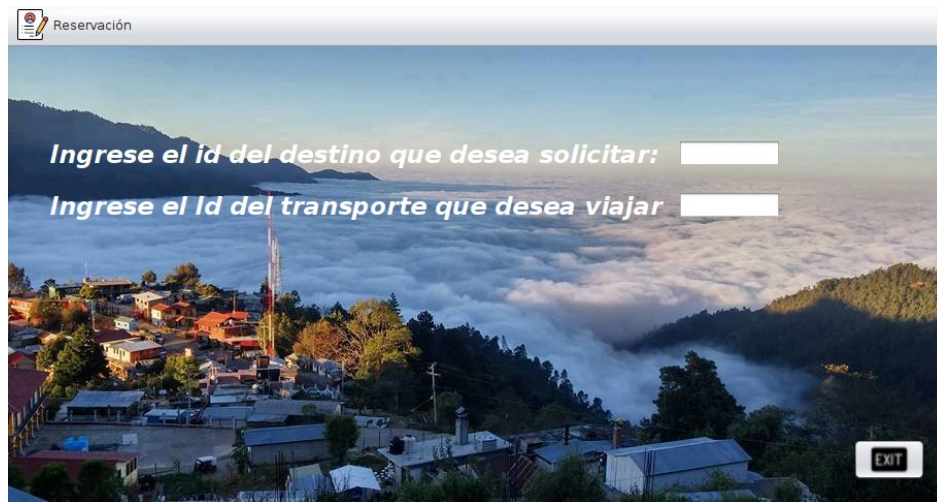
representa la ventana principal donde muestra la lista de los lugares recomendados de Miahuatlán



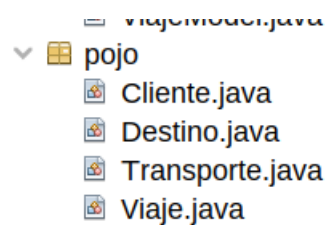
representa la ventana de la lista de los transporte por lo que aparece los nombre de los transportes que pueden viajar.



Paradigma Orientado a objetos



Programa de agencia de viajes



/*

@author: Gonzalo santiago Garcia

Fecha de creación: 1 junio del 2022

Fecha de actualización: 9 de junio del 2022

Descripción: En esta parte del programa contendrá el pojo Destino que es importante dentro del proyecto. por lo tanto, en esta clase se declararán todos los atributos de la clase Destino.

*/

```
package pojo;
```

```
public class Destino {
```

```
    private long idlugar;  
    private String nombre;  
    private String direccion;  
    private String telefono;  
    private int codigoPost;
```

```
    public Destino() {
```

```
    }
```

Paradigma Orientado a objetos

```
public Destino(long idlugar, String nombre, String direccion,
    String telefono, int codigoPost) {
    this.idlugar = idlugar;
    this.nombre = nombre;
    this.direccion = direccion;
    this.telefono = telefono;
    this.codigoPost = codigoPost;
}

public long getIdlugar() {
    return idlugar;
}

public void setIdlugar(long idlugar) {
    this.idlugar = idlugar;
}

public String getNombre() {
    return nombre;
}

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

public String getDireccion() {
    return direccion;
}

public void setDireccion(String direccion) {
    this.direccion = direccion;
}

public String getTelefono() {
    return telefono;
}

public void setTelefono(String telefono) {
    this.telefono = telefono;
}

public int getCodigoPost() {
    return codigoPost;
}
```

Paradigma Orientado a objetos

```
public void setCodigoPost(int codigoPost) {  
    this.codigoPost = codigoPost;  
}
```

```
}
```

Pojo Transporte

```
/*  
 * @autor: Gonzalo Santiago Garcia  
 * Fecha de creación: 01 de mayo del 2022  
 * fecha de actualización: 9 de mayo 2022  
 * Descripción: En la clase Transporte se declaran los atributos de esa clase  
   y además se declara sus métodos get y set de cada atributo.  
 */
```

```
package pojo;
```

```
public class Transporte {
```

```
    private long idTranspor;  
    private int numAsiento;  
    private String nombreTransport;  
    private int precio;
```

```
    public Transporte() {
```

```
    }
```

```
    public Transporte(long idTranspor, int numAsiento, String nombreTransport, int precio) {  
        this.idTranspor = idTranspor;  
        this.numAsiento = numAsiento;  
        this.nombreTransport = nombreTransport;  
        this.precio = precio;  
    }
```

```
    public long getIdTranspor() {  
        return idTranspor;  
    }
```

```
    public void setIdTranspor(long idTranspor) {  
        this.idTranspor = idTranspor;  
    }
```

```
    public int getNumAsiento() {  
        return numAsiento;
```

Paradigma Orientado a objetos

```
}

public void setNumAsiento(int numAsiento) {
    this.numAsiento = numAsiento;
}

public String getNombreTransport() {
    return nombreTransport;
}

public void setNombreTransport(String nombreTransport) {
    this.nombreTransport = nombreTransport;
}

public int getPrecio() {
    return precio;
}

public void setPrecio(int precio) {
    this.precio = precio;
}
}

pojo Viaje
/*
@author: Gonzalo santiago Garcia
Fecha de creación: 25 de mayo del 2022
Fecha de actualización: ----
Descripción: En esta parte del programa contenera el pojo Viaje
que es importante dentro del proyecto. por lo tanto, en está clase se declararán
todos los atributos de la clase Viaje.
*/
package pojo;

public class Viaje {

    private long codigo;
    private String destino;
    private String salida;
    private String regreso;
    private int precio;

    public Viaje() {

    }
}
```

Paradigma Orientado a objetos

```
public Viaje(long codigo, String destino, String salida, String regreso, int precio) {
    this.codigo = codigo;
    this.destino = destino;
    this.salida = salida;
    this.regreso = regreso;
    this.precio = precio;
}

public long getCodigo() {
    return codigo;
}

public void setCodigo(long codigo) {
    this.codigo = codigo;
}

public String getDestino() {
    return destino;
}

public void setDestino(String destino) {
    this.destino = destino;
}

public String getSalida() {
    return salida;
}

public void setSalida(String salida) {
    this.salida = salida;
}

public String getRegreso() {
    return regreso;
}

public void setRegreso(String regreso) {
    this.regreso = regreso;
}

public int getPrecio() {
    return precio;
}
```


Paradigma Orientado a objetos

```
    public void setPrecio(int precio) {  
        this.precio = precio;  
    }  
}
```

pojo Cliente

/*

autor: Gonzalo Santiago Garcia

fecha de creación: 1 de mayo del 2022

fecha de actualización: 9 de junio del 2022

descripción: declaración de los atributos con sus get y set de cada atributo
por lo que se crea los atributos de la clase Cliente

*/

package pojo;

```
public class Cliente {
```

```
    private String nombre;  
    private String apellidos;  
    private String telefono;  
    private String viaje;
```

```
    public Cliente() {
```

```
    }
```

```
    public Cliente(String nombre, String apellidos, String telefono, String viaje) {
```

```
        this.nombre = nombre;  
        this.apellidos = apellidos;  
        this.telefono = telefono;  
        this.viaje = viaje;
```

```
    }
```

```
    public String getNombre() {
```

```
        return nombre;
```

```
    }
```

```
    public void setNombre(String nombre) {
```

```
        this.nombre = nombre;
```

```
    }
```

```
    public String getApellidos() {
```

```
        return apellidos;
```

```
    }
```

Paradigma Orientado a objetos

```
public void setApellidos(String apellidos) {
    this.apellidos = apellidos;
}

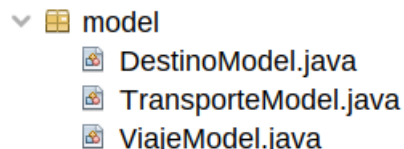
public String getTelefono() {
    return telefono;
}

public void setTelefono(String telefono) {
    this.telefono = telefono;
}

public String getViaje() {
    return viaje;
}

public void setViaje(String viaje) {
    this.viaje = viaje;
}
}
```

model



```
▼ [icon] model
  [icon] DestinoModel.java
  [icon] TransporteModel.java
  [icon] ViaieModel.java
```

```
package model;
```

```
import pojo.Destino;
import java.util.List;
import javax.swing.table.DefaultTableModel;
```

```
public class DestinoModel {

    public void crearDestino(List<Destino> lista, Destino destino) {
        lista.add(destino);
        // return lista;
    }

    // se crea el método obtenerUnDestino que obtiene un id de la
    //clase destino
    public Destino obtenerUnDestino(List<Destino> lista, int codigo) {
```

Paradigma Orientado a objetos

```
for (Destino destino : lista) {
    /*hace la comparación si es igual el id que se esta igualando
    entonces retorna un objeto destino si no lo encuentra retorna
    un null
    */

    if (destino.getCodigoPost() == codigo) {
        return destino;
    }
}
return null;
}

public void eliminarDestino(List<Destino> lista, String nombre) {
    for (int i = 0; i < lista.size(); i++) {
        if (lista.get(i).getNombre().compareTo(nombre) == 0) {
            lista.remove(i);
            break;
        }
    }
}

public void actualizarDestino(List<Destino> lista, Destino destino) {
    for (int i = 0; i < lista.size(); i++) {
        if (lista.get(i).getCodigoPost() == (destino.getCodigoPost())) {
            lista.set(i, destino);

            break;
        }
    }
}

// se crea el método mostrarDestino que muestra los atributos de la clase y los
// presenta en pantalla
public void mostrarDestino(List<Destino> lista, DefaultTableModel modelo) {
    modelo.setRowCount(0);
    for (int i = 0; i < lista.size(); i++) {
        Object[] fila = new Object[6];
        fila[0] = lista.get(i).getIdlugar();
        fila[1] = lista.get(i).getNombre();
        fila[2] = lista.get(i).getCodigoPost();
        fila[3] = lista.get(i).getDireccion();
        fila[4] = lista.get(i).getTelefono();
        modelo.addRow(fila);
    }
}
```

Paradigma Orientado a objetos

```
    }  
}  
}  
}  
  
                                modeloTransporte  
/*  
autor: Gonzalo Santiago Garcia  
fecha de creación: 1 de junio del 2022  
fecha de actualización: 9 de junio del 2022  
descripción: contiene las funciones como crear, obtener, eliminar, actualizar y  
mostrar  
*/  
package model;  
  
import java.util.List;  
import javax.swing.table.DefaultTableModel;  
import pojo.Transporte;  
  
public class TransporteModel {  
  
    public void crearTransporte(List<Transporte> lista, Transporte transporte) {  
        lista.add(transporte);  
    }  
  
    // creamos el metodo obtenerTransporte  
    public Transporte obtenerTransporte(List<Transporte> lista, int codigo) {  
        for (Transporte transporte : lista) {  
            /* hace la comparación de de ide trasnporte si es igual al id  
            entonces retorna el objeto transporte  
            si no encuentra el id retorna null  
            */  
            if (transporte.getIdTranspor() == codigo) {  
                return transporte;  
            }  
        }  
        return null;  
    }  
    //función eliminar  
  
    public void eliminarTransporte(List<Transporte> lista, String nom) {  
        for (int i = 0; i < lista.size(); i++) {  
            if (lista.get(i).getNombreTransport().compareTo(nom) == 0) {  
                lista.remove(i);  
            }  
        }  
    }  
}
```

Paradigma Orientado a objetos

```
        break;
    }
}

//función actualizar
public void actualizarTransporte(List<Transporte> lista, Transporte transport) {
    for (int i = 0; i < lista.size(); i++) {
        if (lista.get(i).getNombreTransport().equals(transport.getNombreTransport())) {
            lista.set(i, transport);
            break;
        }
    }
}

//función mostrar
public void mostrarTransporte(List<Transporte> lista, DefaultTableModel modelo) {
    modelo.setRowCount(0);
    for (int i = 0; i < lista.size(); i++) {
        Object[] fila = new Object[4];
        fila[0] = lista.get(i).getIdTranspor();
        fila[1] = lista.get(i).getNombreTransport();
        fila[2] = lista.get(i).getNumAsiento();

        modelo.addRow(fila);
    }
}
}
```

modeloViaje

package model;

```
import java.util.List;
import pojo.Viaje;
```

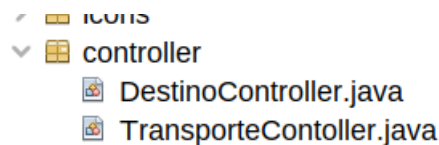
```
public class ViajeModel {

    // lista de viaje
    public List<Viaje> crearViaje(List<Viaje> lista1, Viaje viaje) {
        lista1.add(viaje);
        return lista1;
    }
}
```

Paradigma Orientado a objetos

```
public void mostrarViaje(List<Viaje> lista1) {
    System.out
        .println("-----");
    System.out
        .println("| Idcódigo |      Destino      | Hora de salida | Hora de regreso | Precio |");
    System.out
        .println("-----");
    for (Viaje viaje : lista1) {

        System.out.printf("| %10d | %24s | %14s | %16s | %10s |",
            viaje.getCodigo(), viaje.getDestino(),
            viaje.getSalida(), viaje.getRegreso(), viaje.getPrecio());
        System.out.println(
            "\n-----");
    }
}
}
```



```
/*
autor: Gonzalo Santiago Garcia
fecha de creación: 1 de junio del 2022
fecha de actualización: 9 de junio del 2022
descripción: en esta parte creo la función de crear, eliminar, mostrar
y actualizar
*/
package controller;

import java.util.List;
import javax.swing.table.DefaultTableModel;
import model.DestinoModel;
import pojo.Destino;

public class DestinoController {

    DestinoModel destinoM = new DestinoModel();

    public void crearDestino(List<Destino> lista, Destino destino) {
```

Paradigma Orientado a objetos

```
        destinoM.crearDestino(lista, destino);
    }

    public void eliminarDestino(List<Destino>lista, String nombre) {
        destinoM.eliminarDestino(lista, nombre);
    }

    public void mostrarDestino(List<Destino> lista, DefaultTableModel modelo) {
        destinoM.mostrarDestino(lista, modelo);
    }
    public void actualizarDestino(List<Destino> lista, Destino destino){
        destinoM.actualizarDestino(lista, destino);
    }
    public Destino obtenerUnDestino(List<Destino> lista, int id) {
        return destinoM.obtenerUnDestino(lista, id);
    }
}

/*
autor: Gonzalo Santiago Garcia
fecha de creación: 1 de junio del 2022
fecha de actualización: 9 de junio del 2022
descripción: en esta parte creo la función de crear, eliminar, mostrar
y actualizar
*/
package controller;

import java.util.List;
//import javax.swing.DefaultButtonModel;
import javax.swing.table.DefaultTableModel;
import model.TransporteModel;
import pojo.Transporte;

public class TransporteContoller {

    TransporteModel trasM = new TransporteModel();

    public void crearTransporte(List<Transporte> lista, Transporte transporte) {
        trasM.crearTransporte(lista, transporte);
    }

    //public void eliminarTransorte(List<Transporte> lista, String nombre{
    // trasM.mostrarTransporte(lista, nombre);
    //}

    public void mostrarTransporte(List<Transporte> lista, DefaultTableModel modelo) {
```

Paradigma Orientado a objetos

```
    trasM.mostrarTransporte(lista, modelo);  
}  
  
public void actualizarTransporte(List<Transporte> lista, Transporte transporte) {  
    trasM.actualizarTransporte(lista, transporte);  
}  
  
public Transporte obtenerTransporte(List<Transporte> lista, int id) {  
    return trasM.obtenerTransporte(lista, id);  
}  
}
```

