
Instituto Tecnológico CTC

**Analista Programador
Diseño y Desarrollo de Aplicaciones**

Obligatorio 2

Sistema de gestión de planes turísticos

**Guillermo Morales
Agustín Noy**

2022

Fecha: desde el 03 de noviembre al 30 de noviembre de 2022, a las 23:59.

Una empresa de turismo quiere manejar los planes para ofertar a sus clientes y luego poder asociar esos planes con los clientes que los adquieran. Hay clientes estándar y clientes premium. Los clientes premium son aquellos que ya han adquirido 3 planes de viajes o más. Para ellos, el costo del cuarto plan en adelante tiene un 20% de descuento.

EL SISTEMA DEBE PERMITIR:

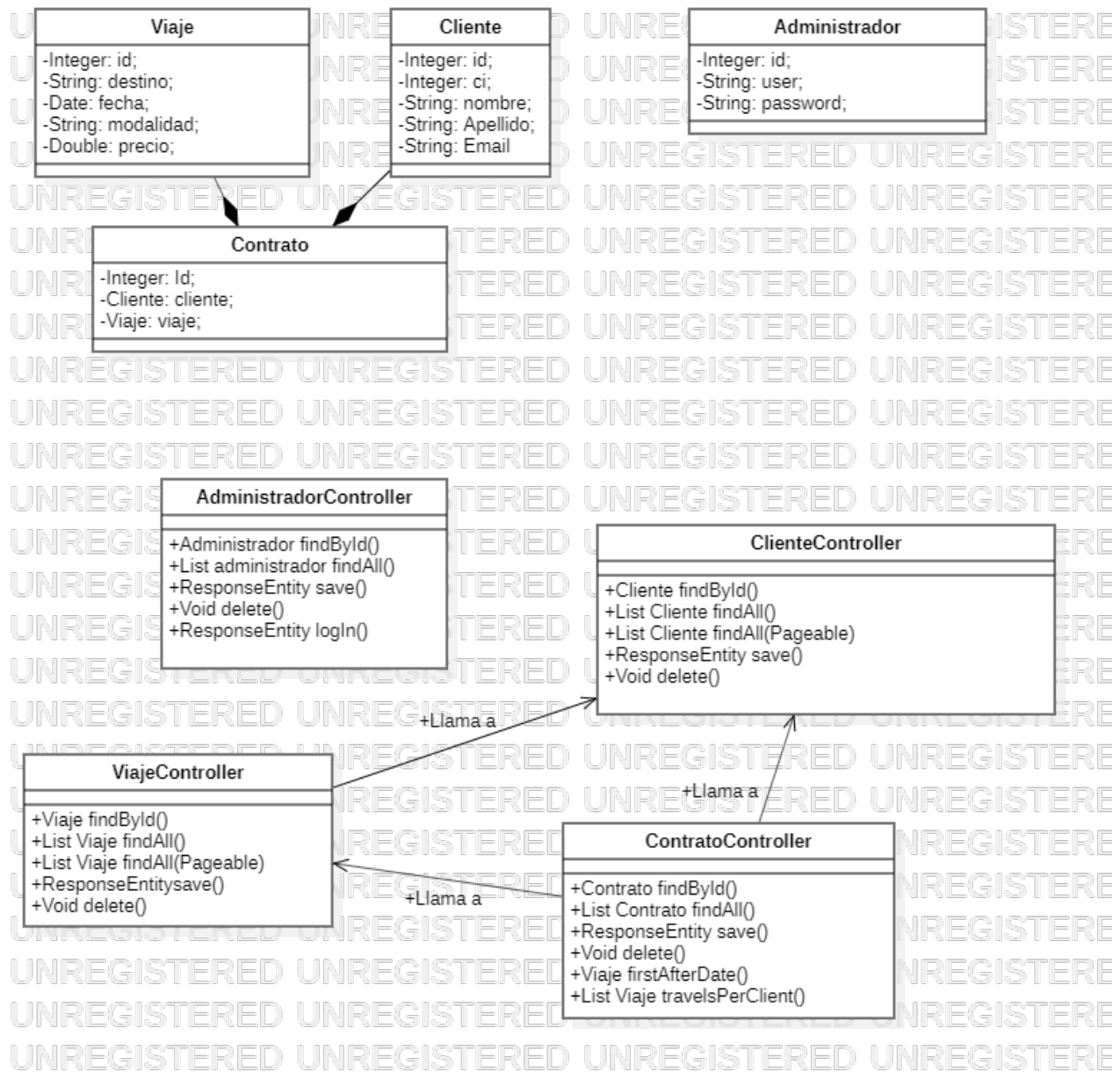
1. Mostrar un dashboard administrativo en ambiente Web o Mobile (sencillo) que permita trabajar con planes de viaje y clientes.
2. Dar de alta, eliminar y modificar planes de viaje. Cada uno de ellos tiene un solo destino (máx. 20 dígitos alfanuméricos), fecha, modalidad (solamente pueden ser aérea, marítima o terrestre), precio en USD, carrusel de fotos (opcional).
3. dar de alta, eliminar y modificar clientes.
Los clientes tienen CI (máx. 8 dígitos, mín. 7 dígitos, sin puntos ni guiones), nombre (máx. 30 dígitos alfanuméricos), apellido (máx. 30 dígitos alfanuméricos), email (máx. 30 dígitos alfanuméricos), planes comprados (si no tiene ninguno se deberá indicar con un mensaje que "no tiene planes comprados").
4. contratar o borrarse de un plan (o más) de viaje.
5. Listar los planes (viajes) de un cliente mostrando todos los atributos de cada viaje.
6. Listar el primer viaje que tendrá un cliente después de una fecha especificada.
7. Controlar que no se permita ingresar viajes (planes) con fechas anteriores a la fecha actual.
8. Controles de errores para ingreso de datos.

Análisis y solución

Para la resolución de los puntos planteados en la letra del problema el Back End se implementara una api utilizando Spring (JPA, Spring WEB, Spring Validation) los constraint seran definidos a nivel de entity y no se utilizaran funciones a nivel de mySQL sino que se implementaran utilizando JPA querys.

Para el Front End la solución sera una pagina con login y ventanas separadas para cada sección (Clientes, Viajes, Contratos) esta ultima conteniendo las dos consultas del primer viaje con una reserva a partir de una fecha y mostrar todos los viajes de un cliente cualquiera. El front end contendra Bootstrap4 para una respuesta de validacion en los campos de los formularios cambiando de color, barra de navegacion responsive y mensajes de alerta/confirmacion.

Diagrama de clases



Codigo de las clases

Administrador:

```
package com.example.ObligatorioSpring.entity;
```

```
import java.io.Serializable;
```

```
import jakarta.persistence.Column;
```

```
import jakarta.persistence.Entity;
```

```
import jakarta.persistence.GeneratedValue;
```

```
import jakarta.persistence.GenerationType;
```

```
import jakarta.persistence.Id;
```

```
import jakarta.persistence.Table;
```

```
import jakarta.validation.constraints.NotNull;
```

```
@Entity
```

```
@Table(name = "administradores")
```

```
public class Administrador implements Serializable{
```

```
    @Id
```

```
    @GeneratedValue(strategy = GenerationType.IDENTITY)
```

```
    private Integer id;
```

```
    @Column
```

```
    @NotNull
```

```
    private String user;
```

```
    @Column
```

```
    @NotNull
```

```
    private String password;
```

```
}
```

Cliente:

```
package com.example.ObligatorioSpring.entity;

import java.io.Serializable;

import org.springframework.beans.factory.annotation.Value;

import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import jakarta.persistence.Table;
import jakarta.validation.constraints.NotNull;
import jakarta.validation.constraints.Size;
```

```
@Entity
```

```
@Table(name = "clientes")
```

```
public class Cliente implements Serializable {
```

```
    @Id
```

```
    @GeneratedValue(strategy = GenerationType.IDENTITY)
```

```
    private Integer id;
```

```
    @Column
```

```
    @NotNull
```

```
    @Size(min = 7, max = 8)
```

```
    private Integer ci;
```

```
    @Column(length = 30)
```

```
    @NotNull
```

```
    private String nombre;
```

```
    @Column(length = 30)
```

```
    @NotNull
```

```
private String apellido;
```

```
@Column(length = 50)
```

```
@NotNull
```

```
private String email;
```

```
@Column
```

```
@NotNull
```

```
@Value("${some.key:false}")
```

```
private boolean premium;
```

```
public Cliente(Integer id, @NotNull @Size(min = 7, max = 8) Integer ci, @NotNull  
String nombre,
```

```
    @NotNull String apellido, @NotNull String email, @NotNull boolean premium) {
```

```
    this.id = id;
```

```
    this.ci = ci;
```

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

```
    this.apellido = apellido;
```

```
    this.email = email;
```

```
    this.premium = premium;
```

```
}
```

```
public Integer getId() {
```

```
    return id;
```

```
}
```

```
public void setId(Integer id) {
```

```
    this.id = id;
```

```
}
```

```
public Integer getCi() {
```

```
    return ci;
```

```
}
```

```
public void setCi(Integer ci) {  
    this.ci = ci;  
}
```

```
public String getNombre() {  
    return nombre;  
}
```

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

```
public String getApellido() {  
    return apellido;  
}
```

```
public void setApellido(String apellido) {  
    this.apellido = apellido;  
}
```

```
public String getEmail() {  
    return email;  
}
```

```
public void setEmail(String email) {  
    this.email = email;  
}
```

```
public Cliente() {  
}
```



```
}
```

Viaje:

```
package com.example.ObligatorioSpring.entity;
```

```
import java.io.Serializable;
```

```
import java.sql.Date;
```

```
import jakarta.persistence.Column;
```

```
import jakarta.persistence.Entity;
```

```
import jakarta.persistence.GeneratedValue;
```

```
import jakarta.persistence.GenerationType;
```

```
import jakarta.persistence.Id;
```

```
import jakarta.persistence.Table;
```

```
import jakarta.persistence.Temporal;
```

```
import jakarta.persistence.TemporalType;
```

```
import jakarta.validation.constraints.NotNull;
```

```
@Entity
```

```
@Table(name = "viajes")
```

```
public class Viaje implements Serializable {
```

```
    @Id
```

```
    @GeneratedValue(strategy = GenerationType.IDENTITY)
```

```
    private Integer id;
```

```
    @Column(length = 20)
```

```
    @NotNull
```

```
    private String destino;
```

```
    @Column
```

```
    @NotNull
```

```
    @Temporal(TemporalType.DATE)
```

```
private Date fecha;
```

```
@Column
```

```
@NotNull
```

```
private String modalidad;
```

```
@Column
```

```
@NotNull
```

```
private Double precio;
```

```
public Integer getId() {  
    return id;  
}
```

```
public void setId(Integer id) {  
    this.id = id;  
}
```

```
public String getDestino() {  
    return destino;  
}
```

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

```
public Date getFecha() {  
    return fecha;  
}
```

```
public void setFecha(Date fecha) {  
    this.fecha = fecha;  
}
```

```

    public String getModalidad() {
        return modalidad;
    }

    public void setModalidad(String modalidad) {
        this.modalidad = modalidad;
    }

    public Double getPrecio() {
        return precio;
    }

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

    public Viaje(Integer id, @NotNull String destino, @NotNull Date fecha, @NotNull
String modalidad,
        @NotNull Double precio) {
        this.id = id;
        this.destino = destino;
        this.fecha = fecha;
        this.modalidad = modalidad;
        this.precio = precio;
    }

    public Viaje() {
    }

}

```

Contrato:

```

package com.example.ObligatorioSpring.entity;

import java.io.Serializable;

import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import jakarta.persistence.OneToMany;
import jakarta.persistence.Table;

@Entity
@Table(name = "contratos")
public class Contrato implements Serializable{

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Integer id;

    @OneToMany
    private Cliente cliente;

    @OneToMany
    private Viaje viaje;

}

```