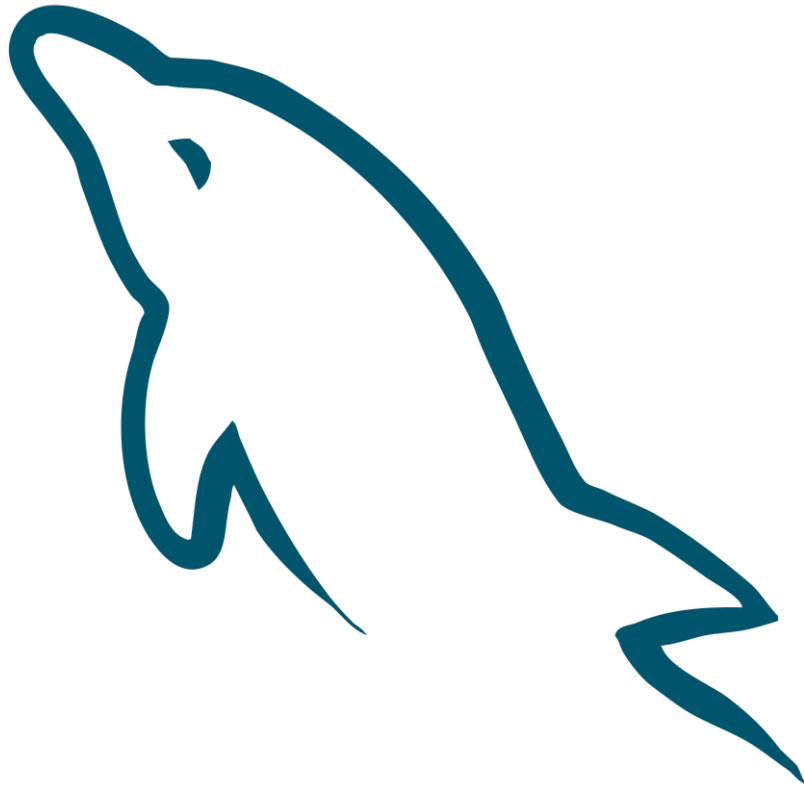


# Metodologías de programación II

Informe final



**Profesor:** Cappelletti Claudia

Comisión: 1

Fecha: 21-11-2023

Alumnos:

**Apellido:** Corsiglia

**Nombre:** Gonzalo Fabian

**Legajo:** 55315

**Apellido:** Duarte

**Nombre:** Rodrigo Ivan

**Legajo:** 64041

**Apellido:** Garcette

**Nombre:** Nicolas Nahuel

**Legajo:** 55396

# Índice

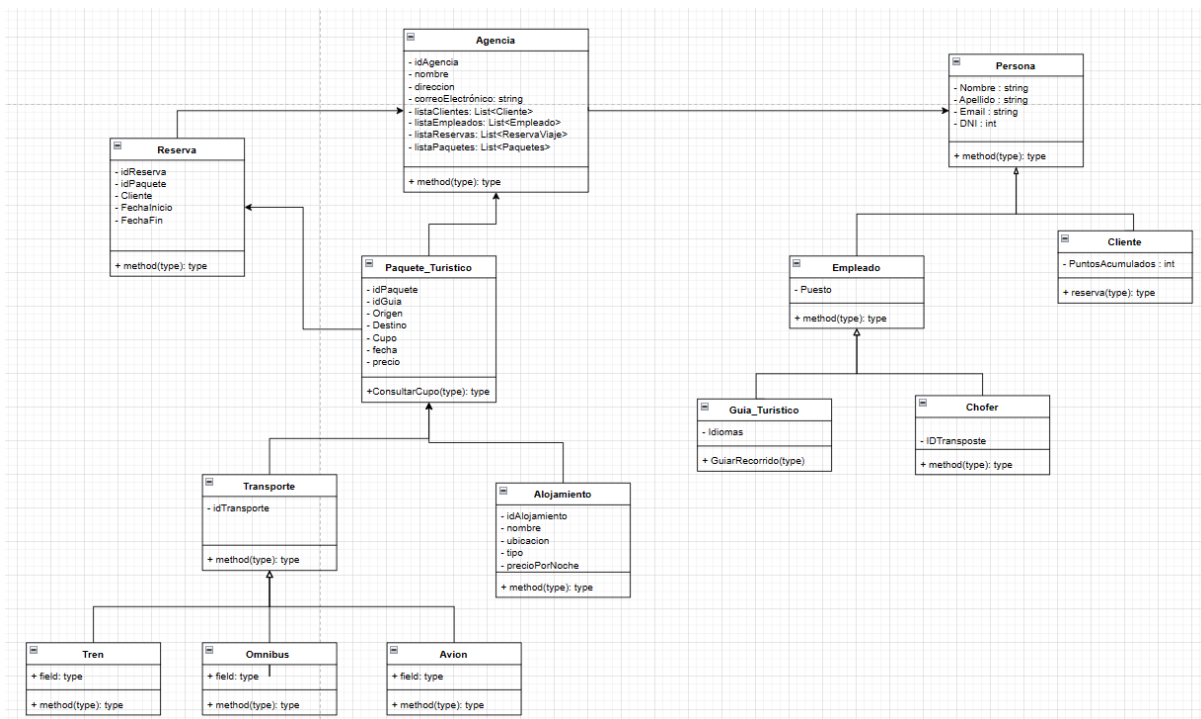
Descripción	2
Especificación e implementación de clases	2
Implementación de la aplicación	4
Descripción de la metodología de trabajo utilizada por los integrantes	5
Breve descripción de los Frameworks que se podrían utilizar para realizar un desarrollo orientado a objetos	5
Especificación de al menos dos casos de uso de Patrones en el contexto del problema planteado	7
Descripción de qué repositorios se podrían utilizar para el desarrollo de software colaborativo	8
Conclusión	8

## Descripción

En el marco de la presente actividad, se planteó el desafío de desarrollar un programa capaz de gestionar reservas de manera eficiente para una agencia de turismo. El objetivo primordial del sistema consistió en automatizar procesos clave, tales como la realización de reservas, la cancelación de las mismas, y la visualización de reservas asociadas a un cliente específico.

## Especificación e implementación de clases

En una primera instancia, se diseñó un diagrama de clases el cual consistía en 13 clases, pero por cuestiones de simplificación se acordó utilizar e implementar 5 clases, las cuales eran completamente necesarias en el flujo de la creación de una reserva, estas eran las clases de Agencia, Persona, Cliente, Reserva y Paquete.

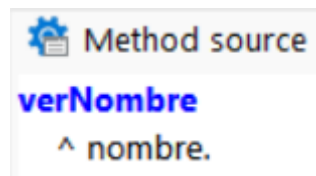


1. Diagrama de clases de la agencia completo.

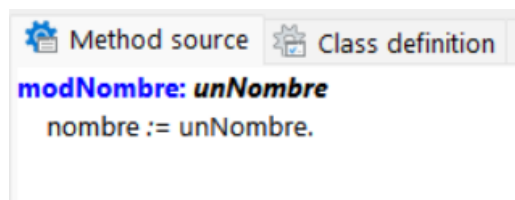


2. Clase Agencia en el class browser con sus variables y algunos de sus métodos de instancia.

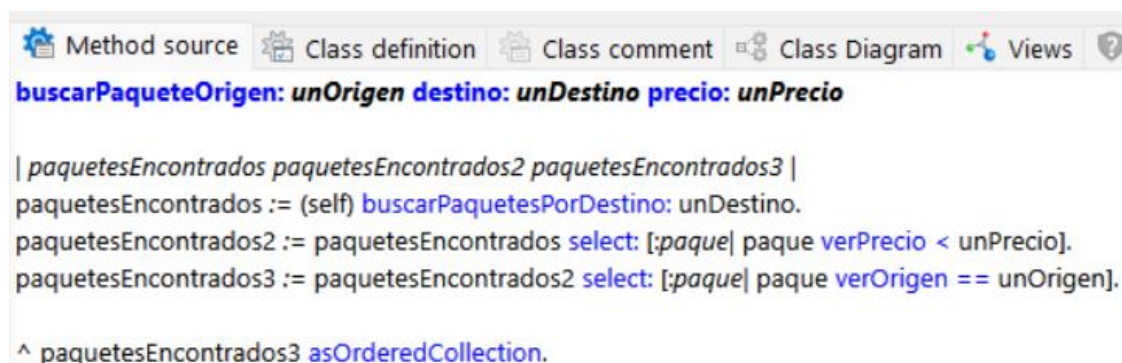
Dentro de las clases se implementaron muchos métodos distintos para lograr cumplir con las distintas soluciones que se probaron durante la realización del programa, métodos comunes tales como gets, set y métodos para la instanciación, por otro lado, también se realizaron métodos más específicos de cada clase relacionado con el manejo de colecciones.



3. Método get de la clase Agencia.



4. Método set de la clase Agencia.



5. Método de la clase Agencia que devuelve una lista con los paquetes que cumplan con los parámetros solicitados.

Corsiglia Gonzalo Fabian  
Duarte Rodrigo Ivan  
Garcette Nicolas Nahuel

### eliminarPaquetesAgotados

```
listaPaquetes := listaPaquetes reject: [:paquete | paquete verCupo <= 0].
```

6. Método de la clase Agencia devuelve una lista con los paquetes que tienen cupo.

### obtenerListaDeDestinos

```
| destinos |
```

```
destinos := listaPaquetes collect: [:paquete | paquete verDestino].
```

```
^ destinos asOrderedCollection.
```

7. Método de la clase Agencia que devuelve una lista con los destinos de los paquetes.

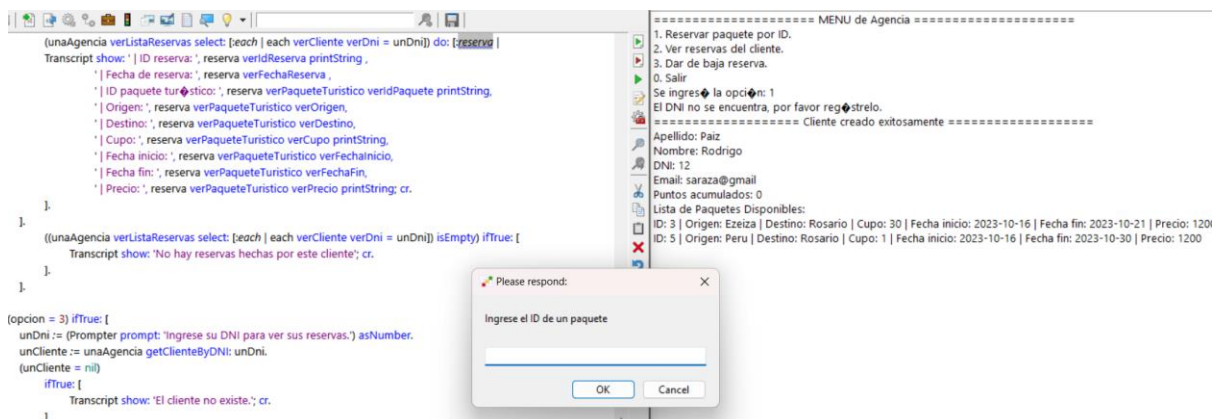
### getClientByDNI:unDni

```
^ listaClientes detect: [:each | each verDni = unDni] ifNone:[^nil].
```

8. Método de la clase Agencia que mediante un detect obtiene el cliente mediante el dni.

## Implementación de la aplicación

El código implementa las clases mencionadas anteriormente y proporciona una interfaz de usuario, a través del system transcript para realizar reservas, consultar disponibilidad y gestionar el estado de las reservas.



9. foto de la aplicación y del system al mismo tiempo durante la ejecución.

## Descripción de la metodología de trabajo utilizada por los integrantes

El equipo adoptó una metodología ágil, específicamente Scrum. Se organizaron sprints cortos. Se llevó a cabo el desarrollo de funcionalidades incrementales, y siempre se estuvo en contacto entre los desarrolladores a través de reuniones regulares de revisión y planificación. Además, se utilizó Trello como herramienta de gestión de proyectos, donde se crearon tableros para visualizar y organizar las tareas pendientes, en progreso y completadas, facilitando la colaboración y la transparencia del equipo.

## Breve descripción de los Frameworks que se podrían utilizar para realizar un desarrollo orientado a objetos

Existen varios frameworks en el ámbito del desarrollo orientado a objetos que facilitan la creación y mantenimiento de software. A continuación, se proporciona una breve descripción de algunos frameworks populares:

### **Spring Framework (Java):**

Spring es un framework de desarrollo de aplicaciones Java ampliamente utilizado. Ofrece soporte integral para la programación orientada a objetos y sigue el principio de inversión de control (IoC) y el contenedor de inversión de control (IoC). También proporciona módulos para el desarrollo de aplicaciones web, acceso a bases de datos, gestión de transacciones, y más.

### **Django (Python):**

Django es un framework de desarrollo web para Python que sigue el patrón de arquitectura Modelo-Vista-Controlador (MVC). Utiliza el enfoque de desarrollo rápido (RAD) y proporciona un ORM (Object-Relational Mapping) para facilitar la interacción con bases de datos. Django se centra en la simplicidad y la productividad.

## **Ruby on Rails (Ruby)**

Ruby on Rails (RoR) es un framework de desarrollo web para el lenguaje de programación Ruby. Utiliza el patrón de arquitectura MVC y sigue la convención sobre configuración, lo que significa que adopta ciertas convenciones preestablecidas para facilitar el desarrollo y reducir la necesidad de configuración manual.

## **.NET Framework (C#)**

.NET Framework es un conjunto de tecnologías de desarrollo de software de Microsoft. Utiliza el enfoque de desarrollo orientado a objetos a través de lenguajes como C#. Proporciona un entorno unificado para el desarrollo de aplicaciones de escritorio, web y móviles, y utiliza el patrón de arquitectura MVC con ASP.NET para aplicaciones web.

## **Laravel (PHP):**

Laravel es un framework de desarrollo web para PHP que sigue el patrón de arquitectura MVC. Ofrece una elegante sintaxis y herramientas poderosas para el desarrollo rápido de aplicaciones web. Laravel utiliza Eloquent ORM para interactuar con bases de datos de manera orientada a objetos.

## **Angular (JavaScript/TypeScript):**

Angular es un framework desarrollado por Google para construir aplicaciones web de una sola página (SPA). Utiliza el patrón de diseño MVVM (Model-View-ViewModel) y está basado en TypeScript, que es un superset de JavaScript que agrega tipado estático.

Estos frameworks proporcionan abstracciones y herramientas que simplifican el desarrollo de software orientado a objetos, aceleran el proceso de desarrollo y mejoran la mantenibilidad del código, por otro lado, cuenta con estructuras que facilitan el desarrollo orientado a objetos, mejoran la eficiencia y mantenibilidad del código, y se adaptan a diferentes lenguajes de programación.. La elección del framework dependerá de los requisitos específicos del proyecto y de las preferencias tecnológicas del equipo de desarrollo.



## Especificación de al menos dos casos de uso de Patrones en el contexto del problema planteado

### **Patrón Observer para Actualización de Disponibilidad de paquetes turísticos:**

Podríamos utilizar el patrón Observer para notificar a los clientes sobre cambios en la disponibilidad de paquetes turísticos. Cuando la disponibilidad de un paquete cambia o se agrega algún paquete, los clientes interesados en este tipo de paquete podrían recibir notificaciones automáticas.

#### **Implementación:**

Cada instancia de la clase “PaqueteTuristico” actúa como el sujeto observado. Las instancias de la clase “Cliente” que están interesadas en un tipo específico de paquete se registran como observadores. Cuando la disponibilidad de un componente en el paquete cambia, la notificación se envía a todos los observadores registrados, asegurando que los clientes estén informados de inmediato. Esta implementación garantiza que los clientes sean notificados directamente cuando haya cambios en la disponibilidad de los paquetes turísticos que les interesan.

### **Patron Decorator para Características Adicionales en Paquetes Turisticos:**

Podríamos utilizar el patrón Decorator para añadir características adicionales de manera dinámica a los paquetes turísticos. Los clientes podrán personalizar sus paquetes con opciones extra, como seguro de viaje, servicio de transporte desde/hacia el aeropuerto, etc.

#### **Implementación:**

La clase base “PaqueteTuristico” representa el paquete estándar. Se crean clases decoradoras concretas, como “DecoratorSeguroViaje”, “DecoratorTransporteAeropuerto”, cada una de las cuales extiende la funcionalidad del paquete base. Los clientes pueden agregar dinámicamente estas características decorativas a sus paquetes turísticos según sus preferencias.

Este patrón permite una extensión flexible de la funcionalidad de los paquetes turísticos sin modificar su estructura, lo que facilita la personalización para adaptarse a las necesidades específicas de cada cliente.



## Descripción de qué repositorios se podrían utilizar para el desarrollo de software colaborativo

Para el desarrollo de software colaborativo, es fundamental utilizar sistemas de control de versiones y alguna plataforma para alojar el repositorio que facilite la colaboración entre desarrolladores.

Uno de los controladores de versiones que se puede utilizar es **git**, el cual permite a los desarrolladores realizar un seguimiento de los cambios en el código fuente de manera eficiente.

### Plataformas para alojar repositorios:

- **GitHub** es una plataforma de desarrollo de software basada en Git, tiene la capacidad de alojar repositorios de código, gestionar problemas y realizar solicitudes de extracción, facilitando la revisión de código.
- **Azure DevOps** es un conjunto de servicios de desarrollo de extremo a extremo ofrecida por Microsoft. Entre sus servicios Azure Repos proporciona alojamiento de repositorios Git y TFVC. Permite la gestión eficiente de ramas, revisiones de código y seguimiento de cambios.
- **Bitbucket** es una plataforma de desarrollo de software basada en Git y Mercurial. Proporciona alojamiento de repositorios con características como bifurcaciones (forks), seguimiento de problemas, revisión de código y wiki integrada.

## Conclusión

En conclusión, en conjunto, la combinación de una metodología ágil, la implementación de clases y métodos específicos, el uso de patrones de diseño, frameworks adecuados y herramientas para el desarrollo colaborativo ha permitido crear una aplicación robusta y eficiente para la gestión de reservas en una agencia de turismo.