Taller de Programación

Tarea 1

Informe del Modelo de Diseño

Grupo 14

Integrantes

Nombre	CI
Manuel Rodriguez	5097757-4
Guido Dinello	5031022-5
Cristian González	4894107-6
Francisco Mauri	5140198-4
Belen Olivera	5598984-3

Docente

Andrea Delgado

Índice

1. Introducción

1.1. Propósito

El propósito de este documento es brindar una descripción general del Modelo de Diseño.

1.2. Alcance

El informe del Modelo de Diseño presenta una abstracción de la solución lógica al problema. Incluye las colaboraciones que realizan cada uno de los casos de uso del Modelo de Casos de Uso.

1.3. Estructura del Documento

El documento está dividido en tres secciones. La segunda sección presenta la organización lógica del sistema en paquetes de diseño. La tercera sección presenta las colaboraciones con la realización de los casos de uso incluidos en el Modelo de Casos de Uso. Por último, la cuarta sección presenta los criterios generales adoptados para el diseño de las colaboraciones.

2. Organización Lógica

A los efectos de organizar la capa lógica se definen paquetes de diseño. Contaremos en una primera instancia con un servidor central el cual contendrá los paquetes: Lógica, DataTypes y Excepciones tal como se muestra en la Figura 2.1.

Asimismo en el paquete correspondiente a Lógica tendremos las interfaces, controladores, manejadores y clases. Los manejadores serán implementados utilizando el patrón Singleton ya que serán los encargados de manejar las colecciones las cuales se quiere que tengan una visibilidad global y se desea que de ellos solo exista una única instancia. Para quebrar la dependencia de la capa visual y lógica se implementa el uso de interfaces.

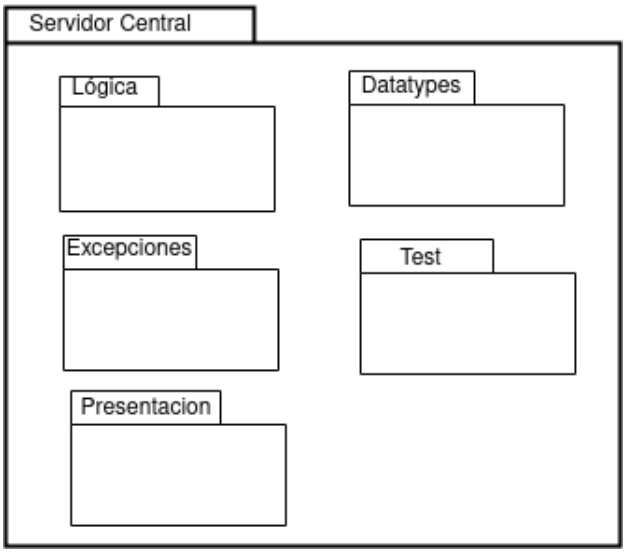

Figura 2.1: Servidor Central

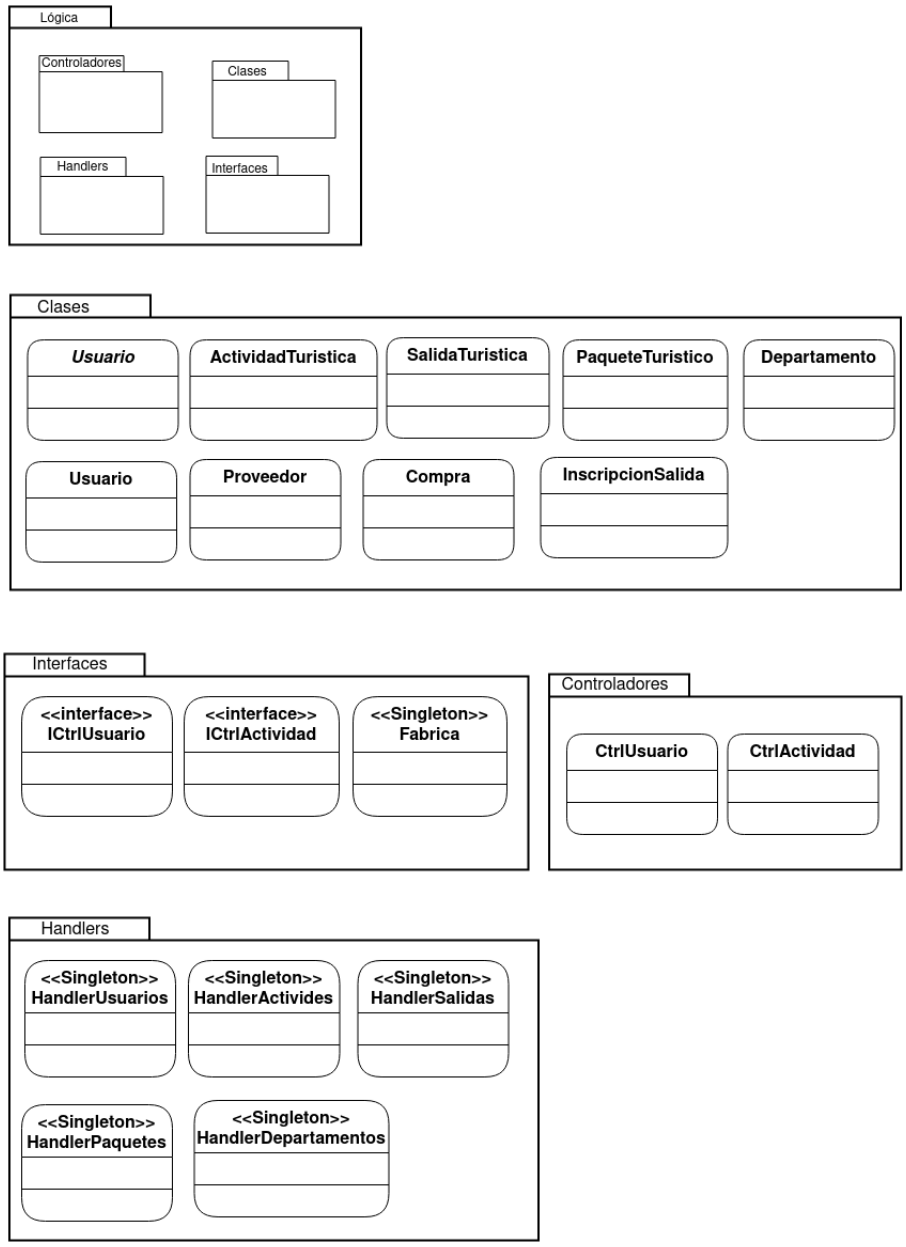

Figura 2.2: Lógica en detalle

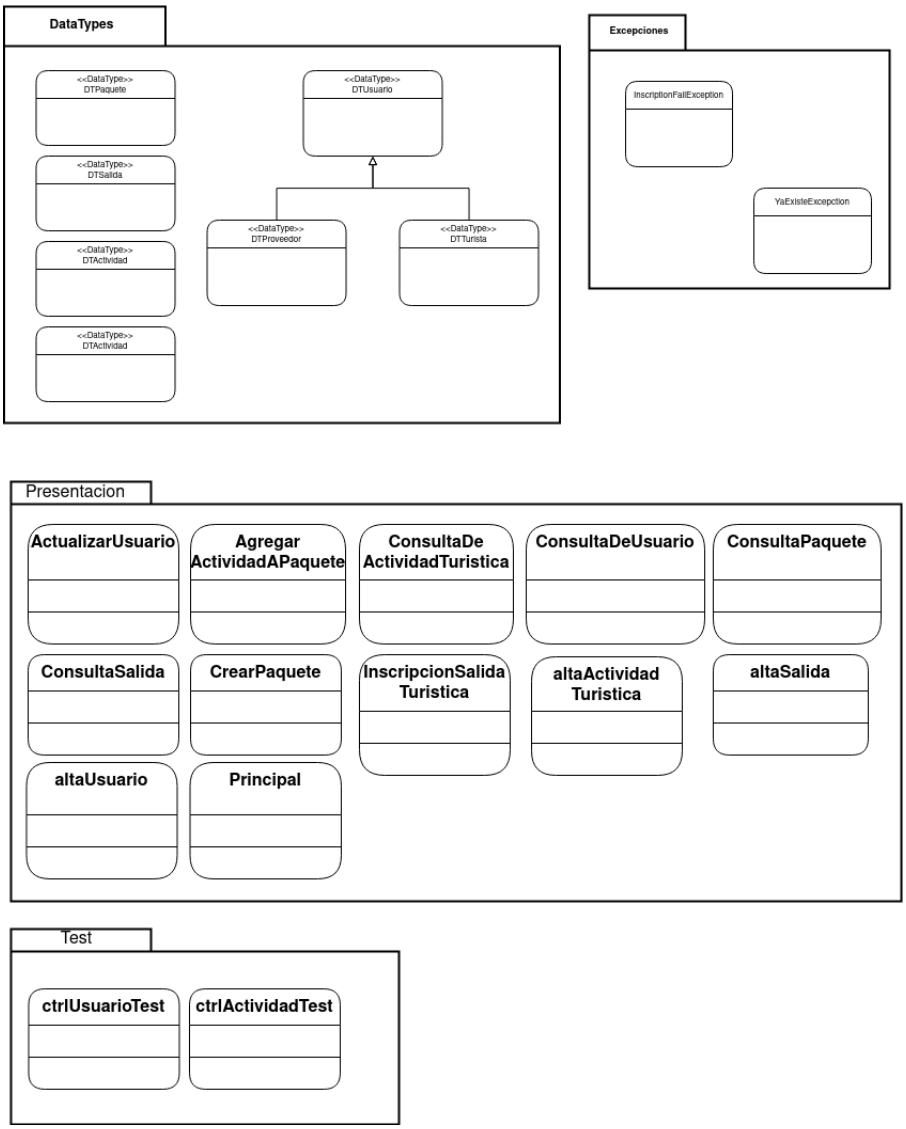

Figura 2.3: DataTypes, Exceptions, Presentacion y Test en detalle

3. Realización de Casos de Uso

Las realizaciones, están expresadas en términos de estructura e interacciones entre instancias que respeten dicha estructura. Concretamente, la parte estructural de una realización es un diagrama de clases conteniendo clases del modelo; la parte dinámica es un conjunto de diagramas de interacción que ilustran el flujo de mensajes entre instancias de las clases de la parte estructural correspondiente.

Esta sección está dividida por colaboración. Una colaboración es la realización de uno o más casos de uso. Para cada colaboración habrá una sección que muestre su diseño. El nombre de una colaboración corresponde al nombre del caso de uso que realiza, o el nombre del "área temática" determinada por el conjunto de casos de uso que realiza. La estructura para presentar la información será la siguiente:

3.1. Colaboraciones

En esta sección se presentan las colaboraciones que realizan los casos de usos involucrados.

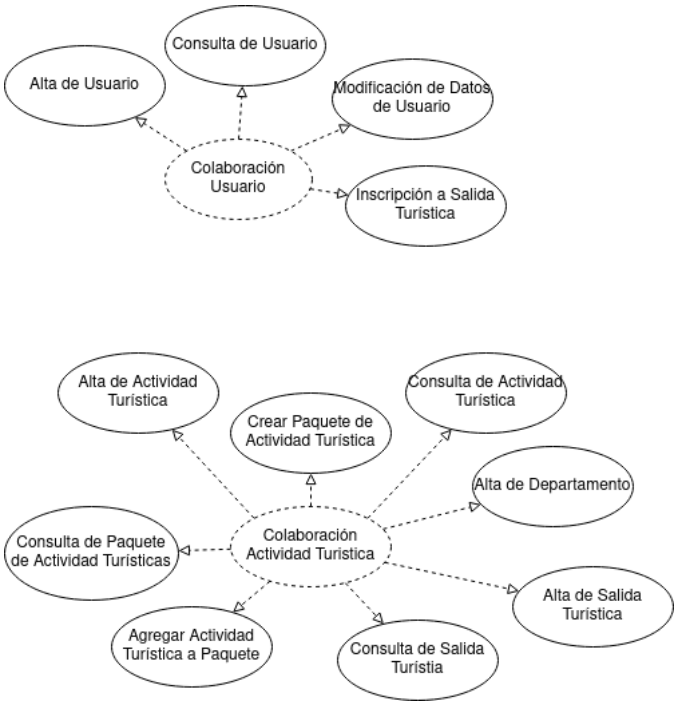

Figura 3.1: Colaboración

3.2. Estructura

En esta sección se presenta el diagrama de clases.

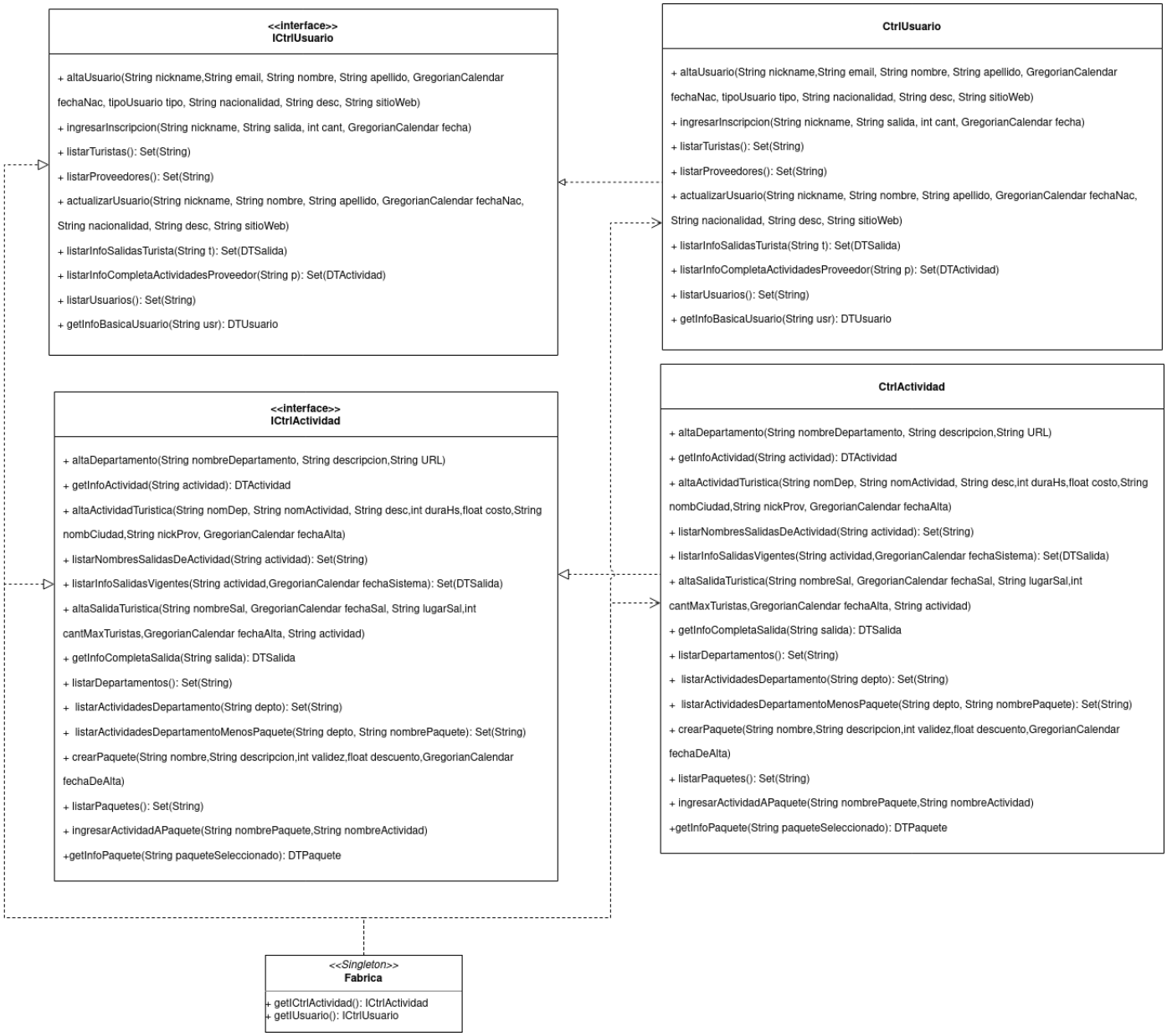

Figura 3.2: Diagrama de Clases de Diseño (Interfaces y controladores)

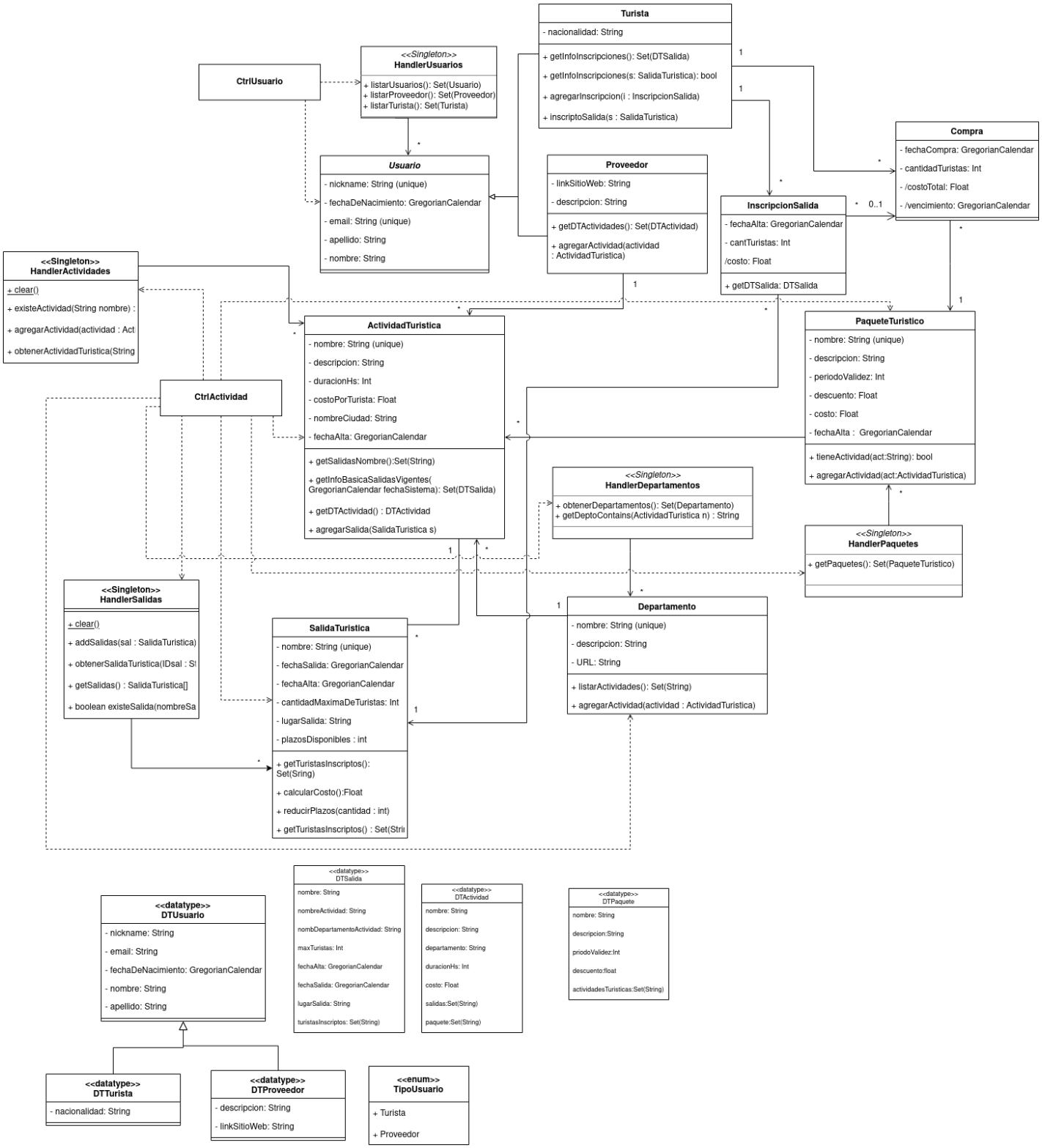

Figura 3.3: Diagrama de Clases de Diseño (Controladores simplificados y las todas las clases)

3.2.1. Design Patterns

En esta sección se indican todos los Design Patterns involucrados en el diseño de la colaboración de esta sección. Los patrones utilizados y que se explicarán a continuación son: Singleton y Factory.

3.2.1.1 Singleton

Utilizamos dicho patrón de diseño para cada uno de los manejadores, es decir, cada una de las clases

- HandlerUsuario.
- HandlerActividades.
- HandlerSalidas
- HandlerDepartamentos
- HandlerPaquetes

Todas ellas siguen dicho patrón como fue brevemente explicado en la sección 2.

3.2.1.2 Factory

Utilizamos dicho patrón de diseño para separar el código de construcción de producto del código que hace uso del producto.

Las clases involucradas son:

- Fabrica, que cumple el rol de Fabrica.
- Interfaces (ejemplo IUsuario), que cumplen el rol de IProveedor.
- Controladores (ejemplo CtrlUsuario), que cumplen el rol de IProveedor Concreto.

4. Criterios Generales

La estrategia utilizada fue la de asignar la responsabilidad de recibir las operaciones del sistema a los controladores conceptualmente asociados, a modo de ejemplo, la función de alta de usuario la va a recibir el controlador usuario y este va a ser el encargado de coordinar y delegar las operaciones que amerite el caso. Por otro lado, también apuntamos a reducir la carga de los handlers a lo mínimo posible puesto que estos son singleton y un uso intensivo de los mismos en un entorno multiproceso puede causar cuellos de botella e introducir problemas de sincronización. Es por esto, que no realizan mas operaciones que las de agregar, remover, encontrar y devolver datos, en ningún momento estos realizan un filtrado y/o preprocesamiento de los datos a devolver ya que esto será responsabilidad del orquestador del caso de uso, es decir, el controlador.