

Trabajo final - Parte 2

integrantes:

- Sandoval Agustin Pedro
- Gonzalez Facundo Nehuen
- Badaracco Iara Ailen
- Iavicoli Nicolas

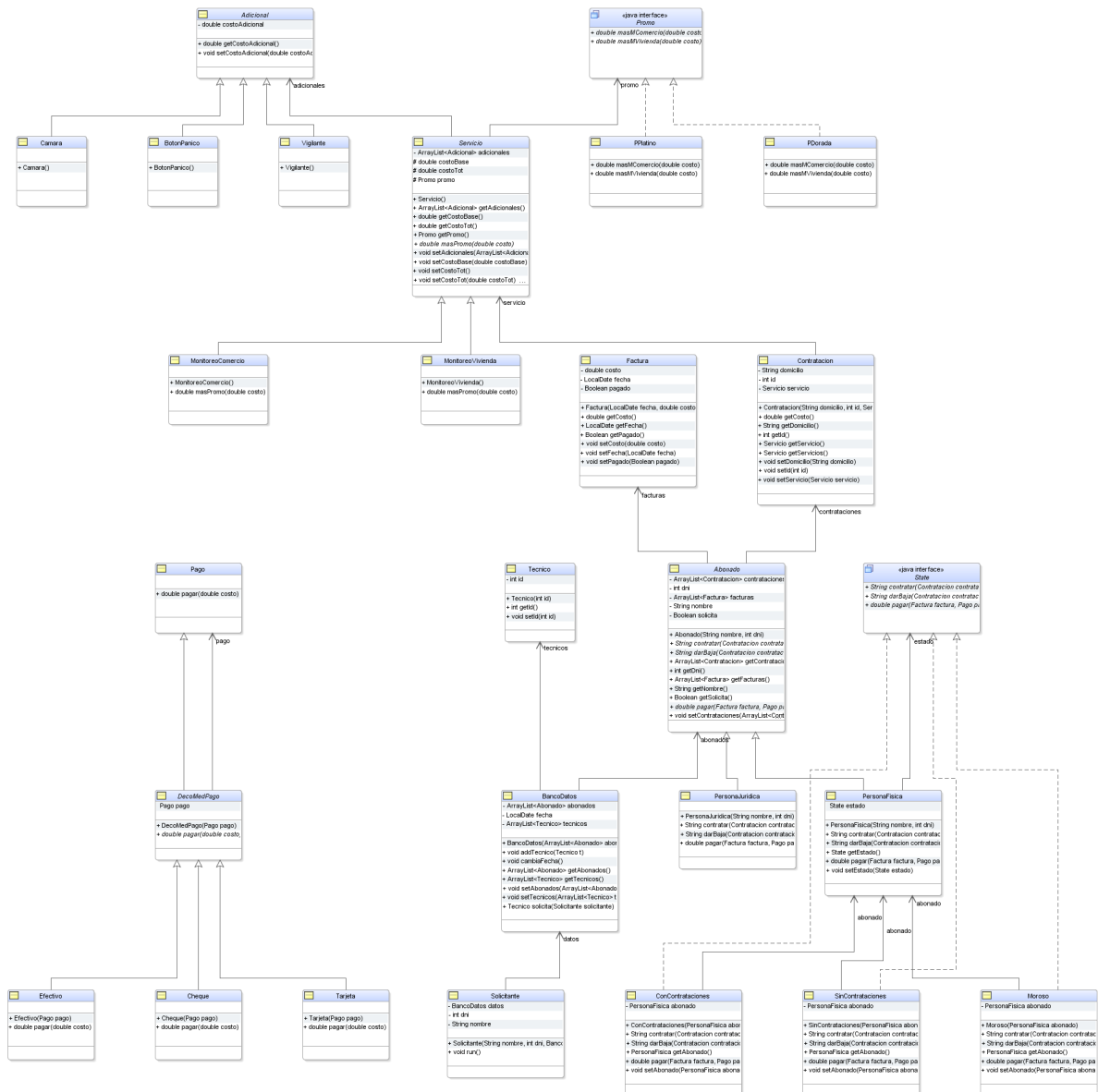
Resumen:

El presente informe detalla la segunda parte del desarrollo de un sistema de contratación de servicios de seguridad. Esta segunda parte tiene como objetivo principal incorporar una interfaz gráfica (unificada o varias ventanas) para poder gestionar el sistema, así como la incorporación de nuevos patrones de diseño a la estructura del programa. El sistema abarca tanto la posibilidad de solicitar la visita de un técnico, viéndose reflejada en un área de texto de la interfaz, hasta poder acceder al histórico de facturas de cada abonado. Para lograrlo, se han implementado los patrones Observer/Observable, State y MVC.

Cambios y correcciones:

Con respecto a la primera parte, se cambió y corrigió la relación de clases entre las entidades para poder cumplir con los requisitos del sistema, ahora las promociones implementan una interfaz para que el polimorfismo decida los valores de sus métodos. ahora abonado implementa dos ArrayList de objetos de tipo contratación y factura, se aprovechan mejor las herencias para crear estructuras mas claras y prolijas. Con ello la lógica de negocio se prepara para interactuar con las nuevas funcionalidades lógicas y gráficas.

UML:



Implementación de las nuevas funcionalidades:

Para poder implementar una función donde el abonado cambia de estado entre “con contratación”, “sin contratación” y “moroso”, se implementó el patrón state para poder cambiar los estados del abonado en tiempo de ejecución, sin necesidad de cambios manuales o intervención del programador.

Para poder gestionar las solicitudes a técnicos, se crean objetos de tipo thread y objetos tipo solicitante (que implementa la interfaz runnable) en la capa de control, el objeto thread

ejecuta el método `star()` usando al objeto solicitante como parametro, asi ejecutando las solicitudes a los técnicos de forma concurrente.

La interfaz gráfica fue implementada como una serie de ventanas que implementan `jframe` en lugar de una sola unificada para tener las funcionalidades más claras.

Estructura del programa:

En este caso se implementó el patrón MVC (modelo - vista - controlador) para definir la estructura del sistema, de esta forma toda la lógica de negocio y las estructuras de datos serán agrupadas en la capa de modelo, en la vista se creó la clase “ventana” que implementa la interfaz “`jframe`” para poder crear la interfaz gráfica del sistema., y la capa controlador que toma las consultas de la vista al modelo.

Conclusión:

En esta segunda parte se añadieron las funciones y patrones de diseño requeridos siendo necesario modificar la estructura del sistema para su correcta organización y funcionamiento, destacando el desarrollo de ventanas gráficas para la interacción del usuario con el sistema, la persistencia en la finalización de la jornada, y uso de concurrencia para implementar el sistema de altas a técnicos..