



Facultad de Ingeniería

Universidad Nacional de Mar del Plata

## **Sistemas de contratación y servicios de seguridad**

Cátedra: Programación III

Grupo 3

Estudiantes:

- Arias Iñaki
- Arias Pedro
- Delgado Facundo
- Rasso Micaela

URL: <https://github.com/facundodelga/SistemasDeSeguridad>

## **Análisis de las soluciones elegidas**

### **Correcciones de la primera parte**

En la última entrega, se logró una correcta aplicación del patrón Decorator al eliminar la clase Pago y crear las clases FacturaFisica y FacturaJuridica. Esta implementación permitió agregar funcionalidades adicionales a las facturas de forma flexible y modular.

Además, como se mencionó en el informe de la entrega anterior, se implementó exitosamente el patrón Factory para la creación de objetos. Esta elección de diseño nos ha permitido encapsular la lógica de creación de instancias en la clase Sistema, delegando responsabilidades específicas a cada clase y mejorando la organización del código.

Estas decisiones han contribuido a mejorar la estructura del sistema y nos han acercado a cumplir con los requerimientos de la consigna.

### **Persistencia**

Para la aplicación de persistencia se optó por utilizar archivos binarios, ya que nos resultó un método más sencillo y eficiente para almacenar y recuperar los datos. Al utilizar archivos binarios, podemos serializar los objetos y guardarlos en disco de forma compacta y estructurada.

Además, al trabajar con archivos binarios, se reduce la posibilidad de alteraciones accidentales o no autorizadas de los datos, ya que no es legible directamente por humanos. Esto brinda un nivel adicional de seguridad para la persistencia de la información.

### **Implementación de las vistas**

Para trabajar con las vistas se utilizó una interfaz llamada IVista que contienen los métodos: setActionListener y addActionListener los cuales se utilizan para asignar el controlador a la vista y para asignar el controlador a los distintos componentes que contiene cada vista respectivamente.

- AgregaDomicilio sirve para gestionar los domicilios que tiene cada persona.
- AgregaPersona lo que hace es crear un nuevo cliente tomando como datos su nombre completo, DNI y si es Persona Física o Jurídica.
- VistaContrataciones nos permite poder gestionar las contrataciones realizadas por la persona seleccionada
- VistaFacturaPersonas muestra el detalle de la facturas de la persona y nos permite pagar mediante los distintos medios de pago
- VistaNuevaContratacion crea nuevas contrataciones en un domicilio pudiendo asignar los servicios adicionales
- VistaSistemaDeSeguridad es la vista principal la cual contiene una lista de personas, un seleccionador de acciones la cual dispara las distintas vistas ya mencionadas, el apartado de Tecnicos el cual muestra la simulación del pedido de servicios técnicos y la pestaña que muestra las facturas históricas de los abonados.

## **Servicio Técnico**

Dentro del paquete simulación se encuentran las clases que se van a dedicar a controlar la concurrencia. las clases `ClienteThread` y `Tecnico` van a ser los hilos y la clase `ServicioTecnico` es el recurso compartido el cual también es de tipo `Observable`, a la vez esta clase se va a dedicar a notificar al Observador (`VistaSistemaDeSeguridad`) si se ha solicitado un servicio técnico o si este ya fue atendido por alguno de los `Tecnicos`.

## **Estados**

Se implementó el patrón state en la Persona Física como se pidió en la consigna. Se resolvió creando la interfaz `IEstado` e implementando los cambios en las clases `ConContratacionEstado`, `MorosoEstado` y `SinContratacionEstado`.

## **Problemas**

### **Implementación de Hilos**

Al realizar la implementación nos encontramos con la disyuntiva de si el controlador iba a contener los hilos o si los iba a almacenar la clase sistema y quien sería el observador. Al final se solucionó almacenando en sistema ya que contiene a la mayoría de objetos y la clase con la que se comunicaría el controlador para realizar las operaciones necesarias además que va a ser la que va a persistir los datos de la aplicación. El observador terminó siendo la Vista principal ya que ésta es el objeto que contenía el log donde se iba a mostrar la comunicación de los hilos con el recurso compartido (`ServicioTecnico`).