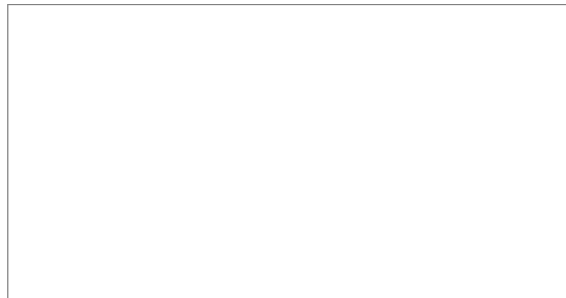


# **Facultad de Ciencia y Tecnología**

## **Ingeniería del Software II**



**Trabajo Práctico Anual**

**Año 2020 – Tercera Parte**



## Índice de contenido

Introducción.....	3
Estados de los Usuarios en la Aplicación.....	5
Flujo de Operaciones de la plataforma.....	6
Actividades a realizar.....	7



## Introducción

En ésta tercer entrega aplicaremos patrones de diseño para resolver problemáticas puntuales con las que nuestra aplicación se encuentra, logrando de ésta manera diseñar soluciones escalables y adaptables basadas en buenas prácticas conocidas.

A la hora de realizar los diagramas debemos tener en cuenta lo siguiente:

- ***Análisis y Justificación de cada Patrón:*** deberemos para cada caso indicar el patrón que vamos a utilizar para resolver cada escenario planteado y justificar su elección.
- ***Aplicación del Patrón:*** deberemos modificar el diagrama de clases para implementar para cada caso los patrones que indiquemos, cabe aclarar que debe verse claramente la implementación del patrón y que las mismas reglas que aplican hasta ahora se siguen manteniendo en la actualización del diagrama.
- ***Correcciones de entregas anteriores:*** en ésta entrega deberemos contar con las correcciones realizadas en la primer y segunda entrega.

La fecha límite de ésta tercer entrega es el día ***Jueves 29 de Octubre de 2020.***



## **Creando nuevos tipos de propiedades**

Dado el crecimiento de la aplicación, se hace necesario el poder agregar nuevos tipos de propiedades (como duplex, quintas y demás) a futuro sin tener que modificar mayormente la aplicación.



## **Creando nuevos tipos de propiedades**

Dado el crecimiento de la aplicación, se hace necesario el poder agregar nuevos tipos de propiedades (como duplex, quintas y demás) a futuro sin tener que modificar mayormente la aplicación.

## **Implementando auditoría y avisos en la aplica**

Se desea a continuación implementar un mecanismo de auditoría, de forma tal de lograr implementar acciones para los avisos, hoy inicialmente se desea que cada vez que un aviso se crea, se notifique por mail al administrador y se persista dicho suceso en una tabla de eventos del sistema. Cabe aclarar que también se desea agregar éste mismo mecanismo para cuando se dan los sucesos en el sistema:

- Cada vez que se registra un usuario.
- Cada vez que se realiza un pago de una operación.

## **Gestionando Performance y unificando lógica**

Se tiene la necesidad de crear una clase para unificar toda la lógica de creación de avisos (operaciones para dar de alta un aviso, poder modificarlo, darlo de baja o listar los avisos), debido a que es una clase que se va a utilizar mucho en la aplicación se desea crear solamente una instancia de dicha clase.



## Actividades a realizar

En ésta segunda entrega se solicita:

1. Especificar para cada uno de los puntos el tipo de patrón que se utilizaría y porqué.
2. Actualizar el diagrama de clases para implementar claramente cada uno de los patrones.