

Examen 1 Diseño de software

Wilson Lopez Rubi

Octubre 2019

1 Introducción

El presente proyecto corresponde al examen I , del curso de diseño de software , impartido en el segundo semestre del 2019 , en el Instituto Tecnológico de Costa Rica. Tiene como objetivo que los estudiantes apliquen el conocimiento en el diseño y desarrollo de software utilizando patrones de diseño y buenas practicas de desarrollo y elaboración de interfaces gráficas que cumplan con los principios S.O.L.I.D , así como los estándares de accesibilidad.

2 Diagrama

A continuación se muestra el diagrama de clases que se plantea como solución al problema.

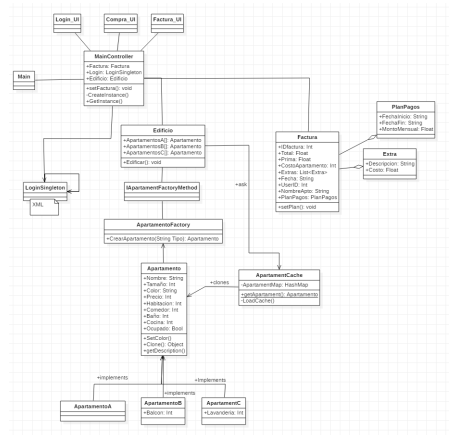


Figure 1: Diagrama de clases

3 Patrones

Para darle solución al problema se tomo la decisión de utilizar tres patrones de diseño , los tres son patrones de diseño de tipo creacional y son **Singleton** , **Factory** y **Prototype** , a continuación se va a dar explicara de manera detallada cada patrón , su uso y justifiacion.

3.1 Singleton

El patrón singleton se emplea en dos ocasiones , la primera para la creación del login , los usuarios registrados se guardan en un archivo de XML , el cual cumple la función de una base de datos de usuarios , la idea de utilizar el singleton es que un usuario solo pueda iniciar una vez. Además el patrón singleton también se utilizar en el controlador del programa , esto con la idea de que el controlador sea único y no se pueda instanciar mas de un controlador haciendo que el programa se replique o cree mas instancias de las necesarias de todas las clases requeridas para el funcionamiento. A continuación se muestra el uso del patrón en el diagrama de clases.

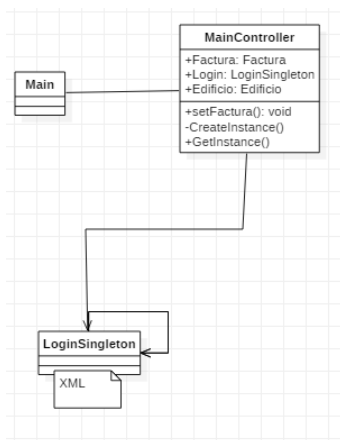


Figure 2: Patrón Singleton

3.2 Factory

El patrón factory se emplea en la creación de los distintos tipos de apartamentos (A,B,C) al ser objetos iguales pero con características distintas (Habitaciones) se considera que el patrón que encaja mas en la construcción de este objeto es el Factory. Una vez que se crea un edificio este activa la fabrica para crear los tres diferentes tipos de apartamentos. A continuacion se muestra su diagrama.

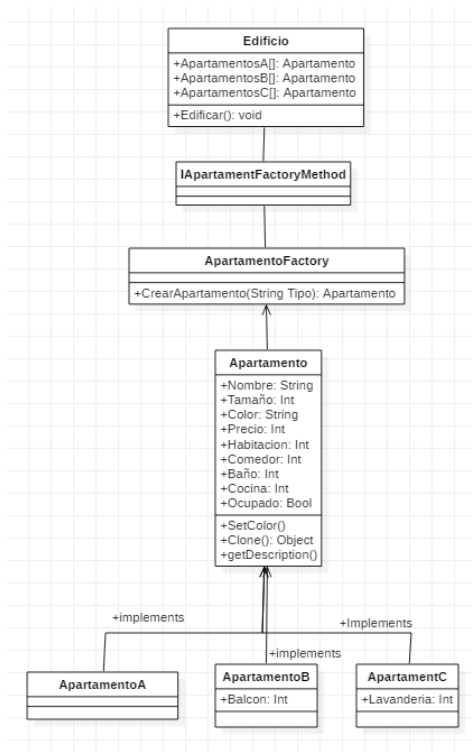


Figure 3: Patrón Factory

3.3 Prototype

Como cada edificio tiene diez apartamentos iguales de cada tipo , se toma la decisión de utilizar el patrón Prototype , para clonar los distintos tipos de apartamentos en lugar de hacer que la fabrica construya los diez de cada tipo. Esto para utilizar menos recursos del sistema en la creación de objetos, además permite escalabilidad pues igual se clonan 30 apartamentos que 300. A continuación se muestra el diagrama del patrón prototype.

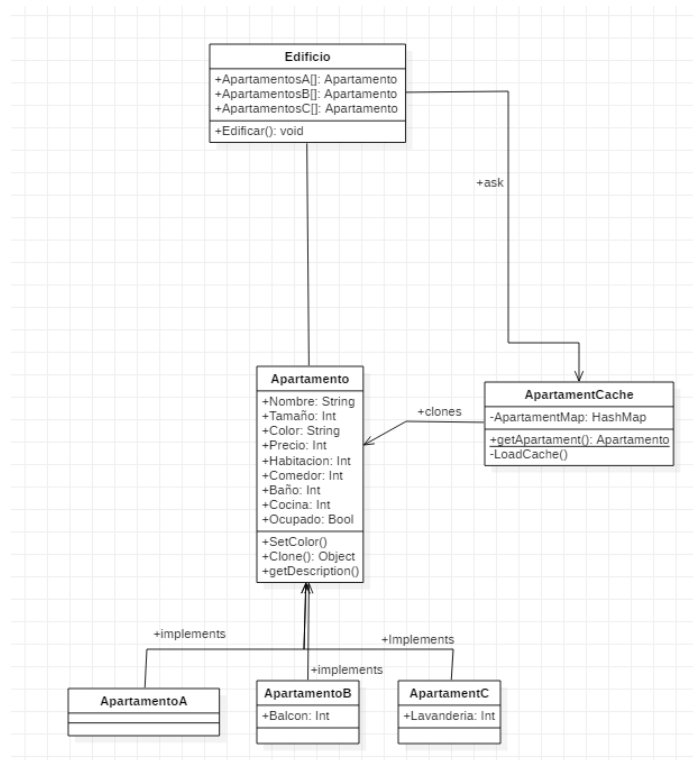


Figure 4: Patrón Prototype

4 Justificación S.O.L.I.D

4.1 single responsability

Todas las clases a excepción del controlador, cumplen única función.

4.2 Open close

Todo el proyecto esta abierto a mejoras y a ampliar la cantida de objetos.

4.3 Liscov sustitucion

Las clases dentro del proyecto tiene bajo acoplamiento por lo tanto cualquier modulo puede ser remplazado fácilmente , ademas se implenta el modelo vista controlador , para generar mejor independencia entre las clases.

4.4 Interface segregation

Al usar patrones como el factory y el tener objetos similares con distintas características el uso de las interfaces son requeridas y son implementadas para facilitar el manejo de los objetos.

4.5 Dependenci inversion

Cada modulo del proyecto esta separado y es independiente , por lo tanto cada grupo de clases es independiente lo que permite hacer cambios en el futuro.

5 Justificación de accesibilidad

La interfaz de usuario se piensa accesible por ese utiliza una paleta de colores clara donde predominan los blancos , además se utiliza un tipo de letra estándar y de buen tamaño para que sea entendible y visible. Las ventanas no se cargan de información innecesaria solamente tienen lo necesario para cada caso de uso únicamente.

6 Conclusión

El uso de los patrones de diseño puede generar que un proyecto sea mas claro , mas escalable y disminuir la cohesión entre las clases , sin embargo el uso mas patrones de la cuenta puede volver un proyecto mas bien mas complejo.

El uso de los patrones es un proceso repetitivo y simple (seguir una receta) , pero es muy importante conocer los casos en que se pueden aplicar.

Los patrones de diseño no son una verdad absoluta , a como un problema se puede solucionar con un patrón , se puede solucionar con otro distinto y no necesariamente significa que la solución esta incorrecta.

Los patrones se pueden combinar , el uso de un patrón de diseño no implica que no se pueda utilizar otro para facilitar llegar a un objetivo.