



UNIVERSIDAD
DE GRANADA

Desarrollo de Software

Práctica 1.1

Granada, 17 de marzo 2021

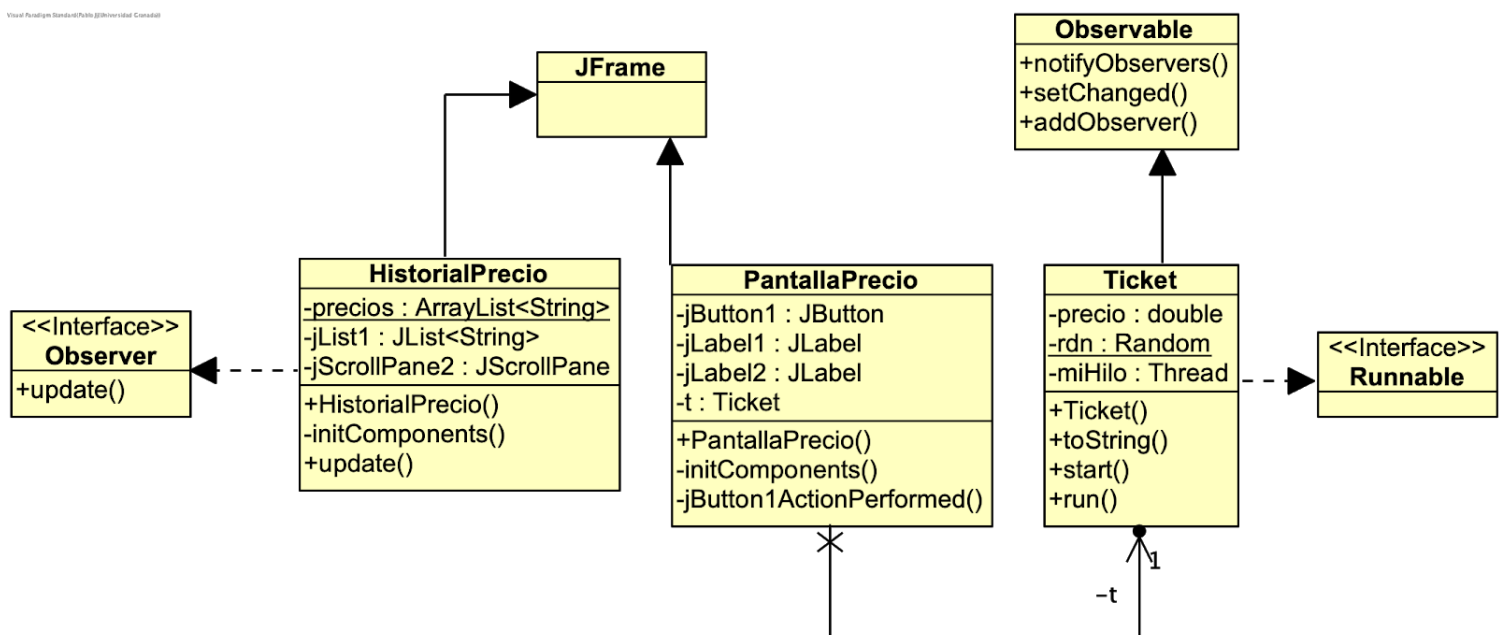
Índice:

Parte Obligatoria	1
Diagrama de clases	1
Parte Opcional	1
Diagrama de clases final	2
Capturas de la aplicación	4

Parte Obligatoria

Hemos implementado la clase `Ticket` como un sujeto *observable*. Tenemos dos *observers* `HistorialPrecio` que está suscrito y es sensible por comunicaciones push mientras que `PantallaPrecio` no está suscrito con el sujeto *observable* mantiene una comunicación pull asíncrona. Mediante una hebra cada 10 segundos se comprueba si ha habido una modificación del *observable*, también puede forzar el usuario la comprobación mediante un botón.

Diagrama de clases



Parte Opcional

Ampliamos el número de observadores suscritos sensibles por notificaciones push: `MapaPrecios` nos muestra una GUI¹ con un mapa y el precio de la entrada según la zona (aplicando el porcentaje de impuestos propio de la zona). `CambiadorPrecio` que además de mostrar el precio de la entrada (sin impuestos) permite modificar el precio del modelo a petición del usuario, igualmente el cambio es notificado a todos sus suscriptores.

¹ Graphical User Interface

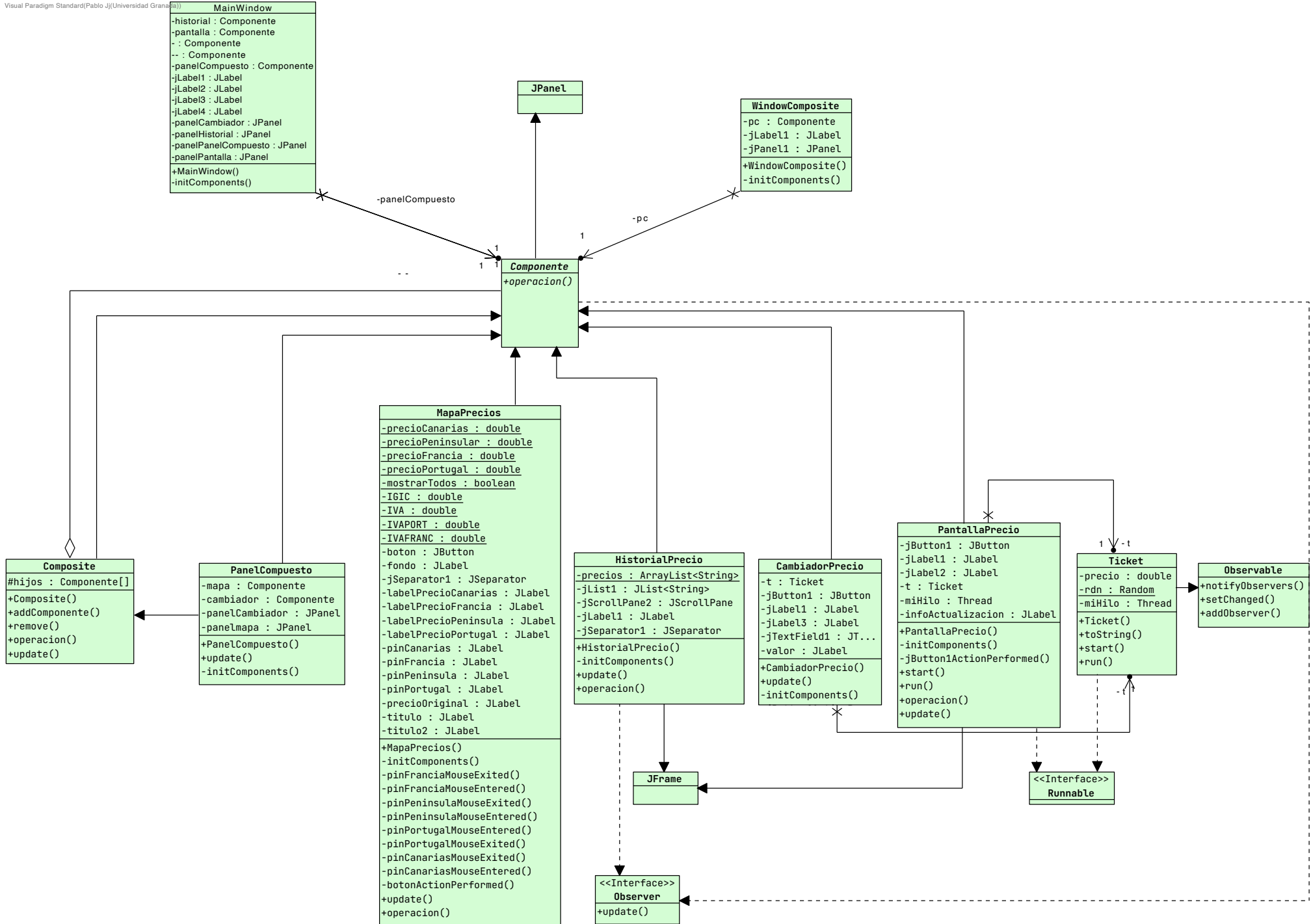
Para la ampliación con el patrón *Composite* hemos reestructurado el diagrama de clases, de manera que a partir de una clase abstracta *Componente* que implementa *Observer* heredan los diferentes observadores que hemos creado hasta ahora, además de la clase *Composite* siguiendo este nuevo patrón, de manera que todos los componentes son observadores.

Para una mayor escalabilidad, de la clase *Composite* hereda la clase de nuestro observador compuesto, *PanelCompuesto*, que es una composición de los componentes *MapaPrecios* y *CambiadorPrecio*.

De manera que al suscribir nuestro objeto compuesto al objeto observable, el compuesto se encargará de manejar la actualización de los objetos que lo componen.

Todo esto se ve mucho más claro en el siguiente diagrama de clases final.

Diagrama de clases final



Capturas de la aplicación

