



DISEÑO DE SISTEMAS

Proyecto Practico de Aplicación integrador
Caso de estudio: Museo Pictórico
Entrega n: 3



Integrantes:

- | | |
|----------------------------------------------------------------------------------------------------------------|---------------|
| • Cañete Julio, Federico (fedejulio3@gmail.com) | Legajo: 83184 |
| • Nóbile, Valentina (valunobile@gmail.com) | Legajo: 82186 |
| • Amormino, Milena Julieta (julietamormino1@gmail.com) | Legajo: 81907 |
| • Rodríguez, Milena Florencia (miluflo.rodriquez@gmail.com) | Legajo: 81323 |
| • Donalisio, Juan Pablo (ipdona@hotmail.com) | Legajo: 79420 |
| • Micheli, Genovés (genomicheli@gmail.com) | Legajo: 78493 |

Docentes:

- Cattaneo, Marcela Fabiana
- Germán Eduardo Vélez
- Martinez, Gonzalo

Fecha de entrega: 09/11/2021

Curso: 3K3

Grupo n: 3

Catedra: Diseño de sistemas



Índice

Tabla de Contenidos

Diagrama de clases – Patrón Observer (CU 102 Registrar venta de entradas)	2
Diagrama de secuencia – Patrón Observer (CU 102 Registrar venta de entradas)	3
Pseudocódigo – Patrón Observer (CU 102 Registrar venta de entradas)	5
Detalles de la implementación.....	6
Identificación del Patrón Observer - Justificación.....	6

Diagrama de clases – Patrón Observer (CU 102 Registrar venta de entradas)

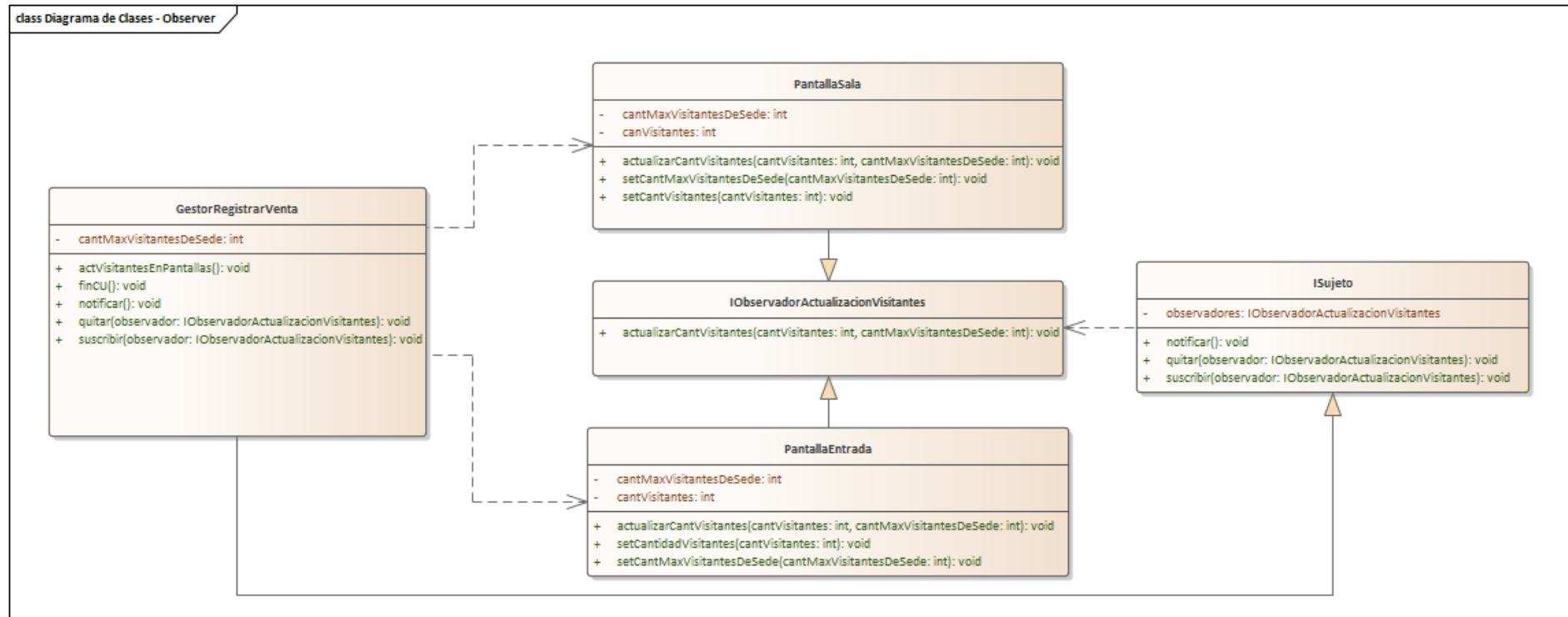
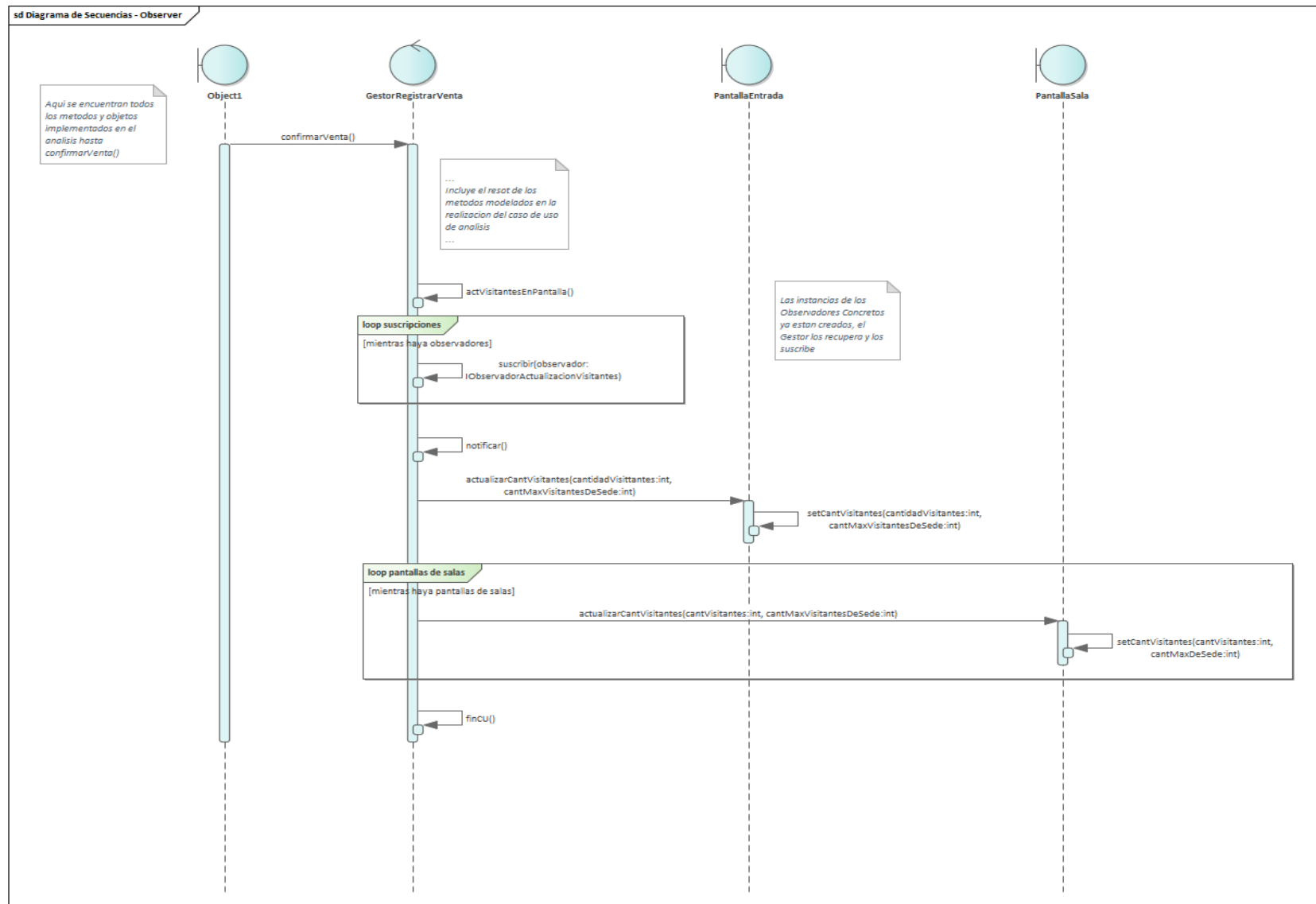


Diagrama de secuencia – Patrón Observer (CU 102 Registrar venta de entradas)



Pseudocódigo – Patrón Observer (CU 102 Registrar venta de entradas)

Interfaces: ISujeto / IObservadorActualizacionVisitantes

Sujeto concreto: GestorRegistrarVenta

Observador concreto: PantallaEntrada / PantallaSala

GestorRegistrarVenta

actVisitantesEnPantalla(): recupera las interfaces que serán observadores, luego llamando en un loop al método suscribir() del sujeto concreto para que vaya suscribiendo los observadores.

suscribir(): suscribe al observador enviándolo por parámetro, en este caso la PantallaSede y la PantallaEntrada

notificar(): llama al método actualizarCantVisitantes(), para cada interfaz suscripta.

Observadores Concretos (PantallaEntrada y PantallaSala)

actualizarCantVisitantes(): Se envía la información necesaria para generar la notificación a las interfaces. Se envían por parámetro la cantidad total actual de visitantes en la sede y la cantidad máxima de visitantes que puede haber en la misma.

setCantVisitantes(): setea la cantidad de visitantes totales en la sede actualmente.

setCantMaxVisitantesDeSede(): setea la cantidad máxima de visitantes que puede haber en la sede.

Detalles de la implementación

- El lenguaje de programación que decidimos utilizar es C# (Lenguaje orientado a objetos).
- Framework de Microsoft .NET que hace énfasis en transparencia de redes con independencia de plataforma de hardware que permite un rápido desarrollo de aplicaciones
- Aplicación de escritorio desarrollada en Visual Studio.
- Está conectada a una base de datos relacional SQL Server

Identificación del Patrón Observer - Justificación

Caso de Uso a rediseñar: 102 – Registrar venta de entradas

Situación a resolver: La forma en la que el sistema notifica a través de pantallas, la cantidad de personas que actualmente se encuentran en el museo junto con la cantidad máxima que puede haber. Es decir, en el paso número 12 de dicho caso de uso se resuelve habiendo actualizado el cambio de personas actuales en la sede, que se verán reflejados en los diferentes observadores concretos, los cuales son las pantallas de las salas y la pantalla de la entrada de la sede.

Patrón a aplicar: Observer. En este caso se utiliza una implementación PUSH del patrón ya que el sujeto actualiza a los observadores en el momento que ocurre la notificación, el observador no decide en qué momento le llega la notificación.