**Trabajo Práctico:**

**Actividades Prácticas – Guía Nro 2.**

**Modelar Sistemas con UML**

|  |  |
| --- | --- |
| Integrantes | |
|  | |
| **Nombre y Apellido** | **Legajo** |
| Adrian Mansilla | 1539619 |
| Alejandro Emanuel Segovia | 1252616 |
| Cristian Baltazar | 1415050 |

Tabla de Contenidos

[1 Supuestos: 3](#_Toc484976052)

[2 Decisiones de diseño: 3](#_Toc484976053)

[3 Diagrama de Clases 4](#_Toc484976054)

[3.1 Diagrama del servidor: 4](#_Toc484976055)

[3.2 Diagrama de la aplicación móvil: 5](#_Toc484976056)

[4 Diagramas de secuencia: 6](#_Toc484976057)

[4.1 Alta desde un punto de venta: 6](#_Toc484976058)

[4.2 Alta y finalización de un estacionamiento desde un celular: 7](#_Toc484976059)

[5 Diagrama de componentes: 8](#_Toc484976060)

[6 Diagrama de despliegue 9](#_Toc484976061)

# 1 Supuestos:

* El diagrama de clases supone la utilización de una interface para manejo de acceso a datos a través de una interface.
* El diagrama de clases supone la utilización de un interface para manejo de las conexiones por internet.
* Las comunicaciones con el sistema se realizan a través del protocolo http. Las comunicaciones contempladas son: consulta de auditoría, estacionamiento por compra en punto de venta, estacionamiento por aplicación móvil.
* Las zonas están precargadas al igual que los puntos de venta fijos. La aplicación del celular determina la zona utilizando la ubicación.

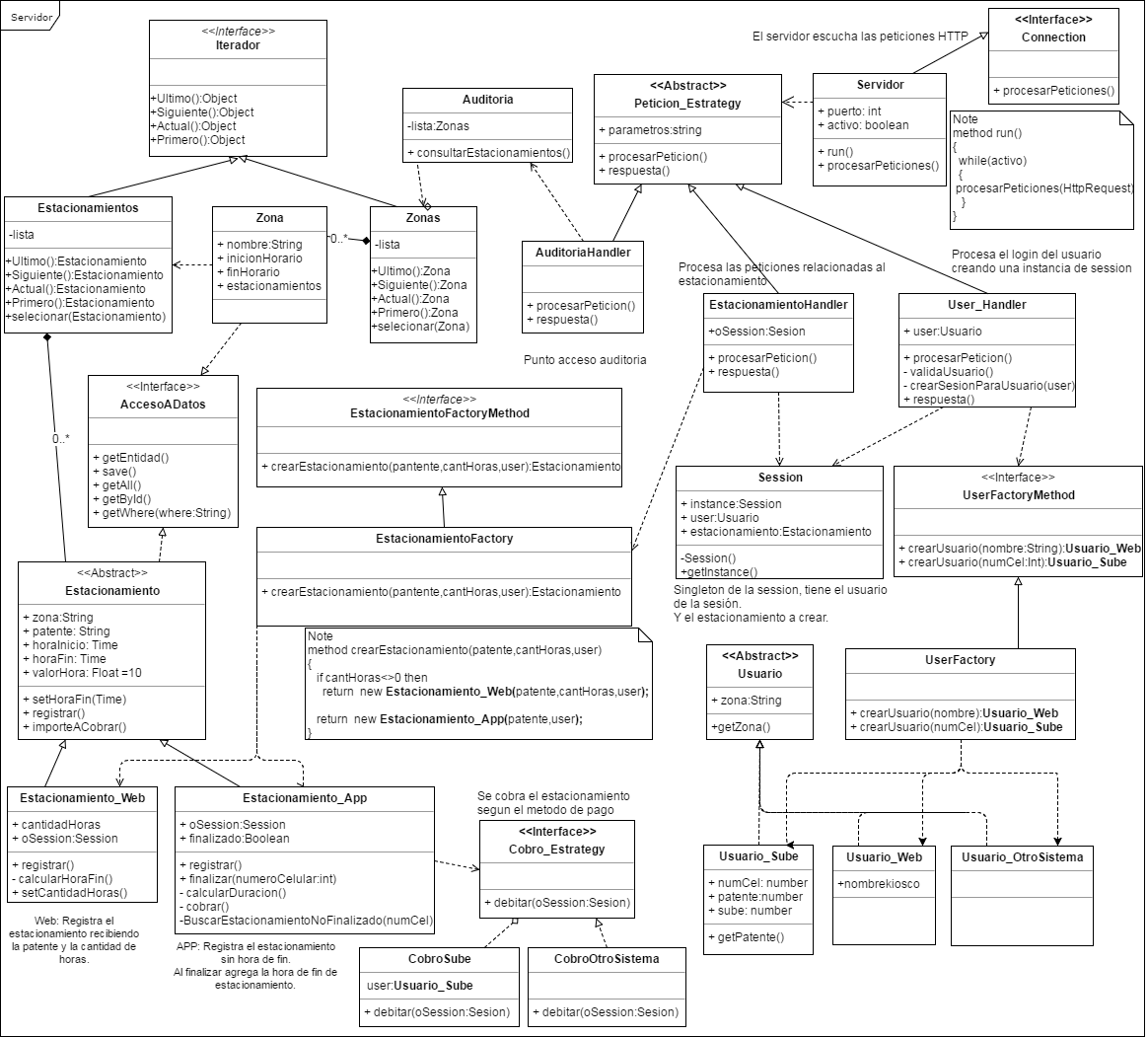
# 2 Decisiones de diseño:

* Se utilizó el patrón factory method para la creación de usuarios dependiendo de los parámetros recibidos del cliente (si se recibe un número de celular es un usuario de app movil).
* Se utilizó el patrón factory method para la creación de estacionamientos de acuerdo a los parámetros recibidos (si no se especifica un número de horas es estacionamiento desde app movil).
* Se utiliza un patrón singleton para mantener la sesión del cliente web desde que inicia la sesión hasta que confirma el estacionamiento posibilitando que el sistema informe el monto a cobrar basado en la cantidad de horas y luego se confirme la transacción.
* Se utiliza un patrón strategy para el cobro del estacionamiento desde la app móvil para que quede lista para un nuevo servicio de cobro además de SUBE.
* Se utiliza un patrón strategy para el procesamiento de las peticiones del cliente.
* Se utiliza un patrón strategy para crear el mensaje que se envía desde la app móvil al servidor dependiendo del método de pago para dejarlo preparado para futuros métodos de pago.
* Se utiliza el patrón iterator para recorrer los estacionamientos y las zonas.

# 3 Diagrama de Clases

## 3.1 Diagrama del servidor:

<https://www.draw.io/?lightbox=1&highlight=0000ff&edit=_blank&layers=1&nav=1#G0By22K6x8JxeFNkR1N0tROTNMc0E>



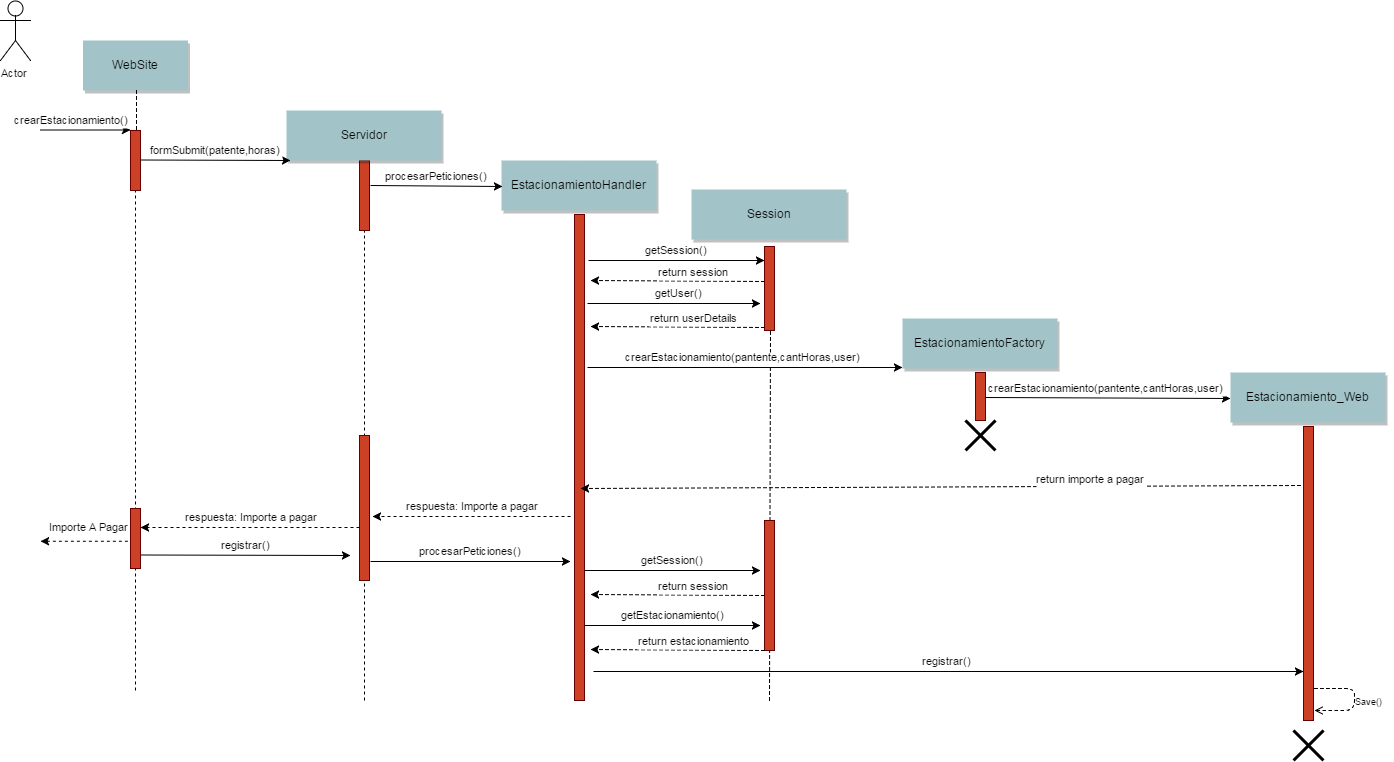
## 3.2 Diagrama de la aplicación móvil:



# 4 Diagramas de secuencia:

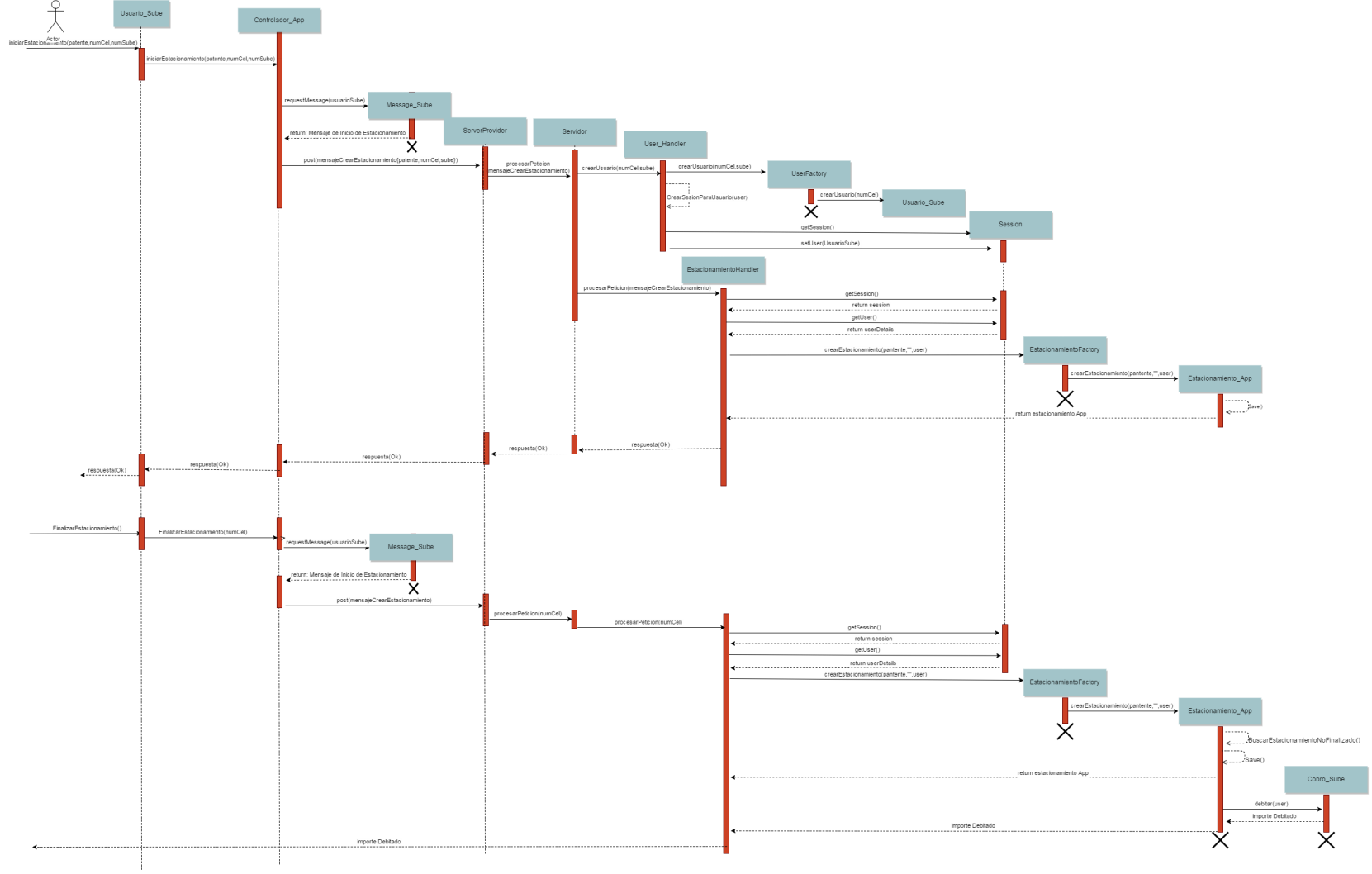
## 4.1 Alta desde un punto de venta:

<https://www.draw.io/?lightbox=1&highlight=0000ff&edit=_blank&layers=1&nav=1#G0By22K6x8JxeFQ0E1LVBlMDZjeGc>

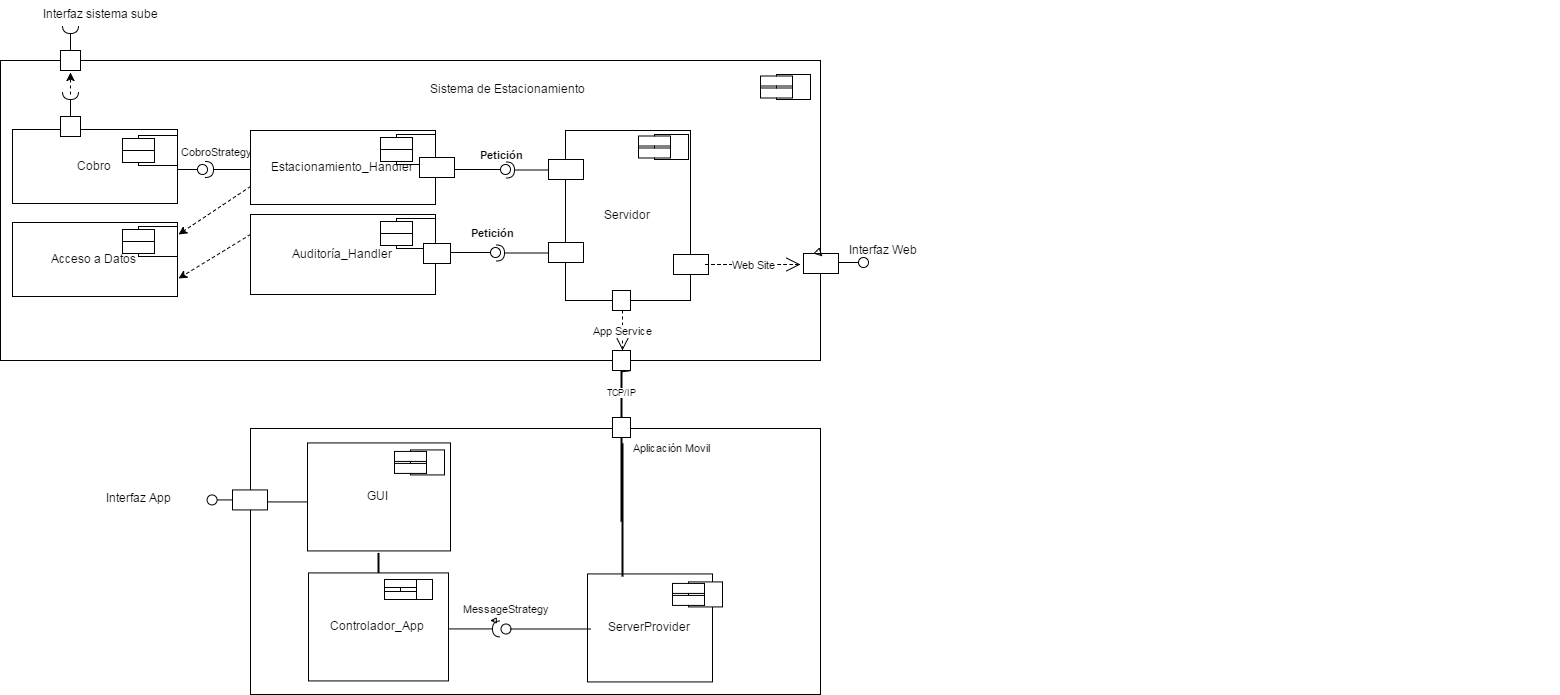


## 4.2 Alta y finalización de un estacionamiento desde un celular:

<https://www.draw.io/?lightbox=1&highlight=0000ff&edit=_blank&layers=1&nav=1#G0By22K6x8JxeFZUpaUG8zaDJxaGM>



# 5 Diagrama de componentes:



# 6 Diagrama de despliegue

