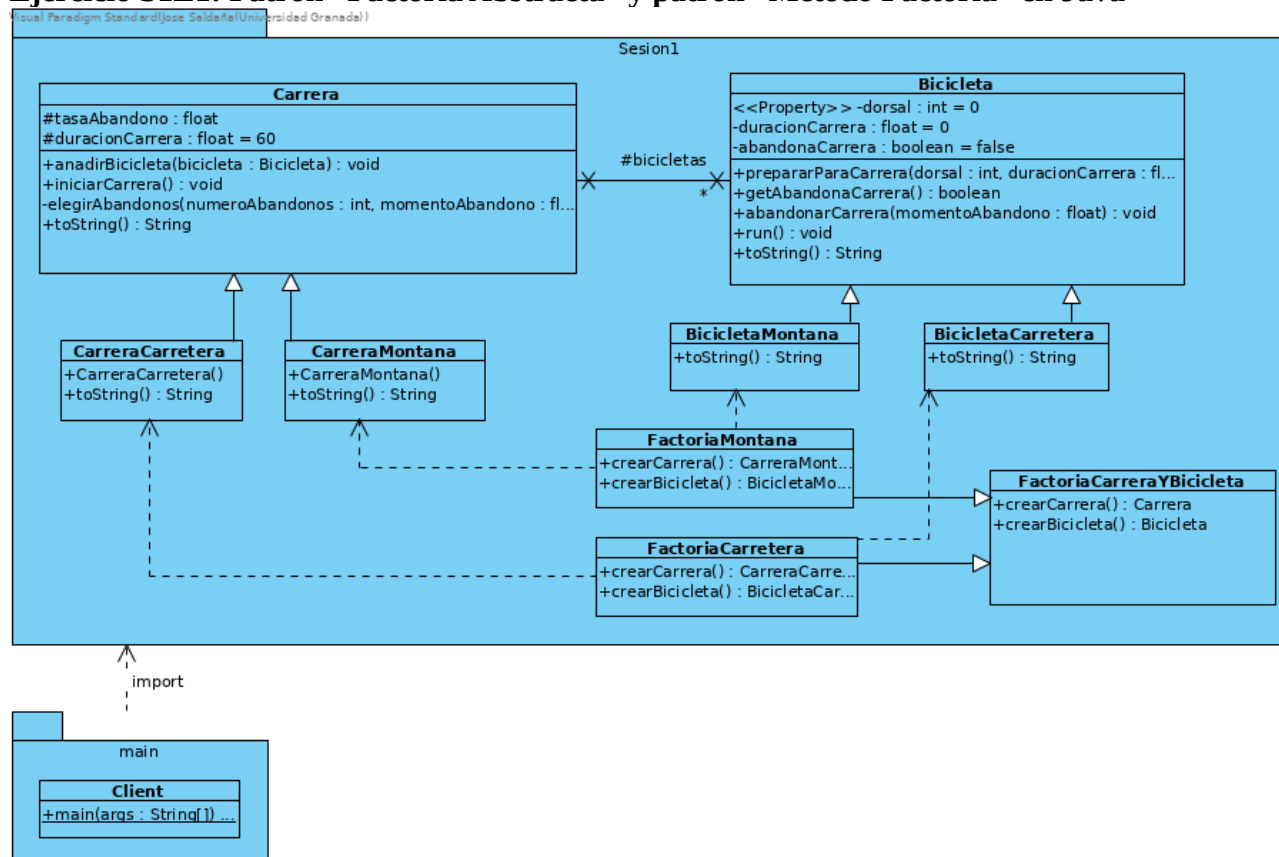


Práctica 1: 4 patrones de diseño en orientación a objetos

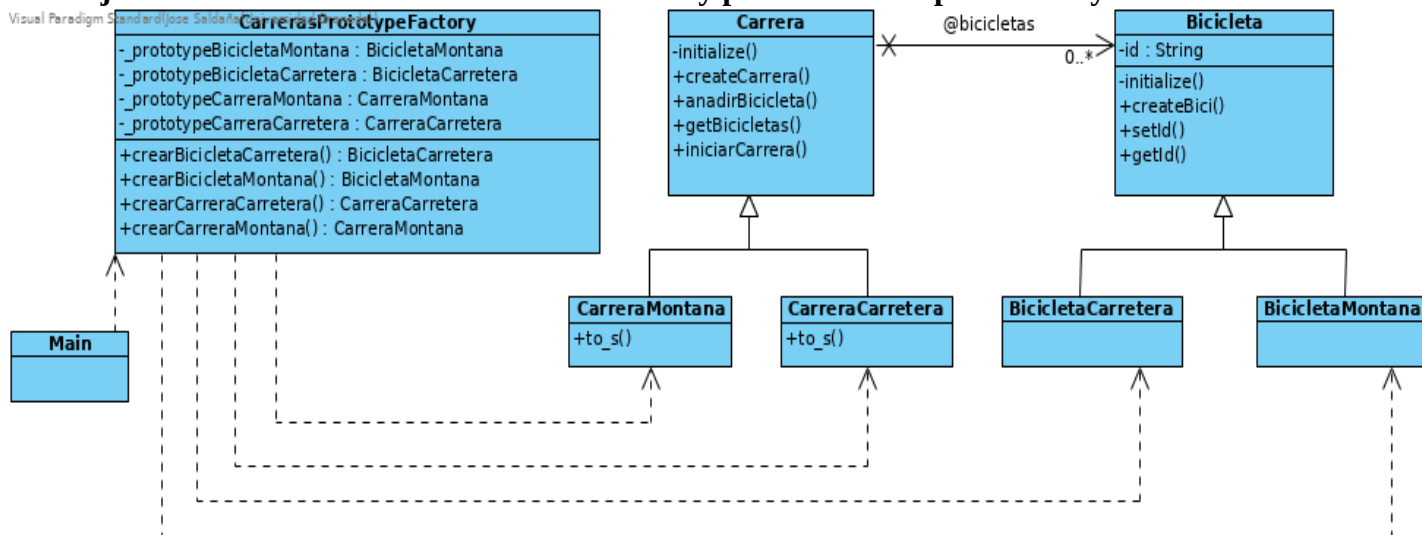
SESIÓN 1ª: Patrones “Factoría Abstracta”, “Método Factoría” y “Prototipo”

Se ha realizado tanto el ejercicio en Java como la parte opcional en Ruby. A continuación se muestran los diagramas de clases.

Ejercicio S1E1: Patrón “Factoría Abstracta” y patrón “Método Factoría” en Java



Ejercicio S1E2: Patrón “Factoría Abstracta” y patrón “Prototipo” en Ruby

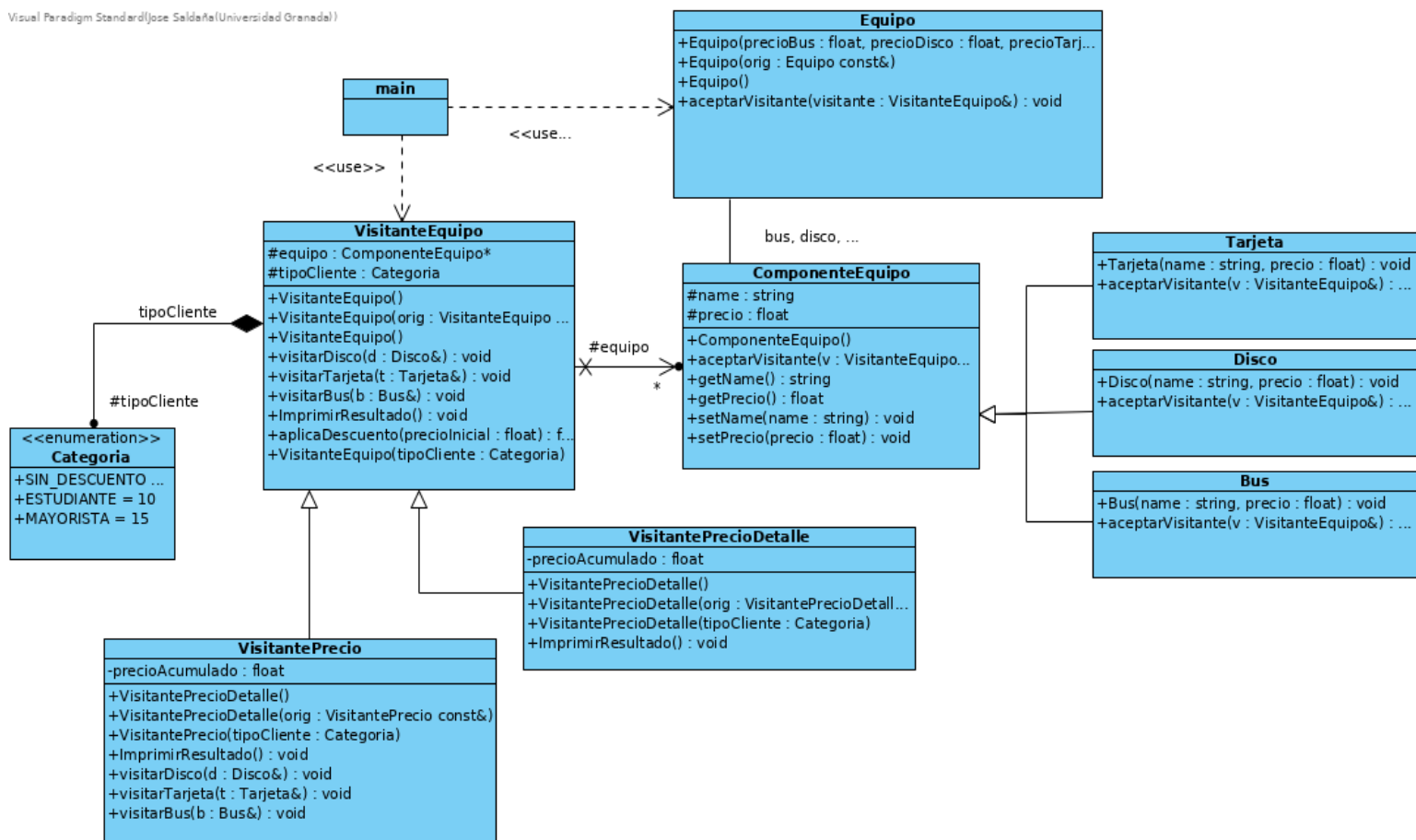


SESIÓN 2ª: Patrón visitante

Se ha realizado tanto el ejercicio obligatorio (sin categorías de visitantes) como el opcional con tres categorías de visitantes distintas, ambos con C++.

Se acompaña directamente el diagrama de **Ejercicio S2E2: Categorías de visitantes** ya que amplía al **Ejercicio S2E1: Visitante básico**

Visual Paradigm Standard(Jose Saldaña(Universidad Granada))

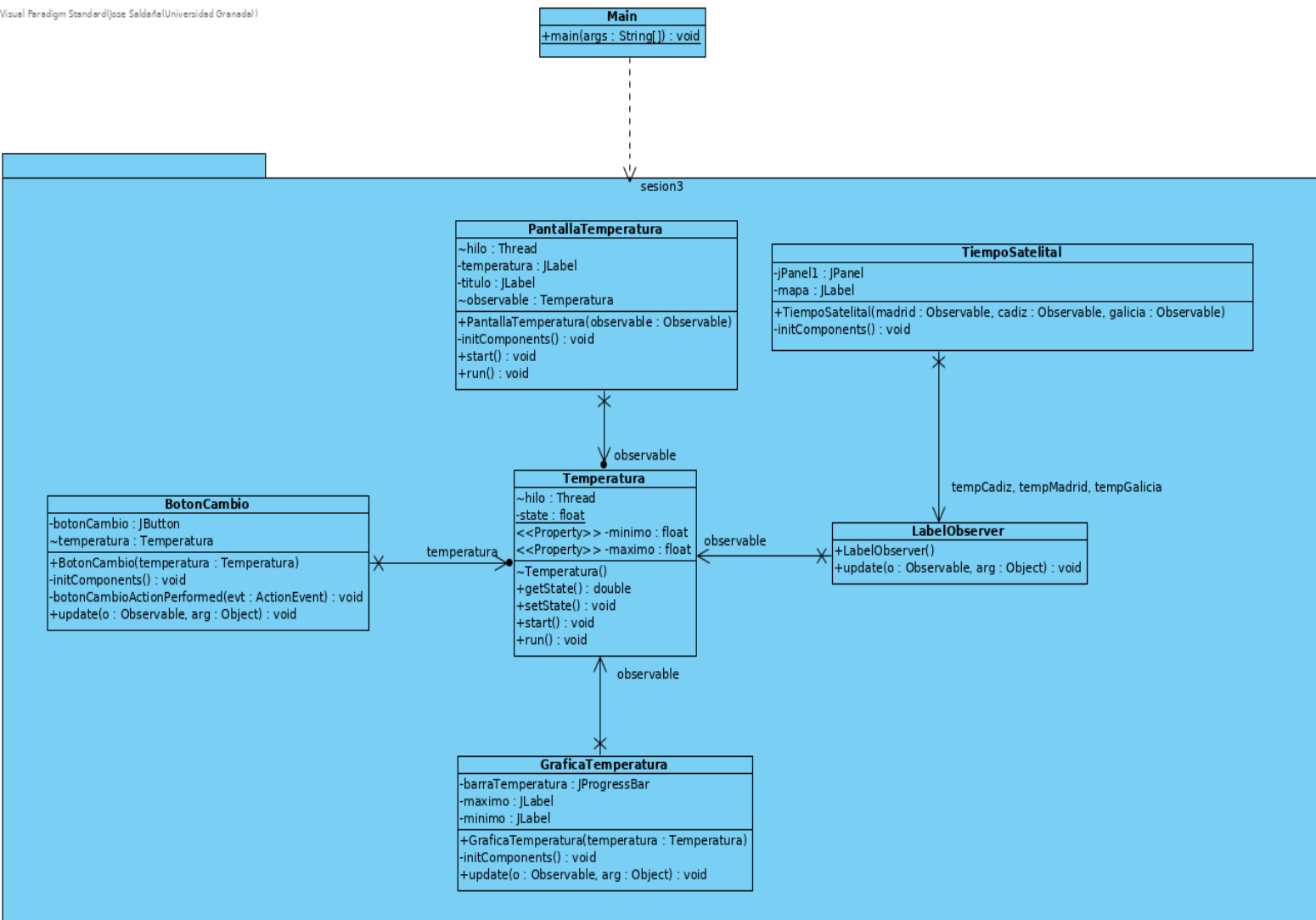


SESIÓN 3ª: Patrón Observador

Se ha realizado tanto el ejercicio obligatorio (sin mapa) como el opcional añadiendo un mapa de la clase TiempoSatelital que muestra 3 temperaturas distintas en 3 partes distintas del mapa. Todo ello implementado con Java y con Java.Swing para la GUI.

Se acompaña directamente el diagrama de **Ejercicio S3E2: Ampliación con información geográfica** ya que amplía a **Ejercicio S3E1: Monitorización de datos meteorológicos**

Visual Paradigm Standard(Jose Saldaña(Universidad Granada))



SESIÓN 4ª: Patrón Filtros de intercepción

Implementado con Java y con Java.Swing para la GUI.

Ejercicio S4E1: Simulador del movimiento de un vehículo con cambio automático

Visual Paradigm Standard(Jose Saldaña(Universidad Granada))

