

Algoritmos y Programación 3 (75.07)

Trabajo práctico 2: GPS Challenge

*2013-2c
(trabajo grupal)*

Fecha de entrega: semana del 13 de Diciembre, en fecha y hora de la práctica en la cual está inscripto el alumno (ver detalles de forma de entrega en este documento)

Nº Grupo: 2

Integrantes:

Apellido y Nombre	Nº Padrón	E-mail
Bérgamo, Ezequiel	94.759	bergamoezequiel@gmail.com
Castro Pippo, Juan Manuel	93.760	jmc.pippo@gmail.com
Di Pietro, Mauro	93.965	mdipietro@live.com.ar
Ruiz Echazú, Manuel	94.965	ruitzei@gmail.com

Corrector: Paez, Nicolás

Calificación:

Observaciones:

--

ÍNDICE

Introducción

Objetivo del trabajo

Consigna general

Descripción de la aplicación a desarrollar

Contexto

Objetivo del juego

Dinámica del juego

Vehículos

Obstáculos

Sorpresas

Escenarios

Puntajes Altos

Entregables

Forma de entrega

Fecha de entrega

Informe

Supuestos

Modelo de dominio

Diagramas de paquetes

Diagramas de clases

Detalles de implementación

Excepciones

Diagramas de secuencia

Interfaz Gráfica

Ventana Inicial

Ventana Nuevo Usuario

Ventana Cargar Usuario

Ventana Ranking

Ventana Pantalla de Juego

Ventana Instrucciones

Ventana Acerca de..

Checklist de corrección

Carpeta

Diagramas

Código

INTRODUCCIÓN

Objetivos del trabajo

Aplicar los conceptos enseñados en la materia a la resolución de un problema, trabajando en forma grupal y utilizando un lenguaje de tipado estático (Java).

Consigna general

Desarrollar la aplicación completa, incluyendo el modelo de clases e interfaz gráfica. La aplicación deberá ser acompañada por pruebas unitarias e integrales y documentación de diseño. En la siguiente sección se describe la aplicación a desarrollar.

DESCRIPCIÓN DE LA APLICACIÓN A DESARROLLAR

Contexto

La empresa Algo Ritmos SA dedicada al desarrollo de videojuegos a decidido contratar a un grupo de programadores para implementar el juego GPS Challenge.

Objetivo del juego

GPS es un juego de estrategia por turnos. El escenario es una ciudad y el objetivo, guías un vehículo a la meta en la menor cantidad de movimientos posibles.

Dinámica del juego

El juego se jugará por turnos, y en cada turno el usuario decide hacia cual de las 4 esquinas posibles avanzará.

Vehículos

El jugador podrá optar por tres diferentes tipos de vehículos.

- Moto
- Auto
- Todoterreno

Obstáculos

Al atravesar una cuadra el jugador se podrá encontrar con alguno de los siguiente obstáculos.

- Pozos: Le suma 3 movimientos de penalización a autos y motos, pero no afecta a los todoterreno.
- Piquetes: Autos y todoterrenos deben pegar la vuelta, no pueden pasar. Las motos pueden pasar con una penalización de 2 movimientos.
- Control Policial: Para todos los vehículos la penalización es de 3 movimientos, sin embargo la probabilidad de que el vehículo quede demorado por el control y sea penalizado es de 30% para las todoterreno, de 50% para los autos y de 80% par las motos ya que nunca llevan el casco puesto.
- Snorlax: La función de este obstáculo es estorbar el paso propiamente dicho. No deja pasar a ningún vehículo ni le aplica penalización alguna.

Sorpresas

También se podrán encontrar diferentes tipos de sorpresas:

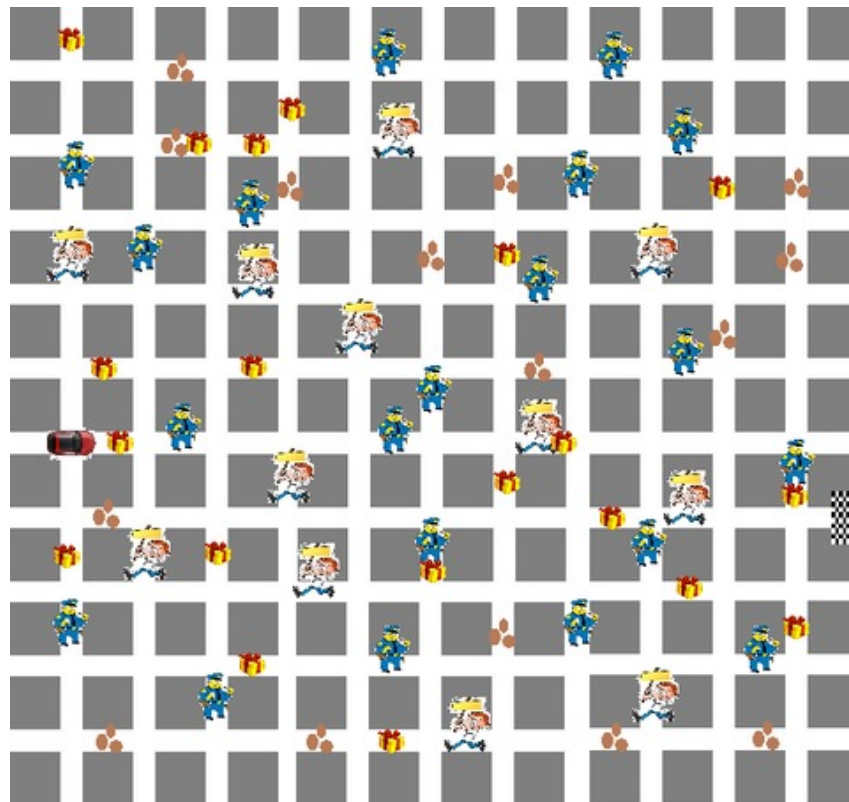
- Sorpresa Favorable: Resta el 20% de los movimientos hechos.
- Sorpresa Desfavorable: Suma el 25% de los movimientos hechos.
- Cambio de Vehículo: Cambia el vehículo del jugador. Si es una moto la convierte en auto. Si es un auto lo convierte en todoterreno. Si es un todoterreno lo convierte en una moto.

Las sorpresas figurarán en el mapa como un regalo y no se sabrá que es hasta que el vehículo la accione.

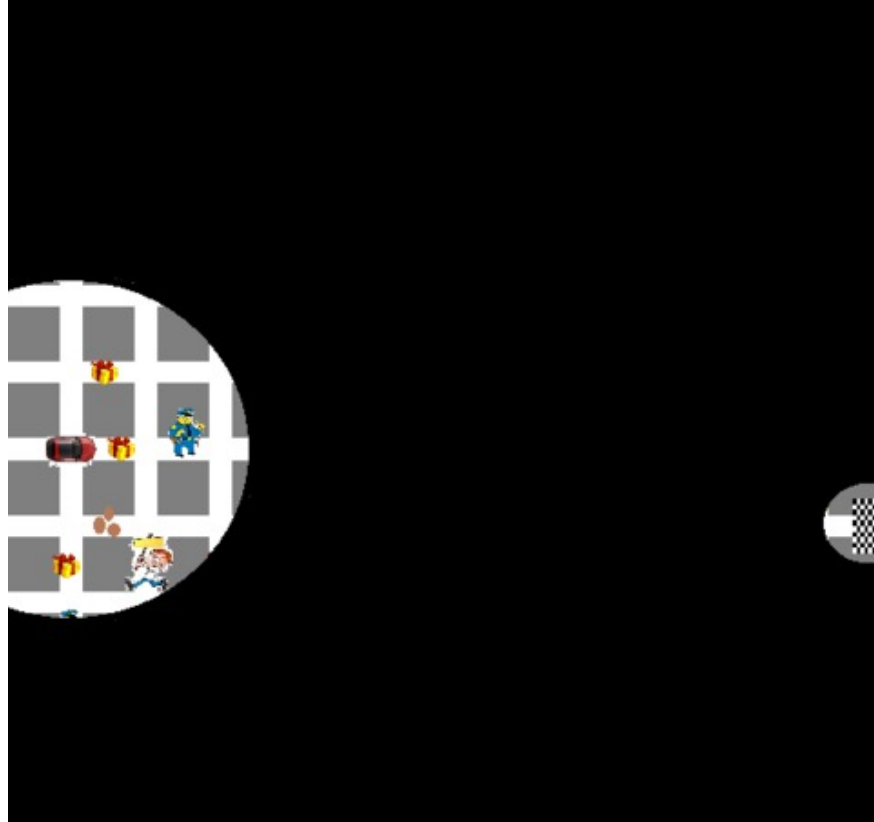
Escenario

Para hacerlo más interesante y jugable, el jugador no podrá ver más que dos manzanas a la redonda de la posición de su vehículo y la bandera a cuadros que marca la meta. El resto del mapa permanecerá en sombras.

El tamaño del escenario no será fijo, y tendrá un punto de partida y una meta.



Ejemplo de escenario



Ejemplo de escenario como lo visualiza el jugador

Puntajes altos

Se debe almacenar un ranking donde figuren los mejores puntajes asociados a un nickname que indique el usuario.

ENTREGABLES

- Código fuente de la aplicación completa, incluyendo también: código de la pruebas, archivos de recursos.
- Script para compilación y ejecución (ant).
- Informe, acorde a lo especificado en este documento.

Forma de entrega

A coordinar con el docente asignado.

Fechas de entrega

Se deberá validar semanalmente con el docente asignado el avance del trabajo. El docente podrá solicitar ítems específicos a entregar en cada revisión semanal.

La entrega final deberá ser en la semana del 13 de diciembre, en la fecha del curso en que se está inscripto.

INFORME

Supuestos

- El conductor no avanza directamente hacia la dirección que le indicamos desde el control por teclado. El conductor en caso de no estar orientado en la dirección que deseamos que se mueva primero se orientará. En cambio, si el conductor ya está orientado en la dirección que le indicamos que debe moverse entonces efectivamente avanzará hacia la siguiente esquina. Derivado de lo anteriormente dicho surge la necesidad de dotar al Conductor de una dirección en la que éste esté mirando.
- Las sorpresas, a diferencia de los obstáculos, desaparecen una vez que se interactúa con ellas.
- En la consigna inicial se indicaba que había tres niveles de dificultad. Al terminar cualquiera de ellos (el que fuera seleccionado por el usuario) se finalizará la partida. Lo que proponemos e implementamos es que no se elija el grado de dificultad sino que arranque a jugar desde un nivel fácil y se vaya pasando de nivel con su correspondiente aumento de dificultad. En caso de terminar todos los niveles o perder en alguno de ellos se finalizará la partida.
- La puntuación final de la partida será equivalente a la suma de los puntajes que haya obtenido en todos los niveles que haya completado con éxito.
- La dificultad del nivel está relacionada con la cantidad de objetos, el tamaño del mapa y el alcance de visión del conductor.
- La visibilidad no es persistente, es decir que si una celda está dentro del rango de visión del conductor ésta se hará visible, en caso contrario, ésta se volverá nuevamente invisible.
- Se decidió agregar un Obstáculo nuevo llamado Snorlax, que no deja pasar a ningún vehículo ni los penaliza. Este obstáculo nos permite mayor grado de flexibilidad a la hora de diseñar niveles y encauzar su dificultad.
- A diferencia de lo que plantea el enunciado decidimos que el Objeto Llegada (la bandera) solo se pueda observar si está dentro del rango de visión del conductor. Ya que de manera contraria se vuelve mucho menos interesante al no tener que buscarla al mismo tiempo que disminuye enormemente el grado de dificultad que buscamos.
- A los jugadores que completen de forma exitosa la totalidad de los niveles se los bonificará con un bonus que multiplica x5 su puntuación total.

- No puede haber más de un objeto encontrable entre dos esquinas.
- Los mapas deben tener dimensiones cuadradas. Esa decisión viene dada por el hecho de que si el mapa del nivel es rectangular en la vista se vería distorsionada al dibujarse sobre un JFrame cuadrado.

Extensión

Se nos planteó pensar como haríamos para que las celdas tengan visibilidad persistente. Es decir, que una vez que se encuentren dentro del alcance de visión del Conductor y se hagan visibles no vuelvan a su estado anterior.

La solución a este problema es bastante sencilla. La celda tiene un método que se llama `esVisible` que calcula si su distancia al Conductor es menor al alcance de visión del mismo y en caso efectivo la celda se hace visible.

Bueno, lo que habría que hacer es guardar en un boolean la visibilidad (que arrancaría en Falso por default). Entonces, al momento de llamar al método `esVisible` lo que se haría es primero preguntar si ya era visible devolver Verdadero. Y solo en caso de que fuera Falsa esa afirmación calcular la distancia como dijimos antes y setear el boolean de visibilidad en función de ello.

MODELO DE DOMINIO

- Patrón Singleton para la implementación de los distintos tipos de Vehículos.
Ya avanzados en el desarrollo del TP notamos que la única función que tenían los distintos tipos de vehículos que podía usar un Conductor era la de poder diferenciar unos de otros a la hora de interactuar con un Objeto Encontrable. Al no notar un cambio de comportamiento o diferencia entre dos posibles instancias de un mismo Vehículo, nos inclinamos por utilizar el patrón Singleton para evitar las innecesarias instanciaciones de esta clase.
- Patrón observador (Ventana - Juego, Conductor)
La Ventana observa al Juego y al Conductor.
Cada vez que el conductor avanza o se reorienta notifica a la ventana para que se redibuje.
Cuando el Juego se gana, se pierde o se pasa de nivel notifica a la ventana.
En caso de que el juego se gane o se pierda se carga el usuario con su puntuación actual a la tabla de posiciones y se muestra la misma.
En caso de que se pase de nivel se refresca la pantalla con el nuevo nivel y setea como observado al nuevo conductor.
- Patrón observador (Nivel - Conductor)
El Nivel observa al Conductor.
Cada vez que el conductor intenta avanzar se notifica al Nivel que revisa si le quedan movimientos al conductor y si éste ya llegó a la meta.
En caso de quedarse sin movimientos notifica al juego de que el jugador perdió.
Por otro lado, si el Conductor alcanza la meta notifica al juego para que actúe en consecuencia.
- Double Dispatch (Objetos Encontrables - Vehículos del Conductor)
Decidimos utilizar este patrón de diseño para modelar la interacción entre los Vehículos del Conductor (Auto, Moto, Todoterreno) con los Objetos Encontrables (Pozo, Sorpresas, etc...)
- MVC
Decidimos aplicar el patrón de diseño Modelo Vista Control (MVC) para poder separar responsabilidades y disminuir el acoplamiento.
El Control por teclado conoce al modelo y le manda señales para que éste cambie su estado. El modelo notifica a la vista consecuentemente, la cual lo refleja en la pantalla para que lo vea el usuario.

Detalles de Implementación

Desarrollo de qué pasa cuando se avanza y se alcanza la meta en un nivel.

Cuando el conductor avanza y se encuentra con la Bandera de Llegada interactúa con ella. Esta interacción consiste en setear su estado de `encontroLlegada` en verdadero.

Como en cada avance el conductor notifica al nivel (por medio del patrón `observer`), éste último revisa los movimientos del conductor y su estado de `encontroLlegada` para decidir si dejar que continúen con el nivel o notificar al juego de que el Conductor Perdió o finalizó un nivel.

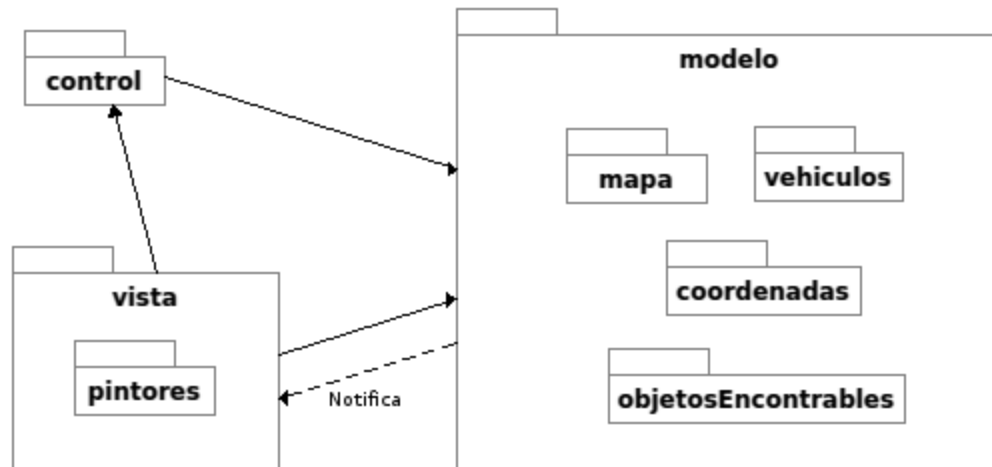
El juego cuando es notificado de que se finalizó un nivel analiza si terminó la partida (porque ya pasó todos los niveles) o si debe setear el `nivelActual` por el siguiente nivel.

Cuando la partida finaliza (sea que el jugador haya perdido o ganado) el juego cargará el nuevo puntaje con el nombre del usuario actual en la tabla de puntuaciones y luego se lo notificará a la ventana (ya que ésta es su observadora).

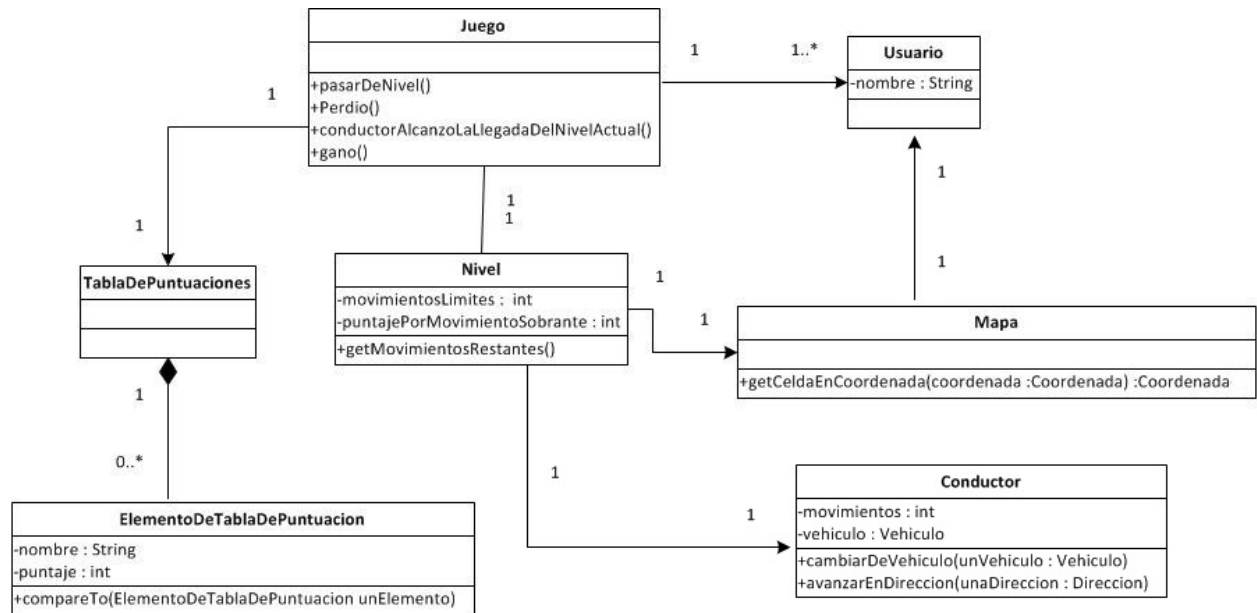
Excepciones

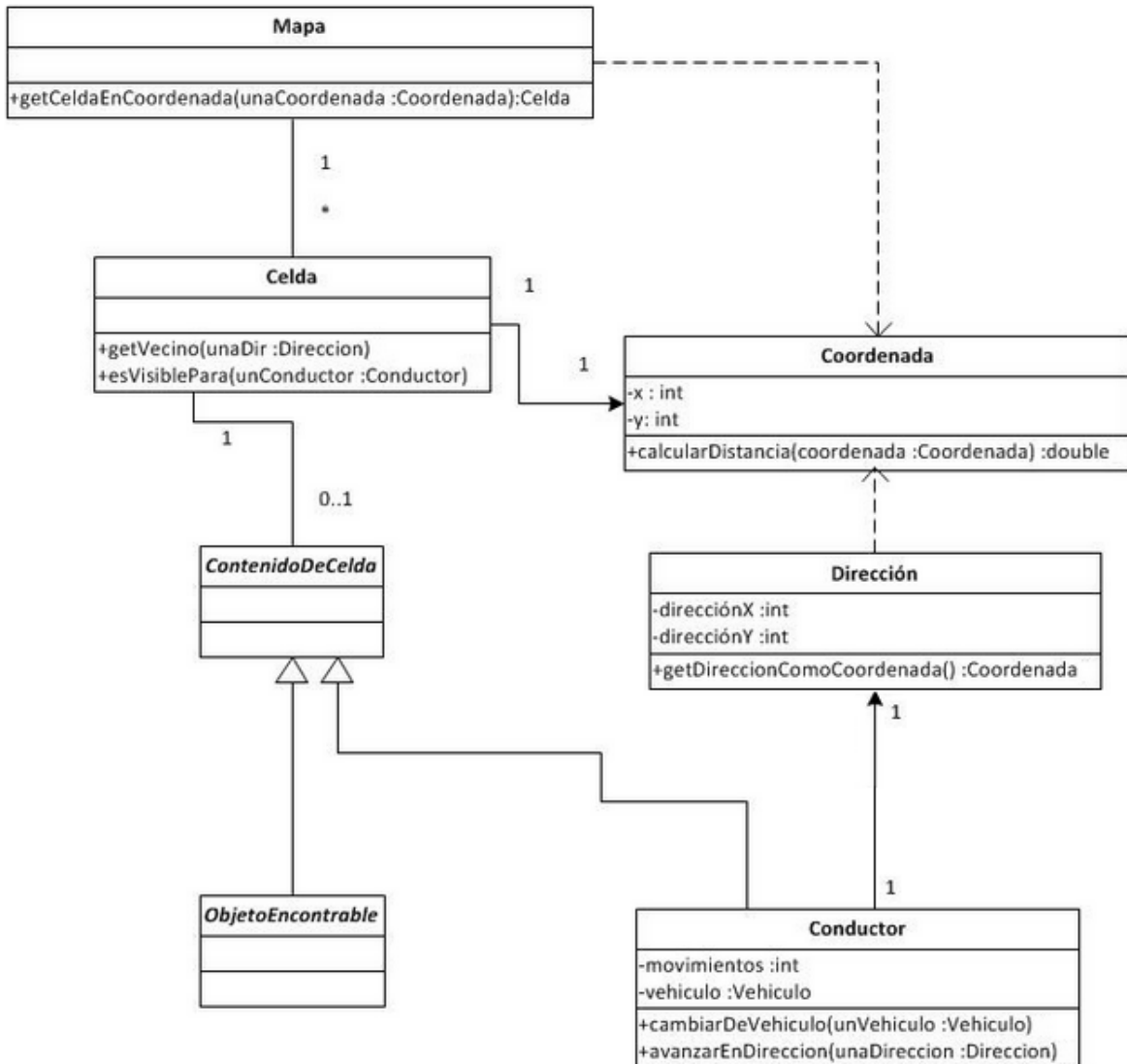
- *UbicacionEnMapaException: Al inicializar un Mapa, este se crea con un determinado número de cuadras*
Esta excepción se lanza en 2 casos:
 - 1- Cuando un Conductor intenta moverse en una dirección que excede los límites del mapa actual.
 - 2- Por la lógica del juego, cuando un conductor se mueve en una dirección, este siempre se mueve de esquina a esquina, y entre esas 2 esquinas puede (o no) haber algún `ObjetoEncontrable`. Para evitar posibles errores de posicionamiento de estos `ObjetosEncontrables`, cuando se intente agregar alguno de estos objetos en una posición del mapa inválida, se lanzará esta excepción.
- *NombreDeUsuarioInvalidoException: esta excepción se lanza cuando se intenta crear un usuario ya existente o que tiene su string nulo (es decir "").*

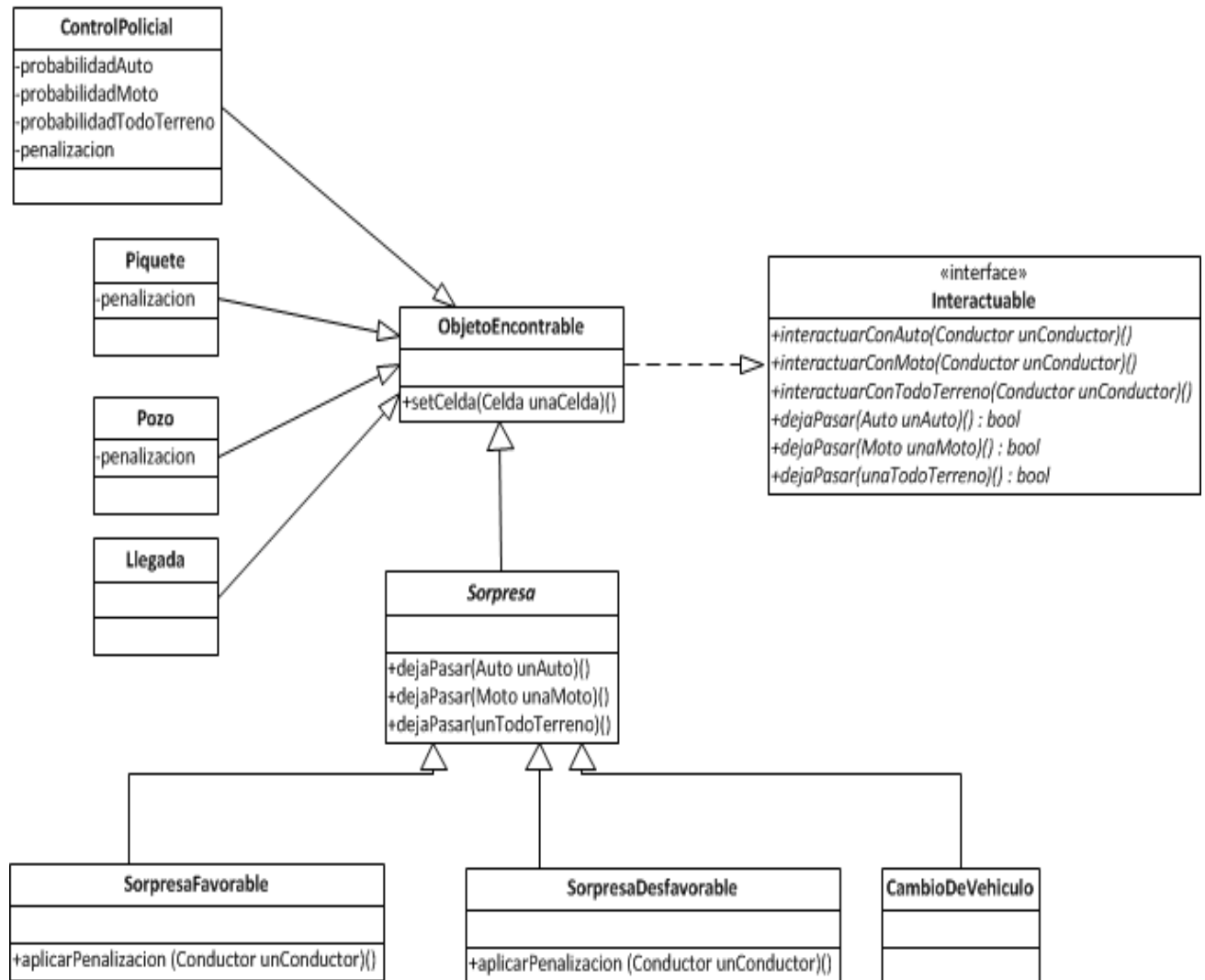
Diagramas de Paquetes



Diagramas de Clases

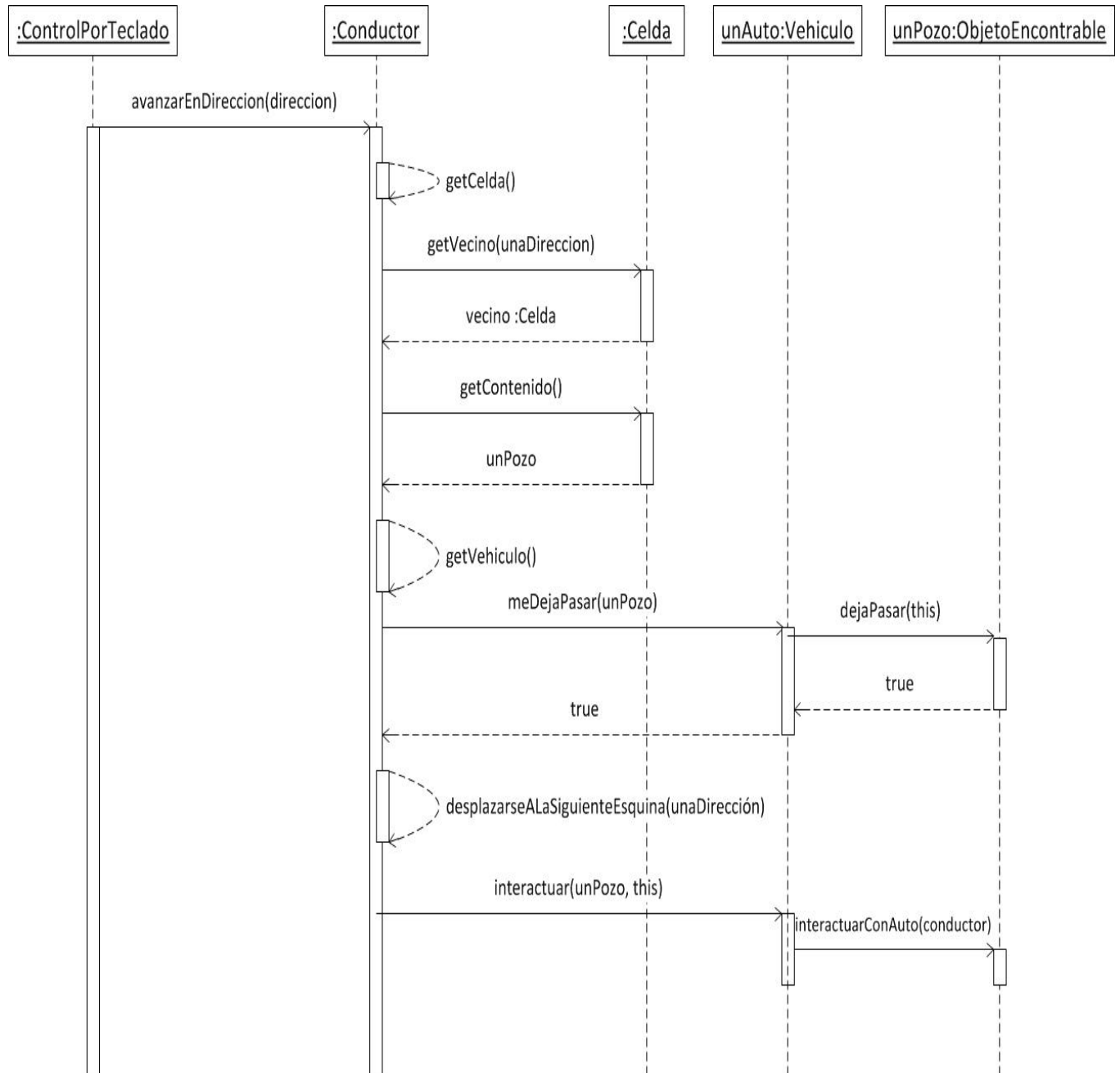




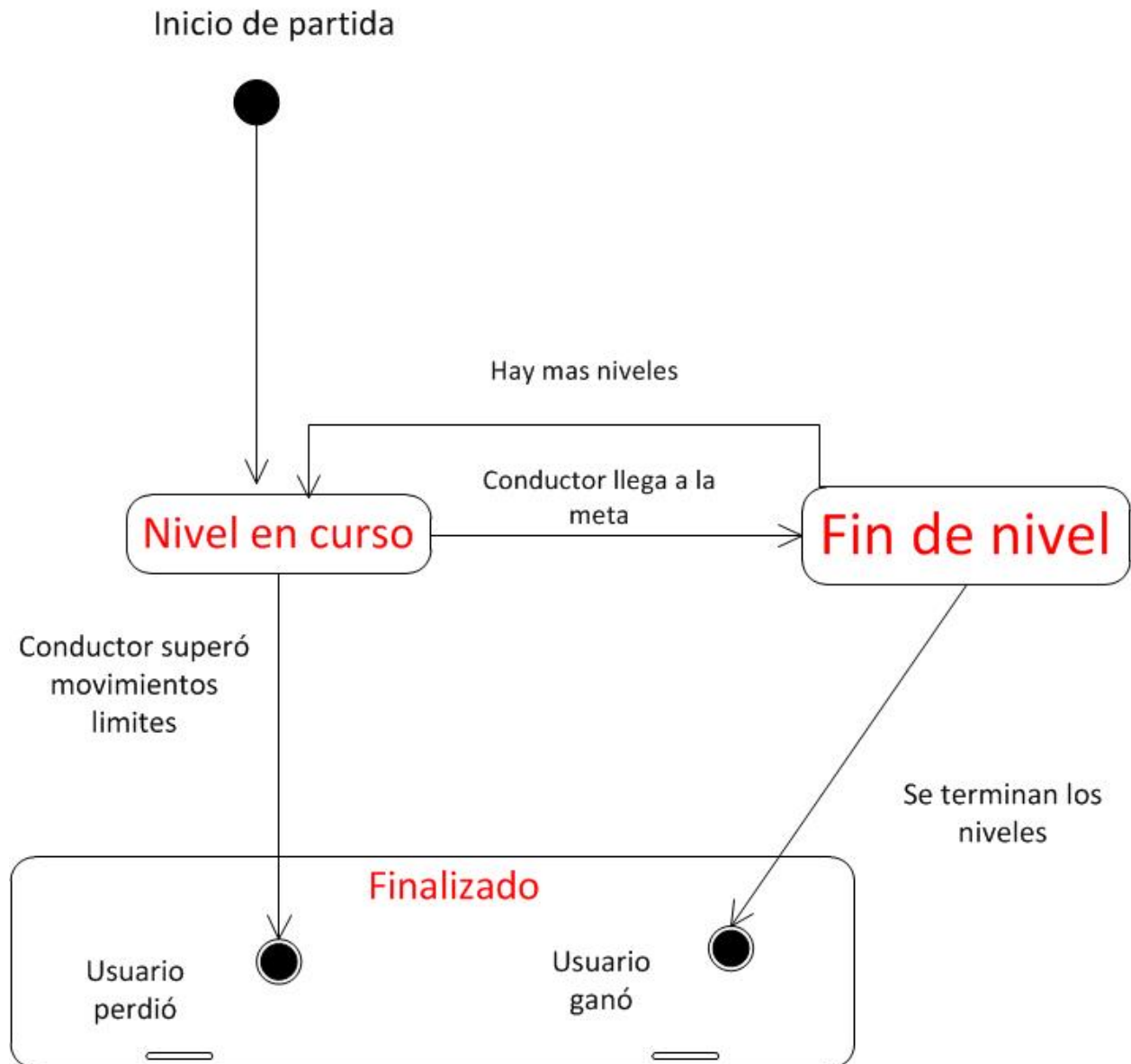


Diagramas de Secuencia

Se asume que el conductor está orientado en la dirección en la que se quiere mover.

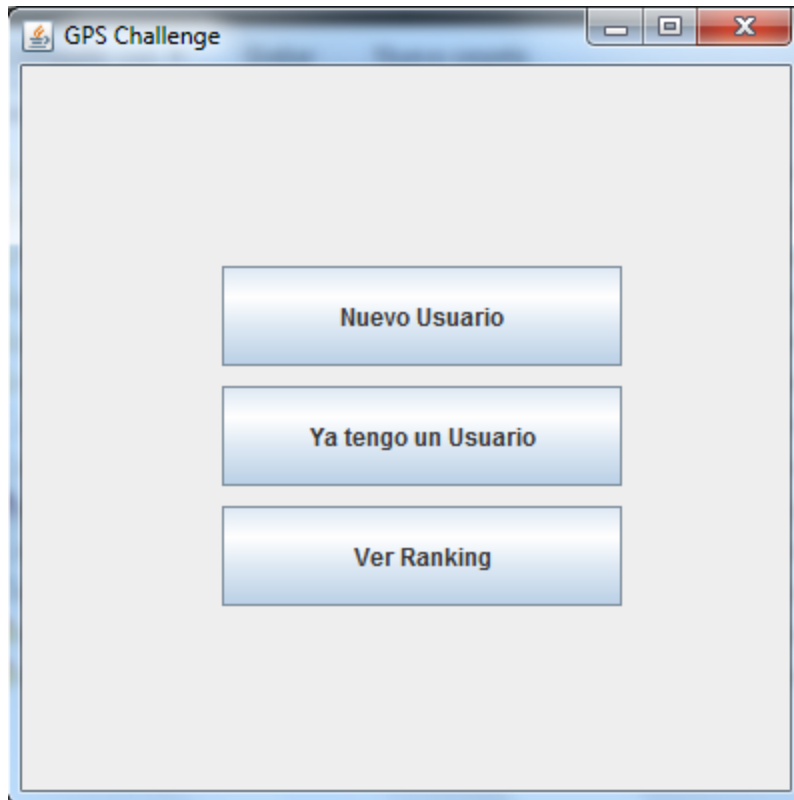


Diagramas de Estado



INTERFÁZ GRÁFICA

Ventana Inicial



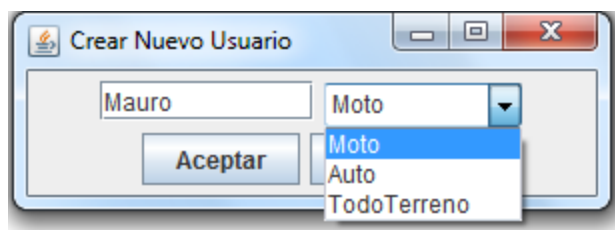
(Imagen 1)

Es la ventana que se muestra cuando se inicia el juego.

Esta ventana consta de tres botones:

- “Nuevo Usuario” (redirige a la ventana Nuevo Usuario (ver Imagen 2).
- “Ya tengo un Usuario” (redirige a la ventana Cargar Usuario (ver Imagen 3).
- “Ver Ranking” (redirige a la ventana Ranking (ver Imagen 4).

Ventana Nuevo Usuario

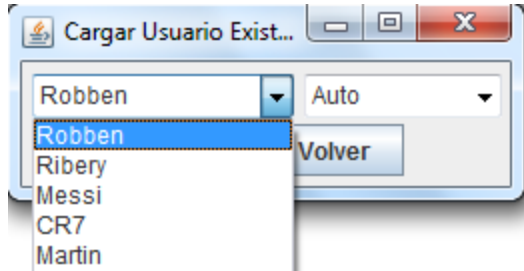


(Imagen 2)

En esta ventana podemos crear un usuario nuevo y seleccionar un tipo de vehículo para

arrancar. Una vez seleccionados los ítems correspondientes se nos redirige a la ventana de Pantalla De Juego (ver imagen 5).

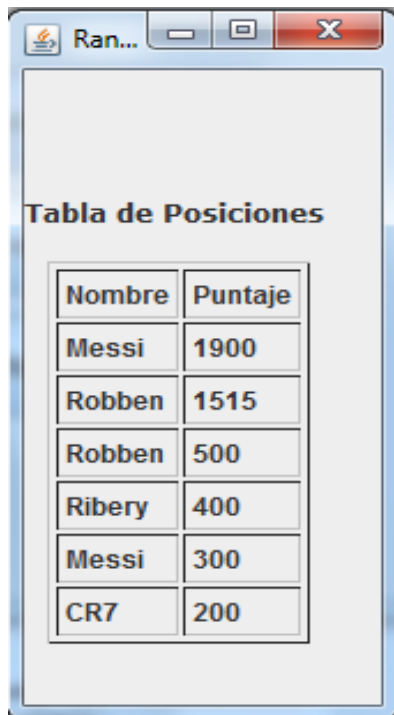
Ventana Cargar Usuario



(Imagen 3)

En esta ventana podemos seleccionar un usuario existente y seleccionar un tipo de vehículo para arrancar. Una vez seleccionados los ítems correspondientes se nos redirige a la ventana de Pantalla De Juego (ver imagen 5).

Ventana Ranking



Nombre	Puntaje
Messi	1900
Robben	1515
Robben	500
Ribery	400
Messi	300
CR7	200

(Imagen 4)

Esta ventana nos muestra la tabla con las mejores puntuaciones.

La tabla no muestra un número ilimitado de puntuaciones, muestra un TOP definido por una constante.

Ventana Pantalla De Juego



(Imagen 5)

En esta ventana se desarrolla el juego.

Esta compuesta por:

- Menús
 - Opciones
 - Nueva Partida
 - Ver Ranking
 - Salir
 - Sistema
 - Guardar Partida
 - Cargar Partida
 - Ayuda
 - Instrucciones (ver imagen 6)
 - Acerca de... (ver imagen 7)
- Pantalla de Mapa de Nivel
- Panel Derecho

Pantalla de Mapa de Nivel

En este frame es el eje del juego. Es donde se maneja al conductor, se muestran los diferentes obstáculos del mapa y donde se ven representadas sus interacciones.

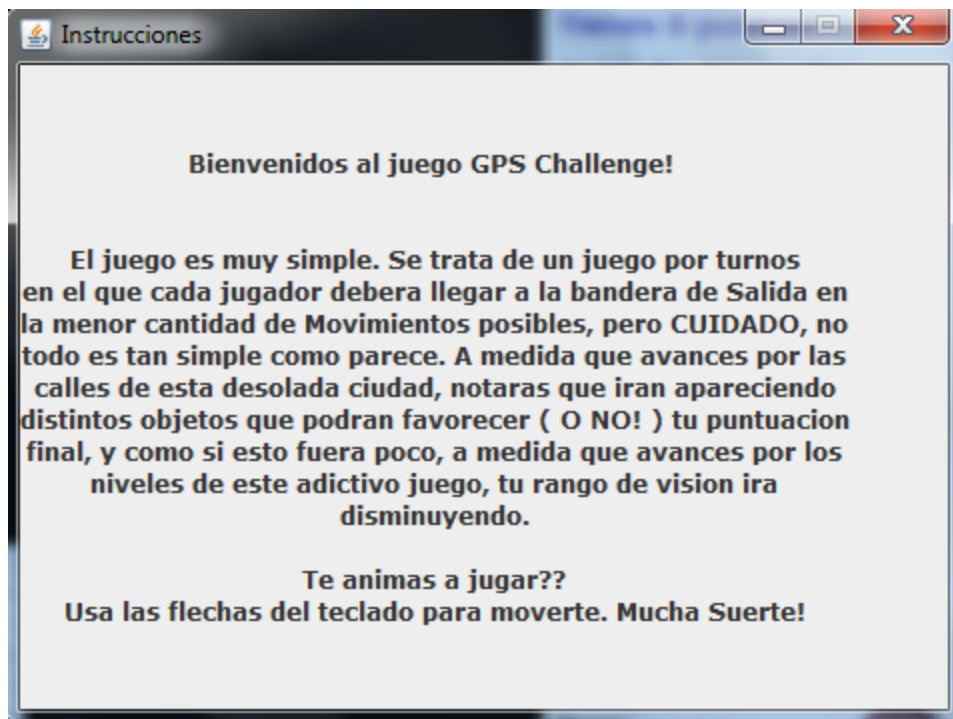
Pantalla Derecho

El panel derecho muestra toda la información relevante que le pueda servir al usuario que está jugando.

Entre estas cosas se encuentran:

- Nombre de usuario
- Puntuación
- Cantidad de movimientos restantes
- Vehículo actual
- Bonus que aplica al nivel.
- Referencias

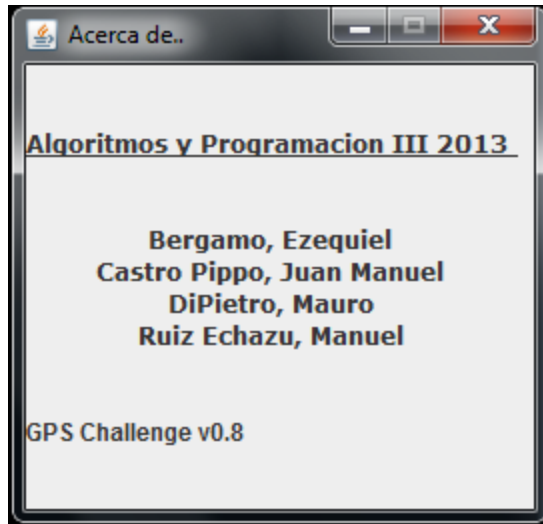
Ventana de Instrucciones



(Imagen 6)

Esta ventana muestra las instrucciones del juego.

Ventana Acerca de..



(Imagen 7)

Esta ventana muestra los integrantes del grupo. Asimismo indica la asignatura para el que fue realizado el proyecto y el año de desarrollo del mismo.

CHECKLIST DE CORRECCIÓN

Esta sección es para uso exclusivo de la cátedra, por favor no modificar.

Carpeta

Generalidades

- ¿Son correctos los supuestos y extensiones?
- ¿Es prolija la presentación? (hojas del mismo tamaño, numeradas y con tipografía uniforme)

Modelo

- ¿Está completo? ¿Contempla la totalidad del problema?
- ¿Respeto encapsulamiento?
- ¿Hace un buen uso de excepciones?
- ¿Utiliza polimorfismo en las situaciones esperadas?

Diagramas

Diagrama de clases

- ¿Está completo?
- ¿Está bien utilizada la notación?

Diagramas de secuencia

- ¿Está completo?
- ¿Es consistente con el diagrama de clases?
- ¿Está bien utilizada la notación?

Diagrama de estados

- ¿Está completo?
- ¿Está bien utilizada la notación?

Código

Generalidades

- ¿Respeto estándares de codificación?
- ¿Está correctamente documentado?