

Trabajo Práctico Grupal

Enunciado del problema

El *Banco de la Plaza* requiere el desarrollo de un prototipo evolutivo para la construcción de un Cajero Automático (ATM – Automatic Teller Machine).

Las funciones que debe implementar el prototipo son las siguientes:

- Identificar Usuario: El usuario deberá ingresar su contraseña numérica de 4 dígitos. Si se valida, el cliente procederá a operar el cajero, caso contrario, se informará que la clave introducida es errónea y se solicitará que la ingrese nuevamente. Si la identificación falla durante 2 intentos más, la tarjeta será retenida por el ATM e inhabilitada para usos futuros.
- Una vez que el cliente es identificado satisfactoriamente, el sistema mostrará la pantalla de *Menú Principal* que permitirá realizar las siguientes operaciones:
 - Cambio de Clave
 - Consulta de Saldo
 - Retirar dinero
 - Depositar dinero
 - Transferencia de Dinero
 - Consulta de Movimientos

Información relevada de la entidad bancaria:

- Los clientes que podrán operar el ATM pertenecen tanto al *Banco de la Plaza* como a otros bancos. Cada banco tiene asignado un rango de números de tarjeta, lo que permitirá determinar el banco al cual pertenece una tarjeta y al cual el ATM deberá pedir datos de validación del cliente, tipo de cuenta, saldos, etc.
- Hay distintos tipos de movimientos de débito o crédito:
 - *Créditos*: Depósito en Efectivo / Acred. Haberes / Liq. de intereses (simulado) / Transferencia desde otra Cuenta
 - *Débitos*: Extracción ATM / Transferencia a otra cuenta / Compra en comercio (simulado) / Cargo por ser *Cliente Otro Banco* / Cargo por *Mantenimiento de Cuenta* (simulado)
 - Nota: Los movimientos simulados no se realizarán operando el ATM sino que se generarán desde un proceso agendado.
- El sistema trabajará con distintos tipos de cuentas bancarias:
 - *Caja de Ahorro*: No permite operaciones en descubierto. Si es un cliente de otro banco, por cada extracción o depósito se debitará de la caja de ahorro del cliente, un monto fijo de \$10. El cargo mensual por mantenimiento de cuenta es de \$150.
 - *Cuenta Sueldo*: Es una Caja de Ahorro a la que, independientemente del Banco a la que pertenezca, no se le debitará ningún importe por transferencias, depósitos, extracciones, ni ninguna otra operación, etc. No tiene cargo de mantenimiento de cuenta. Se debe guardar el CUIT de la empresa que le deposita el sueldo (para simplificar, asumiremos que sólo una empresa le depositará en la cuenta)
 - *Cuenta Corriente*: El cliente tiene acordado un monto de extracciones “en descubierto” (puede ser distinto en cada cliente). Si el cliente es de otro banco, por cada extracción o depósito se le debitarán \$30. Pueden hacerse únicamente 3 extracciones por ATM por

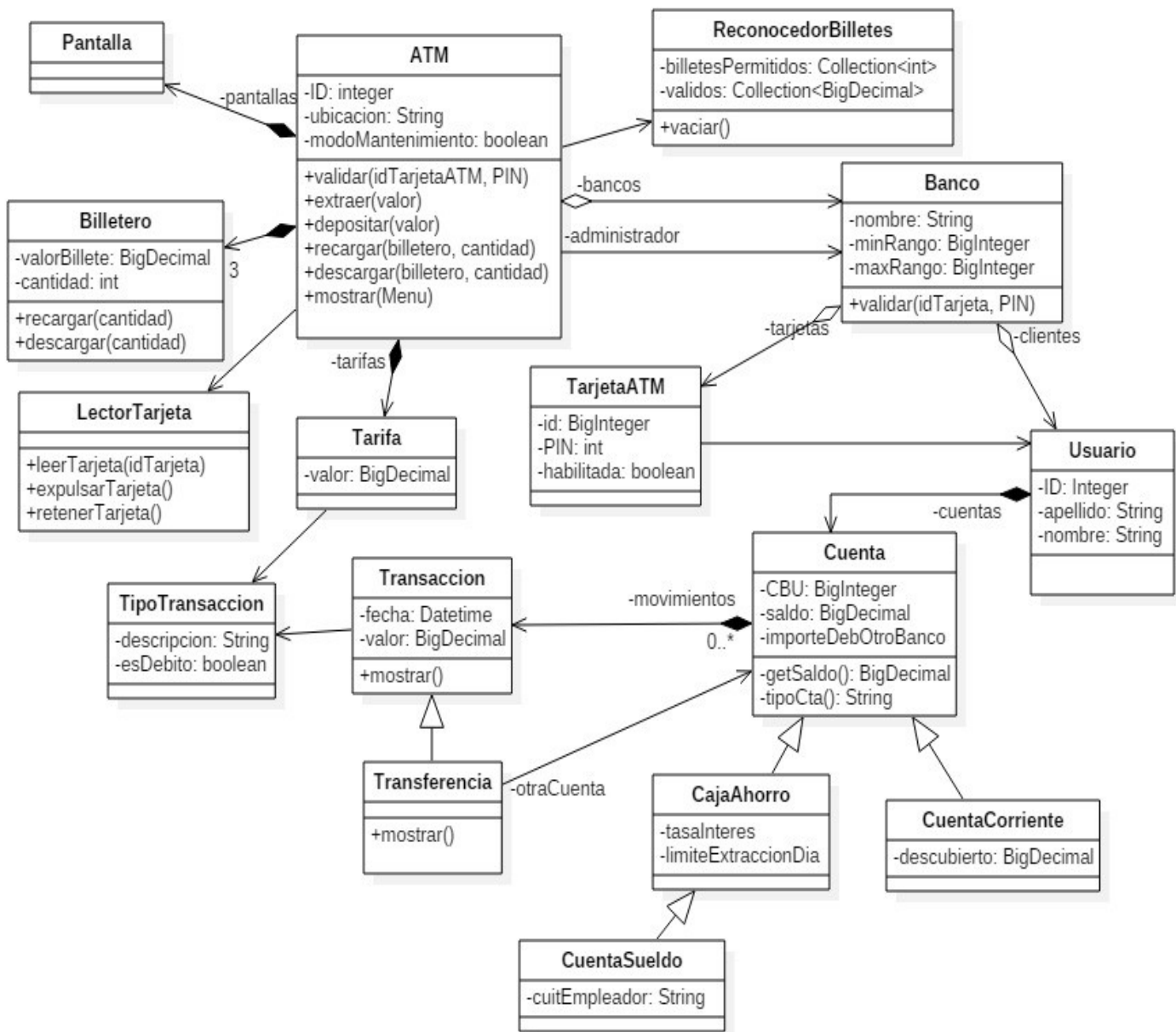
mes sin cargo. A partir de la cuarta extracción, se le debitarán \$15 por cada transacción. El cargo mensual por mantenimiento de cuenta es de \$300.

- Nota: Los valores de los distintos cargos se obtendrán de una lista de tarifas
- Un cliente puede tener más de una Cuenta. Si tuviera más de una, el usuario deberá seleccionar la Cuenta sobre la que trabajará previo a realizar cualquier operación.
- La operación de “Consulta de Movimientos” sólo podrá ser solicitada por clientes del *Banco de la Plaza*
- Al tratarse de un prototipo de ATM, el sistema no interactuará con el hardware del que dispondrá el ATM real:
 - En esta etapa de desarrollo del prototipo, no se dispone de un lector de tarjetas real. No obstante, se simularán las funciones realizadas por dicho dispositivo:
 - leer tarjeta (se seleccionará un Cliente de una lista),
 - retener tarjeta,
 - expulsar tarjeta.
 - El sistema no interactuará con un contador de billetes que entregue billetes reales, aunque sí simulará la función de entrega de dinero que es propia de dicho dispositivo.
 - El sistema no interactuará con un reconocedor de billetes real que reciba e identifique billetes reales, aunque sí simulará las funciones de recibir billetes y rechazar billetes no reconocidos
 - El sistema no emitirá tickets por impresora, aunque si simulará la emisión de los mismos por pantalla

Sugerencias y comentarios

- La **persistencia** se implementará mediante serialización o XML
 - Considerar:
 - El uso de las clases **Contenedoras** provistas por Java para la administración de listas.
 - Desarrollar formularios/procesos ajenos a la funcionalidad del ATM que implementen:
 - ABM de Clientes
 - Liquidación de intereses (para los clientes del propio Banco) (utilizar proceso agendado)
 - Liquidación de cargos de mantenimiento de cuenta (para los clientes del propio Banco) (utilizar procesos agendados)
 - Aplicar el lanzamiento de **excepciones** en las validaciones de las clases del dominio con el objetivo de desacoplarla de la Interfaz de Usuario.
-

Diagrama de Clases sugerido



Condiciones de Aprobación

- Conformar un **grupo** de 3 personas, de ser posible.
- Implementar la totalidad de la **funcionalidad** solicitada en el enunciado del problema.
- Aplicar indefectiblemente en la solución los siguientes conceptos de la Programación orientada a Objetos: **encapsulamiento**, **polimorfismo**, **herencia**.
- Cumplir estrictamente con el **cronograma de entregas**.
- En la Entrega Final:
 - presentar copia digital (CD, por mail, etc) de los **archivos fuentes** del proyecto.
 - los docentes podrán proponer la implementación del sistema con un **lote de datos** propuesto por la cátedra.
 - deberán estar presentes **todos** los integrantes del grupo. El integrante que no lo estuviera se considerará **fuera del grupo** y deberá realizar su propio TP.
- Otros **conceptos** que incidirán en la **aprobación** del trabajo práctico son:
 - Reutilización adecuada del código.
 - Eficiencia en los algoritmos (ej: búsquedas, ordenamientos)
 - Bajo acoplamiento entre interfaz y lógica de dominio
 - Validaciones de ingresos de datos y consistencia de la información.
 - Código prolijo, claro y correctamente comentado. (ej: nombres representativos, crear variables e instancias necesarias, sobrecargar métodos)
 - Amigabilidad de las interfaces de usuario
- *La **nota** del trabajo práctico es **individual**, basada en la participación en la resolución y defensa del trabajo práctico, y en los conocimientos conceptuales exhibidos en la entrega.*

Trabajo Práctico Grupal

Enunciado del problema

La Liga Continental de Fútbol requiere un software para gestionar su Campeonato Anual. La Liga nos brinda la siguiente información sobre el reglamento de dicha competencia:

- Participan 16 equipos divididos en 4 zonas de 4 equipos. En cada zona, cada equipo se enfrenta con los 3 restantes en una sola ocasión, no hay ida y vuelta. En cada partido de la zona, el ganador obtiene 3 puntos, el perdedor 0, y en caso de empate ambos equipos obtienen 1 punto.
- En la tabla de posiciones de cada zona, en el caso de empate en puntos la posición se define por mejor diferencia de gol, y en caso de empate por este criterio, por cantidad de goles a favor. Si persiste la igualdad se tiene en cuenta el resultado entre ambos equipos empatados.
- Los 2 primeros equipos de cada zona pasan a la siguiente instancia de eliminación directa (cuartos de final). Los ganadores de los cuartos de final pasan a la instancia de semifinal y los ganadores de esta instancia a la final del campeonato. Los cuartos de final y semifinales son partidos de ida y vuelta, o sea 2 partidos por cada instancia. La final es un partido único.
- El ganador de los partidos de ida y vuelta es el equipo que haya obtenido mayor puntaje contando ambos partidos. En caso de empate en puntos, se resuelve por mejor diferencia de gol, teniendo en cuenta que los goles de visitante valen doble. Si se mantiene la igualdad, el ganador se define por la ejecución de tiros penales.
- Si la final finaliza en empate en los 90 minutos, también se define con tiros penales.
- Para cada equipo se registra: nombre, país, posición en ranking continental, su plantel de 18 jugadores (2 arqueros, 6 defensores, 5 mediocampistas, 5 delanteros) y su DT.
- Para cada persona que participa del torneo (jugador, DT, referí) se registra su apellido y nombre, fecha de nacimiento, tipo y número de documento.
- Para los jugadores se registra su posición (arquero, defensor, mediocampista, delantero) y puntaje de valoración (a modo de los juegos de consola o cartas)
- Para los DT registrar nacionalidad y cantidad de títulos obtenidos en todos los niveles (nacional o internacional).
- Para los referís se registra nacionalidad y cantidad de años en el referato.
- Para cada partido se registra: fecha, los dos equipos que se enfrentan y el referí. El referí no podrá tener la misma nacionalidad que cualquiera de los equipos, a no ser que se enfrenten dos equipos de la misma nacionalidad.
 - Luego de cargarse la información se emitirán las identificaciones de todas las personas participantes del partido, con sus datos personales y su rol en el evento.
 - El resultado del partido surge de una simulación que tenga en cuenta las posiciones en el ranking de ambos equipos, el promedio de las valoraciones de los jugadores, y los títulos conseguidos por los DT. Además de estos datos, deberá existir un componente aleatorio (*random*) que defina el resultado.
 - Para los partidos de la fase inicial (zonas) se deberá mostrar el estado de la tabla de posiciones antes y después de registrado el resultado. Para cada equipo, mostrar: puntos, partidos jugados, ganados, empatados y perdidos, goles a favor y en contra, diferencia de gol.
 - En los partidos de la fase final (de cuartos de final hasta la final) la simulación se realizará en 2 pasos: en primer lugar, el resultado de los 90 minutos, y luego, de corresponder, el de

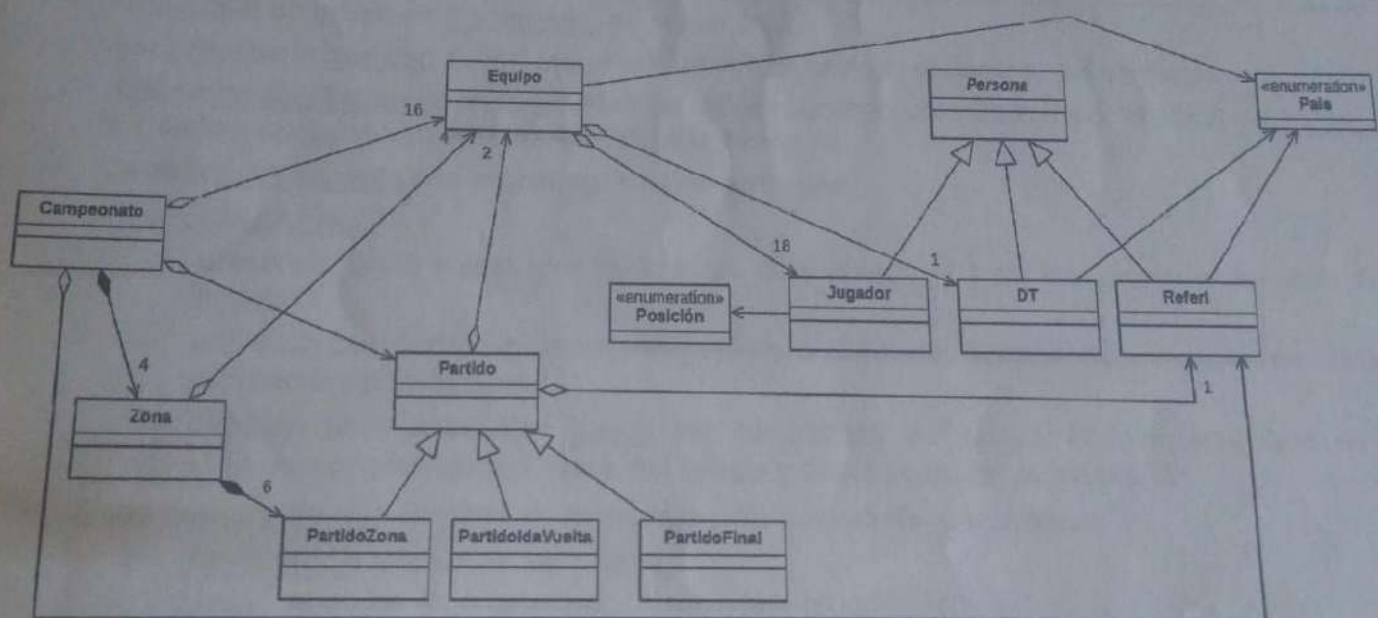
la ejecución de tiros penales. En todos los casos, indicar el criterio que haya determinado al ganador.

- En cualquier momento se podrá emitir los siguientes reportes:
 - Listado alfabético de los equipos mostrando edad promedio de sus jugadores, edad y nacionalidad de su DT, efectividad (porcentaje de puntos obtenidos sobre puntos posibles).
 - Ranking de referís por cantidad de partidos dirigidos en el campeonato. Indicar para cada uno la cantidad de años en el referato, y al final del listado el promedio de los mismos.
 - Listado de jugadores de determinada posición seleccionada por el operador (arquero, defensor, mediocampista, delantero) mostrando toda la información disponible del mismo. En el caso de los arqueros, mostrar la cantidad de Goles en Contra que recibió su equipo y el promedio de gol recibido por partido.

Sugerencias y comentarios

- La carga de los datos de los equipos, sus integrantes y referís puede realizarse a partir de un archivo XML, archivo JSON, ó archivo de texto delimitado por comas u otro separador.
- La interfaz de usuario para la carga de los partidos puede implementarse a través de Swing o modo Consola.
- La persistencia se implementará mediante serialización o XML. Deberá ser posible salir del sistema, volver a ingresar y retomar el campeonato desde el estadio en el que se encontraba al momento de salir.
- Considerar:
 - El uso de las clases *Contenedoras* provistas por Java para la administración de listas.
 - Aplicar el lanzamiento de excepciones en las validaciones de las clases del dominio con el objetivo de desacoplarla de la Interfaz de Usuario.

Diagrama de Clases sugerido





Trabajo Práctico *ExtremeF1*

¡Prepárate para desafiar tus habilidades de planificación y estrategia en el emocionante mundo de las carreras de autos!

ExtremeF1 es un juego de simulación de carreras de autos en el que los jugadores pondrán en juego sus estrategias para tratar de obtener el mejor posicionamiento en cada una de las carreras que componen la temporada de F1.

Acerca de los jugadores

Existen dos tipos de jugadores: Reales (personas) y Simulados (autogestionados por la computadora). Al comienzo de la aplicación, deberá seleccionarse la cantidad de personas que jugarán y cada uno de ellos ingresará su nombre, su avatar/color y seleccionará un piloto de carreras y un automóvil.

De cada piloto, se conocen su nombre, nombre abreviado, su país de origen, la cantidad de carreras corridas y ganadas, la cantidad de campeonatos en los que participó, la cantidad de campeonatos en que fue campeón, la cantidad de pole positions logradas en su historia, el presupuesto con el que cuenta para la temporada (ALTO / MEDIO / BAJO), entre otras cosas.

También se conocen distintas características de manejo y de ejecución de maniobras del piloto (con valores entre 1 y 100, a mayor valor, mejor desempeño):

| | |
|---------------------------|--|
| Sobrepaso | Determina cuántas oportunidades de sobrepaso ve un conductor |
| Defensa de posición | Determina la eficacia con la que un conductor se defiende de los intentos de adelantamientos de otros pilotos |
| Clasificación | Mejora el rendimiento del conductor durante las sesiones de clasificación, lo que le permite obtener una mejor posición de inicio en la grilla de partida |
| Cuidado de los neumáticos | Una nota alta de gestión de neumáticos indica que la forma de manejo del piloto reducirá el desgaste de los neumáticos en las carreras |
| Largada | Una valoración de <i>inicio de carrera</i> más alta aumentará las estadísticas de sobrepaso y de defensa del corredor, durante la primera vuelta de la carrera |

Los jugadores reales usarán el auto seleccionado hasta el final del campeonato. En tanto, los jugadores simulados recibirán un auto al azar entre los autos no seleccionados por los jugadores reales.

Acerca de los autos

Del auto se guardan datos tales como el número, la marca, velocidad máxima, aceleración (0 a 100 km/h, en segundos), potencia (HP), peso del automóvil, consumo promedio de combustible (en litros cada 100 km). También se conocen distintas características de desempeño del auto (con valores entre 1 y 100, a mayor valor, mejor desempeño):

| | |
|-------------------------------|---|
| Performance en los sobrepasos | Determina cómo se desempeña el auto en aceleraciones rápidas y/o rectas prolongadas para sobrepasar a otros autos |
| Performance en las curvas | Determina cómo es el comportamiento del auto en las curvas |



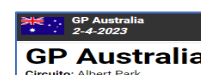
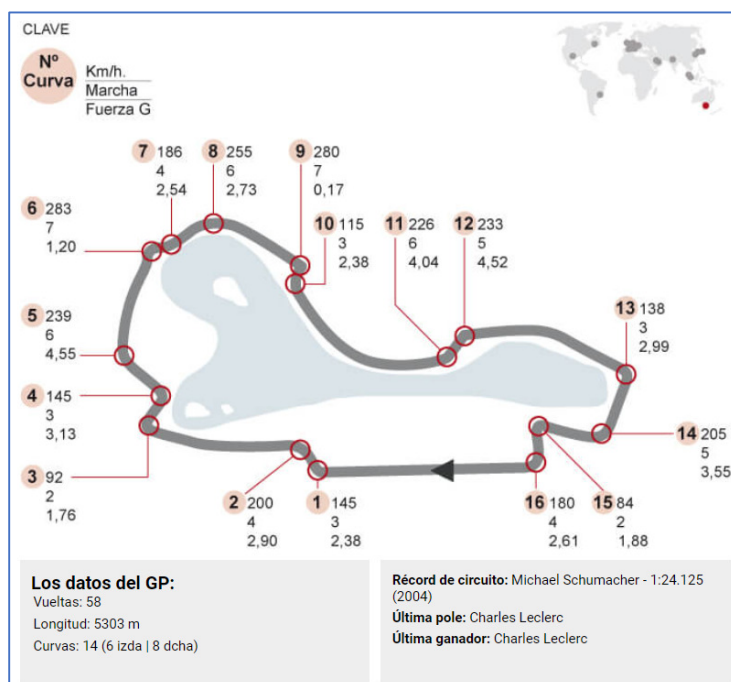
| | |
|------------|--|
| Fiabilidad | Determina qué tan confiable es el auto: A menor valor de este atributo, mayores probabilidades de que vaya a abandonar durante la carrera por desperfectos mecánicos |
|------------|--|

En cuanto a los neumáticos, el auto puede equiparse con 4 tipos de cubiertas diferentes. Los neumáticos serán elegidos por el jugador previo a la largada de la carrera y podrán ser cambiados en cualquier momento, durante la misma:

| | |
|--------|---|
| Soft | Proveen mayor agarre, mayores velocidades, pero menor durabilidad y mayor consumo. Son los neumáticos más rápidos pero también los que se desgastan más rápidamente, lo que probablemente obligue a realizar paradas técnicas para reemplazarlos. |
| Hard | Ofrecen la mayor durabilidad, pero con menor agarre y, por ende, es más difícil la maniobrabilidad a altas velocidades (es decir, el auto no podrá alcanzar la máxima velocidad) |
| Medium | Son una combinación entre los neumáticos Soft y Hard |
| Wet | Son los que mejor performance ofrecen en condiciones de lluvia. Ofrecen gran durabilidad pero sólo si la pista está mojada, en pista seca se rompen rápidamente. |

Acerca de las Circuitos

El campeonato consta de una carrera por cada circuito cargado en la aplicación. Del circuito se conoce el nombre, el país al que pertenece, la longitud de la pista (en metros), la cantidad de vueltas, la infografía del circuito, el récord de la vuelta más rápida.



Fuente: <https://www.elmundo.es/deportes/formula-1/calendario.html>



A los efectos de simular la carrera, se guardarán también como atributos del circuito, la cantidad de zonas de sobrepaso y la cantidad de curvas. Todos estos atributos serán tenidos en cuenta (junto con los propios de cada auto y piloto) para determinar en cada vuelta, los tiempos de vuelta de cada corredor.

Acerca de las Carreras

Cada carrera se corre en una fecha y en un circuito determinado, habiendo una ronda de clasificación y la competencia propiamente dicha.

Ronda de Clasificación

Previo a la carrera, se simulará la ronda clasificatoria, lo que determinará las posiciones de los pilotos en la grilla de partida. Se calculará el tiempo de clasificación para cada piloto teniendo en cuenta las características de la pista y los atributos del conductor y del auto, más alguna componente aleatoria que permita potenciar o degradar tales atributos a los efectos de lograr variabilidad en los resultados. La ronda clasificatoria supone condiciones climáticas ideales, óptima condición de todos los autos e igual configuración de neumáticos para todos los vehículos. Es decir, que no hay nada en especial que el piloto pueda hacer para obtener una mejora en la clasificación, la clasificación dependerá de los atributos que quieran tomarse en cuenta a la hora de simular los tiempos de clasificación junto con el factor aleatorio antes mencionado.

Competencia

Al momento de la carrera, los pilotos conocerán las condiciones climáticas al comienzo de la carrera y las esperadas durante la carrera. Esas condiciones están compuestas de un par conformado por la *condición climática* (Soleado / Nublado / Lluvioso / Niebla) y la *temperatura*.



Previo a la largada, cada jugador deberá especificar su estrategia inicial, teniendo en cuenta 2 aspectos clave que influirán en su desempeño en la pista:

- Combustible: Deberán especificar la cantidad de combustible con que iniciarán la carrera (1/4 tanque | ½ tanque | ¾ tanque | tanque lleno). Esta decisión es relevante ya que llevar demasiado combustible puede ralentizar el auto, pero llevar poco puede ocasionar la necesidad de realizar paradas adicionales en boxes.
- Neumáticos: Los jugadores deberán elegir el tipo de neumáticos con que iniciarán en la carrera.

En el caso de los jugadores simulados, ambas selecciones serán aleatorias (excepto en el caso de los neumáticos, dado a que, si el clima es lluvioso, los neumáticos iniciales serán los específicos para lluvia).

Toma de decisiones durante la carrera

Una vez iniciada la carrera, el corredor visualizará en pantalla en todo momento el número de vueltas que lleva dadas, su posición en la carrera y en el campeonato y también la posición en la



carrera y el campeonato de los otros corredores. Esta información es importante porque saber en qué posición va un contrincante en la carrera/campeonato, puede hacer que se adopte una estrategia más conservadora o más agresiva en la carrera.

Un tablero mostrará al jugador cuál es el estado de su auto: en todo momento conoce el estado de sus neumáticos (en %), de su motor (en %) y del combustible (expresada en cantidad de vueltas de autonomía). Si una condición (neumáticos, motor o combustible) llega a 0, se producirá el abandono de la competencia por parte del jugador, debiendo registrarse el motivo del abandono y los kilómetros recorridos hasta el momento del abandono (los kms recorridos determinan la posición en la carrera de los pilotos que abandonaron).

El abandono de la carrera podrá evitarse si el piloto realiza una parada en boxes antes de romper neumáticos, quedarse sin combustible o romper el motor. Para ello, el jugador tendrá una opción en pantalla para indicar que quiere hacer una parada técnica, especificando qué acciones quiere que se hagan a su auto: reemplazar neumáticos y/o reponer combustible y/o reacondicionar el motor. Hay que tener en cuenta que cada una de esas opciones incrementará el tiempo de detención en el box. Y también, se sabe que, a mayor deterioro del motor, mayor será el tiempo de detención.

Existen 3 modos de manejo que el piloto podrá adoptar en cualquier momento:

- **Rápido:** máxima velocidad y máxima exigencia del motor, de los neumáticos y de consumo de combustible
- **Conservador:** Velocidad reducida, cuidando el estado del motor, de los neumáticos y/o del combustible
- **Moderado:** Modo intermedio, entre el modo rápido y el conservador

Es decir, el modo de manejo seleccionado afecta las condiciones del auto: a mayor velocidad, más rápido se degradarán los neumáticos y el motor y mayor será el consumo de combustibles.

Cambios climáticos durante la carrera

Durante la carrera pueden ocurrir cambios en las condiciones climáticas y la temperatura, lo cuál podrá afectar la performance del auto. Por ejemplo:

- Correr en condiciones de lluvia con neumáticos que no son de lluvia (o a la inversa), afecta negativamente a la velocidad
- Temperaturas extremas (por encima de los 33°C) aceleran enormemente el desgaste de los neumáticos.

En cualquier caso, el piloto podrá optar por ingresar a boxes cuando lo requiera y acondicionar el auto de la forma que considere más conveniente.

Finalización de la carrera - Puntajes

La carrera finalizará una vez que el último auto en carrera complete todas las vueltas. La posición alcanzada en la carrera determinará el puntaje obtenido. A modo de ejemplo, si del campeonato participan 12 corredores, el ganador de la carrera recibe 12 puntos; el segundo, 11 puntos; el tercero, 10 puntos; ...; el undécimo, 2 puntos y el duodécimo, 1 punto.



Ganador del juego

Al correrse la última carrera y finalizar la temporada, el corredor que obtenga mayor puntaje general contabilizando todas las carreras, será el campeón.

Reportes

En cualquier momento se podrán emitir los siguientes **reportes**:

- **Estadística de los pilotos:** Grilla mostrando en las columnas la posición en el campeonato, nombre del piloto y, para cada carrera de la temporada, la posición en la carrera y en la ronda clasificatoria. Se debe poder ordenar la grilla por posición en el campeonato o por orden alfabético
- **Listado de pilotos** conteniendo Nombre, nacionalidad, cantidad de pole-positions en el campeonato, cantidad de carreras ganadas, cantidad de podios, cantidad de vueltas más rápidas realizadas, cantidad de abandonos, cantidad de carreras finalizadas, cantidad de detenciones en boxes.
- **Estadísticas de una carrera:** Dada una carrera, informar posiciones de los pilotos y para cada uno de ellos, detallar tiempo total y diferencia de tiempo respecto del ganador. Si el piloto abandonó, no informar tiempos sino indicar en qué vuelta se produjo el abandono.

En otra grilla, mostrar como fue el resultado de cada vuelta:

Vuelta 1: Piloto posición 1, Piloto posición 2, Piloto posición 3 ... Piloto posición N.

Vuelta 2: Piloto posición 1, Piloto posición 2, Piloto posición 3 ... Piloto posición N.

...

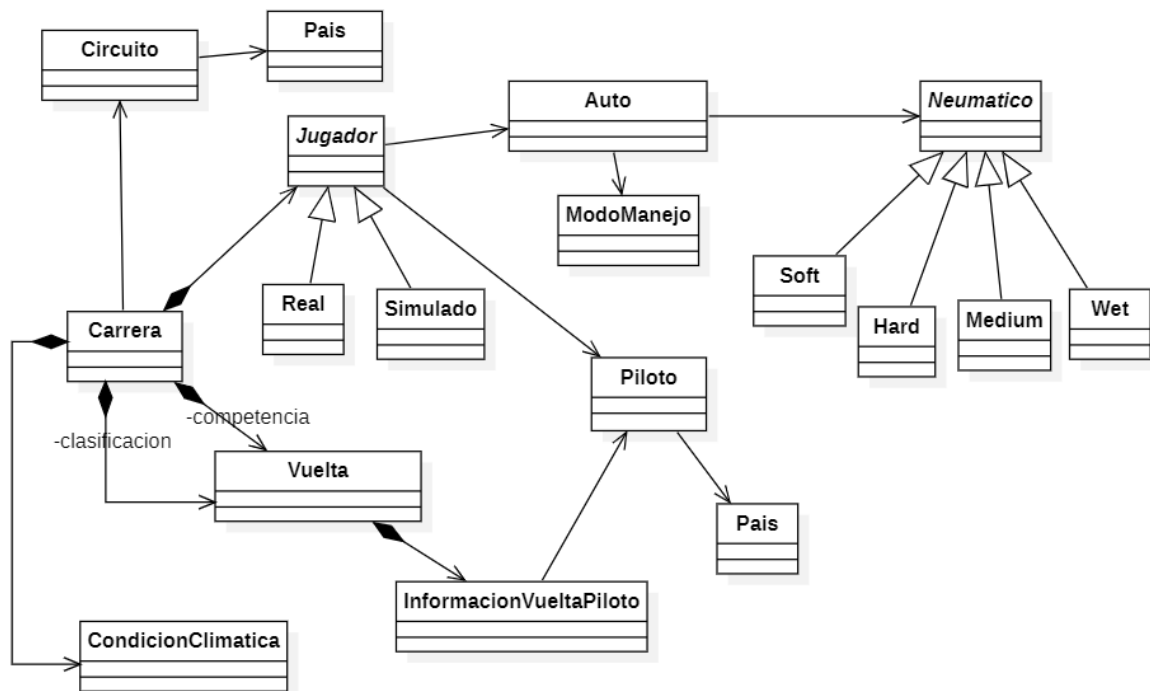
Vuelta N: Piloto posición 1, Piloto posición 2, Piloto posición 3 ... Piloto posición N.

Resaltar de alguna manera en cada vuelta, al piloto más rápido de la vuelta.

Sugerencias y comentarios

- La **carga** de los **datos** de los corredores, autos, circuitos y cronograma de carreras se realizarán a partir de un archivo XML
- La **interfaz de usuario** debe implementarse a través de Swing.
- La **persistencia** se implementará mediante serialización o XML. Deberá ser posible salir del sistema, volver a ingresar y retomar el campeonato desde el estado en el que se encontraba al momento de salir.
- Considerar:
 - El uso de las clases **Contenedoras** provistas por Java para la administración de listas.
 - Aplicar el lanzamiento desde las clases del dominio de **excepciones** en las validaciones y/o la notificación de eventos con el objetivo de desacoplar el dominio de la de la Interfaz de Usuario.

Diagrama de Clases sugerido (sólo clases del dominio)



Condiciones de Aprobación

- Conformar un **grupo** de 3 personas, de ser posible.
- Implementar la totalidad de la **funcionalidad** solicitada en el enunciado del problema.
- Desarrollar el proyecto utilizando la metodología ágil de desarrollo *Scrum*
- Aplicar indefectiblemente en la solución los siguientes conceptos de la Programación Orientada a Objetos: **encapsulamiento, polimorfismo, herencia**.
- Cumplir estrictamente con el **cronograma de entregas**.
- **Uso de idioma inglés**
 - La nomenclatura de *Packages, Clases, atributos, métodos, variables y documentación* debe hacerse en idioma inglés
 - Los nombres de las tareas que conforman el backlog de tareas / sprints, sus descripciones, comentarios, etc., deben redactarse en idioma inglés
- En la Entrega Final:
 - presentar copia digital (por mail, tarea aula virtual, acceso a *git*, etc.) de los **archivos fuentes** del proyecto.
 - los docentes podrán proponer la implementación del sistema con un **lote de datos** propuesto por la cátedra.
 - deberán estar presentes todos los integrantes del grupo. El integrante que no lo estuviera se considerará **fuera del grupo** y deberá realizar su propio TP.
- Otros **conceptos** que incidirán en la **aprobación** del trabajo práctico son:
 - Reutilización adecuada del código.



- Eficiencia en los algoritmos (ej: búsquedas, ordenamientos)
 - Bajo acoplamiento entre interfaz y lógica de dominio
 - Validaciones de ingresos de datos y consistencia de la información.
 - Código prolijo, claro y correctamente comentado. (ej: nombres representativos, crear variables e instancias necesarias, sobrecargar métodos)
 - Amigabilidad de las interfaces de usuario
-
- La **nota** del trabajo práctico es **individual**, basada en la participación en la resolución y defensa del trabajo práctico, y en los conocimientos conceptuales exhibidos en la entrega.

Enunciado del Problema – Juego Racer

Racer es un juego de tablero multijugador. Es un juego que tiene componentes de azar y también de conocimiento, pues se premia a los jugadores que contestan preguntas correctamente, tanto para evitar penalizaciones como también para avanzar más rápidamente hacia la meta. Las características del juego hacen que se mantenga el suspenso de quién resultará ganador hasta el último momento.

Dado que la idea es que puedan participar jugadores de distintas edades y niveles de conocimiento, habrá dos tipos de jugadores: *novatos* y *avanzados*. Al iniciar el juego, deberá indicarse la cantidad de jugadores que participarán en la partida (2 a 4 jugadores)

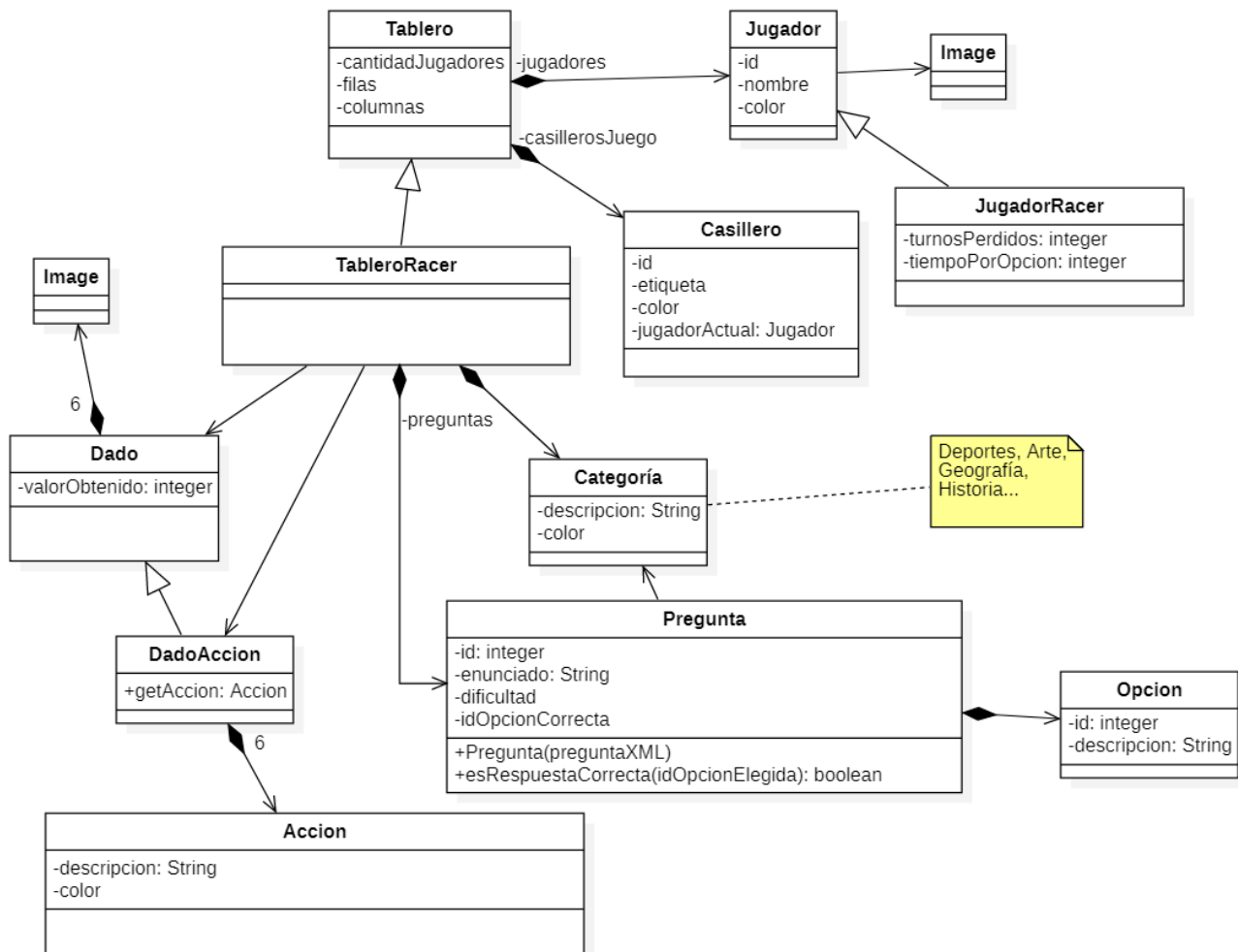
Reglamento del Juego

- Cada jugador elige la ficha del color de su preferencia, una imagen/icono de su preferencia, ingresa su nombre y su nivel (novato o experto). Su ficha se coloca en un casillero de Punto de Partida.
- Existen 2 dados diferentes:
 - o Dado numérico: el tradicional dado con valores de 1 a 6.
 - o Dado de *Acciones*: Cada cara está identificada con un color diferente, cuyos significados son:
 - Rojo: Avanza el valor obtenido en el dado numérico, pero pierde el próximo turno (no podrá lanzar los dados en el siguiente turno)
 - Azul: Exhibe una pregunta al jugador. Duplica casillas a avanzar si contesta bien. Si contesta mal, no avanza y pierde el próximo turno.
 - Naranja: El jugador que lanzó el dado no avanza ni retrocede. El jugador de “la derecha” (el del turno siguiente) deberá responder una pregunta. Si contesta bien, dicho jugador avanza el número obtenido en el dado numérico. Si contesta mal, dicho jugador retrocede esa cantidad de casillas.
 - Amarillo: Avanza si contesta bien. Si contesta mal, no avanza.
 - Verde: Avanza directamente la cantidad de casillas indicada por el dado numérico.
 - Fucsia: Avanza si contesta bien. Retrocede si contesta mal.
- Cada jugador, a su turno, lanza ambos dados. Hecho esto, se procede a analizar ambos dados para determinar la acción a realizar.
- Si un jugador al avanzar cae en la casilla que está ocupando otro jugador, este jugador regresa al punto de partida. Sin embargo, si un jugador al retroceder cae en la casilla que está ocupando otro jugador, quien regresa al punto de partida es el jugador que venía retrocediendo.
- El tablero se conforma de casillas de distintos tipos:
 - o Casillas de *punto de partida*: No pertenecen al tablero en sí, sino que alojan a los jugadores previo al inicio del juego, o bien cuando un jugador es desplazado por otro que cae en la casilla que dicho jugador estaba ocupando.
 - o 37 casillas neutrales: No implican ninguna acción adicional.
 - o 4 casillas de *pregunta*: El jugador que cae en ella (o bien el jugador siguiente) deberá contestar una pregunta. Si responde bien, avanza los números obtenidos en el dado numérico. Si contesta mal, queda en el mismo lugar. Estas casillas de preguntas deberán distribuirse aleatoriamente antes de cada partida. Deben estar separadas por una distancia mayor de 6, unas de otras. Ej:
 - Casillas 4, 11, 20 y 27, es correcto
 - Casillas 4, 12, 16 y 32, no es correcto

- Casilla de *llegada*: Es la última casilla. Quien llega a esta casilla, gana la partida. No es necesario obtener el número exacto con el dado numérico para llegar a esta casilla.
- Las preguntas se importarán desde un archivo XML. Son preguntas de tipo *selección múltiple* con las siguientes consideraciones:
 - La pregunta tendrá una *categoría* y un *grado de dificultad* asignado (1 a 5, donde 1 representa la menor complejidad y 5 la mayor complejidad).
 - La pregunta tendrá N opciones ($N \geq 3$), donde una sola de las opciones es la correcta.
 - En el caso de que la pregunta se exhiba a un jugador novato, se le mostrarán la mitad de las N opciones (Si la $N / 2$ no da un valor entero, redondear la cantidad hacia el número mayor). Dentro de las opciones a mostrar, deberá obviamente incluirse la opción correcta
 - Al jugador experto se le mostrarán todas las opciones
 - Las opciones deben mostrarse en orden aleatorio
 - Los jugadores de nivel novato contestarán preguntas de dificultad 1, 2 y 3 y dispondrán de $N * 15$ segundos como máximo para contestar.
 - Para jugadores de nivel avanzado, los niveles de dificultad serán 3, 4 y 5 y dispondrán de $N * 10$ segundos para contestar. Si el jugador no contesta en el tiempo otorgado se considera que la respuesta fue errónea.
 - Durante el juego, una pregunta sólo podrá exhibirse una vez durante la partida: Si una pregunta fue exhibida a un jugador, no puede volver a exhibirse al mismo o a otro jugador.

Sugerencias y comentarios

Diagrama de clases preliminar



Considerar:

- El uso de las clases contenedoras provistas por Java para administrar listas, conjuntos, etc.
- Disponer de una clase **TableroRacer** que centralice el control del juego y contenga listas de casillas, de jugadores, de preguntas, etc.
- Aplicar el lanzamiento de **excepciones / notificación de eventos** en las clases del dominio (Jugador, Casilleros, etc.) con el objetivo de desacoplarlas de la Interfaz de Usuario.

Requerimientos no funcionales:

- Para todos los formularios / ventanas de la aplicación, verificar que soporten adecuadamente cambios de tamaño y diferentes resoluciones de pantalla, utilizando **Alineaciones, Anclajes, Paneles**, etc
- Deberá exhibirse una pantalla de bienvenida al juego.
- Se deberá crear un único *JPanel* reutilizable que permita ingresar los datos de cada jugador.
- Cuando una partida finaliza, deberá darse la posibilidad de iniciar una nueva partida.
- Deberá ser posible finalizar la partida en cualquier momento
- Las clases, atributos, métodos así como la documentación deberán codificarse en idioma Inglés

Condiciones de Aprobación

- Conformar un **grupo** de 2 o 3 personas, de ser posible.
- Implementar la totalidad de la **funcionalidad** solicitada en el enunciado del problema.
- Aplicar indefectiblemente en la solución los siguientes conceptos de la Programación orientada a Objetos: **encapsulamiento, polimorfismo, herencia**.
- En la Entrega Final:
 - presentar los **archivos fuentes** del proyecto.
 - deberán estar presentes **todos** los integrantes del grupo.
- Otros **conceptos** que serán considerados positivamente en la aprobación del trabajo práctico son:
 - Reutilización adecuada del código.
 - Eficiencia en los algoritmos (ej: búsquedas, ordenamientos)
 - Bajo acoplamiento entre interfaz y lógica de dominio
 - Validaciones de ingresos de datos y consistencia de la información.
 - Código prolijo, claro y correctamente documentado. (ej: nombres representativos, crear variables e instancias necesarias, sobrecargar métodos, uso de *javadoc*)
 - Amigabilidad de las interfaces de usuario
- La **nota** del trabajo práctico es **individual**, basada en la participación en la resolución y defensa del trabajo práctico, y en los conocimientos conceptuales exhibidos en la entrega.

Trabajo Práctico: Simulador de Campeonato de Triatlón

1. Introducción

En este trabajo práctico se debe desarrollar un simulador de campeonato de triatlón utilizando los principios de la programación orientada a objetos. Este simulador deberá ser capaz de correr distintas carreras desarrolladas en diversas ciudades y modalidades (larga distancia, media distancia y distancia olímpica). Los atletas se agrupan en diferentes categorías, por género y edad.

La simulación deberá cumplir con el reglamento oficial del triatlón, en especial, en lo que se deriva de la temperatura del agua al momento de la prueba de natación, los tiempos de paso mínimos, los puestos de reaprovisionamiento, las sanciones por *drafting* en la etapa de ciclismo, etc.

Al finalizar la temporada de carreras, se deberá proclamar a los campeones de las distintas categorías.

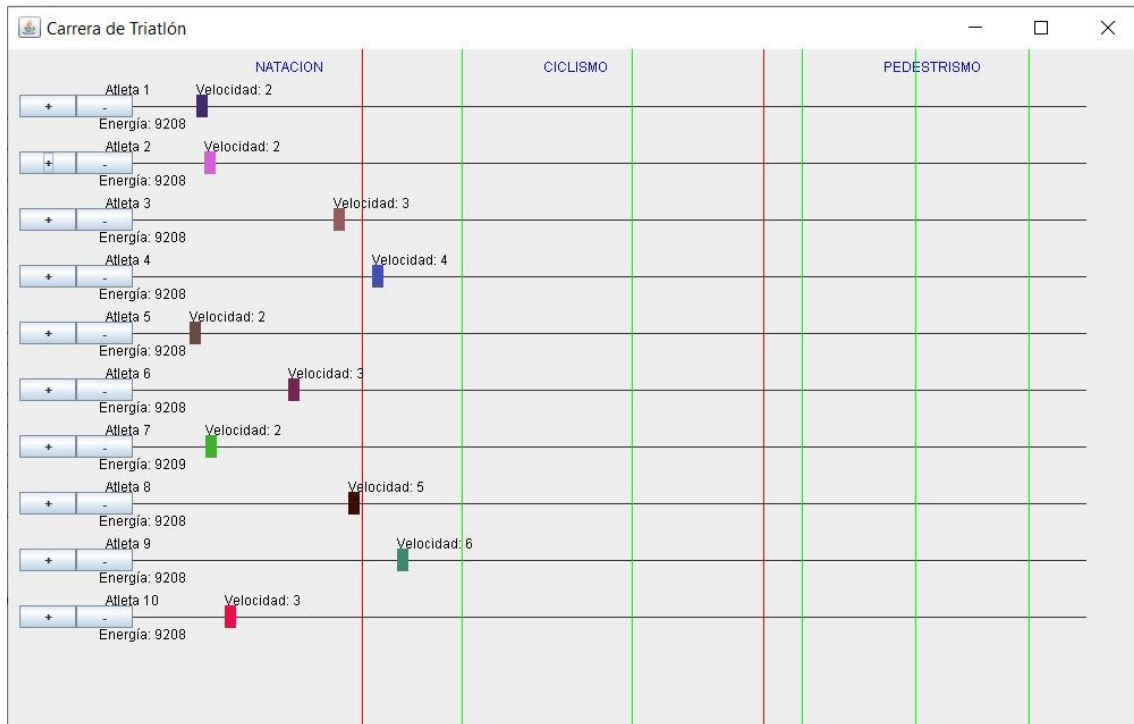
2. Requerimientos del proyecto

El objetivo del proyecto es desarrollar un simulador de campeonato de triatlón aplicando los principios de la programación orientada a objetos, integrando las técnicas y temas que se verán a lo largo del año (programación visual, persistencia, programación multihilos, uso de base de datos, etc.), utilizando el lenguaje de programación Java.

La aplicación debe simular el desempeño de cada atleta participante en la carrera. De los atletas se conocen determinadas capacidades físicas que hacen que algunos se destaquen más en una disciplina que en otra, que tengan mayor o menor resistencia a las condiciones climáticas (temperatura, sol, dirección y velocidad del viento), mayor o menor fortaleza psicológica, etc. La edad puede ser también un condicionante del desempeño, lo mismo que el género. También, como en la vida real, hay factores aleatorios que pueden favorecer o afectar el desempeño de un atleta así como también incidentes (por ej., pinchadura de un neumático de la bicicleta) que puedan retrasarlo.

El simulador debe mostrar en todo momento de una carrera, el tiempo de carrera y la condición climática actual. También debe mostrar información de los corredores y permitir filtrar por categoría, por género o la clasificación general.

Se necesitan ver gráficamente las posiciones de los corredores. A los fines de generar interacción en la aplicación, el usuario debe poder aumentarle/disminuirle la velocidad a un atleta, teniendo en cuenta que a mayor velocidad, mayor consumo de energía (si el atleta se queda sin energía antes de llegar a un puesto de aprovisionamiento, no puede seguir corriendo).



Prototipo de pantalla de una carrera.

Referencias : Líneas rojas delimitan disciplinas. Líneas verdes, representan puestos de aprovisionamiento

a. Sobre las Carreras

El triatlón es una prueba combinada que consiste en natación, ciclismo y pedestrismo, estando enlazados los tramos por dos transiciones para el cambio de disciplina (*T1*, para natación a ciclismo, y *T2*, para ciclismo a pedestrismo).

Hay 4 modalidades de carreras determinadas por las distancias de cada disciplina:

- *Larga distancia*: 3.8 km de natación, 180 km de ciclismo y 42.2 Km de pedestrismo
- *Media distancia*: 1.9 km de natación, 90 km de ciclismo y 21.1 km de pedestrismo
- *Distancia olímpico*: 1.5 km de natación, 40 km de ciclismo y 10 km de pedestrismo
- *Sprint*: 0.75 km de natación, 20 km de ciclismo y 5 km de pedestrismo

Las distintas carreras que componen el campeonato se corren en distintas fechas y ciudades, cada una con una distancia determinada. No se corren diferentes distancias en la misma fecha y ciudad.

Durante una carrera, la aplicación debe variar aleatoriamente las condiciones climáticas teniendo en cuenta que a partir de los 23° C, el desgaste energético de los atletas es significativamente mayor. Los organizadores de las carreras instalan puestos de aprovisionamiento que ofrecen a los atletas la posibilidad de disponer de agua, bebidas con carbohidratos, frutas, barras de cereales, etc.

b. Atletas

De los triatletas se conocen su apellido y nombre, DNI, nacionalidad, fecha de nacimiento, género y categoría: *competición* o *amateur*. Los atletas que se inscriben en la categoría *competición*, a diferencia de los *amateurs*, reciben premios económicos y clasifican a competiciones internacionales según la posición lograda. Las categorías se subdividen por género.

Dentro de la categoría *competición* se distinguen 3 subcategorías:

- Elite
- Sub 23 (U23) (19 a 23 años)
- Júnior (15 a 18 años)

En tanto que a los deportistas *amateurs* se los agrupa en categorías por edad, cada 5 años, desde los 19 años:

| | |
|----------------|------------------|
| • 19 – 24 años | • 50 – 54 años |
| • 25 – 29 años | • 55 – 59 años |
| • 30 – 34 años | • 60 – 64 años |
| • 35 – 39 años | • 65 – 69 años |
| • 40 – 44 años | • 70 – 74 años |
| • 45 – 49 años | • 75 años en más |

c. Acerca del Reglamento y las penalizaciones

El simulador debe aplicar el reglamento oficial del triatlón, incluyendo sanciones por infracciones. Se deben implementar mecanismos para detectar y penalizar a los participantes que incurran en estas infracciones.

Tiempos máximos

Las carreras de *larga distancia* fijan un tiempo máximo para llegar a T2 (es decir, para cubrir los tramos de natación y ciclismo) y un tiempo máximo para arribar a la meta final. Si un atleta no arriba a T2 antes del tiempo máximo, deberá abandonar la competencia. Lo mismo ocurrirá si se supera el tiempo máximo para llegar a la meta.

Reglamento para la Natación

En cuanto al reglamento, para la natación hay normas que indican la obligatoriedad o prohibición del uso de trajes de neoprene, en función de la temperatura. También se establece un tiempo máximo permitido en el agua: si el atleta supera ese tiempo máximo en la etapa de natación, debe abandonar la prueba por razones de su seguridad física.

7.2. Temperatura y tiempo máximos.

7.2.1 Atletas Elite y U23 en Copas Nacionales, Regionales y Campeonatos Argentinos.

Limites superiores e inferiores para el uso de neoprene:

| DISTANCIA DE NATACION | PROHIBIDO ARRIBA DE | LIBERADO ARRIBA DE: | OBLIGATORIO DEBAJO DE | MAXIMO TIEMPO PERMITIDO EN EL AGUA |
|-----------------------|---------------------|---------------------|-----------------------|------------------------------------|
| Hasta 750m | 20 grados C | 15 grados C | 14 grados C | 20 min. |
| 1500m | 20 grados C | 15 grados C | 14 grados C | 40 min. |
| 1501-3000m | 22 grados C | 15 grados C | 15 grados C | 1h 30 min. |
| 3001-4000m | 23 grados C | 16 grados C | 16 grados C | 2h 20 min. |

7.2.2 Atletas de Grupos de Edades Copas Nacionales, Regionales y Campeonatos Argentinos.

Limites superiores e inferiores para el uso de neoprene:

| DISTANCIA NATACION | LIBERADO ARRIBA DE: | OBLIGATORIO ABAJO DE | MAXIMO TIEMPO PERMITIDO EN EL AGUA |
|--------------------|---------------------|----------------------|------------------------------------|
| Hasta 750m | 15 grados C | 14 grados C | 35 min. |
| 1500 m | 15 grados C | 14 grados C | 1h 10 min. |
| 1501-3000m | 16 grados C | 15 grados C | 2h 00 min. |
| 3001-4000m | 17 grados C | 16 grados C | 2h 20 min. |

Fuente: Reglamento de Competencia de la Federación Argentina de Triatlón (www.triatlon.org.ar)

La determinación del uso de traje de neoprene se hará 1 hora antes de la largada en función de la temperatura del agua y será claramente notificada a todos los atletas en una pantalla a tal efecto.

Reglamento para el Ciclismo

La prueba ciclística puede o no permitir *drafting*. Si no lo permite, el simulador deberá detectar estas situaciones y penalizar al atleta que lo cometa. Por cada penalización se suma 1 minuto al tiempo hecho por el atleta.

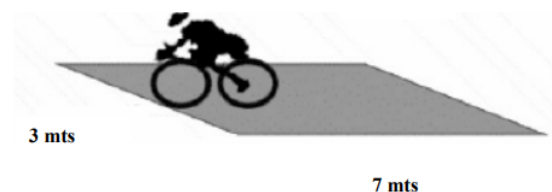
8.5.2 ZONA DE DRAFT

La zona de draft en eventos sin drafting o drafting ilegal, es un rectángulo de siete (7) metros de largo por (3) metros de ancho (*diagrama 1*), la cual rodea a todo atleta y el cual deberán rodear todos y cada uno de los atletas. El extremo trasero de la rueda trasera de la bicicleta será lo que define el inicio de los siete (7) metros del rectángulo hacia atrás.

El atleta tiene un máximo de 15 segundos para atravesar una zona de draft de otro atleta.

Diagrama 1. Distancia de la zona de drafting para todos los atletas

ta.



8.5.4 DEFINICIÓN DE REBASAR O SUPERAR

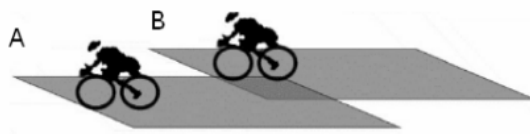
Un atleta es rebasado o superado cuando la rueda delantera de otro atleta ha rebasado o superado la suya.

Una vez rebasado o superado, el atleta inmediatamente deberá salirse de la nueva zona de draft, teniendo 15 segundos para realizarlo.

Los atletas deberán mantenerse a un lado del circuito, y no crear situaciones de bloqueo.

Se denomina bloquear cuando un atleta que se encuentra detrás o de lado de otro atleta, no puede pasar debido a que el atleta que conduce o lidera utiliza en forma innecesaria el espacio disponible, no dejando lugar suficiente en el circuito.

Diagrama 2 Drafting



A esta sobrepasando a B y B esta haciendo drafting con el líder A.
B debe salir de la zona de draft de A antes que termine de

Diagrama 4 *Drafting y no drafting*



A y B mantienen una adecuada separación.
C no está intentando rebasar.

Fuente: Reglamento de Competencia de la Federación Argentina de Triatlón (www.triatlon.org.ar)

d. Resultados de la carrera

Al finalizar cada carrera, se deben calcular y mostrar los resultados, incluyendo los tiempos de cada etapa y el tiempo total de la carrera. Se deben generar clasificaciones por categoría y en general, mostrando a los atletas, sus posiciones y tiempos.

También debe mostrarse las posiciones de los atletas en el campeonato global.

e. Condiciones climáticas

Los datos de condiciones climáticas que pueden presentarse durante una carrera se cargarán desde una Base de Datos. La creación / edición / eliminación de registros de condiciones climáticas en la Base de Datos es parte del Trabajo Práctico que deben desarrollar los alumnos.

Se considerarán distintos tipos de condición climática que pueden afectar el rendimiento de un atleta, considerando rangos de valores. Para cada condición se almacenará entonces una descripción, un valor mínimo (cota inferior), un valor máximo (cota superior) y un porcentaje de desgaste energético adicional que esa condición genera en el atleta (si el porcentaje es negativo, indica que en vez de provocar desgaste, mejora el rendimiento del atleta, por ejemplo: contar con viento a favor).

- **Temperatura:** El calor o el frío extremos pueden afectar la capacidad del cuerpo para regular la temperatura interna. En condiciones de calor intenso, el riesgo de deshidratación y golpe de calor aumenta, lo que puede llevar a una disminución en el rendimiento y aumentar el riesgo de lesiones. Por otro lado, en condiciones de frío extremo, los músculos pueden volverse rígidos y la capacidad de mantener la temperatura corporal puede ser un desafío.
- **Humedad:** La humedad alta puede dificultar la evaporación del sudor, lo que hace que el cuerpo se caliente más rápidamente y aumente el riesgo de deshidratación y agotamiento por calor. Además, la humedad puede afectar la capacidad del cuerpo para enfriarse de manera eficiente durante el ejercicio, lo que puede llevar a un mayor esfuerzo percibido y una disminución en el rendimiento.
- **Viento:** El viento puede tener un impacto significativo en el rendimiento del ciclista, especialmente en tramos largos y expuestos. Un viento en contra puede hacer que sea más difícil mantener la velocidad y aumentar la fatiga muscular, mientras que un viento a favor puede proporcionar un impulso adicional pero también puede dificultar el control de la bicicleta. En la sección de carrera a pie, el viento lateral puede afectar la estabilidad y la eficiencia del corredor.

Ejemplos:

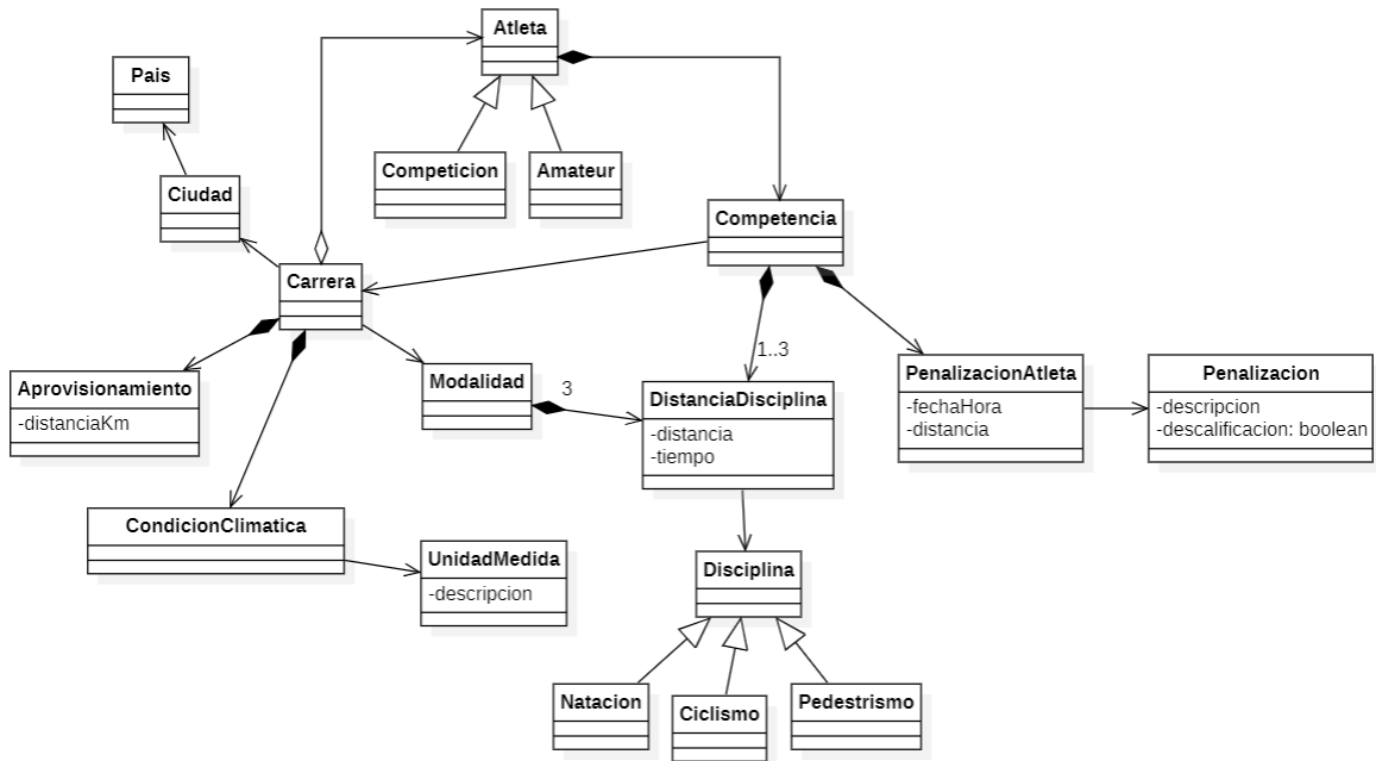
| Descripción | Unidad medida | Cota inferior | Cota superior | % desgaste adicional | | |
|----------------------|---------------|---------------|---------------|----------------------|----------|-------------|
| | | | | Natación | Ciclismo | Pedestrismo |
| Temperatura muy baja | °C | -10 | 5 | 15 | 2 | 8 |
| Temperatura normal | °C | 6 | 22 | 0 | 0 | 1 |
| Temperatura cálida | °C | 23 | 34 | 0 | 10 | 15 |
| Alta temperatura | °C | 35 | 50 | 0 | 15 | 20 |
| Viento a favor | Km/h | 0 | 30 | -1 | -30 | -4 |
| Viento en contra | Km/h | -1 | -30 | 1 | 35 | 5 |
| ... | ... | ... | ... | ... | ... | ... |

f. Reportes

En cualquier momento se podrán emitir los siguientes **reportes**:

- **Estadística de los atletas:** Grilla mostrando en las columnas la posición en el campeonato, nombre y, para cada carrera de la temporada, la posición en la carrera y los tiempos en cada disciplina. Se debe poder ordenar la grilla por posición en el campeonato o por orden alfabético
- **Listado de atletas** conteniendo Nombre, nacionalidad, cantidad de etapas ganadas en cada disciplina en el campeonato, cantidad de carreras ganadas, cantidad de abandonos, cantidad de carreras finalizadas.
- **Estadísticas de una carrera:** Dada una carrera, informar posiciones de los atletas y para cada uno de ellos, detallar tiempo total y diferencia de tiempo respecto del ganador. Si el atleta abandonó, no informar tiempos sino indicar en qué disciplina abandonó y qué distancia había recorrido hasta el momento del abandono

3. Diagrama de Clases sugerido (sólo clases del dominio)



4. Entregables

1. **Código Fuente:** Se deberá entregar el código fuente del simulador, escrito en lenguaje Java.
2. **Documentación:** Se requiere una documentación técnica que explique la estructura del código, las clases y métodos utilizados, así como cualquier decisión de diseño importante.
3. **Gestión del proyecto:** Se deberá tener constancia de la gestión del proyecto grupal donde haya evidencia del trabajo realizado por cada integrante del grupo. La gestión del proyecto debe hacerse empleando metodologías ágiles de desarrollo.
4. **Presentación del trabajo ante la cátedra y los compañeros de cursado:** Es necesario una presentación del trabajo que describa el proceso de desarrollo, los desafíos encontrados y cómo fueron abordados y resueltos desde el punto de vista técnico, así como cualquier mejora o funcionalidad adicional que hayan llevado adelante. Deberán exponerse evidencias de cómo se han aplicado los principios del paradigma de programación orientada a objetos.
5. **Desarrollo de competencias en inglés:** Dado que desde la asignatura se contribuye al desarrollo de competencias en idioma inglés, se requiere que los alumnos:
 - Realicen la gestión del proyecto, la documentación de las tareas, sprints, etc., en idioma inglés, lo mismo que la codificación de las clases y las interfaces de usuario de la aplicación.
 - En la Presentación del trabajo práctico se deben expresar opiniones y juicios, preparar respuestas guionadas y reforzar sonidos propios del idioma inglés

5. Fecha de Entrega

Durante el primer y segundo cuatrimestre habrá instancias de muestra obligatoria de avances del proyecto. La fecha de entrega del trabajo práctico tendrá lugar en el segundo cuatrimestre. Todas las fechas se especifican en el cronograma de la asignatura.

6. Requerimientos no funcionales

- La **carga** de los **datos** de los atletas y de las carreras a correr se realizarán a partir de un archivo XML
- La **interfaz de usuario** debe implementarse a través de Swing.
- La **persistencia** se implementará mediante serialización o XML. Deberá ser posible salir del sistema, volver a ingresar y retomar el campeonato desde el estado en el que se encontraba al momento de salir.
- Considerar:
 - El uso de las clases **Contenedoras** provistas por Java para la administración de listas, conjuntos, diccionarios, etc.
 - Aplicar el lanzamiento desde las clases del dominio de **excepciones** en las validaciones y/o la notificación de eventos con el objetivo de desacoplar el dominio de la de la Interfaz de Usuario.

7. Condiciones de Aprobación

- Conformar un **grupo** de 3 personas, de ser posible.
- Implementar la totalidad de la **funcionalidad** solicitada en el enunciado del problema.
- Desarrollar el proyecto utilizando la **metodología ágil** de desarrollo *Scrum*
- Aplicar indefectiblemente en la solución los siguientes conceptos de la Programación Orientada a Objetos: **encapsulamiento, abstracción, polimorfismo, herencia**.
- Cumplir estrictamente con el **cronograma de entregas**.
- **Uso de idioma inglés**, como se mencionó anteriormente
- En la Entrega Final:
 - presentar copia digital (por mail, tarea aula virtual, acceso a *git*, etc.) de los **archivos fuentes** del proyecto.
 - deberán estar presentes **todos** los integrantes del grupo. El integrante que no lo estuviera se considerará **fuera del grupo** y deberá presentar su propio TP.
- Otros **conceptos** que incidirán en la **aprobación** del trabajo práctico son:
 - Reutilización adecuada del código.
 - Eficiencia en los algoritmos (ej: búsquedas, ordenamientos)
 - Bajo acoplamiento entre interfaz y lógica de dominio
 - Validaciones de ingresos de datos y consistencia de la información.
 - Código prolijo, claro y correctamente comentado. (ej: nombres representativos, crear variables e instancias necesarias, sobrecargar métodos)
 - Amigabilidad de las interfaces de usuario
- La **nota** del trabajo práctico es **individual**, basada en la participación en la resolución y defensa del trabajo práctico, y en los conocimientos conceptuales exhibidos en la entrega. El uso de idioma inglés tendrá una ponderación específica en la Nota final. Oportunamente, se darán a conocer las rúbricas y resultados de aprendizajes esperados con el desarrollo de este TP.

Fuentes de información

- <https://www.triatlon.org.ar/img/reglamentos/reglamento-23.pdf>
- <https://triatletasenred.sport.es/actualidad/quien-tiene-los-records-mundiales-del-triatlon/>
- <https://www.t3team.cat/bloc/miscellania/tabla-conversion-velocidad-ritmo/>
- <https://www.t3team.cat/bloc/miscellania/calculadora-ritmos-natacion-ciclismo-carrera-triatlon/>
- <https://www.sport.es/labolsadelcorredor/sistemas-energeticos-deporte/#:~:text=Hablamos%20de%20sistema%20de%20fosf%C3%A1genos,actividad%20f%C3%ADsica%20y%20la%20intensidad.>
- <https://www.kynet.com.ar/los-pilares-basicos-de-la-nutricion-para-triatletas-de-ironman>
- <https://www.alimmenta.com/dietista-nutricionista-deportivo/dieta-para-triatletas/dieta-para-triatlon-olimpico/#:~:text=De%20forma%20general%2C%20se%20recomienda,0%2C460%2D1%2C150%20g%2FL.>



Programación B - UFASTA

Trabajo Práctico 2025 - Central de Monitoreo Urbano

1. Introducción

En este trabajo práctico se debe desarrollar un simulador de una Central de Monitoreo de una ciudad. Este simulador deberá ser capaz de controlar y visualizar el estado de distintas redes de dispositivos, tales como redes de semáforos, de cámaras de seguridad y de cámaras de fотомultas. Las cámaras y semáforos pueden sufrir fallos que deben ser detectados e informados a la Central de Monitoreo.

Los semáforos y cámaras de fотомulta, al detectar una infracción, reconocen la patente del automóvil infractor y envían la imagen a la Dirección de Tránsito, junto con los datos de la patente reconocida. En la Dirección de Tránsito se generan las multas para ser enviadas a los propietarios de los automóviles infractores.

Las cámaras de seguridad simularán la visualización en tiempo real de las imágenes que están capturando, pudiendo el operador notificar una situación anómala a quién corresponda: servicios médicos, bomberos, policía u Oficina de Tránsito.

El sistema deberá generar distintos tipos de informes a partir de los datos que se generan con el uso de la aplicación.

El objetivo del proyecto es desarrollar un simulador de central de monitoreo aplicando los principios de la programación orientada a objetos, integrando las técnicas y temas que se verán a lo largo del año, utilizando el lenguaje de programación Java. La aplicación deberá aplicar programación visual, programación multihilos, manejo de excepciones, notificación de eventos, persistencia, procesamiento de información desde archivos, conexión a base de datos, entre otras.

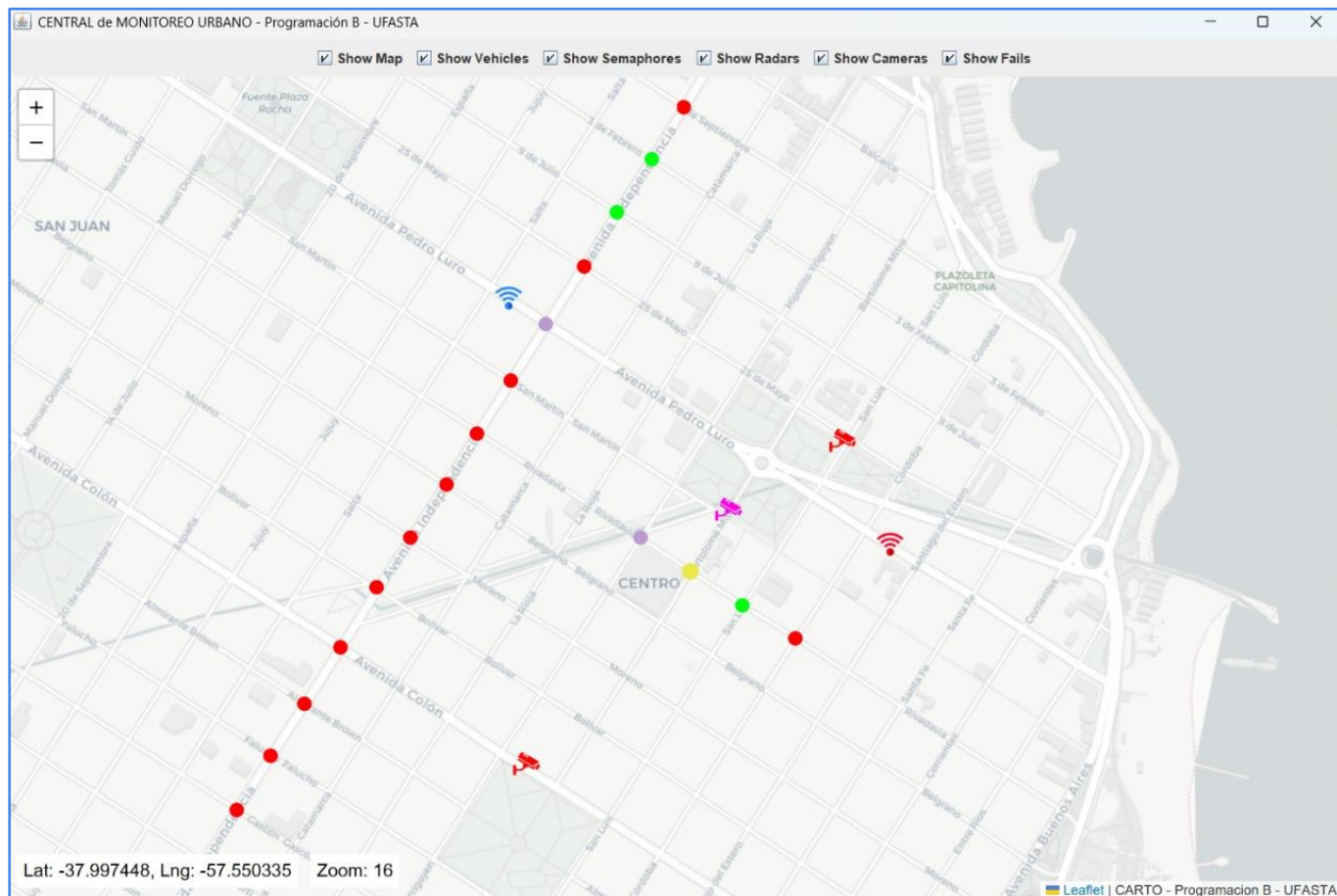




2. Requerimientos del proyecto

a. Mapa

La aplicación debe mostrar gráficamente el mapa de la ciudad y ubicar en él, los distintos dispositivos bajo control de la Central de Monitoreo.



b. Dispositivos

Cada dispositivo tiene una identificación, una ubicación y un estado. Además, cada dispositivo tiene información y funciones propias de su tipo, como se describe a continuación.

Semáforos

Cada intersección de calles que tenga semáforos estará regida por un *Controlador de semáforos*. Para reducir complejidad, inicialmente se trabajará con dos semáforos simples, sin giros a izquierda ni semáforos de peatones. De los dos semáforos, uno de ellos estará identificado como “semáforo principal”. Ejemplos:

- A lo largo de Av. Independencia, todos los semáforos que están sobre la Av. Independencia (intersecciones con Av. Colón, Brown, Falucho, Gascón, ...), son “principales”.
- A lo largo de Av. Colón, los semáforos sobre Av. Colón son los principales, excepto la intersección con Av. Independencia.
- En Catamarca y Gascón, el semáforo principal es el que está sobre la calle Gascón.





En el mapa se muestra gráficamente sólo el estado del semáforo principal: Rojo, amarillo o verde. En caso de error, se mostrará en el mapa un ícono de color violeta. En cualquier caso, al hacer click sobre una intersección se mostrará en detalle, el estado de cada semáforo de la intersección.



Según las luces de los semáforos, el Controlador notificará el estado de los mismos a la Central, acorde a la siguiente tabla de decisión:

Tabla de Decisión – Funcionamiento de Semáforos

| Semáforo A (principal) | Semáforo B | ¿Funcionamiento correcto? | Color Icono | Comentario |
|------------------------|------------|---------------------------|-------------|--|
| Rojo | Rojo | ✗ No | | Ambos detenidos: situación anómala |
| Verde | Verde | ✗ No | | Ambos avanzan: posible riesgo de colisión |
| Rojo | Verde | ✓ Sí | | Funcionamiento normal |
| Verde | Rojo | ✓ Sí | | Funcionamiento normal |
| Intermitente | Rojo | ✗ No | | Uno operativo, el otro en intermitente, posible riesgo de colisión |





| | | | | |
|--------------|--------------|------|--|--|
| Intermitente | Verde | ✗ No | | Uno operativo, el otro en intermitente, posible riesgo de colisión |
| Rojo | Intermitente | ✗ No | | Uno operativo, el otro en intermitente, posible riesgo de colisión |
| Verde | Intermitente | ✗ No | | Uno operativo, el otro en intermitente, posible riesgo de colisión |
| Intermitente | Intermitente | ☑ Sí | | Ambos en modo intermitente |
| Desconocido | Desconocido | ✗ No | | Sin respuesta, posible falla en el suministro eléctrico o la comunicación. |

El controlador guarda las duraciones, en segundos, de los distintos colores del semáforo principal: segundos de Rojo, segundos de Amarillo y segundos de Verde. También se configuran los segundos de “ambos Rojos” (para evitar colisiones entre el que está cruzando entre amarillo/rojo y el que está esperando el paso). Suponiendo que el semáforo principal es el A, se especifican las siguientes duraciones de dicho semáforo:

- Tiempo de Verde: 40 segundos
- Tiempo de Rojo: 30 segundos
- Tiempo de Amarillo: 4 segundos
- Tiempo de Ambos Rojos: 3 segundos

Un ciclo completo sería el siguiente:

| Ciclo | Semáforo A | Semáforo B | Duración | Observaciones |
|---------|------------|------------|----------|--|
| Ciclo 1 | | | 40 seg | |
| | | | 4 seg | |
| | | | 3 seg | No reportar como fallo (es para evitar colisiones) |
| | | | 30 seg | |
| | | | 4 seg | |
| | | | 3 seg | No reportar como fallo (es para evitar colisiones) |
| Ciclo 2 | | | 40 seg | |
| | ... | ... | ... | ... |





De cada *Controlador* de semáforos la Central conoce entre qué horarios la esquina funcionará en modo intermitente (si es que en algún momento lo hace) y le enviará la orden de cambiar a modo intermitente o normal, según corresponda.

Un semáforo puede cambiarse manualmente de modo intermitente a modo normal (o viceversa), desde la Central.

Los semáforos tienen cámaras integradas con capacidad de detectar cruces en rojo. Si detecta una infracción, envía a la Dirección de Tránsito las imágenes captadas, el día, hora y lugar donde se produjo, la patente del automóvil infractor y el tipo de infracción.

Cámaras de Estacionamiento

Las cámaras de fotomultas por estacionamiento detectan vehículos estacionados en lugares indebidos: Hay zonas en las que no está permitido estacionar, las cuales son controladas por cámara. Se conoce el tiempo de tolerancia (en segundos) en que se permite el estacionamiento en la zona prohibida. Pasado ese tiempo, se considera que el vehículo está en infracción.

Si el dispositivo detecta una infracción, envía a la Dirección de Tránsito las imágenes, el momento y lugar donde se produjo, la patente del automóvil infractor, el tipo de infracción y los datos propios de la infracción.

Radares

Los radares detectan infracciones por exceso de velocidad. El radar conoce el límite de velocidad que debe controlar.

Si el dispositivo detecta una infracción, envía a la Dirección de Tránsito las imágenes, el momento y lugar donde se produjo, la patente del automóvil infractor, el tipo de infracción y los datos propios de la infracción.

Cámaras de seguridad

Muestran imágenes en tiempo real. El operador de la Central puede seleccionar la visualización de una o varias cámaras y al detectar una situación anómala, notificará a la policía y/o a los bomberos y/o a el servicio de atención de emergencias médicas, según fuere necesario.

c. Simulación de fallos

El sistema debe generar fallos aleatoriamente en cualquiera de los dispositivos. Los dispositivos pueden volverse a su modo normal de funcionamiento desde la Central. En el caso de un semáforo, al volverlo operativo, se deberá indicar si se lo devuelve al modo normal o al modo intermitente.

d. Revisión de estado de los dispositivos

La Central encuesta cada 1 minuto el estado de situación de los dispositivos. El sistema deberá mostrar en un *Panel de Fallas* todas las fallas detectadas y resaltar sobre el mapa, los dispositivos con anomalías. Visualmente se debe poder filtrar qué dispositivos mostrar, por tipo y estado.

e. Dirección de Tránsito

La *Dirección de Tránsito* recibe las infracciones captadas por los dispositivos y procede a emitir las multas. Al generar la multa (en un documento PDF) deben constar los datos de la *Dirección de Tránsito*, el número de multa (correlativo), la fecha de emisión de la misma, la identificación del titular del automóvil, todos los datos del



automóvil, toda la información propia de la multa (Tipo, lugar, fecha y hora, foto, valor a pagar, puntos de scoring a reducir) y un código de barras conformado por:

- 6 dígitos del número de infracción (justificados con ceros a la izquierda del número)
- seguidos de 12 dígitos correspondientes al importe a cobrar (justificados con ceros a izquierda del número, siendo los dos últimos dígitos, la parte decimal).

Respecto del importe de la infracción, cada tipo de infracción tiene un valor determinado y una cantidad de puntos de scoring de penalización. En el caso de las multas por exceso de velocidad, por cada 10% por encima de la velocidad máxima permitida:

- se adicionará un porcentaje de recargo en forma acumulativa
- Se restará 1 punto de scoring adicional.

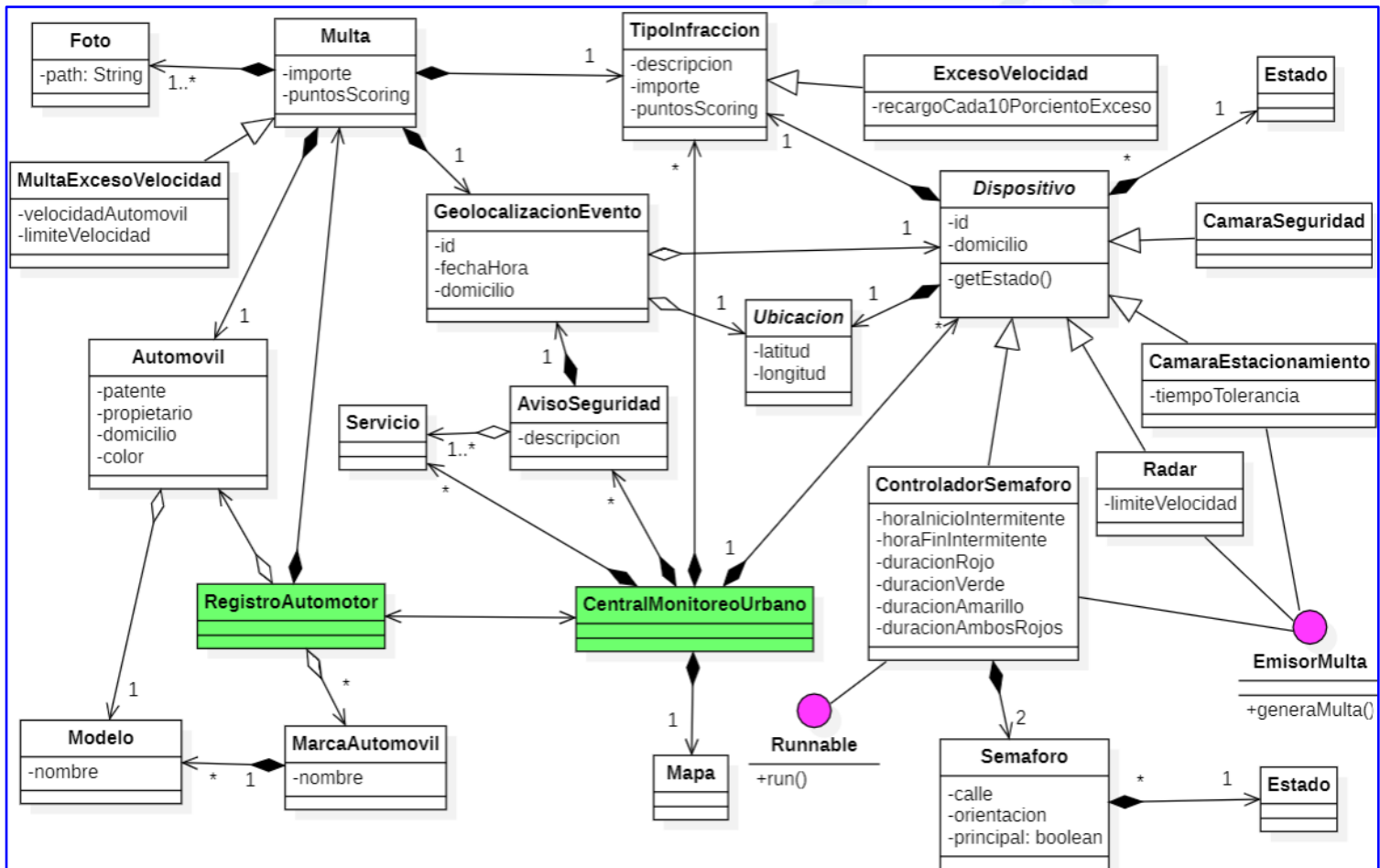
f. Reportes

En cualquier momento se podrán emitir los siguientes **reportes**:

- **Estado de dispositivos:** Grilla mostrando toda la información disponible de cada dispositivo. Se debe poder filtrar la información por tipo de dispositivo y estado del mismo. Si se imprime, en el encabezado, debe constar fecha y hora del mismo. Mostrar cantidad de dispositivos de cada tipo incluidos en el listado y porcentaje de dispositivos con funcionamiento normal y porcentaje de dispositivos con mal funcionamiento.
- **Listado de multas** conteniendo toda la información disponible de las mismas (excepto las fotografías) agrupadas por tipo de multa. Dentro de cada tipo, se las debe ordenar por importe descendente. Como resumen final de cada grupo, mostrar cantidad de multas y sumatoria de importes a pagar. Al final del listado, mostrar cantidad total de multas, sumatoria de importes, multa promedio, multa de menor valor y multa de mayor valor.
- **Consulta de multas de un automóvil:** Dada una patente, listar las multas del mismo.
- **Listado de Avisos de Seguridad:** Se debe poder filtrar entre fechas y horas. Totalizar por tipo de servicio requerido (Policía, Bomberos, Emergencias médicas, etc.) y porcentaje de participación de cada servicio en los avisos. Tener en cuenta que si un servicio no fue llamado nunca, debe de todos modos figurar con porcentaje 0 al pie del reporte.
- **Eventos reportados desde un dispositivo:** Dado un dispositivo determinado, mostrar información de las multas / avisos generados a partir de imágenes generadas por dicho dispositivo.



3. Diagrama de Clases sugerido (sólo clases del dominio)





4. Entregables

1. **Código Fuente:** Se deberá entregar el código fuente del simulador, escrito en lenguaje Java.
2. **Documentación:** Se requiere una documentación técnica que explique la estructura del código, las clases y métodos utilizados, así como cualquier decisión de diseño importante.
3. **Gestión del proyecto:** Se deberá tener constancia de la gestión del proyecto grupal donde haya evidencia del trabajo realizado por cada integrante del grupo. La gestión del proyecto debe hacerse empleando metodologías ágiles de desarrollo.
4. **Presentación del trabajo ante la cátedra y los compañeros de cursado:** Es necesario una presentación del trabajo que describa el proceso de desarrollo, los desafíos encontrados y cómo fueron abordados y resueltos desde el punto de vista técnico, así como cualquier mejora o funcionalidad adicional que hayan llevado adelante. Deberán exponerse evidencias de cómo se han aplicado los principios del paradigma de programación orientada a objetos.
5. **Desarrollo de competencias en inglés:** Dado que desde la asignatura se contribuye al desarrollo de competencias en idioma inglés, se requiere que los alumnos:
 - Realicen la gestión del proyecto, la documentación de las tareas, sprints, etc., en idioma inglés, lo mismo que la codificación de las clases y las interfaces de usuario de la aplicación.
 - En la Presentación del trabajo práctico se deben expresar opiniones y juicios, preparar respuestas guionadas y reforzar sonidos propios del idioma inglés

5. Fecha de Entrega

Durante el primer y segundo cuatrimestre habrá instancias de muestra obligatoria de avances del proyecto. La fecha de entrega del trabajo práctico tendrá lugar en el segundo cuatrimestre. Todas las fechas se especifican en el cronograma de la asignatura.

6. Requerimientos no funcionales

- La **carga** de los **datos** de los dispositivos se realizarán a partir de un archivo JSON / XML
- La **interfaz de usuario** debe implementarse a través de Swing o JavaFX.
- La **persistencia** se implementará mediante serialización o XML. Deberá ser posible salir del sistema, volver a ingresar e iniciar el sistema desde el estado en el que se encontraba al momento de salir.
- Las marcas y modelos de automóviles y los automóviles deben ser creados y editados desde el sistema y persistidos en base de datos
- Considerar:
 - El uso de las clases **Contenedoras** provistas por Java para la administración de listas, conjuntos, diccionarios, etc.
 - Aplicar el lanzamiento desde las clases del dominio de **excepciones** en las validaciones y/o la notificación de eventos con el objetivo de desacoplar el dominio de la de la Interfaz de Usuario.





7. Condiciones de Aprobación

- Conformar un **grupo** de 4 ó 5 personas.
- Implementar la totalidad de la **funcionalidad** solicitada en el enunciado del problema.
- Desarrollar el proyecto utilizando la **metodología ágil** de desarrollo *Scrum*
- Aplicar indefectiblemente en la solución los conceptos de la Programación Orientada a Objetos: **encapsulamiento, abstracción, polimorfismo, herencia**.
- Cumplir estrictamente con el **cronograma de entregas**.
- **Uso de idioma inglés**, como se mencionó anteriormente
- En la Entrega Final:
 - presentar copia digital (por mail, tarea aula virtual, acceso a *git*, etc.) de los **archivos fuentes** del proyecto.
 - deberán estar presentes **todos** los integrantes del grupo. El integrante que no lo estuviera se considerará **fuera del grupo** y deberá presentar su propio TP.
- **Otros conceptos que incidirán en la aprobación del trabajo práctico son:**
 - Reutilización adecuada del código.
 - Eficiencia en los algoritmos (ej: búsquedas, ordenamientos)
 - Bajo acoplamiento entre interfaz y lógica de dominio
 - Validaciones de ingresos de datos y consistencia de la información.
 - Código prolijo, claro y correctamente comentado. (ej: nombres representativos, crear variables e instancias necesarias, sobrecargar métodos)
 - Amigabilidad de las interfaces de usuario
- *La **nota** del trabajo práctico es **individual**, basada en la participación en la resolución y defensa del trabajo práctico, y en los conocimientos conceptuales exhibidos en la entrega. El uso de idioma inglés tendrá una ponderación específica en la Nota final. Oportunamente, se darán a conocer las rúbricas y resultados de aprendizajes esperados con el desarrollo de este TP.*

