Contenido

Trabajo Práctico N° 0: JUnit

Se solicita resolver los siguientes ejercicios, verificando su correcto funcionamiento a través de Junit. Se recomienda aplicar la técnica TDD para obtener mejores resultados y de manera más rápida.

1. Callcenter

   Nos contratan para desarrollar el software de un callcenter para identificar potenciales clientes del servicio de televisión por cable e internet de una nueva empresa llamada "Oeste Cable Color".

   El objetivo de la aplicación es poder incorporar el listado de "Contactos", de los cuales nos interesa conocer:

   - Nombre y Apellido: Valor alfanumérico.
   - Celular: Compuesto del código de país + código de área + número de celular.
   - E-Mail: Debe contener el símbolo '@' y el caracter '.'.
   - Dirección: Valor alfanumérico.
   - Código Postal: Valor numérico.
   - Localidad: Valor alfanumérico.
   - Provincia. Enumerador que contenga las 23 provincias argentinas.
   - Es cliente (Si o No): Inicialmente se carga en "No".
   - Desea ser llamado nuevamente (Si o No): Inicialmente se carga en "Si".

   Cuando el operador del callcenter ingresa a la aplicación, debe seleccionar "Realizar nueva llamada". En ese momento el sistema debe "buscarAlCandidato" y mostrar por pantalla los datos de este. Para determinar qué contacto llamar, el sistema debe seleccionar aleatoriamente un contacto que cumpla con las siguientes condiciones:

   1. El contacto NO debe ser cliente aún.
   2. El contacto desea ser llamado o al menos no informó lo contrario.
   3. El código postal del contacto debe existir en las zonas de cobertura existente.

   Una vez que el operador realiza la llamada debe registrar la misma:

   a. Si la llamada fue exitosa (en ese caso el contacto pasa a ser cliente, y no se lo debe volver a llamar).
   b. Si el contacto no desea ser llamado nuevamente (la llamada pudo no haber sido exitosa, pero se haga un nuevo intento en el futuro).
      Observaciones: Texto libre donde el operador deja registro de lo conversado.

2. Futsal

La asociación argentina de futsal está queriendo realizar un salto de calidad en sus competiciones. Para eso nos contrata para desarrollar el software que les permita anotar las estadísticas de los partidos que se juegan en sus torneos.

Nos comentan que estos torneos están compuestos por 25 equipos que coinciden con los equipos de primera división de AFA (Asociación de Fútbol Argentino).

Para simplificar las pruebas de nuestra primera versión del software acordamos que sólo se cargarán 5 jugadores por equipo, siendo información relevante el precio y la edad de los jugadores que competirán.

Las funcionalidades principales que debe tener el software son las siguientes:

1. Agregar los jugadores a los equipos (los equipos no es necesario dar de alta porque se conforman previo al inicio de la competencia, es decir contaremos de antemano con esa información).
2. Calcular el valor del equipo, esto es la sumatoria del precio de cada jugador.
3. Calcular la edad promedio del equipo.
4. Registrar un nuevo partido. En él se debe poder incorporar al local y al visitante. Una vez creado el partido, se nos solicita poder registrar los siguientes eventos:
    a. Gol, se interesa conocer el autor y el minuto en el que ocurrió
    b. Amonestados, se desea conocer el jugador amonestado y en qué minuto (cuando un jugador es amonestado en dos oportunidades automáticamente debe ser expulsado, y se espera que el sistema informes esta situación).
    c. Expulsados, ya sea por doble amonestación o por expulsión directa, se debe conocer los jugadores que son expulsados del partido.

    Ver el resumen. En este resumen se espera conocer el resultado del partido, y el detalle de los autores de los goles, amonestados y expulsados

3. Sistema
    a. Desarrolle un constructor de la clase Sistema que cumpla con la siguiente firma:

public Sistema(String nombreDelSistema, int cantidadMaximaDeUsuarios)

    b. Desarrolle un método en la clase Sistema que le permita incorporar un usuario a su lista de usuarios. Dicho método devolverá true si el usuario se puede ingresar en el sistema o false en caso contrario (no se puede ingresar un nombre de usuario que ya exista).

public boolean ingresarUsuario (Usuario nuevoUsuario)

    c. Desarrolle el método calcularLaCantidadDeUsuariosLogueados.
    d. Desarrolle el método calcularLaCantidadDeUsuariosBloqueados.
    e. Desarrolle el método calcularElPorcentajeDeUsuariosLogueados.
    f. Desarrolle el método calcularEdadPromedio para conocer a qué público está dirigido el sistema.
    g. En la clase Sistema desarrolle el método loguearUsuario, el cual devolverá true si se logra loguear al usuario y false en caso contrario:

public boolean loguearUsuario (String usuario, String contraseña)

    h. Desarrolle un menú para los administradores del sistema que permita:
        i. Registrar nuevos usuarios en el Sistema
        ii. Ver las estadísticas del sistema (Cantidad de usuarios logueados, bloqueados, porcentaje de usuarios logueados y edad promedio de los usuarios)
        iii. Probar el login. Esto es, como administrador se verifica el acceso al sistema de un usuario determinado. Al salir, el usuario utilizado quedará como logueado para poder evaluar las estadísticas.

4. Reproductor de música

Nos solicitaron desarrollar un programa que permita administrar las listas de reproducción de música de los usuarios de cualquier dispositivo.

El programa debe contar con el siguiente menú de opciones :

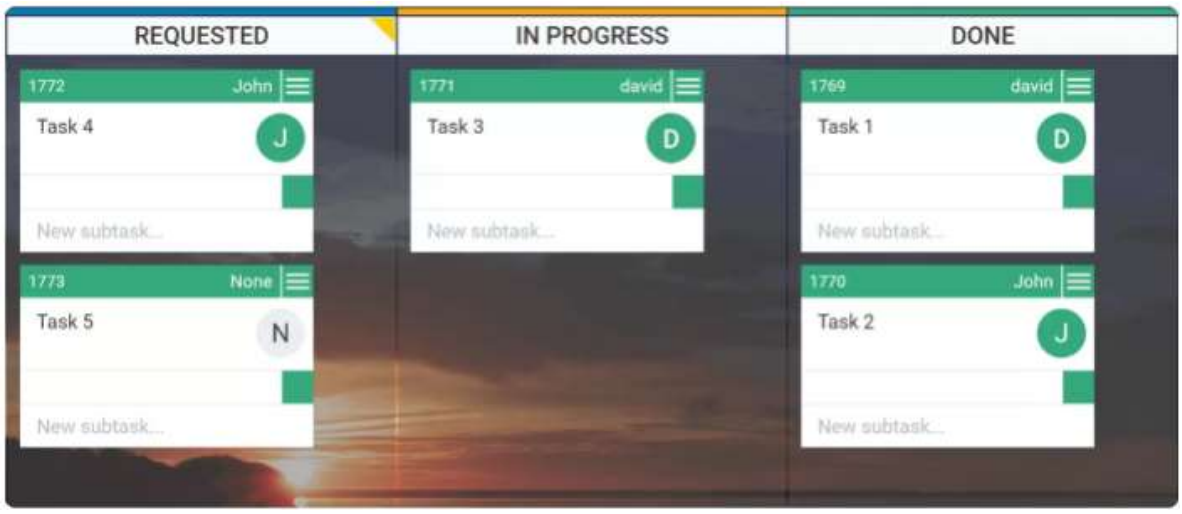
    a. Guardar mis datos personales
    b. Crear una lista de reproducción
    c. c. Agregar canciones a la lista
    d. Reproducir una lista de reproducción.
    e. Del tratamiento del sonido debemos olvidarnos porque eso lo resolverá un equipo específico, pero a nosotros nos toca garantizar que se visualice por pantalla la información de la lista:
        i. El listado de las canciones
        ii. La cantidad de canciones
        iii. La duración de la lista (Se debe mostrar en el formato mm:ss)

5. Vuelo

Desarrollar las clases Pasajero y Vuelo respetando lo descripto en cada método en forma de comentarios.

Trabajo Práctico N° 1: Composición

1. Posiblemente los ejercicios del trabajo práctico N° 0 ya los conocías. Ahora bien, haciendo un repaso de esos mismos ejercicios, ¿dónde encontrás que hayamos aplicado el concepto de composición? ¿Utilizaste alguna relación del tipo agregación en ellos?
2. Desarrollá la clase Montacargas. En él se puede ir cargando distintas Cargas de un peso determinado. El Montacargas tiene un peso máximo, y a medida que se van incorporando cargas al mismo se debe evaluar dicho peso. Si en algún momento el peso máximo del Montacargas es sobrepasado, se debe generar la excepción SobrePeso.
   Verificá el correcto funcionamiento del programa, evitando que la ocurrencia de alguna excepción genere la finalización de la ejecución.
3. Se desea implementar un tablero Kanban en donde se pueda agregar tareas. Cada tarea puede tener distintos estados (al menos se deben considerar los estados PENDIENTE, EN CURSO y FINALIZADA). Una tarea es creada por un usuario determinado, pero mientras se encuentre PENDIENTE, cualquier usuario puede tomarla y pasarla a otro estado, según corresponda. Sin embargo, una tarea EN CURSO, no puede ser tomada por un usuario diferente al que la está realizando porque eso provocaría un solapamiento de actividades.
   A continuación, te mostramos un grafico sobre cómo se visualiza un tablero de estas características para que te sirva de guía en la resolución del ejercicio.



4. Implementar la clase Banco, el cual contiene clientes y cuentas.
   Una cuenta bancaria es un contrato financiero con una entidad bancaria en virtud del cual se registran el balance y los subsiguientes movimientos de dinero del cliente.
   Nosotros vamos a ser menos estrictos, ya que por el momento no nos interesa computar los movimientos de dinero del cliente sino los totales instantáneos (es decir, cuánto dinero tiene el cliente a cada momento luego de realizar cada transacción).
   Tipos de cuentas
   Vamos a establecer, para el contexto de este ejercicio, tres tipos de cuentas:
   a. Cuenta Sueldo
   b. Caja de Ahorros
   c. Cuenta Corriente.

   Cada una cumple con ciertas reglas de negocio, las cuales se detallan a continuación.

   Cuenta Sueldo
   Es el tipo de cuenta más simple, ya que se rige por la premisa de que en tanto y en cuanto se tenga tanto o más dinero en cuenta del que se quiere extraer, la operación se debe efectuar correctamente.

   Caja de Ahorro
   Similar a la anterior, pero se pide que luego de la quinta extracción de dinero se cobre un costo adicional por extracción de $ 6

   Cuenta Corriente
   La más compleja de las cuentas, ésta permite establecer una cantidad de dinero a girar en descubierto. Es por ello que cada vez que se desee extraer dinero, no sólo se considera el que se posee, sino el límite adicional que el banco estará brindando. Por supuesto esto no es gratis, ya que el banco nos cobrará un 5% como comisión sobre todo el monto en descubierto consumido en la operación.
   Por ejemplo, si tuviéramos $ 100 en la cuenta, y quisiéramos retirar $ 200 (con un descubierto de $ 150), podremos hacerlo. Pasaremos a deberle al banco $ 105 en total: los $ 100 que nos cubrió, más el 5% adicional sobre el descubierto otorgado.

   A modo de ayuda, se incluyen 4 casos de prueba que se deberían considerar para verificar el tratamiento de las cuentas corrientes.

Caso de Prueba	numeroDeOperacio	tipoDeOperacion	importe	saldo	descubiertoActual	deuda
queSeCobreCincoPorcientoDeComisionAlDepositarDineroLuegoDeHaberRealizadoUnaExtraccionMayorAlSaldo				$ 0,00	$ 0,00	$ 0,00
	1	Depósito	$ 100,00	$ 100,00	$ 0,00	$ 0,00
	2	Extracción	$ 200,00	$ 0,00	$ 100,00	$ 5,00
	3	Depósito	$ 200,00	$ 95,00	$ 0,00	$ 0,00
queSeCobreElCincoPorCientoDeComisionPorMasQueElProximoDepositoNoSeaSuficieteParaCubrirElDescubierto				$ 0,00	$ 0,00	$ 0,00
	1	Depósito	$ 100,00	$ 100,00	$ 0,00	$ 0,00
	2	Extracción	$ 200,00	$ 0,00	$ 100,00	$ 5,00
	3	Depósito	$ 100,00	$ 0,00	$ 5,00	$ 0,00
queUnaExtraccionCuandoYaSeTieneDeudaIncrementeLaDeuda				$ 0,00	$ 0,00	$ 0,00
	1	Depósito	$ 100,00	$ 100,00	$ 0,00	$ 0,00
	2	Extracción	$ 200,00	$ 0,00	$ 100,00	$ 5,00
	3	Extracción	$ 50,00	$ 0,00	$ 150,00	$ 7,50
queVariasOperacionesDeDepositoYExtraccionGenerenElSaldoYLaDeudaCorrecto				$ 0,00	$ 0,00	$ 0,00
	1	Depósito	$ 100,00	$ 100,00	$ 0,00	$ 0,00
	2	Extracción	$ 200,00	$ 0,00	$ 100,00	$ 5,00
	3	Extracción	$ 50,00	$ 0,00	$ 150,00	$ 7,50
	4	Depósito	$ 50,00	$ 0,00	$ 107,50	$ 0,00
	5	Extracción	$ 40,00	$ 0,00	$ 147,50	$ 2,00
	6	Depósito	$ 150,00	$ 0,50	$ 0,00	$ 0,00

DESCUBIERTO_MAXIMO
$ 150,00

A su vez cada Cliente puede tener una o varias cuentas en el banco. Basado en el saldo que los clientes tienen en sus cuentas, el banco los clasifica en VIP. Actualmente todos aquellos clientes con una sumatoria de saldo mayor a $ 1.000.000 y sin tener ninguna cuenta con saldo negativo, son considerados como VIP. Se solicita tener disponible la lista de clientes VIP.

5. Nos contratan de una empresa de domótica que quiere diseñar un sistema inteligente para el control del flujo de personas y el acceso a los pisos y los ascensores en los edificios donde se encuentran implementados. Para entrar en el mercado se aprovechará el contacto con un cliente que está interesado en implementar el sistema cuanto antes.
   Kim, Ki Nam, CEO de Samsung, nos comenta que sus oficinas de Bouchard 459 tienen 5 ascensores. Actualmente se están usando 2 para los pisos pares y dos para los pisos impares, teniendo un quinto ascensor que es utilizado para las áreas de servicio, el cual soporta una carga mas grande que la de los otros 4 (la capacidad se mide tanto por el peso como por la cantidad de personas que suben).
   El ingreso al edificio requiere la identificación previa de cada empleado o cliente. Los primeros ingresar a través de su huella dactilar y los clientes acceden a través de una tarjeta de invitado.
   Aprovechando la posibilidad que les brinda el acceso controlado Kim desea poder conocer en tiempo real:
   a. El tiempo en que cada empleado se encontró en su piso.
   b. El tiempo promedio en el que los clientes se encuentran dentro del edificio (tasa de espera de atención).
   c. El listado de los empleados ausentes en un día determinado.
   d. Evaluar si la cantidad de ascensores es suficiente. ¿En algún momento se supera la capacidad máxima del ascensor? ¿Cuántas veces por día?

Trabajo Práctico N° 2: Herencia

1. Se desea desarrollar un software para la gestión de los empleados de una empresa. Para cada empleado se desea conocer el nombre, apellido, salario y fecha de nacimiento.
   Como la empresa está organizada en forma de departamentos, también se debe conocer el gerente encargado de cada uno de ellos. Adicionalmente, los gerentes cuentan con la posibilidad de tener una cochera en donde estacionar sus vehículos.
   También se cuenta con un manejo especial para aquellos empleados de tipo “Ingenieros”, dado que su salario base se ve afectado por un concepto adicional denominado “para la productividad”.
   Por último, se encuentran los directores quienes además de tener su cochera, poseen un “sueldo extra” producto de tener la responsabilidad de ser directivo de la empresa.

2. Se desea desarrollar un nuevo modelo de smartwatch para hacer competencia a marcas como Garmin y Polar.
   El requisito fundamental para estos relojes, dedicados específicamente al monitoreo de los distintos deportes, es poder conocer determinadas características del individuo (deportista) como ser:
   - Nombre
   - Edad actual
   - Peso actual
   - Altura actual
   - Pasos diarios

   Esta información podrá ser utilizada como parámetro en los distintos análisis de cada actividad.
   Luego, dependiendo de las preferencias de cada uno, se tendrá una especialización por cada disciplina:
   - Carrera
   - Ciclismo
   - Natación
   - Caminata

   Cada deportista elige el tipo de deporte que realiza, y en función de eso es el comportamiento que se observa al consultar el estado del reloj.

   <u>Carrera</u>

   Los deportistas que sean corredores, desean conocer en tiempo real la siguiente información:

   - Distancia recorrida (en kilómetros)

- Tiempo transcurrido
- Ritmo (minutos que se demora en completar un kilómetro)
- Zona de frecuencia cardíaca actual (*)

Obviamente para poder calcular la zona de frecuencia cardíaca, será necesario conocer los límites de cada zona para el corredor (ver detalle al pie).

Ciclismo

Para los ciclistas, la información que se desea mostrar en línea en la pantalla del reloj es la siguiente:

- Distancia recorrida (en kilómetros)
- Tiempo transcurrido
- Velocidad (expresada en kilómetros por hora)
- Zona de frecuencia cardíaca actual (*)
- Cadencia (en RPM)

Al igual que para los runners, se debe poder calcular la zona de frecuencia cardíaca. Será necesario conocer los límites de cada zona para el ciclista (ver detalle al pie)

Natación

Los nadadores desean contar con esta información:

- Distancia recorrida (en metros)
- Tiempo (en minutos y segundos)
- Ritmo (tiempo, en minutos y segundos que demora en recorrer 100 metros)

Caminata

Para aquellas personas que desean utilizar el reloj como soporte en su caminata, simplemente se mostrará el tiempo y la distancia del recorrido.

(*) Zonas de frecuencia cardíaca: Se deben considerar 5 zonas:

1. Descanso

2. Calentamiento

3. Aeróbico

4. Umbral

5. Máximo

Cada zona tiene definido la frecuencia cardíaca mínima (donde se inicia la zona) y la frecuencia cardíaca máxima (donde finaliza). Es importante señalar que las zonas varían según la actividad física que se esté realizando.

(**) La determinación del estado del reloj se realizará a partir de la recepción de pulsos recibidos desde los sensores que contiene el mismo. Es decir:

- Cada un segundo se recibirá un pulso para incrementar el tiempo.
- Cada un metro se recibirá un pulso del GPS avisando el incremento.
- Cada paso que realice la persona se enviará un pulso informándolo.
- El resto de los parámetros recibirán el valor actualizado cada vez que se establezca la conexión con el sensor (velocidad, cadencia y frecuencia cardíaca).

3. El hospital central de la república ha experimentado varios casos de mala praxis, producto de haberle prescripto determinados alimentos a pacientes que no podían comerlos. Para evitar que esta situación se repita, ha decidido desarrollar un programa en el cual se deba registrar la alimentación indicada para cada paciente.

Para ello, se desea poder clasificar a los pacientes en:

a. Diabéticos
   - Diabetes tipo 1 (Insulino-Dependientes).
   - Diabetes tipo 2.
b. Oncológicos.
c. Celiacos.
d. Hipertensos
e. Generales

Al mismo tiempo, la dirección decidió la terciarización de la cocina bajo la administración de una empresa dedicada a la gastronomía, que, si bien no cuenta con profesionales de salud en su plantel, es muy eficiente en la elaboración, disminución de los costos operativos, y variedad de los platos, lo cual, se dice, ayuda a que los pacientes se encuentren más a gusto y mejora su recuperación.

Teniendo en cuenta que no existirá una comunicación fluida entre los médicos y nutricionistas del hospital con el equipo de cocina, es muy importante que el menú disponible sea armado con el más riguroso sistema de calidad, en donde se detalle cada uno de los ingredientes de cada plato, junto con el proceso de elaboración (si se prepara al horno, frito, en olla, o es un plato sin cocción).

Está planificada una primera etapa de este nuevo proceso, en la cual se tomará una muestra de tipos de pacientes, y se describirá las restricciones que tienen cada uno de estos. A continuación, se detallan estas restricciones:

a. Para los pacientes diabéticos:

   i. Los pacientes diabéticos no pueden consumir nada que contenga azúcar.
   ii. Se debe garantizar que en un determinado momento del día los pacientes con diabetes tipo 2 reciban su dosis de insulina.
b. Los pacientes oncológicos que se encuentren bajo tratamiento de quimioterapia o radioterapia no pueden consumir ningún alimento crudo.
c. Los pacientes celíacos no pueden comer nada que tenga TRIGO, AVENA, CEBADA y CENTENO.
d. A los pacientes hipertensos no se les puede enviar ningún plato que tenga sal.
e. Las pacientes generales no tienen ninguna restricción con respecto a la alimentación.

Se solicita diseñar, codificar y testear las clases principales que permitan atender los requerimientos funcionales del hospital.

Trabajo Práctico N° 3: Polimorfismo

1. Implementar un carrito de compras, que nos sirva para implementar en cualquier negocio, y buscando asemejarlo lo más posible a la realidad. Es decir, cuando nosotros vamos al supermercado, en el carrito podemos poner un objeto de tipo leche o muchos de ese mismo tipo. Luego, al llegar a la caja, se visualiza la cantidad que hemos comprado de cada producto que sean iguales y si llegara a haber un descuento por cantidad, se debe aplicar dicho descuento. No olvides desarrollar las pruebas que garanticen el correcto funcionamiento. Para eso tené en cuenta verificar compras individuales, por cantidad, con y sin descuentos.

Ejercicios integradores

1. Nos contratan de las Fuerzas Armadas para desarrollar una aplicación que les permita planificar las batallas que se pueden librar en las distintas zonas de conflicto.
   Para eso nos comentan que no enviarán hombres al frente sino Vehículos con distintas características.
   Algunos vehículos pueden desempeñarse por tierra, otros por agua y otros por aire. Sin embargo, las últimas tecnologías les permitieron a las fuerzas incorporar nuevos vehículos:
   - Anfibios: Pueden desempeñarse tanto por tierra como por agua
   - Hidroaviones: Son un tipo de avión especial que puede (además de volar) realizar operaciones por agua.

Como información adicional para la construcción del software nos presentan los siguientes requerimientos:

1. Los vehículos de guerra se identifican unívocamente por su código.
2. Las batallas se identifican unívocamente por su nombre.
3. El convoy es el conjunto de Vehículos que disponen (no importa el tipo de vehículo).
4. Para enviar un Vehículo a una batalla se debe confirmar que el mismo forme parte de la fuerza.
5. El Vehículo que se envía a una batalla determinada debe ser apto para esta:
   a. En las batallas aeras, sólo se podrán utilizar Vehículos Voladores.
   b. En las batallas por tierra se podrán utilizar Vehículos Terrestre.
   c. En las batallas por agua sólo se podrán utilizar Vehículos Acuáticos.

Para resolver el ejercicio planteado se recomienda tomar como base el siguiente proyecto (https://github.com/jmonteagudo/recuperatorio), logrando que todos los test se ejecuten de manera exitosa, e incorporando los test adicionales que crea conveniente.

2. Nos solicitan desarrollar el software para un club que se dedica a la organización de eventos deportivos del siguiente tipo:
   a. Natación
   b. Ciclismo
   c. Pedestrismo (Running)
   d. Acuatlon (Combinación entre natación y pedestrismo)
   e. Duatlon (Combinación entre pedestrismo y ciclismo)
   f. Triattlon (Combinación de las tres diciplinas, natación, ciclismo y pedestrismo)

   Para poder participar de los eventos deportivos, primero se tiene que ser socio. Los socios se identifican unívocamente por el número de socio, y, según la disciplina que realizan, se requiere conocer determinada información:
   - Nadadores: Se desea conocer su estilo preferido (Los estilos son Croll, Espalda, Pecho y Mariposa).

- Ciclistas: Se desea conocer el tipo de bicicleta que utilizan (De montaña, de ruta o de triatlón)
- Corredores: Se desea conocer su distancia preferida (5 Km, 10 Km, 21 Km o la Maratón de 42 Km).

Para este club la seguridad de sus socios es muy importante, en consecuencia, nos encargan custodiar que el sistema no permita que un deportista no pueda inscribirse en una competencia para la cual no esté preparado, por tal motivo, para el caso que se intente realizar eso, el sistema tiene que generar una Excepción (NoEstaPreparado). Un deportista "no está preparado" si por ejemplo desea participar de un torneo de natación, pero no sabe nadar.

Para resolver el ejercicio planteado se recomienda tomar como base el siguiente proyecto (https://github.com/jmonteagudo/eva03), logrando que todos los test se ejecuten de manera exitosa, e incorporando los tests adicionales que crea conveniente.

3. Se desea desarrollar un software de monitoreo, el cual sirva para detectar colisiones entre los vehículos que circulan por un área determinada. Sabiendo que se cuenta con distintos tipos de vehículos (medios de transporte), se desea conocer constantemente la ubicación (coordenadas) de cada uno, para poder evaluar si se produce una colisión (las coordenadas de dos vehículos es la misma).
   Para esto se solicita desarrollar el método "actualizarCoordenadas", el cual, ante la ocurrencia de una colisión debe generar una excepción "CollitionException".

   Para resolver el ejercicio planteado se recomienda tomar como base el siguiente proyecto (https://github.com/jmonteagudo/PBIIEjemploSegundoParcial), logrando que todos los test se ejecuten de manera exitosa, e incorporando los tests adicionales que crea conveniente.