IES de Teis
Avda. de Galicia, 101
36216 – Vigo
Tfno: 886 12 04 64
e-mail: ies.teis@edu.xunta.es
http://www.iesteis.es

Nombre y apellidos: Nicolás Primoy Pucheta

DNI: 39455949R

Examen Segunda Evaluación	
Módulo	**Contornos de Desenvolvemento**
Fecha	**03/04/2025**
Hora	**16:00 (llamamiento 15:45)**
Ciclo	**DAM**
Régimen	**Adultos**
Modalidad	**Presencial**
Duración	**2h 15’**
Puntuación máxima	**7**

Calificación

Parte 1	Parte 2	Total
3	**4**	**7**

IES de Teis
Avda. de Galicia, 101
36216 – Vigo

Tfno: 886 12 04 64
e-mail: ies.teis@edu.xunta.es
http://www.iesteis.es

Instrucciones generales

El examen se realizará bajo el **sistema operativo Windows** para realizar todas las actividades propuestas en él.

Tendrás que entregar una memoria, en formato .pdf donde harás las explicaciones y capturas de pantalla de la ejecución de cada una de las preguntas. El nombre del fichero tendrá el formato

apellido_nombre_examen_cd_t2.pdf

Es muy importante recoger en dicho informe TODAS las operaciones realizadas, para que éstas sean evaluables. Puedes utilizar como plantilla para dicho informe el enunciado del examen, y contestar después de cada una de las preguntas.

Para poder realizar alguno de los ejercicios, se proporcionan algunos ficheros adicionales. Puedes copiar todo el conjunto de ficheros para los ejercicios desde la tarea del aula virtual de la entrega del examen.

Además, también tendrás que entregar ficheros adicionales en algunos de los ejercicios. Se indicará en cada uno de ellos qué debes entregar.

Todos los archivos, incluida la memoria realizada, deben añadirse a un archivo comprimido llamado

apellido_nombre_examen_cd_t2.zip

Dentro de ese archivo .zip debes añadir la memoria en pdf y una carpeta de archivos por cada ejercicio, siguiendo el esquema **indicado al final del examen**. El archivo resultante tendrás que entregarlo en la tarea correspondiente del aula virtual.

Podrás responder al examen tanto en castellano como en gallego.

IES de Teis
Avda. de Galicia, 101
36216 – Vigo
Tfno: 886 12 04 64
e-mail: ies.teis@edu.xunta.es
http://www.iesteis.es

Parte 1: Diseño orientado a objetos

Ejercicio 1 - Modelado UML de una Aplicación de Gestión de Fórmula 1 (3 puntos)

Una empresa de software desea desarrollar una aplicación para gestionar los **equipos y carreras de Fórmula 1**. Se necesita modelar el sistema de manera que represente correctamente las entidades principales y sus relaciones.

💡 Nota: Tan solo es necesario modelar las clases indicadas, pudiendo omitir otros detalles o clases auxiliares. No es necesario añadir los getters y setters de los atributos salvo que se considere relevante para el modelo. **Requisitos del modelo:**

Escuderías y Pilotos:

- Cada escudería de Fórmula 1 tiene **un nombre** (por ejemplo, **Mercedes-AMG Petronas**, **Red Bull Racing**) y una sede (ciudad).
- Una escudería **tiene dos pilotos principales** y puede tener o no pilotos de reserva.
- Los pilotos tienen atributos como **nombre, nacionalidad, número de carreras disputadas y puntos acumulados a lo largo de su carrera**.
- Existen **dos tipos de pilotos**:
    - **Piloto Oficial** → Corre en las carreras y suma puntos para el equipo.
    - **Piloto de Reserva** → No participa en carreras, pero puede sustituir a un piloto titular si es necesario. Un piloto reserva puede sustituir a pilotos de diferentes escuderías.

Temporada, carreras y circuitos:

- Una **temporada** de Fórmula 1 se realiza en **un año natural** y cuenta con **un conjunto de equipos participantes y un conjunto de carreras**.
    - Cada temporada se compone de **varios Grandes Premios** (carreras) que se celebran en diferentes circuitos. En una temporada, no puede haber menos de 5 grandes premios, ni más de 24.
    - Cada equipo participa en todas las carreras de la temporada.
- Un **circuito** tiene **nombre, ubicación y longitud en metros**.
- Cada **Gran Premio (carrera)** se realiza en un circuito **específico** y tiene una **fecha de realización**. Sin embargo, en un mismo circuito pueden celebrarse múltiples Grandes Premios (carreras) de una misma temporada. Por ejemplo, en una misma temporada, el circuito Red Bull Ring de Austria albergó dos Grandes Premios: el Gran Premio de Austria y el Gran Premio de Styria.
- Se deben registrar los **resultados de la carrera**, con los puntos obtenidos por cada piloto según su posición en la parrilla de llegada.

IES de Teis
Avda. de Galicia, 101
36216 – Vigo
Tfno: 886 12 04 64
e-mail: ies.teis@edu.xunta.es
http://www.iesteis.es

Instrucciones:

1. **Crear el diagrama de clases UML en Draw.io** que represente los elementos descritos. **Utiliza correctamente los siguientes conceptos:**
    - **Herencia**
    - **Interfaces** (si existiesen).
    - **Relaciones**
    - **Atributos estáticos**
    - **Métodos relevantes** (no incluyas constructores ni getters/setters).
    - **Multiplicidades** (en las relaciones que proceda).

Utiliza la **notación UML** adecuada para la representación de todos los elementos. **Exporta el diagrama en formato PNG** y **adjúntalo al examen**. Escribe en el documento del examen todos los comentarios que consideres necesarios. Trata de añadir las posibles aclaraciones o restricciones del diagrama utilizando notas.

Parte 2: Diseño y realización de pruebas

Ejercicio 2 – Diseño e Implementación de Pruebas para un Sistema de Gestión de Taxis (4 puntos)

Una compañía de taxis necesita un sistema para **calcular el precio de un trayecto** basado en los siguientes criterios:

- **Tarifa base**: El coste mínimo de cualquier viaje es de **3.50€**.
- **coste por distancia**:
    - **Hasta 5 km:** 1.20€/km.
    - **De 5 a 15 km:** 1.00€/km.
    - **Más de 15 km:** 0.80€/km.
- **coste por tiempo** (solo si la velocidad promedio es menor a 10 km/h): **0.50€ por minuto** si el vehículo está detenido o en tráfico.
- **Suplemento nocturno**: Si el viaje ocurre entre **22:00 y 6:00**, se aplica un **recargo del 20%** sobre el total.
- **Suplemento por día festivo**: Si el viaje ocurre en un día festivo, se aplica un **recargo fijo de 5€**.
- **Requisitos adicionales**:
    - Si la distancia o el tiempo ingresado es negativo, el sistema debe lanzar una **excepción** (IllegalArgumentException).
    - El cálculo de la tarifa debe ser **redondeado a dos decimales**.

Además, la compañía quiere analizar el **tipo de trayectos** realizados por sus clientes, por lo que también se incluye una funcionalidad para clasificar las tarifas, que debe cumplir los siguientes requisitos:

IES de Teis
Avda. de Galicia, 101
36216 – Vigo
Tfno: 886 12 04 64
e-mail: ies.teis@edu.xunta.es
http://www.iesteis.es

- Si el **coste total es inferior a 5€**, el trayecto se considera **económico**.
- Si el **coste total es entre 5 y 15€**, el trayecto se considera **estándar**.
- Si el **coste total es superior a 15€**, el trayecto se considera **premium**.

Para ello, se crea una aplicación Java con una clase GestorTaxis que incluye el método calcularTarifa() y clasificarTarifa() :

```
public class GestorTaxi {

    public static double calcularTarifa(double distanciaKm, int tiempoMin, boolean esNocturno,
boolean esFestivo) {
        if (distanciaKm < 0 || tiempoMin < 0) {
            throw new IllegalArgumentException("Distancia y tiempo no pueden ser negativos.");
        }

        double tarifaBase = 3.50;
        double costeDistancia;
        double costeTiempo = 0.0;
        double total;

        if (distanciaKm <= 5) {
            costeDistancia = distanciaKm * 1.20;
        } else if (distanciaKm <= 15) {
            costeDistancia = (5 * 1.20) + ((distanciaKm - 5) * 1.00);
        } else {
            costeDistancia = (5 * 1.20) + (10 * 1.00) + ((distanciaKm - 15) * 0.80);
        }

        if (distanciaKm > 0 && (distanciaKm / (tiempoMin / 60.0)) < 10) {
            costeTiempo = tiempoMin * 0.50;
        }

        total = tarifaBase + costeDistancia + costeTiempo;

        if (esNocturno) {
            total *= 1.20;
        }

        if (esFestivo) {
            total += 5.00;
        }

        return Math.round(total * 100.0) / 100.0;
    }
    /**
```

IES de Teis
Avda. de Galicia, 101
36216 – Vigo
Tfno: 886 12 04 64
e-mail: ies.teis@edu.xunta.es
http://www.iesteis.es


```
 * Método para clasificar la tarifa según su valor.
 * Si el coste total es inferior a 5€, el trayecto se considera económico.
 * Si el coste total es entre 5 y 15€, el trayecto se considera estándar.
 * Si el coste total es superior a 15€, el trayecto se considera premium.
 * Tendrá gestión para valores de tarifa inválidos
 * Se usará para diseñar tests de caja negra sin conocer la implementación interna.
 */
public static String clasificarTarifa(double tarifa) {
    // Implementación pendiente
}
```

Puedes encontrar el proyecto de Intellij IDEA sobre el que trabajar en los archivos proporcionados para el examen.

Apartado 1: Diseño de Pruebas de Caja Negra (1 punto)

Realiza el diseño de pruebas de caja negra para el método clasificarTarifa() siguiendo los siguientes pasos:

- Definir **particiones de equivalencia y valores límite**.
- Identificar los **escenarios relevantes y sus valores de entrada**.
- Define los **casos de prueba** con los datos de entrada y salida esperados.

Añade en esta memoria toda la información necesaria, así como aclaraciones y decisiones tomadas al respecto. Puedes utilizar las tablas propuestas en los apuntes para ello.

Condición de entrada	Clase de equivalencia	Clases válidas	COD	Clases no válidas	COD
CosteTotal			V1	=<0	NV1
CosteTotal	Rango	=1>5	V1	<1	NV2
CosteTotal	Rango	=5>15	V1		NV3
CosteTotal	Rango	>15	V1		NV4
CosteTotal			V1	Strings	NV5

IES de Teis
Avda. de Galicia, 101
36216 – Vigo
Tfno: 886 12 04 64
e-mail: ies.teis@edu.xunta.es
http://www.iesteis.es

Caso de prueba	Clase de equivalencia	Condiciones de entrada	Resultado esperado
CASO 1		0	No válido
CASO 2		1	Trayecto económico
CASO 3		4	Trayecto económico
CASO 4		5	Trayecto estándar
CASO 5		15	Trayecto estándar
CASO 6		16	Trayecto premium
CASO 7		dfaasd	No válido
CASO 8			
CASO 9			

Apartado 2: Diseño de Pruebas de Caja Blanca (1.5 puntos)

Realiza el diseño de pruebas de caja blanca para el método calcularTarifa() siguiendo los siguientes pasos:

- Obtener el **grafo de flujo**. Créalo utilizando Draw.io (versión de escritorio).
  - Etiqueta los nodos con letras mayúsculas para definir más cómodamente los caminos.
  - Numera las regiones resultantes.
- Calcular la **complejidad ciclomática** utilizando las 3 fórmulas conocidas.
- Definir los **caminos de prueba**.
- Diseñar los **casos de prueba** con los datos de entrada y salida esperados.

Deberás entregar el fichero drawio creado (extensión .drawio) así como exportado en formato .png, y añadirlo a la memoria en el apartado correspondiente. Añade también

todas las aclaraciones y decisiones respecto al cálculo de complejidad ciclomática, obtención de caminos de prueba y los casos de prueba resultantes.

Apartado 3: Implementación y Ejecución de Tests en JUnit (1.5 puntos)

Crea una clase GestorTaxisTest que contenga los tests para la clase GestorTaxi.

- Implementar las pruebas unitarias en **JUnit 5** a partir del diseño de tests de caja blanca para el método calcularTarifa().
- Crea un **test parametrizado** para probar al menos 5 conjuntos de valores diferentes de parámetros para el método calcularTarifa(), diferentes a los tests diseñados a partir del método de caja blanca (valores válidos).
- Ejecuta todos los tests, comprueba que todos pasan y realiza capturas de pantalla de la ejecución.
- Utilizando la extensión MetricsReloaded, comprueba que la complejidad ciclomática del método calcularTarifa() es la esperada y coincide con la calculada a partir del grafo de flujo.
- Realiza una ejecución de test con cobertura de código, comprueba que se alcanza una cobertura del 100% con los métodos diseñados mediante la técnica de caja blanca, y realiza capturas de pantalla de la ejecución.

RESUMEN DE ENTREGA

A modo resumen, deberás entregar, tras realizar los ejercicios, un archivo comprimido con la siguiente estructura

- *apellido_nombre_examen_cd_t2.zip*
  - *apellido_nombre_examen_cd_t2.pdf (memoria)*
  - *ejer1/*
    - *apellidos_nombre_ejer1_cd_t2.drawio*
    - *apellidos_nombre_ejer1_cd_t2.png*
  - *ejer2/*
    - *apellidos_nombre_ejer2_cd_t2.drawio*
    - *apellidos_nombre_ejer2_cd_t2/*
      - *.idea*
        - *…*
      - *out*
        - *…*
      - *src*
        - *…*
          - *GestorTaxi.java*
      - *test*
        - *…*

IES de Teis
Avda. de Galicia, 101
36216 – Vigo
Tfno: 886 12 04 64
e-mail: ies.teis@edu.xunta.es
http://www.iesteis.es

    - *GestorTaxiTest.java*
- *pom.xml*