

Al realizar esta prueba de evaluación ACEPTO (individualmente) y ACEPTAMOS (en grupo) la "cláusula de veracidad" por la que no recibiremos ni daremos ayuda en esta prueba y garantizamos la autoría del 100% de los resultados.

El incumplimiento por mi/nuestra parte de los deberes derivados de las buenas prácticas de honestidad académica podrá dar lugar a la adopción de las medidas contenidas en la Normativa de convivencia universitaria y de régimen disciplinario de la Universitat Politècnica de València.

Ejercicio 1 (4 puntos). Obtened el diagrama de casos de uso en UML del siguiente enunciado.

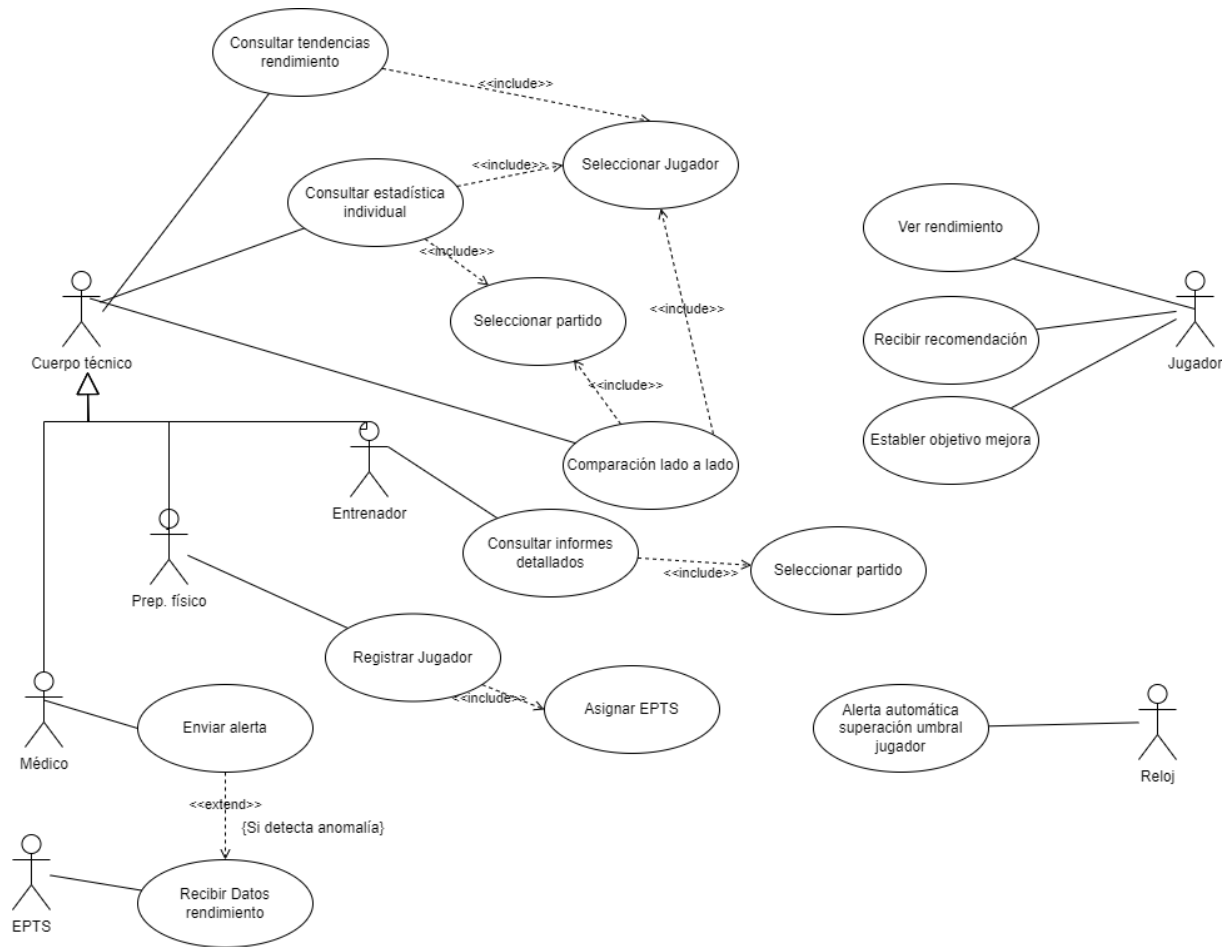
Un equipo de baloncesto profesional desea implementar un sistema para monitorizar el rendimiento de sus jugadores durante los entrenamientos y partidos. El objetivo del sistema es recopilar datos en tiempo real sobre las constantes vitales y el rendimiento físico de los jugadores, así como el acceso personalizado a las estadísticas obtenidas durante los partidos, para mejorar la toma de decisiones del cuerpo técnico (entrenador principal, asistentes, preparadores físicos y cuerpo médico), y optimizar el rendimiento del equipo.

Los preparadores físicos serán los encargados de registrar nuevos jugadores con sus datos personales y características físicas. En el momento de darlo de alta, a cada jugador se le asignará un dispositivo de seguimiento electrónico del rendimiento o EPTS. Cada EPTS es programado para enviar en tiempo real los datos de rendimiento y constantes vitales al sistema, tanto durante los entrenamientos como durante los partidos. El sistema debe emitir alertas si al recibir dicha información detecta cualquier anomalía en las constantes vitales de los jugadores. La alerta será recibida por el equipo médico, que debe verificar su recepción.

Además, el sistema debe permitir al cuerpo técnico la consulta de estadísticas individuales de forma que, tras seleccionar a un jugador y un partido puedan visualizar las estadísticas obtenidas por él: minutos jugados, puntos anotados, rebotes, asistencias, robos, bloqueos y eficiencia general. También debe facilitar al cuerpo técnico la comparación de rendimiento entre dos jugadores en un mismo partido, mostrando una comparación lado a lado de sus estadísticas de juego. Asimismo, el cuerpo técnico podrá acceder a las tendencias en el rendimiento de un jugador a lo largo de la temporada mediante gráficos y reportes que muestren la evolución de las estadísticas clave en cada partido. El primer entrenador, además de lo anterior, podrá acceder a los informes detallados del rendimiento del equipo en los partidos que seleccione (porcentaje de tiros acertados, rebotes totales, asistencias y pérdidas de balón). Finalmente, el sistema debe enviar alertas automáticas al cuerpo técnico si un jugador supera o no alcanza ciertos umbrales de rendimiento predefinidos durante un partido, como anotar más de 30 puntos o tener una eficiencia por debajo de un valor determinado.

Los jugadores también deben poder acceder a sus propios datos a través de una aplicación móvil, donde podrán ver su rendimiento, recibir recomendaciones personalizadas y establecer objetivos de mejora.

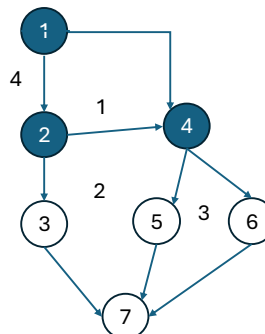
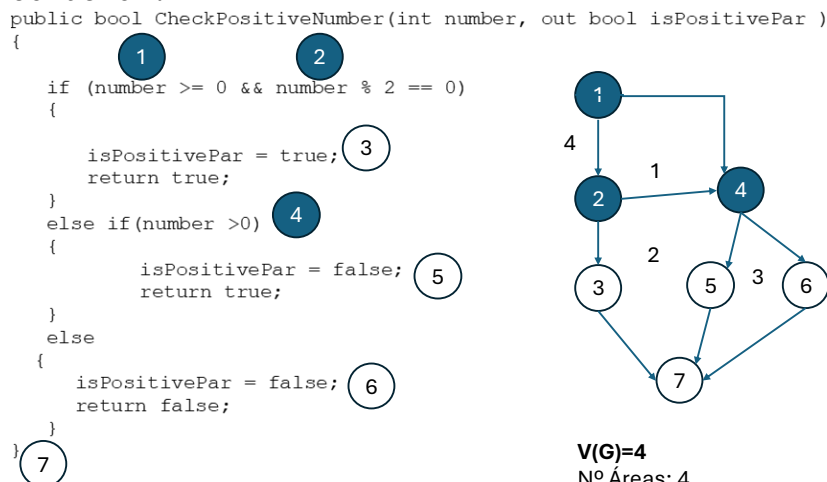
Solución:



Ejercicio 2 (3 puntos). Diseñad los casos de prueba para el método `CheckPositiveNumber` siguiendo la técnica del camino básico de caja blanca, dibujad el grafo de flujo, calculad la complejidad ciclomática, propocionad los caminos independientes y los casos de prueba asociados a éstos.

```
public bool CheckPositiveNumber(int number, out bool isPositivePar )
{
    if (number >= 0 && number % 2 == 0)
    {
        isPositivePar = true;
        return true;
    }
    else if(number >0)
    {
        isPositivePar = false;
        return true;
    }
    else
    {
        isPositivePar = false;
        return false;
    }
}
```

Solución:



Caminos:
1-2-3-7
1-4-5-7
1-4-6-7
1-2-4-5-7

V(G)=4

Nº Áreas: 4

3 nodos predicado +1 = 4

9 aristas – 7 nodos +2= 4

Camino	Entradas	Salidas	
	Number	isPositivaNumber	Return
1-2-3-7	0	True	True
1-4-5-7	No es posible probarlo		
1-4-6-7	-5	False	False
1-2-4-5-7	5	False	True

Ejercicio 3 (3 puntos). Aplicad la técnica de la partición equivalente de caja negra para diseñar los casos de prueba asociados al siguiente método. En concreto, obtened una tabla con las clases de equivalencia numeradas con las

siguientes cuatro columnas: entrada bajo consideración, clases válidas, clases inválidas, heurística aplicada. Además, dos tablas con los casos de prueba asociados, una para las clases válidas y otra para las clases inválidas.

```
public int AsignarPuestoITV (string matricula, string tipoVehiculo, string provincia)
```

Donde:

- string **matricula**: una cadena de 7 caracteres, donde los tres primeros son letras mayúsculas, y los cuatro últimos dígitos.
- string **tipoVehiculo**: una cadena de caracteres que admite los valores “coche”, “moto” o “camión”.
- string **provincia**: una cadena de dos dígitos, que no puede ser “00” ni mayor a “51”.

La salida de este método debe ser:

- 1, si el tipo de vehículo es “coche” y la provincia menor que “26”.
- 2, si el tipo de vehículo es “coche” y la provincia es “26” o superior.
- 3, si el tipo de vehículo es “moto” y la provincia menor que “26”.
- 4, si el tipo de vehículo es “moto” y la provincia es “26” o superior.
- 5, si el tipo de vehículo es “camión” y la provincia menor que “26”.
- 6, si el tipo de vehículo es “camión” y la provincia es “26” o superior.
- En cualquier otro caso: ArgumentException(“Entrada Incorrecta”)

Solución:

Entrada	Clases Válidas	Clases Inválidas	Heurística aplicada
Matrícula	(1) Cadena de 7 char, 3 primeros mayúsculas y últimos números	(2) < 7 char (3) >7char (4) 3 primeros no mayúsculas (5) Últimos no número	Numero finito de valores Boolean Boolean
tipoVehículo	(6)Coche (7)Moto (8)Camión	(9)Otro	Conjunto de valores aceptados
provincia	(10) ['00','25'] (11) ['26','51']	(12) <='00' (13) >'51'	Rango de valores Clases Menores Boolean

Clases válidas cubiertas	Entrada	Salida
(1)(6)(10)	matrícula= “MVP0246”; tipoVehículo=“coche”; Provincia=“25”	1
(1)(7)(11)	matrícula= “MVP0246”; tipoVehículo=“moto”; Provincia=“26”	4
(1)(8)(10)	matrícula= “MVP0246”; tipoVehículo=“camión”; Provincia=“25”	5

Clases inválidas cubiertas	Entrada	Salida
(2)(6)(10)	matrícula= “MVP024”; tipoVehículo=“coche”; Provincia=“25”	ArgumentException(“Entrada Incorrecta”)
(3)(7)(11)	matrícula= “MVP02468”; tipoVehículo=“moto”; Provincia=“26”	ArgumentException(“Entrada Incorrecta”)
(4)(8)(10)	matrícula= “mvp0246”; tipoVehículo=“camión”; Provincia=“25”	ArgumentException(“Entrada Incorrecta”)
(5)(6)(10)	matrícula= “MVPW246”; tipoVehículo=“coche”; Provincia=“25”	ArgumentException(“Entrada Incorrecta”)
(1)(9)(10)	matrícula= “MVP0246”; tipoVehículo=“cualquierValor ”; Provincia=“25”	ArgumentException(“Entrada Incorrecta”)
(1)(6)(12)	matrícula= “MVP0246”; tipoVehículo=“coche”; Provincia=“00”	ArgumentException(“Entrada Incorrecta”)
(1)(6)(13)	matrícula= “MVP0246”; tipoVehículo=“coche”; Provincia=“52”	ArgumentException(“Entrada Incorrecta”)