

Obligatorio 2023

Contexto

Después de la PoC (Proof of Concept) realizada para el primer parcial, la FIA (Federación Internacional del Automovilismo) quedó impresionada con los resultados. Es por esto que ahora, le está solicitando a la facultad de Ingeniera realizar una nueva PoC, pero enfocada en la simulación de carreras del campeonato de F1.

Documentación

Como documentación se pide entregar un diagrama UML de clase de la solución implementada. Para realizar dicho diagrama, se recomienda utilizar la herramienta gratuita draw.io. Deben elegir la opción de OneDrive y registrarse con el mail de la facultad.

Luego, en un archivo README.md, deben hacer una breve descripción de lo que trata la aplicación y explicar cómo ejecutarla.

Funcionalidades

Gestión

En primer lugar, se necesita modelar e implementar un sistema de gestión que permita agregar distintos equipos y los empleados para cada uno de los equipos. Para registrar un empleado se necesita su id (8 números sin guión, con validación), nombre, edad, nacionalidad, fecha de nacimiento y salario. Para simplificar los tipos de empleados a registrar pueden ser:

- **Pilotos:** son quienes corren las carreras. Los equipos tienen un piloto de reserva, en caso de que alguno de los pilotos titulares no pueda correr. Vale recordar que los equipos tienen dos pilotos titulares y uno solo de reserva. Los pilotos cuentan con unos atributos extra: *score* (número entero del 1 al 99) el cual representa la habilidad de cada piloto, número de auto, puntaje en el campeonato y si está lesionado o no.

- **Mecánicos:** son quienes se encargan de preparar, arreglar y ajustar los autos. A su vez, asisten a los pilotos durante la carrera. Un equipo necesita al menos 8 mecánicos para poder correr una carrera. Por último, los mecánicos tienen un atributo score (representa que tan bueno es el auto y es un numero entero entre 1 y 99).
- **Director de equipo:** se encarga de las decisiones más globales y gestiona todo el equipo.

Por otro lado, los equipos tienen un modelo de auto para cada temporada que también deberá ser modelado en la aplicación solicitado. Los datos de un auto son: el modelo, su score (mismo criterio que en los casos anteriores) y color.

Por último, para representar los equipos se requiere ingresar: el nombre del equipo, país de origen, y año de creación. A su vez, se debe poder asociarle los empleados creados y un solo modelo de auto.

Simulación de carrera

Pasando a la simulación de carrera, la modalidad es la siguiente:

- La carrera se dispone entre todos los pilotos titulares registrados y asociados a un equipo o, en caso de que un titular esté incapacitado, correrá el piloto de reserva del equipo. Si por algún motivo ambos pilotos de un equipo no pueden correr, la carrera se llevará adelante con menos pilotos.
- Antes de comenzar la carrera se puede registrar distintos imprevistos que sucedan durante la misma:
 - Abandono de un piloto
 - Error en los pits para un piloto (5pts)
 - Penalidad para un piloto por infringir una norma (8pts)
- Para definir el orden resultado de la carrera, se utilizará la siguiente fórmula:
 - Si el piloto abandona: $score_final = 0$
 - Si no abandona:

$$score_final = suma_score_mecanicos + score_auto + score_piloto - 5*cantidad_errores_en_pits - 8*cantidad_penalidad_infringir_norma$$

Entonces, las posiciones finales de la carrera van a estar determinado por el `score_final` de cada piloto ordenado de manera descendiente.

- Finalizada la carrera se adjudican los puntos a los pilotos de la siguiente manera:

1. 25pts
2. 18pts
3. 15pts
4. 12pts
5. 10pts
6. 8pts
7. 6pts
8. 4pts
9. 2pts
10. 1pt

El resto no suma puntos

Consultas de interés

La aplicación deberá tener una sección donde se puedan realizar una serie de consultas como:

- Top 10 pilotos con más puntos en el campeonato
- Resumen campeonato de constructores (equipos): El puntaje de un equipo corresponde a la suma de puntos que tiene cada uno de sus pilotos.
- Top 5 pilotos mejores pago
- Top 3 pilotos más habilidosos
- Retornar jefes de equipo: Se debe retornar el jefe del equipo y el equipo que lidera, ordenados por nombre del jefe de manera ascendente.

En todos los demás casos se deberá ordenar de manera descendiente por el criterio solicitado.

Experiencia de usuario

Las funcionalidades se deberán representar con un menú a través de la consola de comandos. Deberá tener un ciclo en su clase principal el cual soportará las siguientes acciones desplegadas de la siguiente forma:

1. Alta de empleado
2. Alta de auto
3. Alta de equipo
4. Simular carrera
5. Realizar consultas
6. Finalizar programa

Alta de empleado:

- Ingrese cedula:
- Ingrese nombre:
- Ingrese fecha de nacimiento (DD/MM/AAAA):
- Ingrese nacionalidad:
- Ingrese salario:
- Ingrese cargo (validar que sea alguno de los mencionados previamente):
 - 1 Piloto
 - 2 Piloto de reserva
 - 3 Mecánico
 - 4 Jefe de equipo

Para los casos de pilotos luego solicitar:

- Ingrese score:
- Ingrese número de auto:

Para los casos de mecánicos luego solicitar:

- Ingrese score:

Alta de auto

- Ingrese modelo:
- Ingrese año:
- Ingrese score:

Alta equipo

- Ingrese nombre del equipo:

- Ingrese modelo de auto:
- Ingrese cedula del empleado: (realizar doce veces: dos pilotos de titulares, un piloto de reserva, un jefe de equipo y ocho mecánicos)

Simular carrera

- Ingrese nro auto de todos los pilotos que abandonan separado por coma: *(si es vacío es que ningún piloto abandona)*
- Ingrese nro de auto de todos los pilotos que comente error en pits: *(si es vacío es que ningún piloto abandona y se pueden repetir valores)*
- Ingrese nro de auto de todos los pilotos que reciben penalidad: *(si es vacío es que ningún piloto abandona y se pueden repetir valores)*
- Simular carrera. -> Output: print con pilotos ordenados por orden en el que finalizan la carrera.

Realizar consultas

- Top 10 pilotos con más puntos en el campeonato
- Resumen campeonato de constructores (equipos).
- Top 5 pilotos mejores pago
- Top 3 pilotos más habilidosos
- Retornar jefes de equipo

Finalizar programa

Si el usuario ingresa la opción 6, este debe finalizar.

Aclaraciones:

- Para todos los tipos de datos, validar que la entrada sea del tipo correcto, de lo contrario debe lanzarse una excepción.
- Siempre que se lance una excepción se debe imprimir en consola el error y retornar al menú principal.

Entrega

El proyecto será subido a un repositorio público en [GitHub](https://github.com) el cual tendrá por nombre "obligatorio-p1-2023".

En la configuración del proyecto, se deberá agregar al docente del curso como colaborador.