



universidad  
cenfotec\_  
La U de la informática

# Fundamentos de Programación

Proyecto del curso.  
Cuatrimestre 2020-3.

## Introducción

El **quidditch** es un deporte no tradicional que tiene sus raíces en el deporte ficticio mencionado en la saga de libros de *Harry Potter*. La versión *muggle* fue creada en 1994 en la Universidad de Middlebury en Middlebury, Vermont, Estados Unidos. Ha crecido y se ha desarrollado como deporte propio, y su última versión es del 2005. Desde el 2005 se han jugado 15 torneos, uno cada año.

El juego consiste en que dos equipos de siete jugadores juegan en un campo del tamaño de una pista de hockey, el objetivo final es tener más puntos que el equipo rival. El partido empieza con la *quaffle* (un balón de voleibol) alineada en el centro del campo y los jugadores alineados en sus respectivas porterías. Después de que la *snitch* (una pelota de tenis atada a un jugador) desaparece del campo de visión, el árbitro empieza el partido. Cada gol anotado con la *quaffle* vale 10 puntos, se otorgan 30 puntos al equipo que captura la *snitch* y el equipo con mayor número de puntos gana. El juego termina cuando uno de los equipos captura la *snitch*. Pero si al capturar la *snitch* los equipos empatan en puntos, el partido se va a tiempos extra.

La USQ (*United States Quidditch*), la organización nacional de Estados Unidos de Quidditch, clasifica los equipos de cada torneo en orden descendente, empezando por el equipo de mayor puntaje total. Este puntaje total se calcula sumando tres puntajes:

- El puntaje de los partidos
- El puntaje por número de juegos
- y El puntaje por número de oponentes.

Estos puntajes se calculan teniendo en cuenta las siguientes condiciones:

<b>Puntaje de los partidos:</b>	Total de puntos obtenidos en todos los partidos : Se obtiene de los puntos obtenidos con la <i>quaffle</i> y los puntos obtenidos si se atrapa la <i>snitch</i>		
<b>Puntaje por número de juegos:</b>	1, si número de juegos $\geq 5$		$\sqrt{\text{numJuegos}}/2.25$ , si el número de juegos $< 5$
<b>Puntaje por número de oponentes:</b>	1/3, si número de oponentes = 1	2/3, si número de oponentes = 2	1, si número de oponentes $\geq 3$
<b>Puntaje total:</b>	<b>Puntaje de los partidos + Puntaje por número de juegos + Puntaje por número de oponentes</b>		

## Implementación

Se quiere entonces hacer un sistema que trabaje con la información de un torneo de Quidditch. Para esto se tienen 5 arreglos: un arreglo llamado **equipos** que guarda en *equipos[i]* el nombre del equipo *i* del torneo, un arreglo llamado **numJuegos** que guarda en *numJuegos[i]* el número de juegos ejecutados por el equipo *i* en el torneo, un arreglo llamado **numOponentes** que guarda en *numOponentes[i]* el número de oponentes diferentes con los que ha jugado el equipo *i* en el torneo, un arreglo llamado **resultados** que guarda los resultados del equipo *i* en cada partido del torneo, y un arreglo llamado **puntajeTotal** que guarda el puntaje total acumulado del equipo *i* en el torneo, el cual se calcula sumando los tres puntajes: puntaje de los partidos + puntaje por número de juegos + puntaje por número de oponentes, del equipo *i*. No es necesario que ninguno de los arreglos estén ordenados.

El siguiente es un ejemplo de cómo serían los arreglos en un torneo de Quidditch de 6 equipos:

### equipos

0	1	2	3	4	5
"Maryland"	"Texas Hill"	"New York"	"The Warriors"	"The Lost Boys"	"Boom Train"

### numJuegos

0	1	2	3	4	5
7	5	4	3	2	1

### numOponentes

0	1	2	3	4	5
5	4	3	3	2	1

### Resultados

0	1	2	3	4	5
"Maryland vs Texas Hill 30-80* Maryland vs New York 100*-20 Maryland vs The Warriors 110*-50 Maryland vs The Lost Boys 140*-60 Maryland vs Boom Train 120*-60 Maryland vs Texas Hill 20-70* Maryland vs New York 120*-60"	"Texas Hill vs Maryland 80*-30 Texas Hill vs Maryland 70*-20 Texas Hill vs New York 120-40* Texas Hill vs The Warriors 140*-20 Texas Hill vs The Lost Boys 140-60*"	"New York vs Maryland 20-100* New York vs Maryland 60-120* New York vs Texas Hill 40*-120 New York vs The Warriors 100*-60"	"The Warriors vs Maryland 50-110* The Warriors vs Texas Hill 20-140* The Warriors vs New York 60-100*"	"The Lost Boys vs Maryland 60-140* The Lost Boys vs Texas Hill 60*-140"	"Boom Train vs Maryland 60-120*"

## puntajeTotal

0	1	2	3	4	5
642	552	221.89	131.77	121.29	60.77

El puntaje total de cada equipo se calcula teniendo en cuenta lo mencionado anteriormente:

Equipos Puntaje	Maryland	Texas Hill	New York	The Warriors	The Lost Boys	Boom Train
Puntaje de los partidos	$30 + 100 + 110 + 140 + 20 + 120 + 120$	$80 + 70 + 120 + 140 + 140$	$20 + 60 + 40 + 100$	$50 + 20 + 60$	$60 + 60$	60
Puntaje por número de juegos	1	1	0.89	0.77	0.63	0.44
Puntaje por número de oponentes	1	1	1	1	0.66	0.33

## Entregable 1: 10%

Haga un programa que lea el nombre de un equipo, el puntaje de todos los partidos, el número de juegos ejecutados y el número de oponentes diferentes, y calcule el puntaje total que le corresponde al equipo. El puntaje total se debe calcular teniendo en cuenta las condiciones descritas anteriormente.

El programa debe tener un comportamiento parecido al siguiente:

```
Entradas:  
Nombre del equipo: Maryland  
Puntaje de todos los partidos: 640  
Número de juegos ejecutados: 7  
Número de oponentes diferentes: 5  
  
Salidas:  
El puntaje total del equipo Maryland es: 642
```

La rúbrica con la que se calificará esta Entrega 1 es la siguiente:

- Diagrama general: 5%
- Tabla de entradas y salidas: 10%
- Diagrama explicativo: 10%
- Diagrama de flujo final: 20%
- Casos de prueba: 5% (incluir 2 casos)
- Programa en Java: 50%

El equipo debe entregar lo siguiente:

- Un archivo llamado NombreEstudiante1\_NombreEstudiante2\_Entrega1.pdf, que contenga la documentación solicitada para la entrega.
- Un archivo llamado NombreEstudiante1\_NombreEstudiante2\_Entrega1.java, que es el programa que implementa lo solicitado por la entrega.
- (\*)Opcional: un archivo llamado NombreEstudiante1\_NombreEstudiante2\_Entrega1.drawio, que contiene el diagrama de flujo final, en caso de que el equipo considere no entregar dicho diagrama en la plantilla del curso.

## Entregable 2: 15%

Haga un programa que lea el nombre de un equipo, el número de juegos ejecutados por el equipo y la información correspondiente a cada uno de los juegos, e imprima un reporte de resultados de todos los juegos del equipo, así como el puntaje total de los partidos, obtenido por el equipo. Para cada uno de los juegos se debe preguntar el nombre del equipo oponente, el marcador de cada uno, así como quién atrapó la snitch. El equipo que atrapó la snitch debe aparecer con un \* (asterisco) indicado en el marcador correspondiente, en el reporte de resultados.

El programa debe tener un comportamiento parecido al siguiente:

```
Entradas:
Equipo: Maryland
Número de juegos: 7

Juego 1 contra: Texas Hill
Marcador de Maryland: 30
Atrapó la snitch: no
Marcador de Texas Hill: 80
Atrapó la snitch: si

Juego 2 contra: New York
Marcador de Maryland: 100
Atrapó la snitch: si
Marcador de New York: 20
Atrapó la snitch: no

Juego 3 contra: The Warriors
Marcador de Maryland: 110
Atrapó la snitch: si
Marcador de The Warriors: 50
Atrapó la snitch: no

Juego 4 contra: The Lost Boys
Marcador de Maryland: 140
Atrapó la snitch: si
Marcador de The Lost Boys: 60
Atrapó la snitch: no

Juego 5 contra: Boom Train
Marcador de Maryland: 120
Atrapó la snitch: si
Marcador de Boom Train: 60
Atrapó la snitch: no

Juego 6 contra: Texas Hill
Marcador de Maryland: 20
Atrapó la snitch: no
Marcador de Texas Hill: 70
Atrapó la snitch: si

Juego 7 contra: New York
Marcador de Maryland: 120
Atrapó la snitch: si
Marcador de New York: 60
Atrapó la snitch: no
```

Salidas: Se debe producir el siguiente mensaje o cadena de caracteres e imprimirlo en pantalla, así como el puntaje total del equipo, así:

```
Maryland vs Texas Hill 30-80*
Maryland vs New York 100*-20
Maryland vs The Warriors 110*-50
Maryland vs The Lost Boys 140*-60
Maryland vs Boom Train 120*-60
Maryland vs Texas Hill 20-70*
Maryland vs New York 120*-60
```

El puntaje total de partidos de Maryland es: 640

La rúbrica con la que se calificará esta Entrega 2 es la siguiente:

- Diagrama general: 5%
- Tabla de entradas y salidas: 10%
- Diagrama explicativo: 10%
- Diagrama de flujo final: 20%
- Casos de prueba: 5% (incluir 2 casos)
- Programa en Java: 50%

El equipo debe entregar lo siguiente:

- Un archivo llamado NombreEstudiante1\_NombreEstudiante2\_Entrega2.pdf, que contenga la documentación solicitada para la entrega.
- Un archivo llamado NombreEstudiante1\_NombreEstudiante2\_Entrega2.java, que es el programa que implementa lo solicitado por la entrega.
- (\*)Opcional: un archivo llamado NombreEstudiante1\_NombreEstudiante2\_Entrega2.drawio, que contiene el diagrama de flujo final, en caso de que el equipo considere no entregar dicho diagrama en la plantilla del curso.

## Entrega Final: 15% + Defensa del proyecto: 10%

Para la entrega final el equipo debe entregar todas las funcionalidades descritas más adelante, en la fecha definida para ello. Adicionalmente, debe hacer la defensa ante su profesor(a) del curso, para que esta entrega final pueda calificarse, de lo contrario no tendrá la nota correspondiente.

Para la entrega final del proyecto el equipo debe hacer un menú que tenga las siguientes funcionalidades:

1. Iniciar el programa: inicializar todos los arreglos, para esto debe leer del usuario el total de equipos del torneo.
2. Registrar la información de los arreglos: equipos, *numJuegos* y *numOponentes*. El programa debe preguntarle al usuario la información correspondiente para llenar los tres primeros arreglos.
3. Construir el reporte de resultados de los juegos para cada equipo y calcular el puntaje total de cada equipo del torneo y llenar los arreglos *resultados* y *puntajeTotal*.
4. Imprimir el reporte de resultados de cada uno de los equipos.
5. Imprimir el puntaje total de cada uno de los equipos.
6. Calcular el promedio de oponentes diferentes que tienen todos los equipos del torneo.
7. Imprimir el nombre del equipo que ha jugado más partidos en todo el torneo (Si hay varios equipos, el sistema debe imprimir el primer equipo mayor).
8. Imprimir el nombre del equipo que con menor puntaje en todo el torneo (Si hay varios equipos, el sistema debe imprimir el primer equipo menor).
9. Imprimir los nombres de los equipos que han jugado 5 partidos o más.
10. Imprimir el nombre de los equipos que tengan más oponentes que el promedio de oponentes de todo el torneo.

Por cada una de las funcionalidades, el equipo debe implementar como mínimo una rutina. Los arreglos se trabajan como variables locales al programa principal, por lo que deben usarse como parámetros en las diferentes rutinas. El equipo puede usar lo desarrollado en el entregable 1 y 2, para hacer la entrega final del proyecto.

La rúbrica con la que se calificará la Entrega Final es la siguiente:

- Diagrama general: 1%
- Tabla de entradas y salidas: 10%
- Diagrama explicativo: 4%
- Diagrama de flujo final: 20%
- Casos de prueba: 5%
- Programa en Java: 60%





universidad  
cenfotec\_

La U de la informática