

UCENFOTEC  
SEDE CENTRAL

ESTRUCTURAS DE DATOS 1

PROYECTO FINAL  
PREGUNTADOS

INTEGRANTES  
BRANDON RUTCH MURILLO  
EVELYN SEQUEIRA NAVARRO

PROFESOR  
CHRISTIAN SIBAJA

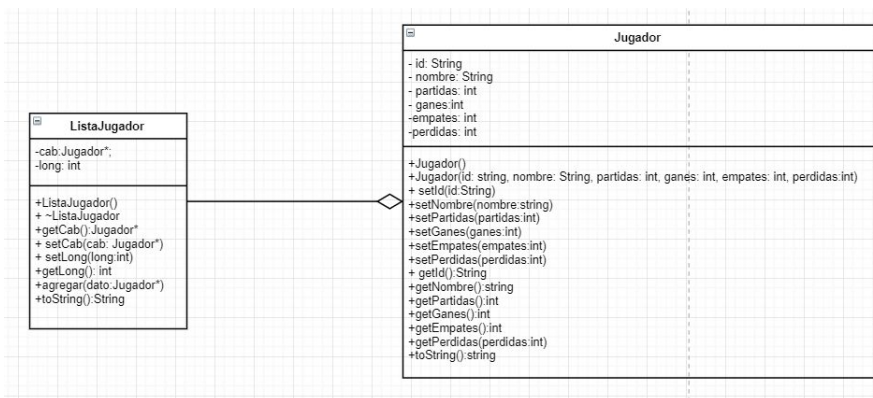
III CUATRIMESTRE 2019

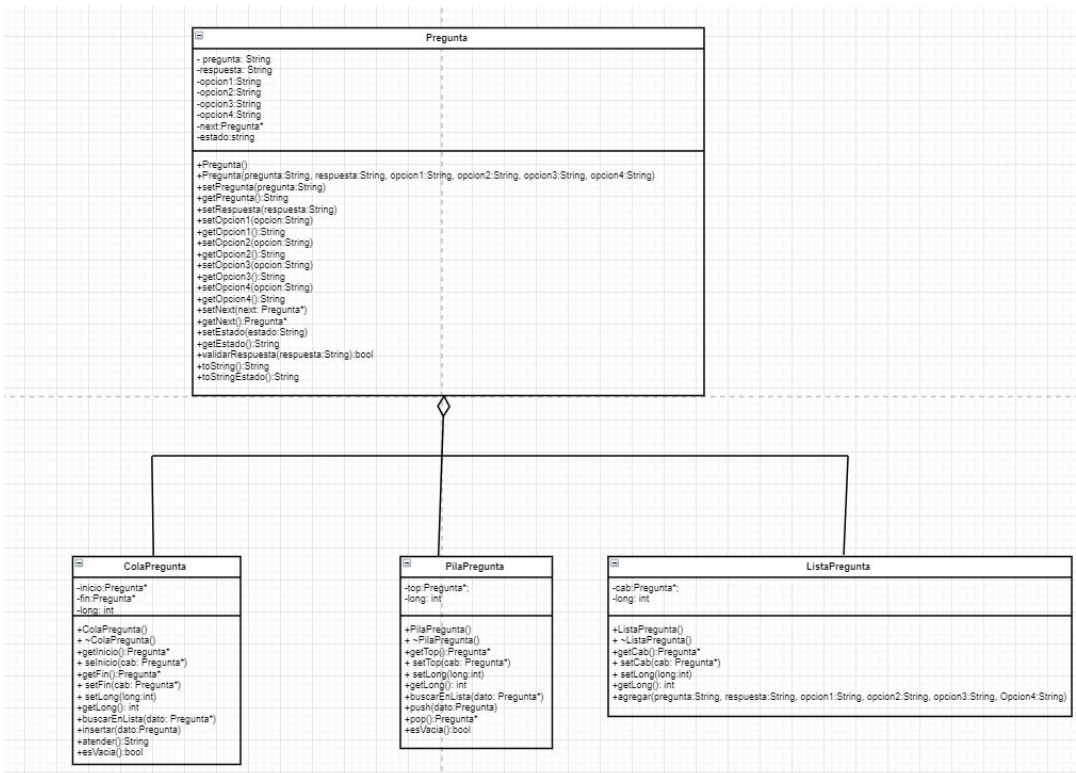
## Descripción del problema

El juego se llama Preguntados. Dicho juego es para dos personas y consiste en la dinámica de responder preguntas sobre cultura general. Cada jugador debe responder diez preguntas alternando entre cada jugador, al final ganará aquel que haya contestado correctamente la mayor cantidad de preguntas.

■ Diagramas completos para las siguientes partes de la aplicación:

### i. UML general de la aplicación

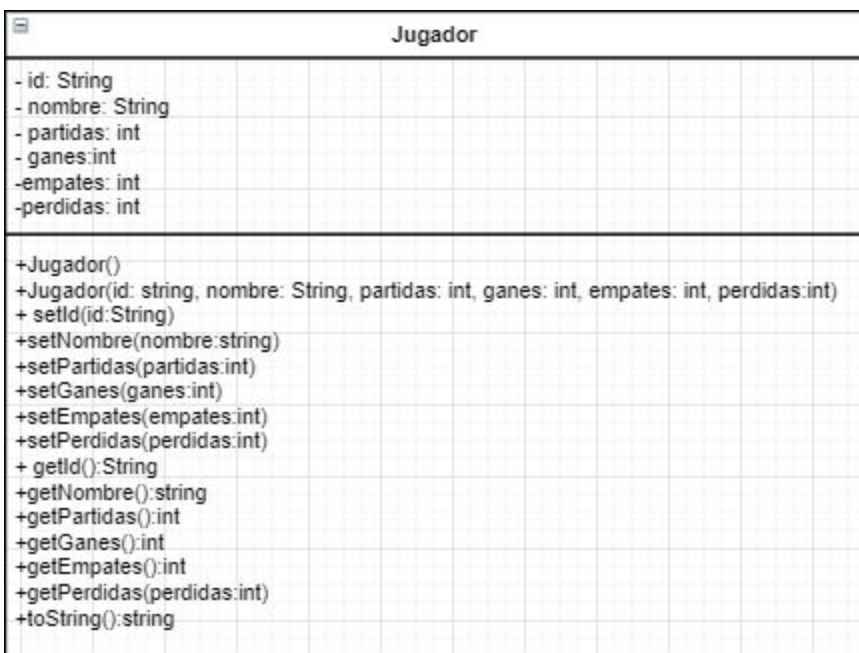




ii. TAD sólo de las estructuras del proyecto (listas, colas, pilas)

## TAD Jugador:

### 1. REPRESENTACIÓN ABSTRACTA



## **2. INVARIANTE**

Los atributos id y nombre deben ser de tipo string. Los atributos partidas, ganés, empates y pérdidas deben ser de tipo int, es decir, número entero positivo.

## **3. OPERACIONES**

Jugador: nada->nada(constructora)// constructor por defecto de Jugador

Jugador: string, string, int, int, int, int -> nada(constructora) // constructor con parámetros

setId: id->nada(modificadora)// modifica el id del Jugador

setNombre: string ->nada (modificadora) //modifica el nombre del Jugador

setPartidas: int->nada (modificadora) //modifica el número de partidas del jugador

setGanes: int->nada (modificadora) //modifica el número de ganés del jugador

setEmpates: int->nada (modificadora) //modifica el número de empates del jugador

setPérdidas: int->nada (modificadora) //modifica el número de pérdidas del jugador

getId: nada->string (analizadora) // retorna el id del jugador

getNombre: nada->string (analizadora) // retorna el nombre del jugador

getPartidas: nada->int(analizadora) // retorna el número de partidas del jugador

getGanes: nada->int(analizadora) // retorna el número de ganés del jugador

getEmpates: nada->int(analizadora) // retorna el número de empates del jugador

getPérdidas: nada->int(analizadora) // retorna el número de pérdidas del jugador

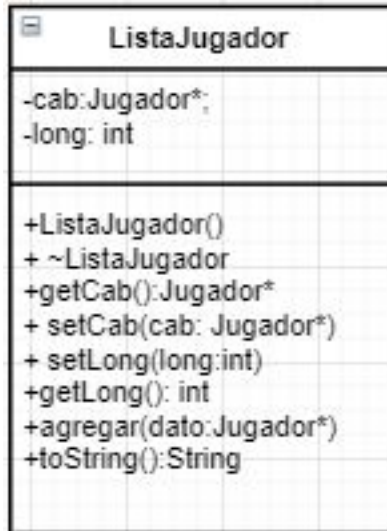
toString: nada->string(analizadora) // retorna un string con la información completa del jugador como su id, nombre, partidas jugadas, partidas ganadas, partidas empatadas y partidas pérdidas.

## **4. MANEJO DE ERROR**

Se revisará que las funciones y procedimientos que utilicen a modo de parámetros y retornos los atributos id y nombre sean de tipo string; y aquellas que usen los atributos partidas, ganés, pérdidas y empates sean de tipo número entero positivo.

## TAD ListaJugador:

### 1. REPRESENTACIÓN ABSTRACTA



### 2. INVARIANTE

El atributo **cab** debe ser un puntero de tipo **Jugador** y el atributo **long** debe ser un número entero positivo.

### 3. OPERACIONES

**ListaJugador nada->nada** (Constructora) //constructor por defecto

**~ListaJugador nada->nada** (Destructor) //destructor por defecto

**getCabnada->Jugador\*** (analiadora) //retorna el puntero cabeza

**setCab: nada-> Jugador\*** (modificadora) //Asigna el puntero cabeza

**setLongint->nada** (modificadora) // Asigna la longitud

**getLong:nada-> int** (analizadora) //Retorna la longitud

**Agregar: Jugador\*->nada** (modificadora) //Recibe el puntero de un jugador y lo agrega

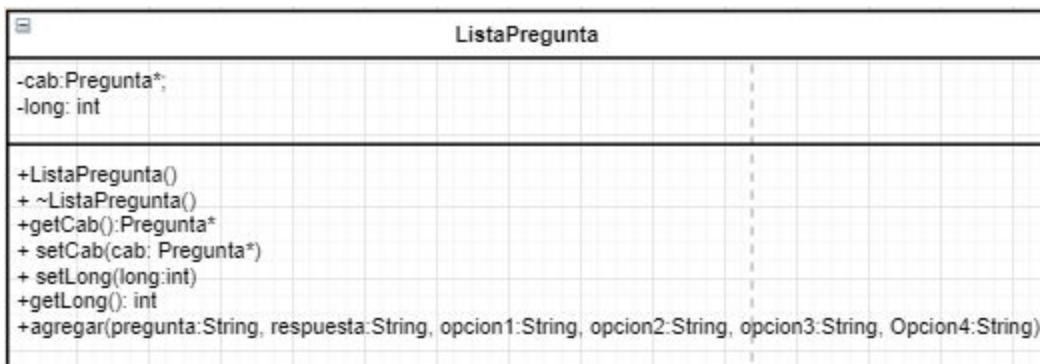
**toString:nada->String** (analizadora) //retorna la información de cada nodo

### 4. MANEJO DE ERROR

Se revisará que los procedimientos y funciones que utilicen el dato cab sea de tipo puntero Jugador, así como se revisará que los procedimientos y funciones que usen el dato long sea de tipo número entero positivo.

## TAD ListaPregunta:

### 1. REPRESENTACIÓN ABSTRACTA



### 2. INVARIANTE

El atributo cab debe ser un puntero de tipo Pregunta y el atributo long debe ser un número entero positivo.

### 3. OPERACIONES

ListaPregunta nada->nada (Constructora) //constructor por defecto

~ListaPregunta nada->nada (Destructor) //destructor por defecto

getCab : nada->Preguntar\* (analiadora) //retorna el puntero cabeza

setCab: nada-> Pregunta\* (modificadora) //Asigna el puntero cabeza

setLongint->nada (modificadora) // Asigna la longitud

getLong:nada-> int (analizadora) //Retorna la longitud

Agregar: string, string, string, string, string, string->nada (modificadora) //Recibe la información del objeto pregunta crea un TAD de ese tipo y lo añade a la lista

### 4. MANEJO DE ERROR

Se revisará que las funciones que utilicen el atributo long de tipo int utilicen un entero positivo en los procedimientos, retornos y parámetros. Por otro lado se revisará que los procedimientos y variables que utilicen el dato cab sea de tipo Pregunta\*.

## **TAD PilaPregunta:**

### **1. REPRESENTACIÓN ABSTRACTA**

PilaPregunta	
-top:Pregunta*;	
-long: int	
+PilaPregunta()	
+ ~PilaPregunta()	
+getTop():Pregunta*	
+ setTop(cab: Pregunta*)	
+ setLong(long:int)	
+getLong(): int	
+buscarEnLista(dato: Pregunta*)	
+push(dato:Pregunta)	
+pop():Pregunta*	
+esVacia():bool	

### **2. INVARIANTE**

El atributo int debe ser un dato entero positivo y el atributo top debe ser de tipo puntero de Pregunta.

### **3. OPERACIONES**

PilaPregunta: nada ->nada(constructora)//Constructor por defecto

~PilaPregunta: nada->nada(destructora)//Destruye el objeto

getTop: nada->Pregunta\*(analizadora)//Retorna el tope de la pila

setTop: Pregunta\* -> nada (modificadora)// Cambia el valor del tope de la pila

setLong: int-> nada (modificadora)// Cambia el valor de la longitud de la pila

getLong(): nada->int (analizadora)// Retorna el valor de la longitud de la pila

buscarEnLista: Pregunta\*->nada (modificadora)// De una lista de preguntas agrega las preguntas a la pila

Push: Pregunta\*->nada (modificadora)//Agrega una pregunta a la pila

Pop: nada->Pregunta\*(modificadora)//Elimina un elemento de la pila

esVacia: nada->bool(analizadora)//Retorna un booleano según la lista esté vacía o no

#### 4. MANEJO DEL ERROR

Se revisará que las funciones que utilicen el atributo long de tipo int utilicen un entero positivo en los procedimientos, retornos y parámetros. Por otro lado se revisará que los procedimientos y variables que utilicen el dato top sea de tipo Pregunta\*.

#### TAD ColaPregunta:

##### 1. REPRESENTACIÓN ABSTRACTA

ColaPregunta		
-inicio:Pregunta*		
-fin:Pregunta*		
-long: int		
<hr/>		
+ColaPregunta()		
+ ~ColaPregunta()		
+getInicio():Pregunta*		
+ setInicio(cab: Pregunta*)		
+getFin():Pregunta*		
+ setFin(cab: Pregunta*)		
+ setLong(long:int)		
+getLong(): int		
+buscarEnLista(dato: Pregunta*)		
+insertar(dato:Pregunta)		
+atender():String		
+esVacia():bool		

##### 2. INVARIANTE

El atributo inicio y el atributo fin debe ser un puntero tipo Pregunta. El atributo long debe ser un número entero positivo.

##### 3. OPERACIONES

ColaPregunta nada->nada (Constructora) //constructor por defecto

~ColaPregunta nada->nada (Destructor) //destructor por defecto

getInicio : nada->Preguntar\* (analiadora) //retorna el puntero del inicio

setInicio: nada-> Pregunta\* (modificadora) //Asigna el puntero del inicio

getFin : nada->Preguntar\* (analiadora) //retorna el puntero final

setFin: nada-> Pregunta\* (modificadora) //Asigna el puntero final

setLong->nada (modificadora) // Asigna la longitud



getLong:nada-> int (analizadora) //Retorna la longitud

insertar: Pregunta\*->nada (modificadora) //Recibe el puntero de la pregunta y la asigna de último en la cola

Atender: nada-> nada (analizadora)// retorna la pregunta y su respuesta correcta

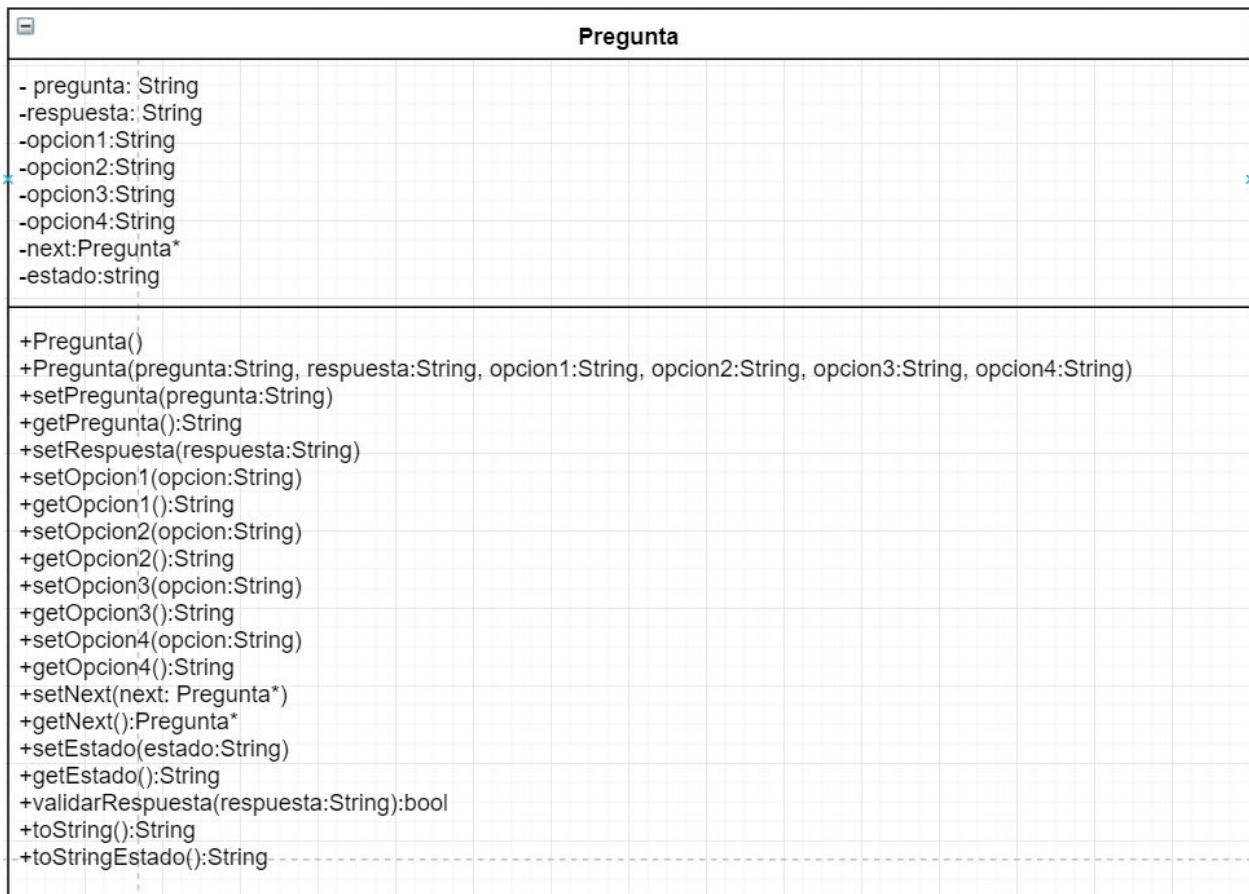
esVacio() nada-> bool (analizadora) // retorna true si la longitud es 0 y false si es mayor a 0

#### **4. MANEJO DEL ERROR**

Se revisará que las funciones que utilicen el atributo long de tipo int utilicen un entero positivo en los procedimientos, retornos y parámetros. Por otro lado se revisará que los procedimientos y variables que utilicen el dato inicio o el dato fin sea de tipo Pregunta\*.

## Pregunta:

### 1. REPRESENTACIÓN ABSTRACTA



### 2. INVARIANTE

Los atributos pregunta, respuesta, opcion1, opcion2, opcion3, opcion4 y estado deben ser tipo string. El atributo next debe ser un puntero de tipo Pregunta\*

### 3. OPERACIONES

Pregunta: nada->nada(constructora) // Constructor por defecto del TAD Pregunta

Pregunta: String, String, String, String, String,String->nada(constructora) // Constructor con parámetros.

setPregunta:string->nada (modificadora)// Modifica el valor de la pregunta

getPreguntanada->string (analizadora) // Retorna la pregunta

setRespuesta:string->nada (modificadora)// Modifica el valor de la respuesta

setOpcion1:string->nada (modificadora)// Modifica el valor de la opción 1  
getOpcion1 nada->string (analizadora) // Retorna la opcion 1  
setOpcion2:string->nada (modificadora)// Modifica el valor de la opción 2  
getOpcion2 nada->string (analizadora) //Retorna la opción 2  
setOpcion3:string->nada (modificadora)// Modifica el valor de la opción 3  
getOpcion3 nada->string (analizadora) // Retorna la opción 3  
setOpcion4:string->nada (modificadora)// Modifica el valor de la opción 4  
getOpcion4 nada->string (analizadora) // Retorna la opción 4  
setNext: Pregunta\* ->nada (modificadora)// Modifica el valor del atributo next  
getNext: nada->Pregunta\*(analizadora) // Retorna el puntero tipo Pregunta ubicado en al tributo next  
setEstado:string->nada (modificadora)// Modifica el valor del estado de la pregunta  
getEstado: nada->string (analizadora) // Retorna el estado de la pregunta.  
validarRespuesta: string->bool(analizadora)//Retorna un dato tipo bool según la respuesta a la pregunta sea correcta o no  
toString: nada->String(analizadora)//Retorna la información de la pregunta como lo es pregunta, respuesta, las 4 opciones y el estado.  
toStringEstado: nada->String(analizadora)//Retorna el estado de la pregunta con valores de incorrecto o correcto

#### **4. MANEJO DEL ERROR**

Se revisará que las funciones que utilicen los atributos pregunta, respuesta, opcion1, opcion2, opcion3, opcion4 y estado sean estrictamente un string en todos los procedimientos, retornos y parámetros. Por otro lado se revisará que los procedimientos y variables que utilicen el dato next sea de tipo Pregunta\*.

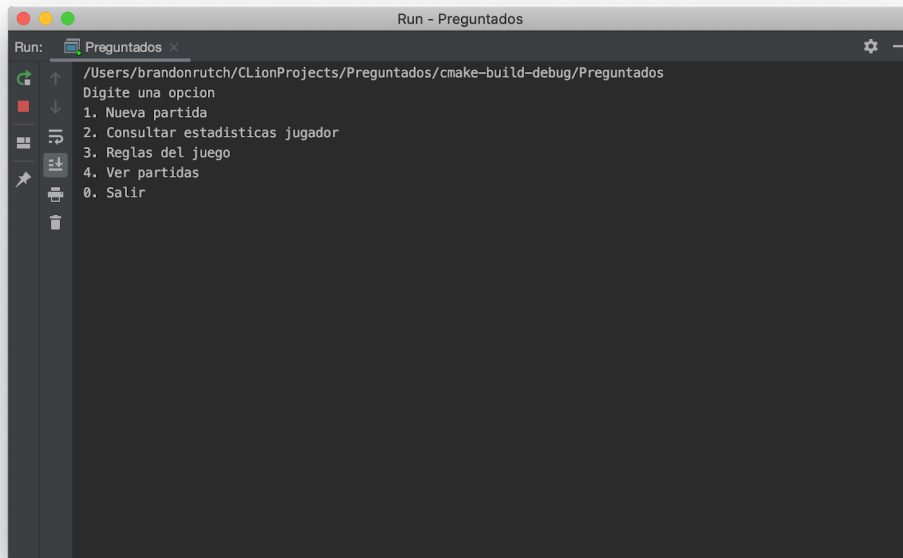
■ Indicar estado de la aplicación (totalmente funcional, parcialmente funcional, etc.)

Estado: Totalmente Funcional.

- Pequeño manual de usuario con pantallazos del funcionamiento.

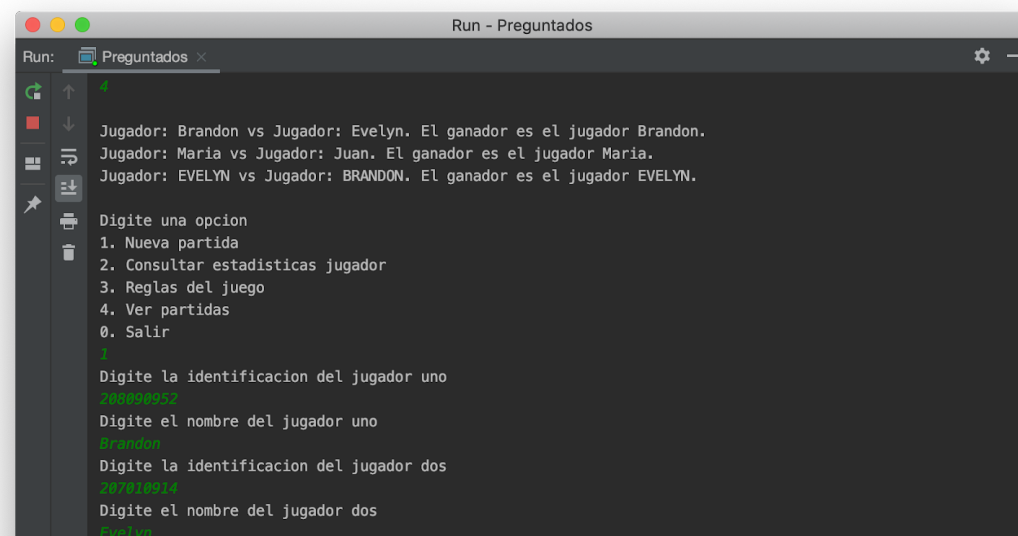
## MANUAL DE USO

Al iniciar el juego se muestra un menú con las opciones con las que cuenta el jugador



Iniciar juego:

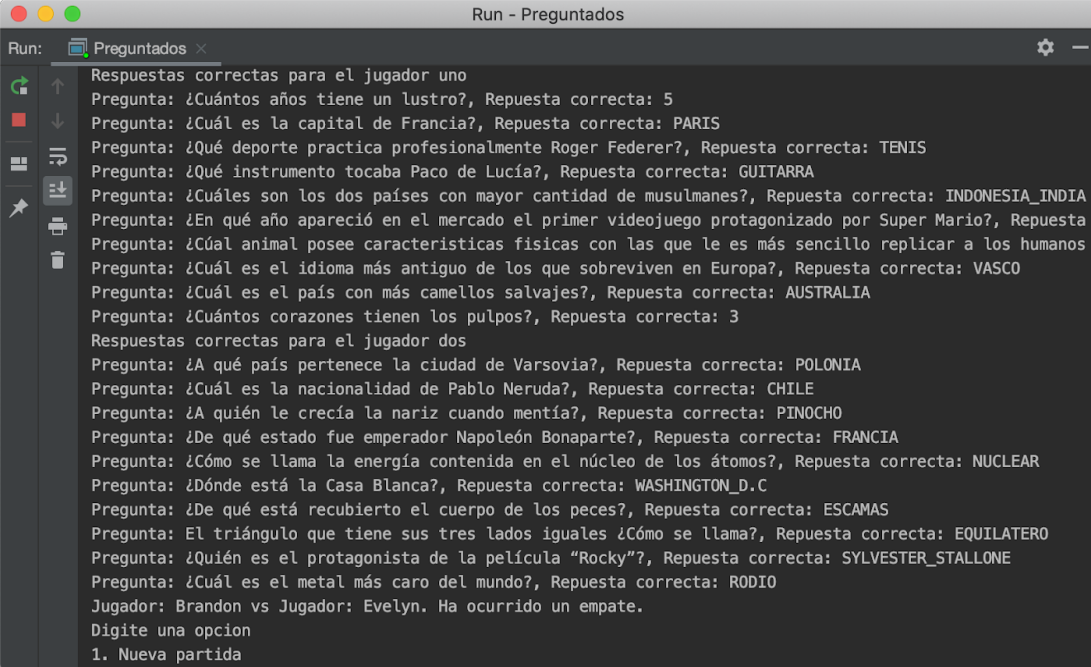
- Quando se selecciona esta opción, el programa solicita la identificación y nombre de cada jugador con el fin de recopilar los datos del juego a los que posteriormente tendrá acceso.



b. Seguidamente se muestra la primer pregunta al jugador uno que debe escribir la respuesta tal y como se muestra en las opciones, dependiendo de si su respuesta es correcta o no, cada vez que responda se muestra si acertó o no. Alternando turnos, continúa el juego hasta completar diez preguntas cada uno. Cuando esto sucede.

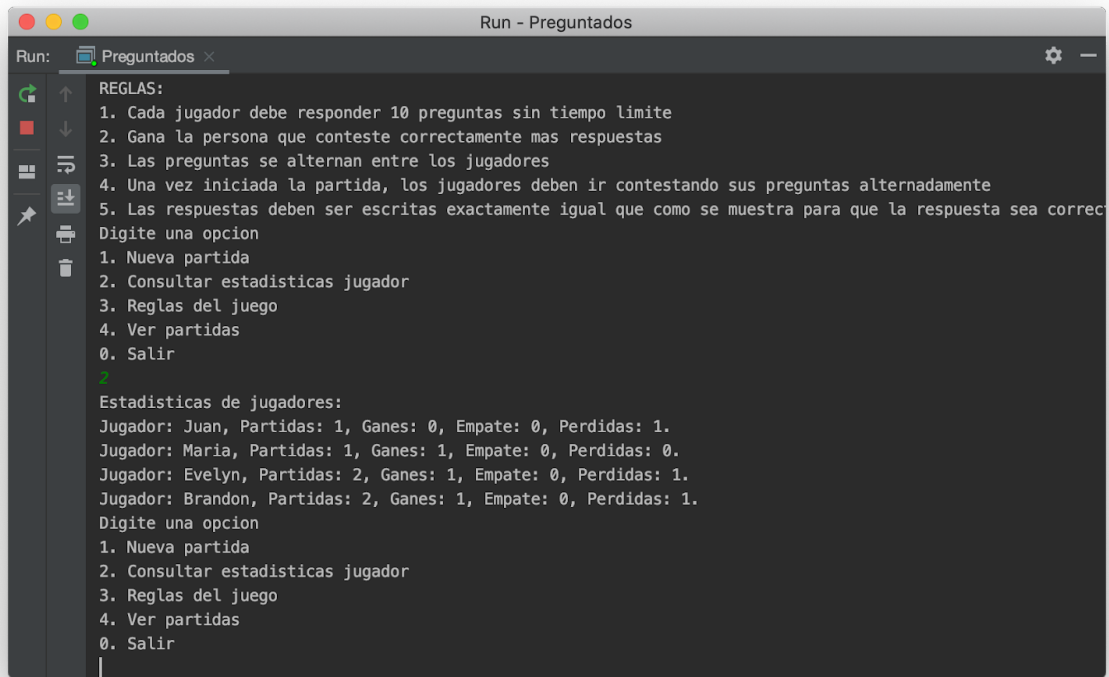
```
Run - Preguntados
Run: Preguntados x
Jugador dos
Pregunta: ¿Cómo se llama la energía contenida en el núcleo de los átomos?
Opciones: POTENCIAL ELECTRICA NUCLEAR QUIMICA
Escriba la opción correcta:NUCLEAR
Correcto
Jugador uno
Pregunta: ¿En qué año apareció en el mercado el primer videojuego protagonizado por Super Mario?
Opciones: 1980 1981 1990 1985
Escriba la opción correcta:1981
Correcto
Jugador dos
Pregunta: ¿Dónde está la Casa Blanca?
Opciones: CALIFORNIA TEXAS NUEVA_YORK WASHINGTON_D.C
Escriba la opción correcta:WASHINGTON_D.C
Correcto
Jugador uno
Pregunta: ¿Cuál animal posee características físicas con las que le es más sencillo replicar a los humanos?
Opciones: LORO LAPA PERICO CUERVO
Escriba la opción correcta:CUERVO
Correcto
Jugador dos
Pregunta: ¿De qué está recubierto el cuerpo de los peces?
Opciones: PELO ESCAMAS PLUMAS PIEL
Escriba la opción correcta:ESCAMAS
Correcto
```

c. El programa muestra un texto con el ganador y finalmente muestra todas las preguntas que respondieron y sus respuestas correctas para que el jugador las aprenda.



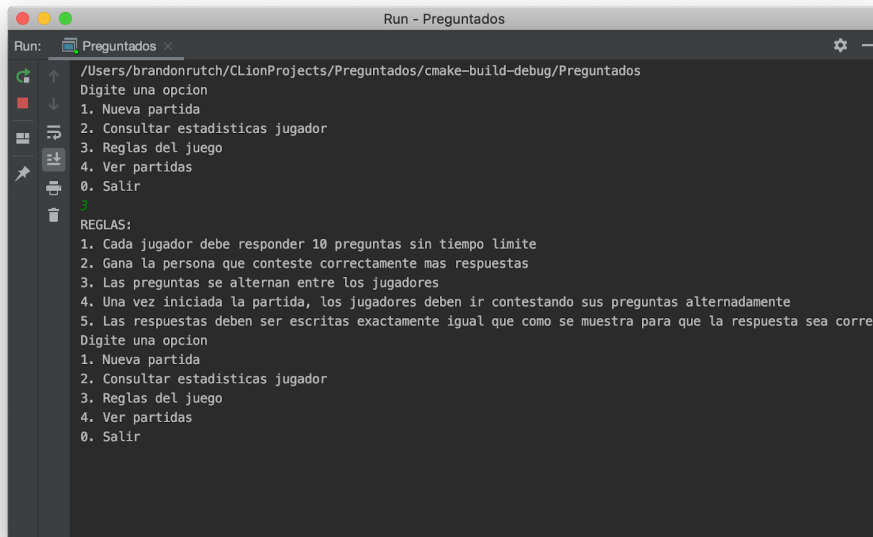
```
Run: Preguntados x
Respuestas correctas para el jugador uno
Pregunta: ¿Cuántos años tiene un lustro?, Repuesta correcta: 5
Pregunta: ¿Cuál es la capital de Francia?, Repuesta correcta: PARIS
Pregunta: ¿Qué deporte practica profesionalmente Roger Federer?, Repuesta correcta: TENIS
Pregunta: ¿Qué instrumento tocaba Paco de Lucía?, Repuesta correcta: GUITARRA
Pregunta: ¿Cuáles son los dos países con mayor cantidad de musulmanes?, Repuesta correcta: INDONESIA_INDIA
Pregunta: ¿En qué año apareció en el mercado el primer videojuego protagonizado por Super Mario?, Repuesta
Pregunta: ¿Cuál animal posee características físicas con las que le es más sencillo replicar a los humanos?
Pregunta: ¿Cuál es el idioma más antiguo de los que sobreviven en Europa?, Repuesta correcta: VASCO
Pregunta: ¿Cuál es el país con más camellos salvajes?, Repuesta correcta: AUSTRALIA
Pregunta: ¿Cuántos corazones tienen los pulpos?, Repuesta correcta: 3
Respuestas correctas para el jugador dos
Pregunta: ¿A qué país pertenece la ciudad de Varsovia?, Repuesta correcta: POLONIA
Pregunta: ¿Cuál es la nacionalidad de Pablo Neruda?, Repuesta correcta: CHILE
Pregunta: ¿A quién le crecía la nariz cuando mentía?, Repuesta correcta: PINOCHO
Pregunta: ¿De qué estado fue emperador Napoleón Bonaparte?, Repuesta correcta: FRANCIA
Pregunta: ¿Cómo se llama la energía contenida en el núcleo de los átomos?, Repuesta correcta: NUCLEAR
Pregunta: ¿Dónde está la Casa Blanca?, Repuesta correcta: WASHINGTON_D.C
Pregunta: ¿De qué está recubierto el cuerpo de los peces?, Repuesta correcta: ESCAMAS
Pregunta: El triángulo que tiene sus tres lados iguales ¿Cómo se llama?, Repuesta correcta: EQUILATERO
Pregunta: ¿Quién es el protagonista de la película "Rocky"?, Repuesta correcta: SYLVESTER_STALLONE
Pregunta: ¿Cuál es el metal más caro del mundo?, Repuesta correcta: RODIO
Jugador: Brandon vs Jugador: Evelyn. Ha ocurrido un empate.
Digite una opcion
1. Nueva partida
```

2. Consultar estadísticas jugador: Cuando se selecciona esta opción, se muestran todas las estadísticas de los jugadores ordenadas del primero en jugar al último



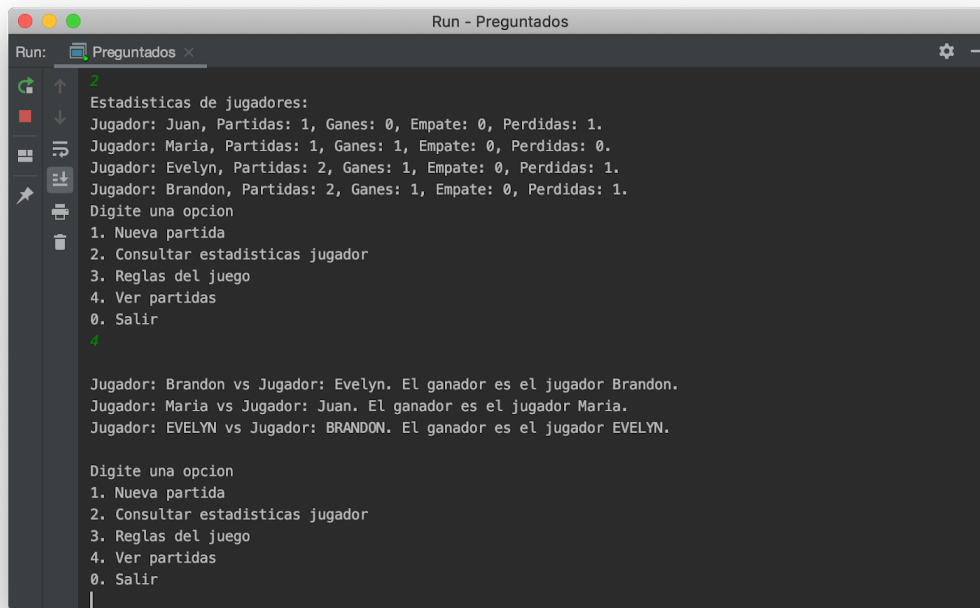
```
Run: Preguntados x
REGLAS:
1. Cada jugador debe responder 10 preguntas sin tiempo limite
2. Gana la persona que conteste correctamente mas respuestas
3. Las preguntas se alternan entre los jugadores
4. Una vez iniciada la partida, los jugadores deben ir contestando sus preguntas alternadamente
5. Las respuestas deben ser escritas exactamente igual que como se muestra para que la respuesta sea correcta
Digite una opcion
1. Nueva partida
2. Consultar estadisticas jugador
3. Reglas del juego
4. Ver partidas
0. Salir
2
Estadisticas de jugadores:
Jugador: Juan, Partidas: 1, Ganes: 0, Empate: 0, Perdidas: 1.
Jugador: Maria, Partidas: 1, Ganes: 1, Empate: 0, Perdidas: 0.
Jugador: Evelyn, Partidas: 2, Ganes: 1, Empate: 0, Perdidas: 1.
Jugador: Brandon, Partidas: 2, Ganes: 1, Empate: 0, Perdidas: 1.
Digite una opcion
1. Nueva partida
2. Consultar estadisticas jugador
3. Reglas del juego
4. Ver partidas
0. Salir
|
```

3. Reglas: Esta opción despliega las reglas del juego



```
Run: Preguntados x
/Users/brandonrutch/ClionProjects/Preguntados/cmake-build-debug/Preguntados
Digite una opcion
1. Nueva partida
2. Consultar estadisticas jugador
3. Reglas del juego
4. Ver partidas
0. Salir
4
REGLAS:
1. Cada jugador debe responder 10 preguntas sin tiempo limite
2. Gana la persona que conteste correctamente mas respuestas
3. Las preguntas se alternan entre los jugadores
4. Una vez iniciada la partida, los jugadores deben ir contestando sus preguntas alternadamente
5. Las respuestas deben ser escritas exactamente igual que como se muestra para que la respuesta sea correcta
Digite una opcion
1. Nueva partida
2. Consultar estadisticas jugador
3. Reglas del juego
4. Ver partidas
0. Salir
```

4. Información partidas: Al seleccionar esto, se despliega el historial de partidas de todos los jugadores que han usado el juego.

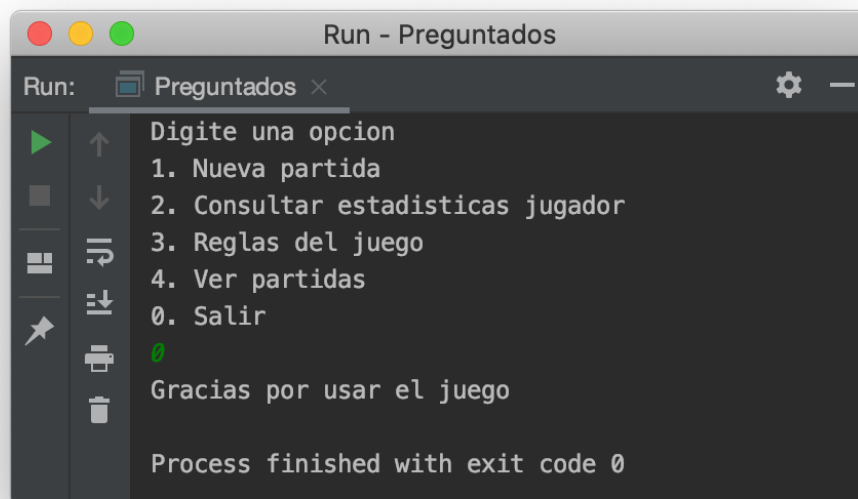


```
Run: Preguntados x
Estadísticas de jugadores:
Jugador: Juan, Partidas: 1, Ganes: 0, Empate: 0, Perdidas: 1.
Jugador: Maria, Partidas: 1, Ganes: 1, Empate: 0, Perdidas: 0.
Jugador: Evelyn, Partidas: 2, Ganes: 1, Empate: 0, Perdidas: 1.
Jugador: Brandon, Partidas: 2, Ganes: 1, Empate: 0, Perdidas: 1.
Digite una opcion
1. Nueva partida
2. Consultar estadísticas jugador
3. Reglas del juego
4. Ver partidas
0. Salir

Jugador: Brandon vs Jugador: Evelyn. El ganador es el jugador Brandon.
Jugador: Maria vs Jugador: Juan. El ganador es el jugador Maria.
Jugador: EVELYN vs Jugador: BRANDON. El ganador es el jugador EVELYN.

Digite una opcion
1. Nueva partida
2. Consultar estadísticas jugador
3. Reglas del juego
4. Ver partidas
0. Salir
|
```

5. Salir: Termina el programa



```
Run: Preguntados x
Digite una opcion
1. Nueva partida
2. Consultar estadísticas jugador
3. Reglas del juego
4. Ver partidas
0. Salir
0
Gracias por usar el juego

Process finished with exit code 0
```