**Mundial de Fútbol Brasil 2014**

32 equipos jugarán el Mundial de Fútbol de Brasil 2014. Inicialmente, estos 32 equipos entran en un sorteo para dividirlos en 8 grupos de 4 equipos. Una vez distribuidos por grupos comienza la primera fase del mundial, en la que en cada grupo se disputa una liguilla a una sola vuelta. Es decir, cada equipo de un grupo juega un partido contra los otros 3 equipos de ese grupo. Cuando un equipo gana un partido se le conceden 3 puntos; si empata, 1 punto. Al finalizar la primera fase, los dos mejores equipos de cada grupo se clasifican para la siguiente fase, la fase de eliminatorias. El objetivo del examen es realizar \_\_un programa que nos permita simular el sorteo y la primera fase\_\_ del Mundial de Fútbol de Brasil 2014.

[](https://github.com/miguelbayon/miguelbayon.github.io/blob/cd218305c70764e1f03110f44a27520944f7c0e8/images/mundial.jpg)

**Consideraciones**

1. Cada equipo cuenta con una valoración de 0 a 100 (ambos incluidos), que el programa calcula de forma aleatoria.
2. Cuando un equipo juega un partido se suma a su valoración un valor entre 0 y 50 (ambos incluidos) que se obtiene de forma aleatoria y que representa la suerte que tendrá ese equipo en ese partido.
3. Por cada 40 puntos de la valoración final el equipo consigue un gol en el partido. Por ejemplo, si un equipo tiene una valoración de 20 y su suerte para el partido vale 12 no marcará ningún gol en ese partido ya que no llega a 40 puntos. En cambio, si la valoración del equipo fuera de 70 y su suerte en ese partido de 12, el equipo marcaría 2 goles.

**Pasos a completar**

1. Crea un repositorio local de git en tu equipo.
2. Asocia dicho repositorio con el repositorio remoto adecuado conforme las instrucciones del anexo II.
3. 20 puntos Codifica el código necesario, conforme a la documentacón de las clases del anexo I, para que se pueda crear un objeto de tipo Mundial desde la interfaz de BlueJ.

Haz un commit en la rama master y realiza un push.

1. 20 puntos Codifica el código necesario, conforme a la documentacón de las clases del anexo I, para que se distribuyan los equipos en grupos (método generaGrupos()) y se pueda ver por pantalla dicha distribución (métodomostrarGrupos()).

Puedes ver [aquí](https://github.com/miguelbayon/miguelbayon.github.io/blob/cd218305c70764e1f03110f44a27520944f7c0e8/images/grupos.png) un ejemlo de cómo debe ser la salida por pantalla.

Haz un commit en la rama master y realiza un push.

1. 20 puntos Codifica el código necesario, conforme a la documentacón de las clases del anexo I, para que se generen los partidos que los equipos deben disputar dentro de cada grupo (métodogeneraPartidosPrimeraFase()) y se pueda ver por pantalla el listado de los mismos por grupo (método mostrarPartidosPrimeraFase()).

Puedes ver [aquí](https://github.com/miguelbayon/miguelbayon.github.io/blob/cd218305c70764e1f03110f44a27520944f7c0e8/images/partidos.png) un ejemlo de cómo debe ser la salida por pantalla.

Haz un commit en la rama master y realiza un push.

1. 40 puntos Codifica el código necesario, conforme a la documentacón de las clases del anexo I, para que se generen los resultados de cada partido de cada grupo y las clasificaciones de cada grupo (métodogeneraResultadosPrimeraFase()) y se pueda ver por pantalla el resultado de los partidos de cada grupo y a clasificacion final de los equipos ordenada por puntos (método mostrarResultadosYClasificacionesPrimeraFase()).

Puedes ver [aquí](https://github.com/miguelbayon/miguelbayon.github.io/blob/cd218305c70764e1f03110f44a27520944f7c0e8/images/resultadosYClasificacion.png) un ejemlo de cómo debe ser la salida por pantalla.

Haz un commit en la rama master y realiza un push.

**Observaciones:**

* Es obligatorio para que se corrija el examen que:
  + El código compile sin errores sobre Java 7.
  + Se utilice git de forma correcta para el control de versiones
* No se permite:
  + El uso de la sentencia break excepto en claúsulas switch.
  + Utilizar más de un return en los métodos.
* Se valora negativamente:
  + La redundancia de código.
  + La incorrección del código.
  + La utilización de código ineficiente.
  + La indentación incorrecta.
  + La utilización de números literales dentro del código en vez de constantes.
  + La incorrección en el diseño.

**Importante**

* No se permite el uso de ninguna forma de comunicación (verbal, escrita, por Internet) entre alumnos o con cualquier otra persona.
* Tampoco se permite ningún mecanismo que permita transmitir el código fuente al resto de alumnos, incluido el simple acceso a repositorios de otros alumnos.

En caso de detectar alguna de estas prácticas, el alumno suspende directamente la convocatoria actual.

**Anexo I: Clases**

A continuación se incluyen enlaces a la documentación de las 4 clases que deben crearse:

1. Clase Mundial: [ver documentación](https://github.com/miguelbayon/miguelbayon.github.io/blob/cd218305c70764e1f03110f44a27520944f7c0e8/paginas/pro013/examen20140606/Mundial.html)
2. Clase Equipo: [ver documentacion](https://github.com/miguelbayon/miguelbayon.github.io/blob/cd218305c70764e1f03110f44a27520944f7c0e8/paginas/pro013/examen20140606/Equipo.html)
3. Clase Grupo: [ver documentacion](https://github.com/miguelbayon/miguelbayon.github.io/blob/cd218305c70764e1f03110f44a27520944f7c0e8/paginas/pro013/examen20140606/Grupo.html)
4. Clase Partido: [ver documentacion](https://github.com/miguelbayon/miguelbayon.github.io/blob/cd218305c70764e1f03110f44a27520944f7c0e8/paginas/pro013/examen20140606/Partido.html)

A continuación se incluye el código completo de la quinta y última clase del programa, la clase NombresEquipos:

{% highlight java %}

/\*\*

* Esta clase contiene los nombres de los 32 equipos clasificados para
* el Mundial de Fúbtol de Brasil 2014
* \*/ public class NombreEquipos { private static String[] nombresEquipos= { "Brasil ", "España ", "Alemania ", "Argentina ", "Colombia ", "Bélgica ", "Uruguay ", "Suiza ", "Holanda ", "Italia ", "Inglaterra ", "Portugal ", "Grecia ", "Bosnia ", "Croacia ", "Rusia ", "Chile ", "Costa de Marfil", "Francia ", "Ecuador ", "Ghana ", "Argelia ", "Nigeria ", "Camerún ", "EEUU ", "México ", "Costa Rica ", "Honduras ", "Japón ", "Irán ", "Corea del Sur ", "Australia " };

/\*

* + Devuelve el nombre de equipo que ocupa la posición index
  + en el array nombresEquipos \*
  + @param index el indice que ocupa el equipo dentro del array
  + @return un nombre de equipo \*/ public static String getNombreEquipo(int index) { return nombresEquipos[index]; } }

{% endhighlight %}

Mundial:

**Class Mundial**

java.lang.Object

**Mundial**

public class **Mundial**

extends java.lang.Object

Esta clase incluye los equipos que disputan el mundial y los distintos grupos en los que quedan agrupados

|  |  |
| --- | --- |
| **Field Summary** | |
| (package private)  Equipo[] | [**equipos**](file:///C:\Users\Clase\AppData\Local\Temp\Mundial.html#equipos)            Almacena todos los equipos que participan en el MUndial |
| (package private)  Grupo[] | [**grupos**](file:///C:\Users\Clase\AppData\Local\Temp\Mundial.html#grupos)            Almacena todos los grupos en los que se dividen los equipos |

|  |  |
| --- | --- |
| **Constructor Summary** | |
| [**Mundial**](file:///C:\Users\Clase\AppData\Local\Temp\Mundial.html#Mundial())()            Construye un objeto Mundial creando todos los equipos que juegan el mundial e introduciendolos en el array equipos |  |

|  |  |
| --- | --- |
| **Method Summary** | |
| void | [**generaGrupos**](file:///C:\Users\Clase\AppData\Local\Temp\Mundial.html#generaGrupos())()            Distribuye aleatoriamente los equipos participantes en tantos grupos como esten definidos en la constante NUMERO\_DE\_GRUPOS. |
| void | [**generaPartidosPrimeraFase**](file:///C:\Users\Clase\AppData\Local\Temp\Mundial.html#generaPartidosPrimeraFase())()            Genera todos los partidos que deben jugarse dentro de un grupo pero sin jugarlos, es decir, sin calcular ning�n resultado |
| void | [**generaResultadosPrimeraFase**](file:///C:\Users\Clase\AppData\Local\Temp\Mundial.html#generaResultadosPrimeraFase())()            Genera los resultados de todos los partidos de todos los grupos |
| void | [**mostrarGrupos**](file:///C:\Users\Clase\AppData\Local\Temp\Mundial.html#mostrarGrupos())()            Muestra por pantalla los datos de todos los grupos (nombre del grupo y equipos pertencientes a dicho grupo) |
| void | [**mostrarPartidosPrimeraFase**](file:///C:\Users\Clase\AppData\Local\Temp\Mundial.html#mostrarPartidosPrimeraFase())()            Muestra por pantalla los partidos que deben jugarse en todos los grupos |
| void | [**mostrarResultadosYClasificacionPrimeraFase**](file:///C:\Users\Clase\AppData\Local\Temp\Mundial.html#mostrarResultadosYClasificacionPrimeraFase())()            Muestra por pantalla los partidos de todos los grupos con sus resultados y las clasificaciones ordenadas de cada grupo |

|  |
| --- |
| **Methods inherited from class** |
| clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait |

|  |
| --- |
| **Field Detail** |

**equipos**

Equipo[] **equipos**

Almacena todos los equipos que participan en el MUndial

**grupos**

Grupo[] **grupos**

Almacena todos los grupos en los que se dividen los equipos

|  |
| --- |
| **Constructor Detail** |

**Mundial**

public **Mundial**()

Construye un objeto Mundial creando todos los equipos que juegan el mundial e introduciendolos en el array equipos

|  |
| --- |
| **Method Detail** |

**generaGrupos**

public void **generaGrupos**()

Distribuye aleatoriamente los equipos participantes en tantos grupos como esten definidos en la constante NUMERO\_DE\_GRUPOS. Todos los grupos quedan almacenados en el array grupos.

**generaPartidosPrimeraFase**

public void **generaPartidosPrimeraFase**()

Genera todos los partidos que deben jugarse dentro de un grupo pero sin jugarlos, es decir, sin calcular ning�n resultado

**generaResultadosPrimeraFase**

public void **generaResultadosPrimeraFase**()

Genera los resultados de todos los partidos de todos los grupos

**mostrarGrupos**

public void **mostrarGrupos**()

Muestra por pantalla los datos de todos los grupos (nombre del grupo y equipos pertencientes a dicho grupo)

**mostrarPartidosPrimeraFase**

public void **mostrarPartidosPrimeraFase**()

Muestra por pantalla los partidos que deben jugarse en todos los grupos

**mostrarResultadosYClasificacionPrimeraFase**

public void **mostrarResultadosYClasificacionPrimeraFase**()

Muestra por pantalla los partidos de todos los grupos con sus resultados y las clasificaciones ordenadas de cada grupo

**Class Equipo**

java.lang.Object

**Equipo**

public class **Equipo**

extends java.lang.Object

Esta clase representa un equipo de futbol

|  |  |
| --- | --- |
| **Field Summary** | |
| (package private)  int | [**golesAFavor**](file:///C:\Users\Clase\AppData\Local\Temp\Equipo.html#golesAFavor)            El total de goles que el equipo ha marcado hasta el momento |
| (package private)  int | [**golesEnContra**](file:///C:\Users\Clase\AppData\Local\Temp\Equipo.html#golesEnContra)            El total de goles que el equipo ha recibido hasta el momento |
| (package private)  java.lang.String | [**nombre**](file:///C:\Users\Clase\AppData\Local\Temp\Equipo.html#nombre)            EL nombre del equipo (por ejemplo, "Espa�a", "Brasil", etc.) |
| (package private)  int | [**partidosEmpatados**](file:///C:\Users\Clase\AppData\Local\Temp\Equipo.html#partidosEmpatados)            El numero de partidos que el equipo ha empatado hasta el momento |
| (package private)  int | [**partidosGanados**](file:///C:\Users\Clase\AppData\Local\Temp\Equipo.html#partidosGanados)            El numero de partidos que el equipo ha ganado hasta el momento |
| (package private)  int | [**partidosJugados**](file:///C:\Users\Clase\AppData\Local\Temp\Equipo.html#partidosJugados)            El numero de partidos que el equipo ha jugado hasta el momento |
| (package private)  int | [**partidosPerdidos**](file:///C:\Users\Clase\AppData\Local\Temp\Equipo.html#partidosPerdidos)            El numero de partidos que el equipo ha perdido hasta el momento |
| (package private)  int | [**puntos**](file:///C:\Users\Clase\AppData\Local\Temp\Equipo.html#puntos)            El numero de puntos acumulados por ese equipo; se obtienen 3 por cada victoria y 1 por cada empate |
| (package private)  int | [**valoracion**](file:///C:\Users\Clase\AppData\Local\Temp\Equipo.html#valoracion)            La valoracion es un valor de 0 a 100 ambos incluidos que representa la calidad de ese equipo y que nos sirve a la hora de simular los partidos |

|  |  |
| --- | --- |
| **Constructor Summary** | |
| [**Equipo**](file:///C:\Users\Clase\AppData\Local\Temp\Equipo.html#Equipo(java.lang.String))(java.lang.String nombre)            Constructor de la clase Equipo; crea un equipo partir de su nombre con todos sus atributos a 0 excepto la valoracion, cuyo valor se obtiene de forma aletaoria entre 0 y 100 (ambos inlcuidos) |  |

|  |  |
| --- | --- |
| **Method Summary** | |
| void | [**display**](file:///C:\Users\Clase\AppData\Local\Temp\Equipo.html#display())()            Muestra por pantalla en una �nica l�nea todos los datos de un equipo excepto la valoraci�n; una salida de ejemplo ser�a "Espa�a 3 2 1 0 10 5 +5 7" donde se muestran por orden el numero de partidos jugados, ganados, empatados, perdidos, goles a favor, goles en contra, diferencia de goles y puntos |
| java.lang.String | [**getNombre**](file:///C:\Users\Clase\AppData\Local\Temp\Equipo.html#getNombre())()            Devuelve el nombre del equipo |
| int | [**getPuntos**](file:///C:\Users\Clase\AppData\Local\Temp\Equipo.html#getPuntos())()            Devuelve los puntos que tiene un equipo actualmente |
| int | [**getValoracion**](file:///C:\Users\Clase\AppData\Local\Temp\Equipo.html#getValoracion())()            Devuelve la valoracion del equipo |
| void | [**incrementarDerrotas**](file:///C:\Users\Clase\AppData\Local\Temp\Equipo.html#incrementarDerrotas())()            Incrementa las derrotas del equipo actualizando convenientemente el numero de partidos jugados y los puntos acumulados |
| void | [**incrementarEmpates**](file:///C:\Users\Clase\AppData\Local\Temp\Equipo.html#incrementarEmpates())()            Incrementa los empates del equipo actualizando convenientemente el numero de partidos jugados y los puntos acumulados |
| void | [**incrementarGolesAFavor**](file:///C:\Users\Clase\AppData\Local\Temp\Equipo.html#incrementarGolesAFavor(int))(int goles)            Incrementa los goles a favor en la cantidad indicada |
| void | [**incrementarGolesEnContra**](file:///C:\Users\Clase\AppData\Local\Temp\Equipo.html#incrementarGolesEnContra(int))(int goles)            Incrementa los goles en contra en la cantidad indicada |
| void | [**incrementarVictorias**](file:///C:\Users\Clase\AppData\Local\Temp\Equipo.html#incrementarVictorias())()            Imcrementa las victorias del equipo actualizando convenientemente el numero de partidos jugados y los puntos acumulados |

|  |
| --- |
| **Methods inherited from class** |
| clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait |

|  |
| --- |
| **Field Detail** |

**golesAFavor**

int **golesAFavor**

El total de goles que el equipo ha marcado hasta el momento

**golesEnContra**

int **golesEnContra**

El total de goles que el equipo ha recibido hasta el momento

**nombre**

java.lang.String **nombre**

EL nombre del equipo (por ejemplo, "Espa�a", "Brasil", etc.)

**partidosEmpatados**

int **partidosEmpatados**

El numero de partidos que el equipo ha empatado hasta el momento

**partidosGanados**

int **partidosGanados**

El numero de partidos que el equipo ha ganado hasta el momento

**partidosJugados**

int **partidosJugados**

El numero de partidos que el equipo ha jugado hasta el momento

**partidosPerdidos**

int **partidosPerdidos**

El numero de partidos que el equipo ha perdido hasta el momento

**puntos**

int **puntos**

El numero de puntos acumulados por ese equipo; se obtienen 3 por cada victoria y 1 por cada empate

**valoracion**

int **valoracion**

La valoracion es un valor de 0 a 100 ambos incluidos que representa la calidad de ese equipo y que nos sirve a la hora de simular los partidos

|  |
| --- |
| **Constructor Detail** |

**Equipo**

public **Equipo**(java.lang.String nombre)

Constructor de la clase Equipo; crea un equipo partir de su nombre con todos sus atributos a 0 excepto la valoracion, cuyo valor se obtiene de forma aletaoria entre 0 y 100 (ambos inlcuidos)

|  |
| --- |
| **Method Detail** |

**display**

public void **display**()

Muestra por pantalla en una �nica l�nea todos los datos de un equipo excepto la valoraci�n; una salida de ejemplo ser�a "Espa�a 3 2 1 0 10 5 +5 7" donde se muestran por orden el numero de partidos jugados, ganados, empatados, perdidos, goles a favor, goles en contra, diferencia de goles y puntos

**getNombre**

public java.lang.String **getNombre**()

Devuelve el nombre del equipo

**Returns:**

el nombre del equipo

**getPuntos**

public int **getPuntos**()

Devuelve los puntos que tiene un equipo actualmente

**Returns:**

el numero de puntos actuales del equipo

**getValoracion**

public int **getValoracion**()

Devuelve la valoracion del equipo

**Returns:**

la valoracion del equipo

**incrementarDerrotas**

public void **incrementarDerrotas**()

Incrementa las derrotas del equipo actualizando convenientemente el numero de partidos jugados y los puntos acumulados

**incrementarEmpates**

public void **incrementarEmpates**()

Incrementa los empates del equipo actualizando convenientemente el numero de partidos jugados y los puntos acumulados

**incrementarGolesAFavor**

public void **incrementarGolesAFavor**(int goles)

Incrementa los goles a favor en la cantidad indicada

**Parameters:**

goles - el numero de goles que ha marcado el equipo en un partido

**incrementarGolesEnContra**

public void **incrementarGolesEnContra**(int goles)

Incrementa los goles en contra en la cantidad indicada

**Parameters:**

goles - el numero de goles que ha recibido el equipo en un partido

**incrementarVictorias**

public void **incrementarVictorias**()

Imcrementa las victorias del equipo actualizando convenientemente el numero de partidos jugados y los puntos acumulados

**Class Grupo**

java.lang.Object

**Grupo**

public class **Grupo**

extends java.lang.Object

Esta clase representa un grupo de los existentes en la primera fase del Mundial de F�tbol 2014 de Brasil

|  |  |
| --- | --- |
| **Field Summary** | |
| (package private)  java.util.ArrayList<Equipo> | [**equipos**](file:///C:\Users\Clase\Desktop\Programacion\EXAMENES\Grupo.html#equipos)            Almacena todos los equipos que estan incluidos en el grupo |
| (package private)  java.lang.String | [**nombreGrupo**](file:///C:\Users\Clase\Desktop\Programacion\EXAMENES\Grupo.html#nombreGrupo)            Almacena el nombre del grupo ("A", "B", etc.) |
| (package private)  java.util.ArrayList<Partido> | [**partidos**](file:///C:\Users\Clase\Desktop\Programacion\EXAMENES\Grupo.html#partidos)            Almacena todos los partidos a disputar dentro de ese grupo |

|  |  |
| --- | --- |
| **Constructor Summary** | |
| [**Grupo**](file:///C:\Users\Clase\Desktop\Programacion\EXAMENES\Grupo.html#Grupo())()            Construye un objeto Grupo con la siguiente letra no asignada; el grupo recien construido no tiene ni partidos ni equipos asignados |  |

|  |  |
| --- | --- |
| **Method Summary** | |
| void | [**addEquipo**](file:///C:\Users\Clase\Desktop\Programacion\EXAMENES\Grupo.html#addEquipo(Equipo))(Equipo equipo)            A�ade un equipo al grupo |
| void | [**displayEquipos**](file:///C:\Users\Clase\Desktop\Programacion\EXAMENES\Grupo.html#displayEquipos())()            Muestra por pantalla el nombre del grupo y el nombre de todos los equipos que pertenecen a ese grupo |
| void | [**displayPartidos**](file:///C:\Users\Clase\Desktop\Programacion\EXAMENES\Grupo.html#displayPartidos())()            Muestra por pantalla el nombre del grupo y los partidos que deben disputarse en ese grupo |
| void | [**displayResultadosYClasificacion**](file:///C:\Users\Clase\Desktop\Programacion\EXAMENES\Grupo.html#displayResultadosYClasificacion())()            Muestra por pantalla el nombre del grupo y los partidos con sus resultados; adem�s, muestra la clasificaci�n del grupo ordenada por puntos |
| void | [**generaPartidos**](file:///C:\Users\Clase\Desktop\Programacion\EXAMENES\Grupo.html#generaPartidos())()            Genera los partidos a disputar dentro del grupo pero no los juega |
| (package private)  void | [**ordenaEquiposEnFuncionDeSusPuntos**](file:///C:\Users\Clase\Desktop\Programacion\EXAMENES\Grupo.html#ordenaEquiposEnFuncionDeSusPuntos())()            Ordena los equipos del grupo almacenados en equipos en funci�n del n�mero de puntos de cada equipo; este m�todo debe ser PRIVADO. |
| void | [**simularPartidos**](file:///C:\Users\Clase\Desktop\Programacion\EXAMENES\Grupo.html#simularPartidos())()            Simula el juego de todos los partidos del grupo y al terminar ordena los equipos en funci�n de sus puntos |

|  |
| --- |
| **Methods inherited from class** |
| clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait |

|  |
| --- |
| **Field Detail** |

**equipos**

java.util.ArrayList<Equipo> **equipos**

Almacena todos los equipos que estan incluidos en el grupo

**nombreGrupo**

java.lang.String **nombreGrupo**

Almacena el nombre del grupo ("A", "B", etc.)

**partidos**

java.util.ArrayList<Partido> **partidos**

Almacena todos los partidos a disputar dentro de ese grupo

|  |
| --- |
| **Constructor Detail** |

**Grupo**

public **Grupo**()

Construye un objeto Grupo con la siguiente letra no asignada; el grupo recien construido no tiene ni partidos ni equipos asignados

|  |
| --- |
| **Method Detail** |

**addEquipo**

public void **addEquipo**(Equipo equipo)

A�ade un equipo al grupo

**Parameters:**

equipo - el equipo a a�adir

**displayEquipos**

public void **displayEquipos**()

Muestra por pantalla el nombre del grupo y el nombre de todos los equipos que pertenecen a ese grupo

**displayPartidos**

public void **displayPartidos**()

Muestra por pantalla el nombre del grupo y los partidos que deben disputarse en ese grupo

**displayResultadosYClasificacion**

public void **displayResultadosYClasificacion**()

Muestra por pantalla el nombre del grupo y los partidos con sus resultados; adem�s, muestra la clasificaci�n del grupo ordenada por puntos

**generaPartidos**

public void **generaPartidos**()

Genera los partidos a disputar dentro del grupo pero no los juega

**ordenaEquiposEnFuncionDeSusPuntos**

void **ordenaEquiposEnFuncionDeSusPuntos**()

Ordena los equipos del grupo almacenados en equipos en funci�n del n�mero de puntos de cada equipo; este m�todo debe ser PRIVADO.

**simularPartidos**

public void **simularPartidos**()

Simula el juego de todos los partidos del grupo y al terminar ordena los equipos en funci�n de sus puntos

**Class Partido**

java.lang.Object

**Partido**

public class **Partido**

extends java.lang.Object

Esta clase representa un partido de futbol. En un partido hay dos equipos: el primero recibe el nombre de equipo local y el segundo recibe el nombre de equipo visitante

|  |  |
| --- | --- |
| **Field Summary** | |
| (package private)  int | [**golesLocal**](file:///C:\Users\Clase\Desktop\Programacion\EXAMENES\Partido.html#golesLocal)            El numero de goles que ha marcado el equipo local |
| (package private)  int | [**golesVisitante**](file:///C:\Users\Clase\Desktop\Programacion\EXAMENES\Partido.html#golesVisitante)            El numero de goles que ha marcado el equipo visitante |
| (package private)  boolean | [**jugado**](file:///C:\Users\Clase\Desktop\Programacion\EXAMENES\Partido.html#jugado)            Indica si el partido se ha jugado ya o no |
| (package private)  Equipo | [**local**](file:///C:\Users\Clase\Desktop\Programacion\EXAMENES\Partido.html#local)            El equipo que actua como local |
| (package private)  Equipo | [**visitante**](file:///C:\Users\Clase\Desktop\Programacion\EXAMENES\Partido.html#visitante)            El equipo que actua como visitante |

|  |  |
| --- | --- |
| **Constructor Summary** | |
| [**Partido**](file:///C:\Users\Clase\Desktop\Programacion\EXAMENES\Partido.html#Partido(Equipo, Equipo))(Equipo local, Equipo visitante)            Construye un objeto Partido a partir de los equipos indicados que representa un partido que a�n no ha sido jugado |  |

|  |  |
| --- | --- |
| **Method Summary** | |
| void | [**display**](file:///C:\Users\Clase\Desktop\Programacion\EXAMENES\Partido.html#display())()            Muestra por pantalla en una �nica l�nea todos los datos de un partido. |
| Equipo | [**getEquipoVencedor**](file:///C:\Users\Clase\Desktop\Programacion\EXAMENES\Partido.html#getEquipoVencedor())()            Devuelve el equipo ganador |
| boolean | [**huboEmpate**](file:///C:\Users\Clase\Desktop\Programacion\EXAMENES\Partido.html#huboEmpate())()            Devuelve true en caso de que una vez jugado el partido haya terminado en empate o false en otro caso |
| void | [**simularPartido**](file:///C:\Users\Clase\Desktop\Programacion\EXAMENES\Partido.html#simularPartido())()            Simula la disputa de un partido de futbol y actualiza los datos de los equipos en funci�n del resultado final del partido; en funci�n de la valoraci�n y de la suerte que tiene cada equipo se obtiene el n�mero de goles que marca. |

|  |
| --- |
| **Methods inherited from class** |
| clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait |

|  |
| --- |
| **Field Detail** |

**golesLocal**

int **golesLocal**

El numero de goles que ha marcado el equipo local

**golesVisitante**

int **golesVisitante**

El numero de goles que ha marcado el equipo visitante

**jugado**

boolean **jugado**

Indica si el partido se ha jugado ya o no

**local**

Equipo **local**

El equipo que actua como local

**visitante**

Equipo **visitante**

El equipo que actua como visitante

|  |
| --- |
| **Constructor Detail** |

**Partido**

public **Partido**(Equipo local,

Equipo visitante)

Construye un objeto Partido a partir de los equipos indicados que representa un partido que a�n no ha sido jugado

|  |
| --- |
| **Method Detail** |

**display**

public void **display**()

Muestra por pantalla en una �nica l�nea todos los datos de un partido. Si el partido se ha jugado ya entonces tambi�n muestra el resultado

**getEquipoVencedor**

public Equipo **getEquipoVencedor**()

Devuelve el equipo ganador

**Returns:**

el equipo vencedor del partido o null en caso de empate

**huboEmpate**

public boolean **huboEmpate**()

Devuelve true en caso de que una vez jugado el partido haya terminado en empate o false en otro caso

**Returns:**

true o false en funcion de si el partido ha terminado en empate

**simularPartido**

public void **simularPartido**()

Simula la disputa de un partido de futbol y actualiza los datos de los equipos en funci�n del resultado final del partido; en funci�n de la valoraci�n y de la suerte que tiene cada equipo se obtiene el n�mero de goles que marca. Por cada 40 puntos de la suma de los valores indicados se obtiene un gol.