

Clases Compartidas

Cluster número 11

Contenido

Especificación	3
1. Clase Partida	
• Atributos	3
• Operaciones	4
2. Clase Usuario	
• Atributos	5
• Operaciones	5
3. Clase Posición	
• Atributos	6
• Operaciones	6
4. Clase Estadísticas	
• Atributos	8
• Operaciones	9
5. Clase Fecha	
• Atributos	11
• Operaciones	13
6. Clase Hora	
• Atributos	15
• Operaciones	15
7. Clase Tablero	
• Atributos	17
• Operaciones	17
8. Clase Repositorio	
• Atributos	19
• Operaciones	19
9. Clase Casilla	
• Atributos	20
• Operaciones	20
Diagrama de Clases Compartidas	22
Tabla de reparto de responsabilidades	23
Autores	24

Especificación

1. Clase Partida

Nombre de la clase:	Partida (public class Partida)
Descripción:	Gestiona usuarios y tableros ya generados, las partidas quedan unívocamente identificadas por su identificador. Una partida solo será finalizada, si se ha explicitado su fin.
Cardinalidad:	Múltiple
Package:	Dominio

Atributos:

private String idPartida;

- Valor inicial: null
- Descripción: Identifica unívocamente la partida dentro del sistema con caracteres alfanúmeros.
- Estático: NO

private String idUsuario;

- Valor inicial: null
- Descripción: Identifica unívocamente al usuario dentro del sistema con caracteres alfanúmeros.
- Estático: NO

private String idTablero;

- Valor inicial: null
- Descripción: Identifica unívocamente un tablero dentro del sistema con caracteres alfanúmeros.
- Estático: NO

private int tiempo;

- Valor inicial: 0
- Descripción: Indica el tiempo transcurrido desde el inicio de la partida.
- Estático: NO

private Boolean finalizada;

- Valor inicial: false
- Descripción: Indica si la partida ha sido finalizada en alguna ocasión.
- Estático: NO

Operaciones:

/* PRE: 'idPartida' tiene que ser un identificador de Partida nuevo, no usado anteriormente. Tanto 'idUserario' como 'idTablero' tienen que ser ids válidos, es decir, todos hacen referencia a objetos existentes. Además los ids sólo pueden contener caracteres alfanuméricos. */

public Partida(String idPartida, String idUsuario, String idTablero)

/* POST: Se ha creado un objeto de la clase Partida con idPartida = 'idPartida', idUsuario = 'idUserario' y idTablero = 'idTablero'. */

/* PRE: Ha de existir 'original' y 'idPartidaNueva' tiene que ser un identificador de partida nuevo, no usado anteriormente. 'idPartidaNueva' solo contiene caracteres alfanumericos. */

public Partida(Partida original, String idPartidaNueva)

/* POST: Se ha creado un nuevo objeto de la clase Partida teniendo los mismo atributos que 'original', excepto el identificador de la partida, que es el indicado por 'idPartidaNueva'. Se ha de tener en cuenta que el tiempo de la nueva partida creada está parado, aunque conservando el tiempo de 'original'. */

/* PRE: - */

public String getIdPartida()

/* POST: Se ha retornado el identificador de partida. */

/* PRE: - */

public String getIdUsuario()

/* POST: Se ha retornado el identificador del usuario de la partida. */

/* PRE: - */

public String getIdTablero()

/* POST: Se ha retornado el identificador del tablero en el que se juega la partida. */

/* PRE: - */

public int getTiempo()

/* POST: Retorna el atributo tiempo. */

/* PRE: - */

public void setTiempo(int tiempo)

/* POST: El atributo tiempo pasa a valer 'tiempo'. */

/* PRE: - */

public Boolean esTerminada()

/* POST: Retorna el atributo finalizada. */

/* PRE: - */

public void setTerminada()

/* POST: Se ha marcado la partida como terminada. Si ya había sido marcada como terminada no tiene ningún efecto. */

2. Clase Usuario

Nombre de la clase: Usuario (public class Usuario)

Descripción: La clase "Usuario" representa cada una de las cuentas de usuario de la aplicación, de manera que cada usuario deberá tener una cuenta en el sistema para poder utilizar la aplicación y/o acceder a zonas exclusivas de cada usuario.

Cardinalidad: Múltiple

Package: Dominio

Atributos:

private String nickname;

- Valor inicial: -
- Descripción: "nickname" es el identificador de cada instancia de la clase "Usuario".
- Estático: NO

private String password;

- Valor inicial: -
- Descripción: "password" es el código de acceso de cada usuario.
- Estático: NO

Operaciones:

```
/* PRE: No existe ninguna instancia de Usuario con nickname = 'nickname'.  
'password' y 'nickname' unicamente contienen caracteres alfanumericos. */  
public Usuario(String nickname, String password)  
/* POST: Se crea una nueva instancia de Usuario en el sistema con  
nickname = 'nickname' y password = 'password'. */
```

```
/* PRE: 'nuevoPassword' unicamente contiene caracteres alfanumericos. */  
public void modificarPassword(String nuevoPassword)  
/* POST: El atributo password pasa a valer 'nuevoPassword'. */
```

```
/* PRE: - */  
public String obtenerUsuario()  
/* POST: Retorna el atributo nickname. */
```

```
/* PRE: - */  
public String obtenerPassword()  
/* POST: Retorna el atributo password. */
```

3. Clase Posición

Nombre de la clase:	Posición (public class Posicion)
Descripción:	Implementa un elemento de ránking, que gestiona los récords de cada usuario.
Cardinalidad:	Múltiple
Package:	Dominio

Atributos:

private String nJugador;

- Valor inicial: null
- Descripción: Contiene el nombre del jugador.
- Estático: NO

private int puntuacion;

- Valor inicial: 0
- Descripción: Contiene la máxima puntuación que ha realizado el jugador en una partida.
- Estático: NO

private Fecha fecha;

- Valor inicial: null
- Descripción: Contiene la fecha en la cual se consiguió la máxima puntuación.
- Estático: NO

Operaciones:

```
/* PRE: puntuacion>=0, 'nombreUsuario' solo utiliza caracteres alfanumericos */  
public Posicion (int puntuacion, String nombreUsuario, Fecha fecha)  
/* POST: crea una instancia de Posicion inicializada a los datos de su record */
```

```
/* PRE: - */  
public String getNjugador()  
/* POST: Retorna el atributo nJugador. */
```

```
/* PRE: - */  
public Integer getPuntuacion()  
/* POST: Retorna el atributo puntuacion. */
```

```
/* PRE: - */  
public Fecha getFecha()  
/* POST: Retorna el atributo fecha. */
```

4. Clase Estadísticas

Nombre de la clase:	Estadísticas (public class Estadisticas)
Descripción:	Esta clase contiene las estadísticas de la aplicación.
Cardinalidad:	Única
Package:	Dominio

Atributos:

private int totalPartidas;

- Valor inicial: 0
- Descripción: Indica el número de partidas jugadas.
- Estático: NO

private int partidasAbandonadas;

- Valor inicial: 0
- Descripción: Indica el número de partidas abandonadas.
- Estático: NO

private int partidasResueltas;

- Valor inicial: 0
- Descripción: Indica el número de partidas finalizadas con éxito.
- Estático: NO

private int partidasFacil;

- Valor inicial: 0
- Descripción: Indica el número de partidas jugadas en nivel fácil.
- Estático: NO

private int partidasMedio;

- Valor inicial: 0
- Descripción: Indica el número de partidas jugadas en nivel medio.
- Estático: NO

private int partidasDificil;

- Valor inicial: 0
- Descripción: Indica el número de partidas jugadas en nivel difícil.
- Estático: NO

private int totalUsuarios;

- Valor inicial: 0
- Descripción: Indica el número de usuarios del sistema.
- Estático: NO

Operaciones:

```
/* PRE: - */
public Estadísticas()
/* POST: Crea una instancia con todos sus atributos inicializados a 0. */

/* PRE: - */
public int obtenerTotalPartidas()
/* POST: Obtiene el valor del atributo totalPartidas. */

/* PRE: - */
public void incrementarTotalPartidas(int inc)
/* POST: Incrementa en inc el valor del atributo totalPartidas. */

/* PRE: - */
public int obtenerPartidasAbandonadas()
/* POST: Obtiene el valor del atributo partidasAbandonadas. */

/* PRE: - */
public void incrementarPartidasAbandonadas(int inc)
/* POST: Incrementa en inc el valor del atributo partidasAbandonadas. */

/* PRE: - */
public int obtenerPartidasResueltas()
/* POST: Obtiene el valor del atributo partidasResueltas. */

/* PRE: - */
public void incrementarPartidasResueltas(int inc)
/* POST: Incrementa en inc el valor del atributo partidasResueltas. */

/* PRE: - */
public int obtenerPartidasFacil()
/* POST: Obtiene el valor del atributo partidasFacil. */

/* PRE: - */
public void incrementarPartidasFacil(int inc)
/* POST: Incrementa en inc el valor del atributo partidasFacil. */

/* PRE: - */
public int obtenerPartidasMedio()
/* POST: Obtiene el valor del atributo partidasMedio. */

/* PRE: - */
public void incrementarPartidasMedio(int inc)
/* POST: Incrementa en inc el valor del atributo partidasMedio. */

/*PRE: - */
public int obtenerPartidasDifícil()
/*POST: Obtiene el valor del atributo partidasDifícil. */

/* PRE: - */
public void incrementarPartidasDifícil(int inc)
/* POST: Incrementa en inc el valor del atributo partidasDifícil */

/* PRE: - */
public int obtenerTotalUsuarios()
/* POST: Obtiene el valor del atributo totalUsuarios. */
```



```
/* PRE: - */  
public void incrementarTotalUsuarios(int inc)  
/* POST: Incrementa en inc el valor del atributo totalUsuarios. */  
  
/* PRE: - */  
public List<Integer> obtenerEstadisticas();  
/* POST: Obtiene una lista con todos los atributos de la clase estadisticas. */
```

5. Clase Fecha

Nombre de la clase: Fecha (public class Fecha)

Descripción: La clase "Fecha" implementa una representación de una fecha mediante el día de la semana, el día del mes, el mes y el año. Nos permite obtener la fecha actual o seleccionar una deseada.

Cardinalidad: Múltiple

Package: Tiempo

Atributos:

public final static int ENERO;

- Valor inicial: 1
- Estático: SI

public final static int FEBRERO;

- Valor inicial: 2
- Estático: SI

public final static int MARZO;

- Valor inicial: 3
- Estático: SI

public final static int ABRIL;

- Valor inicial: 4
- Estático: SI

public final static int MAYO;

- Valor inicial: 5
- Estático: SI

public final static int JUNIO;

- Valor inicial: 6
- Estático: SI

public final static int JULIO;

- Valor inicial: 7
- Estático: SI

public final static int AGOSTO;

- Valor inicial: 8
- Estático: SI

public final static int SEPTIEMBRE;

- Valor inicial: 9
- Estático: SI

public final static int OCTUBRE;

- Valor inicial: 10
- Estático: SI

public final static int NOVIEMBRE;

- Valor inicial: 11
- Estático: SI

public final static int DICIEMBRE;

- Valor inicial: 12
- Estático: SI

private final static String DIAS_DE_LA_SEMANA[]

- Valor inicial: {null, "Domingo", "Lunes", "Martes", "Miercoles", "Jueves", "Viernes", "Sabado" }
- Descripción: "DIAS_DE_LA_SEMANA" es una tabla que contiene los nombres de todos los días de la semana, del 1 al 7 (el día 0 no existe).
- Estático: SI

private final static String MESES[]

- Valor inicial: {null, "Enero", "Febrero", "Marzo", "Abril", "Mayo", "Junio", "Julio", "Agosto", "Septiembre", "Octubre", "Noviembre", "Diciembre" }
- Descripción: "MESES" es una tabla que contiene los nombres de todos los meses del año, del 1 al 12 (el mes 0 no existe).
- Estático: SI

private int diaDelMes;

- Valor inicial: -
- Descripción: "diaDelMes" representa el dia del mes de la fecha en cuestión.
- Estático: NO

private int diaDeLaSemana;

- Valor inicial: -
- Descripción: "diaDeLaSemana" representa el dia de la semana de la fecha en cuestión.
- Estático: NO

private int mes;

- Valor inicial: -
- Descripción: "mes" representa el mes de la fecha en cuestión.
- Estático: NO

private int anio;

- Valor inicial: -
- Descripción: "anio" representa el año de la fecha en cuestión.
- Estático: NO

Operaciones:

```
/* PRE: - */
public Fecha()
/* POST: Se crea una instancia de Fecha con todos sus atributos
inicializados a 0. */

/* PRE: - */
public void hoy()
/* POST: Todos los atributos de Fecha quedan modificados de tal manera
que su valor se corresponde con el dia de hoy. */

/* PRE: 'dia', 'mes' y 'anio' representan una fecha valida segun el calendario
Gregoriano. */
public void ponerFecha(int dia, int mes, int anio)
/* POST: Los atributos diaDelMes, mes y anios quedan modificados con el
valor de 'dia', 'mes' y 'anio' respectivamente. Al atributo diaDeLaSemana se
le asigna el dia de la semana calculado a partir de 'dia', 'mes' y 'anio'. */

/* PRE: - */
public String obtenerFechaFormal()
/* POST: Retorna un string de la forma "DD/MM/AAAA" donde DD es el valor
del atributo diaDelMes, MM es el valor del atributo mes y AAAA es el valor
del atributo anio, convertidos a string. */

/* PRE: - */
public String obtenerFechaCompleta()
/* POST: Retorna un string de la forma "DDDDDD, DD de MMMMMM de
AAAA" donde DDDDDD es el nombre del dia de la semana representado por
el atributo diaDeLaSemana, MMMMMM es el nombre del mes del a(ny)o
representado por el atributo mes y DD es el valor del atributo diaDelMes, y
AAAA es el valor del atributo anio, convertidos a string. */

/* PRE: - */
public int obtenerDiaDelMes()
/* POST: Retorna el atributo diaDelMes. */

/* PRE: - */
public int obtenerDiaDeLaSemanaInt()
/* POST: Retorna el atributo diaDeLaSemana. */

/* PRE: - */
public String obtenerDiaDeLaSemanaStr()
/* POST: Retorna el nombre del dia de la semana representado por el
atributo diaDeLaSemana. */

/* PRE: - */
public int obtenerMesInt()
/* POST: Retorna el atributo mes. */

/* PRE: - */
public String obtenerMesStr()
/* POST: Retorna el nombre del mes del a(ny)o representado por el atributo
mes. */
```

```
/* PRE: - */  
public int obtenerAnio()  
/* POST: Retorna el atributo anio. */  
  
/* PRE: 'dia', 'mes' y 'anio' representan una fecha valida segun el calendario  
Gregoriano. */  
private int calcularDiaDeLaSemana(int dia, int mes, int anio)  
/* POST: Retorna el dia de la semana (numero) calculado a partir de 'dia',  
'mes' y 'anio'. */
```

6. Clase Hora

Nombre de la clase:	Hora (public class Hora)
Descripción:	La clase "Hora" implementa una representación de una hora mediante la hora el minuto y el segundo exactos. Nos permite obtener la hora actual o seleccionar una deseada.
Cardinalidad:	Múltiple
Package:	Tiempo

Atributos:

private int hora;

- Valor inicial: -
- Descripción: Dada una hora en formato HH:MM:SS, 'hora' indica el valor de HH.
- Estático: NO

private int minuto;

- Valor inicial: -
- Descripción: Dada una hora en formato HH:MM:SS, 'minuto' indica el valor de MM.
- Estático: NO

private int segundo;

- Valor inicial: -
- Descripción: Dada una hora en formato HH:MM:SS, 'segundo' indica el valor de SS.
- Estático: NO

Operaciones:

/* PRE: - */

public Hora()

/* POST: Se crea una nueva instancia de Hora con todos sus atributos inicializados a 0. */

/* PRE: - */

public void ahora()

/* POST: Todos los atributos de Hora quedan modificados de tal manera que su valor se corresponde con la hora actual. */

/* PRE: 'hora', 'minuto' y 'segundo' representan una hora valida, es decir, 'hora' esta entre 0 y 23, 'minuto' entre 0 y 59, y 'segundo' entre 0 y 59. */

public void ponerHora(int hora, int minuto, int segundo)

/* POST: Los atributos hora, minuto y segundo quedan modificados con el valor de 'hora', 'minuto' y 'segundo' respectivamente. */

```
/* PRE: - */  
public String obtenerHoraFormal()  
/* POST: Retorna un string en formato HH:MM:SS donde HH es el valor del  
atributo hora, MM es el valor del atributo minuto y SS es el valor del atributo  
segundo, convertidos a string. */  
  
/* PRE: - */  
public int obtenerHora()  
/* POST: Retorna el atributo hora. */  
  
/* PRE: - */  
public int obtenerMinuto()  
/* POST: Retorna el atributo minuto. */  
  
/* PRE: - */  
public int obtenerSegundo()  
/* POST: Retorna el atributo segundo. */
```

7. Clase Tablero

Nombre de la clase:	Tablero (public abstract class Tablero)
Descripción:	Representa el conjunto estructurado de casillas así como sus características (altura y anchura), pero no su contenido.
Cardinalidad:	Múltiple
Package:	Dominio

Atributos:

private String id;

- Valor inicial: -
- Descripción: El atributo "id" tiene como función identificar cada una de las instancias de tablero.
- Estático: NO

private int anchura;

- Valor inicial: -
- Descripción: Representa el número de casillas que tendrá el tablero horizontalmente.
- Estático: NO

private int altura;

- Valor inicial: -
- Descripción: Representa el número de casillas que tendrá el tablero verticalmente.
- Estático: NO

Operaciones:

/* PRE: No existe ninguna instancia de Tablero con id = 'id'. 'id' únicamente contiene caracteres alfanuméricos. 'anchura' y 'altura' son valores mayores que 0. */

public Tablero(String id, int anchura, int altura)

/* POST: Se crea una instancia de Tablero con id = 'id', anchura = 'anchura' y altura = 'altura'. */

/* PRE: No existe ninguna instancia de Tablero con id = 'id'. 'id' únicamente contiene caracteres alfanuméricos. 'tablero' existe. */

public Tablero(String id, Tablero tablero)

/* POST: Se crea una instancia de Tablero con id = 'id' y el resto de atributos como copia de los atributos de 'tablero'. */

/* PRE: - */

public String getTablero()

/* POST: Retorna el atributo id. */


```
/* PRE: - */  
public int getAnchura()  
/* POST: Retorna el atributo anchura. */  
  
/* PRE: - */  
public int getAltura()  
/* POST: Retorna el atributo altura. */  
  
/* PRE: - */  
public abstract void resetearTablero()  
/* POST: Se reconfigura el tablero con el estado que tenía al inicio de la  
partida. */
```

8. Clase Repositorio

Nombre de la clase:	Repositorio (public class Repositorio)
Descripción:	Representa el contenedor de los tableros solución que el sistema ha validado al usuario y los que ha generado a través de los algoritmos.
Cardinalidad:	Única
Package:	Dominio

Atributos:

private List<String> tableros;

- Valor inicial: null
- Descripción: Representa el conjunto de los identificadores de los tableros solución del sistema.
- Estático: NO

Operaciones:

/* PRE: - */

public Repositorio()

/* POST: Crea una instancia de Repositorio sin ningún tablero solución asociado. */

/* PRE: - */

public List<String> getTableros()

/* POST: Devuelve el conjunto de identificadores de todos los tableros solución del repositorio */

/* PRE: El tablero con identificador = 'tablero' no existe en el Repositorio. 'tablero' unicamente contiene caracteres alfanumericos. */

public void addTablero(String tablero)

/* POST: Inserta el tablero con identificador = 'tablero' al repositorio. */

9. Clase Casilla

Nombre de la clase:	Casilla (public class Casilla)
Descripción:	Representa una casilla del juego, ubicada en unas coordenadas concretas de un tablero, ocupada por un valor y que puede estar prefijada al generar el tablero.
Cardinalidad:	Múltiple
Package:	Dominio

Atributos:

private int x;	
• Valor inicial:	-
• Descripción:	Corresponde a la coordenada x de la casilla.
• Estático:	NO
private int y;	
• Valor inicial:	-
• Descripción:	Corresponde a la coordenada y de la casilla.
• Estático:	NO
private int valor;	
• Valor inicial:	0
• Descripción:	Indica el valor de la casilla; 0 si no está ocupada.
• Estático:	NO
private Boolean prefijada;	
• Valor inicial:	false
• Descripción:	Indica si la casilla figura en la configuración inicial del tablero.
• Estático:	NO

Operaciones:

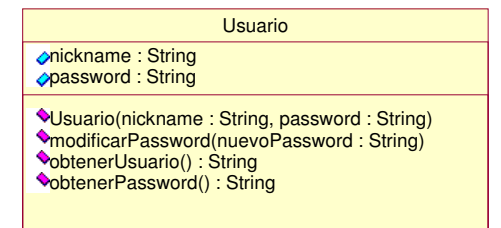
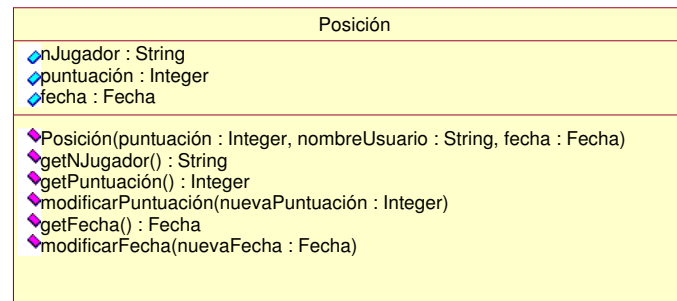
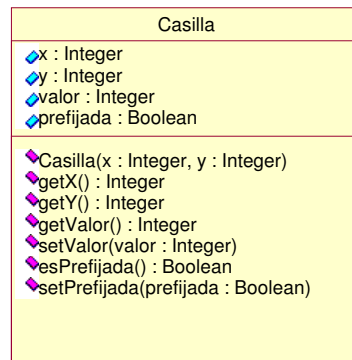
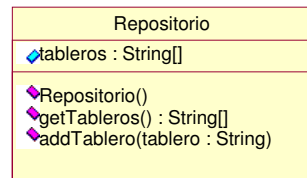
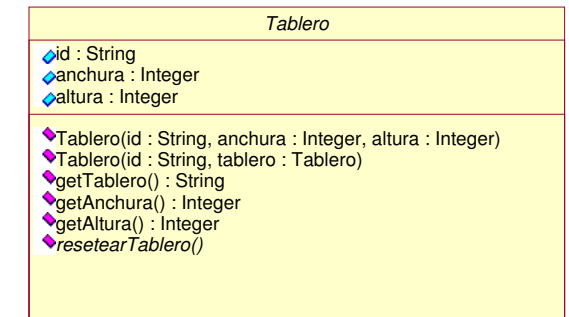
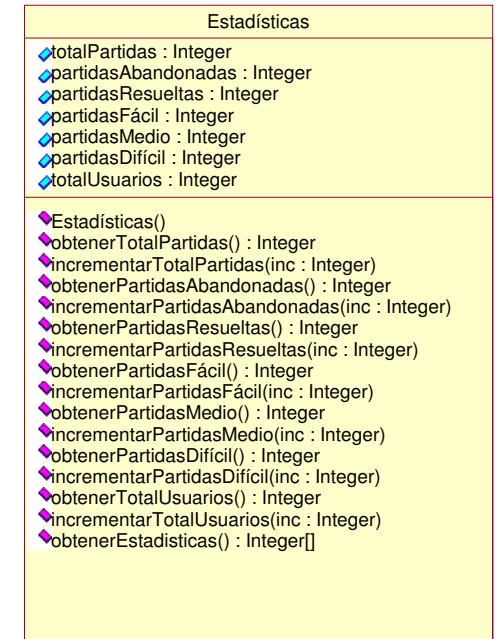
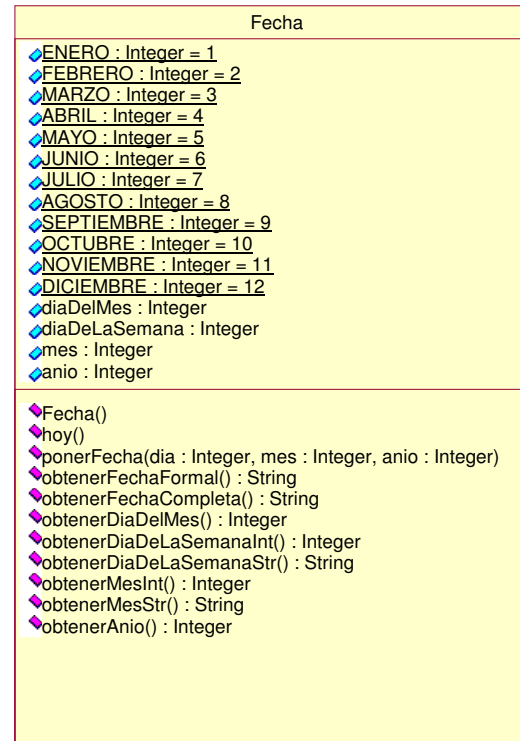
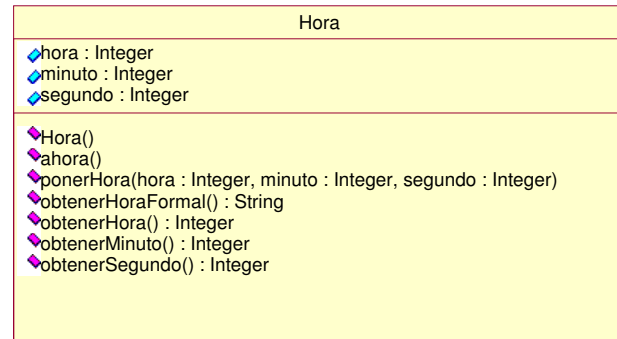
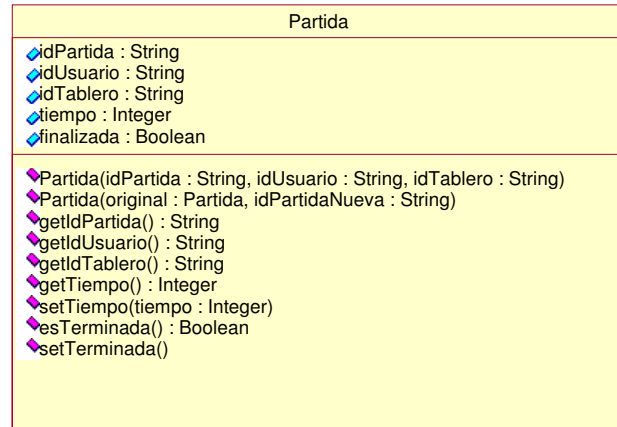
```
/* PRE: x >= 0, y >= 0; no existe una casilla con los mismo 'x' e 'y' en el
tablero y/o bloque al que pertenece la casilla. */
public Casilla(int x, int y)
/* POST: Crea una nueva instancia de casilla con x = 'x' e y = 'y'. */

/* PRE: - */
public int getX()
/* POST: Retorna la coordenada x de la casilla. */

/* PRE: - */
public int getY()
/* POST: Retorna la coordenada y de la casilla. */
```

```
/* PRE: - */  
public int getValor()  
/* POST: Retorna el valor de la casilla; si no está ocupada retorna 0. */  
  
/* PRE: - */  
public Boolean esPrefijada()  
/* POST: Retorna el valor del atributo prefijada. */  
  
/* PRE: Valor es un numero natural mayor o igual que 0. */  
public void setValor(int valor)  
/* POST: Asigna al atributo valor de la casilla el parametro de entrada. */  
  
/* PRE: - */  
public void setPrefijada(Boolean prefijada)  
/* POST: Asigna al atributo prefijada de la casilla el parametro de entrada. */
```

Diagrama de Clases Compartidas



Reparto de responsabilidades de las clases compartidas

<i>Clase</i>	<i>Usada por subgrupo</i>	<i>Desarrollada por subgrupo</i>
Partida	11.1 11.2 11.3	11.1
Usuario	11.1 11.2 11.3	11.3
Posición	11.1 11.2 11.3	11.1
Estadísticas	11.1 11.2 11.3	11.2
Fecha	11.1 11.2 11.3	11.3
Hora	11.1 11.2 11.3	11.3
Tablero	11.2 11.3	11.3
Repositorio	11.2 11.3	11.2
Casilla	11.2 11.3	11.2

Autores

Grupo 11.1

David Castaño González (david.castano.gonzalez)
Sara González Guerrero (sara.gonzalez.guerrero)
Guillermo Muñiz Rodríguez (guillermo.muniz)
Rodolfo Núñez Del Río (rodolfo.nunez)

Grupo 11.2

Beatriz Casín Núñez (beatriz.casin)
Jonathan Florido Moreno (jonathan.florido)
Carlos Sans García (carlos.sans)

Grupo 11.3

Daniel Camarasa Escubedo (daniel.camarasa)
Alex Catarineu Morales (alex.catarineu)
Miguel Ángel Vico Moya (miguel.angel.vico)