Trivial2a

Documentación

Alves De Araujo, Ruan Henrique - UO246819

Baragaño González, Samanta - UO231533

Fraile Muñiz, Ivana - UO216746

García Álvarez, Ángela - UO217608

González Fernández, Cristian - UO231602

Montero Hernández, Jose Antonio - UO179746

Piris García, María - UO230886

Sánchez Illisaca, Jhonny Cristian - UO225242

Velázquez Vico, Álvaro - UO232635

Índice

Indice	
1. Descripción del proyecto	
2. Metodología utilizada	
3. Identificación de stakeholders(Descripción)	
Identificación inicial de los atributos de calidad Primer acercamiento a la solución y riesgos	
5.1 Riesgos relacionados con la solución	
6. Lista actualizada de stakeholders	
Definición (comunes)	
Definición (propias)	
7. Lista de atributos de calidad	
8. Atributos de calidad e interesados	
9. Descripción del negocio de la solución	11
9.1.1. Solución	
9.1.2 Tecnologías	
10. Escenarios de calidad	
11. Vistas	
Sistema Summary	
Details	
_	
UGI	15
gameRequests	16
Business	17
4	
GameAPI	18
dbRequest	10
Persistence	20
PersistenceServices	
Despliegue	
Summary	
Details	
Cliente	23
 UGI	24
<u> </u>	24
GameAPI	24
Game	25
Board	25
Dice	26
Board Ontions Factory	27
BoardOptionsFactory	21
PersistenceServices	27
4	
ScoreDAO	28
QuestionDAO	28
HSQLDB	29
StadisticDAO	20
SIGUISTICUAU	29
UserDao	30
	31

	maryils	
	UGI	33
	Model	33
	Factories	34
	Conf	34
	Business	35
뒫	GameAPI	35
	GameClasses	36
	Exceptions	37
뢷	Game	37
뢷	Board	38
	FileLoaders	38
륃	Dice	39
륃	BoardOptionsFactory	39
	Impl	40
	Persistence	40
Ę	PersistenceServices	41
	Impl	41
2	QuestionDAO	42
Ę	ScoreDAO	42
Ę	StadisticDAO	43
12 M	UserDao	43 44

1. Descripción del proyecto

La empresa NoGame ya posee el sistema que le permite obtener preguntas, y han decidido comenzar la construcción del juego Trivial primeramente disponible solo para ordenadores de sobremesa.

Las preguntas que se pretende utilizar en el juego deben de tomarse de la bases de datos de la primera fase y se mostrará un interfaz gráfico para que los usuarios puedan jugar al trivial.

El trivial original consta de una rueda con seis radios, sin embargo se podrá definir una vista más simplificada, mediante la utilización de una línea recta con una casilla final. El número total de jugadores en una partida debe ser mínimo 1 y máximo el número de colores disponible en los quesitos.

Almacenar información sobre los jugadores del trivial supone un interés a la compañía, por lo cual cada jugador dispondrá de un login y una contraseña. Cuando se inicie el juego se mostrará una ventana de configuración donde se podrá especificar los concursantes que jugarán y donde los mismos se autenticarán. A partir de ahí la maquina será la responsable de lanzar el dado, esperará la respuesta de que casilla moverse y enseñara la pregunta correspondiente a la categoría de la misma.

En el caso de que un jugador acierte una pregunta de una determinada categoría, se le asigna el quesito del color correspondiente y se le permite seguir jugando. En caso contrario, pasa su turno al siguiente concursante. El jugador que complete el quesito con todos los colores podrá proceder a la casilla final. Si acierta la pregunta final, el usuario termina el juego como ganador de la partida.

Es muy importante para la compañía, que el juego sea utilizado como un mecanismo de aprendizaje y por ello desea que se incorpore una ventana de administración para que los usuarios que poseen esos privilegios puedan acceder a estadísticas sobre los usuarios que han jugado, el número de preguntas acertadas, las preguntas más difíciles, etc.

En un futuro la empresa, piensa extender la aplicación para que la misma funcione en distintas plataformas y por ello les gustaría experimentar con distintas opciones visuales del juego. Para ello debe de haber una clara separación entre la representación visual del funcionamiento lógico del juego y de la interacción con el usuario. A raíz de eso, han propuesto a los desarrolladores que el usuarios puedan modificar la interfaz de usuario si así lo desean.

2. Metodología utilizada

Siguiendo las pautas vistas en clases de teoría de la asignatura, emplearemos el método de desarrollo ADD (Atribute-Driven Design).

Identificación de stakeholders(Descripción)

• Empresa (NoGame):

Son los dueños del proyecto y responsables de las decisiones finales, tanto a nivel de requisitos como de presupuesto.

Entre sus objetivos están:

- a) Obtener una aplicación que atraiga al público y que sea funcional.
- b) Conseguir que la aplicación le suponga un valor añadido.
- c) Tener la aplicación en el plazo estipulado.
- d) Conseguir que el proyecto no tenga un coste elevado.
- e) Conseguir que la aplicación sea una herramienta para obtener estadísticas.
- f) Tener la posibilidad de hacer que la aplicación sea multiplataforma.

Desarrolladores (Equipo 2a)

Son los encargados de la construcción del software. Su responsabilidad es codificar, probar y mantener el código del sistema desarrollado.

Entre sus objetivos están:

- a) Conseguir una aplicación modular.
- b) Conseguir una aplicación que permita intercambiar look & fell.
- c) Hacer que el proyecto sea rentable.
- d) A) Hacer un sistema mantenible, que permita ampliar su funcionalidad.

• Arquitectos del sistema (Equipo 2a)

Son los encargados de documentar, estructurar, diseñar y mantener el sistema que da solución al proyecto encargado por la empresa.

Entre sus objetivos están:

- a) Garantizar un sistema reutilizable y de fácil mantenimiento.
- b) Acordar con la empresa unos requisitos que puedan cumplirse.
- c) Conseguir un proyecto rentable
- d) Conseguir un proyecto que les aporte conocimientos nuevos y experiencia

Gestores de bases de datos (Equipo 2a)

Son los encargados de diseñar y mantener las bases de datos. También deben encargarse de la seguridad e integridad de los datos.

Entre sus objetivos están:

- a) Garantizar la seguridad en el acceso a los datos
- b) Garantizar la integridad de los datos almacenados

Cliente (Usuario jugador)

Son aquellas personas que utilizarán la aplicación.

Entre sus objetivos están:

- a) Instalación de la aplicación sencilla
- b) Interfaz intuitiva que haga sencillo el manejo de la aplicación
- c) Confiabilidad en la aplicación

4. Identificación inicial de los atributos de calidad

La aplicación que se debe desarrollar debe cumplir con los siguientes atributos de calidad:

- Interactividad: La aplicación debe comunicarse con el usuario de forma interactiva en todas sus formas, ya que se trata de un juego mayormente visual cuyo propósito es él de ser entretenido.
- **Usabilidad:** Es imprescindible que la aplicación no le suponga ningún tipo de dificultades de utilización al usuario, es sumamente importante que la curva de aprendizaje sea la mínima posible.
- **Seguridad:** La aplicación gestiona información sensible de los usuarios e información importante para la empresa. Por ello se deben tomar todas las medidas de protección necesarias para garantizar la seguridad tanto del sistema como de sus usuarios.
- Mantenible: La evolución del proyecto es un hecho irrefutable, por lo tanto el tiempo en futuras modificaciones y/o extensiones deberían de llevarse a cabo sin muchos problemas.
- **Reusable:** Las partes que integran el sistema deben de ser lo suficientemente independientes para que las mismas puedan ser aplicadas en el futuro. Sean para aplicaciones similares o no.
- Rendimiento: Se debe garantizar un tiempo correcto del procesamiento y ejecución de los algoritmos involucrados en la operación de juego. La aplicación debe correr fluida sin ningún tipo de lag o delay.

5. Primer acercamiento a la solución y riesgos

Para llevar a cabo este proyecto, se

Una vez estudiados los requisitos y tras una serie de propuestas de solución, se opta por desarrollar el juego siguiente el estilo MVC, y separación entre capas.

La primera capa será la GUI, contendrá todas la lógica necesaria para representar las abstracciones de las vistas de nuestro modelo. Dicha capa se implementará utilizando la librería grafica de java Swing.

La segunda capa será la Lógica, contendrá todas las reglas y lógica de negocia asociada al juego y al correcto funcionamiento de la aplicación. Esta capa servirá de puente entre la capa de interfaz y la siguiente capa.

La ultima capa será la capa de Persistencia. responsable de gestionar los datos de los usuarios y las estadísticas, comunicándose con nuestro SGBD.

Esta solución nos brinda una aplicación multiplataforma y totalmente mantenible, ya que reducimos al mínimo el acoplamiento entre las distintas capas que la constituyen.

5.1 Riesgos relacionados con la solución

El sistema así elegido se enfrenta a algunos riesgos que es preciso identificar:

- **Pruebas unitarias.** El programa debe poder ser testeable automáticamente con excepción de la interfaz grafica.
- Extensión a nuevas plataformas. Relacionado con su reusabilidad y mantenibilidad, el hecho de que exista una división entre capas bien definidas nos permite una extensión fácil en el futuro.
- Posibilidad de cambios de interfaz. Relacionado con el punto anterior , una de las principales características buscadas es el poder cambiar el representación visual sin tener que modificar ninguna de las demás capas.

6. Lista actualizada de stakeholders

Definición (comunes)

1. Empresa NoGame

- Son los propietarios y responsables principales de la financiación de la aplicación.
- Objetivos:
 - o Reducir el coste del desarrollo de la aplicación lo máximo posible
 - Obtener una aplicación multiplataforma.
 - o Aumentar al máximo los beneficios conseguidos por ésta misma.
 - o Conseguir que el juego sea funcional, atractivo y barato.

2. Desarrolladores juego Trivial

- Equipo responsable del desarrollo del funcionamiento de la aplicación.
- Objetivos:
 - o Construir una aplicación segura y funcional.
 - o Llevar a cabo implementaciones interesantes.
 - o Desarrollar un proyecto beneficioso intelectual y económicamente.

3. Arquitectos de la aplicación

- Equipo responsable de todo lo que tenga que ver con el análisis del proyecto
- Objetivos:
 - Garantizar una aplicación reutilizable en distintos ámbitos y de fácil mantenimiento.
 - Obtener experiencia con la realización de este proyecto.
 - Seleccionar el mejor estilo arquitectónico para lograr una aplicación que cumpla con creces las necesidades del cliente.

4. Usuarios jugadores

- Son las personas que ejecutaran la aplicación
- Objetivos:
 - o Disfrutar del entretenimiento ofrecido por el juego.
 - o Utilizar el juego como una herramienta de aprendizaje.

5. Usuario administrador

- Son los usuarios con privilegios suficientes para ver las estadísticas de juegos.
- Objetivos:
 - o Encontrarse con una interfaz sencilla e intuitiva.
 - Obtener información suficiente y clara sobre las partidas y los usuarios jugadores.

Definición (propias)

6. Gestores de bases de datos

 Son los responsables de mantener tanto el correcto funcionamiento del sistema de gestión de bases de datos, como de la integridad y consistencia de la bases de datos

Código	Stakeholder	Intereses
ST-01	Empresa NoGame	Bajo conste de desarrollo y mantenibilidad del sistema.
		2. Posibilidad de extensión de la aplicación, y que la misma sea multiplataforma.

ST-02	Desarrolladores juego Trivial		Sencillez a la hora de mantener el sistema y ampliar su funcionalidad. Confiar en la seriedad y profesionalidad del cliente. Llevar a cabo requisitos que se puedan cumplir.
ST-03	Arquitectos de la aplicación	 2. 	Garantizar una aplicación de fácil mantenimiento y reutilizable.
ST-04	Usuarios jugadores	 2. 3. 4. 	Divertirse con la experiencia de juego que proporciona Trivial. Sentirse seguro con la protección de sus datos. Instalar el juego y poder ejecutarlo aunque su máquina no sea demasiado potente. Sencillez de configuración de las partidas.
ST-05	Usuario administrador	1.	Encontrarse con una interfaz sencilla para obtener información estadísticas
ST-06	Gestores de bases de datos	 2. 	Ofrecer un buen nivel de seguridad de acceso a la bases de datos. Construir una bases de datos valiosa y consistente.

7. Lista de atributos de calidad

Código	Descripción	Tipo de Atributo
AT001	El juego debe proporcionarle al usuario una	Interactividad
	interfaz que involucre la interacción con los jugadores.	
AT002	Las representaciones visuales del juego deben ser divertidas y atractivas.	Interactividad
AT003	Las acciones de tirar el dado y mover casillas	Usabilidad

	debe ser sencillas e intuitivas.	
AT004	Se debe asegurar los datos almacenados e introducidos por los usuarios en la aplicación.	Seguridad
AT005	Consistencia e integridad de la bases de datos de la compañía.	Seguridad
AT006	Seguridad de acceso a la base de datos.	Seguridad
AT007	Denegar el acceso a usuarios ilegítimos de la aplicación.	Seguridad
AT008	La aplicación debe estar separada en capas para evitar el acoplamiento.	Mantenible
AT009	Las modificaciones que se lleven a cabo en las implementaciones concretas no deberían implicar la re compilación total de la aplicación.	Mantenible
AT010	Facilidad al indicar los posibles movimientos de un jugador.	Usabilidad
AT011	La lógica de negocio empleada debe ser totalmente independiente de las vistas que se utilicen.	Reusable
AT012	Los algoritmos que calculan las casillas posibles deben garantizar un tiempo de ejecución bajo.	Rendimiento

8. Atributos de calidad e interesados

Atributos	ST-01	ST-02	ST-03	ST-04	ST-05	ST-06
Stakeholders						
AT001				٧	٧	
AT002				٧		
AT003				٧		
AT004	٧	٧	٧	٧	٧	٧
AT005	٧					٧
AT006	٧					٧
AT007						٧
AT008	٧	٧	٧			
AT009		٧	٧			
AT010				٧		
AT011	٧	٧	٧			
AT012		٧		٧		

9. Descripción del negocio de la solución.

En este apartado se describirá el modelo de negocio del programa.

El modelo de negocio de la aplicación está constituido por una aplicación de escritorio multiplataforma con interfaz gráfico, que se ejecuta en el ordenador del cliente.

9.1.1. Solución

Todos los niveles de la aplicación se encuentran separados entre capas, siguiendo las características básicas del patrón MVC con el fin de lograr múltiples vistas del mismo modelo, tal y como se requiere:

- Persistencia: Todas las operaciones y procesos relacionados con tareas de bases de datos (CRUD) se encuentran en esta capa. La utilización de la misma se hará mediante DAOs.
- Lógica: Esta capa debe contener únicamente la lógica de negocio involucrada en el funcionamiento del juego Trivial.
- GUI: Posee las distintas abstracciones visuales del modelo, y debe comunicarse únicamente con la capa de lógica.

Afín de mejorar el mantenimiento de la aplicación y lograr el mínimo acoplamiento posible entre capas, la implementación de factorías y fachadas se hacen imprescindibles.

9.1.2 Tecnologías

En este apartado se exponen las distintas opciones tecnológicas elegidas.

- Lenguaje de programación : El juego se implementara en Java , ya que nos ofrece la posibilidad de ser multiplataforma tal y como necesita el cliente.
- Bases de Datos: La gestión de usuarios y partidas se realizará con el SGBD de bases relacionales HSQLDB, ya que nos brinda la sencillez que necesitamos para esta ocasión.

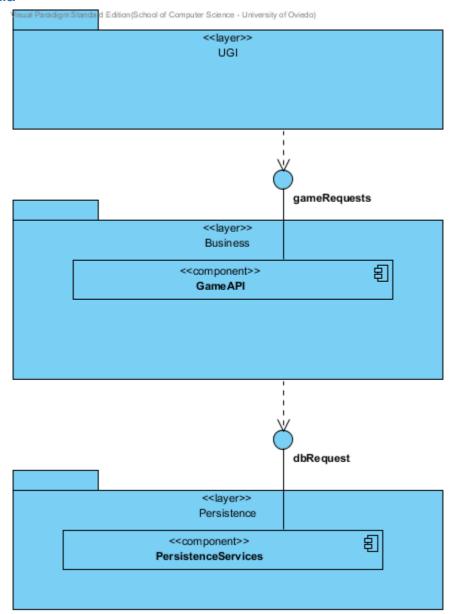
10. Escenarios de calidad

Nº	Fuente	Estímulo	Entorno	Artefa cto	Respuesta	Medición de respuesta	Atribu to de calida d
1	Usuario jugador	Contestar a una pregunta	Producción.	GUI	Mostrar las alternativas posibles.	La selección se debe hacer con un solo clic.	AT001
2	Usuario jugador	Mover a una casilla.	Producción.	GUI	Moverse hasta una casilla.	El movimiento debe ser animado y menor a 1 segundo.	AT002 AT001
3	Usuario jugador	Lanzar el dado	Producción.	GUI	Sacar un numero del dado.	Se obtiene un numero del dado Con apenas un clic y después de una animación en menos de 1 segundo.	AT003 AT002
4	Desarrollad ores	Cambiar la implementació n de una vista.	Desarrollo	GUI	Aceptación e integración de la nueva vista.	Ningún cambio en las demás capas de la arquitectura.	AT008
5	Usuario jugador	Ver informaciones de las partidas, sin privilegios.	Producción.	Lógica	Acceso a la información.	Acceso denegado por no tener suficientes permisos.	AT004
6	Usuario administra dor.	Modificar un valor estadístico.	Producción.	Lógica.	Datos intactos.	100% de los datos siguen intactos.	AT005
7	Intento no autorizado de acceso a BBDD	Intento acceso a BBDD	Producción.	Persistenc ia.	Acceso no autorizado rechazado	No se ha realizado ningún cambio en la base de datos	AT007 AT006
8	Desarrollad ores	Modificación en la lógica.	Desarrollo	Lógica.	Compilar el código	Capa de persistencia y GUI no necesitan ser compiladas.	AT009
9	Aplicación	Terminar partida.	Producción.	Persistenc ia.	No perder información de las partidas y usuarios.	Se mantiene el 100% de la información.	AT005
10	Usuario jugador	Mover a una casilla.	Producción.	GUI.	Ver las opciones posibles.	Los destinos se destacan visiblemente.	AT010
11	Desarrollad ores	Añadir una nueva vista.	Desarrollo.	Lógica.	Funcionamiento correcto.	No es necesario modificar ningún aspecto de la lógica.	AT011
12	Usuario jugador	Hacer un movimiento.	Producción.	Lógica.	Devolver los posibles movimientos.	El algoritmo debe tardar menos de 2 segundos.	AT012

11. Vistas

Class Diagram

Sistema



Name	Value		
Name	Sistema		
Author	José Antonio Montero Hernández		
Create Date Time	12-abr-2015 21:51:37		
Last Modified	12-abr-2015 23:14:39		
Shape Presentation Option	0		

Summary

Name	Documentation			
UGI	User Graphic Interface. Paquete que contiene la interface de usuario, toda la capa de vista.			
gameRequests	Agrupación de los métodos de control del juego que proporciona la API de la capa de negocio.			
Business	Paquete que contiene la lógica del juego, todo lo correspondiente a la capa de negocio.			
GameAPI API utilizada por la Interface Gráfica para comunicarse con la lógica de la aplicación.				
Agrupación de los métodos de acceso a datos que proporciona la sublayer de servicio de la layer d persistencia.				
Persistence	Capa que contiene la lógica de base de datos, toda la capa de persistencia			
PersistenceServ ices	Sublayer de servicio utilizada por la capa de persistencia para ofrecer sus servicios a la capa de negocio.			

Documentation

La vista de sistema describe la división en layers del sistema, así como sus interacciones y clases que las aislan.

Details



Name	Value					
Documentation	User Graphic Interface. Paquete que contiene la interface de usuario, toda la capa de vista.					
Abstract	false	false				
Leaf	false	false				
Root	false					
Stereotypes	layer					
Visibility	public					
Project Management	Name Value					
	Author José Antonio Montero Hernández					
	Create Date Time 12-abr-2015 21:22:14					
	Last Modified	12-abr-2015 22:19:13				

Relationships

Unnamed Import					
То	gameRequests				
Visibility	Unspecified	Unspecified			
Project Management	Name Value				
	Author	José Antonio Montero Hernández			
	Create Date Time 12-abr-2015 22:17:17				
	Last Modified	12-abr-2015 22:29:58			

冒

gameRequests

gamerequests			
Name	Value		
Documentation	Agrupación de los métodos de control del juego que proporciona la API de la capa de negocio.		
Active	false		
Business Key Mutable	true		
Business Model	false	false	
Visibility	public		
Leaf	false		
Root	false		
Stereotypes	Interface		
Project Management	Name Value		
	Author José Antonio Montero Hernández		
	Create Date Time 12-abr-2015 22:18:34		
	Last Modified 12-abr-2015 23:10:30		

Relationships

Unnamed Realization		
То	3 GameAPI	
Visibility	Unspecified	
Project Management	Name Value	
	Author	José Antonio Montero Hernández
	Create Date Time	12-abr-2015 22:18:34

	Last Modified	12-abr-2015 22:29:58	
Unnamed Import	Unnamed Import		
From	□ UGI		
Visibility	Unspecified		
Project Management	Name	Value	
	Author	José Antonio Montero Hernández	
	Create Date Time	12-abr-2015 22:17:17	
	Last Modified	12-abr-2015 22:29:58	

Business

Name	Value	
Documentation	Paquete que contiene la lógica del juego, todo lo correspondiente a la capa de negocio.	
Abstract	false	
Leaf	false	
Root	false	
Stereotypes	layer	
Visibility	public	
Project Management	Name Value	
	Author	José Antonio Montero Hernández
	Create Date Time 10-abr-2015 19:09:45	
	Last Modified	12-abr-2015 22:29:58

Children

Name	Documentation
a GameAPI	API utilizada por la Interface Gráfica para comunicarse con la lógica de la aplicación.
Exceptions	Paquete que contiene las excepciones personalizadas que se puedan lanzar desde la capa de negocio.
FileLoaders	Paquete que contiene las clases que se utilizan para cargar datos desde ficheros.
GameClasses	Paquete que contiene las clases necesarias para el funcionamiento y mantenimiento de la partidas.

- Impi	Paquete que contiene la implementación de las clases que se manejan de forma genérica a
	nivel de la capa de business.

Relationships

Unnamed Dependency		
То	dbRequest	
Visibility	Unspecified	
Project Management	Name Value	
	Author	José Antonio Montero Hernández
	Create Date Time	12-abr-2015 22:22:36
	Last Modified	12-abr-2015 22:29:58

3 GameAPI

Name	Value	
Documentation	API utilizada por la Interface Gráfica para comunicarse con la lógica de la aplicación.	
Active	false	
Business Key Mutable	true	
Business Model	false	
Visibility	public	
Abstract	false	
Leaf	false	
Root	false	
Indirectly Instantiated	true	
Project Management	Name	Value
	Author	José Antonio Montero Hernández
	Create Date Time 12-abr-2015 21:31:36	
	Last Modified 12-abr-2015 22:29:58	

Relationships

Unnamed Realization	
From	gameRequests

Visibility	Unspecified	
Project Management	Name	Value
	Author	José Antonio Montero Hernández
	Create Date Time	12-abr-2015 22:18:34
	Last Modified	12-abr-2015 22:29:58

dbRequest

ubkequest		
Name	Value	
Documentation	Agrupación de los métodos de acceso a datos que proporciona la sublayer de servicio de la layer de persistencia.	
Active	false	
Business Key Mutable	true	
Business Model	false	
Visibility	public	
Leaf	false	
Root	false	
Stereotypes	Interface	
Project Management	Name Value	
	Author	José Antonio Montero Hernández
	Create Date Time 12-abr-2015 22:22:03	
	Last Modified 12-abr-2015 23:10:30	

Relationships

Unnamed Realization		
То	PersistenceServices	
Visibility	Unspecified	
Project Management	Name Value	
	Author	José Antonio Montero Hernández
	Create Date Time	12-abr-2015 22:22:03
	Last Modified	12-abr-2015 22:29:58
Unnamed Dependency		
From	Business	

Visibility	Unspecified	
Project Management	Name	Value
	Author	José Antonio Montero Hernández
	Create Date Time	12-abr-2015 22:22:36
	Last Modified	12-abr-2015 22:29:58

Persistence

Name	Value		
Documentation		Capa que contiene la lógica de base de datos, toda la capa de persistencia	
Abstract	false	false	
Leaf	false	false	
Root	false	false	
Stereotypes	layer	layer	
Visibility	public	public	
Project Management	Name	Value	
	Author	José Antonio Montero Hernández	
	Create Date Time	12-abr-2015 21:21:04	
	Last Modified	12-abr-2015 23:00:03	

Children

Name	Documentation	
PersistenceServices	Sublayer de servicio utilizada por la capa de persistencia para ofrecer sus servicios a la capa de negocio.	
impl impl	Paquete que contiene la implementación de las clases que se manejan de forma genérica a nivel de la capa de persistencia.	
QuestionDAO	Componente que presta servicios de acceso a los datos de preguntas.	
3 StadisticDAO	Componente que presta servicios de acceso a los datos de estadísticas.	
UserDao	Componente que presta servicios de acceso a los datos de usuarios.	
ScoreDAO	Componente que presta servicios de acceso a los datos de puntuaciones.	

PersistenceServices

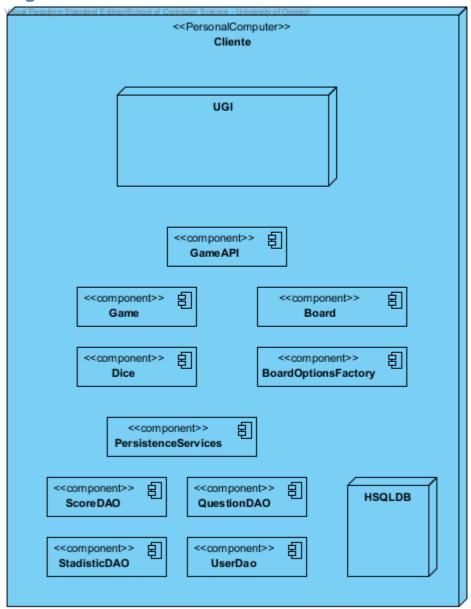
Name	Value	
Documentation	Sublayer de servicio utilizada por la capa de persistencia para ofrecer sus servicios a la capa de negocio.	
Active	false	
Business Key Mutable	true	
Business Model	false	
Visibility	public	
Abstract	false	
Leaf	false	
Root	false	
Indirectly Instantiated	true	
Project Management	Name	Value
	Author	José Antonio Montero Hernández
	Create Date Time	12-abr-2015 22:07:00
	Last Modified	12-abr-2015 22:29:58

Relationships

Unnamed Realization		
From	dbRequest	
Visibility	Unspecified	
Project Management	Name	Value
	Author	José Antonio Montero Hernández
	Create Date Time	12-abr-2015 22:22:03
	Last Modified	12-abr-2015 22:29:58

Deployment Diagram

Despliegue



Name	Value
Name	Despliegue
Author	José Antonio Montero Hernández
Create Date Time	10-abr-2015 19:08:45
Last Modified	12-abr-2015 23:10:30
Shape Presentation Option	0

Summary

Name	Documentation	
Cliente	Ordenador personal de cada usuario en el que ejecutará el juego.	
■ UGI	Nodo que representa las diferentes clases y ventanas de la interface de usuario.	
3 GameAPI	API utilizada por la Interface Gráfica para comunicarse con la lógica de la aplicación.	
3 Game	Gestiona el funcionamiento de la partida y coordina a los componentes implicados en su desarrollo.	
3 Board	Mantiene la representación lógica del tablero durante la partida y provee de información y cálculos sobre su estado.	
Dice	Provee de tiradas aleatorias a la vez que permite mantener un bloque sobre los lanzamientos y sobre el último resultado.	
BoardOptionsFa ctory	Genera las diferentes topologías de tablero disponibles en el sistema.	
PersistenceServ ices	Sublayer de servicio utilizada por la capa de persistencia para ofrecer sus servicios a la capa de negocio.	
ScoreDAO	Componente que presta servicios de acceso a los datos de puntuaciones.	
QuestionDAO	Componente que presta servicios de acceso a los datos de preguntas.	
■ HSQLDB	Base de datos local de la aplicación	
\$ StadisticDAO	Componente que presta servicios de acceso a los datos de estadísticas.	
UserDao	Componente que presta servicios de acceso a los datos de usuarios.	

Details



Name	Value
Documentation	Ordenador personal de cada usuario en el que ejecutará el juego.
Active	false
Business Key Mutable	true
Business Model	false
Visibility	public

Abstract	false	
Leaf	false	
Root	false	
Stereotypes	PersonalComputer	
Project Management	Name	Value
	Author	José Antonio Montero Hernández
	Create Date Time	12-abr-2015 22:36:10
	Last Modified	12-abr-2015 23:00:03

Nested Nodes

Name	Documentation
■ HSQLDB	Base de datos local de la aplicación
■ UGI	Nodo que representa las diferentes clases y ventanas de la interface de usuario.



Name	Value	
Documentation	Nodo que representa las diferentes clases y ventanas de la interface de usuario.	
Active	false	
Business Key Mutable	true	
Business Model	false	
Visibility	public	
Abstract	false	
Leaf	false	
Root	false	
Project Management	Name	Value
	Author	José Antonio Montero Hernández
	Create Date Time	12-abr-2015 22:53:56
	Last Modified	12-abr-2015 22:54:52

GameAPI

Name	Value
Documentation	API utilizada por la Interface Gráfica para

	comunicarse con la lógi	ca de la aplicación.	
Active	false		
Business Key Mutable	true	true	
Business Model	false		
Visibility	public		
Abstract	false		
Leaf	false		
Root	false		
Indirectly Instantiated	true		
Project Management	Name	Value	
	Author	José Antonio Montero Hernández	
	Create Date Time	12-abr-2015 21:31:36	
	Last Modified	12-abr-2015 22:29:58	



Name	Value	
Documentation	Gestiona el funcionamiento de la partida y coordina a los componentes implicados en su desarrollo.	
Active	false	
Business Key Mutable	true	
Business Model	false	
Visibility	public	
Abstract	false	
Leaf	false	
Root	false	
Indirectly Instantiated	true	
Project Management	Name	Value
	Author	José Antonio Montero Hernández
	Create Date Time 12-abr-2015 22:49:37	
	Last Modified 12-abr-2015 23:07:28	



Name

Documentation	Mantiene la representación lógica del tablero durante la partida y provee de información y cálculos sobre su estado.	
Active	false	
Business Key Mutable	true	
Business Model	false	
Visibility	public	
Abstract	false	
Leaf	false	
Root	false	
Indirectly Instantiated	true	
Project Management	Name Value	
	Author	José Antonio Montero Hernández
	Create Date Time 12-abr-2015 22:50:01	
	Last Modified 12-abr-2015 23:07:29	



Name	Value		
Documentation	Provee de tiradas aleatorias a la vez que permite mantener un bloque sobre los lanzamientos y sobre el último resultado.		
Active	false		
Business Key Mutable	true		
Business Model	false	false	
Visibility	public		
Abstract	false		
Leaf	false		
Root	false		
Indirectly Instantiated	true		
Project Management	Name Value		
	Author	José Antonio Montero Hernández	
	Create Date Time	12-abr-2015 22:52:01	
	Last Modified 12-abr-2015 23:07:29		

BoardOptionsFactory

Name	Value	
Documentation	Genera las diferentes topologías de tablero disponibles en el sistema.	
Active	false	
Business Key Mutable	true	
Business Model	false	
Visibility	public	
Abstract	false	
Leaf	false	
Root	false	
Indirectly Instantiated	true	
Project Management	Name Value	
	Author	José Antonio Montero Hernández
	Create Date Time 12-abr-2015 22:52:36	
	Last Modified 12-abr-2015 23:07:29	



PersistenceServices

Name	Value	
Documentation	Sublayer de servicio utilizada por la capa de persistencia para ofrecer sus servicios a la capa de negocio.	
Active	false	
Business Key Mutable	true	
Business Model	false	
Visibility	public	
Abstract	false	
Leaf	false	
Root	false	
Indirectly Instantiated	true	
Project Management	Name Value	
	Author	José Antonio Montero Hernández
	Create Date Time	12-abr-2015 22:07:00
	Last Modified 12-abr-2015 22:29:58	

ScoreDA0

Name	Value	
Documentation	Componente que presta servicios de acceso a los datos de puntuaciones.	
Active	false	
Business Key Mutable	true	
Business Model	false	
Visibility	public	
Abstract	false	
Leaf	false	
Root	false	
Indirectly Instantiated	true	
Project Management	Name Value	
	Author	José Antonio Montero Hernández
	Create Date Time 12-abr-2015 22:45:11	
	Last Modified 12-abr-2015 23:00:03	

QuestionDAO

Name	Value	
Documentation	Componente que presta servicios de acceso a los datos de preguntas.	
Active	false	
Business Key Mutable	true	
Business Model	false	
Visibility	public	
Abstract	false	
Leaf	false	
Root	false	
Indirectly Instantiated	true	
Project Management	Name Value	
	Author	José Antonio Montero Hernández
	Create Date Time 12-abr-2015 22:46:22	
	Last Modified 12-abr-2015 23:00:03	

HSQLDB

Name	Value	
Documentation	Base de datos local de la aplicación	
Active	false	
Business Key Mutable	true	
Business Model	false	
Visibility	public	
Abstract	false	
Leaf	false	
Root	false	
Project Management	Name Value	
	Author José Antonio Montero Hernández	
	Create Date Time 12-abr-2015 22:44:14	
	Last Modified 12-abr-2015 22:50:02	

3 StadisticDAO

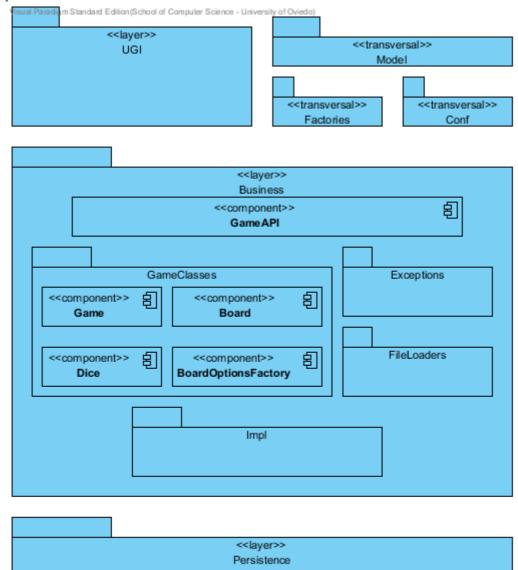
Name	Value	
Documentation	Componente que presta servicios de acceso a los datos de estadísticas.	
Active	false	
Business Key Mutable	true	
Business Model	false	
Visibility	public	
Abstract	false	
Leaf	false	
Root	false	
Indirectly Instantiated	true	
Project Management	Name Value	
	Author	José Antonio Montero Hernández
	Create Date Time 12-abr-2015 22:46:34	
	Last Modified 12-abr-2015 23:00:03	

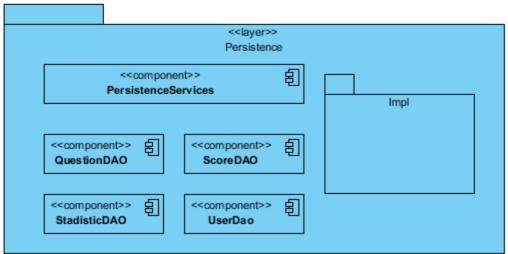
UserDao

Name	Value	
Documentation	Componente que presta servicios de acceso a los datos de usuarios.	
Active	false	
Business Key Mutable	true	
Business Model	false	
Visibility	public	
Abstract	false	
Leaf	false	
Root	false	
Indirectly Instantiated	true	
Project Management	Name Value	
	Author José Antonio Montero Hernández	
	Create Date Time 12-abr-2015 22:46:45	
	Last Modified 12-abr-2015 23:00:03	

Package Diagram

Paquetes





Name	Value
Name	Paquetes

Author	José Antonio Montero Hernández
Create Date Time	10-abr-2015 19:09:15
Last Modified	12-abr-2015 23:10:30
Shape Presentation Option	0

Summary

Nama	Desumentation
Name	Documentation
□ UGI	User Graphic Interface. Paquete que contiene la interface de usuario, toda la capa de vista.
Model	Paquete que contiene las clases que representan el modelo en que se basa la aplicación.
Factories	Paquete que contiene las factorías utilizadas para acceder a las fachadas de las diferentes layers de la aplicación.
Conf	Paquete que contiene clases de utilidad.
Business	Paquete que contiene la lógica del juego, todo lo correspondiente a la capa de negocio.
a GameAPI	API utilizada por la Interface Gráfica para comunicarse con la lógica de la aplicación.
GameClasses	Paquete que contiene las clases necesarias para el funcionamiento y mantenimiento de la partidas.
Exceptions	Paquete que contiene las excepciones personalizadas que se puedan lanzar desde la capa de negocio.
3 Game	Gestiona el funcionamiento de la partida y coordina a los componentes implicados en su desarrollo.
Board	Mantiene la representación lógica del tablero durante la partida y provee de información y cálculos sobre su estado.
FileLoaders	Paquete que contiene las clases que se utilizan para cargar datos desde ficheros.
Dice	Provee de tiradas aleatorias a la vez que permite mantener un bloque sobre los lanzamientos y sobre el último resultado.
BoardOptionsFa ctory	Genera las diferentes topologías de tablero disponibles en el sistema.
impl impl	Paquete que contiene la implementación de las clases que se manejan de forma genérica a nivel de la capa de business.
Persistence	Capa que contiene la lógica de base de datos, toda la capa de persistencia
8	Sublayer de servicio utilizada por la capa de

PersistenceServ ices	persistencia para ofrecer sus servicios a la capa de negocio.
impl impl	Paquete que contiene la implementación de las clases que se manejan de forma genérica a nivel de la capa de persistencia.
QuestionDAO	Componente que presta servicios de acceso a los datos de preguntas.
ScoreDAO	Componente que presta servicios de acceso a los datos de puntuaciones.
\$\frac{1}{2}\$ StadisticDAO	Componente que presta servicios de acceso a los datos de estadísticas.
UserDao	Componente que presta servicios de acceso a los datos de usuarios.

Details



Name	Value	
Documentation	User Graphic Interface. Paquete que contiene la interface de usuario, toda la capa de vista.	
Abstract	false	
Leaf	false	
Root	false	
Stereotypes	layer	
Visibility	public	
Project Management	Name	Value
	Author	José Antonio Montero Hernández
	Create Date Time	12-abr-2015 21:22:14
	Last Modified	12-abr-2015 22:19:13

Model

Name	Value
Documentation	Paquete que contiene las clases que representan el modelo en que se basa la aplicación.
Abstract	false
Leaf	false
Root	false

Stereotypes	transversal	
Visibility	public	
Project Management	Name	Value
	Author	José Antonio Montero Hernández
	Create Date Time	12-abr-2015 21:17:28
	Last Modified	12-abr-2015 23:07:28

Factories

Name	Value		
Documentation	para acceder a las	Paquete que contiene las factorías utilizadas para acceder a las fachadas de las diferentes layers de la aplicación.	
Abstract	false		
Leaf	false		
Root	false		
Stereotypes	transversal		
Visibility	public	public	
Project Management	Name	Value	
	Author	José Antonio Montero Hernández	
	Create Date Time	12-abr-2015 21:18:30	
	Last Modified	12-abr-2015 23:07:28	



Conf

Name	Value	
Documentation	Paquete que contiene cl	ases de utilidad.
Abstract	false	
Leaf	false	
Root	false	
Stereotypes	transversal	
Visibility	public	
Project Management	Name	Value
	Author	José Antonio Montero Hernández
	Create Date Time	12-abr-2015 21:20:50

Look Modified 42 abs 2045 22:07:20		
	Last Modified	12-abr-2015 23:07:28

Business

Name	Value	
Documentation	Paquete que contiene la todo lo correspondiente	
Abstract	false	
Leaf	false	
Root	false	
Stereotypes	layer	
Visibility	public	
Project Management	Name	Value
	Author	José Antonio Montero Hernández
	Create Date Time	10-abr-2015 19:09:45
	Last Modified	12-abr-2015 22:29:58

Children

Name	Documentation
■ GameAPI	API utilizada por la Interface Gráfica para comunicarse con la lógica de la aplicación.
GameClasses	Paquete que contiene las clases necesarias para el funcionamiento y mantenimiento de la partidas.
Exceptions	Paquete que contiene las excepciones personalizadas que se puedan lanzar desde la capa de negocio.
FileLoaders	Paquete que contiene las clases que se utilizan para cargar datos desde ficheros.
impl impl	Paquete que contiene la implementación de las clases que se manejan de forma genérica a nivel de la capa de business.



GameAPI

Name	Value
Documentation	API utilizada por la Interface Gráfica para comunicarse con la lógica de la aplicación.

Active	false	
Business Key Mutable	true	
Business Model	false	
Visibility	public	
Abstract	false	
Leaf	false	
Root	false	
Indirectly Instantiated	true	
Project Management	Name Value	
	Author	José Antonio Montero Hernández
	Create Date Time	12-abr-2015 21:31:36
	Last Modified	12-abr-2015 22:29:58

GameClasses

Name	Value		
Documentation	Paquete que contiene las clases necesarias para el funcionamiento y mantenimiento de la partidas.		
Abstract	false	false	
Leaf	false		
Root	false		
Visibility	public		
Project Management	Name Value		
	Author	José Antonio Montero Hernández	
	Create Date Time	12-abr-2015 21:33:41	
	Last Modified 12-abr-2015 23:00:03		

Children

Name	Documentation
3 Game	Gestiona el funcionamiento de la partida y coordina a los componentes implicados en su desarrollo.
3 Board	Mantiene la representación lógica del tablero durante la partida y provee de información y cálculos sobre su estado.
a Dice	Provee de tiradas aleatorias a la vez que

	permite mantener un bloque sobre los lanzamientos y sobre el último resultado.
BoardOptionsFactory	Genera las diferentes topologías de tablero disponibles en el sistema.

Exceptions

Name	Value		
Documentation	Paquete que contiene las excepciones personalizadas que se puedan lanzar desde la capa de negocio.		
Abstract	false	false	
Leaf	false		
Root	false		
Visibility	public		
Project Management	Name Value		
	Author	José Antonio Montero Hernández	
	Create Date Time	12-abr-2015 21:32:30	
	Last Modified	12-abr-2015 21:58:10	



Name	Value	
Documentation	Gestiona el funcionamiento de la partida y coordina a los componentes implicados en su desarrollo.	
Active	false	
Business Key Mutable	true	
Business Model	false	
Visibility	public	
Abstract	false	
Leaf	false	
Root	false	
Indirectly Instantiated	true	
Project Management	Name Value	
	Author	José Antonio Montero Hernández

Create Date Time	12-abr-2015 22:49:37
Last Modified	12-abr-2015 23:07:28

Board

Name	Value		
Documentation	Mantiene la representación lógica del tablero durante la partida y provee de información y cálculos sobre su estado.		
Active	false		
Business Key Mutable	true		
Business Model	false	false	
Visibility	public		
Abstract	false		
Leaf	false		
Root	false		
Indirectly Instantiated	true		
Project Management	Name Value		
	Author	José Antonio Montero Hernández	
	Create Date Time	12-abr-2015 22:50:01	
	Last Modified	12-abr-2015 23:07:29	

FileLoaders

Name	Value		
Documentation	Paquete que contiene las clases que se utilizan para cargar datos desde ficheros.		
Abstract	false		
Leaf	false	false	
Root	false		
Visibility	public		
Project Management	Name Value		
	Author	José Antonio Montero Hernández	
	Create Date Time	12-abr-2015 21:33:18	
	Last Modified	12-abr-2015 21:58:11	



Name	Value		
Documentation	Provee de tiradas aleatorias a la vez que permite mantener un bloque sobre los lanzamientos y sobre el último resultado.		
Active	false		
Business Key Mutable	true		
Business Model	false	false	
Visibility	public	public	
Abstract	false		
Leaf	false		
Root	false		
Indirectly Instantiated	true		
Project Management	Name	Value	
	Author	José Antonio Montero Hernández	
	Create Date Time	12-abr-2015 22:52:01	
	Last Modified	12-abr-2015 23:07:29	

BoardOptionsFactory

Name	Value		
Documentation	Genera las diferentes topologías de tablero disponibles en el sistema.		
Active	false		
Business Key Mutable	true		
Business Model	false		
Visibility	public	public	
Abstract	false		
Leaf	false		
Root	false		
Indirectly Instantiated	true		
Project Management	Name	Value	
	Author	José Antonio Montero Hernández	
	Create Date Time 12-abr-2015 22:52:36		
	Last Modified 12-abr-2015 23:07:29		

impl

Name	Value	
Documentation	Paquete que contiene la implementación de las clases que se manejan de forma genérica a nivel de la capa de business.	
Abstract	false	
Leaf	false	
Root	false	
Visibility	public	
Project Management	Name Value	
	Author	José Antonio Montero Hernández
	Create Date Time	12-abr-2015 21:34:25
	Last Modified	12-abr-2015 21:47:47

Pers:

Persistence

Name	Value		
Documentation	Capa que contiene la lógica de base de datos, toda la capa de persistencia		
Abstract	false	false	
Leaf	false		
Root	false		
Stereotypes	layer		
Visibility	public		
Project Management	Name Value		
	Author José Antonio Montero Hernández		
	Create Date Time 12-abr-2015 21:21:04		
	Last Modified 12-abr-2015 23:00:03		

Children

Name	Documentation
PersistenceServices	Sublayer de servicio utilizada por la capa de persistencia para ofrecer sus servicios a la capa de negocio.
impl Impl	Paquete que contiene la implementación de las clases que se manejan de forma genérica a nivel de la capa de persistencia.

QuestionDAO	Componente que presta servicios de acceso a los datos de preguntas.
ScoreDAO	Componente que presta servicios de acceso a los datos de puntuaciones.
StadisticDAO	Componente que presta servicios de acceso a los datos de estadísticas.
1 UserDao	Componente que presta servicios de acceso a los datos de usuarios.

PersistenceServices

Name	Value	
Documentation	Sublayer de servicio utilizada por la capa de persistencia para ofrecer sus servicios a la capa de negocio.	
Active	false	
Business Key Mutable	true	
Business Model	false	
Visibility	public	
Abstract	false	
Leaf	false	
Root	false	
Indirectly Instantiated	true	
Project Management	Name Value	
	Author José Antonio Montero Hernández	
	Create Date Time 12-abr-2015 22:07:00	
	Last Modified 12-abr-2015 22:29:58	



Name	Value
Documentation	Paquete que contiene la implementación de las clases que se manejan de forma genérica a nivel de la capa de persistencia.
Abstract	false
Leaf	false
Root	false
Visibility	public

Project Management	Name	Value
	Author	José Antonio Montero Hernández
	Create Date Time	12-abr-2015 21:35:40
	Last Modified	12-abr-2015 21:47:48

QuestionDAO

= Questionidad			
Name	Value		
Documentation	Componente que presta servicios de acceso a los datos de preguntas.		
Active	false	false	
Business Key Mutable	true		
Business Model	false		
Visibility	public	public	
Abstract	false		
Leaf	false		
Root	false		
Indirectly Instantiated	true		
Project Management	Name Value		
	Author José Antonio Montero Hernández		
	Create Date Time 12-abr-2015 22:46:22		
	Last Modified 12-abr-2015 23:00:03		



ScoreDA0

Name	Value
Documentation	Componente que presta servicios de acceso a los datos de puntuaciones.
Active	false
Business Key Mutable	true
Business Model	false
Visibility	public
Abstract	false
Leaf	false
Root	false
Indirectly Instantiated	true

Project Management	Name	Value
	Author	José Antonio Montero Hernández
	Create Date Time	12-abr-2015 22:45:11
	Last Modified	12-abr-2015 23:00:03

StadisticDAO

Name	Value		
Documentation	Componente que presta servicios de acceso a los datos de estadísticas.		
Active	false		
Business Key Mutable	true		
Business Model	false		
Visibility	public	public	
Abstract	false		
Leaf	false		
Root	false		
Indirectly Instantiated	true		
Project Management	Name Value		
	Author José Antonio Montero Hernández		
	Create Date Time 12-abr-2015 22:46:34		
	Last Modified 12-abr-2015 23:00:03		



UserDao

Name	Value
Documentation	Componente que presta servicios de acceso a los datos de usuarios.
Active	false
Business Key Mutable	true
Business Model	false
Visibility	public
Abstract	false
Leaf	false
Root	false
Indirectly Instantiated	true

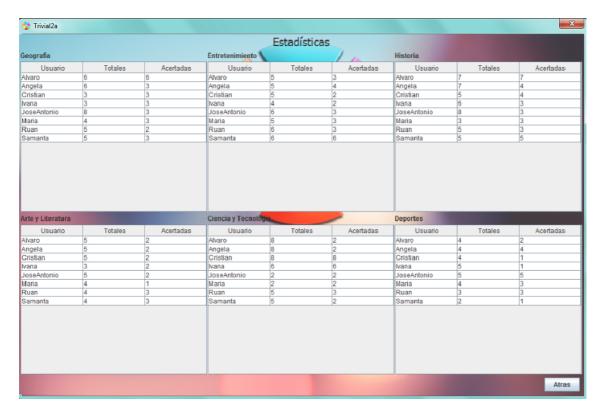
Project Management	Name	Value
	Author	José Antonio Montero Hernández
	Create Date Time	12-abr-2015 22:46:45
	Last Modified	12-abr-2015 23:00:03

12. Manual de usuario

En primer lugar se muestra la pantalla de inicio en el que el usuario puede escoger iniciar una nueva partida, escoger entre diversas opciones en relación al diseño del tablero, consultar estadísticas en base a los aciertos y fallos de las preguntas o salir.



Si el jugador opta por visualizar las estadísticas podrá ver las respuestas acertadas y totales de cada categoría para cada usuario.



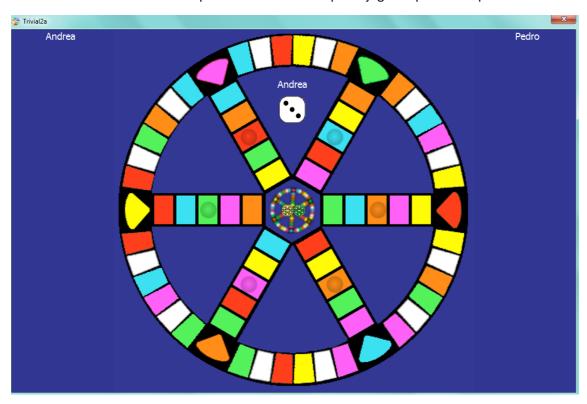
El escoger la opción de jugar se muestra una pantalla en la que deben introducirse los jugadores de la partida.



Si en la ventana anterior el usuario pulsa sobre jugar se presenta el tablero en la parte central, con el nombre de los jugadores a los lados. Puede observarse un dado, pulsar sobre el equivaldría a tirar el dado, por lo que el valor que se muestra en pantalla es el número de casillas que puede desplazarse el jugador, sobre éste aparece el nombre del jugador del turno actual.



Al tirar el dado se muestran las posibles casillas a las que el jugador podría desplazarse.



Al pulsar sobre una de las mismas se muestra la pregunta que dicho jugador debe contestar.



Si escoge la respuesta correcta s ele mostrará un mensaje de acierto y podrá volver a tirar el dado, en caso contrario será el turno del siguiente jugador.

Si la casilla en la que se encontraba el jugador era una casilla de pieza, al acertar la pregunta obtendrá el triángulo del color correspondiente.

El jugador que se posicione en la casilla central habiendo obtenido un triángulo de cada color, será el ganador.