

DOCUMENTACIÓN

TRIVIAL 3A

Versión 3.1



- *Marcos Abdel-Fattah Martínez* - (UO230956)
- *Carmen Peñalver San Cristóbal* - (UO229705)
- *Pablo Bravo Mediavilla* - (UO223531)
- *Álvaro Panizo Romano* - (UO227748)
- *Javier Iglesias García* - (UO232562)
- *Raúl Herrador Colino* - (UO234551)
- *Liliana Villar Iglesias* - (UO232510)
- *Iñigo Llana Allier* - (UO206367)

Grupo Trivial 3a

UNIVERSIDAD DE OVIEDO - Arquitectura del Software

Contenido

MANUAL – JUEGO TRIVIAL	2
Manual de instalación	2
Manual de usuario	2
Usuario cliente	2
Usuario administrador	8
Manual de sistema	10
Paquete business	10
<i>Paquete core</i>	10
<i>Paquete game</i>	10
<i>Paquete Tablero</i>	11
Paquete modelo	13
<i>Clase Pregunta</i>	13
<i>Clase Usuario</i>	13
Paquete persistencia.....	13
Paquete controllers	13
<i>Clase Application</i>	13
Paquete views	14
<i>Game.scala.html</i>	14
<i>Index.scala.html</i>	14
<i>Menu_admin.scala.html</i>	14
DOCUMENTACIÓN – JUEGO TRIVIAL	15
Ámbito y alcance del problema	15
Planteamiento del problema	15
Metodología usada	16
Identificación de los interesados (Stakeholders)	16
Lista de interesados (Stakeholders)	17
Lista de atributos de calidad	18
Atributos de calidad e interesados	20
Escenarios de calidad.	20
Descripción de negocio de la solución	22
<i>Tecnologías utilizadas.</i>	23
<i>Vistas</i>	26
DOCUMENTACIÓN CASE	26
BUSINESS MODELING (BPM)	73

MANUAL – JUEGO TRIVIAL

Manual de instalación

En este caso el manual de instalación tratará de un breve tutorial que orientará al usuario desarrollador u observador consentido sobre como probar la reléase en fase beta del juego del trivial.

Para empezar comenzaremos descargando el proyecto desde git:

<https://github.com/Arquisoft/Trivial3a/tree/DesignPlay>

Tras descargar el proyecto nos situamos en una línea de comando sobre la carpeta del proyecto y escribimos el comando **activator**, que nos cargara el proyecto gracias a dicha instrucción añadida en la última versión de play.

Si quisiéramos ver el código simplemente con la instrucción **eclipse** ya se nos cargaría el proyecto con todo el código a nuestra disposición.

Si quisiéramos probar la funcionalidad a nivel local bastaría con utilizar la instrucción **run** y acceder desde nuestro navegador a **localhost:9000**, puerto que escucha el servidor de play.

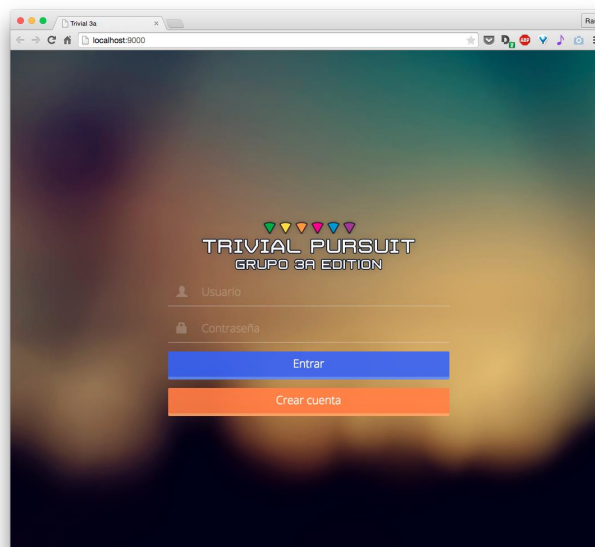
A un nivel más alto de abstracción, si el proyecto se utilizara dentro de un servidor podríamos utilizar la instrucción **start** para arrancarlo pero desde nuestro proyecto nos ha dado algunos problemas con Windows aunque para IOS funciona.

Manual de usuario

Descripción del funcionamiento de la interfaz que será utilizada por un usuario administrador o un usuario cliente, éste usuario podrá registrarse en la aplicación, validarse para poder participar en una partida de trivial y consultar sus estadísticas de las partidas ganadas, preguntas acertadas, etc...

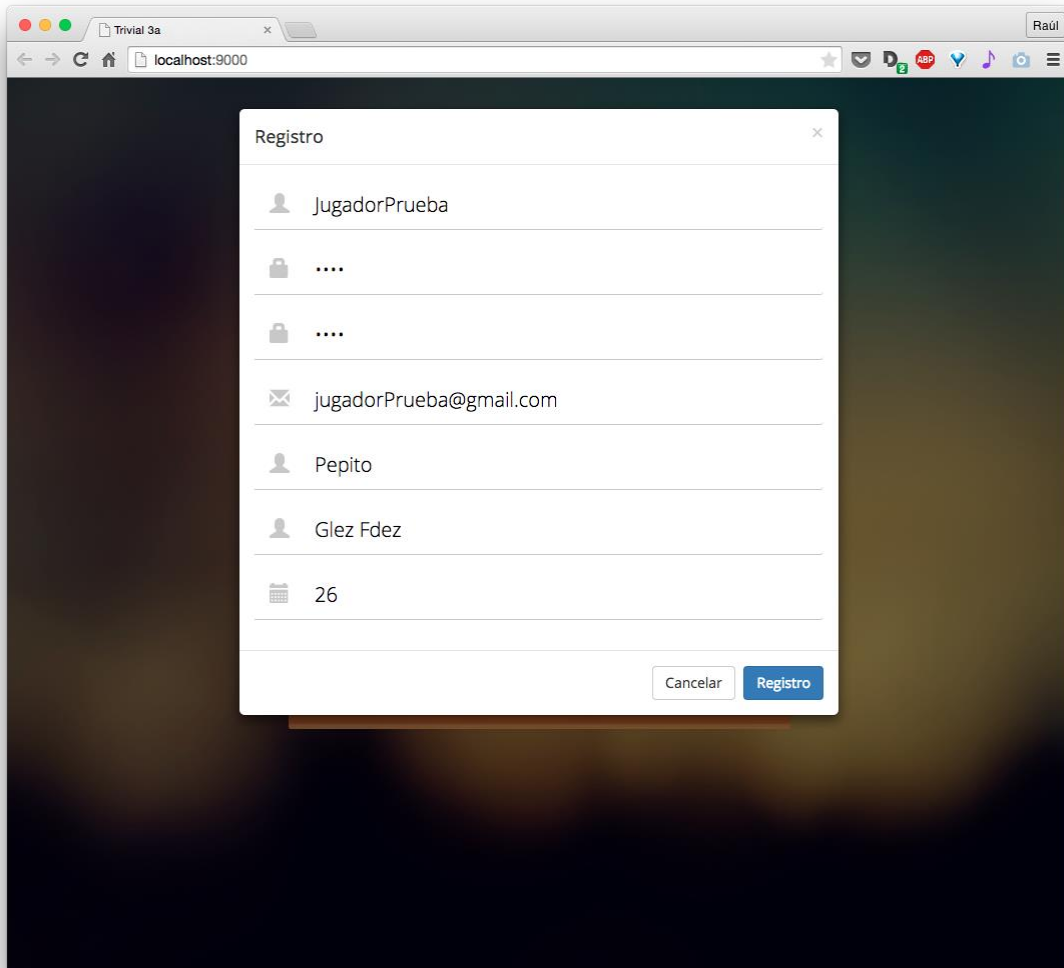
Usuario cliente

Al cargar la aplicación nos encontraremos con la siguiente ventana de inicio:



En ella, tenemos dos alternativas.

- Crear cuenta. Al pulsar sobre este botón se nos abrirá una nueva ventana en la que podremos ver un formulario de registro:



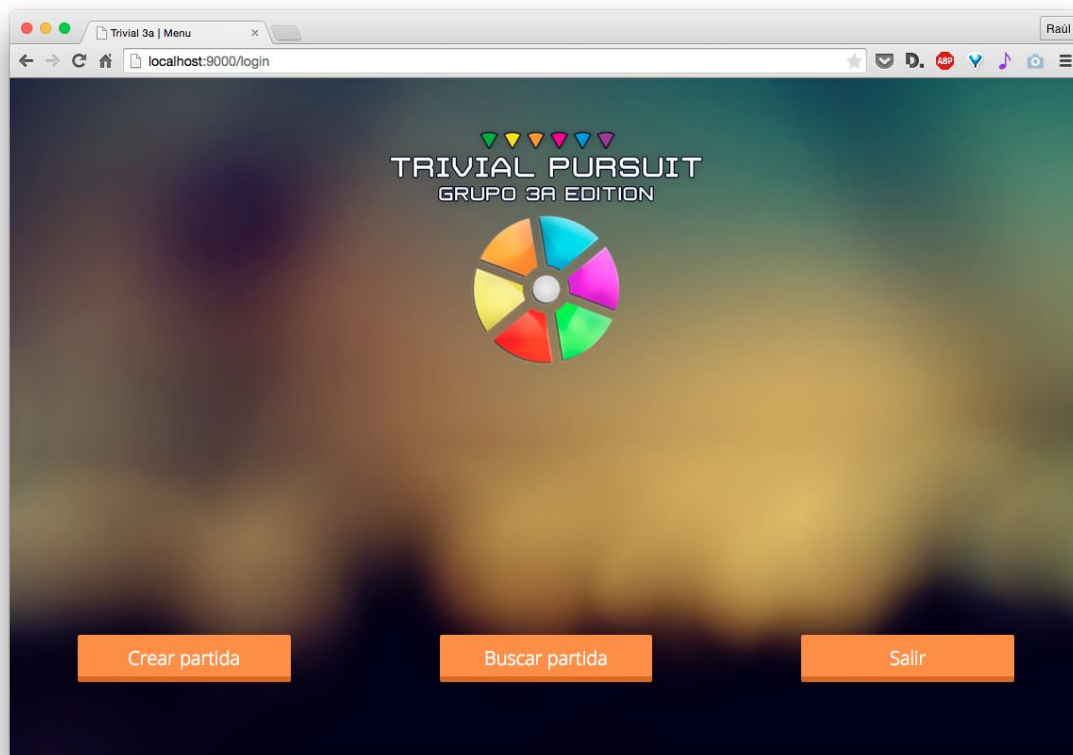
The screenshot shows a web browser window with the address bar set to localhost:9000. A modal window titled 'Registro' is open, containing a registration form with the following fields and values:

Icono	Valor
Persona	JugadorPrueba
Candado
Candado
Correo	jugadorPrueba@gmail.com
Persona	Pepito
Persona	Glez Fdez
Calendario	26

At the bottom of the form are two buttons: 'Cancelar' and 'Registro'.

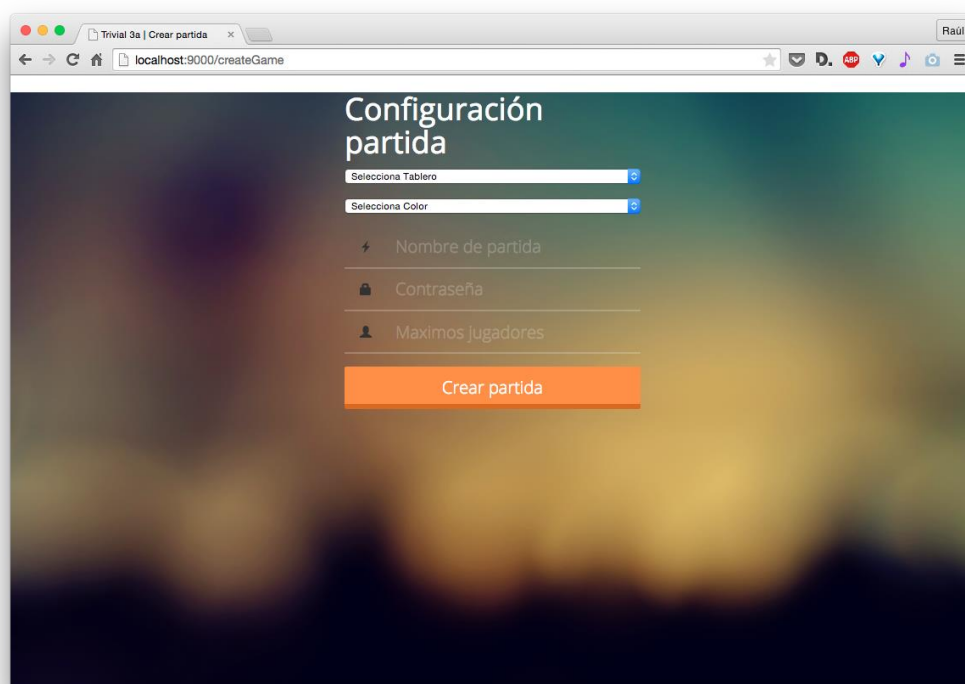
En este formulario se nos pedirá que introduzcamos datos del usuario: nombre de usuario, contraseña (por duplicado), dirección de correo electrónico, nombre, apellidos y edad. Pinchamos sobre el botón “registro” y si los datos introducidos no se corresponden con los de ningún otro jugador ya registrado, se creará nuestra cuenta y podremos empezar a jugar.

- Entrar. Para poder entrar a la aplicación es necesario introducir nuestro nombre de usuario y contraseña correctamente. Cuando los hayamos introducido se nos abrirá una nueva ventana:



En esta nueva ventana, tendremos tres opciones:

- Crear partida: al pulsar el botón para crear la partida se nos abrirá una nueva ventana en la que podremos configurar las opciones para jugar:
- Buscar partida: para buscar una partida que otro usuario tenga iniciada y unirse a ella.
- Salir: cerrará la aplicación.

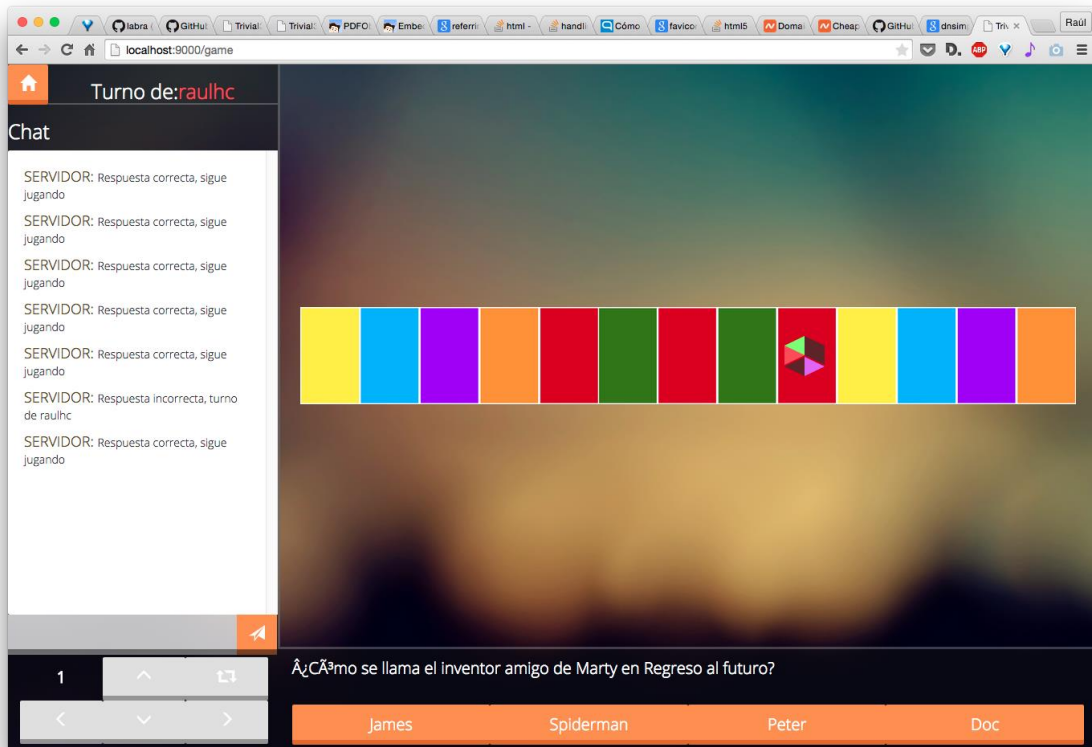
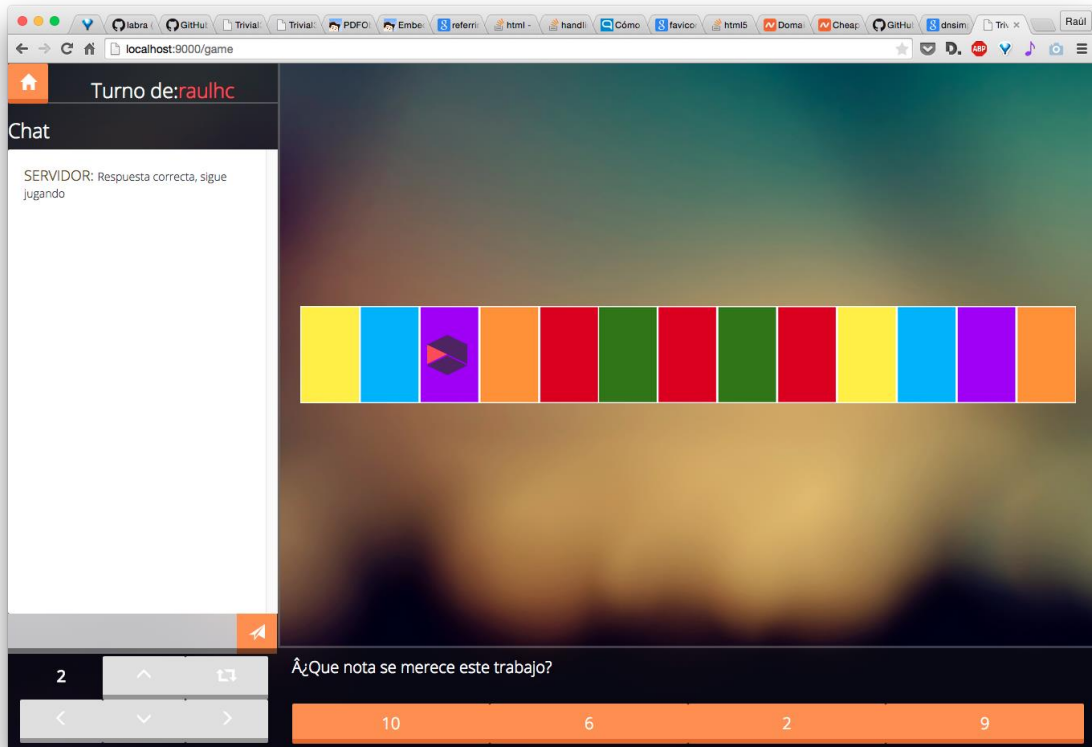


Lo primero que debemos elegir es el tablero con el que queremos jugar, con forma circular, forma recta... En segundo lugar seleccionaremos el color que queremos ser (rojo, amarillo, verde...), a continuación introduciremos una serie de datos, el nombre de la partida, una contraseña y el máximo de jugadores que queremos permitir.

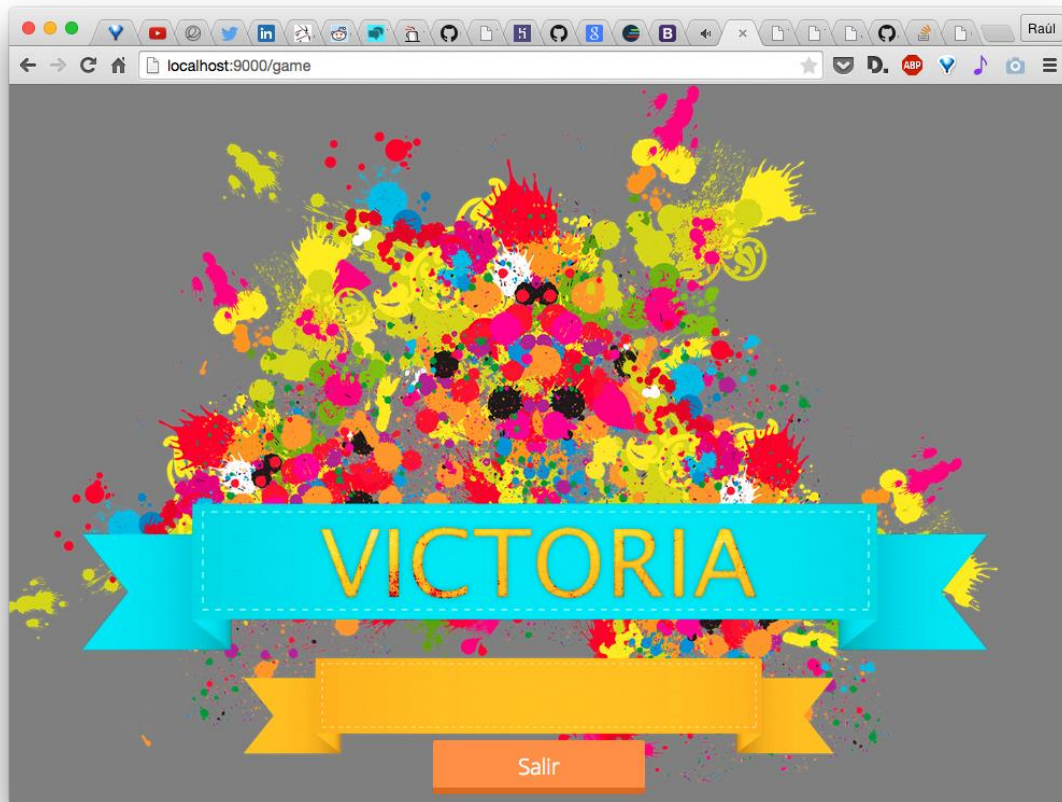


Al presionar el botón de crear partida tendremos acceso a la ventana con el tablero.

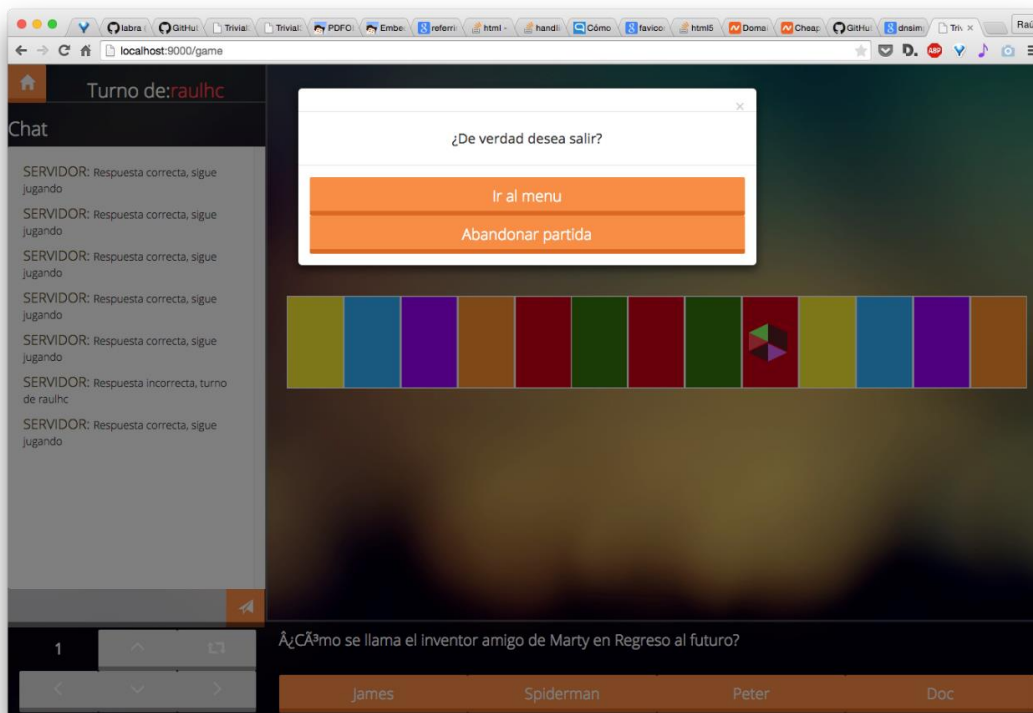
En esta ventana, tiraremos el dado pulsando sobre el botón con el botón con una figura de un cuadrado con flechas, nos aparecerá a su izquierda el resultado, elegiremos donde queremos movernos con las flechas y nos aparecerá una pregunta con cuatro posibles respuestas. Si acertamos se nos añadirá un quesito, si no, jugará otro jugador en caso de partidas multijugador.



Cuando tengamos los quesitos de todos los colores, finalizará el juego indicando que hemos ganado.



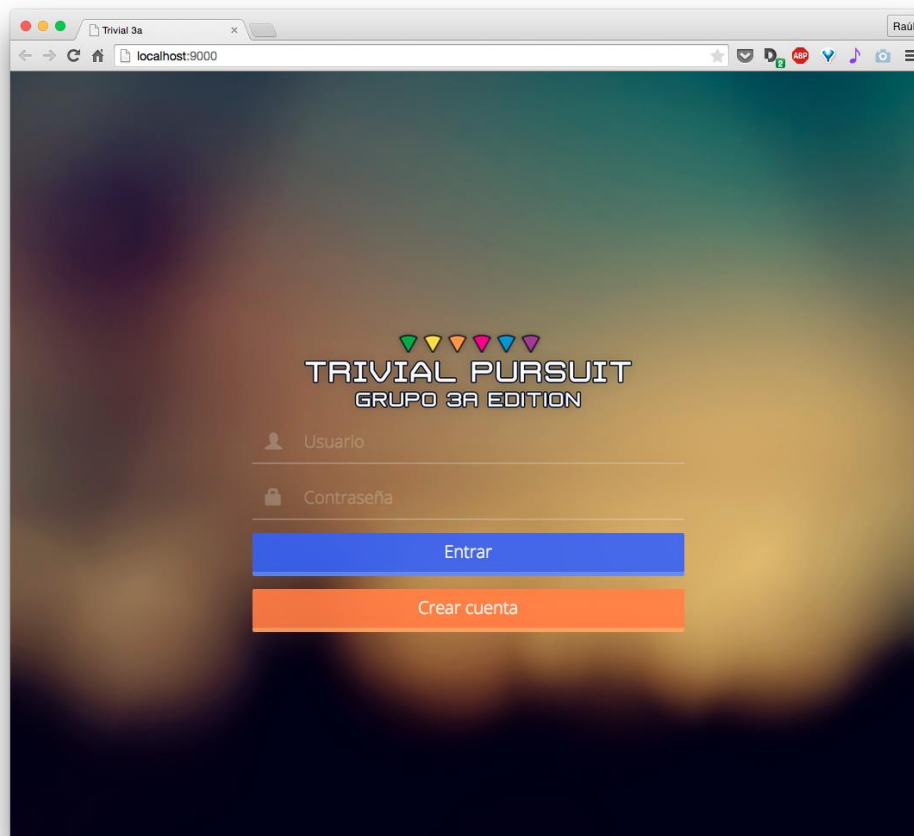
Si queremos salir de la partida, pincharemos sobre el botón con una casita y nos aparecerán dos opciones:



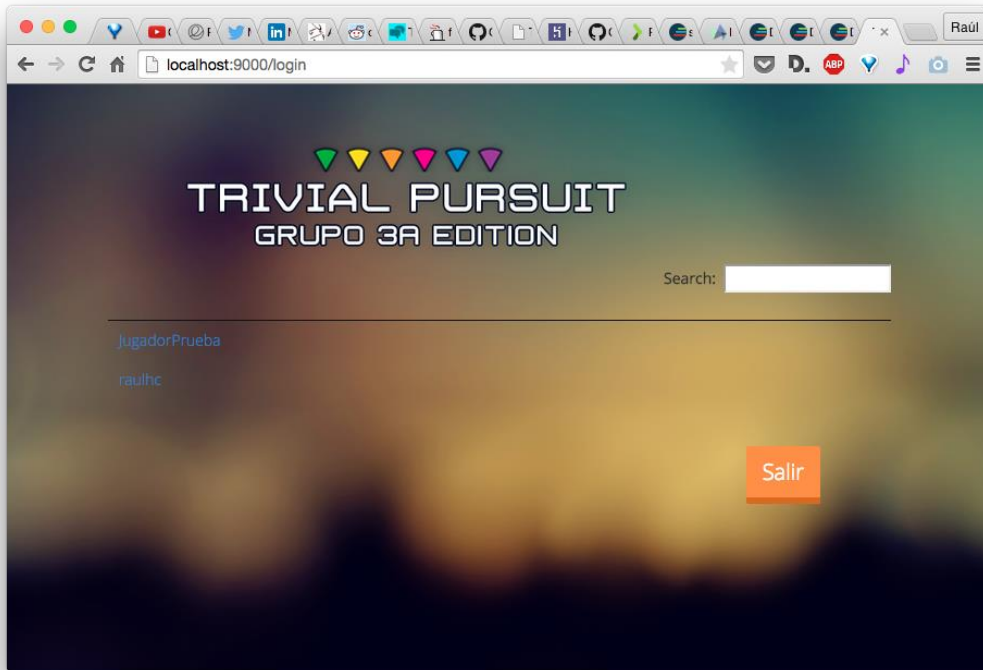
Podemos volver al menú para iniciar otra partida, o finalizar el juego.

Usuario administrador

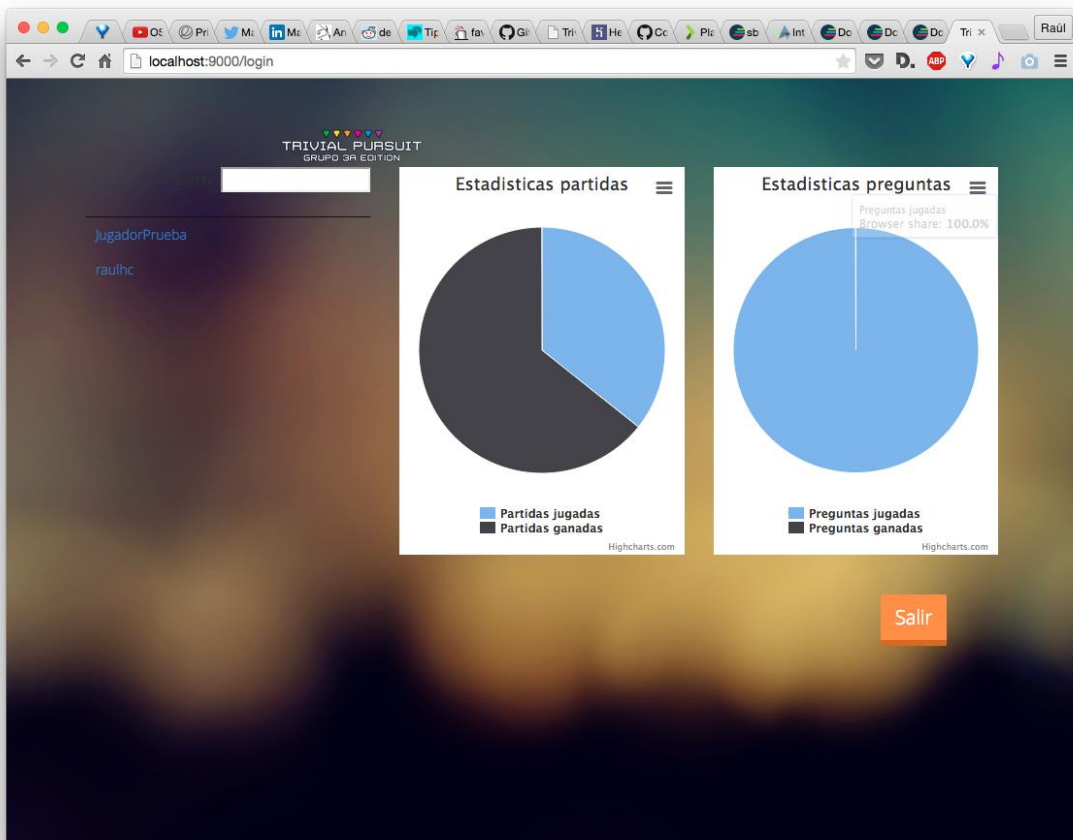
El usuario administrador, deberá iniciar sesión en la ventana principal, para ello, introducirá su nombre de usuario y contraseña.

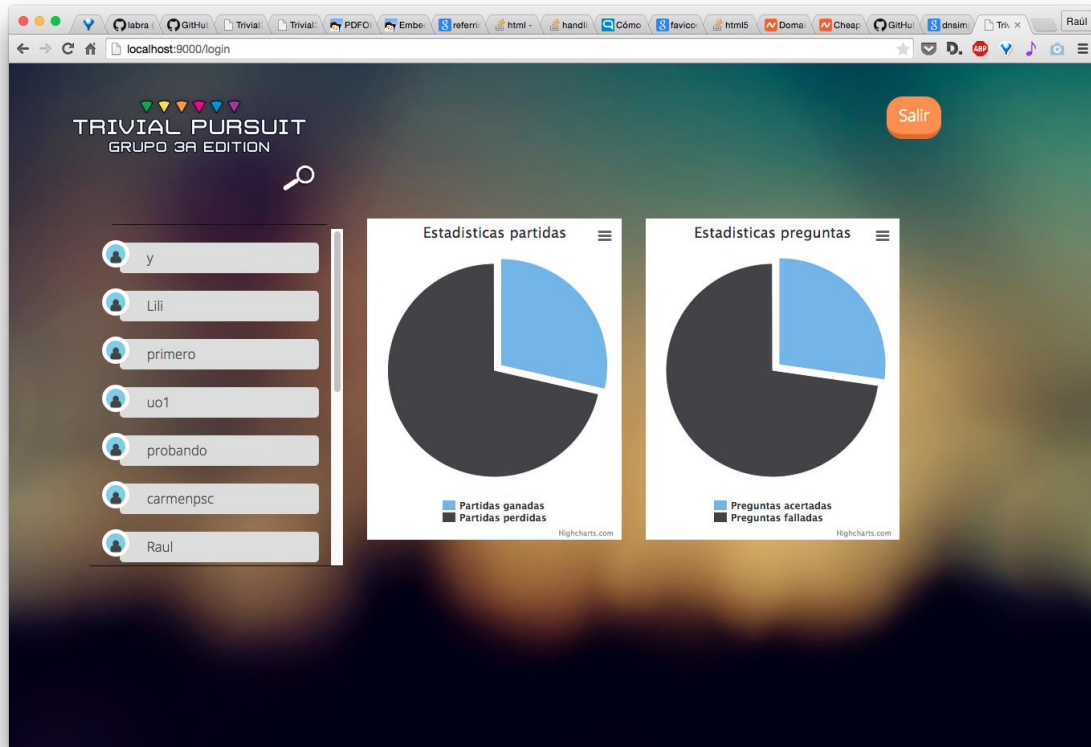


Una vez introducidos los datos, se nos abrirá una nueva ventana:



En esta ventana podemos ver todos los usuarios de la aplicación o buscarlos por sus nombres de usuario. Además también tenemos la opción de ver sus estadísticas:





En esta ventana podemos ver todas las estadísticas de los jugadores, partidas jugadas, ganadas, preguntas contestadas, preguntadas acertadas... Además podremos buscar un jugador en concreto filtrando su búsqueda por nombre, por login, por apellidos, etc.

Manual de sistema

El sistema está implementado con un conjunto de paquetes que se irán explicando a continuación. Se utiliza el patrón factoría para comunicar las capas de la lógica de negocio y persistencia con la capa de presentación.

Paquete business

En este paquete se almacena toda la lógica del juego, está subdividido en otros tres paquetes.

Paquete core

Este paquete contiene la clase Core que es la encargada crear y ejecutar la ventana principal de la aplicación en la que el usuario podrá realizar el registro y comenzar el juego.

Paquete game

En este paquete se encuentran las clases encargadas de iniciar el juego.

Clase IniciarJuegoAction

Esta clase tiene una lista en la que se guardan los usuarios que van a participar en la partida junto con el color. Métodos:

- IniciarJuego: almacena en la lista anterior los usuarios que van a participar.
- Execute: añade la lista de usuarios a la base de datos.

Clase RegistrarseAction

Métodos:

- RegistrarseAction: crea un nuevo Usuario con los datos pasados como parámetros (Login, contraseña, nombre, apellidos, email y edad).
- ExistUserNameEmail: comprueba que en la base de datos no esté ningún usuario con el mismo nombre de usuario, email o nombre.
- Execute: añade a la base de datos el nuevo usuario.

Clase ValidarseAction

Métodos:

- GetUsuario: este método obtiene el usuario que se está validando a través de los datos que se le pasan: login y contraseña.
- UsuarioDuplicado: se comprueba que un jugador no se pueda validar dos veces en el juego para que no se repita ninguno en el momento de iniciar la partida.
- IsCorrecto: muestra mensajes de error si el usuario ya se ha validado, si el color ya se ha elegido o si la contraseña o login son incorrectos.

Clase Action

Es la encargada de realizar la conexión con la base de datos y obtener los datos almacenados.

Clase Partida

Esta clase tiene los métodos encargados de añadir usuarios a una lista y de eliminarlos.

Paquete Tablero

Contiene los diferentes paquetes:

Paquete colores

Este paquete tiene la interface color que puede ser amarillo, azul, incoloro, morado, naranja, rojo o verde.

Paquete jugadores

Este paquete contiene otro patrón factoría que se utiliza para crear nuevos jugadores.

Clase jugador

Tiene los atributos:

- Quesitos: lista de quesitos donde se añadirán los colores de los quesitos que se obtengan.
- Actual: nodo con la posición del tablero en la que se encuentra.
- Tablero: atributo de la clase Tablero.

Métodos:

- MoverDerecha/MoverIzquierda: se le pasa el número de posiciones que se quiere mover y se actualiza la posición del atributo "actual", al nuevo nodo.
- AñadirQuesito: añade un color a la lista de quesitos.

- `ComprobarQuesitos`: comprueba si la lista `quesitos` contiene todos los colores y devuelve `true` o `false`.
- `RemoveQuesito`: quita un `quesito` de la lista.

Paquete nodos

Este paquete tiene una factoría para generar nodos, y las siguientes clases:

Clase `NodoFinal`

Será el último nodo del tablero.

Clase `NodoQuesito`

Serán los nodos en los que se puedan conseguir `quesitos`.

Clase `NodoNormal`

Serán el resto de nodos del tablero que tendrán un color y un nodo anterior y otro siguiente.

Paquete `tableros`

Éste paquete genera clases `tableros` que pueden ser circulares, lineales o de tipo rueda. Estos `tableros` contienen un nodo raíz que será el primero, un nodo pivote y un contenedor donde se encuentran los datos de todos los nodos que se van a enlazar.

Clase `TableroCircular`

El constructor se encarga de enlazar los nodos de los extremos para que el tablero esté en forma de círculo.

Clase `TableroLineal`

No se enlazan los nodos extremos ya que se ponen a `null`, de esta manera solo se podrá ir y volver en la misma línea.

Clase `TableroRueda`

El constructor se encarga de enlazar los nodos de los extremos para que el tablero esté en forma de círculo y además, los nodos con `quesito` se enlazan con nodos hacia arriba para que el círculo tenga sus radios.

Paquete `Util`

Este paquete tiene dos clases encargadas de cargar los nodos de los diferentes tipos de `tableros`.

Clase `LectorCircular`

Crea un contenedor de nodos y lo llena de `quesitos`.

Clase `LectorFormaRueda`

Carga un fichero que se le pasa como parámetro y va introduciendo cada nodo en el contenedor de nodos.

Clase `ContenedorDeNodos`

Clase que guarda en listas los nodos exteriores del tablero, los interiores y los nodos `quesitos`.

Clase `PreguntasAleatorias`

Crea una instancia de sí misma y tiene el método `dado` que obtiene un número aleatorio de 1 a 6.

Clase SelectorDeColores

Permite obtener uno de los colores del trivial pasándole un String con su nombre.

Paquete modelo

Este paquete contiene las clases que se añadirán a la base de datos.

Clase Pregunta

Tiene los atributos id, pregunta, respuestas incorrectas y respuesta correcta.

Clase Usuario

Tiene los atributos login, contraseña, nombre, apellidos, email, edad, número de jugadas y número partidas ganadas.

Paquete persistencia

Este paquete se encarga del acceso a la base de datos y de realizar las consultas para insertar datos. Esto se realiza mediante las clases:

Clase Connection

Conecta con la base de datos.

Clase Insert

Inserta elementos en la base de datos.

Clase Estadísticas

Esta clase permite crear jugadores con sus estadísticas de juego iniciadas a 0, guardará las partidas jugadas, las ganadas, y el número total. Tiene una lista con todas las estadísticas de todos los jugadores y un método que permite actualizarlas.

Clase GetPreguntas

Esta clase tiene un método que devuelve una pregunta de la base de datos de una categoría pasada como parámetro.

Clase GetUsuario

Esta clase tiene un método que devuelve un usuario de la base de datos. Lo busca por su nombre de usuario que se le pasa como parámetro.

Clase GetUsuarios

Esta clase devuelve una lista con todos los usuarios de la base de datos.

Paquete controllers

Clase Application

Esta clase controla casi toda la lógica del juego. Tiene diferentes métodos:

- **NewRegister:** Permite registrarse a un nuevo usuario obteniendo los datos del formulario de registro.

- NewLogin: Obtiene el nombre de usuario y la contraseña introducidas por el usuario en la ventana de login y comprueba que existen. Si todo es correcto nos envía a una nueva ventana, si no, muestra un mensaje de error.
- Logout: Para borrar los datos guardados en la sesión.
- Game: Crea todos los elementos necesarios para jugar una partida: los jugadores, el tablero, las estadísticas y lo guarda todo en sesión.
- GetPosicion: Obtiene la posición actual del jugador y actualiza sus datos.
- Move: A partir del parámetro “dirección” que se le pasa, mueve al jugador hacia uno de los lados indicados.
- IsFinished: Comprueba si el jugador es diferente de null para saber si ha cerrado sesión o no.
- GetTokens: Obtiene los quesitos del jugador en sesión.

Paquete views

Este paquete tiene todos los html con los elementos de las partidas que veremos en nuestro navegador al lanzar la aplicación.

Game.scala.html

Este fichero html contiene todos los elementos de la vista del juego cuando hemos empezado una partida. Los botones para tirar dado, movernos, salir, el tablero y el chat con los mensajes informativos.

Index.scala.html

En este fichero tenemos los elementos de la ventana de inicio, los botones para registrarse o iniciar sesión, con los campos para rellenar los datos.

Menu_admin.scala.html

En este fichero están los elementos de la ventana del administrador donde tiene una lista de todos los usuarios y sus estadísticas.

DOCUMENTACIÓN – JUEGO TRIVIAL

Ámbito y alcance del problema

Para esta tercera y última entrega se nos plantea un problema ligeramente diferente y que viene a ser una ampliación del trabajo realizado en las dos primeras entregas. Los requisitos han sido ampliados de manera sustancial con el objetivo de lograr una aplicación *completamente funcional para su uso en la web*.

Las limitaciones de esta tercera entrega vienen a ser que la aplicación funcionara en la web de una manera similar a la que lo hacía en el escritorio, siendo además necesario que los jugadores dispongan de una cuenta para poder buscar una partida existente y unirse a ella o comenzar una partida nueva.

La parte que varía con respecto a la segunda entrega es únicamente la parte de las vistas, teniendo que existir ahora una versión web. Es decir, hemos reutilizado la mayor parte de la lógica de negocio, teniendo que hacer pequeñas adaptaciones puntuales.

En cuanto al alcance del problema nos encontramos con que tenemos que llevar a cabo el desarrollo de una aplicación completamente funcional que **permita jugar partidas de trivial tanto en escritorio como en la web** a diferentes usuarios y con diferentes modos de juego. Además se plantea que han de registrarse una serie de datos de interés acerca de las partidas jugadas por cada usuario con el objetivo de conocer *datos estadísticos* sobre su rendimiento durante las partidas que podrán ser consultados por el administrador.

Planteamiento del problema

En esta ocasión la empresa “NoGame” nos plantea la continuación de las dos primeras partes del juego ya desarrolladas. Esta vez el desarrollo consistirá en la realización de la versión web del juego trivial con el que habíamos estado trabajando.

Dicha aplicación tendrá una interfaz web gráfica y permitirá que los usuarios se registren y loguen en la aplicación. Dado que cada usuario quedara guardado en el sistema también se guardarán datos relativos a las partidas que haya jugado. De esta forma el administrador podrá llevar a cabo un análisis estadístico del rendimiento de los jugadores durante las partidas. En la interfaz web se mostrará un tablero similar al tablero de trivial clásico con el objetivo de que los usuarios puedan conocer en que posiciones se encuentran y sus puntuaciones.

La interfaz gráfica llevará a cabo la representación visual de todos los datos contenidos en la base de datos utilizada por la aplicación, tales como usuarios registrados, preguntas, etc

Metodología usada

El estudio y diseño de esta arquitectura se llevara a cabo mediante la utilización del método Attribute-Driven-Design (**ADD**) y el estándar del **SEI** (ANSI/IEEE 1471, 2000).

El método ADD esta basado en la identificación de los atributos de calidad y la creación de una serie de escenarios que esos atributos deben cumplir.

Identificación de los interesados (Stakeholders)

Este proyecto esta nombrado como “Trivial3a” por lo que los interesados están contextualizados dentro del dicho proyecto.

En este caso los interesados son:

- **Responsables de la empresa:**
 - Los dirigentes que toman las decisiones sobre el proyecto, presupuesto y evolución. Sus tareas son hacer que el sistema sea tan fácil de modificar y escalar como sea posible. Dicho objetivo viene motivado por el intento de conseguir que el coste de desarrollo sea el mínimo posible.
- **Usuario administrador:**
 - El administrador o administradores son los encargados de tratar con la aplicación de una forma más amplia que los usuarios corrientes. Tienen la función de controlar que la aplicación este funcionando bien así como de llevar un control de las estadísticas de juego durante el funcionamiento del mismo.
- **Equipo de desarrollo:**
 - Los programadores tratan de lograr un diseño modular que permita construir la aplicación del modo mas independiente posible para que así los módulos existentes no se vean influidos por el desarrollo de otros módulos.
- **Jugadores / Usuarios de la aplicación:**
 - Los usuarios de la aplicación son aquellas personas interesadas en dar uso al juego que hemos desarrollado. Ahora esas personas incluyen también a las dispuestas a jugar desde la web, además de las que jugaban en local sobre la aplicación de escritorio.
 - Son la clave para que una aplicación triunfe o no, por lo que el diseño ha de ser tan sencillo como sea posible. Además el usuario tiene que encontrar entretenida la aplicación porque si no lo hace dejara de usarla.

Lista de interesados (Stakeholders)

Código	StakeHolder	Intereses
ST-01	Responsables de la empresa	Bajo coste de desarrollo, esto es, el desarrollo del proyecto debe ser corto y con un coste reducido. Eficiencia e independencia de los distintos componentes del juego para poder ampliar funcionalidad en un futuro sin que esto suponga un coste elevado.
ST-02	Usuario administrador	Simplificación de aprendizaje y utilización. Fácil interacción con la interfaz. Rápida detección de posibles problemas y rápida y sencilla solución de ellos
ST-03	Equipo de desarrollo	Proyecto escalable y 'modularizado'. Reparto de tareas y desarrollo eficiente.
ST-04	Jugadores / Usuarios de la aplicación	Facilidad de uso ante todo. Entretenimiento además de buena experiencia de uso.

Lista de atributos de calidad

Código	Descripción	Tipo de atributo
AT001	Posibilidad de cambio del look&feel del juego.	Modificabilidad
AT002	Posibilidad de modificar el tipo de tablero así como el numero de jugadores que participan en una partida.	Modificabilidad
AT003	Garantizar que la aplicación procese los datos correctamente y en un tiempo óptimo	Rendimiento
AT004	Las estadísticas de cada usuario solo pueden ser conocidas por el mismo y por los administradores del juego	Confidencialidad
AT005	Coste de desarrollo bajo. Coste-Beneficio	Coste-Beneficio
AT006	Facilidad de uso de la aplicación	Usabilidad
AT007	Facilidad de visualización de las estadísticas	Usabilidad
AT008	Evitar la inconsistencia de los datos por accesos simultáneos	Seguridad
AT009	Los usuarios no registrados y logeados no podrán acceder a la aplicación	Seguridad
AT010	Las BBDD han de funcionar a tiempo completo sin producirse fallos de lectura/escritura	Disponibilidad

AT011	Tiempo de desarrollo corto	Time to market
AT012	El juego ha de estar dividido en componentes lo suficientemente pequeños como para ser reutilizados en otras aplicaciones de la empresa	Adaptabilidad
AT013	La aplicación debe de ser fácilmente probada con el objetivo de encontrar posibles bugs lo mas rápida y fácilmente posible.	Testeabilidad
AT014	El servidor en el que esta desplegada la aplicación tiene que funcionar sin fallos durante las 24 horas del día, los 365 días del año	Disponibilidad
AT015	La aplicación web tiene que incorporar medidas de seguridad contra el robo de información, ejecución de código JavaScript, 'sanitizar' la entrada de texto de los formularios, etc.	Seguridad
AT016	La pagina web debería de ser responsiva hasta que se desarrolle una versión del juego especifica para dispositivos móviles	Usabilidad
AT017	La pagina podría incluir mecanismos de marketing como botón de compartir con RRSS. Además nos interesa que la pagina este bien posicionada en los principales buscadores por lo que habría que utilizar técnicas SEO para ello.	Marketing
AT018	Escalabilidad. Ahora que nuestro juego va a albergar a una gran cantidad de usuarios jugando al mismo tiempo es necesario que los sistemas aguanten bien esa carga extra	Disponibilidad

Atributos de calidad e interesados

Atributos vs Interesados	ST-01	ST-02	ST-03	ST-04
AT001	X			X
AT002	X			X
AT003	X			X
AT004	X	X		X
AT005	X			
AT006				X
AT007		X		
AT008	X	X	X	X
AT009	X	X		
AT010	X	X		X
AT011	X			
AT012	X		X	
AT013			X	
AT014	X	X		X
AT015	X	X	X	X
AT016	X			X
AT017	X			
AT018	X		X	

Escenarios de calidad.

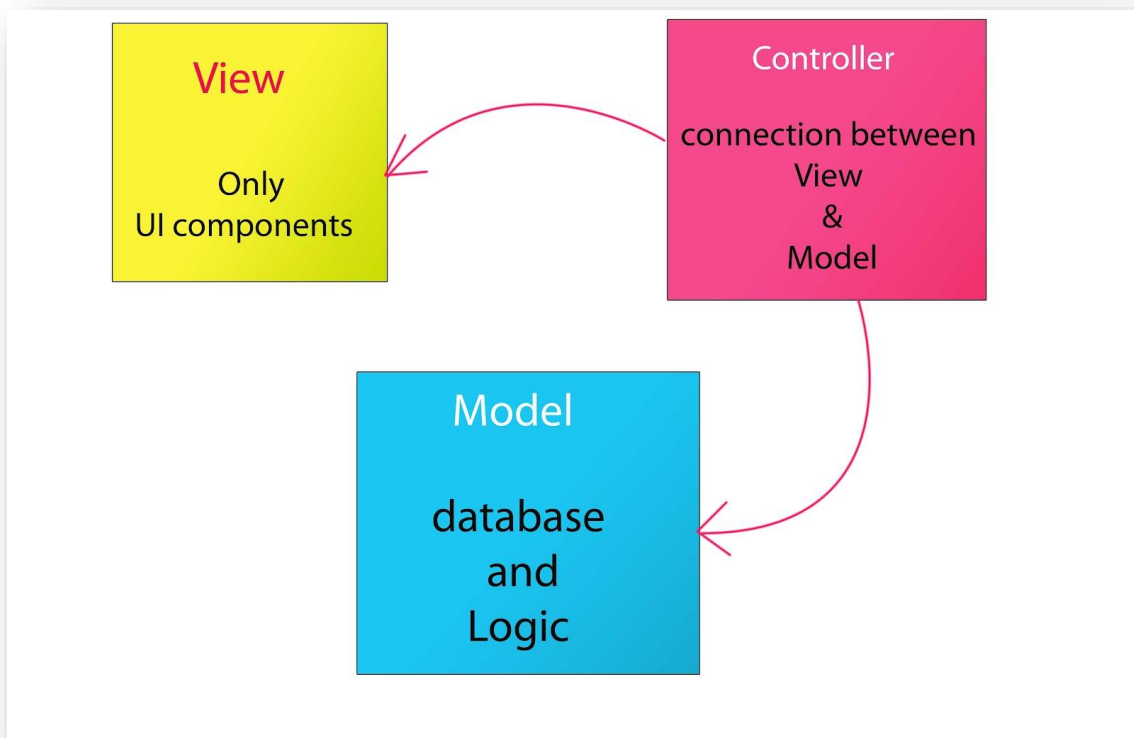
Escenario Nº	Fuente de estímulo	Estímulo	Entorno	Artefacto	Respuesta	Medición de la respuesta	Atributo de calidad afectado
1	Usuario	Comienzo de una partida	Explotación	Sistema	El sistema muestra el tablero elegido y le presenta al usuario preguntas	Tiempo no superior a 1 segundo desde que se lleva a cabo la acción	AT009 AT006 AT001 AT014 AT016
2	Usuario	Seleccionar una apariencia de tablero distinta	Explotación	Interfaz	Cambio de la interfaz dependiendo de la opción seleccionada	Tiempo no superior a 3 segundos desde que se lleva a cabo la acción	AT006 AT002 AT001 AT016

3	Usuario	Acceso a sus estadísticas de juego	Explotación	Sistema	El sistema muestra al usuario las estadísticas de sus juegos	Datos bien calculados a partir de los datos almacenados sobre el usuario	AT010 AT007 AT004 AT016
4	Administrador de la aplicación	Acceso al panel de estadísticas globales	Explotación	Sistema	Se presentan en pantalla los datos relativos a las estadísticas globales del juego	Datos bien calculados a partir de los datos almacenados sobre las partidas jugadas por los usuarios	AT010 AT007 AT006 AT004
5	Directivos	Cambio de las reglas del juego	Desarrollo	Sistema	Cambio de la funcionalidad de la aplicación hasta que cumpla con lo pedido	Coste desarrollo < 6000€ Tiempo desarrollo < 15 días Reutilización de módulos ya implementados	AT011 AT012 AT005
6	Directivos	Ampliación de la funcionalidad del juego	Desarrollo	Sistema	Implementación de las nuevas características requeridas	Coste desarrollo < 9000€ Tiempo desarrollo < 45 días	AT011 AT012 AT005 AT018 AT015 AT017
7	Usuario	Juego simultaneo de varios usuarios	Explotación	MongoDB	No se pueden producir fallos de lectura ni dejar la BBDD en un estado inconsistente	Sin errores. Tiempo de lectura < 500ms Tiempo de escritura < 800ms	AT008 AT003 AT018 AT014

8	Herramientas de desarrollo	Ejecución de test	Explotación	Modelo de datos. Lógica de Negocio	El sistema nos produce unos resultados de todo lo ocurrido durante la ejecución de los test	Test ejecutados con cada build. Cobertura de los test > 50% Test todos en verde	AT013 AT011 AT012
---	----------------------------	-------------------	-------------	---------------------------------------	---	---	---------------------------------

Descripción de negocio de la solución

Para el desarrollo de esta segunda ampliación de nuestra aplicación Trivial hemos optado por implementar una solución basada en la arquitectura MVC (Modelo – Vista – Controlador). Esta decisión ha sido tomada en base a que se deja intuir que el próximas versiones del juego la representación visual del mismo variará. Con MVC solucionamos de un plumazo el problema de tener que hacer una aplicación diferente para cada tipo de vista, ya que lo que vamos a tener de esta manera será un modelo que no variará (en la medida de lo posible) junto a un controlador que es el encargado de comunicar el modelo con las diferentes vistas que va a tener el juego.



Al haber seguido este modelo de manera más o menos fiel la estructura del proyecto resultante está compuesta de las siguientes partes (entre otras):

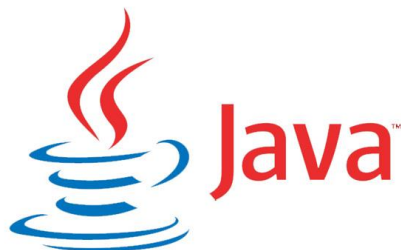
- Modelo
 - Aquí está representada la lógica de negocio y el estado. Es independiente del controlador y de las distintas vistas
- Business / Controlador
 - Esta parte es la encargada de procesar los eventos de los jugadores y realizar las acciones acordes a esos eventos.
- Presentación / Vista
 - Muestra los contenidos del modelo

Además de las tres partes principales que ya han sido nombradas se usan otra serie de patrones del software y metodologías con la intención de que nuestra aplicación posea un alto grado de cohesión y poco acoplamiento.

En esta última entrega para la realización de la aplicación web hemos optado por utilizar el framework MVC Play para aplicaciones altamente escalables.

Tecnologías utilizadas.

- Java: Lenguaje de programación principal elegido por su robustez y la base de conocimiento de la que los miembros del equipo tenemos



- Eclipse: Entorno de desarrollo que hemos utilizado para desarrollar la aplicación en combinación con algunos plugins como el plugin de Git y el de Gradle



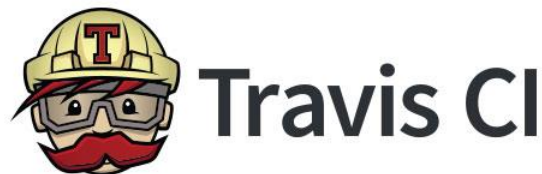
- Github: es un repositorio de Git basado en la web para alojar proyectos utilizando dicho sistema de control de versiones.



- MongoDB: Base de datos documental Open-source que provee un gran rendimiento y una alta disponibilidad además de ser fácilmente escalable.



- TravisCI: es un servidor de integración continua que está integrado con Github y ofrece soporte para un gran número de lenguajes



- Cucumber: Framework de pruebas que permite a los equipos de desarrollo describir como debe comportarse el programa en texto plano. Este texto es escrito en un lenguaje de dominio específico en el que resulta fácil leer los términos de negocio y sirve de documentación.



- GSON: Es una librería java que puede ser usadas para convertir objetos java a su representación JSON. También puede ser usada para convertir desde un Sting JSON a su objeto java equivalente.



- Sbt: es una herramienta que utilizamos para automatizar las tareas de construcción de nuestra aplicación



- Framework Play: framework para construir aplicaciones web con Java y Scala. Está basado en una arquitectura ligera y sin estado.



- Less: preprocesador de CSS que extiende el lenguaje para añadir características adicionales para hacer el código CSS mas mantenible y fácil de extender.



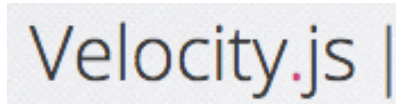
- Datatables: plugin de JQuery que utilizamos para realizar algunas acciones con algunas tables, como ordenarlas.



- Highcharts: librería JavaScript que utilizamos para mostrar las estadísticas de los juegos con gráficos interactivos.



- Velocity.js: Es un motor de animaciones que mejora algunas características del método \$.animate() de JQuery. Lo utilizamos para la animación del logo, para mover la ficha, etc.



- Bootstrap: framework que utilizamos para construir las distintas páginas web de las que se compone el juego.



- JUnit: framework utilizado para probar la parte de lógica de negocio desarrollada en java.

JUnit

- Template engine de Play: utiliza Html5 en conjunción con una serie de características adicionales que nos proporciona el framework Play.

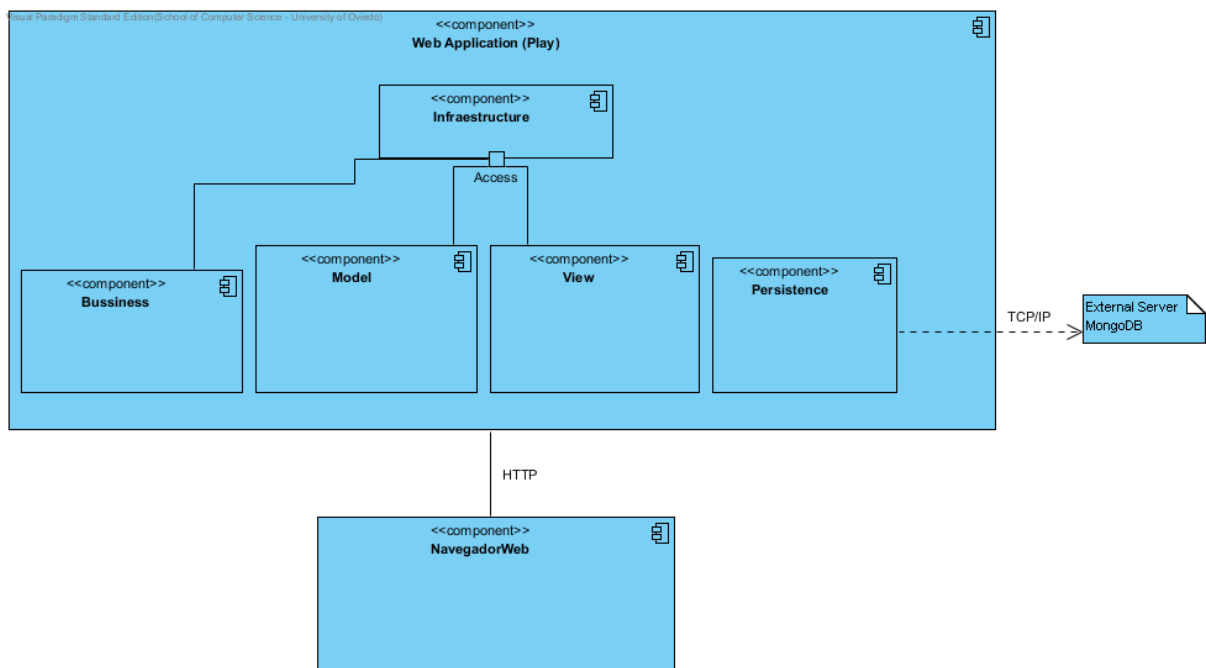
Vistas

En esta parte de la documentación pasamos a comentar las diferentes vistas que hemos realizado con el objetivo de representar de forma visual las diferentes partes de la aplicación y que además nos sirven para entender de una forma mas intuitiva el funcionamiento del juego en su conjunto. También podríamos encuadrar esta sección en la documentación generada a partir de la información creada dentro de la **herramienta CASE**.










DOCUMENTACIÓN CASE

Diagrama de Componentes

Trivial WebAppPlay



Resumen

Nombre	Documentación
 Web Application (Play)	
 Infraestructure	Nos proporciona un punto de entrada único a la lógica de la aplicación, las factorías resuelven los accesos de manera eficiente para la memoria.
 Access	
 Model	El modelo conforma la estructura de objetos de la aplicación, puede verse con más detalle en el diagrama de clases.
 View	Las vistas conforman la interfaz web de la aplicación, las tecnologías usadas son html, bootstraps, less, javascript, less.. etc.
 Persistence	La base de datos de mongo se ha alojado en un servidor para facilitar el acceso la misma.
 Bussiness	Componente con la lógica de negocio (business) de la aplicación, contiene los elementos que aportan funcionalidad a la aplicación para resolver las acciones emitidas por las vistas.
 NavegadorWeb	En un release final el navegador web accedería por medio de http al servidor donde estuviera alojado, en nuestro caso accede de modo local mediante localhost.
 N/A	External Server MongoDB






Detalles

Web Application (Play)






Nombre	Valor
Activo	false
Business Key Mutable	true
Modelo de Negocio	false
Visibilidad	public
Abstracto	false
Hoja	false
Raíz	false
Quality Score	Fair
Quality Reason	Problem Suggestion

	Model element at root	Add model element to a container
	Name does not contain glossary terms	Define name as term in glossary
Indirectly Instantiated	true	

Hijos

Nombre	Documentación
 Infraestructure	Nos proporciona un punto de entrada único a la lógica de la aplicación, las factorías resuelven los accesos de manera eficiente para la memoria.
 Model	El modelo conforma la estructura de objetos de la aplicación, puede verse con más detalle en el diagrama de clases.
 View	Las vistas conforman la interfaz web de la aplicación, las tecnologías usadas son html, bootstraps, less, javascript, less.. etc.
 Persistence	La base de datos de mongo se ha alojado en un servidor para facilitar el acceso la misma.
 Bussiness	Componente con la lógica de negocio (business) de la aplicación, contiene los elementos que aportan funcionalidad a la aplicación para resolver las acciones emitidas por las vistas.

Resident Components


Nombre	Documentación
 Bussiness	Componente con la lógica de negocio (business) de la aplicación, contiene los elementos que aportan funcionalidad a la aplicación para resolver las acciones emitidas por las vistas.
 Model	El modelo conforma la estructura de objetos de la aplicación, puede verse con más detalle en el diagrama de clases.
 View	Las vistas conforman la interfaz web de la aplicación, las tecnologías usadas son html, bootstraps, less, javascript, less.. etc.
 Persistence	La base de datos de mongo se ha alojado en un servidor para facilitar el acceso la misma.
 Infraestructure	Nos proporciona un punto de entrada único a la lógica de la aplicación, las factorías

	resuelven los accesos de manera eficiente para la memoria.
--	--


Infraestructure

Nombre	Valor	
Documentación	Nos proporciona un punto de entrada único a la lógica de la aplicación, las factorías resuelven los accesos de manera eficiente para la memoria.	
Activo	false	
Business Key Mutable	true	
Modelo de Negocio	false	
Visibilidad	public	
Abstracto	false	
Hoja	false	
Raíz	false	
Quality Score	Very Bad	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary
	Model element without any relationship	Connect model element with relationship
Indirectly Instantiated	true	

Hijos

Nombre	Documentación
 Access	

Ports

Nombre	Documentación
 Access	

Access

Nombre	Valor
--------	-------

Servicio	true
Comportamiento	false
Conjugated	false
Derivado	false
Is ID	false
Derived Union	false
Aggregation	Unspecified
Multiplicidad	Unspecified
Read Only	false
Static	false
Hoja	false
Visibilidad	public

Model

Nombre	Valor	
Documentación	El modelo conforma la estructura de objetos de la aplicación, puede verse con más detalle en el diagrama de clases.	
Activo	false	
Business Key Mutable	true	
Modelo de Negocio	false	
Visibilidad	public	
Abstracto	false	
Hoja	false	
Raíz	false	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary
Indirectly Instantiated	true	

View

Nombre	Valor
Documentación	Las vistas conforman la interfaz web de la aplicación, las tecnologías usadas son html, bootstraps, less, javascript, less.. etc.
Activo	false

Business Key Mutable	true	
Modelo de Negocio	false	
Visibilidad	public	
Abstracto	false	
Hoja	false	
Raíz	false	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary
Indirectly Instantiated	true	

 Persistence

Nombre	Valor	
Documentación	La base de datos de mongo se ha alojado en un servidor para facilitar el acceso la misma.	
Activo	false	
Business Key Mutable	true	
Modelo de Negocio	false	
Visibilidad	public	
Abstracto	false	
Hoja	false	
Raíz	false	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary
Indirectly Instantiated	true	

 Bussiness

Nombre	Valor
Documentación	Componente con la lógica de negocio (business) de la aplicación, contiene los elementos que aportan funcionalidad a la aplicación para resolver las acciones emitidas por las vistas.
Activo	false

Business Key Mutable	true	
Modelo de Negocio	false	
Visibilidad	public	
Abstracto	false	
Hoja	false	
Raíz	false	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary
Indirectly Instantiated	true	



NavegadorWeb

Nombre	Valor	
Documentación	En un release final el navegador web accedería por medio de http al servidor donde estuviera alojado, en nuestro caso accede de modo local mediante localhost.	
Activo	false	
Business Key Mutable	true	
Modelo de Negocio	false	
Visibilidad	public	
Abstracto	false	
Hoja	false	
Raíz	false	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Model element at root	Add model element to a container
	Name does not contain glossary terms	Define name as term in glossary
Indirectly Instantiated	true	



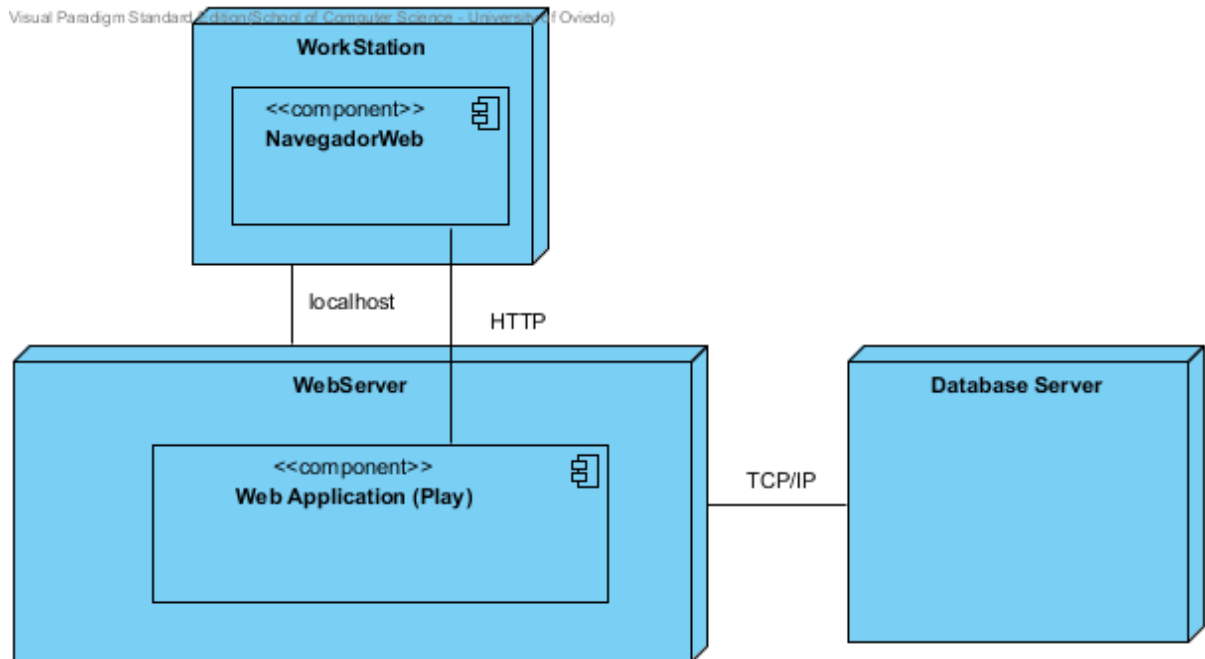
N/A

Nombre	Valor
Documentación	External Server MongoDB
Quality Score	Fair






Quality Reason	Problem	Suggestion
	Model element at root	Add model element to a container

Diagrama de Despliegue

Despliegue Trivial3a



Resumen

Nombre	Documentación
 WorkStation	
 NavegadorWeb	En un release final el navegador web accedería por medio de http al servidor donde estuviera alojado, en nuestro caso accede de modo local mediante localhost.
 WebServer	En el web Server se situara la aplicación desarrollada en play, se hace alusión al componente concreto del diagrama de componentes y se especifica que la conexión actual es local.
 Database Server	Servidor de mongo donde se aloja la bdd.
 Web Application (Play)	

Documentación

Estación de trabajo local.

Detalles

 WorkStation

Nombre	Valor
Activo	false
Business Key Mutable	true
Modelo de Negocio	false
Visibilidad	public
Abstracto	false
Hoja	false
Raíz	false

 NavegadorWeb

Nombre	Valor	
Documentación	En un release final el navegador web accedería por medio de http al servidor donde estuviera alojado, en nuestro caso accede de modo local mediante localhost.	
Activo	false	
Business Key Mutable	true	
Modelo de Negocio	false	
Visibilidad	public	
Abstracto	false	
Hoja	false	
Raíz	false	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Model element at root	Add model element to a container
	Name does not contain glossary terms	Define name as term in glossary
Indirectly Instantiated	true	

 WebServer

Nombre	Valor
Documentación	En el web Server se situara la aplicación desarrollada en play, se hace alusion al componente concreto del diagrama de componentes y se especifica que la conexion actual es local.
Activo	false
Business Key Mutable	true
Modelo de Negocio	false
Visibilidad	public
Abstracto	false
Hoja	false
Raíz	false

 Database Server






Nombre	Valor
Documentación	Servidor de mongo donde se aloja la bdd.
Activo	false
Business Key Mutable	true
Modelo de Negocio	false
Visibilidad	public
Abstracto	false
Hoja	false
Raíz	false

 Web Application (Play)





Nombre	Valor
Activo	false
Business Key Mutable	true
Modelo de Negocio	false
Visibilidad	public
Abstracto	false
Hoja	false
Raíz	false
Quality Score	Fair

Quality Reason	Problem	Suggestion
	Model element at root	Add model element to a container
	Name does not contain glossary terms	Define name as term in glossary
Indirectly Instantiated	true	

Hijos

Nombre	Documentación
 Bussiness	Componente con la lógica de negocio (business) de la aplicación, contiene los elementos que aportan funcionalidad a la aplicación para resolver las acciones emitidas por las vistas.
 Model	El modelo conforma la estructura de objetos de la aplicación, puede verse con más detalle en el diagrama de clases.
 View	Las vistas conforman la interfaz web de la aplicación, las tecnologías usadas son html, bootstraps, less, javascript, less.. etc.
 Persistence	La base de datos de mongo se ha alojado en un servidor para facilitar el acceso la misma.
 Infraestructure	Nos proporciona un punto de entrada único a la lógica de la aplicación, las factorías resuelven los accesos de manera eficiente para la memoria.

Resident Components

Nombre	Documentación
 Bussiness	Componente con la lógica de negocio (business) de la aplicación, contiene los elementos que aportan funcionalidad a la aplicación para resolver las acciones emitidas por las vistas.
 Model	El modelo conforma la estructura de objetos de la aplicación, puede verse con más detalle en el diagrama de clases.
 View	Las vistas conforman la interfaz web de la aplicación, las tecnologías usadas son html, bootstraps, less, javascript, less.. etc.
 Persistence	La base de datos de mongo se ha alojado en un servidor para facilitar el acceso la misma.




 Infraestructure	Nos proporciona un punto de entrada único a la lógica de la aplicación, las factorías resuelven los accesos de manera eficiente para la memoria.
--	--

Diagrama de Clases

Diagrama de clases - Login



Resumen

Nombre	Documentación
 Diagrama de clases - Modelo del juego	Referencia a las clases del juego ya que usan como punto de entrada los jugadores que se han registrado en el login.
 Partida	Clase Partida del Login, guarda los jugadores que han iniciado sesión con sus colores elegidos.

Detalles

Diagrama de clases - Modelo del juego

Nombre	Valor	
Documentación	Referencia a las clases del juego ya que usan como punto de entrada los jugadores que se han registrado en el login.	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary
Diagram Id	t7uzqTKGAqG4AQ8w	

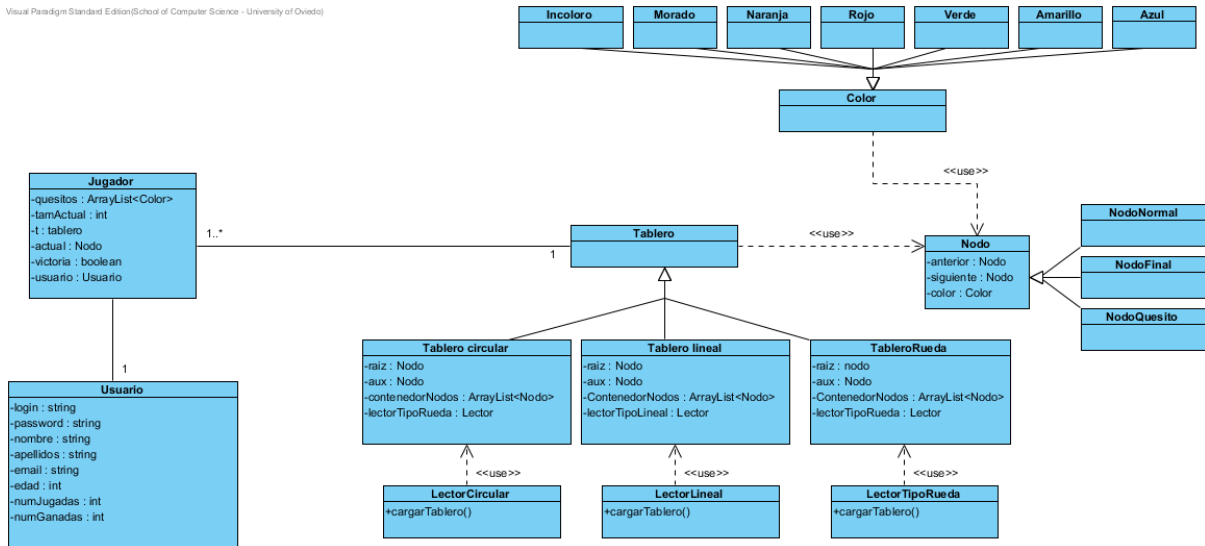

Partida

Nombre	Valor	
Documentación	Clase Partida del Login, guarda los jugadores que han iniciado sesión con sus colores elegidos.	
Activo	false	
Business Key Mutable	true	
Modelo de Negocio	false	
Visibilidad	public	
Hoja	false	
Raíz	false	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary













Diagrama de Clases










Diagrama de clases - Modelo del juego

Visual Paradigm Standard Edition (School of Computer Science - University of Oviedo)



Resumen

Nombre	Documentación
 Incoloro	
 Morado	
 Naranja	
 Rojo	
 Verde	
 Amarillo	
 Azul	
 Color	
 Jugador	Clase Jugador, representa una entidad que puede desarrollar las acciones de juego, a su vez tiene asociado un Usuario como referencia a los atributos de la base de datos asociados a dicho jugador.
 NodoNormal	Implementacion de las casillas corrientes.
 Tablero	El tablero es la representacion del grafo con todos los atributos y referencias necesarias a la hora de realizar las acciones que desencadenara el juego.
 Nodo	Entidad que representa cada casilla dentro del tablero. La equivalencia de "tablero" a grafo es

	equitativa para "casilla" y nodo.
 NodoFinal	Implementacion del Nodo central.
 NodoQuesito	
 Tablero circular	Implementación del tablero circular sin barras centrales.
 Tablero lineal	Tablero lineal, cuadrado, termina el tablero en un nodo determinado.
 TableroRueda	Tablero rueda, convencional de trivial con barras centrales.
 Usuario	Entidad que representa a un usuario en la base de datos.
 LectorCircular	Lector del ficheros para la creación de tablero circular.
 LectorLineal	Lector del ficheros para la creación de tablero lineal.
 LectorTipoRueda	Lector del ficheros para la creación de tablero rueda.

Detalles

Incoloro

Nombre	Valor	
Activo	false	
Business Key Mutable	true	
Modelo de Negocio	false	
Visibilidad	public	
Hoja	false	
Raíz	false	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary

Morado

Nombre	Valor
Activo	false
Business Key Mutable	true

Modelo de Negocio	false	
Visibilidad	public	
Hoja	false	
Raíz	false	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary



Naranja

Nombre	Valor	
Activo	false	
Business Key Mutable	true	
Modelo de Negocio	false	
Visibilidad	public	
Hoja	false	
Raíz	false	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary



Rojo

Nombre	Valor	
Activo	false	
Business Key Mutable	true	
Modelo de Negocio	false	
Visibilidad	public	
Hoja	false	
Raíz	false	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary

 Verde

Nombre	Valor	
Activo	false	
Business Key Mutable	true	
Modelo de Negocio	false	
Visibilidad	public	
Hoja	false	
Raíz	false	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary

 Amarillo

Nombre	Valor	
Activo	false	
Business Key Mutable	true	
Modelo de Negocio	false	
Visibilidad	public	
Hoja	false	
Raíz	false	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary

 Azul

Nombre	Valor
Activo	false
Business Key Mutable	true
Modelo de Negocio	false
Visibilidad	public
Hoja	false
Raíz	false
Quality Score	Fair

Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary

 Color

Nombre	Valor	
Activo	false	
Business Key Mutable	true	
Modelo de Negocio	false	
Visibilidad	public	
Hoja	false	
Raíz	false	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary

 Jugador

Nombre	Valor	
Documentación	Clase Jugador, representa una entidad que puede desarrollar las acciones de juego, a su vez tiene asociado un Usuario como referencia a los atributos de la base de datos asociados a dicho jugador.	
Activo	false	
Business Key Mutable	true	
Modelo de Negocio	false	
Visibilidad	public	
Hoja	false	
Raíz	false	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary


NodoNormal

Nombre	Valor	
Documentación	Implementacion de las casillas corrientes.	
Activo	false	
Business Key Mutable	true	
Modelo de Negocio	false	
Visibilidad	public	
Hoja	false	
Raíz	false	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary


Tablero

Nombre	Valor	
Documentación	El tablero es la representacion del grafo con todos los atributos y referencias necesarias a la hora de realizar las acciones que desencadenara el juego.	
Activo	false	
Business Key Mutable	true	
Modelo de Negocio	false	
Visibilidad	public	
Hoja	false	
Raíz	false	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary


Nodo

Nombre	Valor
Documentación	Entidad que representa cada casilla dentro del tablero. La equivalencia de "tablero" a grafo es equitativa para "casilla" y nodo.
Activo	false

Business Key Mutable	true	
Modelo de Negocio	false	
Visibilidad	public	
Hoja	false	
Raíz	false	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary

NodoFinal

Nombre	Valor	
Documentación	Implementacion del Nodo central.	
Activo	false	
Business Key Mutable	true	
Modelo de Negocio	false	
Visibilidad	public	
Hoja	false	
Raíz	false	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary

NodoQuesito

Nombre	Valor	
Activo	false	
Business Key Mutable	true	
Modelo de Negocio	false	
Visibilidad	public	
Hoja	false	
Raíz	false	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary


Tablero circular

Nombre	Valor	
Documentación	Implementación del tablero circular sin barras centrales.	
Activo	false	
Business Key Mutable	true	
Modelo de Negocio	false	
Visibilidad	public	
Hoja	false	
Raíz	false	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary


Tablero lineal

Nombre	Valor	
Documentación	Tablero lineal, cuadrado, termina el tablero en un nodo determinado.	
Activo	false	
Business Key Mutable	true	
Modelo de Negocio	false	
Visibilidad	public	
Hoja	false	
Raíz	false	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary


TableroRueda

Nombre	Valor
Documentación	Tablero rueda, convencional de trivial con barras centrales.
Activo	false

Business Key Mutable	true	
Modelo de Negocio	false	
Visibilidad	public	
Hoja	false	
Raíz	false	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary



Usuario

Nombre	Valor	
Documentación	Entidad que representa a un usuario en la base de datos.	
Activo	false	
Business Key Mutable	true	
Modelo de Negocio	false	
Visibilidad	public	
Hoja	false	
Raíz	false	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary



LectorCircular

Nombre	Valor	
Documentación	Lector del ficheros para la creación de tablero circular.	
Activo	false	
Business Key Mutable	true	
Modelo de Negocio	false	
Visibilidad	public	
Hoja	false	
Raíz	false	
Quality Score	Fair	

Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary



LectorLineal

Nombre	Valor	
Documentación	Lector del ficheros para la creación de tablero lineal.	
Activo	false	
Business Key Mutable	true	
Modelo de Negocio	false	
Visibilidad	public	
Hoja	false	
Raíz	false	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary

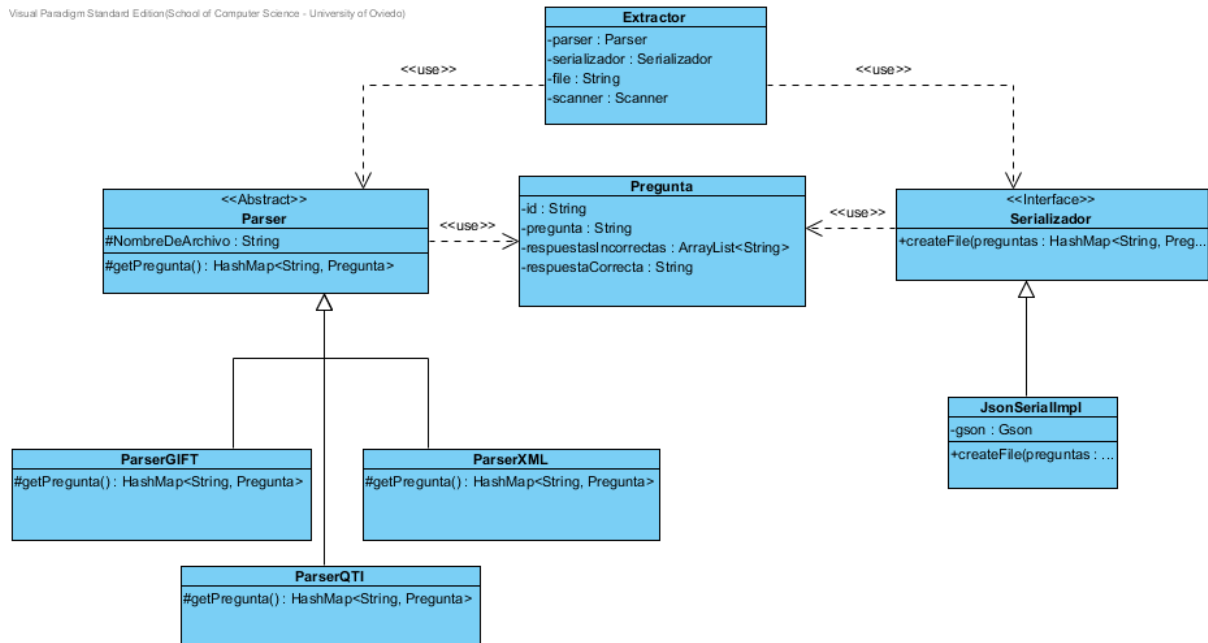


LectorTipoRueda





Nombre	Valor	
Documentación	Lector del ficheros para la creación de tablero rueda.	
Activo	false	
Business Key Mutable	true	
Modelo de Negocio	false	
Visibilidad	public	
Hoja	false	
Raíz	false	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary





Diagrama de Clases

Diagrama de clases - Parser



Resumen

Nombre	Documentación
 Extractor	Clase de integración (controladora) de todos los submodulos del modulo principal de parseo, que crea a partir de fichero de entrada el formato de salida desado.
 Pregunta	Es la clase principal de parseo puesto que contiene el formato intermedio entre el formato de entrada de la implementación concreta del parser y el formato de salida del serializador. Contiene métodos getter y setter para todos los atributos privados, así como los métodos de hashCode, equals y toString que aportan funcionalidad para comparaciones y otras utilidades.
 Parser	Esta clase abstracta actua como superclase para las clases hijas que especificaran el tipo de parser según el formato de entrada del archivo que es el atributo de esta clase (nombreArchivo)
 Serializador	Esta interfaz forma la superclase de las implementaciones concretas de serializado a formatos de salida concretos. Utiliza el formato intermedio generado por el parser y crea el fichero de salida según la implementación concreta.

 JsonSerializerImpl	Implementación concreta de la superclase para transformar a formato JSON.
 ParserGIFT	Esta es la implementación de la superclase para un formato de entrada GIFT, toda la funcionalidad está en el método protegido <code>getPregunta()</code>
 ParserXML	Esta es la implementación de la superclase para un formato de entrada XML plano, toda la funcionalidad está en el método protegido <code>getPregunta()</code>
 ParserQTI	Esta es la implementación de la superclase para un formato de entrada XML QTI, toda la funcionalidad está en el método protegido <code>getPregunta()</code>

Detalles

Extractor

Nombre	Valor	
Documentación	Clase de integración (controladora) de todos los submodulos del modulo principal de parseo, que crea a partir de fichero de entrada el formato de salida desado.	
Activo	false	
Business Key Mutable	true	
Modelo de Negocio	false	
Visibilidad	public	
Hoja	false	
Raíz	false	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary

Pregunta

Nombre	Valor
Documentación	Es la clase principal de parseo puesto que contiene el formato intermedio entre el formato de entrada de la implementación concreta del parser y el formato de salida del serializador. Contiene métodos getter y setter para todos los atributos privados, así como los métodos de <code>hashCode</code> , <code>equals</code> y <code>toString</code> que aportan funcionalidad para comparaciones y otras utilidades.

Activo	false	
Business Key Mutable	true	
Modelo de Negocio	false	
Visibilidad	public	
Hoja	false	
Raíz	false	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary

Parser

Nombre	Valor	
Documentación	Esta clase abstracta actua como superclase para las clases hijas que esepificicaran el tipo de parser según el formato de entrada del archivo que es el atributo de esta clase (nombreArchivo)	
Activo	false	
Business Key Mutable	true	
Modelo de Negocio	false	
Visibilidad	public	
Hoja	false	
Raíz	false	
Estereotipos	Abstract	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary

Serializador

Nombre	Valor
Documentación	Esta interfaz forma la superclase de las implementaciones concretas de serializado a formatos de salida concretos. Utiliza el formato intermedio generado por el parser y crea el fichero de salida según la implementación concreta.
Activo	false

Business Key Mutable	true	
Modelo de Negocio	false	
Visibilidad	public	
Hoja	false	
Raíz	false	
Estereotipos	Interface	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary



JsonSerialImpl

Nombre	Valor	
Documentación	Implementación concreta de la superclase para transformar a formato JSON.	
Activo	false	
Business Key Mutable	true	
Modelo de Negocio	false	
Visibilidad	public	
Hoja	false	
Raíz	false	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary



N/A

Nombre	Valor
Visibilidad	Unspecified
Quality Score	Good
Covering	false
Disjoint	false



ParserGIFT

Nombre	Valor
--------	-------

Documentación	Esta es la implementación de la superclase para un formato de entrada GIFT, toda la funcionalidad está en el método protegido <code>getPregunta()</code>	
Activo	false	
Business Key Mutable	true	
Modelo de Negocio	false	
Visibilidad	public	
Hoja	false	
Raíz	false	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary



ParserXML

Nombre	Valor	
Documentación	Esta es la implementación de la superclase para un formato de entrada XML plano, toda la funcionalidad está en el método protegido <code>getPregunta()</code>	
Activo	false	
Business Key Mutable	true	
Modelo de Negocio	false	
Visibilidad	public	
Hoja	false	
Raíz	false	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary



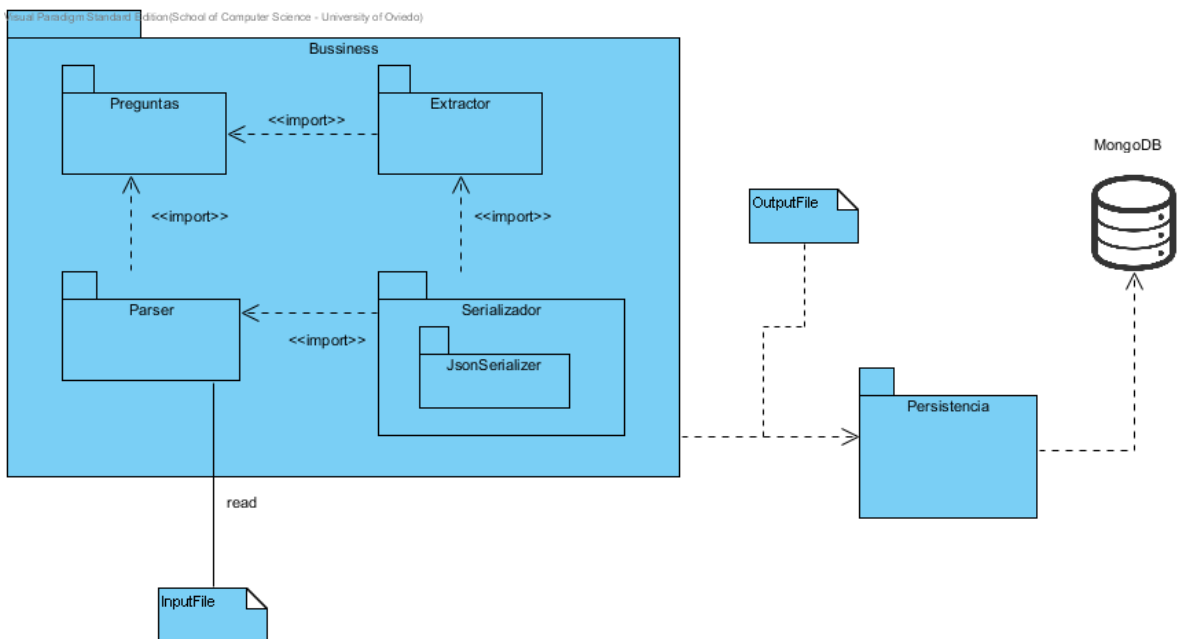
ParserQTI

Nombre	Valor	
Documentación	Esta es la implementación de la superclase para un formato de entrada XML QTI, toda la funcionalidad está en el método protegido <code>getPregunta()</code>	
Activo	false	




Business Key Mutable	true	
Modelo de Negocio	false	
Visibilidad	public	
Hoja	false	
Raíz	false	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary








Diagrama de paquetes

Diagrama de paquetes - Extractor



Resumen

Nombre	Documentación
 Bussiness	Paquete que contiene toda la lógica de nuestra aplicación.
 Preguntas	Representación interna que usa la aplicación. Es el tipo de objeto creado a partir de los datos leídos por uno de los parsers disponibles al leer un fichero de entrada.
 Extractor	Paquete que actúa a modo de clase controladora de la



	aplicación. Es en este paquete en el que se presentan los modos de interacción disponibles al usuario del sistema.
 MongoDB	Base de datos de nuestra aplicación, MongoDB
 Parser	Paquete parser: Engloba a todos los tipos de parsers que soporta la aplicación, habiendo un tipo de parser por cada formato de fichero de entrada aceptado.
 Serializador	
 JsonSerializer	Paquete en el que se lleva a cabo la serialización al formato JSON a partir de el formato de representación interna que utiliza la aplicación.
 Persistencia	Modulo encargado de la persistencia de los datos a la base de datos.
 N/A	OutputFile
 N/A	InputFile



Detalles

Bussiness

Nombre	Valor	
Documentación	Paquete que contiene toda la lógica de nuestra aplicación.	
Abstracto	false	
Hoja	false	
Raíz	false	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary
Visibilidad	public	

Hijos

Nombre	Documentación
 Preguntas	Representación interna que usa la aplicación. Es el tipo de objeto creado a partir de los datos leídos por uno de los parsers disponibles al leer un fichero de entrada.
 Extractor	Paquete que actúa a modo de clase controladora de la aplicación. Es en este

	paquete en el que se presentan los modos de interacción disponibles al usuario del sistema.
 Parser	Paquete parser: Engloba a todos los tipos de parsers que soporta la aplicación, habiendo un tipo de parser por cada formato de fichero de entrada aceptado.
 Serializador	

Preguntas

Nombre	Valor	
Documentación	Representación interna que usa la aplicación. Es el tipo de objeto creado a partir de los datos leídos por uno de los parsers disponibles al leer un fichero de entrada.	
Abstracto	false	
Hoja	false	
Raíz	false	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary
Visibilidad	public	

Extractor

Nombre	Valor	
Documentación	Paquete que actúa a modo de clase controladora de la aplicación. Es en este paquete en el que se presentan los modos de interacción disponibles al usuario del sistema.	
Abstracto	false	
Hoja	false	
Raíz	false	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary

Visibilidad	public
-------------	--------

 MongoDB

Nombre	Valor	
Documentación	Base de datos de nuestra aplicación, MongoDB	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary


 Parser

Nombre	Valor	
Documentación	Paquete parser: Engloba a todos los tipos de parsers que soporta la aplicación, habiendo un tipo de parser por cada formato de fichero de entrada aceptado.	
Abstracto	false	
Hoja	false	
Raíz	false	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary
Visibilidad	public	

 Serializador

Nombre	Valor	
Abstracto	false	
Hoja	false	
Raíz	false	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary
Visibilidad	public	

Hijos

Nombre	Documentación
 JsonSerializer	Paquete en el que se lleva a cabo la serialización al formato JSON a partir de el formato de representación interna que utiliza la aplicación.

 JsonSerializer

Nombre	Valor	
Documentación	Paquete en el que se lleva a cabo la serialización al formato JSON a partir de el formato de representación interna que utiliza la aplicación.	
Abstracto	false	
Hoja	false	
Raíz	false	
Quality Score	Very Bad	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary
	Model element without any relationship	Connect model element with relationship
Visibilidad	public	

 Persistencia

Nombre	Valor	
Documentación	Modulo encargado de la persistencia de los datos a la base de datos.	
Abstracto	false	
Hoja	false	
Raíz	false	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary
Visibilidad	public	

 N/A

Nombre	Valor
Documentación	OutputFile
Quality Score	Good

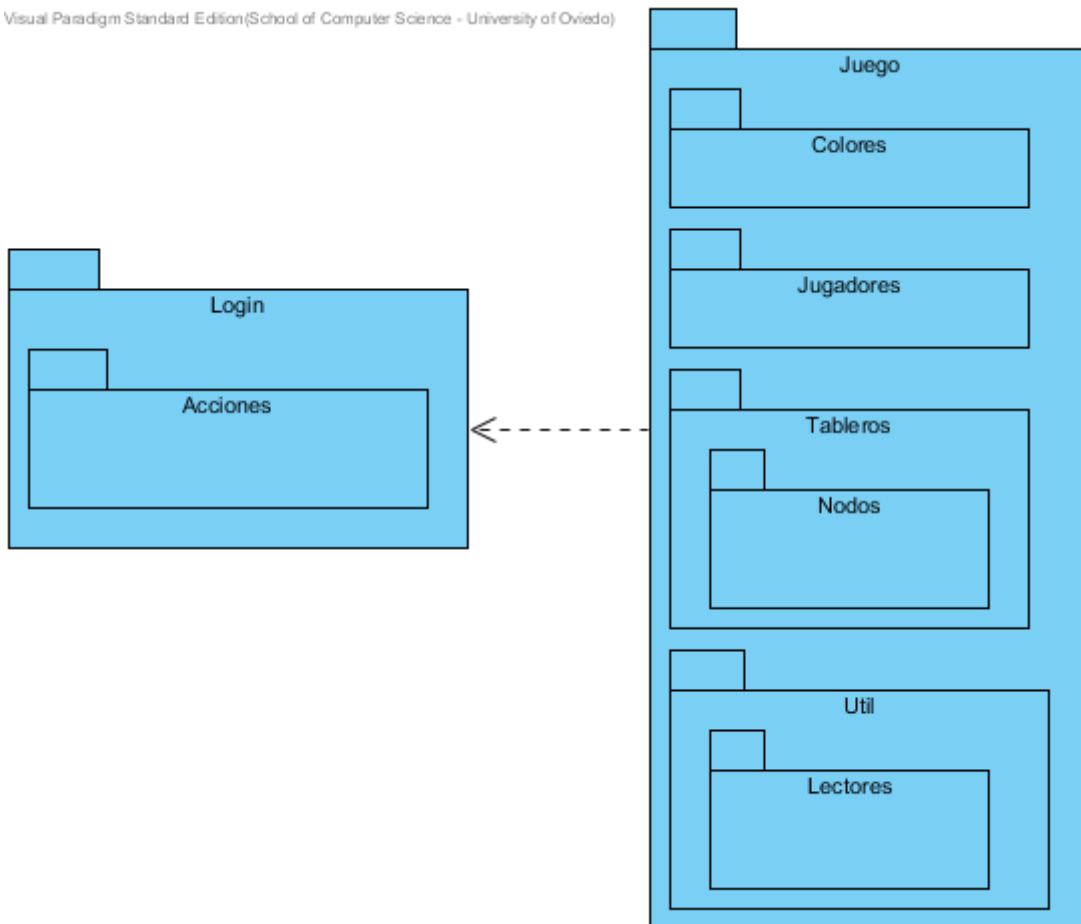
 N/A

Nombre	Valor
Documentación	InputFile
Quality Score	Good










Diagrama de paquetes

Diagrama de paquetes - Juego

Visual Paradigm Standard Edition (School of Computer Science - University of Oviedo)



Resumen





Nombre	Documentación
 Juego	
 Colores	
 Jugadores	
 Login	
 Acciones	
 Tableros	
 Nodos	
 Util	
 Lectores	

Detalles

Juego

Nombre	Valor	
Abstracto	false	
Hoja	false	
Raíz	false	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary
Visibilidad	public	

Hijos

Nombre	Documentación
 Colores	
 Jugadores	
 Tableros	
 Util	

 Colores

Nombre	Valor	
Abstracto	false	
Hoja	false	
Raíz	false	
Quality Score	Very Bad	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary
	Model element without any relationship	Connect model element with relationship
Visibilidad	public	


 Jugadores

Nombre	Valor	
Abstracto	false	
Hoja	false	
Raíz	false	
Quality Score	Very Bad	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary
	Model element without any relationship	Connect model element with relationship
Visibilidad	public	

 Login

Nombre	Valor	
Abstracto	false	
Hoja	false	
Raíz	false	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary
Visibilidad	public	

Hijos

Nombre	Documentación
 Acciones	


 Acciones

Nombre	Valor	
Abstracto	false	
Hoja	false	
Raíz	false	
Quality Score	Very Bad	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary
	Model element without any relationship	Connect model element with relationship
Visibilidad	public	

 Tableros

Nombre	Valor	
Abstracto	false	
Hoja	false	
Raíz	false	
Quality Score	Very Bad	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary
	Model element without any relationship	Connect model element with relationship
Visibilidad	public	

Hijos

Nombre	Documentación
 Nodos	


 Nodos

Nombre	Valor	
Abstracto	false	
Hoja	false	
Raíz	false	
Quality Score	Very Bad	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary
	Model element without any relationship	Connect model element with relationship
Visibilidad	public	

 Util

Nombre	Valor	
Abstracto	false	
Hoja	false	
Raíz	false	
Quality Score	Very Bad	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary
	Model element without any relationship	Connect model element with relationship
Visibilidad	public	

Hijos

Nombre	Documentación
 Lectores	

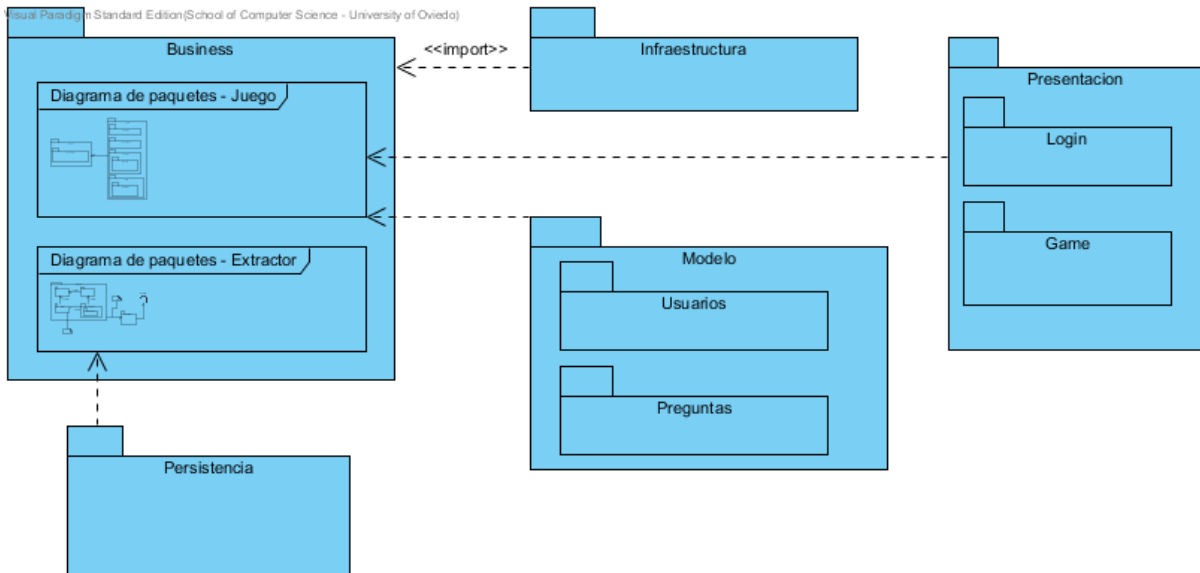
 Lectores

Nombre	Valor
--------	-------












Abstracto	false	
Hoja	false	
Raíz	false	
Quality Score	Very Bad	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary
	Model element without any relationship	Connect model element with relationship
Visibilidad	public	

Diagrama de paquetes

Diagrama de paquetes - Trivial



Resumen



Nombre	Documentación
 Business	
 Infraestructura	
 Presentacion	
 Diagrama de paquetes - Juego	
 Login	
 Game	
 Modelo	
 Diagrama de paquetes - Extractor	
 Usuarios	
 Preguntas	
 Persistencia	Modulo encargado de la persistencia de los datos a la base de datos.

Detalles

Business

Nombre	Valor	
Abstracto	false	
Hoja	false	
Raíz	false	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary
Visibilidad	public	

Hijos

Nombre	Documentación
 Diagrama de paquetes - Juego	
 Diagrama de paquetes - Extractor	

Infraestructura

Nombre	Valor	
Abstracto	false	
Hoja	false	
Raíz	false	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary
Visibilidad	public	

Presentacion

Nombre	Valor
--------	-------

Abstracto	false	
Hoja	false	
Raíz	false	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary
Visibilidad	public	

Hijos



Nombre	Documentación
 Login	
 Game	

 Diagrama de paquetes - Juego

Nombre	Valor	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary
Diagram Id	yfIQaTKGAqG4ARRI	

 Login

Nombre	Valor	
Abstracto	false	
Hoja	false	
Raíz	false	
Quality Score	Very Bad	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary
	Model element without any relationship	Connect model element with relationship
Visibilidad	public	

 Game

Nombre	Valor	
Abstracto	false	
Hoja	false	
Raíz	false	
Quality Score	Very Bad	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary
	Model element without any relationship	Connect model element with relationship
Visibilidad	public	

 Modelo

Nombre	Valor	
Abstracto	false	
Hoja	false	
Raíz	false	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary
Visibilidad	public	

Hijos



Nombre	Documentación
 Usuarios	
 Preguntas	

 Diagrama de paquetes - Extractor

Nombre	Valor
Quality Score	Fair

Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary
Diagram Id	BeWJ.9KGAqG4AQUo	

Usuarios

Nombre	Valor	
Abstracto	false	
Hoja	false	
Raíz	false	
Quality Score	Very Bad	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary
	Model element without any relationship	Connect model element with relationship
Visibilidad	public	

Preguntas

Nombre	Valor	
Abstracto	false	
Hoja	false	
Raíz	false	
Quality Score	Very Bad	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary
	Model element without any relationship	Connect model element with relationship
Visibilidad	public	

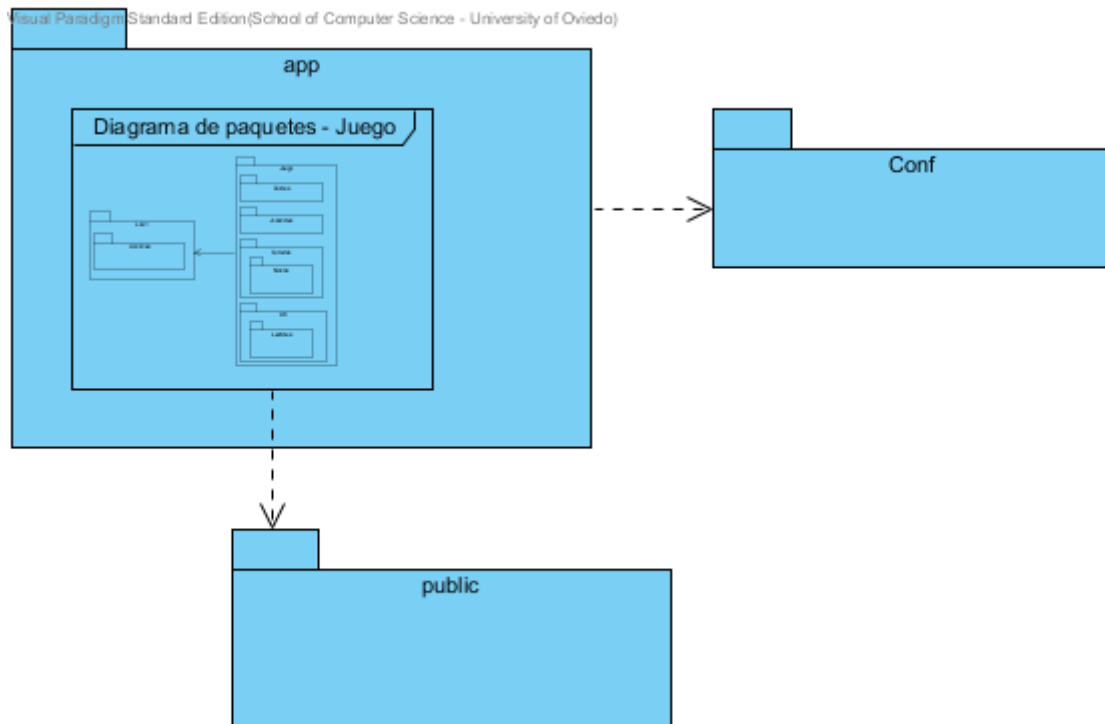
Persistencia

Nombre	Valor
Documentación	Modulo encargado de la persistencia de los datos a la base de datos.
Abstracto	false




Hoja	false	
Raíz	false	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary
Visibilidad	public	


Diagrama de paquetes

WebApplication



Resumen

Nombre	Documentación
 app	En el framework play alojamos toda la funcionalidad dentro del paquete app. Toda nuestra arquitectura MVC se sitúa dentro de este paquete.
 Diagrama de paquetes - Juego	
 Conf	Paquete en el que se sitúan los elementos de configuración como la hoja de rutas de la

	aplicación, o la configuración de la app.
 public	Aquí se situaran elementos físicos (gráficos) que se utilizaran en las vistas, vease, imagenes, videos, musica..

Detalles

app

Nombre	Valor	
Documentación	En el framework play alojamos toda la funcionalidad dentro del paquete app. Toda nuestra arquitectura MVC se situa dentro de este paquete.	
Abstracto	false	
Hoja	false	
Raíz	false	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary
Visibilidad	public	

Hijos


Nombre	Documentación
 Diagrama de paquetes - Juego	

Diagrama de paquetes - Juego

Nombre	Valor	
Quality Score	Very Bad	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary
	Model element without any relationship	Connect model element with relationship
Diagram Id	yfIQaTKGAqG4ARRI	

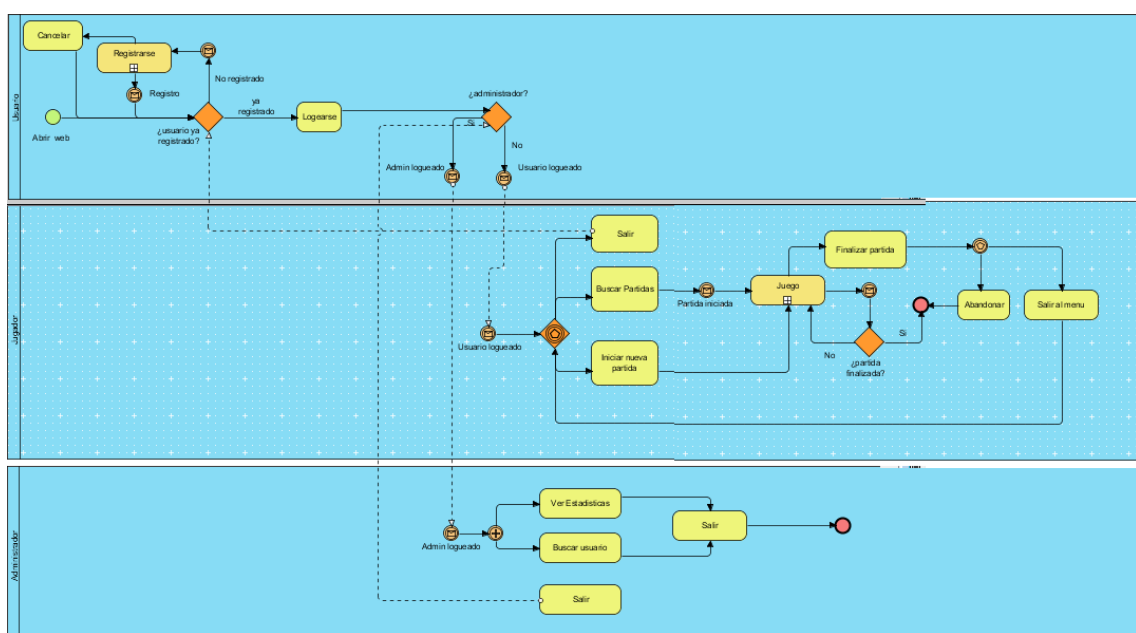
 Conf

Nombre	Valor	
Documentación	Paquete en el que se sitúan los elementos de configuración como la hoja de rutas de la aplicación, o la configuración de la app.	
Abstracto	false	
Hoja	false	
Raíz	false	
Quality Score	Fair	
Quality Reason	Problem	Suggestion
	Name does not contain glossary terms	Define name as term in glossary
Visibilidad	public	













 public


Nombre	Valor
Documentación	Aquí se situarían elementos físicos (gráficos) que se utilizarían en las vistas, vease, imágenes, videos, música..
Abstracto	false
Hoja	false
Raíz	false
Visibilidad	public

BUSINESS MODELING (BPM)

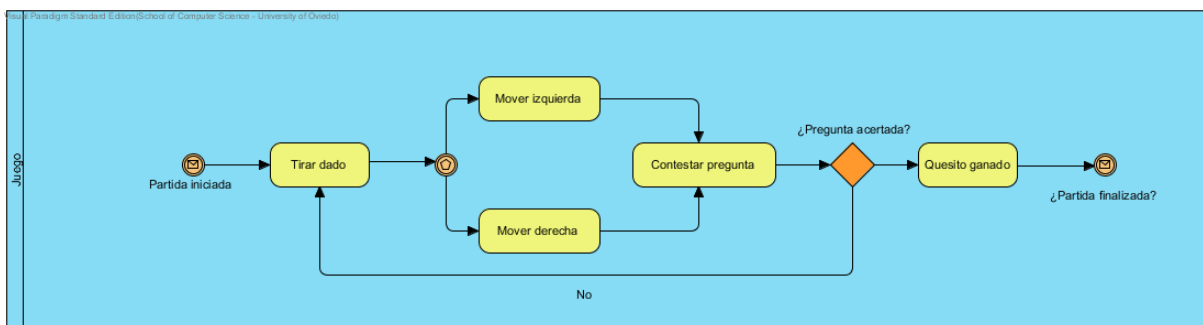


Resumen










Nombre	Documentación
 Usuario	
 Administrador	
 Jugador	
 Abrir web	
 ¿usuario ya registrado?	
 No registrado	
 Registrarse	
 Registro	
 Cancelar	
 Logearse	
 ¿administrador?	
 Admin logueado	

 Admin logueado	
 Ver Estadísticas	
 Usuario logueado	
 Usuario logueado	
 Buscar Partidas	
 Partida iniciada	
 Iniciar nueva partida	
 Salir	
 Juego	
 ¿partida finalizada?	
 Finalizar partida	
 Salir al menu	
 Abandonar	
 Buscar usuario	
 Salir	
 Salir	

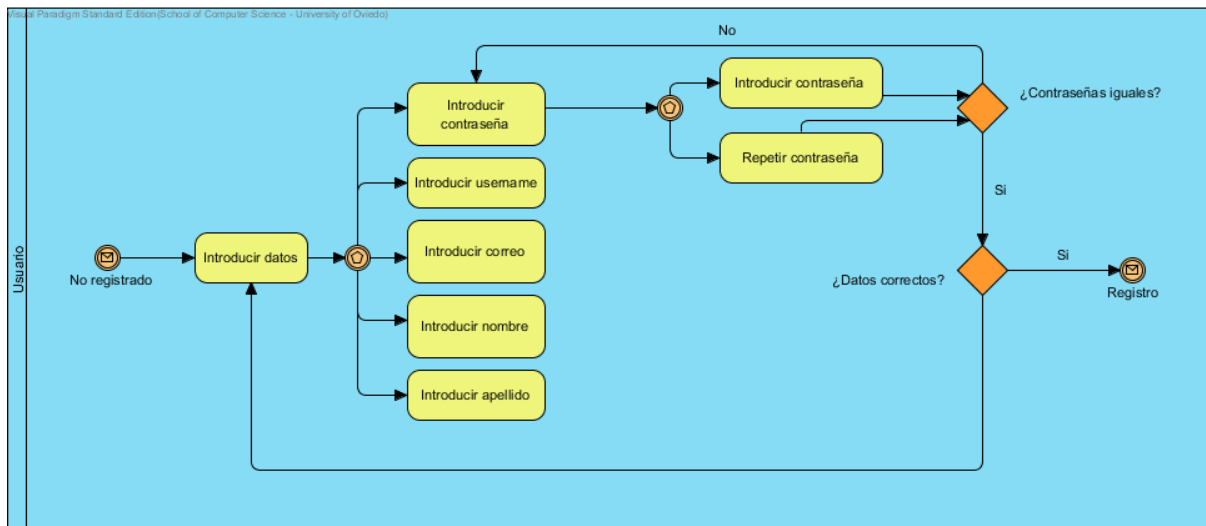
Juego




Resumen




Nombre	Documentación
 Juego	
 Partida iniciada	
 Tirar dado	
 Mover izquierda	
 Mover derecha	
 Contestar pregunta	
 ¿Pregunta acertada?	
 Quesito ganado	
 ¿Partida finalizada?	

Registrarse



Resumen

Nombre	Documentación
 Usuario	
 No	

registrado	
<input type="checkbox"/> Introducir datos	
<input type="checkbox"/> Introducir username	
<input type="checkbox"/> Introducir contraseña	
<input type="checkbox"/> Repetir contraseña	
<input type="checkbox"/> Introducir contraseña	
 ¿Contraseñas iguales?	
 ¿Datos correctos?	
 Registro	
<input type="checkbox"/> Introducir correo	
<input type="checkbox"/> Introducir nombre	
<input type="checkbox"/> Introducir apellido	