

2015

Versión 3.0

Manual E3



- *Marcos Abdel-Fattah Martínez* - (UO230956)
- *Carmen Peñalver San Cristóbal* - (UO229705)
- *Pablo Bravo Mediavilla* - (UO223531)
- *Álvaro Panizo Romano* - (UO227748)
- *Javier Iglesias García* - (UO232562)
- *Pelayo Gómez Fanjul* - (UO221205)
- *Raúl Herrador Colino* - (UO234551)
- *Liliana Villar Iglesias* - (UO232510)
- *Iñigo Llaneza Aller* - (UO206367)

Grupo Trivial 3a

UNIVERSIDAD DE OVIEDO - Arquitectura del Software



Contenido

Visión general.....	3
Manual de usuario	4
Usuario cliente	4
Usuario administrador	12
Manual de sistema	16
Paquete business	16
Paquete core	16
Paquete game	16
Clase IniciarJuegoAction.....	16
Clase RegistrarseAction.....	16
Clase ValidarseAction	16
Clase Action	16
Clase Partida.....	17
Paquete Tablero	17
Paquete colores.....	17
Paquete jugadores	17
Clase jugador	17
Paquete nodos	17
Clase NodoFinal.....	17
Clase NodoQuesito	17
Clase NodoNormal.....	17
Paquete tableros	17
Clase TableroCircular.....	18
Clase TableroLineal.....	18
Clase TableroRueda	18
Paquete Util.....	18
Clase LectorCircular	18
Clase LectorFormaRueda.....	18
Clase ContenedorDeNodos	18
Clase PreguntasAleatorias	18

Clase SelectorDeColores.....	18
Paquete modelo	18
Clase Pregunta.....	18
Clase Usuario.....	18
Paquete persistencia	19
Clase Connection.....	19
Clase Insert.....	19
Clase Estadísticas.....	19
Clase GetPreguntas	19
Clase GetUsuario	19
Clase GetUsuarios	19
Paquete controllers	19
Clase Application	19
Paquete views	20
Game.scala.html.....	20
Index.scala.html	20
Menu_admin.scala.html.....	20

Visión general

En este manual, se describe el funcionamiento del módulo, que permite a un usuario, registrarse en el juego, iniciar una partida de trivial seleccionando el número de jugadores con los que desea jugar y consultar sus estadísticas e información de los jugadores. Nos centraremos en dos tipos diferentes de visiones:

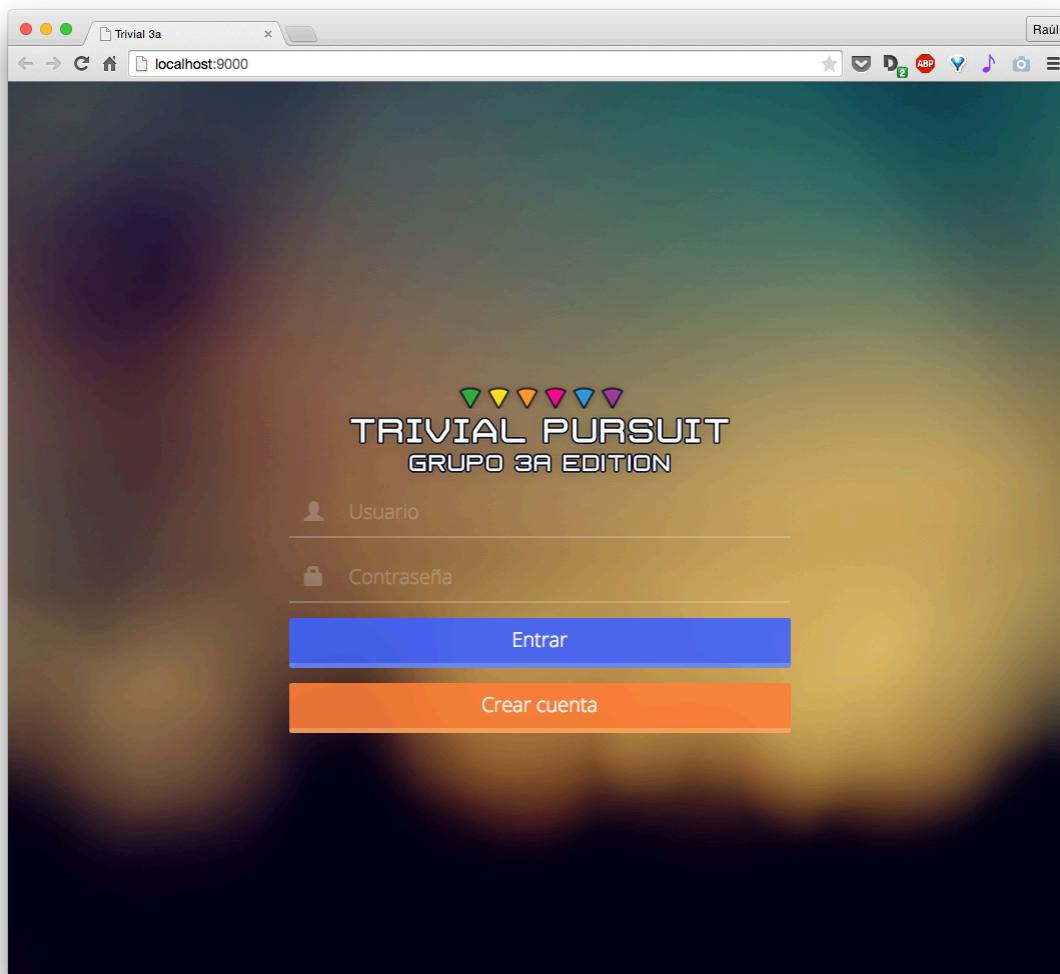
1. **A nivel de usuario:** esta parte de la aplicación está pensada para un usuario cliente haga uso de ella. Se explicará cómo se realiza la interacción con el juego.
2. **A nivel de sistema:** instrucciones para la utilización de este código por distintos desarrolladores. Gracias a éste se permitirá una lectura rápida que nos ayudará a entender el código y funcionamiento del proyecto para su utilización y modificación posterior. Se explicará detalladamente la funcionalidad de las diversas clases que conforman el proyecto. El sistema está formado por una serie de jerarquías de clases que cooperan entre sí para lograr el correcto funcionamiento de la aplicación.

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...

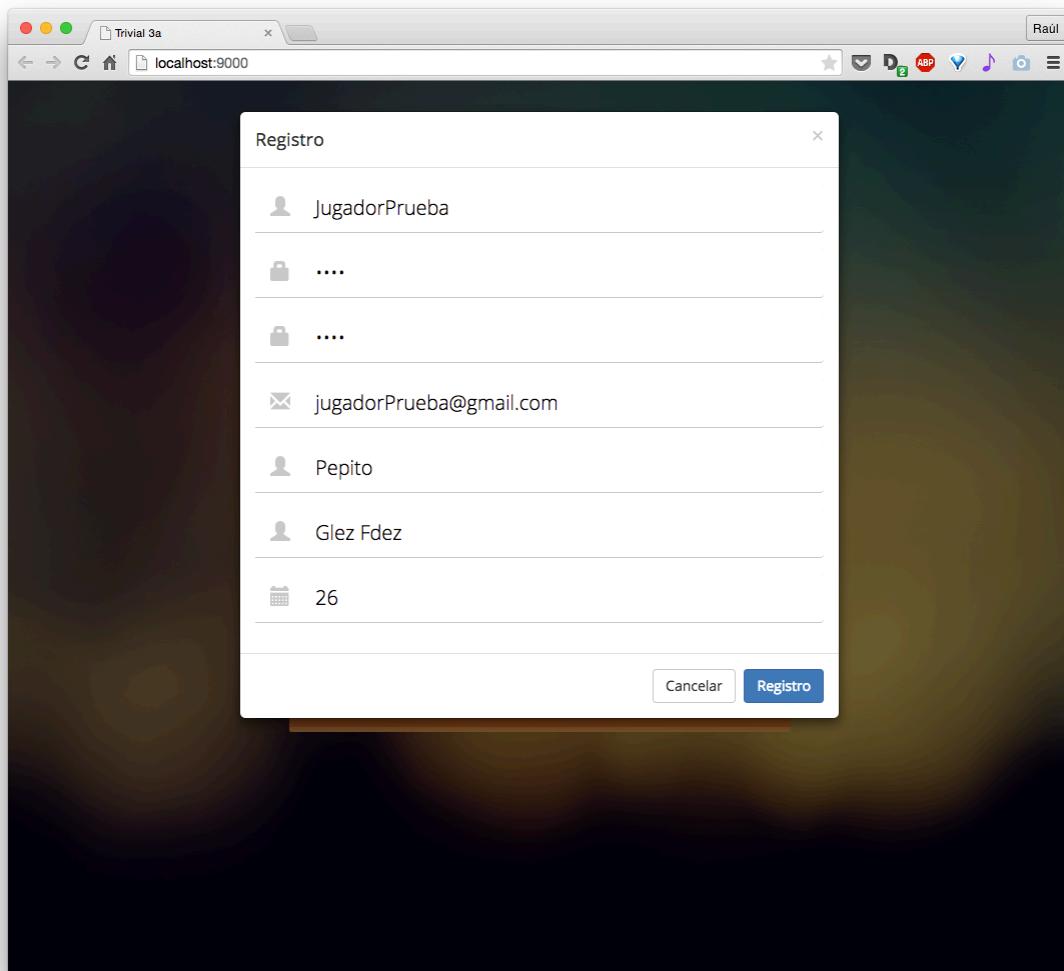
Usuario cliente

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



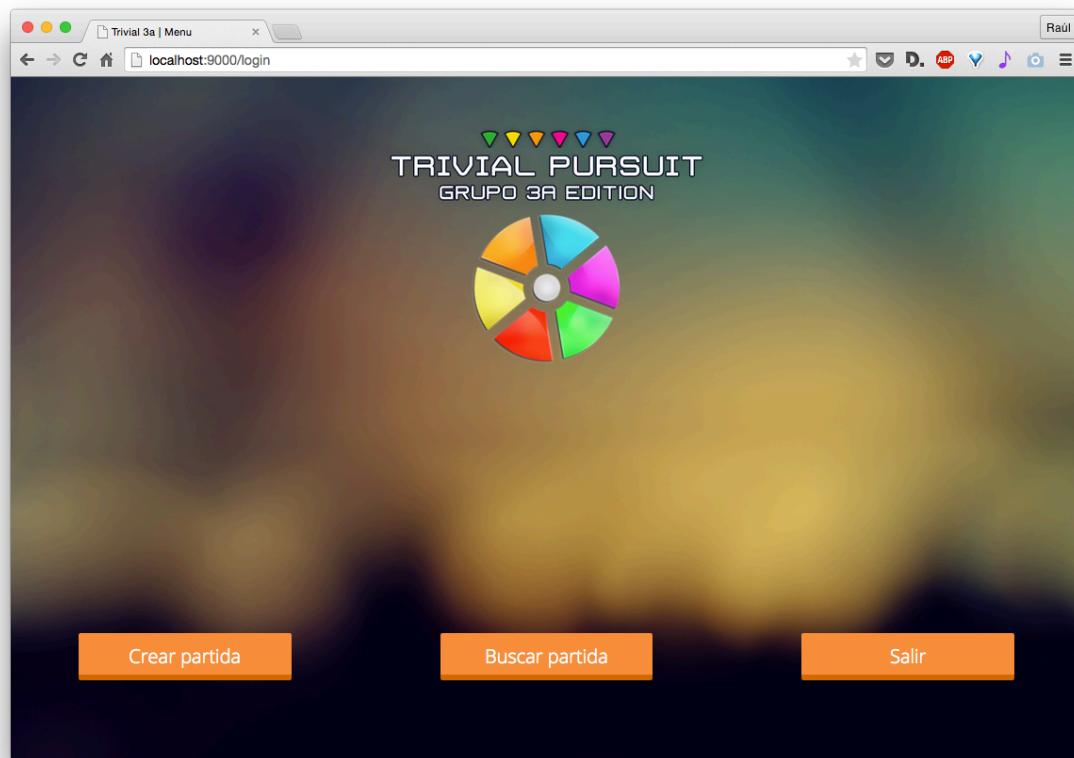
En ella, tenemos dos alternativas.

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



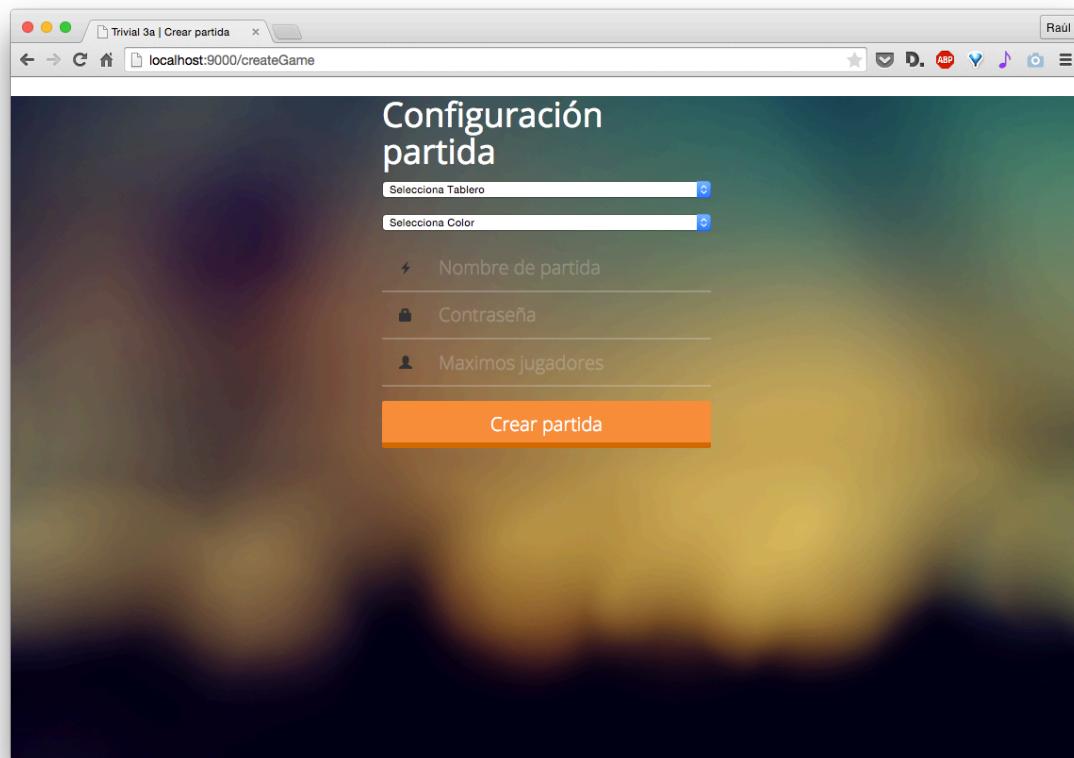
En este formulario se nos pedirá que introduzcamos datos del usuario: nombre de usuario, contraseña (por duplicado), dirección de correo electrónico, nombre, apellidos y edad. Pinchamos sobre el botón “registro” y si los datos introducidos no se corresponden con los de ningún otro jugador ya registrado,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

- a. 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:
- b. Buscar partida: para buscar una partida que otro usuario tenga iniciada y unirse a ella.
- c. 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 queramos ser (rojo, amarillo, verde...), a continuación introduciremos una serie de datos, el nombre de la partida, una contraseña y el máximo de jugadores que queramos 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.

localhost:9000/game

Turno de:raulhc

Chat

SERVIDOR: Respuesta correcta, sigue jugando



2 ▲ ↻ ↺ ↻ ↺ 10 6 2 9

Â Que nota se merece este trabajo?

localhost:9000/game

Turno de:raulhc

Chat

SERVIDOR: Respuesta correcta, sigue jugando

SERVIDOR: Respuesta incorrecta, turno de raulhc

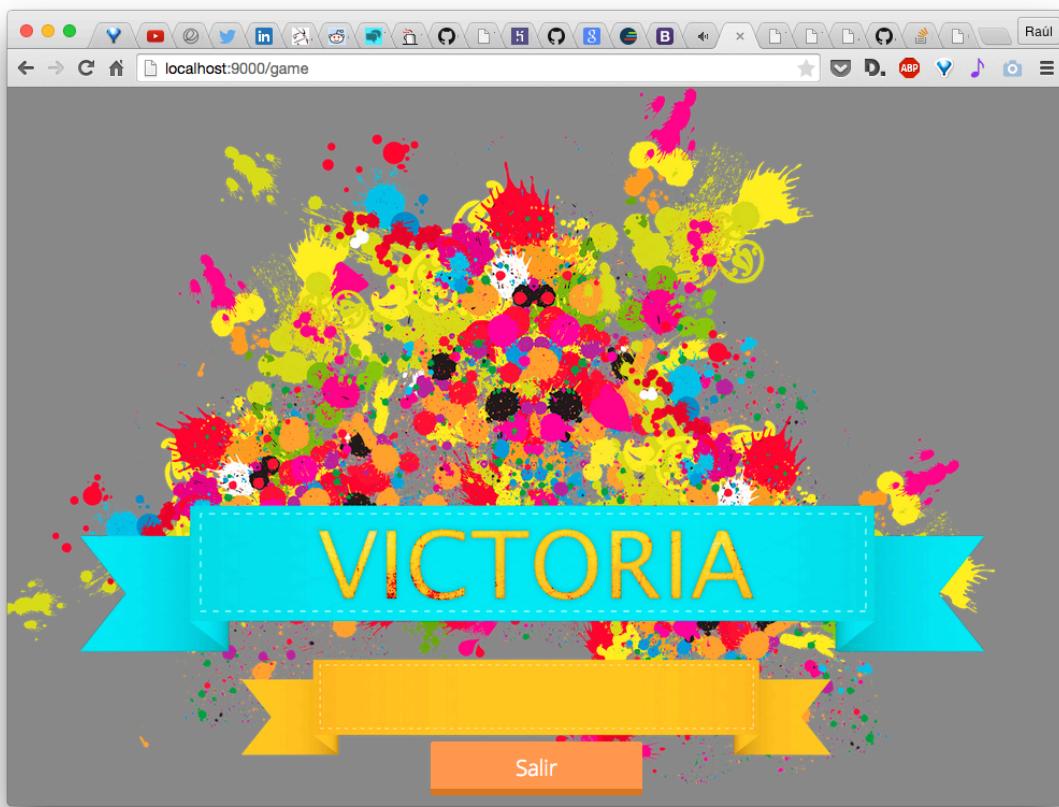
SERVIDOR: Respuesta correcta, sigue jugando



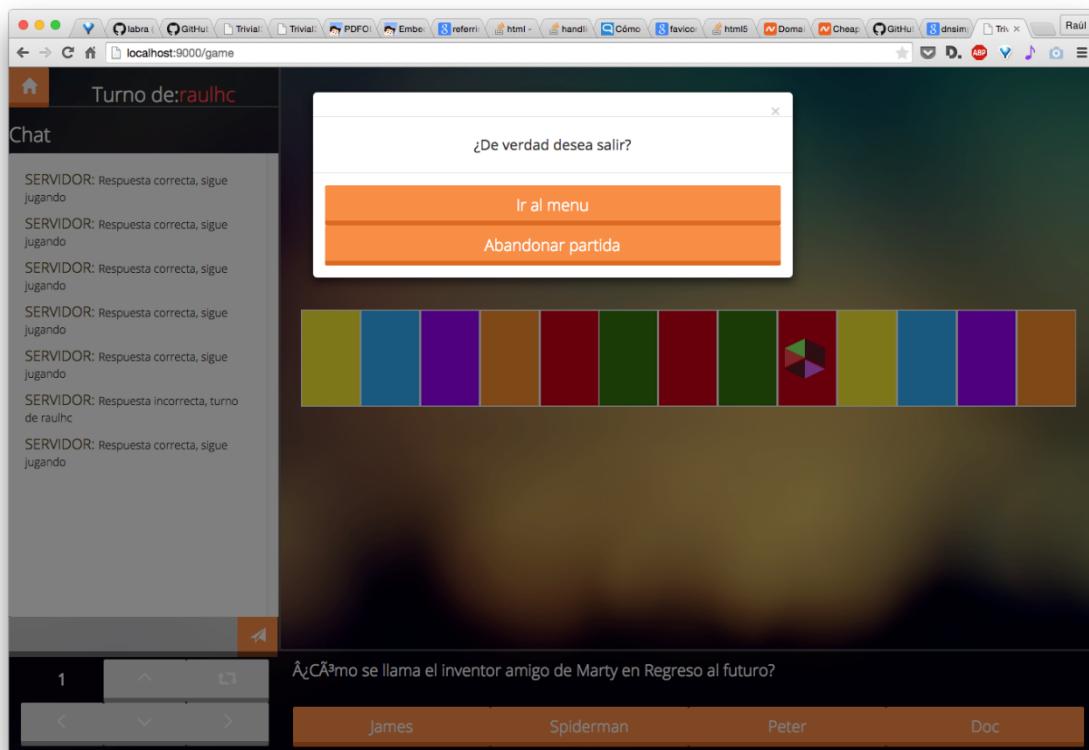
1 ▲ ↻ ↺ ↻ ↺ James Spiderman Peter Doc

Â CÃ³mo se llama el inventor amigo de Marty en Regreso al futuro?

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



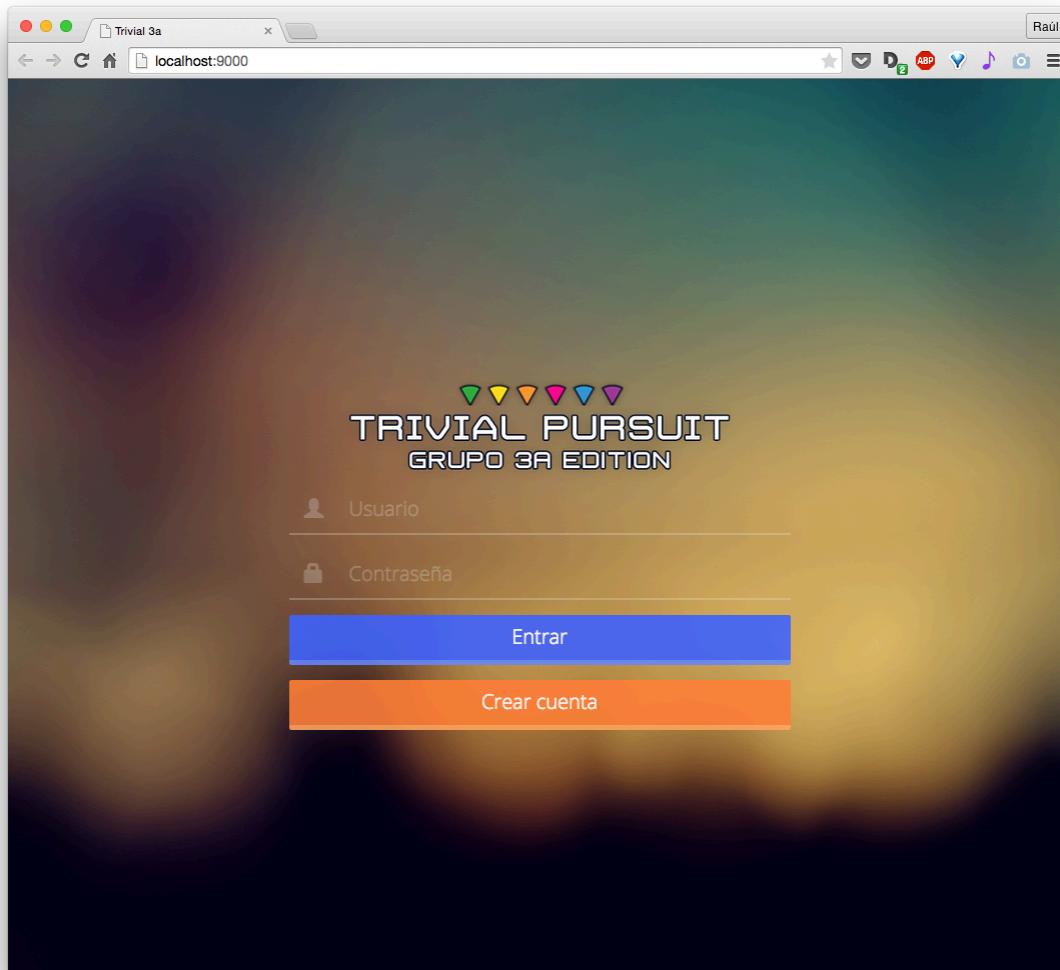
Si queremos salir de la partida, pincharemos sobre el botón con una casita y nos aparecerán dos opciones:



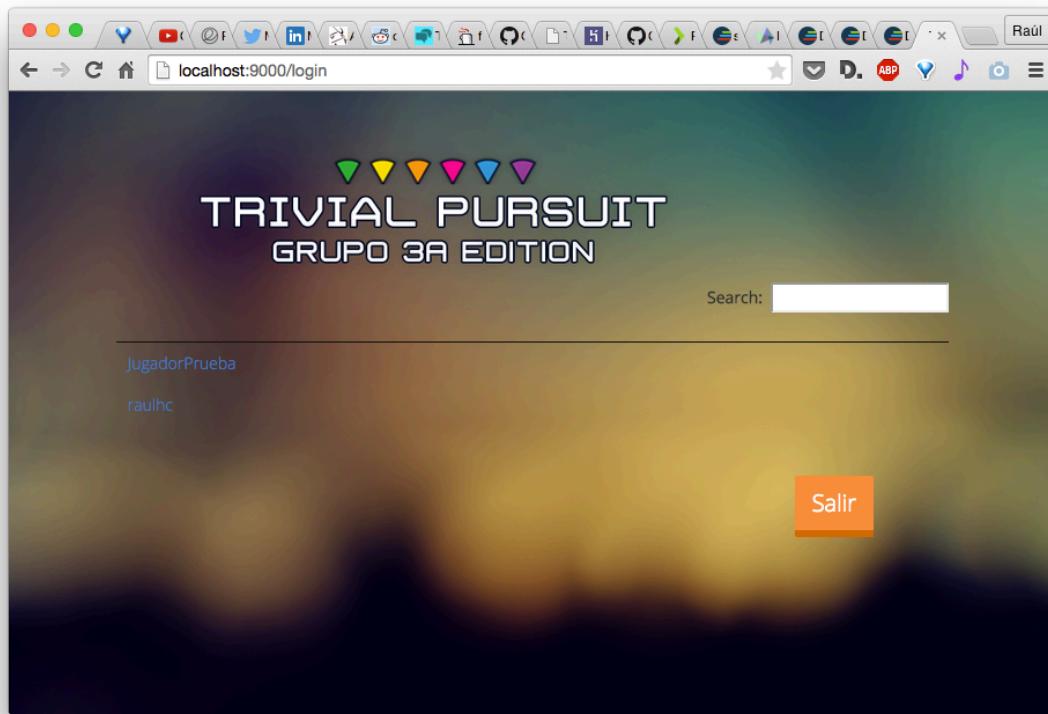
Podemos volver al menú para iniciar otra partida, o finalizar el juego.

Usuario administrador

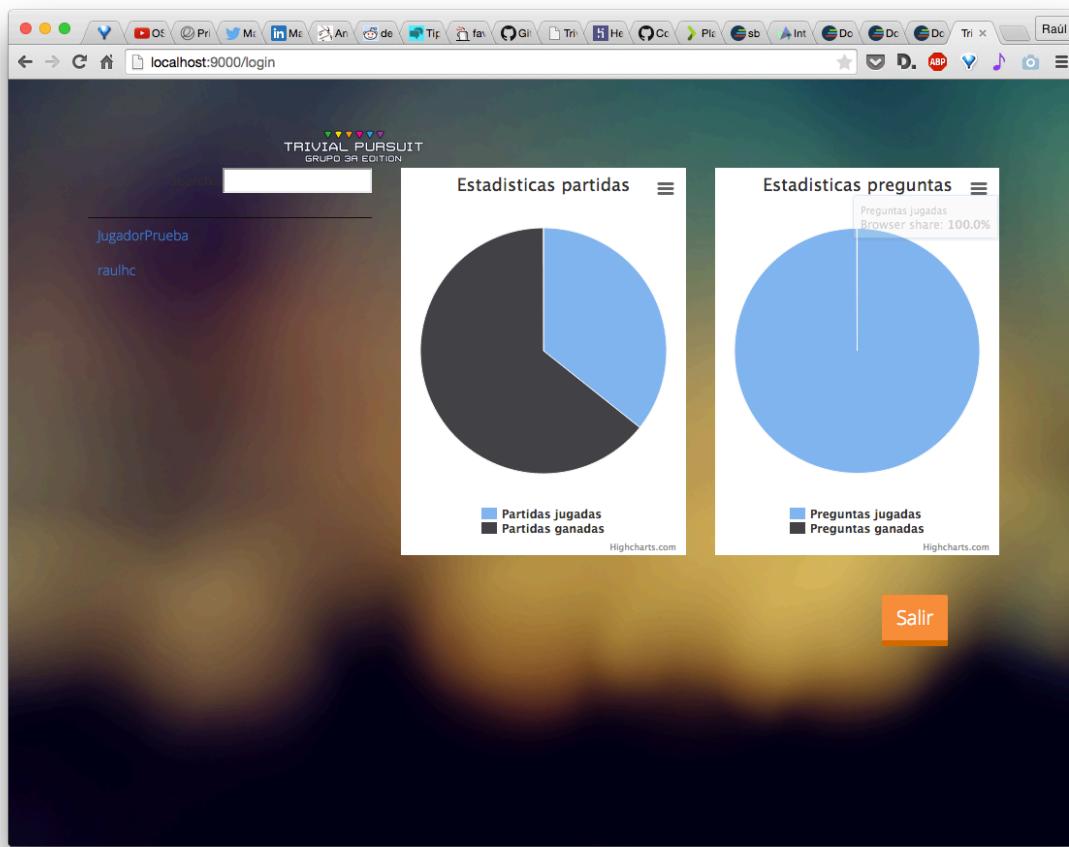
El usuario administrador, deberá iniciar sesión en la ventana principal, para ello, introducirá su nombre de usuario y contraseña.

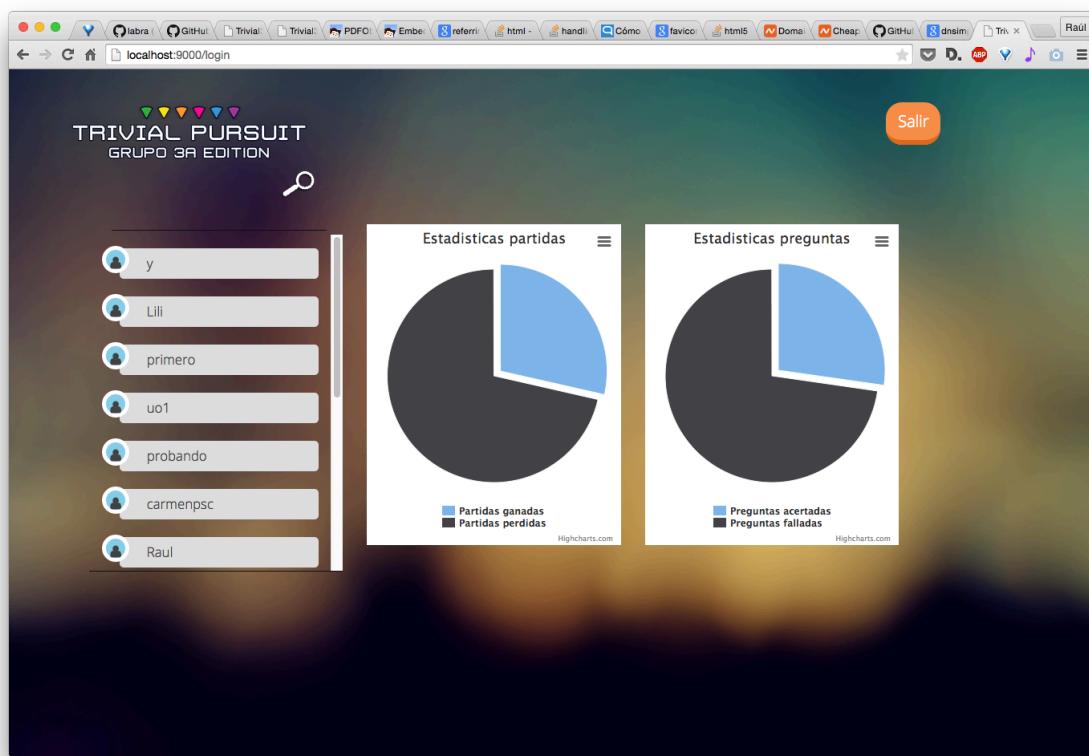


Una vez introducidos los datos, se nos abrirá una nueva ventana:



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

Este 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 pivot 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 llenar 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.