Trivial4a

Escuela de Ingeniería Informática - Universidad de Oviedo

Grado en Ingeniería Informática del Software

Índice de contenidos

[Hoja de identificación 3](#_Toc418680577)

[Introducción 4](#_Toc418680578)

[Planteamiento del problema 4](#_Toc418680579)

[Metodología usada 5](#_Toc418680580)

[Metodología de trabajo 5](#_Toc418680581)

[Identificación de Interesados 5](#_Toc418680582)

[Descripción de los interesados 5](#_Toc418680583)

[Usuarios 5](#_Toc418680584)

[Administradores 5](#_Toc418680585)

[Responsables de NoGame 5](#_Toc418680586)

[Desarrolladores 5](#_Toc418680587)

[Roles e historias de usuario 6](#_Toc418680588)

[Roles 6](#_Toc418680589)

[Historias de usuarios 6](#_Toc418680590)

[Identificación de atributos de calidad 6](#_Toc418680591)

[Lista de atributos de calidad e interesados 7](#_Toc418680592)

[Solución planteada 8](#_Toc418680593)

[Módulo 1 – Extract 8](#_Toc418680594)

[Módulo 2 – Core 9](#_Toc418680595)

[Módulo 3 – Trivial web 9](#_Toc418680596)

[Escenarios de calidad 10](#_Toc418680597)

[Vistas 13](#_Toc418680598)

[Vistas de sistema 13](#_Toc418680599)

[Diagrama de componentes 14](#_Toc418680600)

[Componente Base de datos 15](#_Toc418680601)

[Componente Trivial4a 16](#_Toc418680602)

[Componente Cometidos 16](#_Toc418680603)

[Componente Procesos aplicación 17](#_Toc418680604)

[Componente Proceso Login 17](#_Toc418680605)

[Componente Proceso gestión de datos 17](#_Toc418680606)

[Componente Interfaz web 18](#_Toc418680607)

[Componente Procesamiento de sesión 18](#_Toc418680608)

[Componente Procesamiento de datos 18](#_Toc418680609)

[Diagrama de paquetes 19](#_Toc418680610)

[Anexos 22](#_Toc418680611)

[Diagrama BPMN 22](#_Toc418680612)

[Índice de figuras 23](#_Toc418680613)

[Índice de tablas 23](#_Toc418680614)

# Hoja de identificación

|  |  |
| --- | --- |
| Nombre del trabajo: | Trivial4a |
| Fecha: | Mayo 2015 |
| Autores: | |  |  | | --- | --- | | Nombre | UO | | Jorge Sierra Miranda | 226566 | | David Sariego Fernández | 223935 | | Santiago Marqués Rivera | 225086 | | Lucía Sánchez Díaz | 205169 | | Cristina Montes Rubio | 226259 | | Ángel Rodríguez Bobes | 214459 | | Pablo Romero Martín | 221220 | |  |  | |

# Introducción

La arquitectura descrita en este documento pretende dar cabida al desarrollo de una plataforma de juego. Esta plataforma pretende imitar al juego “Trivial” además de ofrecer funcionamiento adicional.

En el documento se describirá la arquitectura del sistema comparándolo con los patrones arquitectónicos conocidos resaltando aquellas características que se hayan intentado obtener con dichos patrones.

# Planteamiento del problema

NoGame, una empresa del sector de los videojuegos está interesada en la creación de una aplicación con un funcionamiento similar al juego de mesa “Trivial”.

La aplicación no será sólo una plataforma de juego, sino que deberá además guardar registros de las actividades de los jugadores.

Debido a la naturaleza del juego las preguntas deberán poder añadirse, quitarse, y cambiarse; permitiendo jugar cada vez con un set de preguntas distinto.

Adaptándose a la las nuevas tecnologías el sistema estará disponible para jugar en aplicaciones de escritorio o a través de la web.

Por último, los datos recopilados por la información deber estar disponibles para los administradores del sistema. Esta funcionalidad permitirá la reutilización de la plataforma en entornos profesionales o educacionales para la realización de controles o estadísticas de personal.



Ilustración - Diagrama de contexto

# Metodología usada

El estudio de la arquitectura se realiza siguiendo el método de ADD (Atribute-Driven Design) (Bass, Clements, & Kazman, Software Architecture in Practice, Second Edition, 2003).

## Metodología de trabajo

La metodología para el desarrollo de la aplicación se basa en el reparto de tareas.

Las tareas repartidas se podrán repartir a más de una persona creando la posibilidad de dividir la tarea en subtareas o de realizar la tarea conjuntamente con técnicas como *pair porgramming*.

La comunicación del equipo se realizará presencialmente en el centro, a partir de mensajería instantánea en Skype y a través de Github, aprovechando todas las herramientas disponibles en este.

El estilo de desarrollo del software ha sido estilo incremental (*Incremental piecemeal*) por la manera en la que se nos ha entregado a los alumnos los enunciados.

# Identificación de Interesados

Los interesados en la aplicación, también llamados *stakeholders* son los siguientes:

* Usuarios
* Administradores
* Desarrolladores

## Descripción de los interesados

Se tendrán en cuenta las distintas características de los *stakeholders* para el desarrollo del sistema. Cada tipo de *stakeholder* tendrá unos objetivos distintos en la aplicación.

### Usuarios

Son los que utilizarán el sistema participando en juegos y respondiendo preguntas.

Deberán estar registrados en el sistema.

Se guardarán los datos de cómo estos usuarios interactúan con el sistema.

### Administradores

Son los usuarios encargados del sistema que tendrán acceso a los datos recopilados por la aplicación.

### Responsables de NoGame

NoGame como empresa contratante es un interesado en la aplicación.

No participarán directamente en la aplicación pero sus necesidades son aquellas que la aplicación intenta cubrir.

### Desarrolladores

Son aquellas personas designadas al desarrollo de la aplicación.

Interactuarán con el sistema en la fase de desarrollo y tendrán como objetivo crear un sistema que sea fácil de mantener, con un coste reducido y abierto para la extensión de su funcionalidad.

# Roles e historias de usuario

## Roles

Los roles en el sistema se corresponden con dos tipos de interesados.

* Usuarios
* Administradores

## Historias de usuarios

|  |
| --- |
| Como administrador quiero poder añadir preguntas a la aplicación desde ficheros GIFT. |
| Como administrador quiero poder ver las estadísticas de todos los usuarios. |
| Como administrados quiero poder modificar los perfiles de otros usuarios. |
| Como usuario quiero poder identificarme para utilizar la plataforma. |
| Como usuario quiero poder cerrar sesión. |
| Como usuario quiero poder ver mis estadísticas. |
| Como usuario quiero poder empezar una partida. |
| Como usuario quiero poder cambiar mis datos en la aplicación. |

Tabla - Historias de usuario

# Identificación de atributos de calidad

Los atributos de calidad identificados para este sistema son:

#### Disponibilidad

El sistema debe poder estar disponible las 24 del día.

#### Modificabilidad

El sistema tiene que poder sufrir cambios como nuevos ficheros fuente, nuevos formatos de representación y posibles cambios de base de datos sin que estos resulten costosos.

#### Escalabilidad

El incremento del número de usuarios y de las preguntas almacenadas no debe suponer cambio alguno para la aplicación.

#### Rendimiento

Las acciones interactivas realizadas sobre el sistema deben ser instantáneas a la vista de los usuarios.

#### Seguridad

Los accesos a la aplicación deben ser controlados con cuentas de usuario y contraseña.

Se debe asegurar la integridad de los datos almacenados.

#### Testeabilidad

El funcionamiento de la aplicación se debe probar con facilidad.

#### Usabilidad

La aplicación debe ser amigable con el usuario, es decir, fácil de usar y entender.

# Lista de atributos de calidad e interesados

|  |  |  |
| --- | --- | --- |
| CÓDIGO | STAKEHOLDER | INTERESES |
| S1 | Usuario | * Facilidad de uso * Rapidez en la interacción |
| S2 | Administrador | * Facilidad de uso * Seguridad de acceso |
| S3 | Responsable de NoGame | * Bajo coste de desarrollo * Aplicación resultante correcta * Entrega dentro del plazo * Fácilmente ampliable |
| S4 | Desarrollador | * Aplicación resultante correcta * Facilidad para mantener el sistema |

Tabla - Stakeholders

|  |  |  |
| --- | --- | --- |
| CÓDIGO | DESCRIPCIÓN | TIPO DE ATRIBUTO |
| A1 | El sistema debe poder estar disponible las 24 del día | Disponibilidad |
| A2 | El sistema tiene que poder sufrir cambios como nuevos ficheros fuente, nuevos formatos de representación y posibles cambios de base de datos sin que estos resulten costosos. | Modificabilidad |
| A3 | El incremento del número de usuarios y de las preguntas almacenadas no debe suponer cambio alguno para la aplicación. | Escalabilidad |
| A4 | Las acciones interactivas realizadas sobre el sistema deben ser instantáneas a la vista de los usuarios. | Rendimiento |
| A5 | Los accesos a la aplicación deben ser controlados con cuentas de usuario y contraseña | Seguridad |
| A6 | Se debe asegurar la integridad de los datos almacenados. | Seguridad |
| A7 | El funcionamiento de la aplicación se debe probar con facilidad | Testeabilidad |
| A8 | La aplicación debe ser amigable con el usuario, es decir, fácil de usar y entender | Usabilidad |

Tabla - Atributos de calidad

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | S1 | S2 | S3 | S4 |
| A1 | √ | √ | √ |  |
| A2 |  |  | √ | √ |
| A3 |  | √ | √ | √ |
| A4 | √ | √ |  |  |
| A5 |  | √ | √ |  |
| A6 |  | √ |  | √ |
| A7 |  |  |  | √ |
| A8 | √ |  |  |  |

Tabla - Relacion Steakeholder/Atributo de calidad

# Solución planteada

La solución planteada para los requisitos del sistema es la realización de una aplicación basada en tres módulos relacionados.

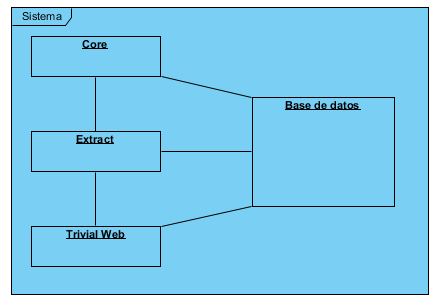


Ilustración - Solución planteada

Los módulos comparten una única base de datos de forma similar a estilo arquitectónico de **base de datos compartida**, obteniendo de esta manera algunas de las ventajas de dicho estilo. Sin embargo los módulos de este sistema no son independientes lo que desvirtúa el patrón.

## Módulo 1 – Extract

Este módulo es el inicial del sistema. Desde este módulo se añaden preguntas a la base de datos desde ficheros en formato GIFT. Es decir, es un parser que lee ficheros y actualiza la base de datos. También se puede usar para escribir ficheros intermedios para el sistema.

El diseño de software interno consiste en la separación de responsabilidades. Independientemente de su diseño interno, este módulo tiene el comportamiento de una aplicación secuencial o batch. Por las características de este módulo y las necesidades que debe cubrir este comportamiento es el óptimo.

## Módulo 2 – Core

Este módulo consiste en una aplicación de escritorio dónde uno o varios usuarios pueden identificarse para participar en un juego. Al finalizar la partida los resultados de la partida son almacenados en la base de datos.

Este módulo se corresponde con el patrón *modelo vista controlador* (**MVC)**.Este patrón cubre todas las necesidades de este módulo, adaptándose al modelo de dominio, permitiendo distintas representaciones (vistas) del juego a la vez que controla el desarrollo de la partida.

## Módulo 3 – Trivial web

Este módulo es más interesante de la aplicación y el que más se explotará una vez se desarrolle la plataforma.

Este módulo permite jugar partidas personales a partir de un navegador web. Además dispone de vistas para los datos que almacena la aplicación para que los usuarios vean sus propios datos y los administradores controlen los datos de todos.

Para cubrir todas las necesidades y requisitos del sistema este módulo se desarrolla con el Framework Play. Este framework permite crear aplicaciones web basadas en el **MVC**. Al basarse en **MVC** tiene todas las ventajas que este patrón ofrece. Además al ser una aplicación web puede instalarse en un servidor, permitiendo el estilo de integración Cliente-Servidor. Cualquier cliente puede conectarse a un servidor que tenga la plataforma Trivial4a instalada y participar en juegos o revisar los datos.

# Escenarios de calidad

| Código | Fuente del estímulo | Estímulo | Entorno | Artefacto | Respuesta | Medición de la Respuesta | Atributo afectado |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 01 | Usuarios del sistema | Poder jugar o jugar sin interrupción una partida | Explotación | Aplicación Web y Base de datos | Aplicación disponible | Todas las partes del sistema den respuesta al usuario | A1 |
| 02 | Incrementar los datos del sistema | Obtener nuevos datos | Comunicación con el servidor | Extractor | Procesamiento de nuevas preguntas y/o categorías | Las preguntas son correctamente guardadas con su categoría y respuestas | A2 |
| 03 | Actualizar o modificar ficheros, formatos o base de datos | Actualizar los datos existente o los datos que se guardan sobre ellos | Comunicación con el servidor | Extractor y base de datos | Procesamiento de nuevas preguntas y/o categorías  Y los datos sobre estas | Las preguntas y sus datos son correctamente guardadas | A2 |
| 04 | Crecimiento del número de usuarios y/o preguntas en el sistema | Crecimiento en la cantidad de accesos al portal y de la comunicación de este con la base de datos | Explotación | Administración de sesión y procesamiento de datos | Disponer de servidores adicionales que respondan antes la demanda de usuarios que accedan al portal | La página siga en pie ante un gran número de peticiones | A3 |
| 05 | Usuarios que utilizan la aplicación | Jugar una partida online | Explotación | Base de datos y sistema de administración de preguntas y respuestas | Tiempo de respuesta mínimo | Tiempo de procesamiento < 2 seg | A4 |
| 06 | Actualización de las preguntas y datos | Añadir/actualizar preguntas y/o respuestas | Explotación | Base de datos y Extractor | Tiempo de respuesta mínimo | Tiempo de procesamiento < 1 min | A4 |
| 07 | Usuario con privilegios consulta estadísticas | Realizar consulta de las estadísticas del juego | Explotación | AdminView | Tiempo de respuesta mínimo | Tiempo de procesamiento < 5 seg | A4 |
| 08 | Seguridad en las conexiones de usuarios | Conexiones seguras de los usuarios | Explotación | Loggin | Inicio de sesión o registro | No poder acceder a la cuenta de un usuario sin su contraseña correcta | A5 |
| 09 | Discriminación de tipos de usuarios | Diferir entre administrador y otros usuarios | Explotación | Loggin | Inicio de sesión del rol correcto | Ningún usuario normal tiene acceso a funciones de administrador | A5 |
| 10 | Integridad de los datos | Integridad de las preguntas y respuestas almacenadas | Explotación | Extractor y Gestor de Datos | Respuestas correctas a cada pregunta | Obtener siempre las misma/a respuesta/s correcta/s a las preguntas | A6 |
| 11 | Integridad de los datos procesados | Integridad de las respuestas guardadas | Explotación | GameObject | Almacenamiento correcto de los datos de cada respuesta del usuario | Reflejado en base de datos pregunta, respuesta, si correcta y fecha correctas de cada respuesta proporcionada | A6 |
| 12 | Estabilidad para la aplicación | Estabilidad | Desarrollo | Sistemas y equipos de desarrollo | Aplicación robusta | Sin fallos inesperados no controlados | A7 |
| 13 | Testing del equipo de desarrollo | Casos de prueba | Desarrollo | Sistemas, equipos de desarrollo y procesamiento de datos | Preparar y realizar periódicamente pruebas de caso de todos los niveles | Tras cada fase superada del proyecto | A7 |
| 14 | Aplicación fácil de usar | Usabilidad | Explotación | Interfaz de configuración y juego | Interfaz usable | <5% de rechazo | A8 |
| 15 | Aplicación accesible | Accesibilidad | Explotación | Controlador de peticiones de usuario | Interfaz accesible | Chequeado desde W3C cumpliendo más de AA | A8 |

Tabla - Escenarios de calidad

# Vistas

## Vistas de sistema





Ilustración - Diagrama del sistema

|  |  |
| --- | --- |
| Elemento del  sistema | Descripción |
| Usuario | Usuarios de la aplicación (jugadores) que interactuarán constantemente con el Servidor Trivial4a para registrarse y acceder a sus cuentas |
| Servidor Trivial4a | Subsistema encargado de la interacción con el Usuario y el Servidor Trivial4a. |
| MongoDB | Base de datos en la que se almacenará toda la información que será utilizada por la aplicación de trivial (cuentas, preguntas y estadísticas) o que será obtenida a través de la misma (registro). |

Tabla - Elementos del sistema

## Diagrama de componentes

## C:\Users\Cris\Desktop\Captura.PNG

Ilustración - Diagrama de componentes (I)

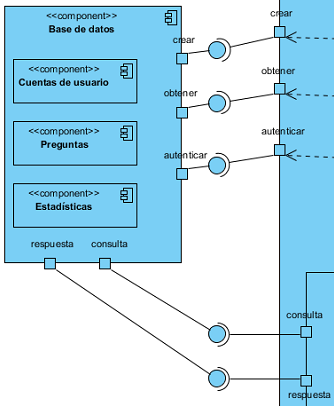


Ilustración - Diagrama de componentes (II)

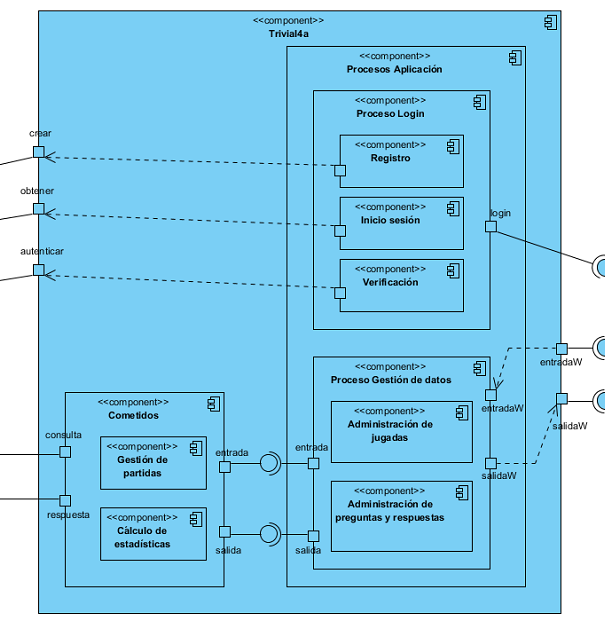


Ilustración - Diagrama de componentes (III)

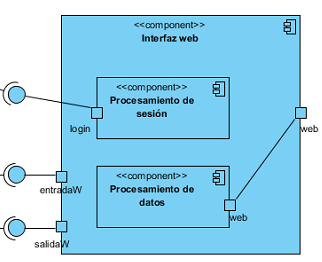


Ilustración - Diagrama de componentes (IV)

### Componente Base de datos

Contiene los datos que han sido tanto proporcionados por los usuarios como generados por el juego.

|  |  |
| --- | --- |
| Componentes | Descripción |
| Cuentas de usuario | Representa la información contenida en la base de datos acerca de las cuentas de los usuarios registrados. |
| Preguntas | Representa la información contenida en la base de datos acerca de los bancos de preguntas utilizadas. |
| Estadísticas | Representa la información estadística contenida en la base de datos generadas a partir de las partidas jugadas por cada jugador. |

Tabla - Componentes Base de datos

|  |  |
| --- | --- |
| Puertos | Descripción |
| Crear | Añadir un nuevo usuario a la base de datos. |
| Obtener | Obtener un usuario ya registrado en la base de datos. |
| Autenticar | Verificar la identidad del usuario contrastando la información de su cuenta en la base de datos. |
| Consulta | Recibimiento de información a la base de datos. |
| Respuesta | Salida de información procedente de la base de datos. |

Tabla - Puertos Base de datos

### Componente Trivial4a

En él se llevan a cabo todos los procesos referentes a la gestión central de datos y las funcionalidades principales del sistema.

|  |  |
| --- | --- |
| Componentes | Descripción |
| Cometidos | Engloba las funcionalidades principales del juego. |
| Procesos aplicación | Engloba todo lo referente a la gestión de procesamiento de datos de la web. |

Tabla - Componentes Trivial4a

|  |  |
| --- | --- |
| Puertos | Descripción |
| Crear | Añadir un nuevo usuario a la base de datos. |
| Obtener | Obtener un usuario ya registrado en la base de datos. |
| Autenticar | Verificar la identidad del usuario contrastando la información de su cuenta en la base de datos. |
| EntradaW | Recibimiento de información procedente de datos obtenidos de la web. |
| SalidaW | Salida de información encaminada a la web. |

Tabla - Puertos Trivial4a

### Componente Cometidos

Engloba las funcionalidades principales del juego.

|  |  |
| --- | --- |
| Componentes | Descripción |
| Gestión de partidas | Controla los eventos sucesivos producidos durante el transcurso de una partida. |
| Cálculo de estadísticas | Recopila información acerca de los resultados obtenidos durante el transcurso de la partida para su posterior tratamiento. |

Tabla - Componentes Cometidos

|  |  |
| --- | --- |
| Puertos | Descripción |
| Consulta | Recibimiento de información a la base de datos. |
| Respuesta | Salida de información procedente de la base de datos. |
| Entrada | Conjunto de datos de entrada procesados durante una partida. |
| Salida | Conjunto de datos de salida procesados durante una partida. |

Tabla - Puertos Cometidos

### Componente Procesos aplicación

Gestor del procesamiento de datos de la aplicación.

|  |  |
| --- | --- |
| Componentes | Descripción |
| Proceso login | Inicio de sesión en la aplicación. |
| Proceso gestión de datos | Procesamiento de datos generados en cada evento de una partida. |

Tabla - Componentes Procesos aplicación

### Componente Proceso Login

Gestiona el inicio de sesión o registro por parte de los usuarios.

|  |  |
| --- | --- |
| Componentes | Descripción |
| Registro | Registra un nuevo usuario. |
| Inicio sesión | Inicia la sesión de un usuario. |
| Verificación | Encargado de verificar los datos de sesión aportados por los usuarios. |

Tabla - Componentes Proceso login

|  |  |
| --- | --- |
| Puertos | Descripción |
| Login | Proceso de inicio de sesión. |

Tabla - Puertos Proceso login

### Componente Proceso gestión de datos

Procesamiento de datos generados en cada evento de una partida.

|  |  |
| --- | --- |
| Componentes | Descripción |
| Administración de jugadas | Tratamiento de datos relativos a las jugadas realizadas por cada jugador a lo largo de una partida. |
| Administración de preguntas y respuestas | Proporciona y registra las preguntas y respuestas utilizadas durante el transcurso de una partida. |

Tabla - Componentes Proceso gestión de datos

|  |  |
| --- | --- |
| Puertos | Descripción |
| Entrada | Conjunto de datos de entrada procesados durante una partida. |
| Salida | Conjunto de datos de salida procesados durante una partida. |
| EntradaW | Recibimiento de información procedente de datos obtenidos de la web. |
| SalidaW | Salida de información encaminada a la web. |

Tabla 17 - Puertos Proceso gestión de datos

### Componente Interfaz web

Se encarga de la recolección de datos para su posterior visualización, así como del mantenimiento activo constante de la sesión actual de un usuario.

|  |  |
| --- | --- |
| Componentes | Descripción |
| Procesamiento de sesión | Mantenimiento de la sesión. |
| Procesamiento de datos | Recogida y visualización de los datos. |

Tabla 18 - Componentes Interfaz web

|  |  |
| --- | --- |
| Puertos | Descripción |
| EntradaW | Recibimiento de información procedente de datos obtenidos de la web. |
| SalidaW | Salida de información encaminada a la web. |

Tabla 19 - Puertos Interfaz web

### Componente Procesamiento de sesión

Mantenimiento activo de la sesión de un usuario.

|  |  |
| --- | --- |
| Puertos | Descripción |
| Login | Datos necesarios para el inicio de sesión o el registro de usuarios. |

Tabla 20 - Componentes Procesamiento de sesión

### Componente Procesamiento de datos

Se encarga de la recolección de datos para su posterior visualización.

|  |  |
| --- | --- |
| Puertos | Descripción |
| Web | Obtención de datos de la web. |

Tabla 21 - Puertos Procesamiento de datos

## Diagrama de paquetes

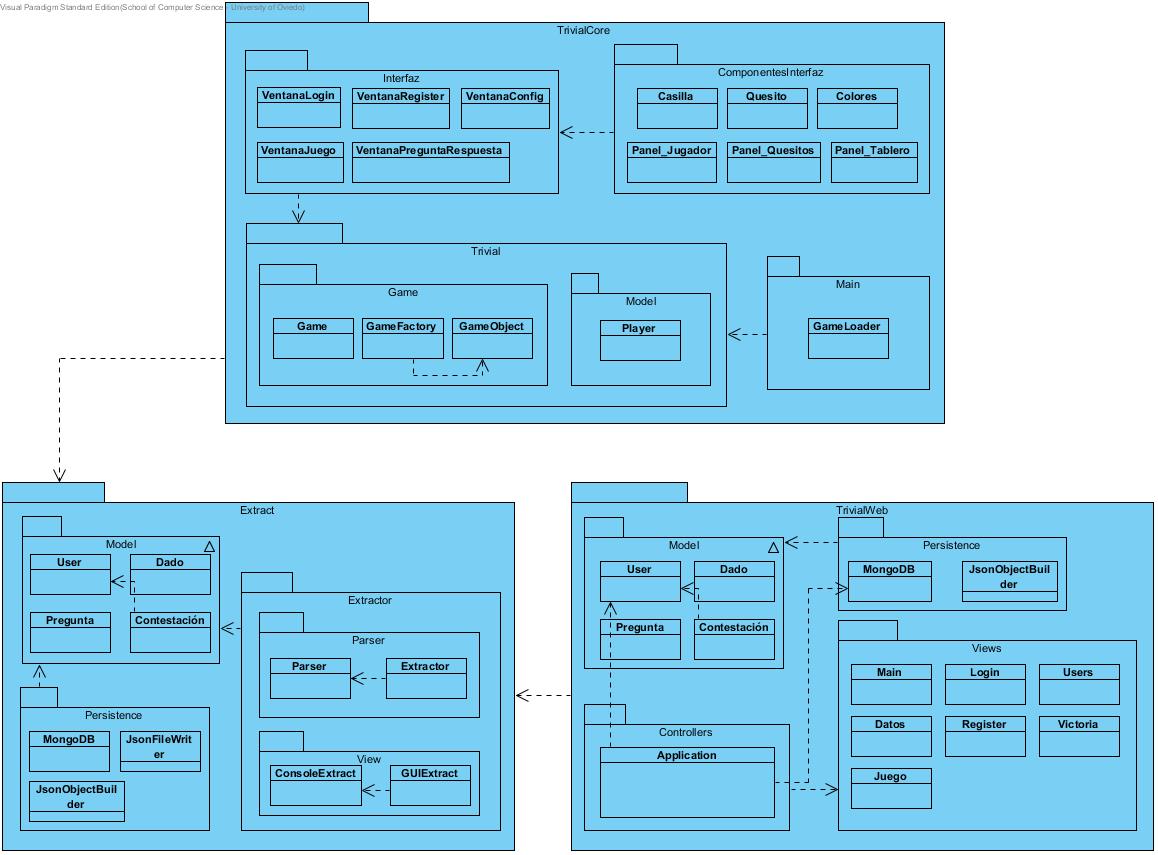


Ilustración - Diagrama de paquetes

|  |  |
| --- | --- |
| Elemento | Descripción |
| Extract | Añade preguntas a la base de datos. Se encarga de leer/parsear ficheros y actualizar la base de datos. |
| TrivialCore | Juego. |
| TrivialWeb | Juego en versión web |

|  |  |
| --- | --- |
| Elemento EXTRACT | Descripción |
| Model | Modelo de dominio. |
| Extractor | Paquete para la aplicación que extrae las preguntas. |
| Persistence | Paquete para la gestión de la base de datos. |

|  |  |
| --- | --- |
| Elemento Extract/Model |  |
| User | Representación del usuario de la aplicación. Incluye sus datos y datos relativos a las partidas. |
| Dado | Representación del dado. |
| Pregunta | Representación de la pregunta del juego. |
| Contestación | Representa la respuesta de un usuario a una pregunta del juego. |
| Elemento Extract/Extractor | |
| Parser | Paquete con el parser de ficheros. |
| View | Paquete con la parte visual del extractor. |
| Elemento Extract/Extractor/Parser | |
| Extractor | Parsea un fichero de preguntas |
| Parser |  |
| Elemento Extract/Extractor/View | |
| ConsoleExtract | Aplicación del extractor de preguntas. |
| GUIExtract | Arranca el programa extractor de preguntas |
| Elemento Extract/Persistence | |
| MongoDB | Operaciones de gestión de datos de usuarios y preguntas. |
| JsonFileWriter | Escribe el fichero JSON |
| JsonObjectBuilder | Creación de preguntas y/o usuarios a partir de un fichero JSON |

|  |  |
| --- | --- |
| Elemento TRIVIALCORE | Descripción |
| Interfaz | Paquete con la parte gráfica del juego. |
| ComponentesInterfaz | Paquete con componentes gráficos del juego. |
| Trivial | Lógica del juego. |
| Main | Arranque de la aplicación. |

|  |  |
| --- | --- |
| Elemento TrivialCore/Interfaz |  |
| VentanaLogin | Ventana para que el usuario entre en sesión. |
| VentanaRegister | Ventana para que un usuario nuevo se registre en la aplicación. |
| VentanaConfig | Ventana donde el usuario configure el juego |
| VentanaJuego | Ventana principal donde el usuario juega. |
| VentanaPreguntaRespuesta | Ventana donde el jugador responde a la pregunta que se le muestra. |
| Elemento TrivialCore/ComponentesInterfaz | |
| Casilla | Representación de las casillas del tablero. |
| Quesito | Respresenta un quesito. |
| Colores | Asocia un color con una categoría. |
| Panel\_Jugador | Panel del jugador correspondiente |
| Panel\_Quesitos | Para visualizar los quesitos del jugador correspondiente |
| Panel\_Tablero | Visualización del tablero. |
| Elemento TrivialCore/Trivial | |
| Game | Paquete para la creación de partidas. |
| Model | Modelo para partidas. |
| Elemento TrivialCore/Trivial/Game | |
| Game | Interfaz con las operaciones propias de la partida. |
| GameFactory | Crea la partida |
| GameObject | Representa una partida. |
| Elemento TrivialCore/Trivial/Model | |
| Player | Implementa las operaciones del jugador en una partida. |
| Elemento TrivialCore/Main |  |
| GameLoader | Carga las preguntas y crea la partida. |

|  |  |
| --- | --- |
| Elemento TrivialWeb | Descripción |
| Model | Modelo de dominio. |
| Controllers | Controlador de la aplicación. |
| Persistence | Paquete para la gestión de la base de datos. |
| Views | Paquete con las vistas. |

|  |  |
| --- | --- |
| Elemento TrivialWeb/Model | |
| User | Representación del usuario de la aplicación. Incluye sus datos y datos relativos a las partidas. |
| Dado | Representación del dado. |
| Pregunta | Representación de la pregunta del juego. |
| Contestación | Representa la respuesta de un usuario a una pregunta del juego. |
| Elemento TrivialWeb/Controllers | |
| Application | Controlador de la aplicación. |
| Elemento TrivialWeb/Persistence | |
| MongoDB | Operaciones de gestión de datos de usuarios y preguntas. |
| JsonObjectBuilder | Creación de preguntas y/o usuarios a partir de un fichero JSON |
| Elemento TrivialWeb/Views | |
| Main | Pantalla de principal del usuario en sesión. |
| Datos | Formulario para visualizar y cambiar datos propios. |
| Juego | Pantalla principal del juego. |
| Login | Formulario de inicio de sesión. |
| Register | Formulario de registro en la aplicación. |
| Users | Lista para la gestión de usuarios. |
| Victoria | Pantalla para partida ganada. |

# Anexos

## Diagrama BPMN

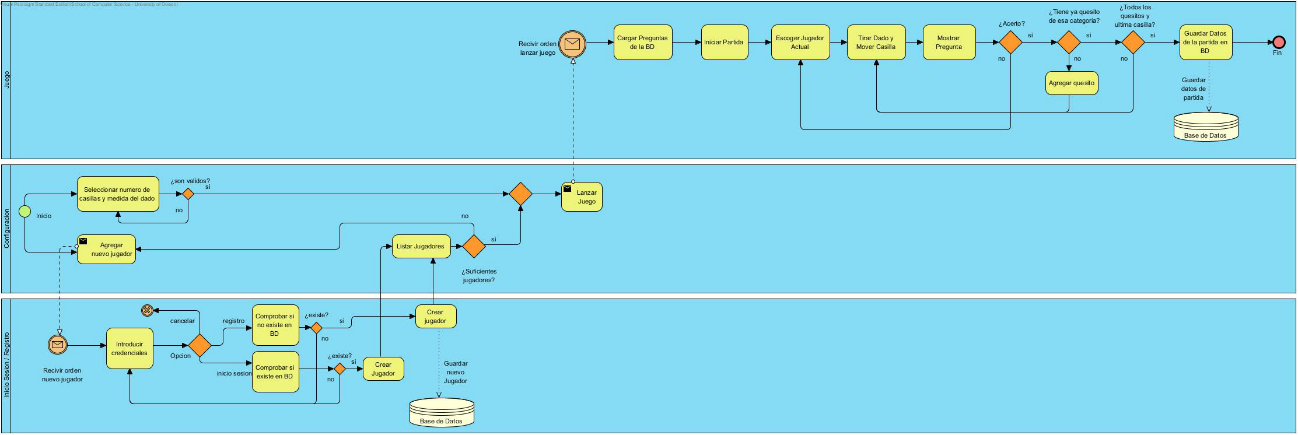


Ilustración - Diagrama BPMN

Un diagrama a tamaño real se puede encontrar en la siguiente dirección:

<http://alniarez.host56.com/ASW/Trivial4a/BPMN.png>

## Índice de figuras

[Ilustración 1 - Diagrama de contexto 4](#_Toc418683173)

[Ilustración 2 - Solución planteada 8](#_Toc418683174)

[Ilustración 3 - Diagrama del sistema 13](#_Toc418683175)

[Ilustración 4 - Diagrama de componentes (I) 14](#_Toc418683176)

[Ilustración 5 - Diagrama de componentes (II) 14](#_Toc418683177)

[Ilustración 6 - Diagrama de componentes (III) 15](#_Toc418683178)

[Ilustración 7 - Diagrama de componentes (IV) 15](#_Toc418683179)

[Ilustración 8 - Diagrama de paquetes 19](#_Toc418683180)

[Ilustración 9 - Diagrama BPMN 22](#_Toc418683181)

## Índice de tablas

[Tabla 1 - Historias de usuario 6](#_Toc418684025)

[Tabla 2 - Stakeholders 7](#_Toc418684026)

[Tabla 3 - Atributos de calidad 7](#_Toc418684027)

[Tabla 4 - Relacion Steakeholder/Atributo de calidad 8](#_Toc418684028)

[Tabla 5 - Escenarios de calidad 12](#_Toc418684029)

[Tabla 6 - Elementos del sistema 14](#_Toc418684030)

[Tabla 7 - Componentes Base de datos 16](#_Toc418684031)

[Tabla 8 - Puertos Base de datos 16](#_Toc418684032)

[Tabla 9 - Componentes Trivial4a 16](#_Toc418684033)

[Tabla 10 - Puertos Trivial4a 16](#_Toc418684034)

[Tabla 11 - Componentes Cometidos 17](#_Toc418684035)

[Tabla 12 - Puertos Cometidos 17](#_Toc418684036)

[Tabla 13 - Componentes Procesos aplicación 17](#_Toc418684037)

[Tabla 14 - Componentes Proceso login 17](#_Toc418684038)

[Tabla 15 - Puertos Proceso login 17](#_Toc418684039)

[Tabla 16 - Componentes Proceso gestión de datos 18](#_Toc418684040)

[Tabla 17 - Puertos Proceso gestión de datos 18](#_Toc418684041)

[Tabla 18 - Componentes Interfaz web 18](#_Toc418684042)

[Tabla 19 - Puertos Interfaz web 18](#_Toc418684043)

[Tabla 20 - Componentes Procesamiento de sesión 18](#_Toc418684044)

## Manual de usuario

[enlace](http://alniarez.host56.com/ASW/Trivial4a/manual/)