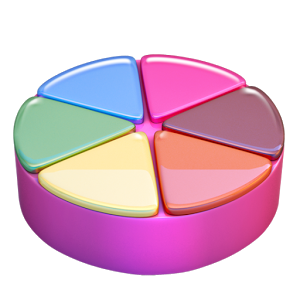
UNIVERSIDAD DE OVIEDO – ARQUITECTURA DEL SOFTWARE.



# Proyecto Trivial4A.

**Autor**: Grupo 4-A – Arquitectura Software.

**Titulación**: Grado en Ingeniería Informática del Software

15/04/2015

## Índice de contenido

[Proyecto Trivial4A. 1](#_Toc416813678)

[Índice de contenido 3](#_Toc416813679)

[Índice de figuras 6](#_Toc416813680)

[Índice de tablas 7](#_Toc416813681)

[Índice de requisitos 8](#_Toc416813682)

[Índice de pruebas 8](#_Toc416813683)

[1. Introducción 9](#_Toc416813684)

[1.1. Motivación del proyecto 9](#_Toc416813685)

[1.2. Objetivos 11](#_Toc416813686)

[1.3. Contenido de la memoria 12](#_Toc416813687)

[2. Análisis 13](#_Toc416813688)

[2.1. Introducción 13](#_Toc416813689)

[2.2. Aplicaciones similares 14](#_Toc416813690)

[2.2.1 Trivial Pursuit 14](#_Toc416813691)

[2.2.2 Trivinet 16](#_Toc416813692)

[2.2.3 Comparación de resultados 17](#_Toc416813693)

[2.4. Análisis de tecnologías y alternativas 18](#_Toc416813694)

[2.4.1. Plataforma 18](#_Toc416813695)

[2.4.1.1. Escritorio 18](#_Toc416813696)

[2.4.2. Tecnologías de despliegue 19](#_Toc416813697)

[2.4.2.1. JAVA 19](#_Toc416813698)

[2.4.3. Tecnologías de almacenamiento 19](#_Toc416813699)

[2.4.3.1. MongoDB 20](#_Toc416813700)

[2.5. Diagrama de casos de uso 21](#_Toc416813701)

[2.6. Requisitos de software 22](#_Toc416813702)

[2.6.1. Formato seguido en la especificación de requisitos 22](#_Toc416813703)

[2.6.2. Requisitos funcionales 23](#_Toc416813704)

[2.7.2.1. Requisitos funcionales de operación 24](#_Toc416813705)

[2.7.2.2. Requisitos funcionales de datos 25](#_Toc416813706)

[2.7.3. Requisitos no funcionales 26](#_Toc416813707)

[2.7.3.1. Requisitos no funcionales de rendimiento 26](#_Toc416813708)

[2.7.3.2. Requisitos no funcionales de interoperabilidad 26](#_Toc416813709)

[2.7.3.3. Requisitos no funcionales de interfaz 26](#_Toc416813710)

[2.7.3.4. Requisitos no funcionales de accesibilidad 27](#_Toc416813711)

[2.8. Plan de pruebas Error! Bookmark not defined.](#_Toc416813712)

[2.8.1. Formato de las pruebas **Error! Bookmark not defined.**](#_Toc416813713)

[2.8.2. Catálogo de pruebas **Error! Bookmark not defined.**](#_Toc416813714)

[3. Diseño 28](#_Toc416813715)

[3.1. Arquitectura del sistema 28](#_Toc416813716)

[3.1.1. Modelo a seguir 28](#_Toc416813717)

[3.1.2. Diagrama de componentes 31](#_Toc416813718)

[3.2 Diseño detallado Error! Bookmark not defined.](#_Toc416813719)

[3.2.1. Diseño de la Interfaz **Error! Bookmark not defined.**](#_Toc416813720)

[3.2.2. Diseño del controlador **Error! Bookmark not defined.**](#_Toc416813721)

[3.2.3. Diseño del modelo **Error! Bookmark not defined.**](#_Toc416813722)

[3.3. Diagramas de secuencia Error! Bookmark not defined.](#_Toc416813723)

[4. Atributos de calidad 32](#_Toc416813724)

[4.1 Atributos de calidad 32](#_Toc416813725)

[4.1.1 Disponibilidad 32](#_Toc416813726)

[4.1.2 Modificabilidad 32](#_Toc416813727)

[4.1.3 Seguridad 32](#_Toc416813728)

[4.1.4 Testabilidad 32](#_Toc416813729)

[4.1.5 Usabilidad 32](#_Toc416813730)

[4.2 Stakeholders 33](#_Toc416813731)

[4.3 Atributos de Calidad e Interesados 34](#_Toc416813732)

[4.4 Escenarios de Calidad 35](#_Toc416813733)

[5. Gestión del proyecto 36](#_Toc416813734)

[5.1. Planificación 36](#_Toc416813735)

[5.2. Medios técnicos empleados 36](#_Toc416813736)

[5.2.1. Hardware 36](#_Toc416813737)

[5.2.2. Software 37](#_Toc416813738)

[6. Conclusiones Error! Bookmark not defined.](#_Toc416813739)

[6.1. Conclusiones del proyecto Error! Bookmark not defined.](#_Toc416813740)

[6.2. Líneas futuras Error! Bookmark not defined.](#_Toc416813741)

[ANEXO I - Manual de usuario 38](#_Toc416813742)

[1. Interfaz Inicio Aplicación. 38](#_Toc416813743)

[2. Interfaz Login. 39](#_Toc416813744)

[3. Ventana de Juego. 40](#_Toc416813745)

[4. Parte dedicada al Administrador 43](#_Toc416813746)

[4.1. Gestión de preguntas de juego 43](#_Toc416813747)

[Fuentes de las imágenes utilizadas 44](#_Toc416813748)

[Bibliografía 44](#_Toc416813749)

## Índice de figuras

Figura 1: xxxxx xx

## Índice de tablas

Tabla 1: Comparación de resultados según los objetivos xx

## Índice de requisitos

Requisitos 1: Requisitos funcionales de operación xx

Requisitos 2: Requisitos funcionales de datos xx

Requisitos 3: Requisitos no funcionales de rendimiento xx

Requisitos 4: Requisitos no funcionales de interoperabilidad xx

Requisitos 5: Requisitos no funcionales de interfaz xx

Requisitos 6: Requisitos no funcionales de accesibilidad xx

## Índice de pruebas

Pruebas 1: Catálogo de pruebas xx

# Introducción

## Motivación del proyecto

Una vez desarrollado el sistema que permite obtener preguntas, la empresa NoGame ha decidido comenzar la fase de creación del juego Trivial. Para ello ha decidido desarrollar un primer juego que funcione como una aplicación tradicional en ordenadores de sobremesa.

El juego tomará las preguntas de la base de preguntas desarrollada en la primera fase y mostrará un interfaz gráfico en el que los concursantes puedan jugar al trivial.

Las reglas del juego Trivial vienen definidas en [1]. Aunque el tablero oficial está formado por una rueda con seis radios se podrá definir una versión simplificada en la que en lugar de una rueda, se siga una línea recta con una casilla final. El número de jugadores que podrán participar en cada partida será 1..N donde N es el número de colores disponibles en los quesitos.

La compañía está interesada en almacenar información sobre los jugadores por lo cual cada jugador dispondrá de un nombre de usuario y una password. Al iniciar el juego se mostrará una pantalla de configuración en la que se podrán especificar los jugadores y cada jugador introducirá sus credenciales. A partir de ahí, la máquina irá generando tiradas de dado para cada jugador, le preguntará a qué casilla quiere moverse y le mostrará la pregunta que corresponda según la categoría de la casilla.

Cuando un jugador acierta una pregunta de una determinada categoría, se le asigna el quesito del color correspondiente y puede continuar jugando. Si no acierta, pasa el turno al siguiente jugador. El jugador que tenga los quesitos de todos los colores podrá proceder a la última casilla. Si acierta la pregunta de dicha casilla finaliza el juego como ganador.

La compañía está muy interesada en la utilización del juego como herramienta de aprendizaje, por lo que desea incorporar una pantalla de administración en la que los usuarios que tengan esos privilegios puedan obtener información sobre qué usuarios han jugado, cuántas preguntas han acertado, qué preguntas son las más difíciles, etc.

Además, dado que la compañía se está planteando que el juego funcione en el futuro en diferentes plataformas (ordenadores de escritorio con diferentes sistemas operativos, dispositivos móviles, etc.) desea experimentar con diferentes representaciones visuales. Para ello desea que pueda separarse claramente la visualización del juego del resto de funcionalidad como de la interacción con el usuario o el funcionamiento del juego. De hecho han propuesto a los desarrolladores que exista la posibilidad de que los usuarios puedan modificar la apariencia de la aplicación cuando ellos lo deseen.

[1] Trivial Pursuit: http://wikipedia.org/wiki/Trivial\_Pursuit

En el siguiente apartado se encuentran definidos más a fondo todos los objetivos del proyecto a desarrollar.

## Objetivos

El principal objetivo perseguido **es el desarrollo de una aplicación llamada Trivial4a que permita a los Administradores evaluar a los jugadores de una forma diferente, consiguiendo que estos aprendan jugando y colaborando unos con otros. Dicha aplicación debe permitir registrar y analizar todas las puntuaciones obtenidas, para poder tener un control total sobre el aprendizaje de los jugadores y su forma de responder a tales preguntas.** La aplicación debe permitir a los administradores administrar las preguntas y respuestas dispuestas en un tablero de juego. Así podrán tener control total sobre el conocimiento y la curva de aprendizaje de todos los jugadores.

Los objetivos específicos de dicho proyecto se encuentran enumerados a continuación:

* + - La aplicación a desarrollar debe permitir a los alumnos registrados **jugar un número indefinido de partidas respondiendo las preguntas propuestas en cada una de ellas**. Dichas preguntas deben sucederse según el avance en un tablero de juego, que debe estar formado por casillas de diferente color según el tema de la pregunta.
    - Todas las puntuaciones obtenidas deben almacenarse y disponerse en forma de rankings, cuyo objetivo es **aumentar el grado de competitividad** y por lo tanto, la frecuencia de juego por parte de los alumnos.
    - En lo que a los administradores se refiere, la aplicación tiene como objetivo **el control total sobre la curva de aprendizaje de los jugadores, sabiendo en todo momento que preguntas fallan y cuales aciertan.**

## Contenido de la memoria

La presente memoria se encuentra estructurada de la siguiente forma:

* + - **Capítulo 1, Introducción**: En este primer capítulo se introduce al lector en el contexto de realización de dicho proyecto, indicando la motivación por la que se ha llevado a cabo y especificando los objetivos perseguidos.
    - **Capítulo 2, Análisis**: En dicho apartado se procede a realizar el análisis de tecnologías y alternativas candidatas para el despliegue, y aplicaciones similares implementadas, estableciendo en qué medida satisfacen o no los objetivos perseguidos. En este mismo apartado también se especifican los casos de uso, los requisitos y el plan de pruebas propio de la aplicación.
    - **Capítulo 3, Diseño**: En este capítulo enfocado al diseño de la aplicación, se establece la arquitectura del sistema identificando el tipo de aplicación a desarrollar, el modelo seguido y el diseño de los componentes que la forman.
    - **Capítulo 4, Implementación**: Este apartado, relacionado con la implementación del sistema, aborda a nivel de código aspectos y técnicas relevantes para implementar la aplicación. De la misma forma, se encarga de enunciar el resultado obtenido tras la realización del plan de pruebas.
    - **Capítulo 5, Gestión del Proyecto**: Otro de los puntos a desarrollar en la presente memoria es el dedicado a la gestión del proyecto, cuyo objetivo es analizar la planificación inicial y final del proyecto y el coste económico que ha supuesto. De la misma forma, se analizan diferentes opciones candidatas como posible vía de venta del proyecto.
    - **Capítulo 6, Conclusiones y Líneas futuras**: Se dispone el apartado que enuncia la conclusión personal y del proyecto, así como las posibles mejoras que podrían ser implementadas en la aplicación.
    - **Anexo I, Manual de usuario**: Este apartado del anexo es el encargado de albergar el manual que debe seguir un usuario para hacer un correcto uso de la aplicación. En dicho punto se especifican, con ayuda de imágenes detalladas, todas las funcionalidades de la aplicación y los pasos necesarios para llevarlas a cabo.
    - **Referencias:** Apartado final de la memoria donde se incluyen todas las fuentes de información consultadas.

# Análisis

## Introducción

A continuación se tratará de realizar un análisis completo sobre la aplicación a desarrollar, obteniendo así una especificación detallada del sistema que pueda servir de base para el diseño del mismo.

El análisis se realizará sobre los siguientes apartados:

* **Aplicaciones similares:** Se analizarán aplicaciones parecidas, diferenciándolas de la aplicación a desarrollar en este proyecto e identificando que carencias tienen en cuanto al objetivo perseguido por Trivial4a.
* **Análisis de tecnologías y alternativas:** Se analizarán todas las tecnologías pensadas para desarrollar la aplicación, tanto las elegidas finalmente cómo las posibles alternativas.
* **Elección final:** Se proporcionarán todas las razones que explican que tecnologías se han usado finalmente para la realización de la aplicación, indicando todas las ventajas e inconvenientes que supone dicha elección.
* **Diagrama de casos de uso:** Se incluirán diagramas que representan el comportamiento del sistema ante la acción distintos actores, en este caso Administrador y Usuario.
* **Requisitos Software**: En este apartado se analizarán todos los requisitos necesarios para dar una descripción completa de la aplicación a desarrollar.
* **Plan de Pruebas:** Finalmente, se establecerá un plan de pruebas que sea capaz de verificar la funcionalidad de los requisitos de la aplicación.

## 2.2. Aplicaciones similares

A continuación se van a analizar diferentes aplicaciones ya implementadas parecidas al software desarrollado en este proyecto. Se explicará su desarrollo, objetivos y características y se elaborará una lista de requisitos que servirá para comparar las ventajas e inconvenientes de dichas aplicaciones y Trivial4a.

### Trivial Pursuit

Trivial Pursuit es un juego de mesa. Scott Abbott, un editor deportivo del diario Canadian Press y Chris Haney, fotógrafo de la revista Montreal Gazette, desarrollaron la idea en diciembre de 1979; su juego fue lanzado al mercado dos años después.

En Estados Unidos, el juego se hizo popular en 1984, año en que se vendieron alrededor de 20 millones de ejemplares. Parker Brothers (ahora parte de Hasbro) compró los derechos del juego en 1988. Hasta 2004, se han vendido cerca de 88 millones de ejemplares, en 26 países y 17 idiomas. Este famoso juego de mesa se basa en ir respondiendo preguntas completando los radios de una rueda con los ‘quesitos’ obtenidos como premio al conseguir acertar cada una de las seis preguntas especiales dispuestas en el tablero.

El trivial ha pasado de ser un juego de mesa a estar desarrollado para una gran cantidad de plataformas. Lo podemos encontrar como juego de PC, de múltiples videoconsolas, tablets o dispositivos móviles.



Interfaz de juego de 'Trivial Pursuit' 1

El tablero está compuesto por casillas que forman una rueda con seis radios. Cada casilla lleva el color de un tipo de preguntas, con una casilla especial de cada color en el lugar en que los radios se unen a la rueda, donde aparece el quesito del respectivo color sostenido por dos querubines.

La edición Genus está compuesta por 4800 preguntas y respuestas en 800 tarjetas, 6 fichas de jugador, 36 cuñas de puntuación (quesitos), dos dados y un tablero. Se recomienda para personas de 10 años en adelante y pueden llegar a jugar de 2 a 36 jugadores (jugando en equipos de hasta 6 participantes). Los temas tienen un código de colores en la edición Genus: azul (geografía), rosa (espectáculos), amarillo (historia), marrón (arte y literatura), verde (ciencias y naturaleza) y naranja (deportes y pasatiempos).

Los jugadores comienzan en el centro de la rueda. Lanzan los dados en orden y mueven sus fichas con forma de queso en la dirección que elijan, el número de casillas que indica el dado. Tras mover la ficha, otro jugador coge una carta del mazo y hace al jugador que ha movido la ficha una pregunta que se corresponde con el color de la casilla a la que la ficha ha ido a parar. Si el jugador la responde correctamente puede tirar el dado de nuevo. Si la casilla es una de las casillas especiales situadas en la intersección, obtiene como premio una de las seis cuñas (popularmente llamadas "quesitos") del color de la casilla,

Si es que no la ha conseguido ya. Si el jugador responde incorrectamente, el turno pasa al siguiente jugador.

El juego prosigue hasta que un jugador obtiene los quesitos de los seis colores y

se dirige al centro. Una vez que llega allí el jugador debe responder una pregunta final, cuyo color escogen los rivales. Si la responde correctamente, gana el juego. Si no, se sigue jugando hasta que alguien lo consiga. Es común que, para ganar la partida, en vez de tener que responder una pregunta final a elegir por los rivales valga con responder correctamente cuatro de las seis preguntas de una tarjeta elegida al azar. También es común aplicar una regla en la que si un jugador cae en una casilla especial de premio de cuña (quesito) habiendo conseguido anteriormente esa cuña, si responde incorrectamente, puede perder esa cuña ganada.

### Trivinet

En trivinet.com queremos ser tu**juego de trivial** de referencia. El juego donde puedas pasar el rato respondiendo preguntas. Para ello puedes jugar al trivial con todo tipo de preguntas o puedes configurar tus propias partidas de trivial eligiendo temáticas y dificultades además del idioma.

Si optas por registrarte en nuestro trivial tienes que saber que es **totalmente gratuito** y que una vez registrado podrás crear tus propios grupos de preguntas y respuestas donde podrás competir directamente contra tus amigos o contra el mundo entero. También puedes optar por **crear tus propios grupos de trivial** de la temática que tú quieras, nuestro juego de trivial es totalmente configurable y gratuito.

Trivinet no es como trivial pursuit en el sentido de jugar con quesitos sino que se trata de un **juego de preguntas y respuestas** de muchas y variadas temáticas y con distintas dificultades para cada pregunta.



Interfaz de juego de 'Trivinet' 1

### Comparación de resultados

|  |  |  |  |
| --- | --- | --- | --- |
| **Aplicaciones**  **Objetivos** | **Trivial Pursuit** | **Trivinet** | **Trivial4a** |
| **Juego educativo** | x | x | x |
| **Online** | x | x |  |
| **Gratuito** |  | x | x |
| **Interfaz llamativa** | x |  | x |
| **Interacción alumno- alumno** | x | x | x |
| **Editable** |  |  | x |
| **Clasificación por temas** | x | x | x |
| **Modos de pregunta** | x | x | x |
| **Baja complejidad** | x |  | x |
| **Ameno** | x |  | x |
| **Implementado en red social** | x | x |  |

A continuación se muestra una tabla con los objetivos que debería cumplir una aplicación que persiga el mismo fin que la desarrollada en el proyecto. El objetivo de esta matriz es comparar las aplicaciones similares a Trivial4a para saber si pueden suplir o no su funcionalidad.

Tabla: Comparación de resultados 1

## 2.4. Análisis de tecnologías y alternativas

A continuación se van a describir todas las tecnologías candidatas para implementar la aplicación, explicando sus características, sus ventajas e inconvenientes, su instalación etc. De esta forma se recopilarán una serie de datos que servirán en el próximo apartado para poder identificar cuáles son las herramientas más adecuadas para la implementación de la aplicación.

### Plataforma

En este apartado se va a proceder a realizar un análisis de las posibles plataformas en las que se puede implementar la aplicación a desarrollar. Las candidatas son las siguientes:

### Escritorio

### Tecnologías de despliegue

A continuación se analizan las diferentes tecnologías sobre las que se puede realizar el despliegue de la aplicación. Las tecnologías candidatas son **JAVA**.

### JAVA

Actualmente la aplicación ha sido programada en su totalidad usando el lenguaje de programación JAVA, tanto la interfaz como la lógica interna del juego.



Logo Java 1

Java es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por Sun Microsystems. Hay muchas aplicaciones y sitios web que no funcionarán a menos que tenga Java instalado y cada día se crean más. Java es rápido, seguro y fiable. Desde portátiles hasta centros de datos, desde consolas para juegos hasta súper computadoras, desde teléfonos móviles hasta Internet, Java está en todas partes.

### Tecnologías de almacenamiento

Finalmente se procede a realizar el análisis de las tecnologías candidatas para llevar a cabo el almacenamiento de datos de la aplicación a desarrollar. Estas son las siguientes:

### MongoDB

Se usara MongoDB como base de datos para la gestión de preguntas y respuestas del juego.

MongoDB (de la palabra en inglés “humongous” que significa enorme) es un sistema de base de datos NoSQL orientado a documentos, desarrollado bajo el concepto de código abierto.

MongoDB forma parte de la nueva familia de sistemas de base de datos NoSQL. En vez de guardar los datos en tablas como se hace en las base de datos relacionales, MongoDB guarda estructuras de datos en documentos tipo JSON con un esquema dinámico (MongoDB llama ese formato BSON), haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida.

El desarrollo de MongoDB empezó en octubre de 2007 por la compañía de software 10gen. Ahora MongoDB es una base de datos lista para su uso en producción y con muchas características. Esta base de datos se utiliza mucho en la industria1 y MTV Network,2 Craiglist3 y Foursquare4 son algunas de las empresas que utilizan esta base de datos.



Logo MongoDB 1

Una de las diferencias más importantes con respecto a las bases de datos relacionales, es que no es necesario seguir un esquema. Los documentos de una misma colección – concepto similar a una tabla de una base de datos relacional -, pueden tener esquemas diferentes.

## Diagrama de casos de uso

A continuación se van a incluir los diagramas de caso de uso para los diferentes actores que pueden hacer uso de la aplicación…

## Requisitos de software

A continuación se van a especificar los requisitos de software, que valdrán para dar una descripción completa de la aplicación a desarrollar y del comportamiento de la misma. Para comenzar se especificará que formato siguen las tablas de requisitos y posteriormente se pasará a establecer cuáles son los requisitos funcionales y no funcionales de la aplicación.

La especificación de requisitos de software es la actividad en la cual se genera el documento, con el mismo nombre, que contiene una descripción completa de las necesidades y funcionalidades del sistema que será desarrollado; describe el alcance del sistema y la forma en como hará sus funciones, definiendo los requerimientos funcionales y los no funcionales. 

### Formato seguido en la especificación de requisitos

Las tablas de requisitos estarán compuestas por:

* + - * **Identificador**: Clave única que identifica cada requisito. Estará compuesta por las siglas de Requisito Funcional (RF) o Requisito No Funcional (RNF) seguido de un número único asignado de dos cifras.
      * **Nombre**: Nombre del requisito. Debe ser significativo en relación a la descripción.
      * **Descripción**: Explicación detallada del requisito.
      * **Prioridad**: Nivel de preferencia que tiene el requisito para el equipo de desarrollo.
      * **Verificabilidad**: Apartado formado por los identificadores de las pruebas que verifican el cumplimiento del requisito.

### Requisitos funcionales

A continuación se especifican los requisitos funcionales, necesarios para entender la aplicación y su comportamiento. Estos se dispondrán en dos grupos diferenciados, los requisitos de operación y los de datos.

El modelo que deben seguir las tablas de los requisitos funcionales es la siguiente:

|  |  |  |
| --- | --- | --- |
| Identificador | Descripción | Prioridad |
| RF – XX |  | Alta Media Baja |

Formato requisitos funcionales 1

### Requisitos funcionales de operación

|  |  |  |
| --- | --- | --- |
| Identificador | Descripción | Prioridad |
| RF - 01 | El sistema debe dar la posibilidad al usuario de la creación de su propio usuario con el que identificarse | Alta |
| RF - 02 |  | A |
| RF - 03 |  |  |

### Requisitos funcionales de datos

|  |  |  |
| --- | --- | --- |
| Identificador | Descripción | Prioridad |
|  | El sistema deberá procesar un fichero de entrada con preguntas y sus respectivas respuestas en formato GIFT. |  |
|  | El sistema deberá generar un fichero de salida con preguntas y sus respectivas respuestas en formato JSON. |  |
|  | El sistema deberá almacenar las preguntas obtenidas en una base de datos externa. |  |
|  | La conversión debe poder automatizarse para que se realice en cualquier momento. |  |
|  | El sistema no deberá realizar la conversión de forma interactiva. |  |

### Requisitos no funcionales

A continuación se especifican los requisitos no funcionales de la aplicación, cuya finalidad es indicar restricciones del sistema. El modelo que deben seguir las tablas de dichos requisitos es el mismo empleado para especificar los requisitos funcionales del apartado anterior.

### Requisitos no funcionales de rendimiento

|  |  |  |
| --- | --- | --- |
| Identificador | Descripción | Prioridad |
| RNF – 01 | El equipo en el que se ejecute la aplicación deberá tener instalado Java para su correcto funcionamiento. | Alta |

### Requisitos no funcionales de interoperabilidad

|  |  |  |
| --- | --- | --- |
| Identificador | Descripción | Prioridad |
| RNF-02 | El sistema deberá funcionar ininterrumpidamente y a tiempo completo (sistema 24/7). | Alta |

### Requisitos no funcionales de interfaz

|  |  |  |
| --- | --- | --- |
| Identificador | Descripción | Prioridad |
| RNF-03 | El sistema deberá trabajar en una resolución óptima para cualquier tipo de ordenador. | Alta |

**Requisitos 5: Requisitos no funcionales de interfaz**

### Requisitos no funcionales de accesibilidad

|  |  |  |
| --- | --- | --- |
| Identificador | Descripción | Prioridad |
| RNF – 04 | La aplicación debe mostrar todo su contenido en español | Alta |

# Diseño

En este capítulo se van a distinguir tres apartados diferenciados. En primer lugar se explica de forma detallada la arquitectura del sistema, que modelo se ha seguido para su desarrollo y los componentes que lo forma. Posteriormente, a más bajo nivel, se detallan todas las clases y métodos que forman las diferentes partes de la aplicación así como el diseño de todas las interfaces con las que el usuario puede interactuar. Por último, se presentarán algunos diagramas de secuencia que detallarán algunas acciones relevantes a realizar en la aplicación.

## Arquitectura del sistema

A continuación se explicará detalladamente la arquitectura del sistema a desarrollar. Se establecerá el modelo que sigue y los componentes que la forman.

### Modelo a seguir

El modelo seguido para realizar el diseño de la arquitectura del sistema será el Model-View-Controller(MVC). Como su propio nombre indica, dicho modelo divide la interacción entre el usuario y la aplicación en tres roles, el modelo (Lógica de negocio), la vista (Interfaz de usuario) y el controlador. El principal objetivo de este modelo es facilitar el crecimiento por separado de cada una de las partes, incrementar así la reutilización y flexibilidad y facilitar el desarrollo independiente, las pruebas y el mantenimiento de cada rol.

El patrón MVC fue una de las primeras ideas en el campo de las interfaces gráficas de usuario y uno de los primeros trabajos en describir e implementar aplicaciones software en términos de sus diferentes funciones.

MVC fue introducido por Trygve Reenskaug (web personal) en Smalltalk-76 durante su visita a Xerox Parc en los años 70 y, seguidamente, en los años 80, Jim Althoff y otros implementaron una versión de MVC para la biblioteca de clases de Smalltalk-80. Solo más tarde, en 1988, MVC se expresó como un concepto general en un artículo sobre Smalltalk-80.

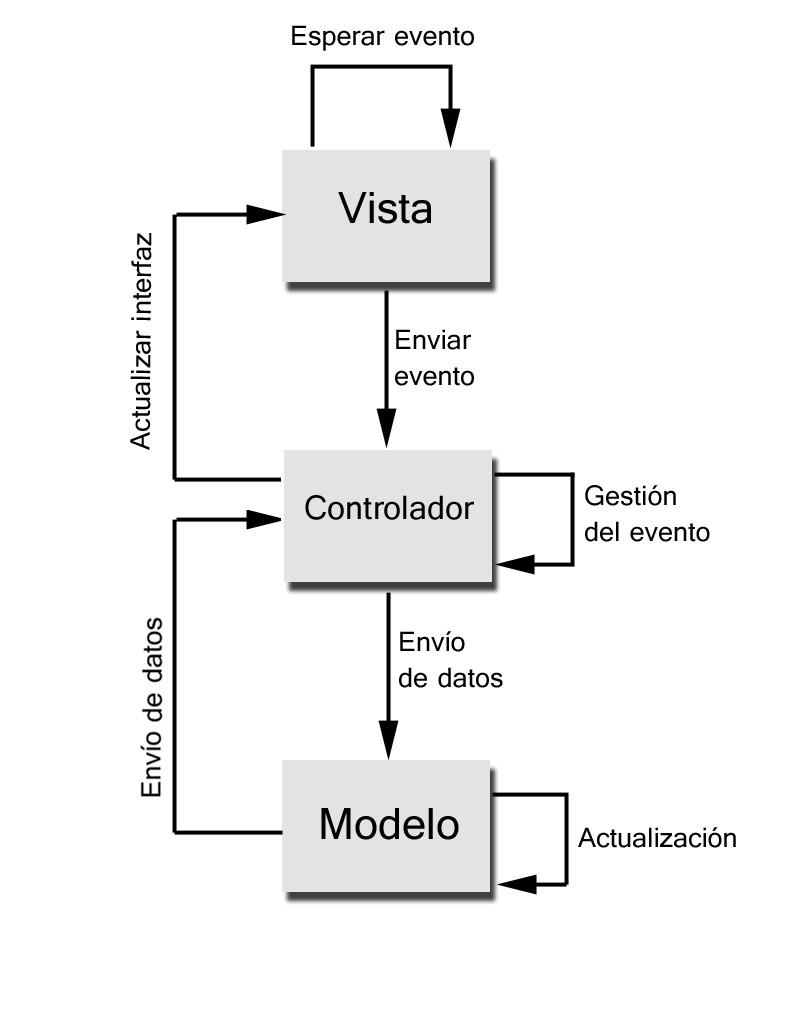
La función que desempeña cada rol de este modelo es la siguiente:

* **Modelo:** Es la parte de la arquitectura de la aplicación que representa la información con la que opera el sistema. El modelo responde a las peticiones de información, principalmente desde la vista, y a las instrucciones recibidas para realizar cambios, generalmente desde el controlador. Estos cambios de notifican a los usuarios a través de la vista.
* **Vista:** Es la parte de la arquitectura de la aplicación encargada de presentar el modelo al usuario de forma adecuada para la interacción, generalmente en forma de interfaz. Pueden existir múltiples vistas para cubrir varios propósitos.
* **Controlador:** Parte de la arquitectura de la aplicación encargada de recibir, y responder los eventos que vienen de parte del usuario. Se ocupa del tratamiento de datos y realiza peticiones tanto a la vista como al modelo.

El proceso que sigue esta arquitectura es el siguiente:

* El usuario interactúa de alguna forma con la parte de la vista (Interfaz), por ejemplo, pulsando un botón encargado de enviar un formulario.
* El controlador recibe el evento enviado por la vista y lo gestiona.
* Posteriormente, el controlador accede al modelo y este se actualiza. Por ejemplo, en este caso almacena los datos del formulario.
* La vista obtiene los datos que el modelo actualizado manda pasando por el controlador (Si requiere tratamiento de datos), actualizando así la interfaz notificando los cambios producidos. En este caso, por ejemplo se mostraría al usuario por pantalla los datos que envió pulsando el botón.
* La interfaz espera de nuevo el envío de eventos por parte del usuario de la aplicación.

Todo este ciclo se puede observar en la siguiente imagen:



Modelo Vista-Controlador 1

### Diagrama de componentes

A continuación se detalla el diagrama que representa los componentes que forman la aplicación a más alto nivel. Este se encuentra dividido principalmente en tres partes: La parte correspondiente a la vista, formada por todas las interfaces con las que los usuarios pueden interactuar; La parte dedicada al controlador de la aplicación, formada por aquellos componentes encargados de gestionar las preguntas, gestionar las peticiones y usuarios, gestionar el ranking y gestionar la partida; La tercera parte es la que representa el modelo, o lo que es lo mismo, la parte encargada de almacenar y proporcionar todos los datos necesarios para el funcionamiento de la aplicación. Esta última parte contiene componentes como los referidos a los datos de las preguntas, los datos propios de cada usuario, los datos de las peticiones y los datos de la partida. El diagrama que muestra las relaciones entre todos los componentes es el siguiente:

# Atributos de calidad

A continuación se describen los requerimientos que especifican los criterios para juzgar la operación de un sistema en lugar de su comportamiento específico.

## Atributos de calidad

### Disponibilidad

Disponibilidad del sistema para ejecutar la aplicación 24x7.

### Modificabilidad

Facilidad de cambio: nuevos ficheros fuente, nuevos formatos de representación de las preguntas, posibles cambios de base de datos.

Escalabilidad del sistema: incremento del número de usuarios y de las preguntas almacenadas.

### Seguridad

Seguridad de los datos almacenados: preguntas y respuestas, datos de los usuarios.

### Testabilidad

Facilidad para probar la fiabilidad de sistema, garantía de que las partidas se desarrollan correctamente (logs).

### Usabilidad

Facilidad de uso para los usuarios finales del juego, y para los encargados del mantenimiento del sistema.

|  |  |  |
| --- | --- | --- |
| Código | Descripción | Tipo de Atributo |
| AT001 | Disponibilidad del sistema para procesar datos 24x7. | Disponibilidad |
| AT002 | Facilidad de cambio: nuevos ficheros fuente, nuevos formatos de representación de las preguntas, posibles cambios de base de datos. | Modificabilidad |
| AT003 | Escalabilidad del sistema, por posible incremento del número de usuarios y de las preguntas almacenadas. | Modificabilidad |
| AT004 | Seguridad de los datos almacenados: preguntas y respuestas, datos de los usuarios. | Seguridad |
| AT005 | Integridad de los datos almacenados | Seguridad |
| AT006 | Facilidad para probar la fiabilidad | Testabilidad |
| AT007 | Facilidad de uso para los usuarios finales del juego, y para los encargados del mantenimiento del sistema. | Usabilidad |

## Stakeholders

|  |  |  |
| --- | --- | --- |
| Código | Stakeholder | Intereses |
| ST01 | Responsables de NoGame | Bajo coste de desarrollo, es decir, ajustar el tiempo de desarrollo del proyecto y el coste del mismo. |
| ST02 | Desarrolladores de NoGame | Baja intensidad tecnológica en el proyecto, para obtener un proceso de desarrollo controlado y predecible. |
| ST03 | Usuarios | Facilidad de uso de la aplicación. |

## 

## Atributos de Calidad e Interesados

|  |  |  |  |
| --- | --- | --- | --- |
| Atributos  Vs.  Interesados | ST01 | ST02 | ST03 |
| AT001 |  |  | x |
| AT002 |  | x |  |
| AT003 | x |  |  |
| AT004 | x |  | x |
| AT005 |  | x |  |
| AT006 |  | x |  |
| AT007 |  | x | x |

## 

## Escenarios de Calidad

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Nº | Fuente de estímulo | Estímulo | Entorno | Artefacto | Respuesta | Medición de la respuesta | Atributo de Calidad afectado |
| 1 | Desarrollador | Arranque del sistema | Desarrollo | Sistema | Tiempo para que el sistema esté funcionando razonablemente corto | Tiempo de arranque de la aplicación < 30 seg | AT001 |
| 2 | Desarrollador | Nuevo formato de preguntas | Desarrollo | Sistema | Implementar la nueva funcionalidad en un tiempo razonablemente corto | Tiempo de desarrollo < 2 días | AT002 |
| 3 | Desarrollador | Aumento del repositorio de preguntas | Desarrollo | Sistema/Base de datos | Adición de preguntas a la base de datos del sistema | Tiempo para la inclusión de las preguntas de un fichero < 2 min | AT003 |
| 4 | Usuario | Intento de acceso a datos privados del sistema | Explotación | Base de datos | Negación del sistema a realizar la operación pertinente | Nº de intrusiones y accesos no permitidos =0 | AT004 |
| 5 | Usuario | Almacenamiento de datos en el sistema | Explotación | Base de datos | Almacenamiento correcto de los datos del usuario en el sistema. | Sin pérdida de información | AT005 |
| 6 | Desarrollador | Pruebas del sistema | Desarrollo | Sistema | Ejecución de pruebas del sistema | Superación de las pruebas definidas para la aplicación | AT006 |
| 7 | Usuario | Utilización del sistema | Explotación | Sistema | El sistema será simple e intuitivo | Tiempo de aprendizaje para la utilización del sistema < 5 min | AT007 |

# Gestión del proyecto

En este apartado se pretende explicar detalladamente todos los aspectos que forman la gestión del proyecto. Para ello se van a distinguir dos puntos claramente diferenciados: Primero, se van a explicar cada una de las fases planificadas por las que ha pasado el proyecto,; en segundo lugar, se van a especificar todas las herramientas *software* y *hardware* que han sido necesarias para el desarrollo de la aplicación.

## Planificación

En primer lugar se encuentra la tarea de gestión de preguntas. En ella se realizó el Parser para la creación de todas las preguntas y respuestas del sistema que formarían la base de datos.

Más tarde daría comienzo la programación del sistema tanto lógica como visual.

En este punto, da comienzo la fase de diseño de la aplicación. En dicha fase se planifica analizar la arquitectura del sistema, identificando el tipo de aplicación a desarrollar y los componentes que la forman.

## Medios técnicos empleados

A continuación se van a detallar todos los medios empleados para el desarrollo de la aplicación. El apartado se encontrará dividido en dos partes, la primera para especificar las herramientas *hardware* y la segunda para especificar las de tipo *software*.

### Hardware

Como herramientas de tipo hardware se han empleado un ordenador de sobremesa para la implementación y codificación de la aplicación y un ordenador portátil para la elaboración de la documentación. Las características de dichos dispositivos se disponen en las siguientes tablas:

|  |  |
| --- | --- |
| Dispositivo | Sony Vaio VPECEH |
| Procesador | Intel Core i3-2330 2.20 Ghz |
| Memoria Ram | 5,00 Gb |
| Disco Duro | 500 Gb |
| Tarjeta Gráfica | Intel HD Graphics 3000 |

|  |  |
| --- | --- |
| Dispositivo | Ordenador Sobremesa Clase |
| Procesador | Pentium Dual-Core E5400 2.70 GHz |
| Memoria Ram | 2,00 Gb |
| Disco Duro | 200 Gb |
| Tarjeta Gráfica | Intel G33/G31 |

### Software

A continuación se va a mostrar una lista elaborada con los nombres de aquellas herramientas de tipo *software* empleadas para llevar a cabo el desarrollo de la aplicación. Dicha lista contendrá los siguientes campos: Tipo de software, nombre de la herramienta y página web donde se puede encontrar.

|  |  |  |
| --- | --- | --- |
| Tipo de software | Nombre de la herramienta | Sitio Web |
| Sistema Operativo | Windows 8 | http://emea.microsoftstore.com/es/es-ES/Microsoft/Windows/Windows-8 |
| Procesador de textos | Microsoft Office Word 2010 | <http://emea.microsoftstore.com/es/es-ES/Microsoft/Office> |
| Edición de diapositivas | Microsoft Office PowerPoint 2010 | <http://emea.microsoftstore.com/es/es-ES/Microsoft/Office> |
| Entorno de programación | Eclipse | http://www.eclipse.org/downloads/ |
| Control de versiones | Github | https://www.dropbox.com/ |
| Hoja de cálculo | Microsoft Office Excel 2010 | <http://emea.microsoftstore.com/es/es-ES/Microsoft/Office> |
| Editor de imágenes | Adobe Photoshop Cs6 | http://www.adobe.com/cfusion/tdrc/ index.cfm?product=photoshop&loc=e ses |
| Conversor PDF | PDF Creator v1.2 | http://www.pdfforge.org/download |
| Visor PDF | Adobe Reader X 10.1 | http://get.adobe.com/es/reader/ |
| Creación de diagramas de secuencia | Visual Paradigm for UML Enterprise edition | http://www.visual- paradigm.com/download/vpuml.jsp |
| Creación de Base de Datos | MongoDB | https://www.mongodb.org/downloads |
| Creación de diagramas de clases | Star UML v5.0 | http://staruml.sourceforge.net/en/ |

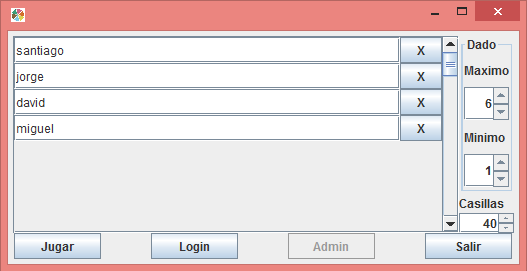
# ANEXO I - Manual de usuario

A continuación se desarrolla el manual que debe seguir el usuario para realizar un uso correcto de la aplicación implementada. En dicho documento se identifican, con ayuda de imágenes detalladas, todos los elementos que forman las diferentes partes del juego. Esto se lleva a cabo tanto para la parte dedicada a los usuarios que posean el rol de alumno, como para los profesores que desempeñen la función de administrador.

## Interfaz Inicio Aplicación.

Al iniciar el juego, al usuario se le presenta una ventana de **Configuración**. A primera vista se le mostraran todos los **usuarios que van a estar presentes en el juego**, con su nombre seguido del botón “**X**” donde se podrá eliminar a ese jugador de la partida.

A mano derecha tendrá la opción de poder escoger las opciones del **dado**, pudiendo configurar el **máximo/mínimo** valor que un jugador puede sacar en una tirada y el número de **casillas** que tendrá el tablero en esa partida.



Para poder **añadir tu nombre** a la partida deberás hacer **login** con tu propia cuenta en el sistema, para acceder a este menú bastará con pulsar la tecla “**Login**” de la ventana de configuración de la partida. Sin cuenta es imposible acceder a la partida, por lo tanto cada jugador deberá tener su **propia cuenta**.

Cuando un administrador haga **Login** en el sistema, se le habilitará automáticamente el botón **“Admin**” donde entrara en un menú para poder **configurar las preguntas y respuestas** del sistema. Este punto se verá más adelante del manual.

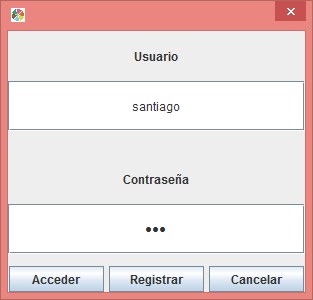
Una vez introducido todos los datos correctamente se pulsará el botón **“Jugar”** que dará comienzo a la partida, donde estarán presentes tantos jugadores como hayan inscritos en el menú de configuración.

Si se desea **cancelar la operación** en cualquier momento se podrá pulsar el botón “**Salir**” y se cerrará la ventana actual.

## Interfaz Login.

Si accedemos al menú “**Login**” pulsando el botón correspondiente en la ventana de **configuración** se accederá a la ventana donde podrá que meter sus credenciales para poder jugar con su cuenta y pulsar el botón **“Acceder”**. Si el usuario no dispusiese de cuenta propia se le daría la opción de crear una nueva cuenta usando el botón **“Registrar”.**

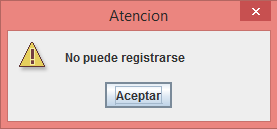
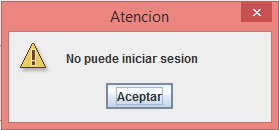
Si el usuario dese cancelar la operación puede pulsar en cualquier momento el botón **“Cancelar”** el cual terminará la acción en curso.



Ventana de Login 1

Si el usuario inserta los datos **incorrectamente** se le mostrara la siguiente pantalla notificando el problema para que pueda revisar los datos.

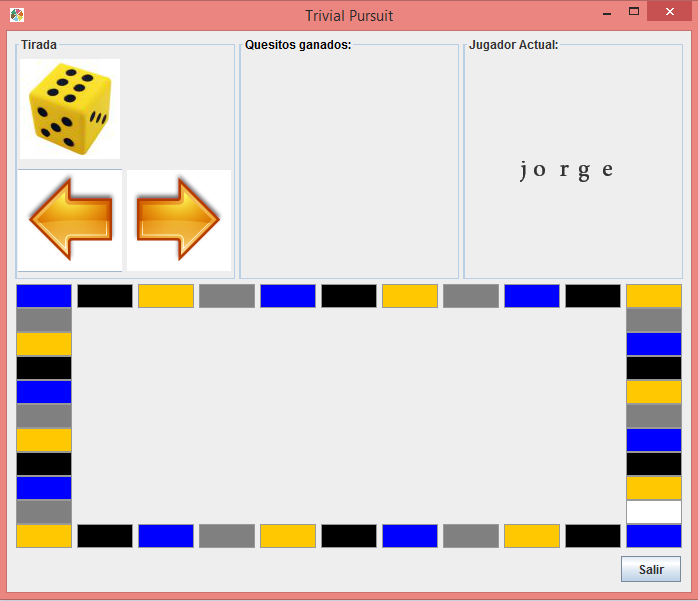
Lo mismo al usar la opción de **registro**, donde podrá revisar el error para registrar bien su cuenta.



## Ventana de Juego.

Una vez configurada todas las opciones y pulsado el botón “**Jugar**” en la venta de **configuración** se procederá al inicio del juego.

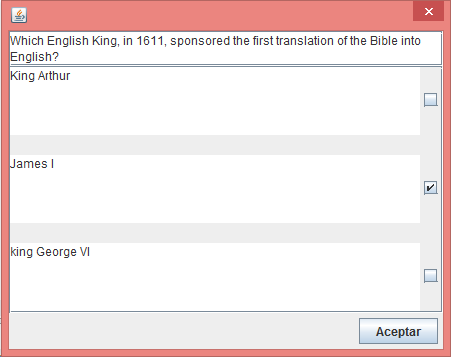
Aquí comienza el juego modo Trivial. **El turno del jugador** se mostrará en la sección **superior derecha**, donde se mostrara el **nombre** del usuario a quien le toca la tirada de dado.



En la sección **superior izquierda**, se mostrara el **dado** y las **flechas de jugada**. Cuando un usuario desea efectuar una tirada bastará con pulsar encima del **dado** y se procederá a calcular un numero aleatorio entre **el mínimo/máximo** que se ha establecido en la ventada de **configuración**.

Una vez calculado ese número podrá seleccionar decidiendo donde moverse en el tablero con las **flechas de dirección**.

Cuando el usuario caiga en **una casilla**, se le mostrará la **pregunta** con diferentes **respuestas** donde tendrá que seleccionar la **respuesta correcta** seleccionándola en la parte **derecha**.

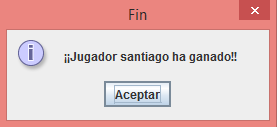


Pulsando el botón “**Aceptar**” volverá a la ventana del juego y se mostrara un mensaje indicando si la respuesta ha sido correcta o incorrecta.

En todo momento el usuario podrá visualizar todos **los quesitos ganados** al acertar preguntas en la sección medio-superior de la ventana de juego, donde se le mostrara todos los quesitos ganados.

Cuando un usuario **complete todos los quesitos** en todas las categorías de preguntas será proclamado vencedor y será el **ganador del juego**.

Se mostrara una ventana informando del **nombre del ganador**.



## Parte dedicada al Administrador

### Gestión de preguntas de juego

## Fuentes de las imágenes utilizadas

Imagen Trivial Pursuit : http://www.pcprofessionale.it/stappro61/uploads/2009/01/trivial.jpg

Imagen Trivinet: http://imagenes.es.sftcdn.net/es/scrn/11000/11918/trivinet-4.jpg

Imagen Java: http://www.tutorialspoint.com/images/java-mini-logo.png

Imagen MongoDB: http://photos3.meetupstatic.com/photos/event/c/9/7/c/highres\_14391580.jpeg

## Bibliografía

* 1. http://es.wikipedia.org/wiki/Trivial\_Pursuit
  2. http://foro.desdelinux.net/viewtopic.php?id=2079
  3. http://aplicacionesandroid.info/open-trivial-espanol/
  4. http://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93controlador
  5. http://www.trivinet.com/
  6. https://www.java.com/es/download/faq/whatis\_java.xml
  7. http://es.wikipedia.org/wiki/MongoDB
  8. http://www.genbetadev.com/bases-de-datos/mongodb-que-es-como-funciona-y-cuando-podemos-usarlo-o-no