# Descripción del proyecto

NoGame, empresa dedicada a la creación y el desarrollo de videojuegos, va a crear una colección de juegos de preguntas/respuestas para varias plataformas. Su primer proyecto es crear una variante del Trivial.

## Primera Fase

NoGame desea una aplicación que lea ficheros de preguntas y respuestas en formato GIFT (en un futuro podría ampliarse a otros formatos) y los procese a formato JSON. Después almacenará las preguntas en una base de datos. Estas dos fases de la aplicación deben estar claramente diferenciadas

## Segunda Fase

NoGame desea una aplicación de escritorio multiplataforma que simule el juego del Trivial. Para ello se tomarán las preguntas de la base de datos y se ofrecerá al jugador un interfaz gráfico para jugar cuya apariencia podrá ser modificada por el jugador.

Además del propio juego, la empresa desea almacenar estadísticas sobre los jugadores (identificados con usuario y contraseña) para que sirva de herramienta de aprendizaje. Esta utilidad estará disponible únicamente para los administradores y en ella se mostrará información sobre preguntas acertadas, usuarios que juegan…

## Tercera Fase

NoGame desea ampliar el juego del Trivial para que pueda usarse a través de la web aumentando así su potencial.

Se reutilizará la lógica de la fase anterior añadiendo una capa para conectarse vía web y jugar partidas. Deben tratarse los problemas de conexión que podrán tener los jugadores. De nuevo desean diferentes visualizaciones y también compatibilidad con distintos dispositivos.

La empresa desea añadir la posibilidad de que los jugadores puedan conocer sus datos históricos al conectarse, y también si hay otros usuarios en línea.

# Metodología utilizada

Siguiendo las indicaciones de las clases teóricas de “Arquitectura del Software”, se empleará el método ADD (Attribute-Driven Design Method)

# Identificación de stakeholders (Descripción)

* **Empresa (NoGame):**  
  Son los dueños del proyecto y responsables de las decisiones finales, tanto a nivel de requisitos como de presupuesto.  
  Entre sus objetivos están:
  + Obtener aplicaciones (web y de escritorio) que atraigan al público y que sean funcional
  + Obtener una aplicación (de escritorio) multiplataforma (Android, Windows, IOS…)
  + Obtener una aplicación (web) apta para diferentes dispositivos (móvil, Tablet, Ordenador…), que cumpla los principios y estándares web de diseño responsable
  + Conseguir que el sistema sea una herramienta para obtener estadísticas de juego
  + Conseguir que el sistema le suponga un valor añadido
  + Conseguir que las aplicaciones (web y de escritorio) tengan unos tiempos de respuesta bajos, inferiores a 4-5 segundos (basado en estudios sobre el tiempo de espera de los usuarios)
  + Conseguir que el sistema tenga un diseño modular
  + Tener el sistema en el plazo estipulado
  + Conseguir que el proyecto tenga un coste de desarrollo bajo
* **Operario de la empresa**Es la persona que ejecuta la aplicación encargada del procesamiento de ficheros de preguntas o activa la ejecución automática cada cierto tiempo.  
  Entre sus objetivos están:
  + Interactuar con una interfaz fácil de comprender y utilizar
  + Aplicación que permita varias opciones de entrada
  + Aplicación que permita observar resultados intermedios de la conversión
  + No tener que resolver fallos de la aplicación
* **Desarrolladores (Equipo 2a)**Son los encargados de la construcción del software. Su responsabilidad es hacer, probar y mantener el código del sistema a desarrollar.  
  Entre sus objetivos están:
  + Hacer que el proyecto sea rentable
  + Hacer un sistema mantenible que permita ampliar su funcionalidad
  + Gozar de tiempo suficiente para la implementación del sistema
  + Conseguir aplicaciones (web y de escritorio) modular
  + Conseguir aplicaciones (web y de escritorio) que permitan intercambiar look & feel
  + Conseguir que las aplicaciones (web y de escritorio) tengan unos tiempos de respuesta bajos, inferiores a 4-5 segundos (basado en estudios sobre el tiempo de espera de los usuarios)
  + Conseguir un proyecto que les aporte conocimientos nuevos y experiencia
* **Arquitectos del sistema (Equipo 2a)**Son los encargados de documentar, estructurar, diseñar y mantener la arquitectura del sistema que da solución al proyecto encargado por la empresa.  
  Entre sus objetivos están:
  + Conseguir un sistema escalable y fácil de mantener
  + Hacer que el proyecto sea rentable
  + Acordar con la empresa unos requisitos que puedan cumplirse
  + Conseguir un proyecto que les aporte conocimientos nuevos y experiencia
* **Gestores de bases de datos (Equipo 2a)**Son los encargados de diseñar y mantener las bases de datos. También deben encargarse de la seguridad e integridad de los datos.  
  Entre sus objetivos están:
  + Garantizar la seguridad en el acceso a los datos
  + Garantizar la integridad de los datos almacenados
* **Cliente (Usuario jugador)**Son aquellas personas que utilizarán las aplicaciones de escritorio y web.  
  Entre sus objetivos están:
  + Registro en la aplicación (web) sencillo
  + Interfaz intuitiva y sencilla
  + Confiabilidad en la aplicación, es decir, que no produzca errores
  + Instalación de la aplicación (de escritorio) sencilla
  + Posibilidad de visualizar fácilmente su histórico de partidas y estadísticas de juego.
* **Administradores de NoGame (Usuario administrador)**Son aquellas personas que utilizarán las aplicaciones de escritorio y web como herramientas para ver las estadísticas del juego y los usuarios.  
  Entre sus objetivos están:
  + Visualizar fácilmente las estadísticas
  + Interfaz intuitiva

# Identificación inicial de atributos de calidad

El sistema a desarrollar debe cumplir con los siguientes atributos de calidad:

* **Disponibilidad**: Todo el sistema (aplicación de procesamiento de ficheros, aplicación de escritorio y aplicación web) debe poder ejecutarse en cualquier momento (disponibilidad 24/7)
* **Usabilidad**: Debe garantizarse la facilidad de uso de las aplicaciones, independientemente de si disponen de interfaz interactivo (aplicación de escritorio y web) o no (aplicación de procesamiento de ficheros). Debe conseguirse que la curva de aprendizaje sea lo más empinada posible (mayor eficiencia de aprendizaje-en poco tiempo se aprende mucho).
* **Mantenibilidad**
  + ***Modificabilidad***: Futuras modificaciones en el sistema deben poder llevarse a cabo sin problemas
  + ***Escalabiliad***: Futuras extensiones en el sistema deben poder llevarse a cabo sin problemas
  + ***Reutilización***: Los diferentes módulos del sistema deben ser independientes para poder ser utilizados en aplicaciones similares
* **Rendimiento**: Se debe garantizar un tiempo bajo de procesamiento y ejecución de los algoritmos del sistema. Así como una ejecución de la aplicación fluida sin ningún lag o delay. En ningún caso deben sobrepasarse 5 segundos de espera por parte del usuario.
* **Seguridad**: Todos los datos almacenados en la base de datos son información sensible para la empresa y para los propios usuarios, por tanto no debe disponerse de medidas de seguridad para acceder a éstos.
  + Sólo los administradores podrán acceder a los datos de todos los usuarios
  + Cada jugador particular podrá acceder únicamente a sus propios datos de usuario
* **Testabilidad**: Todo el sistema debe poder probarse de forma sencilla.
  + En la aplicación de procesamiento de ficheros se debe garantizar que la conversión de formatos es correcta y no se pierde información
  + En la aplicación web se debe garantizar el tratamiento de los errores de conexión de los distintos usuarios del juego
* **Interactividad**: Las aplicaciones web y de escritorio deben comunicarse con el usuario de forma interactiva. El usuario se comunica directamente con la aplicación y viceversa, y es el primero el que regula la comunicación (es el que toma las decisiones)
* **Accesibilidad**: La aplicación debe poder utilizarse por cualquier usuario independientemente de sus limitaciones individuales o derivadas del contexto de acceso
  + ***Accesibilidad web***: debe cumplir las Pautas de Accesibilidad al Contenido de la Web ([WCAG](http://www.w3.org/TR/WCAG10/)). Como mínimo debe alcanzarse el nivel de conformidad “A”

# Primer acercamiento a la solución

### Aplicación de procesamiento de ficheros y almacenamiento de preguntas

Se opta por desarrollar un traductor en dos etapas mediante el patrón arquitectónico de Pipe-and-filter.

La primera etapa analizará los ficheros con las preguntas y generará un formato intermedio. Esta etapa contendrá un analizador léxico y un analizador sintáctico que validarán el fichero de entrada y ayudarán a depurar los errores que se puedan cometer.

La segunda etapa tomará las preguntas en formato intermedio y las procesará para poder almacenarlas en la base de datos. Estará compuesta por un Visitor que genere el formato final y un controlador que añadirá las preguntas a la base de datos.

Esta solución facilita tratar diferentes formatos de entrada, permite observar los resultados intermedios y una ejecución en diferentes fases. Cumpliendo así los requisitos solicitados por la empresa.

### Aplicación de escritorio del juego del Trivial

Se opta por desarrollar el juego siguiendo el estilo MVC y una separación en capas.

La primera capa será la GUI y contendrá toda la lógica necesaria para representar las abstracciones de las vistas de nuestro modelo. Esta capa se implementará utilizando la librería gráfica de java.Swing.

La segunda capa será la lógica y contendrá todas las reglas y lógica de negocio asociada al juego y a su correcto funcionamiento. Esta capa servirá de puente entre la interfaz y la capa siguiente.

La última capa es la persistencia, responsable de gestionar los datos de los usuarios y las estadísticas comunicándose con el SGBD.

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

### Aplicación web del juego del Trivial

Se opta por desarrollar el juego siguiendo el patrón MVC aplicado a la arquitectura web. Este patrón divide la aplicación en capas bien diferenciadas, la presentación que incluye vista y controlador, y el modelo.

La capa de modelo es la representación de dominio específico de la información con la que opera la aplicación. Esta capa contiene la lógica que permite que el juego funcione correctamente. Además encapsula la capa de persistencia, que permite gestionar los datos de los diferentes usuarios así como sus históricos, comunicándose con nuestro SGBD.

La vista, contenida en la capa de presentación, se trata de la interfaz de usuario, que permite que el modelo pueda ser utilizado mediante interacción con el usuario. Pueden existir diferentes vistas para un mismo modelo, lo que permite que la aplicación sea multiplataforma sin necesidad de modificar el modelo.

El controlador (capa de presentación) es el encargado de responder y procesar los eventos, peticiones HTTP. Además, también puede invocar a los cambios en el modelo.

Esta arquitectura permite una aplicación con bajo acoplamiento entre la lógica y la presentación, aunque el acoplamiento entre vista y controlador es elevado. Al existir la separación de responsabilidades, se puede adaptar el juego a nuevas plataformas sin modificar la lógica.

## Riesgos relacionados con la solución

### Aplicación de procesamiento de ficheros y almacenamiento de preguntas

El sistema elegido tiene algunos riesgos que es preciso especificar:

* **Detección de errores en los ficheros de entrada**: Hay que tener en cuenta que nos e puede presuponer que los datos de entrada van a ser correctos y se podrán procesar. Para solucionarlo, cada filtro debe procesar los datos cuando son correctos y detectar cuando no lo son.
* **Garantizar la conversión correcta de los datos:** Relacionado con el problema anterior y con la testabilidad, es importante que los datos se procesen adecuadamente por cada filtro. Para solucionarlo, el programa garantizará la trazabilidad del flujo de datos. Se añadirán a cada filtro funciones que permitan reconocer, localizar y mostrar los datos encontrados en los ficheros de entrada. Además se generará una salida legible de los datos intermedios entre las dos etapas, y así poder realizar una comprobación independiente de los formatos de entrada y salida.
* **Posibilidad de admitir otros formatos de entrada o salida**: Relacionado con la modificabilidad, la aplicación debe poder ampliarse y aceptar diferentes formatos y adecuar el formato de salida a lo que se necesita. Para solucionarlo, los componentes de cada etapa estarán desacoplados entre sí. Así se garantiza que, mientras el formato intermedio se mantenga, las etapas se podrán modificar según los formatos de entrada y salida que se necesiten.

### Aplicación de escritorio del juego del Trivial

El sistema elegido se enfrenta a los siguientes riesgos:

* **Pruebas unitarias**: El programa debe poder testarse automáticamente (con excepción de la interfaz gráfica)
* **Extensión a nuevas plataformas**: Relacionado con la mantenibilidad, debe existir una división en capas bien definida que nos permite que el sistema sea fácilmente extensible.
* **Posibilidad de cambios de interfaz:** Una de los requisitos deseados es poder cambiar la representación visual sin tener que modificar las demás capas.

### Aplicación web del juego del Trivial

La solución escogida se enfrenta a algunos riesgos que es necesario identificar:

* **Pruebas unitarias**: El programa debe poder testarse automáticamente, con excepción de la interfaz gráfica que debe ser probada manualmente.
* **Extensión a nuevas plataformas**: Relacionado con la mantenibilidad, debe existir una división en capas bien definida que nos permite que el sistema sea fácilmente extensible.
* **Posibilidad de cambios de interfaz**: Una de los requisitos deseados es poder cambiar la representación visual sin tener que modificar las demás capas.

Si comparten problemas tal vez se pueden unir

# Lista actualizada de stakeholders

## Definición (comunes)

1. **Empresa NoGame**
   * Son los propietarios y responsables principales de la financiación de la aplicación.
   * Objetivos:
     1. Obtener aplicaciones (web y de escritorio) que atraigan al público y que sean funcional
     2. Obtener una aplicación (de escritorio) multiplataforma (Android, Windows, IOS…)
     3. Obtener una aplicación (web) apta para diferentes dispositivos (móvil, Tablet, Ordenador…), que cumpla los principios y estándares web de diseño responsable
     4. Conseguir que el sistema sea una herramienta para obtener estadísticas de juego
     5. Conseguir que el sistema le suponga un valor añadido
     6. Conseguir que las aplicaciones (web y de escritorio) tengan unos tiempos de respuesta bajos, inferiores a 4-5 segundos (basado en estudios sobre el tiempo de espera de los usuarios)
     7. Conseguir que el sistema tenga un diseño modular
     8. Tener el sistema en el plazo estipulado
     9. Conseguir que el proyecto tenga un coste de desarrollo bajo
2. **Desarrolladores (Equipo 2a)**
   * Equipo responsable del desarrollo del funcionamiento de la aplicación
   * Objetivos:
     1. Hacer que el proyecto sea rentable
     2. Hacer un sistema mantenible que permita ampliar su funcionalidad
     3. Gozar de tiempo suficiente para la implementación del sistema
     4. Conseguir aplicaciones (web y de escritorio) modular
     5. Conseguir aplicaciones (web y de escritorio) que permitan intercambiar look & feel
     6. Conseguir que las aplicaciones (web y de escritorio) tengan unos tiempos de respuesta bajos, inferiores a 4-5 segundos (basado en estudios sobre el tiempo de espera de los usuarios)
     7. Conseguir un proyecto que les aporte conocimientos nuevos y experiencia
3. **Arquitectos del sistema (Equipo 2a)**
   * Equipo responsable de todo lo que tenga que ver con el análisis del desarrollo del proyecto
   * Objetivos:
     1. Conseguir un sistema escalable y fácil de mantener
     2. Hacer que el proyecto sea rentable
     3. Acordar con la empresa unos requisitos que puedan cumplirse
     4. Conseguir un proyecto que les aporte conocimientos nuevos y experiencia
4. **Gestores de bases de datos (Equipo 2a)**
   * Equipo responsable del diseño, mantenimiento y seguridad de la base de datos.
   * Objetivos:
     1. Garantizar la seguridad en el acceso a los datos
     2. Garantizar la integridad de los datos almacenados

## Definición (propias)

### Aplicación procesamiento ficheros y almacenamiento preguntas

1. **Operario de NoGame**
   * Ejecuta la aplicación de forma manual o activa la ejecución automática
   * Objetivos:
     1. Interactuar con una interfaz fácil de comprender y utilizar
     2. Aplicación que permita varias opciones de entrada
     3. Aplicación que permita observar resultados intermedios de la conversión
     4. No tener que resolver fallos de la aplicación

### Aplicaciones de escritorio y web del juego del Trivial

1. **Cliente (Usuario jugador)**
   * Son las personas que ejecutaran la aplicación
   * Objetivos:
     1. Registro en la aplicación (web) sencillo
     2. Interfaz intuitiva y sencilla
     3. Confiabilidad en la aplicación, es decir, que no produzca errores
     4. Instalación de la aplicación (de escritorio) sencilla
     5. Posibilidad de visualizar fácilmente su histórico de partidas y estadísticas de juego
2. Administradores de NoGame (Usuario administrador)
   * Son los usuarios con privilegios suficientes para ver las estadísticas de juegos
   * Objetivos:
     1. Visualizar fácilmente las estadísticas
     2. Interfaz intuitiva

|  |  |  |
| --- | --- | --- |
| Código | Stakeholder | Intereses |
| ST-01 | Empresa NoGame | 1. Aplicaciones (web y de escritorio) atrayentes y funcionales. 2. Aplicación (de escritorio) multiplataforma. 3. Aplicación (web) apta para diferentes dispositivos que cumpla los principios y estándares Web de diseño responsable. 4. Sistema como herramienta para obtener estadísticas de juego. 5. Sistema suponga un valor añadido 6. Aplicaciones (web y de escritorio) con tiempos de respuesta inferiores a 5 segundos. 7. Sistema con diseño modular y mantenible 8. Cumplir los plazos estipulados 9. Coste de desarrollo bajo |
| ST-02 | Desarrolladores del sistema | 1. Proyecto rentable. 2. Gozar de tiempo suficiente 3. Sistema con diseño modular y mantenible 4. Aplicaciones (web y de escritorio) que permitan intercambiar look & feel 5. Aplicaciones (web y de escritorio) con tiempos de respuesta inferiores a 5 segundos 6. Proyecto que les aporte nuevos conocimientos y experiencia |
| ST-03 | Arquitectos del sistema | 1. Sistema escalable y fácil de mantener, debe estar preparado para ser usado por un gran número de usuarios, así como para que su funcionalidad sea ampliada. 2. Proyecto rentable. 3. Acordar con la empresa requisitos que puedan cumplirse 4. Proyecto que les aporte nuevos conocimientos y experiencia |
| ST-04 | Gestores de bases de datos | 1. Garantizar la seguridad en el acceso a los datos. 2. Garantizar la integridad de los datos almacenados. |
| ST-05 | Operario de NoGame | 1. Interfaz de fácil de comprender y utilizar. 2. Aplicación que permita varias opciones de entrada 3. Aplicación que permita observar resultados intermedios de la conversión 4. No tener que resolver fallos de la aplicación |
| ST-06 | Usuarios jugadores | 1. Registro sencillo en la aplicación web. 2. Interfaz de fácil de comprender y utilizar. 3. Confiabilidad en la aplicación 4. Instalación de la aplicación de escritorio sencilla 5. Posibilidad y facilidad de visualizar fácilmente su histórico de partidas y estadísticas de juego |
| ST-07 | Usuarios administradores | 1. Visualización sencilla de las estadísticas 2. Interfaz fácil de comprender y utilizar |

# Lista actualizada de atributos de calidad

|  |  |  |
| --- | --- | --- |
| Código | Descripción | Tipo de Atributo |
| AT001 | Las aplicaciones deben proporcionarle al usuario una interfaz sencilla. | Usabilidad y Accesibilidad |
| AT002 | Las representaciones visuales del juego deben ser atractivas. | Usabilidad y Accesibilidad |
| AT003 | Las acciones de tirar el dado y mover casillas deben ser sencillas e intuitivas. | Usabilidad y Accesibilidad |
| AT004 | Se debe asegurar la integridad de los datos almacenados e introducidos por los usuarios en la aplicación. | Seguridad |
| AT005 | Consistencia e integridad de la bases de datos del sistema. | Seguridad |
| AT006 | Seguridad de acceso a la base de datos. | Seguridad |
| AT007 | Denegar el acceso a usuarios ilegítimos de la aplicación. | Seguridad |
| AT008 | La aplicación debe estar separada en capas para evitar el acoplamiento. | Mantenibilidad |
| AT009 | Las modificaciones que se lleven a cabo en las implementaciones concretas no deberían implicar al total de la aplicación. | Mantenibilidad |
| AT010 | El jugador debe ver de forma clara a qué casillas puede desplazarse. | Usabilidad y Accesibilidad |
| AT011 | La lógica de negocio empleada debe ser totalmente independiente de las vistas que se utilicen. | Mantenibilidad-Reutilización |
| AT012 | Los algoritmos que calculan las casillas posibles deben garantizar un tiempo de ejecución inferior a 2 segundos. | Rendimiento |
| AT013 | El tiempo de respuesta debe ser inferior a 5 segundos independientemente del número de jugadores activos. | Rendimiento |
| AT014 | Un jugador puede acceder en todo momento independientemente del volumen de usuarios activos. Disponibilidad 24/7 | Disponibilidad |
| AT015 | La aplicación debe estar preparada para añadir nuevas funcionalidades. | Mantenibilidad-Modificabilidad |
| AT016 | Disponibilidad 24/7 del sistema para procesar los ficheros | Disponibilidad |
| AT017 | Facilidad de cambio de la entrada o salida de los archivos | Mantenibilidad- Modificabilidad |
| AT018 | Facilidad para probar la fiabilidad del sistema para procesar los ficheros | Testabilidad |
| AT019 | Garantía de conversión de formatos correcta y sin pérdida de datos | Testabilidad |
| AT020 | Facilidad al indicar los ficheros a procesar | Usabilidad |

# Atributos de calidad y stakeholders

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Atributos  Stakeholders | ST-01 | ST-02 | ST-03 | ST-04 | ST-05 | ST-06 | ST-07 |
| AT001 |  |  |  |  |  | √ | √ |
| AT002 |  |  |  |  |  | √ |  |
| AT003 |  |  |  |  |  | √ |  |
| AT004 | √ | √ | √ | √ | √ | √ | √ |
| AT005 | √ |  |  | √ | √ | √ | √ |
| AT006 | √ |  |  | √ |  | √ | √ |
| AT007 | √ |  |  | √ |  | √ | √ |
| AT008 | √ | √ | √ |  |  |  |  |
| AT009 | √ | √ | √ |  |  |  |  |
| AT010 |  |  |  |  |  | √ |  |
| AT011 | √ | √ | √ |  |  |  |  |
| AT012 |  | √ |  |  |  | √ |  |
| AT013 | √ | √ |  |  |  | √ |  |
| AT014 | √ | √ |  |  |  | √ |  |
| AT015 | √ | √ | √ |  |  |  |  |
| AT016 | √ | Antes lo teníamos puesto |  |  | √ |  |  |
| AT017 | √ | Antes lo teníamos puesto |  |  | √ |  |  |
| AT018 |  | Antes lo teníamos puesto |  |  | √ |  |  |
| AT019 | √ | Antes lo teníamos puesto |  |  | √ |  |  |
| AT020 |  | Antes lo teníamos puesto |  |  | √ |  |  |

# Descripción del negocio de la solución

En este apartado se describirá el modelo de negocio de al aplicación

### Aplicación de procesamiento de ficheros y almacenamiento de preguntas

Está constituido por un programa ubicado en el ordenador del operador que se ejecuta mediante línea de comandos. El operador lo ejecutará indicando el fichero de entrada y el resultado (preguntas y sus respuestas) se almacena en la base de datos.

#### Solución

* El Parser recibe el nombre de un fichero en formato GIFT que será introducido por el operador, lo procesa e indica por consola los resultados intermedios obtenidos en la conversión.
* Una vez finalizada la traducción el Saver, tiene la responsabilidad de generar un fichero de salida “salida.txt ” con el formato indicado, en este caso en formato JSON.

#### Tecnologías

Por completar

### Aplicación de escritorio del juego del Trivial

Está constituido por una aplicación de escritorio multiplataforma, con interfaz gráfico intercambiable y que se ejecuta en el ordenador del cliente

#### Solución

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

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

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

#### Tecnologías

En este apartado se exponen las distintas opciones tecnológicas elegidas.

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

### Aplicación web del juego del Trivial

Está constituido por una aplicación web con interfaz gráfico.

#### Solución

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

* Modelo: Esta capa debe contener la lógica de negocio involucrada en el funcionamiento del juego Trivial, así como las operaciones y procesos relacionados con tareas de bases de datos (CRUD) mediante DAOs.
* Vista: Posee la abstracción visual del modelo y se comunica, únicamente con de controlador.
* Controlador: Recibe las interacciones del usuario y las coordina con las acciones a realizar por el modelo. Por lo tanto debe comunicarse con la lógica de negocio del modelo, no con la persistencia, y con la vista.

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

#### Tecnologías

En este apartado se exponen las distintas opciones tecnológicas elegidas.

* **Lenguaje de programación**: Como lenguaje principal hemos utilizado Java, debido a que es el lenguaje con el que el equipo está más familiarizado, permitiendo así un desarrollo ágil. Para el desarrollo de la aplicación web se ha utilizado el framework “Play”, que incluye una utilidad que nos facilita la creación de las vistas mediante lenguaje HTML5 con bloques de Scala (scala-based template engine). Además, se ha utilizado JavaScript para añadir funcionalidad a la web.
* **Bases de Datos**: La gestión de usuarios y partidas se realizará con el SGBD de bases relacionales HSQLDB , ya que nos brinda la sencillez que necesitamos para esta ocasión.
* **Testabilidad**: para realizar las pruebas automáticas que permiten verificar el correcto funcionamiento de la aplicación se han usado dos tecnologías: Travis y JUnit.

# Escenarios de calidad

Por revisar

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Nº | Fuente | Estímulo | Entorno | Artefacto | Respuesta | Medición de respuesta | Atributo de calidad |
| 1 | Usuario jugador | Contestar a una pregunta | Producción | Controlador / GUI | Mostrar las alternativas posibles. | La selección se debe hacer con un solo clic. | AT001 |
| 2 | Usuario  jugador | Mover a una casilla. | Producción | Controlador / GUI | Moverse hasta una casilla. | El movimiento debe ser animado y menor a 1 segundo. | AT002  AT001 |
| 3 | Usuario administrador | Visualizar estadísticas de jugadores | Producción | Controlador | Muestra las estadísticas de los jugadores | El acceso a las estadísticas debe ser a través de un botón (un solo click) e introduciendo usuario y contraseña | AT001 |
| 4 | Usuario  jugador | Lanzar el dado | Producción | Controlador/ GUI | Sacar un número del dado. | Se obtiene un número del dado  Con apenas un clic y después de una animación en menos de 1 segundo. | AT003  AT002 |
| 5 | Usuario jugador | Guardar sus estadísticas de juego (automático) | Producción | Modelo (Persistencia) | Las estadísticas del jugador se guardan en la BD | El proceso debe hacerse automáticamente | AT004  AT005 |
| 6 | Operario | Almacenar preguntas procesadas en la BD | Desarrollo | Modelo (Persistencia) | Las preguntas y sus respuestas se guardan en la BD | El proceso debe hacerse automáticamente al finalizar el procesamiento de los datos | AT004  AT005 |
| 7 | Usuario jugador | Ver informaciones de las partidas de otros jugadores, sin privilegios. | Producción | Modelo (Lógica) | Acceso a la información. | Acceso denegado por no tener suficientes permisos. | AT006  AT007  Antes estaba AT004 |
| 8 | Usuario administrador. | Modificar un valor estadístico. | Producción | Modelo (Lógica) | Datos intactos. | 100% de los datos siguen intactos. | AT005 |
| 9 | Aplicación | Terminar partida | Producción | Modelo (Persistencia) | No perder información de las partidas y usuarios. | Se mantiene el 100% de la información. | AT005 |
| 10 | Usuario administrador | Ver informaciones de partidas todos los jugadores, con privilegios | Producción | Modelo (lógica) | Acceso a la información | Acceso permitido ya que tiene suficientes permisos | AT006  AT007 |
| 11 | Usuario jugador | Ver información de sus partidas | Producción | Modelo (Lógica) | Acceso a la información | Acceso permitido ya que el usuario puede ver sus propios datos | AT006  AT007 |
| 12 | Intento no autorizado de acceso a BBDD | Intento acceso a BBDD | Producción | Modelo (Persistencia) | Acceso no autorizado rechazado | No se ha realizado ningún cambio en la base de datos | AT007  AT006 |
| 13 | Desarrolladores | Cambiar la implementación de una vista. | Desarrollo | Vista / GUI | Aceptación e integración de la nueva vista. | Ningún cambio en el modelo | AT008  AT009 |
| 14 | Empresa | Pide un cambio en una vista | Desarrollo | Vista | Aceptación e integración fácil del cambio en la vista | Ningún cambio en el modelo | AT008  AT009 |
| 15 | Desarrolladores | Modificación en la lógica. | Desarrollo | Modelo (Lógica) | Compilar el código | Capa de persistencia y GUI no necesitan ser compiladas. | AT009  AT008  AT011 |
| 16 | Usuario jugador | Mover a una casilla. | Producción | Vista / GUI | Ver las opciones posibles. | Los destinos se destacan visiblemente. | AT010 |
| 17 | Desarrolladores | Añadir una nueva vista. | Desarrollo | Modelo (Lógica) | Funcionamiento correcto. | No es necesario modificar ningún aspecto de la lógica. | AT011  AT008  AT009 |
| 18 | Usuario jugador | Hacer un movimiento. | Producción | Modelo (Lógica) | Devolver los posibles movimientos. | El algoritmo debe tardar menos de 2 segundos. | AT012 |
| 19 | Usuario jugador | Contestar una pregunta en una partida con 5 jugadores | Producción |  | Devolver las posibles respuestas | El algoritmo debe tardar menos de 5 segundos | AT013 |
| 20 | Usuario jugador | Conectarse al trivial web cuando ya hay 20 jugadores en línea | Producción |  | Acceder a la aplicación | La aplicación le permite acceder aunque el número de usuarios activos sea elevado | AT014 |
| 21 | Desarrolladores | Incluir un temporizador | Desarrollo | Modelo (Lógica) | Aceptación e integración fácil de la extensión | Ningún cambio en el modelo | AT015 |
| 22 | Operario | Acceso a la base de datos | Desarrollo | Extracción de las preguntas | Procesar las preguntas extraídas | Disponibilidad del sistema 24/7 | AT016 |
| 23 | Empresa | Nueva procedencia de los datos, cambio de fuente | Desarrollo | Parser | Las preguntas ser convierten con éxito | El índice de preguntas convertidas se mantiene | AT017 |
| 24 | Desarrolladores | Añadir un nuevo formato de entrada | Desarrollo | Parser | Aceptar nuevo formato de entrada sin modificar todo el Parser | Cambios mínimos en el código fuente existente | AT017 |
| 25 | Desarrolladores | Añadir nuevo formato de salida | Desarrollo | Parser | Aceptar nuevo formato de entrada sin modificar todo el Parser | Cambios mínimos en el código fuente existente | AT017 |
| 26 | Desarrolladores | Fallo en la conversión de preguntas | Desarrollo | Parser | Reparar errores de código fuente | Tiempo de reparación inferior a un día | AT018 |
| 27 | Operario | Conversión de preguntas | Producción | Parser | No perder información de preguntas | Se mantiene el 100% de la información de las preguntas válidas | AT019 |
| 28 | Operario | Uso del conversor de preguntas | Producción | Parser | El operario no tiene problemas para indicar el fichero a procesar | El operario selecciona el fichero con apenas un click | AT020 |

# Vistas

# Manual de usuario

### Aplicación de procesamiento de ficheros y almacenamiento de preguntas

Para utilizar la aplicación manualmente es necesario abrir un terminal de comandos y hay dos opciones de ejecución:

#### Sin parámetros

Ejecutar la aplicación con el comando  
***java –jar Trivial2a.jar***  
Donde:

* Trivial2a.jar es el nombre del ejecutable java proporcionado por los desarrolladores del equipo 2ª

La aplicación funcionará con dos parámetros predefinidos:

* entrada.txt 🡪 fichero de entrada en formato GIFT
* salida.txt 🡪 fichero de salida en formato JSON

La aplicación mostrará por consola mensajes descriptivos del estado en el que se encuentra el proceso de traducción. El fichero de salida se ubicará en el mismo directorio que el ejecutablel Trivial2a.jar

#### Con parámetros

Ejecutar la aplicación con el comando:  
***java –jar Trivial2a.jar ficheroEntrada formatoEntrada ficheroSalida formatoSalida***

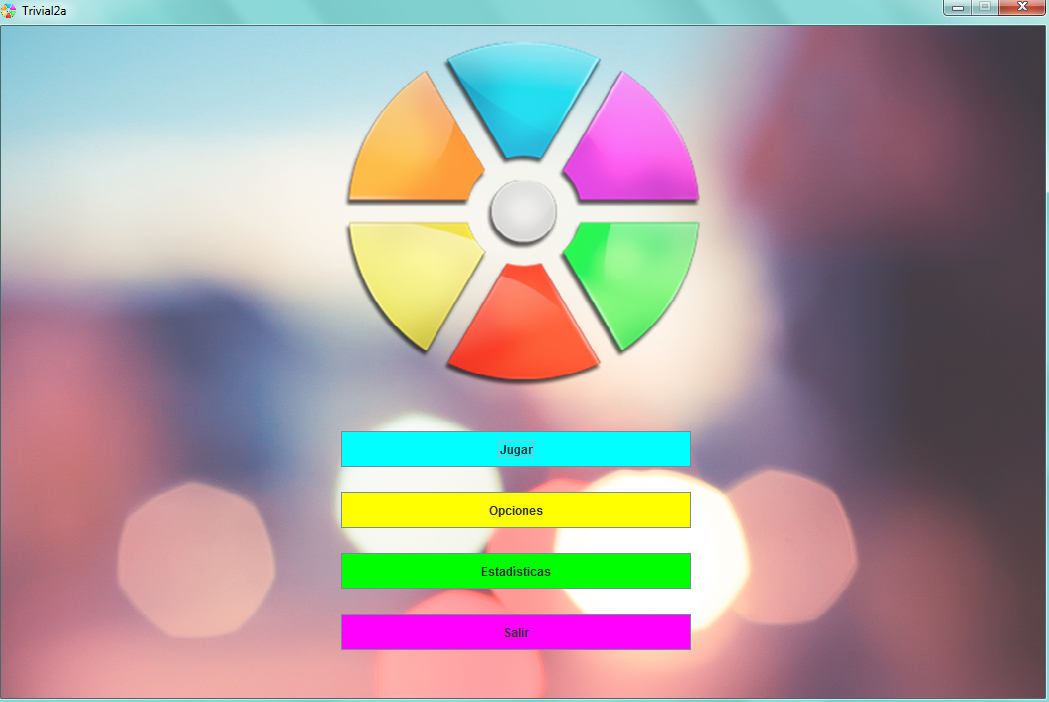
Por ejemplo, el procesamiento de un fichero de entrada llamado ejemplo.txt en formato XML para obtener un fichero en formato JSON denominado ejemplo\_salida.txt, se lograría ejecutando:  
***java –jar Trivial2a.jar ejemplo.txt XML ejemplo\_salida.txt JSON***

Para facilitar la ejecución automatizada de la aplicación, se proporciona un fichero .bat parametrizado para coger las preguntas en formato GIFT del fichero entrada.txt y generar el fichero equivalente salida.txt en formato JSON.

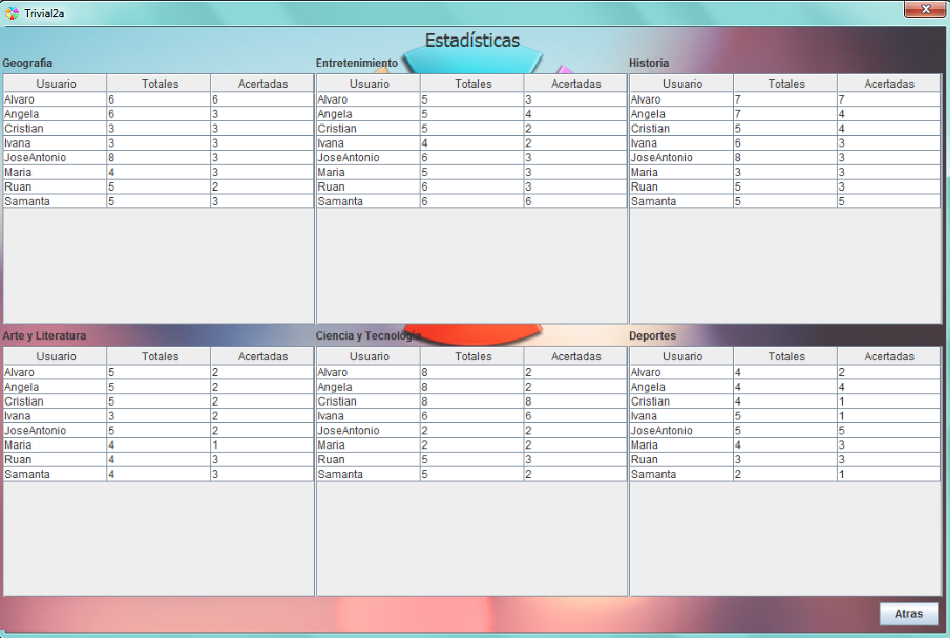
Para utilizar el fichero .bat basta con hacer doble click sobre el mismo

### Aplicación de escritorio del juego del Trivial

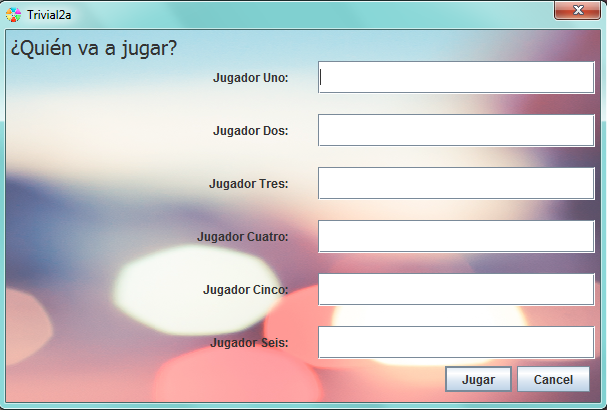
En primer lugar se muestra la pantalla de inicio en el que el usuario puede escoger iniciar una nueva partida, escoger entre diversas opciones en relación al diseño del tablero, consultar estadísticas en base a los aciertos y fallos de las preguntas o salir.



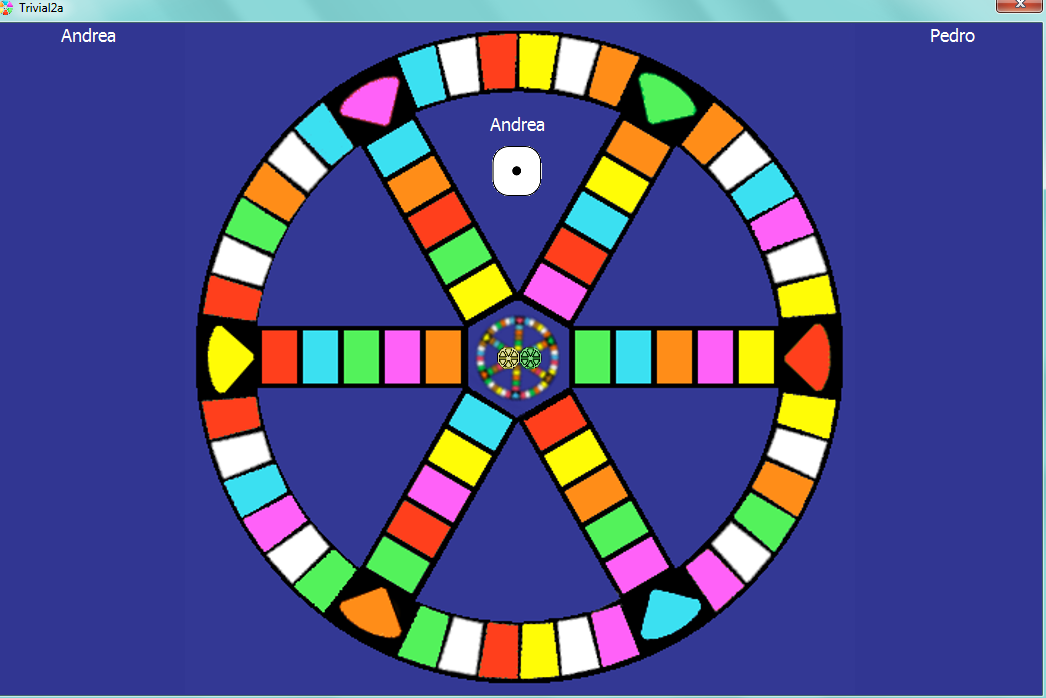
Si el jugador opta por visualizar las estadísticas podrá ver las respuestas acertadas y totales de cada categoría para cada usuario.



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



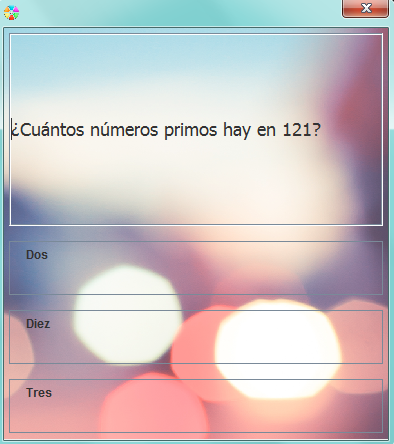
Si en la ventana anterior el usuario pulsa sobre jugar se presenta el tablero en la parte central, con el nombre de los jugadores a los lados. Puede observarse un dado, pulsar sobre el equivaldría a tirar el dado, por lo que el valor que se muestra en pantalla es el número de casillas que puede desplazarse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



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



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



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

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

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

### Aplicación web del juego del Trivial

# Manual del sistema

### Aplicación de procesamiento de ficheros y almacenamiento de preguntas

### Aplicación de escritorio del juego del Trivial

### Aplicación web del juego del Trivial