## 1. Descripción del proyecto

La empresa NoGame ya posee el sistema que le permite obtener preguntas, y han decidido comenzar la construcción del juego Trivial primeramente disponible solo para ordenadores de sobremesa.

Las preguntas que se pretende utilizar en el juego deben de tomarse de la bases de datos de la primera fase y se mostrará un interfaz gráfico para que los usuarios puedan jugar al trivial.

El trivial original consta de una rueda con seis radios, sin embargo se podrá definir una vista más simplificada, mediante la utilización de una línea recta con una casilla final. El número total de jugadores en una partida debe ser mínimo 1 y máximo el número de colores disponible en los quesitos.

Almacenar información sobre los jugadores del trivial supone un interés a la compañía, por lo cual cada jugador dispondrá de un login y una contraseña. Cuando se inicie el juego se mostrará una ventana de configuración donde se podrá especificar los concursantes que jugarán y donde los mismos se autenticarán. A partir de ahí la maquina será la responsable de lanzar el dado, esperará la respuesta de que casilla moverse y enseñara la pregunta correspondiente a la categoría de la misma.

En el caso de que un jugador acierte una pregunta de una determinada categoría, se le asigna el quesito del color correspondiente y se le permite seguir jugando. En caso contrario, pasa su turno al siguiente concursante. El jugador que complete el quesito con todos los colores podrá proceder a la casilla final. Si acierta la pregunta final, el usuario termina el juego como ganador de la partida.

Es muy importante para la compañía, que el juego sea utilizado como un mecanismo de aprendizaje y por ello desea que se incorpore una ventana de administración para que los usuarios que poseen esos privilegios puedan acceder a estadísticas sobre los usuarios que han jugado, el número de preguntas acertadas, las preguntas más difíciles, etc.

En un futuro la empresa, piensa extender la aplicación para que la misma funcione en distintas plataformas y por ello les gustaría experimentar con distintas opciones visuales del juego. Para ello debe de haber una clara separación entre la representación visual del funcionamiento lógico del juego y de la interacción con el usuario. A raíz de eso, han propuesto a los desarrolladores que el usuarios puedan modificar la interfaz de usuario si así lo desean.

## 2. Metodología utilizada

Siguiendo las pautas vistas en clases de teoría de la asignatura, emplearemos el método de desarrollo ADD (Atribute-Driven Design).

## 3. 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:

1. Obtener una aplicación que atraiga al público y que sea funcional.
2. Conseguir que la aplicación le suponga un valor añadido.
3. Tener la aplicación en el plazo estipulado.
4. Conseguir que el proyecto no tenga un coste elevado.
5. Conseguir que la aplicación sea una herramienta para obtener estadísticas.
6. Tener la posibilidad de hacer que la aplicación sea multiplataforma.

* **Desarrolladores (Equipo 2a)**

Son los encargados de la construcción del software. Su responsabilidad es codificar, probar y mantener el código del sistema desarrollado.

Entre sus objetivos están:

1. Conseguir una aplicación modular.
2. Conseguir una aplicación que permita intercambiar look & fell.
3. Hacer que el proyecto sea rentable.
4. A) Hacer un sistema mantenible, que permita ampliar su funcionalidad.

* **Arquitectos del sistema (Equipo 2a)**

Son los encargados de documentar, estructurar, diseñar y mantener el sistema que da solución al proyecto encargado por la empresa.

Entre sus objetivos están:

1. Garantizar un sistema reutilizable y de fácil mantenimiento.
2. Acordar con la empresa unos requisitos que puedan cumplirse.
3. Conseguir un proyecto rentable
4. 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:

1. Garantizar la seguridad en el acceso a los datos
2. Garantizar la integridad de los datos almacenados

* **Cliente (Usuario jugador)**

Son aquellas personas que utilizarán la aplicación.

Entre sus objetivos están:

1. Instalación de la aplicación sencilla
2. Interfaz intuitiva que haga sencillo el manejo de la aplicación
3. Confiabilidad en la aplicación

## 4. Identificación inicial de los atributos de calidad

La aplicación que se debe desarrollar debe cumplir con los siguientes atributos de calidad:

* **Interactividad:**  La aplicación debe comunicarse con el usuario de forma interactiva en todas sus formas, ya que se trata de un juego mayormente visual cuyo propósito es él de ser entretenido.
* **Usabilidad:** Es imprescindible que la aplicación no le suponga ningún tipo de dificultades de utilización al usuario, es sumamente importante que la curva de aprendizaje sea la mínima posible.
* **Seguridad:** La aplicación gestiona información sensible de los usuarios e información importante para la empresa. Por ello se deben tomar todas las medidas de protección necesarias para garantizar la seguridad tanto del sistema como de sus usuarios.
* **Mantenible:** La evolución del proyecto es un hecho irrefutable, por lo tanto el tiempo en futuras modificaciones y/o extensiones deberían de llevarse a cabo sin muchos problemas.
* **Reusable:** Las partes que integran el sistema deben de ser lo suficientemente independientes para que las mismas puedan ser aplicadas en el futuro. Sean para aplicaciones similares o no.
* **Rendimiento:** Se debe garantizar un tiempo correcto del procesamiento y ejecución de los algoritmos involucrados en la operación de juego. La aplicación debe correr fluida sin ningún tipo de lag o delay.

## 5. Primer acercamiento a la solución y riesgos

Para llevar a cabo este proyecto , se

Una vez estudiados los requisitos y tras una serie de propuestas de solución, se opta por desarrollar el juego siguiente el estilo MVC , y separación entre capas.

La primera capa será la GUI, contendrá todas la lógica necesaria para representar las abstracciones de las vistas de nuestro modelo. Dicha capa se implementará utilizando la librería grafica de java Swing.

La segunda capa será la Lógica, contendrá todas las reglas y lógica de negocia asociada al juego y al correcto funcionamiento de la aplicación. Esta capa servirá de puente entre la capa de interfaz y la siguiente capa.

La ultima capa será la capa de Persistencia. responsable de gestionar los datos de los usuarios y las estadísticas, comunicándose con nuestro SGBD.

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

### 5.1 Riesgos relacionados con la solución

El sistema así elegido se enfrenta a algunos riesgos que es preciso identificar:

* **Pruebas unitarias.** El programa debe poder ser testeable automáticamente con excepción de la interfaz grafica.
* **Extensión a nuevas plataformas.** Relacionado con su reusabilidad y mantenibilidad, el hecho de que exista una división entre capas bien definidas nos permite una extensión fácil en el futuro.
* **Posibilidad de cambios de interfaz.** Relacionado con el punto anterior , una de las principales características buscadas es el poder cambiar el representación visual sin tener que modificar ninguna de las demás capas.

## 6. Lista actualizada de stakeholders

### Definición (comunes)

#### Empresa NoGame

* Son los propietarios y responsables principales de la financiación de la aplicación.
* Objetivos:
  + Reducir el coste del desarrollo de la aplicación lo máximo posible
  + Obtener una aplicación multiplataforma.
  + Aumentar al máximo los beneficios conseguidos por ésta misma.
  + Conseguir que el juego sea funcional, atractivo y barato.

#### Desarrolladores juego Trivial

* Equipo responsable del desarrollo del funcionamiento de la aplicación.
* Objetivos:
  + Construir una aplicación segura y funcional.
  + Llevar a cabo implementaciones interesantes.
  + Desarrollar un proyecto beneficioso intelectual y económicamente.

#### Arquitectos de la aplicación

* Equipo responsable de todo lo que tenga que ver con el análisis del proyecto
* Objetivos:
  + Garantizar una aplicación reutilizable en distintos ámbitos y de fácil mantenimiento.
  + Obtener experiencia con la realización de este proyecto.
  + Seleccionar el mejor estilo arquitectónico para lograr una aplicación que cumpla con creces las necesidades del cliente.

#### Usuarios jugadores

* Son las personas que ejecutaran la aplicación
* Objetivos:
  + Disfrutar del entretenimiento ofrecido por el juego.
  + Utilizar el juego como una herramienta de aprendizaje.

#### Usuario administrador

* Son los usuarios con privilegios suficientes para ver las estadísticas de juegos.
* Objetivos:
  + Encontrarse con una interfaz sencilla e intuitiva.
  + Obtener información suficiente y clara sobre las partidas y los usuarios jugadores.

### Definición (propias)

#### Gestores de bases de datos

* Son los responsables de mantener tanto el correcto funcionamiento del sistema de gestión de bases de datos, como de la integridad y consistencia de la bases de datos

|  |  |  |
| --- | --- | --- |
| Código | Stakeholder | Intereses |
| ST-01 | Empresa NoGame | 1. Bajo conste de desarrollo y mantenibilidad del sistema. 2. Posibilidad de extensión de la aplicación, y que la misma sea multiplataforma. |
| ST-02 | Desarrolladores juego Trivial | 1. Sencillez a la hora de mantener el sistema y ampliar su funcionalidad. 2. Confiar en la seriedad y profesionalidad del cliente. 3. Llevar a cabo requisitos que se puedan cumplir. |
| ST-03 | Arquitectos de la aplicación | 1. Garantizar una aplicación de fácil mantenimiento y reutilizable. 2. Aumentar su experiencia como arquitecto. |
| ST-04 | Usuarios jugadores | 1. Divertirse con la experiencia de juego que proporciona Trivial. 2. Sentirse seguro con la protección de sus datos. 3. Instalar el juego y poder ejecutarlo aunque su máquina no sea demasiado potente. 4. Sencillez de configuración de las partidas. |
| ST-05 | Usuario administrador | 1. Encontrarse con una interfaz sencilla para obtener información estadísticas |
| ST-06 | Gestores de bases de datos | 1. Ofrecer un buen nivel de seguridad de acceso a la bases de datos. 2. Construir una bases de datos valiosa y consistente. |

## 7. Lista de atributos de calidad

|  |  |  |
| --- | --- | --- |
| Código | Descripción | Tipo de Atributo |
| AT001 | El juego debe proporcionarle al usuario una interfaz que involucre la interacción con los jugadores. | Interactividad |
| AT002 | Las representaciones visuales del juego deben ser divertidas y atractivas. | Interactividad |
| AT003 | Las acciones de tirar el dado y mover casillas debe ser sencillas e intuitivas. | Usabilidad |
| AT004 | Se debe asegurar los datos almacenados e introducidos por los usuarios en la aplicación. | Seguridad |
| AT005 | Consistencia e integridad de la bases de datos de la compañía. | 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. | Mantenible |
| AT009 | Las modificaciones que se lleven a cabo en las implementaciones concretas no deberían implicar la re compilación total de la aplicación. | Mantenible |
| AT010 | Facilidad al indicar los posibles movimientos de un jugador. | Usabilidad |
| AT011 | La lógica de negocio empleada debe ser totalmente independiente de las vistas que se utilicen. | Reusable |
| AT012 | Los algoritmos que calculan las casillas posibles deben garantizar un tiempo de ejecución bajo. | Rendimiento |

## 8. Atributos de calidad e interesados

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributos  Stakeholders | ST-01 | ST-02 | ST-03 | ST-04 | ST-05 | ST-06 |
| AT001 |  |  |  | √ | √ |  |
| AT002 |  |  |  | √ |  |  |
| AT003 |  |  |  | √ |  |  |
| AT004 | √ | √ | √ | √ | √ | √ |
| AT005 | √ |  |  |  |  | √ |
| AT006 | √ |  |  |  |  | √ |
| AT007 |  |  |  |  |  | √ |
| AT008 | √ | √ | √ |  |  |  |
| AT009 |  | √ | √ |  |  |  |
| AT010 |  |  |  | √ |  |  |
| AT011 | √ | √ | √ |  |  |  |
| AT012 |  | √ |  | √ |  |  |

## 9. Descripción del negocio de la solución.

En este apartado se describirá el modelo de negocio del programa.

El modelo de negocio de la aplicación está constituido por una aplicación de escritorio multiplataforma con interfaz gráfico, que se ejecuta en el ordenador del cliente.

### 9.1.1. 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.

### 9.1.2 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.

## 10. Escenarios de calidad

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Nº | Fuente | Estímulo | Entorno | Artefacto | Respuesta | Medición de respuesta | Atributo de calidad |
| 1 | Usuario jugador | Contestar a una pregunta | Producción. | GUI | Mostrar las alternativas posibles. | La selección se debe hacer con un solo clic. | AT001 |
| 2 | Usuario  jugador | Mover a una casilla. | Producción. | GUI | Moverse hasta una casilla. | El movimiento debe ser animado y menor a 1 segundo. | AT002  AT001 |
| 3 | Usuario  jugador | Lanzar el dado | Producción. | GUI | Sacar un numero del dado. | Se obtiene un numero del dado  Con apenas un clic y después de una animación en menos de 1 segundo. | AT003  AT002 |
| 4 | Desarrolladores | Cambiar la implementación de una vista. | Desarrollo | GUI | Aceptación e integración de la nueva vista. | Ningún cambio en las demás capas de la arquitectura. | AT008 |
| 5 | Usuario jugador | Ver informaciones de las partidas, sin privilegios. | Producción. | Lógica | Acceso a la información. | Acceso denegado por no tener suficientes permisos. | AT004 |
| 6 | Usuario administrador. | Modificar un valor estadístico. | Producción. | Lógica. | Datos intactos. | 100% de los datos siguen intactos. | AT005 |
| 7 | Intento no autorizado de acceso a BBDD | Intento acceso a BBDD | Producción. | Persistencia. | Acceso no autorizado rechazado | No se ha realizado ningún cambio en la base de datos | AT007  AT006 |
| 8 | Desarrolladores | Modificación en la lógica. | Desarrollo | Lógica. | Compilar el código | Capa de persistencia y GUI no necesitan ser compiladas. | AT009 |
| 9 | Aplicación | Terminar partida. | Producción. | Persistencia. | No perder información de las partidas y usuarios. | Se mantiene el 100% de la información. | AT005 |
| 10 | Usuario jugador | Mover a una casilla. | Producción. | GUI. | Ver las opciones posibles. | Los destinos se destacan visiblemente. | AT010 |
| 11 | Desarrolladores | Añadir una nueva vista. | Desarrollo. | Lógica. | Funcionamiento correcto. | No es necesario modificar ningún aspecto de la lógica. | AT011 |
| 12 | Usuario jugador | Hacer un movimiento. | Producción. | Lógica. | Devolver los posibles movimientos. | El algoritmo debe tardar menos de 2 segundos. | AT012 |