Plan de Prueba

**Ceballos Martínez Enrique**

**Martínez Sosa Jatniel Jasdekj**

**Prueba de Software**

**27/09/17**

# Introducción

El proyecto MingMang será sometido a pruebas de software para verificar y validar que el juego cumpla con los requerimientos establecidos en el ERS de este proyecto y con las expectativas de los stakeholders; así como para garantizar un producto y trabajo de calidad aceptable y sin errores.

Nuestro proyecto se tratará del juego MingMang y será construido en JavaFX. Este sistema será elaborado como parte del proyecto final de la experiencia educativa Tecnologías para la Construcción de Software. Los clientes que utilizarán este sistema serán los integrantes del grupo ajenos a la construcción de este sistema y el académico de la experiencia educativa antes mencionada.

El objetivo de la prueba es checar si el juego funciona de manera correcta: si las fichas se mueven correctamente, si existe conexión entre un equipo y otro, si los marcadores están actualizados, si el sistema en sí está bien traducido.

El plan de prueba que se detallará a continuación estará dividido en las siguientes secciones:

* **Límites.** Establece que abarcará la prueba y qué no:
  + **Alcance.** Indica el rango en el cual se llevará la prueba (qué sí y qué no), en términos de:
    - Enfoques a probar.
    - Elementos a probar.
  + **Definiciones.** Glosario que permite a cualquier persona (aún sin ser especialista en ingeniería de software) entender los conceptos que serán usados en el documento del plan.
  + **Entorno.** Identificar la ubicación física del equipo de desarrollo y la(s) ubicación(es) de los grupos de prueba, así como la organización.
* **Riesgos de calidad.** Relaciona de manera sintética los riesgos de calidad para lo que se sugiere organizar dichos riesgos con base a la tabla anexa.
* **Propuesta de plan por hito.** Desglosa los hitos para la prueba y las fechas relacionadas.
* **Transiciones.** Para cada fase, el sistema bajo prueba debe satisfacer un conjunto mínimo de calificaciones antes de que el grupo de prueba pueda ejecutar efectiva y eficientemente las pruebas; se deben listar.
  + **Criterio de entrada.** Los requisitos que deben cumplirse para iniciar una fase particular de la prueba, considerando contestar las siguientes preguntas:
    - ¿Tenemos la información necesaria de documentación, diseño y requerimientos disponibles para que los testers operen el sistema y puedan juzgar si su comportamiento es correcto?
    - ¿El sistema está listo para ejecutar la fase de la prueba que pretendemos iniciar?
    - ¿Está disponible el soporte sobre las utilidades, accesorios y prerrequisitos de tal forma que los testers los puedan utilizar?
    - ¿Tiene el sistema un nivel de calidad adecuado?
    - ¿Está listo el ambiente de prueba (laboratorio, hardware, software y sistema de soporte administrativo)?
  + **Criterio de continuación.** Define las condiciones y situaciones que deben prevalecer, mismas que permiten continuar de manera eficaz y eficiente el proceso de prueba.

Estos criterios hablan de la estabilidad requerida para que la prueba pueda continuar sin necesidad de ser redefinida. En términos del código, las estructuras de datos y las bases de datos.

En algunos casos se ponen límites con relación a la estabilidad del propio software que se está probando, pues un exceso de defecto puede hablar de un trabajo inconcluso.

* + **Criterio de salida.** Indican como determinar cuándo se ha terminado un hito de una prueba.
* **Desarrollo de pruebas.** Los conjuntos de pruebas o casos de prueba pueden generarse con base a la especificación del sistema.

En algunos casos se re-ejecutan conjuntos de pruebas previamente elaborados.

Los objetos que se pueden incluir son: casos de pruebas, herramientas de pruebas, procedimientos de pruebas, conjuntos de pruebas, scripts de pruebas automatizadas, incluso conjuntos de información para pruebas (BD) y otros elementos.

A nivel de plan de prueba se deben indicar qué tipos de objetos serán creados, con qué método y artefactos/templates.

* **Configuración y ambiente de pruebas.** En esta sección se documenta qué hardware, software, redes y laboratorio, será utilizado para la prueba.

Para cada elemento se debe indicar su configuración.

De igual forma, se deben identificar versiones de software base, configuración de manejadores de bases de datos.

* **Ejecución de pruebas.** Esta es una sección relevante, pues se definirán algunos aspectos que puedan afectar el performance del proceso de prueba que será ejecutado:
  + El orden en que serán ejecutadas las pruebas
  + Si se recibirán recursos externos
  + Recursos indispensables para la ejecución de la prueba
  + Los sistemas a probar

Se definen los datos que serán obtenidos de las pruebas, cómo serán procesados, analizados y reportados.

* + **Recursos.** Se definen los participantes clave en el esfuerzo de la prueba y los roles que cubrirán y algunos otros recursos que no se hubieran ya mencionado en el plan. Esto incluye las personas externas que participarán y sus datos de contacto. Se puede incluir una síntesis de las responsabilidades y a quién reportará.
  + **Casos de prueba y rastreo de defectos.** En esta subsección se define la forma en que se gestionará y dará seguimiento a los casos de prueba y los defectos. Esto es, la hoja de cálculo, base de datos o herramienta que se utilizará para llevar un control.

Se puede describir el proceso con dos secciones:

* + - Proceso.
      * Identificación del proceso
      * Diagrama de flujo del proceso
      * Descripción del proceso
    - Instrumentos.
      * Identificación del instrumento
      * Guía de llenado del instrumento
  + **Identificación y clasificación de defectos.** La identificación de defectos se logra por medio de la ejecución del software bajo prueba con los casos diseñados, para encontrar las variables relacionadas o posibles causas.

Identificar, por lo tanto, también permite “aislar” el defecto; esto ayuda a los desarrolladores en la tarea de resolver los defectos.

Como respuesta a los defectos reportados se podría tener el siguiente estatus:

* *Falla en requerimientos:* El defecto reportado se relaciona con un fallo en el sistema que proviene de los requerimientos. Deberá ser resuelto.
* *Falla no relacionada con requerimientos:* El reporte de defecto no se relaciona con los requerimientos del sistema, pero puede afectar significativamente la calidad del sistema, por lo que no sería aceptable. Deberá ser resuelto.
* *Solicitud denegada:* Aunque el defecto reportado describe un fallo, no será resuelto pues a criterio de los desarrolladores, no afectará de manera significativa la experiencia del usuario ni la del cliente.
* *Fallo externo:* El defecto reportado es una falla proveniente de un factor externo más allá del control del sistema bajo prueba.
* *Fallo de la prueba:* Los desarrolladores observan que la prueba da resultados incorrectos, lo cual invalida el error reportado.

En algunos casos se asignan prioridades para poder dedicar al equipo de desarrollo en la reparación de los defectos.

* + **Gestión de liberaciones de pruebas.** Debe incluirse al equipo de prueba en la planeación de las liberaciones de versiones o hitos de desarrollo.

Se debe planear considerando en que no sólo se ejecuta la prueba, sino que deben repararse los defectos.

Cuando existen actualizaciones de software o nuevas liberaciones se deben realizar las pruebas que él mismo ya había pasado (pues una modificación puede inyectar algún error en una sección ya aprobada), estas son las *pruebas de regresión*.

Igual si se han reparado algunos errores, deben realizarse de nuevo los conjuntos de prueba, pues puede ser que la reparación inserte condiciones diferentes que cambien los resultados.

* + **Ciclos de pruebas.** Un ciclo de prueba es la ejecución de uno, más de uno o todos los conjuntos de prueba planeados a ejecutar la fase de pruebas.

Se pueden asociar un ciclo de prueba con una liberación de pruebas.

* + **Horas de pruebas.** En algunos proyectos es necesario definir las horas que serán dedicadas a las pruebas.

Se pueden utilizar múltiplos de 8 horas por día hábil.

Si se tiene algún parámetro de productividad que indique el número de horas efectivas a contar puede aplicarse.

* **Riesgos y contingencias.** Si bien es cierto que la prueba es un elemento para gestionar los riesgos, la prueba como proceso también es un sujeto de riesgo.

Pueden suceder cosas que dificulten o imposibiliten la ejecución del plan de pruebas, por lo que deben realizarse un plan de riesgos del proceso de pruebas, que considere acciones para:

* + *Mitigar los riesgos.* Esto es, reducir la probabilidad de que ocurra.
  + *Generar un plan de contingencia.* Es decir, acciones que una vez que ocurra un riesgo, disminuya su impacto.
  + *Transferir el riesgo*. Esto es, identificar qué otros miembros del equipo pueden aceptar la consecuencia del riesgo ocurrido y cómo lo manejarían.
  + *Aceptar o ignorar.* Identificar que riesgos no pueden ser gestionados y sólo deben ser asumidos.
* **Histórico de cambios.** Esta parte del plan de prueba registra los cambios y revisiones del propio documento del plan de prueba, indicando:
  + Número de revisión
  + Nombre de la persona que realiza los cambios
  + Cuáles son los cambios
  + La fecha de la liberación del cambio
* **Documentos referidos.** Es una lista de otros documentos que son referidos en el plan de prueba.

Se pueden codificar con un “nick”, el cual será empleado en todo el documento para su referencia.

* **Preguntas frecuentes.** Sección de apoyo para las personas no conocedoras, que pudieran tener acceso al plan y su correcto entendimiento.

# Límites

## Alcance

La prueba abarcará la funcionalidad de lo siguiente:

* El sistema guarda el registro del nuevo jugador.
* El inicio de sesión se realiza de manera correcta.
* Se ha creado la partida con éxito.
* El jugador se unió exitosamente a la partida.
* En el caso de que la partida a unirse requiera de una contraseña, el sistema debe validar correctamente dicha contraseña.
* La partida se está ejecutando correctamente (las fichas se mueven de acuerdo a las reglas, y los turnos se van alternando de acuerdo a las reglas).
* La victoria de un jugador se concederá de acuerdo a las reglas del juego.
* Actualización de los marcadores tras la finalización de la partida.
* El sistema se ejecuta en inglés y en español.

No se evaluará la cantidad de memoria RAM que consumirá la aplicación, ni la cantidad de memoria en disco que se requerirá para su instalación.

## Definiciones

* **Ming Mang:** Juego de mesa de estrategia de dos jugadores originario de Tibet.

## Entorno

El equipo de desarrollo se ubicará con frecuencia en la Facultad de Estadística e Informática, localizada en Xalapa; así como en las respectivas residencias de los integrantes del equipo. La prueba se ejecutará en los equipos de cómputo de todos los integrantes del equipo.

# Riesgos de calidad

|  |  |  |
| --- | --- | --- |
| CATEGORÍA DE RIESGO DE CALIDAD | ELEMENTO DE RIESGO | ESTRATEGIA DE PRUEBA |
| Validación de datos | Error al tratar de verificar si los datos ingresados fueron los correctos. | * Evaluar uno por uno los datos que el usuario ingresó. |
| Rendimiento del software | El software responde a los eventos con tardanza. | * Recomendar algo de paciencia a los usuarios del juego. |
| Respuesta errónea a los eventos ocurridos | El software puede responder a los eventos de forma inesperada | * Ir probando cada evento con cuidado para verificar si el evento dio un resultado correcto. |

# Propuesta de plan por hito

|  |  |
| --- | --- |
| **HITO** | **FECHA** |
| Realizar la plantilla de plan de prueba | 31/10/2017 |
| Funcionalidad completa del juego | 07/11/2017 |
| Ejecución de los casos de prueba | 20/11/2017 |

# Transiciones

## Criterio de entrada

La prueba de sistema iniciará con:

* Se haya evaluado este plan de prueba.
* Cuando el sistema a evaluar esté construido en su totalidad.

## Criterio de continuación

La prueba de sistema se realizará mientras:

* El periodo de prueba se encuentre en curso.
* Se realicen evaluaciones periódicas de los trabajos hechos durante el curso de Prueba de Software.

## Criterio de salida

La prueba del sistema se terminará cuando:

* La docente de Prueba de Software lo indique.
* El periodo semestral agosto 2017 – enero 2018 concluya.
* Las pruebas realizadas hayan sido exitosas.

# Desarrollo de pruebas

|  |  |
| --- | --- |
| **PRUEBA A REALIZAR** | **EVALUACIÓN** |
| Internacionalización correcta del sistema (en español y en inglés) |  |
| Verificación y validación de registro |  |
| Verificación y validación de inicio de sesión |  |
| Redireccionamiento a otras interfaces mediante los botones |  |
| Creación exitosa de la partida |  |
| Acceso exitoso a la partida compartida |  |
| Despliegue de todas las partidas disponibles |  |
| El transcurso de juego sigue sus reglas (Las fichas se mueven como deben, el cambio de color de las fichas es el debido y la victoria del ganador es correcta y justa) |  |
| Los marcadores se van actualizando conforme van terminando las partidas de los jugadores. |  |

# Configuración y ambiente de pruebas

# Ejecución de pruebas

## Recursos

|  |  |  |
| --- | --- | --- |
| **PERSONA** | **ROL** | **DATOS DE CONTACTO** |
| Ceballos Martínez Enrique | Analista, diseñador, programador y evaluador | ceballos\_quiq@hotmail.com |
| Martínez Sosa Jatniel Jasdekj | Analista, diseñador, programador y evaluador | jatnielmtz@gmail.com |

## Casos de prueba y rastreo de defectos

## Identificación y clasificación de defectos

## Gestión de liberaciones de pruebas

## Ciclos de prueba

## Horas de prueba

# Riesgos y contingencias

# Historial de cambios

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Autor** | **Actualización** |
| **27/09/17** | **1.0** | **Ceballos Martínez Enrique, Martínez Sosa Jatniel Jasdekj** |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Documentos referidos

# Preguntas frecuentes