**Logotipo

Descripción generada automáticamente con confianza media**

**INSTITUTO TECNOLÓGICO DE LOS MOCHIS.**

Ingeniería Informática

**Materias:**

Administración de proyectos, programación ambiente cliente-servidor.

**Trabajo:**

Juego Batalla Naval.

**Equipo:**

Castro López Andrés – 19440377.

Jiménez Bacasegua José Carlos – 19440094.

Villa Zavala José Mario – 19440189.

**Profesor:**

  ALGARA NORZAGARAY JUAN FRANCISCO

**Índice**

[**Nombre del proyecto:** 3](#_Toc121244742)

[**Descripción del proyecto:** 3](#_Toc121244743)

[**Alcance del proyecto:** 3](#_Toc121244744)

[**Roles:** 4](#_Toc121244745)

[**Descripción de los roles:** 4](#_Toc121244746)

[**WBS:** 5](#_Toc121244747)

[**RBS:** 6](#_Toc121244748)

[**Historias de uso.** 8](#_Toc121244749)

[**Calendarización.** 11](#_Toc121244750)

[**Bitácora de errores durante el desarrollo:** 12](#_Toc121244751)

[**Resultados:** 13](#_Toc121244752)

**Nombre del proyecto:** Sistema de comunicación por medio de consola de comandos para juego Batalla Naval.

**Descripción del proyecto:** El proyecto consta de un sistema de información en donde se implementa la conexión cliente-servidor, para que varios clientes puedan jugar entre ellos Batalla Naval, el sistema está construido con el lenguaje de programación JAVA para ser ejecutado en la consola de comandos de Windows.

# **Alcance del proyecto:**

* El sistema tiene que funcionar en la línea de comandos.
* Debe ser multihilo.
* Asíncrono.
* Debe permitir jugar de forma remota y simultánea.
* El juego debe cumplir con las reglas básicas de Batalla Naval.
* La interacción debe de darse mediante comandos o preguntas al servidor, algunos ejemplos de comandos son:
* Mostrar los tableros.
* Tabla de ganadores.
* Mostrar peor enemigo.
* A quien le has ganado más.
* Un jugador puede tener varias partidas simultáneas con varios jugadores, pero no 2 veces con el mismo.
* Para iniciar una partida se manda una invitación a otro jugador que puede o no ser aceptada.
* Durante una partida uno de los participantes se puede rendir.
* Si durante una partida el servidor se apaga no se registra nada.
* Guardar registro de victorias (cada victoria vale 3 puntos).

# **Roles:**

Jiménez Bacasegua José Carlos: Programador.

Castro López Andrés: Diseñador de la arquitectura del programa, tester.

Villa Zavala José Mario: Administrador del proyecto, documentador.

# **Descripción de los roles:**

**Administrador del proyecto:** Las responsabilidades del administrador del proyecto son la de estudiar que tan viable es el desarrollo del sistema, los objetivos del proyecto, qué recursos se disponen para dicho desarrollo, que actividades se tienen que asignar a los diferentes roles del proyecto y sus fechas.

**Programador:** La responsabilidad del programador es plasmar las funcionalidades y arquitectura planteada del sistema, en código de programación para que el proyecto vaya cumpliendo con los objetivos.

**Diseñador de la arquitectura del programa:** La responsabilidad del diseñador de la arquitectura es plantear la lógica del sistema, las interacciones de los usuarios con el sistema de información.

**Documentador:** La responsabilidad de este rol es de asegurar el control de toda la documentación que surja con la historia del proyecto. La importancia de la documentación es tener un registro de la historia del proyecto, para que así los demás roles estén al tanto de los avances de las fases y se reducir la distorsión de ideas en el proceso de desarrollo.

**Téster:** Se trata del responsable de los procesos de detección y eliminación de errores o defectos que existan en el proceso de la construcción del software.

# **WBS:**

**SISTEMA DE JUEGO BATALLA NAVAL EN CONSOLA**

DOCUMENTACION

DESARROLLO

IMPLEMENTACION

Calendarización

Asignación de roles

Descripción del proyecto

Alcance del proyecto

WBS

Historias de uso

RBS

Repositorio del proyecto en GitHub

Estructuración de clases del programa

Codificación en java

Pruebas

Entrega del proyecto

# **RBS:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Proyecto: Batalla Naval en chat multihilos.** | **Técnicos** | **Requisitos** | Falta de cumplir requisitos del proyecto | 40% |
| **Tecnología** | Fallas en la implementación de tecnologías que se involucren en el proyecto | 30% |
| Fallas en el equipo de trabajo | 20% |
| Falta de experiencia utilizando tecnologías que se requieran | 50% |
| Perdida de avances del proyecto **\*** | 20% |
| Malas practicas | 40% |
| **Calidad** | Mala optimización del proyecto | 40% |
| **Externos** | **Académicos** | Destinación de tiempo a actividades o tareas de otras materias | 20% |
| **Salud** | Enfermedad de un miembro del  equipo | 30% |
| **Gestión del proyecto** | **Estimación** | Mala estimación de tiempos \* | 45% |
| Adelantar la fecha de entrega \* | 10% |
| **Planificación** | Horarios incompatibles | 10% |
| Mala organización de las actividades | 25% |
| **Comunicación** | Mala organización entre los  miembros del equipo \* | 30% |
| Problemas de comunicación entre  los miembros | 30% |
| Falla de un miembro del equipo | 30% |

(\*) Riesgos críticos.

(%) probabilidad de que ocurran.

Los riesgos con menos del 40% de que suceda se aceptan.

# **Historias de uso.**

|  |  |  |
| --- | --- | --- |
| **Historia de Uso** | **Descripción** | **Puntos**  **(días de trabajo** |
| **Servidor para chat multihilos** | Crear un proyecto que será el servidor que manejará los hilos de comunicación entre clientes para que puedan mandar mensajes entre si | 1 |
| **Cliente para chat multihilos** | Crear proyecto que será el cliente que se conectará al servidor, que se comunique con otros clientes del mismo tipo conectados a dicho servidor. Cada cliente ingresa su nombre de usuario al conectarse, si ya existe no lo deja entrar | 1 |
| **Mandar mensajes a un solo cliente** | Desde un cliente, escribir el mensaje que quiere enviar y el usuario al que desea mandarle un mensaje privado, por ejemplo (“hola juan @juan”), el mensaje solo se imprimirá al usuario que se le especificó | 1 |
| **Establecer comandos** | El cliente podrá mandar comandos para activar el juego de batalla naval y responder la solicitud de juego, escribiendo en mayúsculas el comando deseado (por ejemplo, el comando jugar, “JUGAR @juan”) | 1 |
| **Manejar comandos** | Cuando el cliente mande un mensaje al servidor, debe verificar si es un mensaje normal o un comando, en caso de ser comando, ejecutar el tipo de comando que se le envió | 2 |
| **Iniciar Juego Batalla naval** | Al escribir el comando “JUGAR @usuario” le debe mandar una solicitud de aceptar juego, si el otro usuario la acepta (con el comando “SI”) inicia el juego, le envía al cliente que envió la solicitud que fue aceptada e inicia el juego. Si no la acepta (con el comando “NO”), solo regresa al cliente que envió la solicitud que fue rechazada. | 1 |
| **Elegir cantidad de Barcos** | Al aceptar una solicitud de juego, se inicia el juego de batalla Naval y se manda a llamar un método para elegir la cantidad de barcos que vamos a usar en nuestra partida, dando como opción 3 tipos de elección con diferente cantidad de barcos cada una. | 2 |
| **Ingresar coordenadas de barcos propios** | Una vez que se eligió la opción de cantidad de barcos, empieza a llenar las coordenadas en donde se acomodaran los barcos propios, ingresando la coordenada inicial y final de cada barco y guardarlas en el tablero, una vez guardadas las coordenadas, mandar al servidor la matriz con los barcos para que el otro jugador pueda disparar. Una vez que se establezcan las posiciones de los barcos, enviar un mensaje al servidor para indicar al otro jugador que estas listo para empezar la partida | 3 |
| **Identificar coordenadas** | Para saber si lo que estamos mandando es una coordenada o un mensaje normal, es necesario verificar que cumpla con las condiciones para una coordenada, si cumple, entonces si te deje ingresar dicha coordenada en la posición de los barcos o disparos que realices | 2 |
| **Creación de tablero de juego** | Crear un tablero de 10 x 10 rellenado con [0] y [1], en donde las casillas que contengan [0] representan el agua y las casillas con [1] representan una posición del barco | 2 |
| Comando Mostrar tableros | Establecer como nuevo comando, un comando TABLEROS que, al mandarlo, te muestre el tablero del enemigo y el tablero con las posiciones atinadas y el tablero propio con tus barcos elegidos | 3 |
| Disparar al enemigo | Ya que ambos jugadores eligieron las posiciones de sus barcos, verifica si ambos están listos para jugar y empiezan a disparar por turnos, empezando por el jugador que mando la solicitud, ingresa la coordenada donde va a disparar y una vez que la ingresa, le dice si atino el disparo o si cayó al agua, si atinó, en su tablero enemigo se muestra la posición atinada, termina su turno y espera al disparo del otro jugador | 3 |
| Terminar juego | El juego se termina una vez que alguno de los dos jugadores haya atinado todas las posiciones del contrario o alguno se haya desconectado. El jugador que haya atinado las posiciones se convierte en el ganador de la partida y obtiene 3 puntos. En caso que alguno se haya desconectado, solo se sale del juego y no guarda nada | 1 |

# **Calendarización.**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Proyecto: Batalla naval** | | | | | | | | | | | | | | | |
| **Actividad** | **Día** | | | | | | | | | | | | | | |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** |
| **Definir descripción del proyecto** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Definir alcance del proyecto** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Asignación de roles** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Calendarización de actividades** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Crear historias de usuario** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Crear WBS** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Crear RBS** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Construir el repositorio del proyecto** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Estructuración de clases del programa** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Codificación en java** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Pruebas** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Entrega del proyecto** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

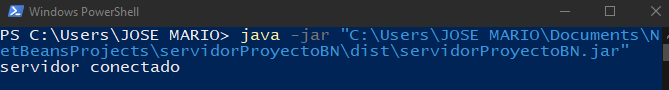
# **Bitácora de errores durante el desarrollo:**

|  |  |  |
| --- | --- | --- |
| Error | Descripción | fecha |
| 1 | El cliente no reconocía cuando lo que ingresaba o recibía era un comando o cuando era mensaje normal. | 28/11/2022 |
| 2 | No se podía acceder al valor de la variable booleana “solicitud” que es true si llega un mensaje de solicitud de juego de otro jugador. | 28/11/2022 |
| 3 | Error al declarar variables, no estaban inicializados los arreglos que verificaban si los mensajes ingresados eran comandos. | 29/11/2022 |
| 4 | Error- al ingresar las coordenadas propias, el hilo para enviar era el que entraba y mandaba la coordenada al server, y el cliente no la leía. | 1/12/2022 |
| 5 | Error al llenar las posiciones de los barcos, solo se rellenaban dos posiciones de cada barco. | 2/12/2022 |
| 6 | Error al imprimir la matriz con el tablero- no se mostraba en forma de cuadricula. | 3/12/2022 |

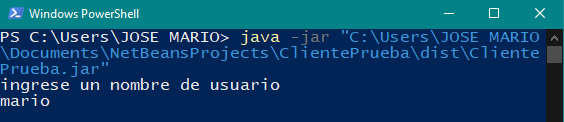
# **Resultados:**

En esta sección se muestran algunas capturas de pantalla sobre el funcionamiento del proyecto.

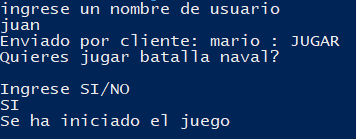
1. Se inicia el servidor en la consola de comandos, con el comando java -jar y la ruta donde esté ubicado el archivo ejecutable .jar del servidor.



1. Ya en otra consola, se ingresa el mismo comando, pero para ejecutar el .jar del cliente, una vez en ejecución, el programa nos pedirá un nombre de usuario para identificarnos y le damos enter.



1. Para jugar una partida con otro usuario es necesario el uso del comando JUGAR @” nombre de usuario” para que le llegue la solicitud de juego al otro usuario, por ejemplo, del usuario “mario” le mandamos solicitud al usuario “juan” en donde puede aceptar o no, en este caso daremos que SI.



1. Una vez que el cliente juan ha aceptado la solicitud, el cliente mario debe ingresar la posición en donde se quiere colocar los barcos en sus respectivas coordenadas, para posicionar, se utiliza una coordenada inicial y una coordenada final, y los espacios en medio se rellenan en el tablero correspondiente.

