

# Universidad Autónoma del Estado de México

# Unidad Académica Profesional Tianguistenco

## Ingeniería en Software

## Unidad de Aprendizaje:

# Sistemas Distribuidos

## Profesor:

# Jovani Armaga García

## Titulo:

# Proyecto: Juego de Ahorcado

López Palma Carlos Jonathan ....... N.C: 1413313

David Velázquez Ramírez

Uriel García Martínez

Edwin Misael Vázquez Rivera

César Pedraza Hernández …........ N.C: 1774280

### índice

Antecedentes 2

Herramientas por utilizar 3

Hipótesis 3

Planteamiento del problema 3

Videojuegos y sistemas distribuidos 2

## Juego Ahorcado

### Antecedentes

Historia del juego del ahorcado

Al parecer surgió en la época victoriana," dice Tony Augarde autor de "La Guía de Oxford de Juegos de palabras" (Oxford University Press).

El juego es mencionado en 1894 en "The Traditional Games of England, Scotland, and Ireland - Vol I" (Juegos tradicionales de Inglaterra, Escocia e Irlanda) ​ de Alice Bertha Gomme bajo el nombre "Birds, Beasts, and Fishes" (Pájaros, Bestias y Peces). Las reglas eran simples: un jugador anota la primera y última letra de una palabra de un animal, y el otro jugador adivina las letras en el medio.

En otras fuentes el juego se llama "Horca", "El Juego de Hangin'", o "Suspensión".

El Ahorcado ha aparecido en el sistema de videojuegos Speak & Spell de 1978 bajo el nombre de "Palabra Misterio" y en la actualidad (2014), a veces, se juega en los foros de Internet.

### Videojuegos y sistemas distribuidos

La posibilidad de utilizar un ordenador para jugar en red comenzó en torno a 1979, cuando un grupo de estudiantes de la Universidad de Essex crearon una versión informática multiusuario de un juego de rol llamado “Dungeons & Dragons”, basado en una interfaz de texto mediante el cual los usuarios podían ir avanzando en la historia. Así surgió un nuevo tipo de juegos conocidos como MUD (Multi-User Dungeons o Domains) que se desarrollaría rápidamente por la aún poco conocida Internet, surgiendo así las primeras comunidades virtuales.

El primer juego multiusuario que incorporó imágenes fue Hábitat en 1986, creado por Lucas Films Games y destinado para el Commodore 64. De él surgieron posteriormente juegos como el EverQuest, Asheron’s o Ultima Online.

Pero la verdadera revolución de los juegos en red surgió en 1993 con la creación de la World Wide Web. Los usuarios tenían la posibilidad de acceder gratuitamente a versiones reducidas de videojuegos para ordenador con fines básicamente promocionales, como es el caso de la primera entrega del Doom. Además, la rápida difusión de Internet como medio de entretenimiento facilitó la mejora de las tecnologías para la conexión en red de usuarios y su acercamiento a la sociedad.

### Planteamiento del problema

Se plantea la creación de un videojuego basado en el juego del ahorcado, con la finalidad de que más de dos personas puedan interactuar mediante una red informática, donde se espera podrán jugar desde distintos equipos de cómputo mediante una aplicación.

### Hipótesis

La programación en red siempre ha sido dificultosa, el programador debía de conocer la mayoría de los detalles de la red, incluyendo el hardware utilizado, los distintos niveles en que se divide la capa de red, las librerías necesarias para programar en cada capa, etc.

Pero, la idea simplemente consiste en obtener información desde otra máquina, aportada por otra aplicación software. Por lo tanto, de cierto modo se puede reducir al mero hecho de leer y escribir archivos, con ciertas salvedades.

Es por eso por lo que se plantea utilizar la tecnología Java para comunicar diversos ordenadores y que puedan participar en un juego de Ahorcado

### Herramientas por utilizar

Una de las características más importantes de Java es su capacidad y, a la vez, facilidad para realizar aplicaciones que funcionen en red. La mayoría de los detalles de implementación a bajo nivel están ocultos y son tratados de forma transparente por la JVM (Java Virtual Machine).

Los programas son independientes de la arquitectura y se ejecutan indistintamente en una gran variedad de equipos con diferentes tipos de microprocesadores y sistemas operativos.

En las aplicaciones en red es muy común el paradigma cliente-servidor. El servidor es el que espera las conexiones del cliente (en un lugar claramente definido) y el cliente es el que lanza las peticiones a la maquina donde se está ejecutando el servidor, y al lugar donde está esperando el servidor (el puerto(s) específico que atiende). Una vez establecida la conexión, ésta es tratada como un stream (flujo) típico de entrada/salida.

A través de las clases del paquete java.net, los programas Java pueden utilizar TCP o UDP para comunicarse a través de Internet. Las clases URL, URLConnection, Socket, y SocketServer utilizan TCP para comunicarse a través de la Red. Las clases DatagramPacket y DatagramServer utilizan UDP.

TCP proporciona un canal de comunicación fiable punto a punto, lo que utilizan para comunicarse las aplicaciones cliente-servidor en Internet. Las clases Socket y ServerSocket del paquete java.net proporcionan un canal de comunicación independiente del sistema utilizando TCP, cada una de las cuales implementa el lado del cliente y el servidor respectivamente.

Así el paquete java.net proporciona, entre otras, las siguientes clases, que son las que se verán con detalle:

- Socket: Implementa un extremo de la conexión TCP.

- ServerSocket: Se encarga de implementar el extremo Servidor de la conexión en la que se esperarán las conexiones de los clientes.

- DatagramSocket: Implementa tanto el servidor como el cliente cuando se utiliza UDP.

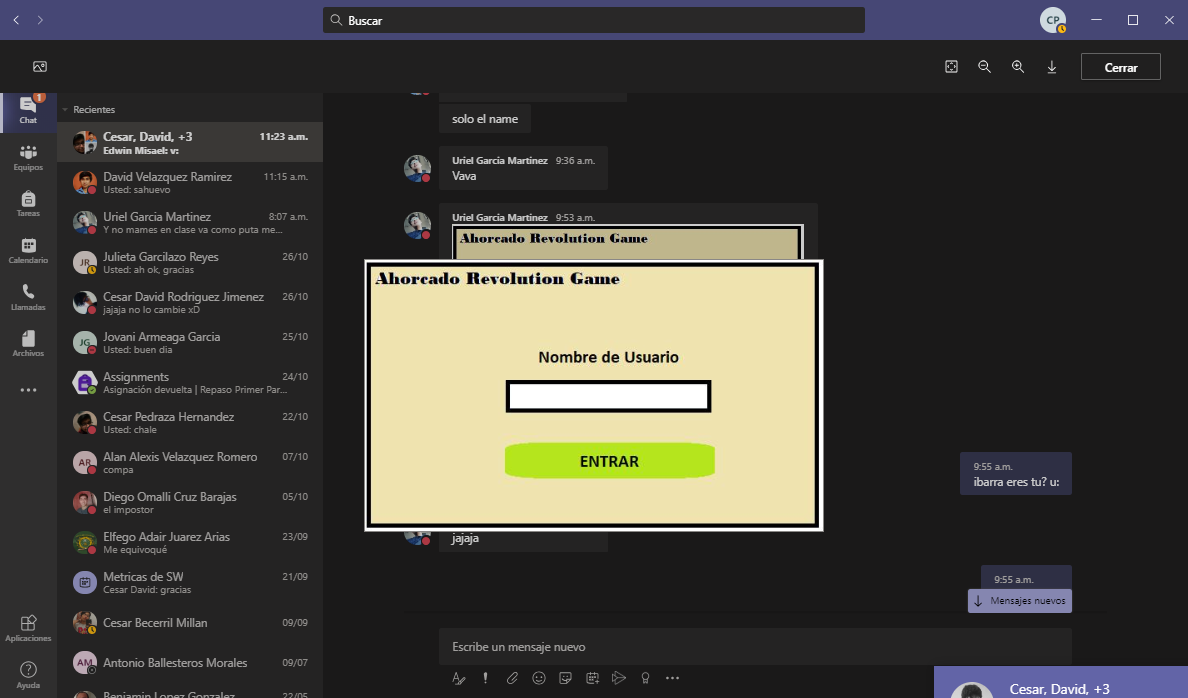
- DatagramPacket: Implementa un datagram packet, que se utiliza para la creación de servicios de reparto de paquetes sin conexión.

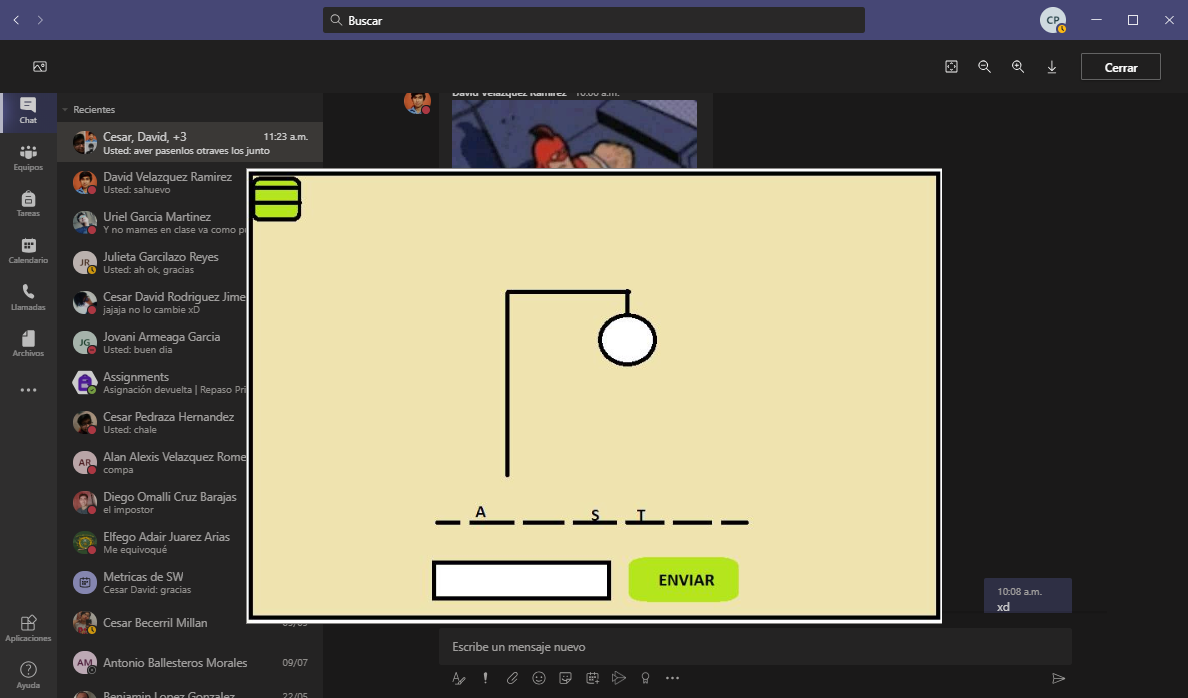
Sistemas Distribuidos: Sockets en Java EUI-SG/INFOR.UVA.ES 3

- InetAddress: Se encarga de implementar la dirección IP.

Se utilizará el IDE Netbeans ya que puede ser usada para desarrollar cualquier tipo de aplicación, Reutilización de módulos, Permite el uso de herramientas AWT para generacion de interfaces y requiere de una Instalación y actualización simple.

**Diseño de prototipo**





|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | RF1 | | |
| **Nombre** | Ingresar usuario | | |
| **Descripción** | El usuario debe ingresar un nombre de jugador, para entrar a una partida. | | |
| **Funcionamiento** | El usuario debe ingresar en un campo de texto (textfield) un nombre de usuario que lo identifique dentro de la sala.  El nombre debe cumplir con las siguientes características:   * Longitud mínima: 5 * Longitud máxima: 15 * El nombre puede llevar letras (a-z, A-Z), dígitos (0-9) y caracteres especiales (. (punto) - (guion), \_ (guion bajo)). * No usar caracteres que no se encuentren entre los enumerados anteriormente.   Botón entrar: al presionar, muestra la pantalla de sala.  Casos:  Al entrar a la sala de partida, se comprueba que el nombre cumpla con las especificaciones anteriores, de lo contrario, se mantiene en la pantalla de Ingresar usuario y muestra un mensaje con el problema. Si el nombre ingresado es el mismo que el de un jugador dentro de la sala de partida, muestra un mensaje al usuario, solicitando cambiar su nombre.  Si el campo está en blanco, no podrá ingresar a la sala.  Para entrar a la sala, se debe presionar el botón “entrar”. | | |
| **Prioridad de Requerimiento** | Alta | **Exigencia de requerimiento** | Exigible |
| **Relación** | RF2 | | |
| **Editor** | Carlos Jonathan López Palma | | |
| **Fecha** | 21 de octubre de 2020 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | RF2 | | |
| **Nombre** | Sala de partida | | |
| **Descripción** | Se inicia una sala cuando un usuario entra a una partida. | | |
| **Funcionamiento** | La sala se inicia, cuando un usuario ingresa con su usuario, o se une a la partida si ya hay más usuarios dentro de la sala.  Una sala permite hasta 5 jugadores. Los jugadores dentro de la partida, se enlistan de lado izquierdo, en orden alfabético. Al lado derecho del nombre del jugador, se muestra su contador de puntos (Al comienzo de la partida, los puntos son 0 para todos los jugadores):   * Usuario – 0pts   La partida comienza cuando entran 5 jugadores. | | |
| **Prioridad de Requerimiento** | Alta | **Exigencia de requerimiento** | Exigible |
| **Relación** |  | | |
| **Editor** | David Velázquez Ramírez (Diego :u) | | |
| **Fecha** | 21 de octubre 2020 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | RF3 | | |
| **Nombre** | Seleccionar Jugador | | |
| **Descripción** | Al comenzar una ronda se selecciona un jugador para establecer una palabra. | | |
| **Funcionamiento** | Cuando una ronda comienza, se selecciona un jugador, para que sea el que establezca la palabra a adivinar. El jugador se selecciona tomando el orden de la lista de jugadores.  En la primera ronda, se tomará el primer jugador de la lista, y se irán seleccionando en ese orden.  El jugador seleccionado, no podrá escribir durante esta ronda letras para adivinar, pero en un textfield, aparte, puede poner pistas para el resto de jugadores. | | |
| **Prioridad de Requerimiento** | Alta | **Exigencia de requerimiento** | Exigible |
| **Relación** |  | | |
| **Editor** | Uriel García Martínez | | |
| **Fecha** | 21 de octubre 2020 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | RF4 | | |
| **Nombre** | Partida | | |
| **Descripción** | Los buscadores, deben ingresar por turnos una letra, para tratar de adivinar la palabra u oración. | | |
| **Funcionamiento** | Al entrar a la pantalla de juego, se muestran los jugadores en lista de lado izquierdo y junto al nombre, la cantidad de puntos acumulados. Al entrar a partida, los putos de un jugador se establece en 0.  Los jugadores se enlistan alfabéticamente y siguiendo el orden de esta lista se van asignando los turnos para establecer la palabra a adivinar.  La partida se iniciará cuando la cantidad de jugadores sea igual a 5 y al que se posicione en primer lugar de la lista, se le dará el primer turno para establecer la palabra u oración.  La palabra u oración establecida por el jugador seleccionado, debe cumplir las siguientes características:   * Longitud mínima: 5 * Longitud máxima: 20 * Solo puede incluir caracteres alfabéticos (A-Z) y dígitos (0-9) * Espacios en blanco no cuentan como dígitos   Los jugadores que deben adivinar deben, por turnos, ingresar en el campo una sola letra. Solo puede introducir:   * Solo puede incluir caracteres alfabéticos (A-Z) y dígitos (0-9)   No se permiten caracteres especiales | | |
| **Prioridad de Requerimiento** | Alta | **Exigencia de requerimiento** | Exigible |
| **Relación** |  | | |
| **Editor** | Cesar Pedraza | | |
| **Fecha** | 21 de octubre 2020 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | RF5 | | |
| **Nombre** | Partida - Ronda | | |
| **Descripción** | La ronda comienza cuando el jugador seleccionado ingresa la palabra u oración, y termina cuando el resto de jugadores encuentren la palabra. | | |
| **Funcionamiento** | Cuando la ronda comienza, siguiendo el orden de la lista, cada jugador buscador, podrá ingresar por turnos, un carácter en el textfield. Si el carácter coincide con algún carácter de la oración o palabra, el carácter se agrega a en la posición en la que aparece y el jugador que la adivino, gana 25 puntos.  Si los jugadores logran encontrar la respuesta, ganan la ronda y cada jugador obtiene 50 puntos.  Por cada ronda en la que ningún jugador hay adivinado una letra, se agrega una extremidad al personaje:   * Una cabeza * Dos brazos * Dos piernas * Un torso | | |
| **Prioridad de Requerimiento** | Alta | **Exigencia de requerimiento** | Exigible |
| **Relación** |  | | |
| **Editor** | Edwin Misael Velázquez | | |
| **Fecha** | 21 de octubre 2020 | | |