Práctica 1 – Aplicaciones de Internet usando API de Sockets

Objetivos

 Analizar los mecanismos de comunicación de la capa de transporte
 Emplear el modelo Cliente-Servidor para construir aplicaciones de Internet
 Programar aplicaciones de internet utilizando sockets de flujo o de mensaje

Desarrollo

Implementar una aplicación cliente y una aplicación servidor que se puedan comunicar en el dominio de Internet, usando SOCKETS.

El alumno debe diseñar un algoritmo para calcular el número de días que ha vivido hasta el 10 de marzo del 2025. Hacer la operación *R = númeroDeDías % 3*. El resultado de la operación determinará el juego que implementará:
R = 0. Buscaminas
R = 1. Gato Dummy
R = 2. Memoria
El alumno debe elegir si va a usar sockets de flujo (TCP) o sockets de mensajes (UDP) para su implementación y deberá justificar su elección.
No es necesario que el juego tenga una interfaz gráfica. Sin embargo, el cliente y el servidor tendrán un tablero "impreso" en consola.
En las aplicaciones cliente y servidor se usará el intercambio de mensajes tipo "coordenadas" y "control". Un mensaje de coordenadas es enviado del cliente al servidor para indicar la posición del tiro. Un mensaje de control es un mensaje que envía el servidor indicando el resultado del tiro, por ejemplo "Ok", "siguiente tiro". "perdiste".

Buscaminas.

Deberá implementarse una aplicación cliente con las siguientes características:

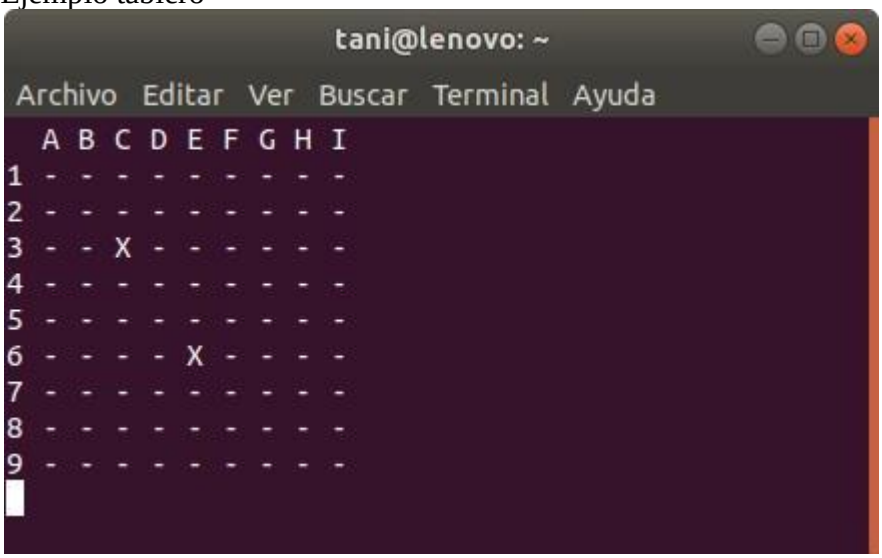
- La aplicación cliente deberá conectarse con la aplicación servidor a través de un socket (la dirección y puerto destino deberán ser proporcionados por el jugador).
- Una vez conectado, el jugador recibe un mensaje de confirmación que indica la dificultad de la partida. En ese momento se imprime un tablero local (vacío) que sólo sirve para mostrar la información que manda el servidor.
- El jugador tendrá la capacidad de descubrir una casilla. La aplicación cliente manda al servidor las **coordenadas** de la casilla para que el servidor valide su contenido. Espera la respuesta del servidor y actualiza la información del tablero.
- En el caso que descubra una casilla que contiene una mina, la aplicación cliente deberá descubrir todas las minas del tablero y notificar al jugador que ha perdido.
- En caso de el jugador quiera descubrir una casilla que ya se encuentra ocupada, la aplicación cliente debe enviar la coordenada para que el servidor pueda validarlo y recibirá un mensaje tipo "casilla ocupada" y repetirá el tiro.

- En el caso que se descubran todas las casillas del tablero, menos las que contienen minas, la aplicación cliente deberá notificar al usuario que ha ganado la partida, y mostrar la duración del juego.

Deberá implementarse una aplicación servidor con las siguientes características:

- la aplicación solicita al usuario especificar la IP y el puerto en el que recibirá solicitudes de conexión usando sockets
- La aplicación solicita la dificultad del juego, puede elegir entre:
  - principiante: tablero de 9 × 9 casillas y 10 minas.
  - avanzado: tablero de 16 × 16 casillas y 40 minas.
- Una vez iniciado el servidor, este recibirá la conexión del jugador.
- En cuanto se conecte un cliente, la aplicación servidor generará un tablero (según la dificultad) que contendrá el número de minas correspondientes a la dificultad, colocadas de forma aleatoria dentro del tablero.
- El servidor envía un mensaje de confirmación indicando que está listo para recibir y validar cada una de las acciones del jugador (destapar y validar)
- El servidor recibe las coordenadas del tiro y valida que la casilla se encuentre libre, si es así manda un mensaje de control tipo “casilla libre”. Si la casilla tiene mina, el servidor manda un mensaje tipo “mina pisada”.
- El servidor deberá registrar una marca de tiempo al inicio y al final de la partida para determinar la duración del juego.
- Al finalizar la partida, (es decir, cuando todas las minas han sido marcadas, cuando todas las casillas han sido destapadas, excepto las que contienen minas, o cuando se ha destapado una casilla que contiene mina), el servidor deberá informar al jugador si ganó o perdió la partida, así como mostrar un registro del tiempo que duró la partida.
- El servidor debe enviar un mensaje de fin de transferencia para indicar al cliente que el juego terminó.

Ejemplo tablero



Gato Dummy

Deberá implementarse una aplicación cliente con las siguientes características:

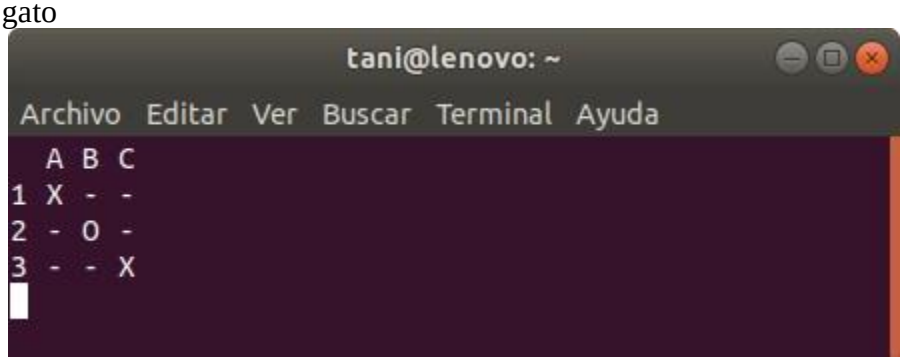
- La aplicación cliente deberá conectarse con la aplicación servidor a través de un socket (la dirección y el puerto destino deberán ser proporcionados por el jugador).
- La aplicación cliente recibe un mensaje de confirmación que indica la dificultad de la partida. En ese momento, la aplicación imprime un tablero local (vacío) que sólo sirve para mostrar la información que manda el servidor.
- En ese momento, el jugador podrá elegir una casilla donde colocará su marca. La aplicación envía la elección del jugador y espera la jugada del servidor.

- Cuando recibe una jugada del servidor, la aplicación actualiza el tablero con la información recibida, usará un caracter diferente al del usuario para poder diferenciar los tiros.
- El juego termina cuando el servidor manda la cadena de control “empate” (todas las casillas están ocupadas), o “ganaste” (cuando algún jugador logre conseguir K casillas en línea).

Deberá implementarse una aplicación servidor con las siguientes características:

- la aplicación solicita al usuario especificar la IP y el puerto en el que recibirá solicitudes de conexión usando sockets
- la aplicación solicita la dificultad del juego según el nivel:
  - principiante, tres en línea: m=3, n=3, k=3.
  - avanzado, cinco en línea: m=5, n=5, k=5.
  - En cualquier caso, el servidor construye un tablero de m x n y el objetivo es conseguir k en línea (horizontal o vertical).
- Una vez iniciado el servidor, este recibirá la conexión del cliente.
- El servidor debe enviar un mensaje de confirmación al cliente, indicando la dificultad de la partida. Posteriormente, el servidor deberá prepararse para recibir y validar cada una de las acciones del jugador (elegir casilla, validar).
- La aplicación servidor deberá registrar una marca de tiempo al inicio y al final de la partida para determinar la duración del juego.
- Durante el juego, el servidor deberá elegir una casilla de manera aleatoria. Deberá marcarla en el tablero y enviar la actualización del tablero mediante mensajes de confirmación al cliente.
- El servidor deberá validar si algún jugador logra marcar k casillas en línea. Si es el caso, el servidor finaliza la partida.
- Al finalizar la partida, (empate o k en línea), el servidor deberá informar al jugador si ganó, perdió o empató la partida, así como mostrar el tiempo de duración de la partida.

Ejemplo tablero gato



MEMORIA

Deberá implementarse una aplicación cliente con las siguientes características:

- La aplicación cliente deberá conectarse con el servidor a través de un socket (la dirección y el puerto destino deberán ser proporcionados por el jugador).
- La aplicación cliente recibe un mensaje de confirmación del servidor que contiene la dificultad del tablero. Imprime el tablero (según el mensaje de confirmación anterior) y el jugador puede elegir un par de casillas para destapar.

La aplicación muestra el contenido de estas casillas. La aplicación envía el par de casillas para que el servidor valide si el contenido es igual. En caso de que el contenido de las casillas sea igual se contará un punto a favor del jugador y podrá repetir la elección de dos casillas. En el caso contrario, será turno del servidor.

La aplicación cliente espera hasta que el servidor haya elegido sus casillas y actualiza el tablero para visualizar el contenido de las casillas correspondientes.

El juego termina cuando todas las casillas han sido destapadas

Deberá implementarse un servidor con las siguientes características:

- la aplicación solicita al usuario especificar la IP y el puerto en el que recibirá solicitudes de conexión usando sockets
- La aplicación solicita la dificultad del juego según el nivel:
  - Nivel principiante: Tablero de 4 x 4 casillas.
  - Nivel avanzado: Tablero de 6 x 6 casillas.
- Una vez iniciado el servidor, este recibirá la conexión del cliente.
- En cuanto se conecte un cliente, el servidor envía un mensaje de confirmación que contiene la dificultad. Posteriormente, genera un tablero que contendrá el número de casillas correspondiente. El contenido de las casillas serán palabras duplicadas, por ejemplo, la casilla 1 y la casilla 3 contendrán la palabra “árbol”, el jugador deberá encontrar ese par de palabras para poder hacer un punto. El servidor tendrá una colección de 8 palabras duplicadas para el nivel principiante y de 18 palabras duplicadas para el nivel avanzado. El servidor las colocará de manera aleatoria en las casillas.
- El servidor debe enviar un mensaje de confirmación al cliente indicando que está listo para iniciar la partida. Posteriormente, el servidor deberá prepararse para recibir y validar cada una de las acciones del jugador (destapar dos casillas, validar).
- La aplicación servidor deberá registrar una marca de tiempo al inicio y al final de la partida para determinar la duración del juego.
- Durante el juego, el servidor deberá elegir un par de casillas de manera aleatoria. Deberá destaparlas en el tablero y enviar la actualización del tablero al cliente.
- El servidor deberá validar si algún jugador logra destapar dos casillas con la misma palabra. Si es el caso, el servidor contabiliza el punto para el jugador correspondiente.
- Al finalizar la partida, (todas las casillas son destapadas), el servidor deberá informar al jugador si ganó, perdió o empató la partida, así como mostrar el tiempo de duración de la partida.