

Práctica 3: Othello

UNIDAD DE APRENDIZAJE : Aplicaciones para comunicaciones en red UNIDADES TEMÁTICAS I y II: Sockets de flujo, Sockets de datagrama	
No. Y Título de la práctica: Práctica no. 3 Othello	Tiempo de realización: 4.5 horas
Objetivo de la práctica: El estudiante migrará una aplicación de escritorio a un modelo cliente-servidor utilizando sockets no bloqueantes.	
Situación problemática: En el mundo laboral, existen muchas situaciones en las que se requiere migrar sistemas pensados para ejecutarse inicialmente en un entorno de escritorio (standalone) a un entorno en red. Esta no es una tarea sencilla, ya que independientemente de las buenas costumbres de programación/documentación. Se necesita tener una idea clara del diseño de sistemas para decidir donde se implementará la lógica (cliente, servidor, o ambos) para tener una aplicación eficiente y funcional. ¿Qué tipo de socket no bloqueante se necesitará para implementar una aplicación cliente servidor eficiente?	
Competencia específica: Desarrolla aplicaciones en red, con base en el modelo cliente-servidor y utilizando únicamente la interfaz de sockets no bloqueantes, para el envío de datos.	
Competencias genéricas: <ul style="list-style-type: none">• Aplica los conocimientos en la práctica• Demuestra habilidad para trabajar en equipo• Demuestra capacidad de investigación• Desarrolla aplicaciones en red con base en la tecnología más adecuada	Elementos de competencia: <ul style="list-style-type: none">• Programa aplicaciones en red con base en el modelo Cliente-Servidor y la interfaz de aplicaciones de sockets de flujo• Analiza los servicios definidos en la capa de transporte• Emplea el modelo Cliente-Servidor para construir aplicaciones en red• Programa aplicaciones Cliente-Servidor utilizando sockets de flujo bloqueantes

Introducción

El juego de Othello o Yang es un juego para dos jugadores, que comparten 64 fichas iguales, de caras distintas, que se van colocando por turnos en un tablero dividido en 64 casillas. Las caras de las fichas se distinguen por su color y cada jugador tiene asignado uno de esos colores, ganando quien tenga más fichas sobre el tablero al finalizar la partida. Sus orígenes se sitúan en Inglaterra, en el siglo XIX, en donde los londinenses Lewis Waterman y John W. Mollett comercializaron en 1880, cada uno, juegos con reglas parecidas. En 1971 el japonés Goro Hasegawa cambió dos reglas del juego y lo registró bajo el nombre de Othello®, inspirado por la obra de Shakespeare del mismo nombre. ("Oteló" en español).

Actualmente únicamente se juega con las reglas de Othello (en idioma castellano también se utiliza el nombre de otelo, y en Argentina bajo el nombre comercial de "Yang").

Recursos y/o materiales

- | | |
|---|--|
| <ul style="list-style-type: none">Manual de prácticas de laboratorio de Aplicaciones para Comunicaciones en RedPlumonesBibliografía | <ul style="list-style-type: none">InternetComputadoraIDE de desarrolloApuntes |
|---|--|

Instrucciones

En esta práctica debes migrar la aplicación “juego de Othello” a un modelo cliente-servidor, en el cual el servidor se comportará como un servidor de juegos, llevando el control de hasta 2 partidas de dos jugadores c/u de manera concurrente. Deberá existir al menos otro servidor de juegos, de modo que cuando el jugador número 5 intente conectarse al servidor de juegos, éste le devolverá un mensaje indicando la dirección IP y puerto del siguiente servidor al que deberá conectarse, dado que el no puede iniciar una nueva partida.

Para poder iniciar una partida, deberán existir al menos dos jugadores conectados al servidor.

Desarrollo de la práctica

A partir de los programas contenidos en el archivo Oteló.zip que te serán proporcionados por el profesor deberás realizar lo que a continuación se te pide:

- A partir de los programas contenidos en el archivo Oteló.zip crea un diagrama de clases para documentar la funcionalidad de cada una de las clases que componen a la aplicación Oteló.
- Separa la lógica de la aplicación, de forma que la validación de los movimientos del usuario se realice del lado del servidor y la vista del lado del cliente. Ilústralo a partir de un diagrama de arquitectura física.
- Implementa tanto el lado del servidor, así como el lado del cliente.
- Implementa un módulo que se encargue de validar la cantidad de partidas registradas en el servidor, de modo que cuando se supere el límite de éstas (solo 2 partidas por servidor con 2 jugadores por partida), el servidor no cree más partidas y a los clientes que intenten conectarse a él les enviará un mensaje indicando la dirección IP y número de puerto de otro servidor de juegos disponible.

Cierre de la práctica

Preguntas:

- ¿Qué tipo de socket no bloqueante utilizaste y por qué?
- ¿Qué comportamiento mostró la aplicación utilizando este tipo de socket (retardo en envío, paquetes perdidos, etc.)?
- ¿Qué diferencias experimentaste con respecto al uso de sockets bloqueantes?