

Proyecto #2 – Valor 20%. I Ciclo 2019

Objetivos

- Describir el uso de diversas API para creación y gestión de hilos
- Implementar técnicas de programación asociadas al uso de APIs del sistema
- Instalar y configurar distintas tecnologías y protocolos de red
- Desarrollar e implementar librerías dinámicas (DLLs)
- Instalar y configurar un Domain Controller (DC) con Windows Server 2016
- Implementar una solución de autenticación de usuarios (Active Directory)
- Implementar técnicas de comunicación y sincronización entre procesos

Descripción

El proyecto consiste en desarrollar una versión simplificada del juego de cartas llamado “Texas Hold’em” en su versión sin límites. La aplicación debe permitir la interacción de los jugadores de manera distribuida, eso quiere decir que cada jugador se conecta a un módulo centralizado a través de la red – se implementa el modelo cliente/servidor.

La Baraja (Mazo)

En el juego se utiliza una baraja inglesa de 52 cartas. Las 52 cartas se dividen en 4 palos (2 de color rojo y 2 de color negro):



Espadas (también llamado picas o bastos)



Corazones



Diamantes



Tréboles

Cada palo está formado por 13 cartas, que ordenadas de menor a mayor forman la siguiente secuencia:

2 3 4 5 6 7 8 9 10 J Q K A

¿En qué consiste el juego?

En este juego participan un número de jugadores mientras que el “*dealer*” (conocido como la casa) no participa activamente en el juego. El objetivo del juego consiste en acumular la mayor cantidad de fichas posible de manera que todos los demás jugadores queden sin fichas y se retiren. Gana el jugador que quede con todas las fichas al final del juego.

A cada jugador se le reparte una cierta cantidad de fichas para jugar – la misma cantidad a todos por igual. Las fichas tienen diferentes denominaciones, pero para efectos de este juego puede asumir que todas las fichas tienen la misma denominación, al final lo relevante es la cantidad de la apuesta.

El juego se desarrollará en rondas. En cada ronda, se designa un jugador "dealer" a partir del cual se reparten las cartas (hacia la izquierda) y se designan dos jugadores para las apuestas ciegas (los jugadores a la izquierda del dealer): una apuesta pequeña y otra apuesta grande. El monto de las apuestas ciegas se determina en cada ronda, usualmente se van aumentando conforme pasa el tiempo de juego o por un número determinado de rondas. La designación del dealer cambia en cada ronda de manera que las apuestas ciegas vayan rotando. Luego de las apuestas ciegas, se reparten dos cartas a cada jugador. Cada jugador esperará su turno para hacer la jugada, pero se comienza con el siguiente jugador que se encuentre a la izquierda del que realizó la apuesta grande. En este punto, cada jugador determinará de acuerdo con las cartas que tenga si realiza alguna de las siguientes acciones:

- Pagar la apuesta y jugar (call). El jugador iguala la apuesta mayor.
- No pagar la apuesta y retirarse (fold). El jugador no juega ese turno.
- Subir la apuesta y jugar (raise). El jugador aumenta la apuesta mayor.
- Pasar (check). En caso de que el jugador ya haya apostado la apuesta mayor.

Se tomarán turnos para que cada jugador determiné la acción a realizar hasta que todos hayan participado. En este punto, se procede a mostrar las primeras tres cartas comunes (el flop). Nuevamente se procede por turnos preguntando a cada jugador que acción desea realizar. El proceso continúa hasta que todos los jugadores hayan igualado la apuesta o hasta que solamente un jugador haya apostado si, por ejemplo, los demás jugadores se retiraron, en cuyo caso el jugador restante es el ganador del "pozo" y recoge las apuestas de todos. Si ocurre lo segundo, se recogen las cartas de los jugadores y procede un nuevo turno. En caso de que todos los jugadores hayan aceptado, entonces el dealer coloca una nueva carta en la mesa (el turn). Se realiza el mismo proceso anterior.

Finalmente, si la jugada continúa se coloca la última carta común (el river) y los jugadores proceden con las apuestas finales. De los jugadores que queden, se selecciona el ganador tomando en cuenta el mejor juego de 5 cartas a partir de todas las cartas (las cartas comunes más las dos cartas del jugador). La siguiente tabla muestra las posibles jugadas y el orden en importancia:

	Jugada	Jugada en inglés	Descripción	Ejemplo
1	Escalera real o Flor Imperial	Royal flush	Cinco cartas seguidas del mismo palo del 10 al As.	
2	Escalera de color o flor corrida	Straight flush	Cinco cartas seguidas del mismo palo, pero que no tiene As como carta alta.	
3	Póquer	Four of a kind	Cuatro cartas iguales en su valor.	
4	Full	Full house	Tres cartas iguales (trío), más otras dos iguales (par).	
5	Color	Flush	Cinco cartas del mismo palo sin ser seguidas.	
6	Escalera	Straight	Cinco cartas seguidas de palos diferentes.	
7	Trio, Tercio, Pierna o Trucha	Three of a kind	Tres cartas iguales en su valor.	
8	Doble par	Two pair	Dos pares de cartas.	
9	Par	Pair	Dos cartas iguales y tres diferentes.	
10	Carta alta o "Carta Mayor"	High card	Gana quien tiene la carta más alta.	

Puede consultar los siguientes enlaces para obtener información acerca de las reglas del juego y la metodología:

<https://www.pokerstars.com/espanol/poker/games/texas-holdem/>

<https://www.poquer.com.es/reglas.html>

<https://www.pokerlistings.es/texas-holdem-poker-reglas>

https://es.wikipedia.org/wiki/Texas_hold_%27em

Especificaciones

Para el desarrollo del proyecto, se deben implementar los siguientes módulos:

1. **Módulo de Autenticación:** Este módulo permitirá validar la información del usuario que va a ingresar al juego. Únicamente los usuarios autorizados podrán ingresar a la sala de juego. Si un usuario no tiene una cuenta creada, entonces se le debe dar la opción de crear una. El programa debe gestionar la autenticación y creación de usuario a través de un *controlador de dominio Active Directory*. El controlador de dominio debe instalarse y configurarse en un Windows Server 2016 o 2019.

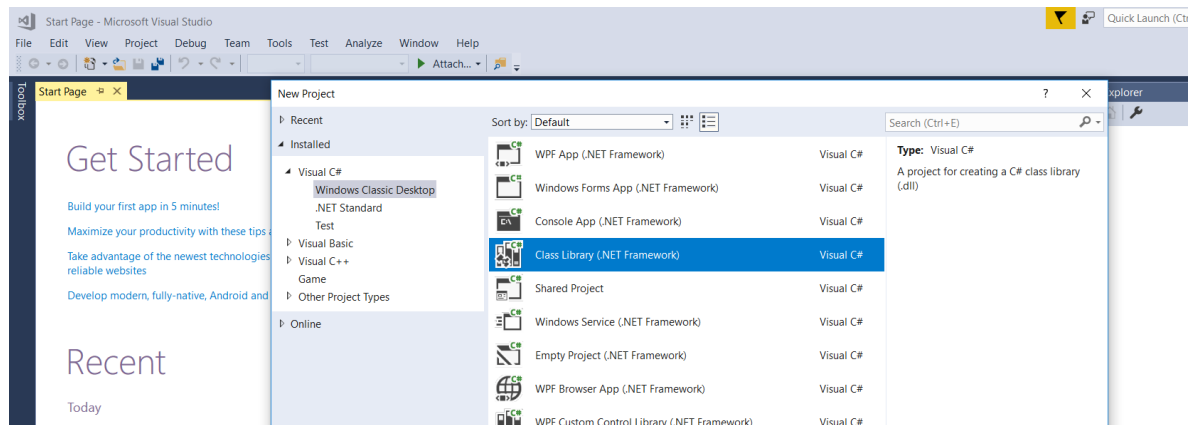
El módulo de autenticación debe encapsular todas las funciones que ejecuten la comunicación desde el juego hacia el Active Directory para validar las credenciales de usuario, así como la creación de usuarios nuevos.

Las funciones se almacenarán en una librería dinámica (.dll). Por lo tanto, la aplicación deberá importar esa librería .dll para tener acceso a las funciones de autenticación.

```
Función 1{  
}  
  
Funcion 2{  
}
```

autenticacion.dll

Para crear la librería dinámica, deberá seleccionar como tipo de proyecto “*Class Library*” (Biblioteca de Clase) desde el menú de opciones de C#.



Cuando compile el proyecto, se generará como resultado un archivo de tipo .dll

2. **Módulo Controlador (server):** Esta es la interfaz principal del programa, que permite la interacción entre los jugadores de Texax Hold'em y la casa (crupier). El módulo controlador deberá realizar las siguientes acciones:

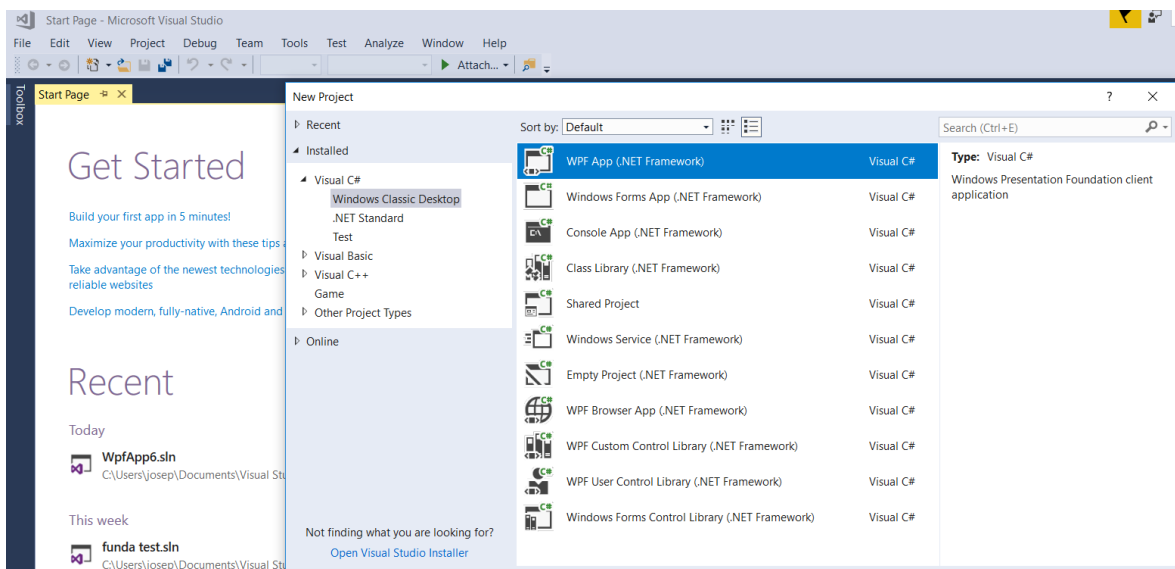
- Escuchar en un socket por conexiones externas por parte de los jugadores a través de una red TCP/IP.
- Crear y gestionar los hilos para cada uno de los jugadores que ingresen a la sala. Se deberá crear un hilo por cada jugador que ingrese a la sala de juego.
- Controlar las acciones del juego (repartir cartas, asignar el dealer y apuestas ciegas, mantener el registro de apuestas, administrar los turnos de cada jugador y las acciones que realicen así como la verificación del ganador en cada ronda.)
- Implementar los mecanismos de comunicación con los programas cliente, de manera que se reciban las acciones de los jugadores. La comunicación entre el módulo controlador y los programas cliente debe implementarse mediante objetos serializados con JSON.
- La cantidad máxima de jugadores para una mesa de juego es de 4. En caso de que más de 4 jugadores intenten ingresar a la mesa del juego, se impedirá la conexión de jugadores extra.
- Verificar la identidad de los usuarios y crear usuarios nuevos. Para ello el módulo controlador deberá importar la librería .dll (módulo de autenticación) con las funciones que hacen las llamadas al Controlador de Dominio.

El módulo controlador no requiere de una interfaz de usuario, pero queda a discreción de los programadores imprimir mensajes en la consola para ver los eventos del juego. Se recomienda utilizar una aplicación de tipo Consola desde los proyectos de C# en Visual Studio.

3. **Módulo Cliente:** El módulo cliente implementará la interfaz de usuario para cada uno de los jugadores. Para cada jugador se debe visualizar la mesa de juego, junto con las cartas del jugador y las cartas comunes, el "pozo", la cantidad total de fichas (o el monto) del jugador y la apuesta realizada, y muestra el nombre de cada jugador en la posición que corresponda en la mesa. Este módulo deberá realizar las siguientes acciones:

- Mostrar una pantalla de inicio de juego, donde se debe indicar el nombre/dirección IP y el número de puerto del servidor del juego al cuál conectarse. El usuario escribirá los datos y se completará la conexión con el servidor (módulo controlador).
- Una vez establecida la comunicación con el servidor, la ventana deberá mostrar un formulario de autenticación al usuario, para digitar el nombre de usuario y contraseña, o para registrar un usuario nuevo.
- Luego de realizar la autenticación, se deberá mostrar en la ventana la mesa del juego se hay espacio en la misma.
- El jugador podrá observar las jugadas de los demás jugadores (call, raise, fold) así como la apuesta hecha por cada jugador. Además, este módulo es el que le permite al usuario interactuar en el juego.
- En todo momento deberá existir una opción dentro de la ventana para salir del juego, si el usuario la selecciona abandonará la mesa. También abandonará la mesa automáticamente el jugador que no tenga dinero disponible para apostar o si la cantidad de fichas es menor a la apuesta grande.

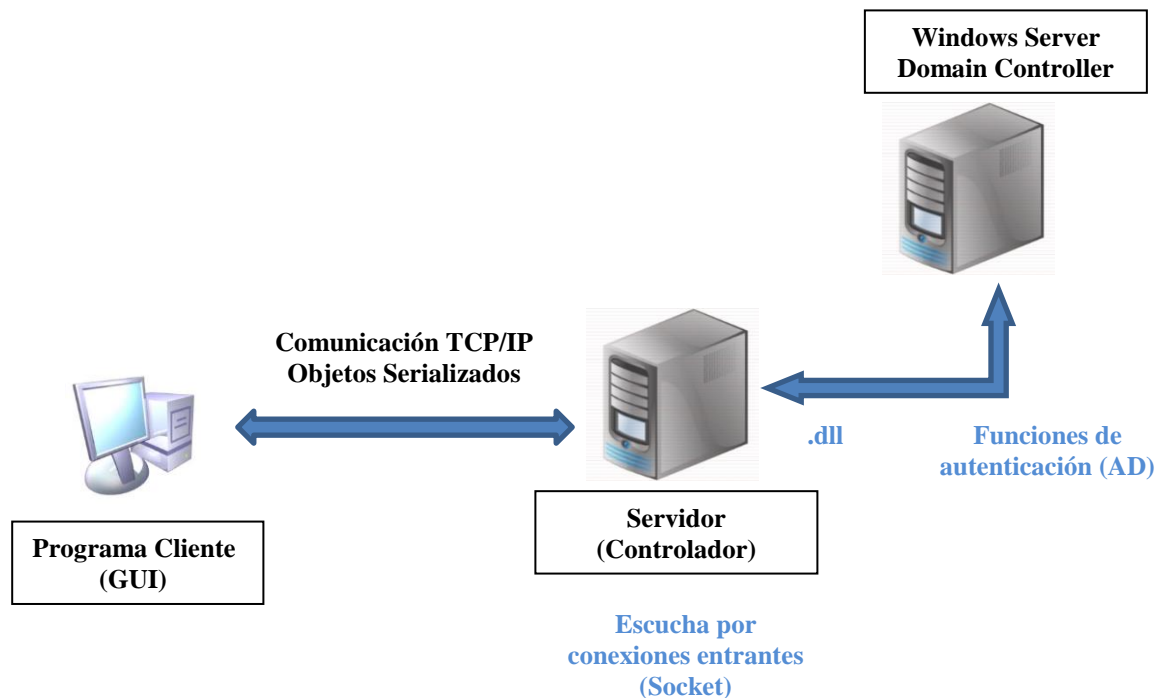
El módulo cliente deberá implementarse a través de WPF (Windows Presentation Foundation), que es una tecnología de Microsoft utilizada para el desarrollo de interfaces de usuario. Para crear el proyecto, seleccione la opción “*WPF App (.Net Framework)*” desde las opciones de proyecto de C#.



Flujo del Programa

La aplicación completa contará con dos archivos ejecutables y una librería dinámica (.dll):

- Programa Cliente – ejecutable que contiene el módulo cliente (interfaz de usuario)
- Programa Servidor – ejecutable que contiene el módulo servidor
- Shared Library - .dll que contiene las funciones de autenticación (se importan desde el servidor)



Consideraciones Adicionales

- En el módulo controlador, debe manejar las condiciones de carrera al utilizar datos compartidos entre los hilos (utilice la clase *Mutex* de *System.Threading* de .Net). Además debe evitar deadlocks.
- Debe manejar adecuadamente las situaciones de error de conexión de la red (si un cliente deja de responder o hubo un problema en la red)
- En el caso del manejo de las apuestas, cada vez que un usuario se conecte, puede asignar un monto fijo de fichas con el cuál puede realizar apuestas. Los jugadores inician el juego con la misma cantidad de fichas.

Observaciones

- Para la implementación de Active Directory, se recomienda utilizar Windows Server 2016 o Windows Server 2019. Puede implementarlo mediante una máquina virtual. Investigue acerca de la instalación de un servidor de dominio.
- Utilice el lenguaje de programación C# en Visual Studio 2019 (recomendado) o Visual Studio 2017 para el desarrollo del proyecto. Para el desarrollo de la interfaz gráfica, debe utilizar WPF (Windows Presentation Foundation).
- La tarea debe ser realizada en grupos de cuatro personas como máximo.
- Se debe entregar el proyecto con el código fuente, ejecutables y librerías necesarias para su ejecución. Así mismo, debe crear un instalador del programa para que pueda ejecutarse desde cualquier máquina (sin necesidad de instalar el respectivo framework / compilador). El instalador debe incluir todos los archivos ejecutables y la librería dinámica (cliente, servidor, .dll).

- Tome en cuenta que debe desarrollar tres proyectos distintos en Visual Studio (puede crear una solución, y que la solución tenga los 3 proyectos).
- La fecha límite para la entrega es el día viernes 5 de junio en horas de clase.
- En caso de copia (dos proyectos o más con mucha similitud que lo demuestren) o plagio (códigos descargados de Internet, libros o cualquier otro material), la nota de la tarea es 0.
- Debe indicar todos los recursos consultados para la elaboración del proyecto.

Recursos útiles

Threads en .NET

<https://docs.microsoft.com/en-us/dotnet/api/system.threading.thread?view=netframework-4.8>

<https://docs.microsoft.com/en-us/dotnet/standard/threading/using-threads-and-threading>

<https://www.c-sharpcorner.com/article/understanding-threading-in-net-framework/>

Mutex en .NET

<https://docs.microsoft.com/en-us/dotnet/api/system.threading.mutex?view=netframework-4.8>

WPF

<https://docs.microsoft.com/en-us/dotnet/framework/wpf/getting-started/walkthrough-my-first-wpf-desktop-application>

<https://wpf-tutorial.com/>

<https://programarfacil.com/blog/programacion-net-blog/que-es-wpf/>

<https://www.c-sharpcorner.com/technologies/wpf>

C#

<https://csharp.net-tutorials.com/>

<https://www.tutorialspoint.com/csharp/>

<https://www.guru99.com/c-sharp-tutorial.html>

C# y Sockets

<https://docs.microsoft.com/en-us/dotnet/api/system.net.sockets.socket?view=netframework-4.8>

<http://csharp.net-informations.com/communications/csharp-multi-threaded-socket.htm>

<https://www.c-sharpcorner.com/article/socket-programming-in-C-Sharp/>

<http://codigoprogramacion.com/cursos/c-sharp/sockets-en-c-parte-1.html#.XMCXNKRMFPY>

C# y JSON

<https://docs.microsoft.com/en-us/dotnet/framework/wcf/feature-details/how-to-serialize-and-deserialize-json-data>

<https://www.c-sharpcorner.com/article/json-serialization-and-deserialization-in-c-sharp/>

C#, LDAP y Active Directory

<https://docs.microsoft.com/en-us/dotnet/api/system.directoryservices?view=netframework-4.8>

<https://docs.microsoft.com/en-us/dotnet/api/system.directoryservices.directoryentry?view=netframework-4.8>

<https://www.c-sharpcorner.com/article/authenticate-a-user-against-the-active-directory/>

<https://auth0.com/blog/using-ldap-with-c-sharp/>