



## PONTIFICIA UNIVERSIDAD JAVERIANA

### FACULTAD DE INGENIERÍA

*Departamento de Ingeniería de Sistemas*

Profesor: Camilo Cañón Correa

*Octubre de 2016*

### **MyBook**

Gracias a la gran expansión de herramientas sociales una importante firma del mundo del entretenimiento está pensando en crear una aplicación similar. Esta empresa se enteró del excelente trabajo de los estudiantes Pensamiento Algorítmico de la PUJ y está haciendo una licitación para la construcción del software que tenga ciertas características similares a otras en Internet. Por lo anterior, usted ha sido seleccionado para participar en esta licitación y deberá construir un programa en C++ que cumpla con los siguientes requerimientos:

Debe tener un menú con las siguientes opciones:

1. Inscribirse: Permite al usuario crear una cuenta en esta aplicación. La información básica que será utilizada para cada usuario es: Nombre, apellido, correo electrónico y contraseña. No se puede crear una cuenta en MyBook con un correo electrónico repetido. De cada dato del usuario, excepto el nombre y apellido, se podrá elegir si es público o no. Si la información es pública, aparecerá en las búsquedas de hagan otros usuarios del directorio de MyBook.
2. Ingresar a la cuenta (Sign-in): con esta opción, el programa deberá pedir el correo electrónico y la contraseña, para ingresar. Una vez se ha ingresado, deberá mostrar el siguiente menú:
  - 2.1. Editar perfil:
    - 2.1.1. Cambiar información básica (Nombre y apellido).
    - 2.1.2. Cambiar/ingresar fecha de nacimiento.
    - 2.1.3. Administración de seguridad: Se debe permitir indicar si los datos básicos del usuario así como el año de nacimiento serán datos públicos o no.
    - 2.1.4. Deshabilitar perfil: Esta función hará que este contacto “se oculte” cuando sus amigos lo listen. Ocultarse corresponde a que se verá que hay un amigo, pero no se verán sus datos, solamente como nombre se verá “Usuario MyBook”.
  - 2.2. Amigos
    - 2.2.1. Ver amigos: Permitirá ver la lista de amigos con sus detalles públicos.
    - 2.2.2. Buscar amigos, pedirá el nombre o apellido de un usuario y mostrará el listado de los usuarios que tienen ese nombre o ese apellido, mostrando solamente su información pública.
    - 2.2.3. Aceptar o rechazar amigos: Esta opción permitirá ver las solicitudes de amistad que se han hecho de otras cuentas de MyBook, dando la opción de aceptarlas o rechazarlas. Si una solicitud es aceptada, ese contacto deberá aparecer dentro de la lista de amigos. Cuando se acepta a alguien como amigo, se deberá pedir el grupo a la cual pertenecerá dicho amigo (siempre se debe pedir el grupo y éste debe existir dentro de los grupos creados). Si la solicitud es rechazada, este contacto no aparecerá en la lista de amigos. En cualquiera de los casos, aceptación o rechazo, la

solicitud ya no deberá aparecer en la lista de solicitudes. Cada solicitud deberá tener la siguiente información: Nombre y apellido del contacto que hace la solicitud y una lista con los nombres y apellidos de los amigos mutuos.

2.2.4. Eliminar amigos: Esta opción deberá pedir el correo electrónico del amigo que se desea eliminar de la lista de amigos y deberá eliminarlo de dicha lista.

2.2.5. Mover un amigo a una lista: Deberá preguntar el nombre del amigo y a lista a la que se va a cambiar dicho amigo, éste aparecerá entonces en la nueva lista.

2.2.6. Administrar Listas de amigos: Mediante esta opción se podrá:

2.2.6.1. Crear una lista: Se deberá preguntar el nombre de la lista. Siempre, existirá la lista llamada "AMIGOS". No se debe permitir la creación de una lista repetida.

2.2.6.2. Eliminar una lista: Se pedirá el nombre de la lista a eliminar. La única lista que no se puede eliminar es la lista "AMIGOS". Cuando se elimina una lista y a esta lista pertenecen un grupo de contactos, esos contactos deberán quedar en la lista "AMIGOS"

2.2.6.3. Ver amigos de una lista: Esta opción debe mostrar las listas de amigos que existen, y deberá pedir la lista que se quiere ver. Una vez ingresado el nombre de la lista, deberá mostrar todos los contactos que pertenecen a esta lista. Si la lista elegida es AMIGOS, se deberán mostrar TODOS los amigos sin importar las listas a las que pertenezcan.

2.2.7. Hacer solicitudes de amistad: Esta opción deberá permitir ver una lista de todos los usuarios de MyBook, y deberá permitir pedir el correo electrónico del contacto al cual se le enviará la solicitud de amistad. Cada solicitud deberá tener la siguiente información: Nombre y apellido del contacto que hace la solicitud y una lista con los nombres y apellidos de los amigos mutuos. **La información de la solicitud se debe crear automáticamente.**

## 2.3. Juegos

### 2.3.1. Triqui:

2.3.1.1. Jugar: pedirá el correo y la contraseña del amigo contra el cual va a jugar y les permitirá jugar triqui entre los dos. Se almacenará en la opción de juegos, en triqui, las estadísticas de los partidos jugados, perdidos, ganados y empatados.

2.3.1.2. Ver estadísticas: mostrará las estadísticas del triqui.

## 2.4. Salir de la cuenta (Sign-out o cerrar sesión).

## 3. Salir: Sale de MyBook.

El programa deberá permitir tener un MAXIMO de 50 cuentas. Cada cuenta podrá tener hasta MAXIMO 60 amigos. Así mismo para cada cuenta se podrá manejar un máximo de 20 solicitudes de amistad. MAXIMO 10 listas de amigos por cada cuenta.

**Entregas:**

5% Primera entrega. Documento de diseño. Se deberá entregar un documento que contiene el diseño del programa (Modelo conceptual de problema e implementación del menú) y el menú en C++. Fecha máxima 31 de octubre 23:59:

10% Entrega final y sustentación. Esta entrega tiene dos componentes

- a. 20% Ejecución del programa: El programa deberá ejecutarse correctamente, de acuerdo a lo especificado en este documento.
- b. 80% Sustentación: El programa deberá ser sustentado individualmente.

La entrega final se realizará durante la semana segunda semana después del tercer parcial.



### **Snake III**

**Snake** es el nombre común para el concepto de videojuegos en donde el jugador maneja una línea la cual crece en longitud, siendo ella misma el obstáculo primario. El concepto se originó en 1976. Después de que una variación fue precargada en los teléfonos móviles NOKIA en 1998, hubo un resurgimiento en el interés del concepto Snake pues encontró una mayor audiencia.

Snake vive en un mundo rectangular, de máximo 60 columnas x 20 filas. Snake tiene una longitud inicial de 2 cuadros, y en la medida en que va consumiendo ciertas frutas su longitud va aumentando 2 cuadros por cada fruta. Cada vez que el usuario presiona una de las flechas, Snake irá en esa dirección, siempre y cuando la rotación sea de 90°.

Una fruta aparece aleatoriamente en una posición libre del tablero, y existen dos tipos de frutas, las que hacen crecer a Snake y las que hacen disminuir su tamaño. Cada fruta se queda presente en el tablero máximo 40 movimientos de Snake, y si no ha sido consumida, desaparecerá.

Una vez que Snake ha comido una de las frutas, inmediatamente aparecerá otra en otro lugar disponible del tablero. Cada una de las frutas da un puntaje de 20 puntos si hace crecer a Snake (la hace crecer 2 cuadros) y de -10 puntos si la hace disminuir (la hace disminuir 2 cuadros).

El propósito del juego consiste en ganar la mayor cantidad de puntos sin que Snake golpee alguno de los muros o se muerda ella misma.

Usted ha sido seleccionado para elaborar un programa en C++ que permita jugar Snake con las siguientes opciones:

1. Registrar usuario: esta opción permitirá crear una cuenta con la siguiente información: correo electrónico (es único por usuario), contraseña, nombres, apellidos, carrera y semestre
2. Administrar juego: permitirá crear nuevos mundos con diferentes configuraciones de muros y administrar usuarios.
  - 2.1. Ver información de usuarios: mostrará el listado de los usuarios con su información básica
  - 2.2. Eliminar usuario: Se elimina la información completa de un usuario, seleccionando el correo electrónico del jugador
  - 2.3. Crear mundo: pedirá información de nombre del mundo y ubicación de los muros. Cada muro ocupará una casilla del tablero. Se deberán ubicar tantos muros cuantos quiera el usuario. También se permitirá eliminar muros que se hayan creado de mundo actual. Cada mundo tendrá un código único asignado automáticamente. Máximo se pueden crear hasta 10 mundos.
  - 2.4. Eliminar mundo: se eliminará uno de los mundos creados, ingresando el código del mundo que se va a eliminar.

- 2.5. Modificar mundo: permitirá modificar un mundo ya creado, eliminando o creando muros de acuerdo al código del mundo que se seleccione para ser modificado.
- 3. Jugar
  - 3.1. Ingresar al sistema: Permitirá ingresar con el correo electrónico y la contraseña para poder jugar
  - 3.2. Ver estadísticas:
    - 3.2.1. Ver estadísticas: mostrará las estadísticas del juego
    - 3.2.2. Ordenar: permitirá ordenar la información de los jugadores, por alguno de los siguientes criterios:
      - 3.2.2.1. Por puntaje
      - 3.2.2.2. Por correo
      - 3.2.2.3. Por apellido
  - 3.3. Modificar información de jugador:
    - 3.3.1. Modificar contraseña
    - 3.3.2. Modificar nombres
    - 3.3.3. Modificar apellidos
    - 3.3.4. Modificar carrera
    - 3.3.5. Modificar semestre
  - 3.4. Elegir configuración y dificultad
    - 3.4.1. Configuración: listará el código de los mundos creados para elegir uno de ellos
    - 3.4.2. Dificultad: la dificultad de 1 fácil a 9 difícil. La dificultad se refiere a la velocidad. Cada nivel mayor de dificultad le adiciona 1 punto más a las frutas.
  - 3.5. Jugar: permitirá al usuario jugar según la configuración seleccionada. Si no se ha seleccionado ninguna configuración específica, el juego iniciará con el tablero rectangular, con muros en los bordes del mundo y sin muros interiores. Cuando el usuario pierda, el puntaje que lleva en esta ocasión será guardado dentro de las estadísticas para ese jugador.

### **Entregas:**

5% Primera entrega. Documento de diseño. Se deberá entregar un documento que contiene el diseño del programa (Modelo conceptual de problema e implementación del menú) y el menú en C++. Fecha máxima 31 de octubre 23:59:

10% Entrega final y sustentación. Esta entrega tiene dos componentes

- a. 20% Ejecución del programa: El programa deberá ejecutarse correctamente, de acuerdo a lo especificado en este documento.
- b. 80% Sustentación: El programa deberá ser sustentado individualmente.

La entrega final se realizará durante la semana segunda semana después del tercer parcial.