

## Proyecto 2

---

### Objetivos

#### Objetivo General

- Que el desarrollador pueda aplicar los conocimientos adquiridos en el curso y que éste sea capaz de construir un sistema complejo mediante el lenguaje ensamblador haciendo uso de algoritmos creativos para solucionar los distintos requerimientos solicitados

#### Objetivos Específicos

- Implementar soluciones creativas para algoritmos complejos
- Manipular correctamente la memoria del sistema
- Mezclar diferentes familias de funciones en interrupción.
- Aplicar instrucciones aritméticas a la solución de algoritmos
- Aplicar instrucciones lógicas a la solución de algoritmos
- Aprender a utilizar como entrada teclas auxiliares.
- Comprender y aplicar el manejo de memoria de video en ensamblador

### Descripción

Se desea desarrollar una variante del clásico juego galaga, el cuál contará con distintos niveles, enemigos y desplazamientos posibles.

En el mismo proyecto se evaluarán la implementación de funciones para manipular la información del juego y / o generar salidas de datos con criterios específicos

#### Teclas de función

Dentro del desarrollo de los menús se solicitará que haga uso de las teclas de función del teclado, por ejemplo, F7 se referiría a la tecla de función con dicho nombre, lo mismo a ESC, HOME u otra que se solicite.

---

# Desarrollo del proyecto

---

## Identificación del desarrollador

Cada vez que se ejecute la práctica se deberá mostrar la identificación del desarrollador, ejemplo:

```
Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas
Arquitectura de Compiladores y ensambladores 1
Sección <A|B>
< Nombre del desarrollador >
< Registro Académico >
```

Luego de la identificación deberá esperar a que se presione Enter para poder continuar hacia el menú.

---

## Menú Principal

El menú del desarrollo debe incluir las siguientes opciones

```
Menu
=====

F1. Login
F5. Register
F9. Exit
```

## Login

Permitirá ingresar al menú de juego por medio de las credenciales del usuario.

Se solicitarán las credenciales del usuario, Username y Password; el password no debe ser visible y debe reemplazarse por asteriscos.

```
Login
=====

Username: pruebaUsuario
Password: *****
```

El usuario y el password es Case Sensitive, ~~el ingresar 3 veces erróneas el password bloqueará el usuario.~~

Si el usuario se encuentra bloqueado es necesario mostrar un mensaje de advertencia y negar el acceso aunque la contraseña sea correcta. **Únicamente el administrador puede desbloquear un usuario**

Login

---

Username: pruebaUsuario

Password: \*\*\*\*\*

```
>>          Permission denied          <<
>> There where 3 failed login attempts <<
>>   Please contact the administrator  <<

>>   Press Enter   to go back to menu   <<
```

En caso de tratarse del administrador, este no se bloqueará totalmente, se establece una política de cooldown de 30 segundos para el siguiente intento. Esto aplica únicamente luego del tercero erróneo, cada nuevo intento erróneo lo mostrará y volverá a durar 30 segundos más; se desactivará luego del siguiente intento correcto.

Login

---

Username: AdminB19387

Password: \*\*\*\*\*

```
>>          Permission denied          <<
>> There where 3 failed login attempts <<
>> Wait 18 more seconds and try again <<

>>   Press Enter   to go back to menu   <<
```

~~Es obligatorio mostrar los segundos pendientes en esta etapa; al momento del bloqueo mostrará este mensaje inmediatamente indicando los 30 segundos.~~

## Register

Permitirá registrarse en el sistema para poder tener acceso al juego

- ~~Se solicitará el nombre de usuario y se harán las siguientes validaciones:~~
  - No puede empezar por número
  - La longitud debe validarse entre 8 y 15 caracteres
  - El nombre de usuario no debe existir

4. Los únicos caracteres especiales que puede contener serán:

- Guión (-)
- Guión bajo (\_)
- Punto (.)

2. ~~Se solicitará el password para el usuario y se harán las siguientes validaciones~~

1. Debe tener al menos una mayúscula
2. Debe tener al menos un número
3. Debe tener al menos un caracter especial de los siguientes:
  - Admiración (!)
  - Mayor qué (>)
  - Porcentaje (%)
  - Punto y coma (;)
  - Asterisco (\*)
4. Longitud mínima de 16 caracteres
5. Longitud máxima queda a discreción del desarrollador

~~Todas las validaciones que no se cumplan debe notificarse explícitamente.~~

Register

---

Username: 5Juan

Password: \*\*\*\*\*

```
>> Action Rejected <<
>> <<
>> Missed requirements: <<
>> Username begins with a letter <<
>> Username length between 8 and 15 characters <<
>> Password must contain at least one number <<
>> Password length at least 16 characters <<

>> Press Enter to go back to menu <<
```

Exit

Terminará la ejecución del programa

## Menú usuario

El menú de usuario debe incluir las siguientes opciones

```
User menu                user: Juan2000
```

---

```
F2. Play game
F3. Show top 10 scoreboard
F5. Show my top 10 scoreboard
F9. Logout
```

### Play Game

Dirigirá al usuario hacia el juego directamente (explicado más adelante)

### Show top 10 scoreboard

Mostrará en pantalla el scoreboard top 10 ordenado por Puntos de forma descendente, esperará que se presione ENTER para volver al menú anterior.

```
Top10 Scoreboard        user: Juan2000
```

---

Rank	Player	N	Points	Time
------	--------	---	--------	------

---

1	Juan2000	3	280	300
2	Pedro9876	3	240	290
3	Luis11111	3	230	280
4	Walter8788	3	200	280
5	Juan2000	3	199	250
6	Juan2000	3	195	225
7	Fernando09	2	170	210
8	Jorge5555	2	150	203
9	Javier000	2	130	201
10	Alejandro	1	80	150

```
>>  Press Enter to go back to menu  <<
```

### Show my top 10 scoreboard

Mostrará en pantalla el scoreboard top 10, del usuario en cuestión, ordenado por Puntos de forma descendente; debe mostrar el rank en el que se encuentra el punteo en relación al rank global; esperará que se presione enter para volver al menú anterior.

My Top10 Scoreboard      user: Juan2000

Rank	Player	Points
------	--------	--------

Rank	Player	N	Points	Time
------	--------	---	--------	------

1	Juan2000	3	280	300
5	Juan2000	3	199	250
6	Juan2000	3	195	225
15	Juan2000	1	75	130

>>    Press Enter to go back to menu    <<

## Logout

Cierra sesión y vuelve al menú principal

## Menú Administración

El menú de administración debe incluir las siguientes opciones

```
Admin menu                user: AdminB19387
```

---

```
F1. Unlock User
F2. Promote user to admin
F3. Demote user from admin
F5. Bubble sort
F6. Heap sort
F7. Tim sort
F9. Logout
```

- Funciones en F2 y F3 solo disponibles para el usuario administrador general
- Para un usuario administrador normal mostrará en F2 Show top 10 scoreboard, F3 Show my top 10 scoreboard y F4 Play Game

```
Admin menu                user: Juan2000
```

---

```
F1. Unlock User
F2. Show top 10 scoreboard
F3. Show my top 10 scoreboard
F4. Play Game
F5. Bubble sort
F6. Sort Method 2
F7. Sort Method 3
F9. Logout
```

## Unlock User

Quita el bloqueo de un usuario

```
Unlock User                user: AdminB19387
```

---

```
Username to unlock: Juan2000
```

Se debe notificar explícitamente la acción tomada; en caso de no haber estado bloqueado el usuario previamente también debe notificarlo.

Debe esperar a que se presione enter para volver a mostrar el menú.

```
Unlock User                                user: AdminB19387
```

---

```
Username to unlock: Juan2000
```

```
>>      Successfully unlocked user      <<
```

```
>>      Press Enter to go back to menu  <<
```

```
Unlock User                                user: AdminB19387
```

---

```
Username to unlock: Juan2000
```

```
>>      Error, User wasn't locked      <<
```

```
>>      Press Enter to go back to menu  <<
```

### ~~Promote user to admin~~

Permitirá que un usuario posea los permisos de administración completos del sistema, a excepción de promote y demote, es decir, solo el usuario administrador general podrá promover o degradar usuarios admin.

```
Promote to admin                          user: AdminB19387
```

---

```
Username to promote: Juan2000
```

Se debe notificar explícitamente la acción tomada; en caso de haber contado previamente con los permisos de administrador también debe notificarlo.



Debe esperar a que se presione enter para volver a mostrar el menú.

```
Promote to admin          user: AdminB19387
```

---

```
Username to promote: Juan2000
```

```
>>      Successfully promoted user      <<
```

```
>>      Press Enter to go back to menu  <<
```

```
Promote to admin          user: AdminB19387
```

---

```
Username to promote: Juan2000
```

```
>>      Error, User already admin      <<
```

```
>>      Press Enter to go back to menu  <<
```

## ~~Demote user from admin~~

Permitirá remover los permisos de administración del sistema a un usuario específico

```
Demote from admin          user: AdminB19387
```

---

```
Username to demote: Juan2000
```

Se debe notificar explícitamente la acción tomada; en caso de no contar previamente con los permisos de administrador también debe notificarlo.

Debe esperar a que se presione enter para volver a mostrar el menú.

```
Demote from admin          user: AdminB19387
```

---

```
Username to demote: Juan2000
```

```
>>      Successfully demoted user      <<
```

```
>>      Press Enter to go back to menu  <<
```

```
Demote from admin          user: AdminB19387
```

---

```
Username to demote: Juan2000
```

```
>>      Error, User wasn't admin      <<
```

```
>>      Press Enter to go back to menu  <<
```

## ~~Usuarios ADMIN~~

Existirá un usuario ADMIN general para cada sección, este responderá a:

- Carnet del estudiante: 200012345
- Sección: A|B
- Identificador admin: ADM

Según el ejemplo anterior tendríamos el usuario: 200012345BADM

El password será el carnet del estudiante en orden inverso, para el ejemplo: 543210002

Este usuario no puede estar en memoria si no se encuentra en uso, deberá ser almacenado en el archivo de credenciales `users.gal`

## Métodos de ordenamiento

El objetivo de la visualización, en modo video, de los métodos de ordenamiento es validar la correcta implementación de los algoritmos solicitados

Para poder tener derecho a la calificación de estos se deberá tener la representación animada del método de ordenamiento implementada; de lo contrario se tendrá una nota de 0 puntos.

El seleccionar cualquier ordenamiento derivará en un submenú que solicitará el sentido del ordenamiento; mostrará el nombre del ordenamiento seleccionado en el título del submenú

```
<Nombre del ordenamiento> Sort    user: Juan2000
```

---

```
F1. Ascending
```

```
F2. Descending
```

```
F9. Go back
```

Luego de seleccionar el ordenamiento se deberá indicar la métrica a mostrar.

```
<Nombre del ordenamiento> Metric user: Juan2000
```

---

F1. Points  
F2. Time  
F9. Go back

Por último solicitará la velocidad a la que se ejecutará el ordenamiento.

```
<Nombre del ordenamiento> Speed user: AdminB19387
```

---

F1. 0  
F2. 1  
F3. 2  
F4. 3  
F5. 4  
F6. 5  
F7. 6  
F8. 7  
F9. Go back

Al ingresar se mostrará la grafica inicial (en modo video), mostrando la métrica seleccionada y la gráfica que la representa, en el orden como esté en el archivo de scoreboard; se debe mostrar un mensaje que indique que se presione la tecla "HOME" para iniciar, el contador de tiempo del algoritmo iniciará hasta luego de presionar la tecla "HOME" y se detendrá al finalizar el ordenamiento.

Para representar las barras en pantalla se requiere:

- Serán barras horizontales, de izquierda a derecha
- El alto mínimo de una barra será de 7 pixeles
- Si son muy pocas barras se debe ajustar el alto al espacio disponible en pantalla; esto quiere decir que si se representarán 4 barras, el alto de la pantalla disponible para gráficas se dividirá en 4.
- El valor que representa la barra debe ser visible numéricamente en pantalla, a la izquierda de la misma, con espacio mínimo para 3 números pero máximo de 5; el desarrollador debe elegir la mejor forma de coincidir el texto con la barra representada de manera no exista confusión
- El ancho de las barras serán calculado en base al mayor número en de la métrica evaluada
- El máximo número de columnas a representar en pantalla será de 20 (por motivos de espacio), cuando sea este el caso, se tomarán las últimas 20 líneas del archivo de scoreboard para realizar la animación.

Para la ejecución del ordenamiento se requiere:

- Identificar con flechas las barras que están siendo evaluadas en un momento específico, esto se hará a la izquierda del número que pertenece a cada barra
- Realizar una pausa antes de avanzar al siguiente paso
- En caso ser viable el intercambio de las barras
  - Colocar el color de las barras involucradas en rojo
  - Hacer una pausa antes de ejecutar el movimiento
  - Realizar el movimiento
  - Hacer una pausa luego de ello
  - Devolver el color original a la barra

En pantalla se requiere:

- Nombre del ordenamiento
- Dirección del ordenamiento (Flecha arriba -Ascendente- o abajo -Descendente- )
- Tiempo transcurrido actualmente en el algoritmo actual
- Velocidad seleccionada
- Nombre de la métrica evaluada (para el ejemplo fue TIME)



Al finalizar el ordenamiento se deberá quedar mostrando la gráfica ordenada, el tiempo que tomó el completarse el algoritmo, los datos solicitados en pantalla y un mensaje que indique que se presione la tecla "END" para salir.

## Reporte

La ejecución de cada algoritmo de ordenamiento genera un reporte en archivo donde siempre se deben mostrar todos los datos que se guardan por usuario pero ordenados por la métrica seleccionada y contendrán únicamente el top10 de los datos tomados para la ejecución del ordenamiento.

## Ejemplo de archivo por puntos ascendente

```
-----  
Universidad de San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Ciencias y Sistemas  
Arquitectura de Compiladores y ensambladores 1  
Sección <A|B>  
-----
```

```
      < Nombre del desarrollador >  
      < Registro Académico >
```

```
-----  
Tipo: < Ordenamiento >      Sentido: < Sentido >  
Fecha: 03/03/2022           Hora: 07:05:15  
=====
```

Rank	Player	N	Points	Time
1	Juan2000	3	280	300
2	Pedro9876	3	240	290
3	Luis11111	3	230	280
4	Walter8788	3	200	280
5	Juan2000	3	199	250
6	Juan2000	3	195	225
7	Fernando09	2	170	210
8	Jorge5555	2	150	203
9	Javier000	2	130	201
10	Alejandro	1	80	150

El nombre del archivo del reporte es `LASTSORT.REP` y será reemplazada cada vez que se genere uno nuevo.

Debido a que el mismo reporte será reemplazado sin importar el ordenamiento, es obligatorio que el tipo y sentido de ordenamiento se encuentre dentro del reporte.

Los reportes que no cuenten con la identificación, completa, del estudiante o tipo y sentido tendrán una nota de cero puntos.

## Archivos

- Los usuarios se guardarán en un archivo llamado `users.gal`
- El scoreboard se guardará en un archivo llamado `scores.gal`
- El contenido de los archivos queda a discreción del desarrollador
- Ningún dato de scoreboard o usuarios estará previamente en memoria, ni el administrador general, todos deben ser leídos desde archivo; de lo contrario se tendrá una nota de cero puntos en dónde se evalúen su manejo

Se recomienda trabajar en archivos separados por comas

## Juego

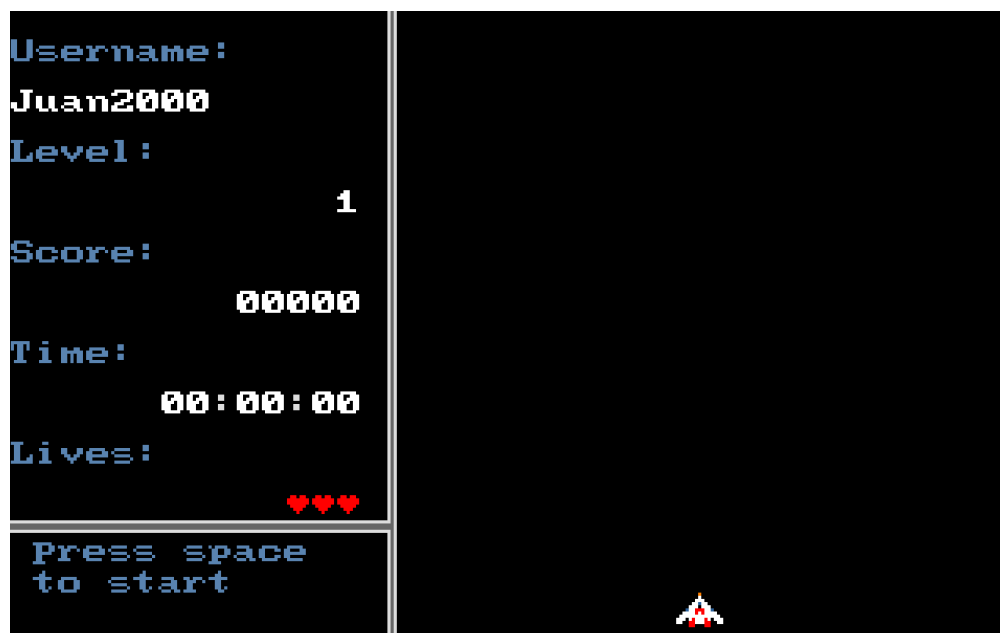
Se trata de una variación del juego galaga.

En su pantalla del juego debe incluir información del nombre del jugador actual, nivel en curso, punteo hasta el momento, tiempo (minutos, segundo, centisegundo) y vidas.

A pesar que el diseño del área de juego queda a discreción del estudiante se solicita que toda la información previa se encuentre en pantalla y que el área de juego sea, al menos, de 193 pixeles de ancho por 200 de alto.

## Iniciar Nivel

Al inicio de cada nivel se deberá mostrar todos los datos detallados previamente, en el último estado obtenido en el nivel previo, además de la nave correspondiente al nivel y un mensaje de presionar espacio para iniciar.



Hasta que se presione espacio para iniciar se mostrarán los enemigos para el nivel en curso.

## Pausa

Se contará con la tecla escape (ESC) como botón de pausa.

Al estar en un juego en curso, si se presiona la tecla detendrá el juego momentáneamente.

Al estar en pausa y presionar la tecla **ESC** saldrá al menú del usuario específico, de ser así se guardará en ese momento el score obtenido.

Si se presionase la tecla espacio, mientras se está en pausa, reanudará el juego en el estado previo a la pausa.

Durante la pausa deberá mostrar un mensaje que indique la función de cada tecla.

## Enemigos

Los enemigos serán de tres tipos, cada uno más resistente que el anterior, para diferenciarlos deberán de ser de distintos colores entre sí, se requiere que sus dimensiones sean de **8 x 8 pixeles**.

Color	Resistencia
Azul	1
Verde	2
Amarillo	3

Luego de un tiempo que el desarrollador considere prudente, pero que sea visible dentro de la jugabilidad, los enemigos uno por uno lanzarán un ataque kamikaze verticalmente hacia abajo, en caso de no impactar la nave se perderán en la parte baja de la pantalla, en caso de impactar, se perderá una vida (se debe quitar la vida de pantalla) y **se destruirá el enemigo sin sumar puntos**.

El puntaje de cada tipo de enemigo queda a discreción del estudiante, se recomienda que el punteo sea coherente con la resistencia que tiene cada uno y tomando en cuenta la cantidad de enemigos a destruir luego de 3 niveles y el dominio de numeración con 16 bits. Si se desea se pueden llegar a valores de 5 dígitos reduciendo el ancho de las columnas en los métodos de ordenamiento.

## Nave

La nave estará en la parte baja de la pantalla y se podrá mover, únicamente, de forma horizontal (izquierda a derecha); no podrá salirse del marco de juego que debe estar dibujado para delimitar claramente el área.

Se iniciará con una nave con un cañón, cada nivel agregará un cañón más a la nave y cada cañón será más fuerte que el inicial. Los cañones que no pertenecen a un nivel no deben estar presentes.

- En el nivel uno se inicia con un cañón central.
- En el nivel dos se agrega un cañón a la izquierda del primer cañón
- En el nivel tres se agrega un cañón, a los dos anteriores, a la derecha del primer cañón.



Aunque el diseño de la nave es a criterio del desarrollador se requiere un ancho de 15 pixeles exactamente y los cañones estarán en los pixeles 1, 8 y 15.

Los daños son incrementales en uno por cada cañón y las balas deben corresponder a un color específico como se detalla a continuación:

Cañón	Daño	Color
1	1	Gris Claro
2	2	Morado
3	3	Rojo

Cada bala destruirá un enemigo si su daño es igual o mayor al del enemigo; en caso la bala no pueda destruir al enemigo este no disminuirá su resistencia.

Si el daño posible de la bala es mayor que el del enemigo, la bala continuará su camino, habiendo destruido al enemigo, el daño posible de la bala luego del impacto será igual a su daño previo menos el la resistencia del enemigo que haya destruido.

Si una bala del cañón 3 (daño 3) impacta con un enemigo verde (resistencia 2) destruirá el enemigo, la bala continúa su camino con un daño posible de 1

Si esa misma bala continúa su camino e impacta con otro enemigo verde (**resistencia 2**) no destruirá el enemigo, **el enemigo seguirá teniendo resistencia 2** y la bala ya no podrá continuar.

La bala tendrá dimensiones de **1 pixel de ancho por 3 de alto**

Cada cañón contará con una tecla de disparo que solo funcionará si el cañón está presente.

Cañón	Tecla
1	v
2	b
3	n

Se podrá disparar los tres cañones al mismo tiempo con la tecla espacio.

En pantalla, en un momento dado, **solo se puede tener una bala de cada tipo**, si se presiona varias veces una tecla que requiera el disparo de una bala pero esa bala ya se encuentra en pantalla estas solicitudes deberán de ser ignoradas; si fuese la tecla espacio se ignorará únicamente las balas que estén previamente en pantalla, las otras deberán ser disparadas.

## Niveles

Cada nivel iniciará con, al menos, **7** enemigos de cada tipo multiplicados por el nivel. (Pueden ser más)

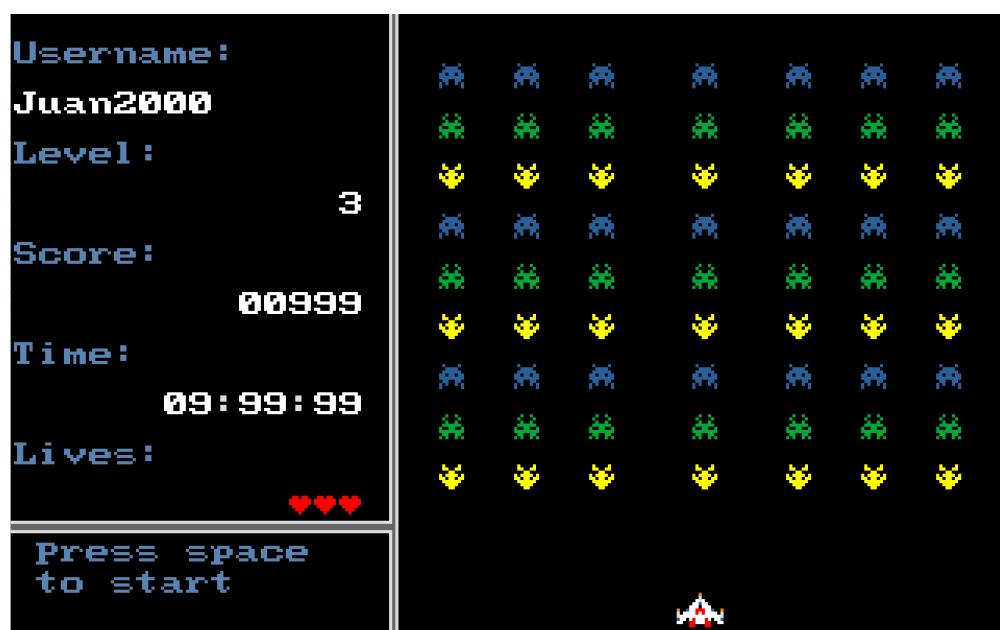


Nivel	Azul	Verde	Amarillo
1	7	7	7
2	14	14	14
3	21	21	21

Estos deben estar distribuidos equitativamente de manera horizontal, deben estar ordenados por filas, dónde los más resistentes estarán al frente.



Para facilitar la implementación, se puede repetir la distribución de enemigos de la imagen anterior por cada nivel; a continuación un ejemplo del nivel 3.



Aspectos a considerar:

- La velocidad de los eventos debe aumentar con cada nivel.
- Cada nivel empieza en pausa esperando que el usuario indique el inicio del nivel.
- El nivel terminará cuándo se hayan eliminado todos los enemigos en pantalla.
- El juego terminará cuándo se hayan perdido todas las vidas o superado el nivel 3.
- Al terminar el juego se escribirá el score a disco.
- Los scores serán almacenados en orden cronológico de juego, el último jugado deberá estar al final del archivo.
- Solo se puede guardar un scoreboard y será el general; no está permitido guardar un scoreboard ordenado previamente.

## Consideraciones

---

- ☐ El usuario Administrador general no puede degradarse a sí mismo
- ☐ El usuario Administrador general no puede jugar
- ☐ Cualquier otro usuario admin sí puede jugar
- ☐ Siempre que esté un usuario logueado debe mostrarse el nombre de usuario en el encabezado del menú, en caso que en el presente enunciado, por error u olvido, no se muestre en alguno de los ejemplos, el desarrollador debe recordar colocarlo.
- ☐ Dónde se requiera tomar en cuenta el tiempo se deben mostrar las fracciones de segundo y el tiempo debe coincidir con el paso del tiempo real
- ☐ Para ser válidos los reportes a la hora de la calificación deben incluir la identificación del desarrollador completa, la fecha y hora de generación del reporte, incluyendo los segundos además de la identificación y sentido del algoritmo aplicado; de lo contrario tendrá una nota de 0 puntos en donde se evalúe cada uno

=====

## Referencias

Titulo	Referencia
Tabla ASCII	<a href="https://elcodigoascii.com.ar">https://elcodigoascii.com.ar</a>
Libro de Texto	Brey, Barry B. Microprocesadores Intel 8086/8088, 80186/80188, 80286, 80386, 80486, Pentium, Pentium Pro Processor, Pentium II, Pentium III, and Pentium 4 Arquitectura, Programación e Interfaz. Pearson Prentice Hall. Séptima Edición.
Libro sugerido	Assembly language for x86 processors. Irvine - Pearson - 2015

## Requerimientos mínimos

1. Debe ser jugable por lo menos un nivel completo, hasta terminar ese nivel, ya que es la fuente de todos los datos que se evaluarán, de lo contrario no se calificará.
2. Directiva .MODEL máxima a usar **small**
3. Lectura de archivos
4. Generación de reportes
5. Se debe entregar los manuales técnico y de usuario, de lo contrario se asumirá que el estudiante copió.

Debido a la naturaleza de ensamblador, los resultados completos solo se pueden evaluar si se tienen las entradas y salidas correctamente implementadas.

## Observaciones y Restricciones

- **La realización del proyecto es de forma individual**
- No se calificará ninguna práctica que no tenga la identificación completa del estudiante detallada como aparece al inicio del presente enunciado
- El código del programa debe ser estrictamente ensamblador
- No se permite el uso de alguna librería.
- Se debe presentar el proyecto en Dosbox.
- Lenguaje ensamblador a utilizar MASM 6.11
- No está permitido el uso de estructuras de control if, if else, while, repeat, for.
- No está permitido el uso de STRUCT
- Se evaluará que cada tecla solicitada explícitamente sea la que se use para la manipulación del proyecto.
- El día de la calificación se harán preguntas, modificación de código sobre aspectos utilizados en la elaboración del proyecto, las cuales se considerarán en la nota final.
- Subir todos los archivos para que la práctica funcione así como los manuales para antes de 23:59 horas del jueves 28 de abril.
- **Copias parciales o totales tendrán una nota de 0 puntos y los involucrados serán reportados a la Escuela de Ciencias y Sistemas**
- Todos estos entregables deben de ser adjuntados en un archivo de extensión rar. El archivo debe ser nombrado como [ACE1]Pro2\_#Carnet.rar.