**Portal de Super Space invasion war 3000**

****

Yino Segura 8-912-2030

Eloy Gonzales 8-914-1631

Hector Vasquez 8-913-2324

**Tabla de Contenidos**

**1. Introducción**

1.1 Propósito

1.2 Alcance

**2. Estructura de Modelo de Caso de Uso**

2.1 Caso de Uso

2.2 Diagrama de Caso de Uso

2.3 wireframe

2.4 Pantallas

**3. Diagrama General de los Casos de Uso**

**Resumen del modelo de casos de uso**

# **Introducción**

Este documento nos muestra los diferentes casos de uso, que se planean desarrollar para la implementación del proyecto.

1.1 **Propósito**

En este documento se muestra cómo se estructura el modelo de casos de uso y brinda una descripción de sus elementos: actores y casos de uso. Además de pantallas iniciales y finales de la página web y del juego

1.2 **Alcance**

El alcance de este documento presenta desde el Caso de Uso de acceso al Portal hasta el Caso de ver historial de compras.

Este proyecto crea un portal para el juego Super space invansion war 3000 y mantener un seguimiento de todas las compras, la página estará ligada a un juego que se descargara desde la misma, además de realizar consultas de todas las compras y editar información.

Este documento está disponible para los directores del proyecto y el personal encargado de la realización de esta aplicación.

# 

# 

# 

# 

# 

# 

# **2. Estructura del modelo de casos de usos**

En esta sección se describen uno a uno los casos de uso que constituyen el sistema. De acuerdo con el grado de importancia que se le asigne, serán descritos en orden, haciendo mención primero a aquellos de mayor relevancia. Además, se incluye el diagrama de casos de uso al igual que la responsabilidad de cada actor que presenta en el sistema.

**Caso de Uso:**

Página Principal (SSIW3000)

1. Nombre: Página Principal (SSIW3000)
2. Descripción: En esta página se promociona el juego, usuario puede descarga el juego desde esta.
3. Actores: Usuario
4. Precondiciones: ninguna
5. Flujo de eventos:
   1. Flujo básico:
      1. El sistema carga y muestra el contenido.
      2. Condición:
         1. El usuario presiona el botón *Descargar*. El Sistema sigue el flujo alterno *Navegar a Registrar*.
         2. El usuario presiona el botón *Iniciar Sesión*. El Sistema sigue el flujo alterno *Navegar a Iniciar Sesión.*
      3. Fin del caso de uso.
   2. Flujo alterno:
      1. Navegar a Registrar:
         1. El Sistema Redirige a la pantalla Iniciar Sesión.
      2. Navegar a Iniciar Sesión:
         1. El Sistema Redirige a la pantalla Registrar con el parámetro *source*, cuyo valor será ssiw3000.
         2. Fin del caso de uso

* **Escenario de Caso de Uso:**

El usuario entra al Portal.

El sistema despliega la Pantalla de Inicio.

El usuario descarga el juego

El sistema inicia la descarga

.

**Caso de Uso**

Iniciar Sesión

1. Nombre: Iniciar Sesión
2. Descripción: En esta página el usuario puede iniciar sesión en el sitio web.
3. Actores: Usuario
4. Precondiciones: No debe existir una sesión abierta.
5. Flujo de eventos:
   1. Flujo básico:
      1. El sistema carga y muestra el contenido.
      2. El usuario presiona el botón *Crear Cuenta*. Ir al flujo alterno *Navegar a Registrar*.
      3. El usuario llena los campos.
      4. El usuario presiona el botón *Iniciar Sesión*.
      5. El sistema envía los datos al servidor.
      6. El sistema valida los datos.
      7. El sistema registra la sesión si los datos son correctos.
      8. El servidor prepara respuesta.
      9. El servidor retorna respuesta. Condición:
         1. La respuesta contiene “success:false”. Indicar mensajes de error con la razón de error que indica la respuesta.
         2. La respuesta contiene “success:true”. Redireccionar a la pantalla de Perfil.
      10. Fin del caso de uso.
   2. Flujo alterno:
      1. Navegar a Registrar:
         1. El Sistema Redirige a la pantalla *Registrar*.
         2. Fin del caso de uso.

**Escenarios:**

El usuario ingresa a la página.

Elsistema: Presenta la pantalla de login con opción de ingresar.

El usuario Ingresa su id y password y presiona ingresar.

El sistema Valida si el id y password son correctos.

El Sistema Si son correctos los datos se le presentará las opciones de comprar mercancia, ver historial y editar información.

**Casos de usos:**

Pantalla Principal de SSIW3000 (Juego)

1. Nombre: Pantalla Principal de SSIW3000 (Juego)
2. Descripción: La pantalla que carga al iniciar el juego.
3. Actores: Usuario
4. Precondiciones: Ninguna.
5. Flujo de eventos:
   1. Flujo básico:
      1. El sistema carga y muestra la pantalla.
      2. El sistema valida que haya sesiones abiertas. Condición:
         1. Si la hay, deshabilitar el botón *Iniciar Sesión* y mostrar el nombre del usuario.
         2. Si no la hay, habilitar el botón *Iniciar Sesión*.
      3. El usuario presiona un botón de la pantalla:
         1. *Iniciar Sesión*:
6. El sistema muestra la pantalla *Iniciar Sesión.*
   * + 1. Jugar:
          1. El sistema muestra la pantalla *Selección de Nave*.
       2. Tienda:
7. El sistema muestra el *Panel Tienda*.
   * + 1. Opciones:
          1. El sistema muestra la pantalla *Opciones.*
       2. Salir:
8. El sistema detiene el tiempo de ejecución de la aplicación.

Fin del caso de uso.

**Escenarios:**

El usuario inicia el juego

El sistema valida que haya una sesión abierta

El sistema despliega las diferentes opciones: Iniciar sesion, jugar, tienda, opciones y salir.

**Casos de usos:**

Registrar

1. Nombre: Registrar
2. Descripción: En esta página el usuario puede crear una cuenta en la plataforma.
3. Actores: Usuario
4. Precondiciones: No debe existir una sesión abierta.
5. Flujo de eventos:
   1. Flujo básico:
      1. El sistema carga y muestra el contenido.
      2. El sistema valida que no haya sesiones abiertas. Condición:
         1. Si la hay, redireccionar a la pantalla *Perfil*.
         2. Si no la hay, continuar el flujo.
      3. El usuario presiona el botón *Iniciar Sesión*. Ir al flujo alterno *Navegar a Iniciar Sesión.*
      4. El usuario llena los campos.
      5. El usuario presiona el botón *Registrar*.
      6. El sistema envía los datos al servidor.
      7. El servidor valida los datos.
      8. El servidor crea la cuenta si los datos validados son correctos.
      9. El servidor prepara una respuesta.
      10. El servidor retorna respuesta. Condición:
          1. La respuesta contiene “success:false”. Indicar mensajes de error con la razón de error que indica la respuesta.
          2. La respuesta contiene “success:true”. Redireccionar a la pantalla de Perfil.
      11. Fin del caso de uso.
   2. Flujo alternativo.
      1. Navegar a Iniciar Sesión:

El sistema redirecciona a la pantalla *Iniciar Sesión.*

**Escenarios**

El usuario entra a la pagina

El usuario sin cuenta presiona registrar cuenta

El sistema despliega un formulario que el usuario llena

El sistema verifica que todos los campos estén llenos de manera correcta

El usuario se ha registrado

**Casos de usos:**

Iniciar Sesión

1. Nombre: Iniciar Sesión
2. Descripción: En esta pantalla el usuario puede iniciar sesión en el juego.
3. Actores: Usuario
4. Precondiciones: No debe existir una sesión abierta.
5. Flujo de eventos:
   1. Flujo básico:
      1. El sistema carga y muestra el contenido.
      2. El usuario presiona el botón *Crear Cuenta*. Ir al flujo alterno *Navegar a Registrar*.
      3. El usuario llena los campos.
      4. El usuario presiona el botón *Iniciar Sesión*.
      5. El sistema envía los datos al servidor.
      6. El sistema valida los datos.
      7. El sistema registra la sesión si los datos son correctos.
      8. El servidor prepara respuesta.
      9. El servidor retorna respuesta. Condición:
         1. La respuesta contiene “success:false”. Indicar mensajes de error con la razón de error que indica la respuesta.
         2. La respuesta contiene “success:true”. Mostrar la *Pantalla Principal*.
      10. Fin del caso de uso.
   2. Flujo alterno:
      1. Navegar a Registrar:

El Sistema abre la pantalla *Registrar* en el navegador.

**Escenarios**

El usuario entra a la pagina

El sistema despliega la página principal

El usuario presiona iniciar sesión e ingresa sus datos

El sistema valida los datos y de ser correctos muestra la página inicial

**Casos de usos:**

Opciones (Juego)

1. Nombre: Opciones (Juego)
2. Descripción: Pantalla en la que el usuario puede configurar el juego.
3. Actores: Usuario
4. Precondiciones: Ninguna.
5. Flujo de eventos:
   1. Flujo básico:
      1. El sistema carga y muestra la pantalla.
      2. El sistema valida si hay una sesión abierta. Condición:
         1. Si la hay, mostrar los botones *Mi Perfil* y *Cerrar Sesión*.
         2. Si no la hay, mostrar el botón *Borrar datos.*
      3. El usuario presiona un botón de la pantalla:
         1. Atrás:
            1. El sistema muestra la *Pantalla Principal*.
         2. Música (Activar/Desactivar):
            1. El sistema ejecuta el comando para habilitar o deshabilitar la música de fondo.
         3. Efectos de Sonido (Activar/Desactivar):
            1. El sistema ejecuta el comando para habilitar o deshabilitar los efectos de sonido.
         4. Mi Perfil:
            1. El sistema abre *Mi Perfil* en el navegador.
         5. Cerrar Sesión
            1. El sistema ejecuta una petición al servidor para desvincular la sesión.
            2. El sistema ejecuta el comando para eliminar los datos del juego.
         6. Borrar Datos
            1. El sistema ejecuta el comando para eliminar los datos del juego.

Fin del caso de uso

**Escenarios**

El usuario entra al juego

El usuario introduce sus datos y el sistema los valida

El sistema despliega la pantalla inicial del juego y el usuario toca opciones

El sistema despliega la pantalla de opciones

**Casos de usos:**

Panel Tienda (Juego)

1. Nombre: Panel Tienda (Juego)
2. Descripción: Menú en el que el usuario puede comprar elementos para el juego. PayPal es el método de pago utilizado en el sistema.
3. Actores: Usuario
4. Precondiciones: Ninguna.
5. Flujo de eventos:
   1. Flujo básico:
      1. El sistema carga y muestra el Panel.
      2. El usuario presiona un botón de la pantalla:
         1. Canjear código:
            1. Ir a flujo alterno *Canjear*.
         2. Atrás:
            1. El sistema oculta el *Panel Tienda*.
         3. Comprar (Aplica a cada elemento de la lista):
            1. Ir a flujo alterno *Comprar*.
      3. Fin de caso de uso
   2. Flujo alterno:
      1. Canjear:
         1. El sistema muestra el *Panel Canjear.*
         2. El usuario pega el código del portapapeles.
         3. El usuario presiona el botón Canjear.
         4. El sistema envía datos al servidor.
         5. El servidor valida los datos.
         6. El servidor retorna una respuesta.
         7. El sistema procesa la respuesta. Condición:
            1. Si la respuesta contiene “sucsess:false”, el sistema mostrará mensajes de error.
            2. Si la respuesta contiene “success:true”, el sistema mostrará ocultará el *Panel Canjear* y mostrará al usuario la compra que realizó.
      2. Comprar:
         1. El sistema envía datos al servidor.
         2. El servidor procesa los datos.
         3. El servidor retorna una respuesta.
         4. El sistema procesa la respuesta. Condición:
            1. Si la respuesta contiene “success:false”, el sistema mostrará mensajes de error.

Si la respuesta contiene “success:true”, el ejecutará en el navegador el URL que se indica en la propiedad “approval\_url”.

**Escenarios**

El usuario está en la página inicial del juego

El usuario presiona el botón tienda

El usuario canjea codigo y el sistema lo valida

El usuario comprar el objeto deseado

**Casos de usos:**

Selección de Nave (Juego)

1. Nombre: Selección de Nave (Juego)
2. Descripción: Pantalla en la que el usuario elige la *nave* con la que iniciará la partida.
3. Actores: Usuario
4. Precondiciones: Ninguna.
5. Flujo de eventos:
   1. Flujo básico:
      1. El sistema carga y muestra la pantalla.
      2. El usuario presiona las *flechas de navegación*.
      3. El sistema valida la navegación y carga la nave correspondiente.
      4. El sistema valida que la nave haya sido desbloqueada. Condición:
         1. Si no ha sido desbloqueada, mostrar el botón *Desbloquear*.
         2. Si está desbloqueada, mostrar el botón *Seleccionar.*
      5. El usuario presiona el botón:
         1. *Desbloquear:*
            1. Si el usuario cuenta con suficientes *diamantes.* La nave se desbloquea. Si no, ir al flujo alterno *Diamantes insuficientes.*
            2. El sistema guardar los datos.
         2. Seleccionar:
            1. El sistema muestra la pantalla *Jugar*.
         3. Tienda:
            1. El sistema muestra el *Panel Tienda*.
         4. Atrás:
            1. El sistema muestra la *Pantalla Principal*.
      6. Fin del caso de uso.
   2. Flujo alternativo.
      1. Diamantes insuficientes:
         1. El sistema muestra el mensaje “¡Obtén más diamantes!”*.*
         2. El usuario presiona el botón:
6. Cancelar:
   1. El sistema oculta el mensaje.
7. Tienda:

El sistema muestra el *Panel Tienda.*

**Escenarios**

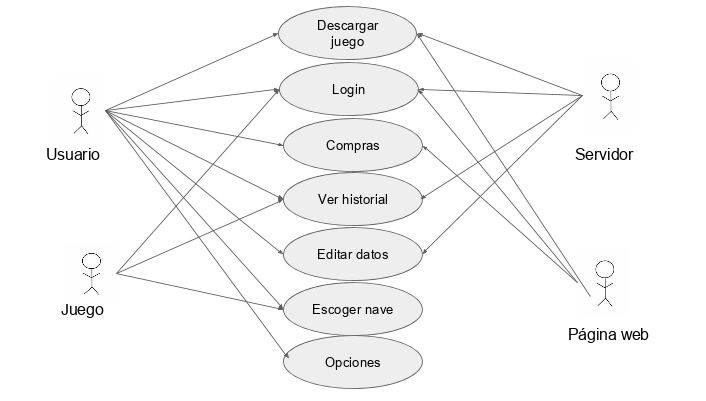
El usuario se encuentra en la pantalla inicial del juego

El usuario presiona selecionar nave

El sistema despliega las nave que posee el usuario

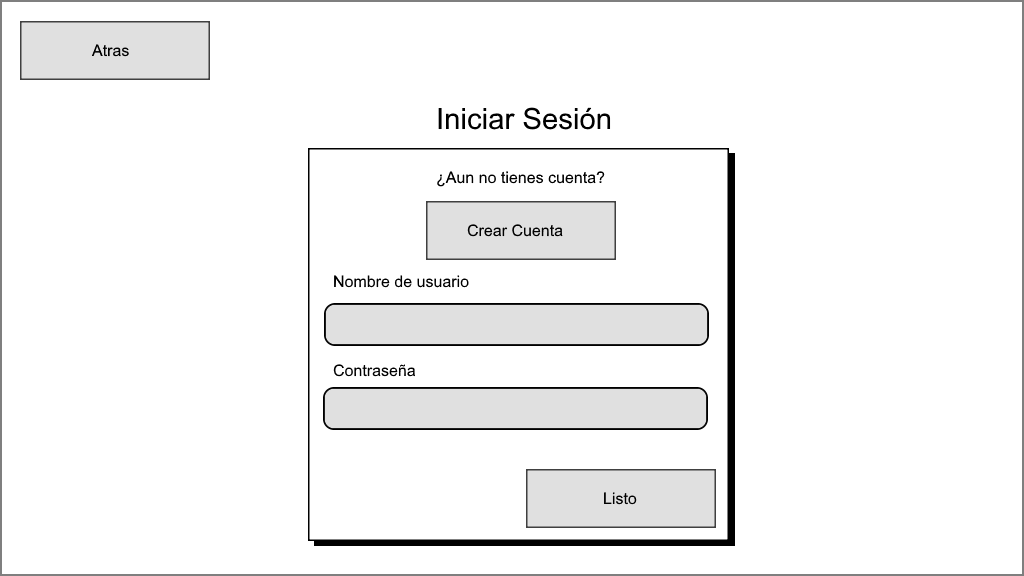
El usuario selecciona la que quiere usar

**2.2 Diagrama de Caso de Uso General**

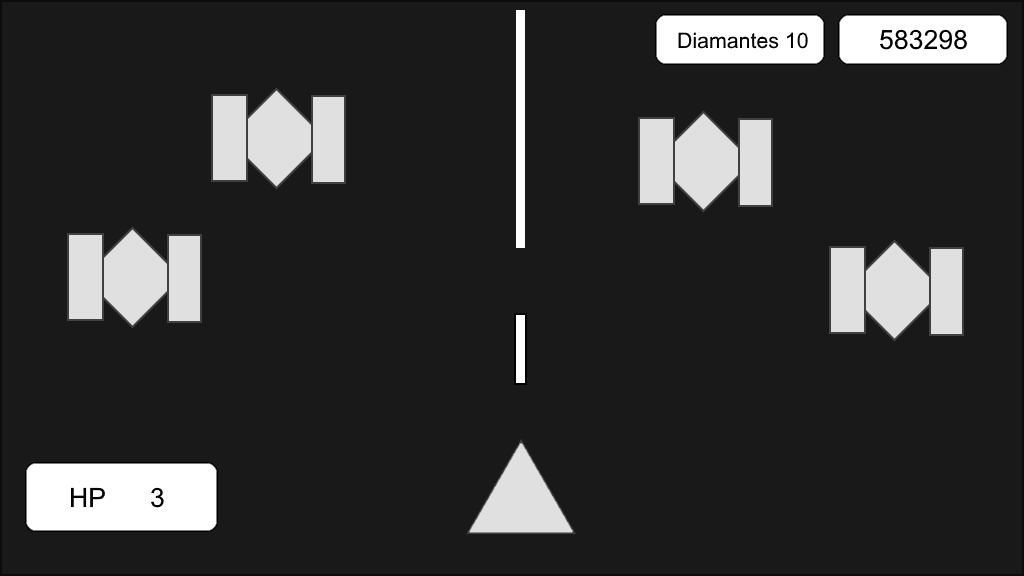


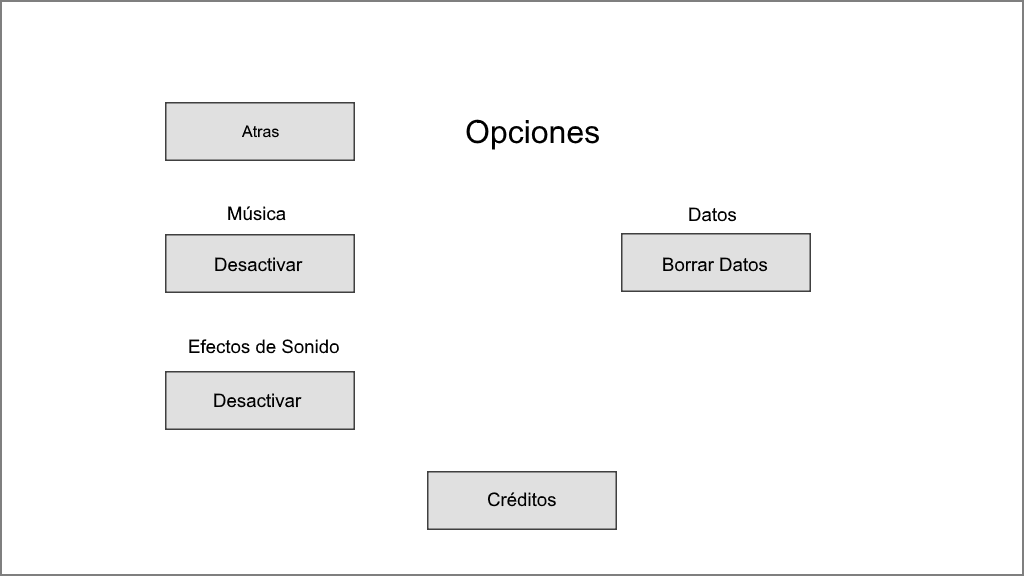
**2.3 Wireframes**

Diseño inicial de la página de iniciar sesión

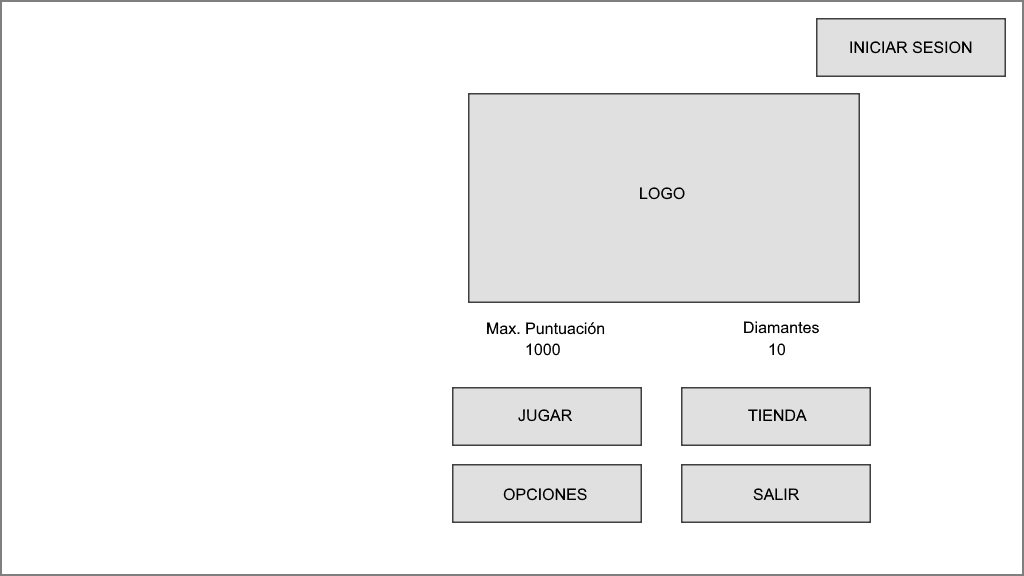


Pantalla inicial del juego

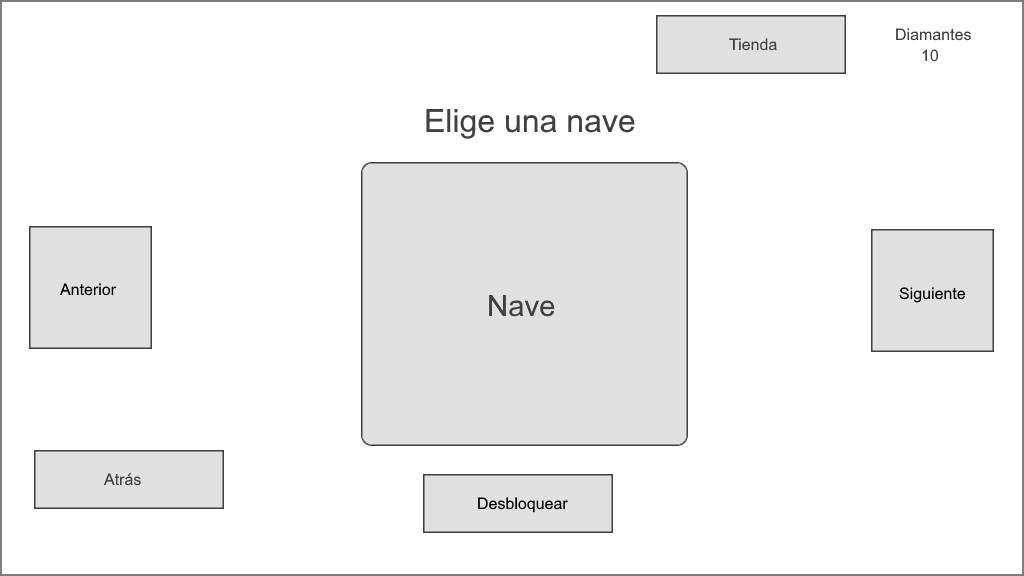


Pantalla inicial de opciones dentro del juego 

Pantalla inicial del juego

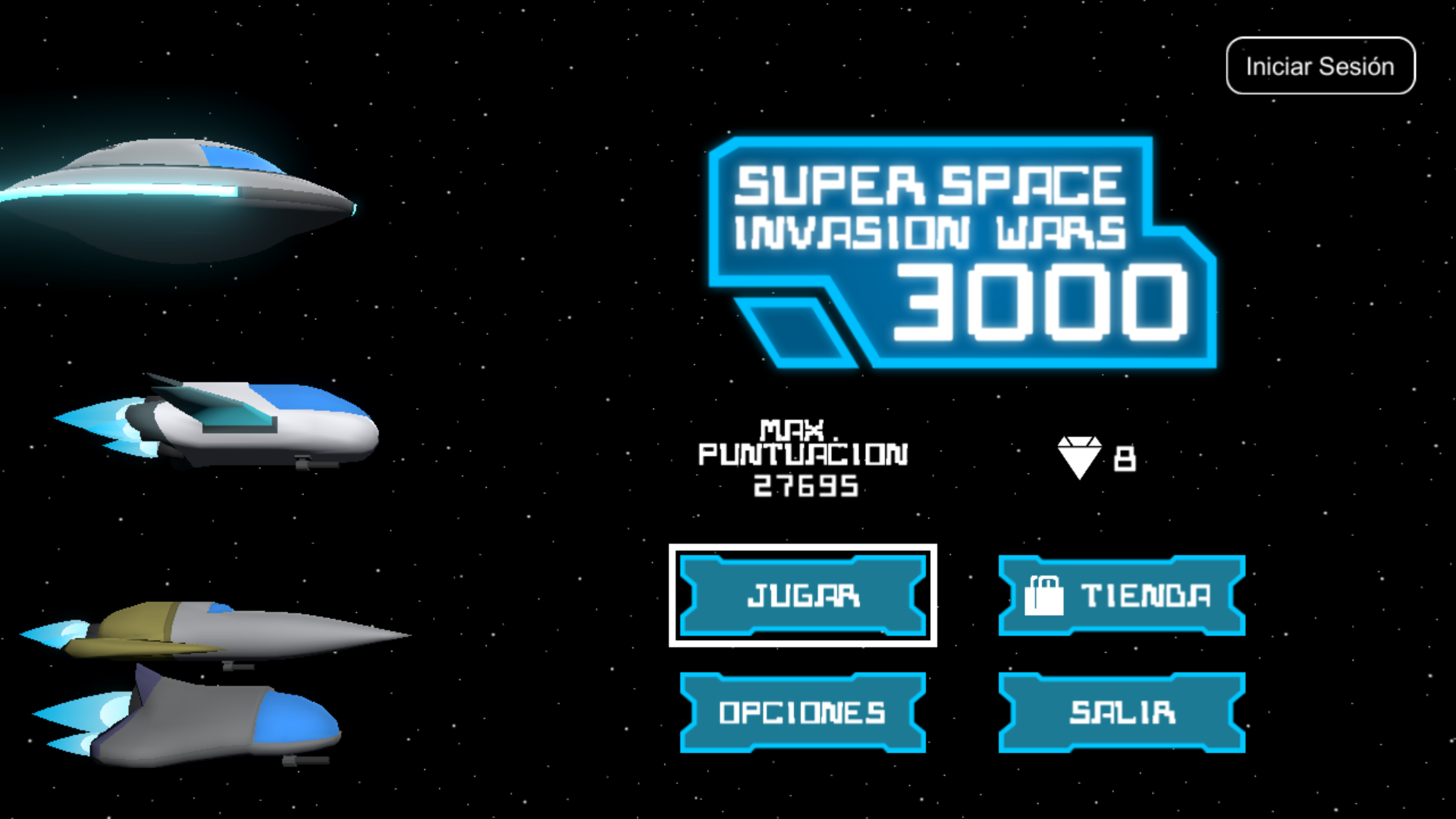


Pantalla inicial para seleccionar una nave

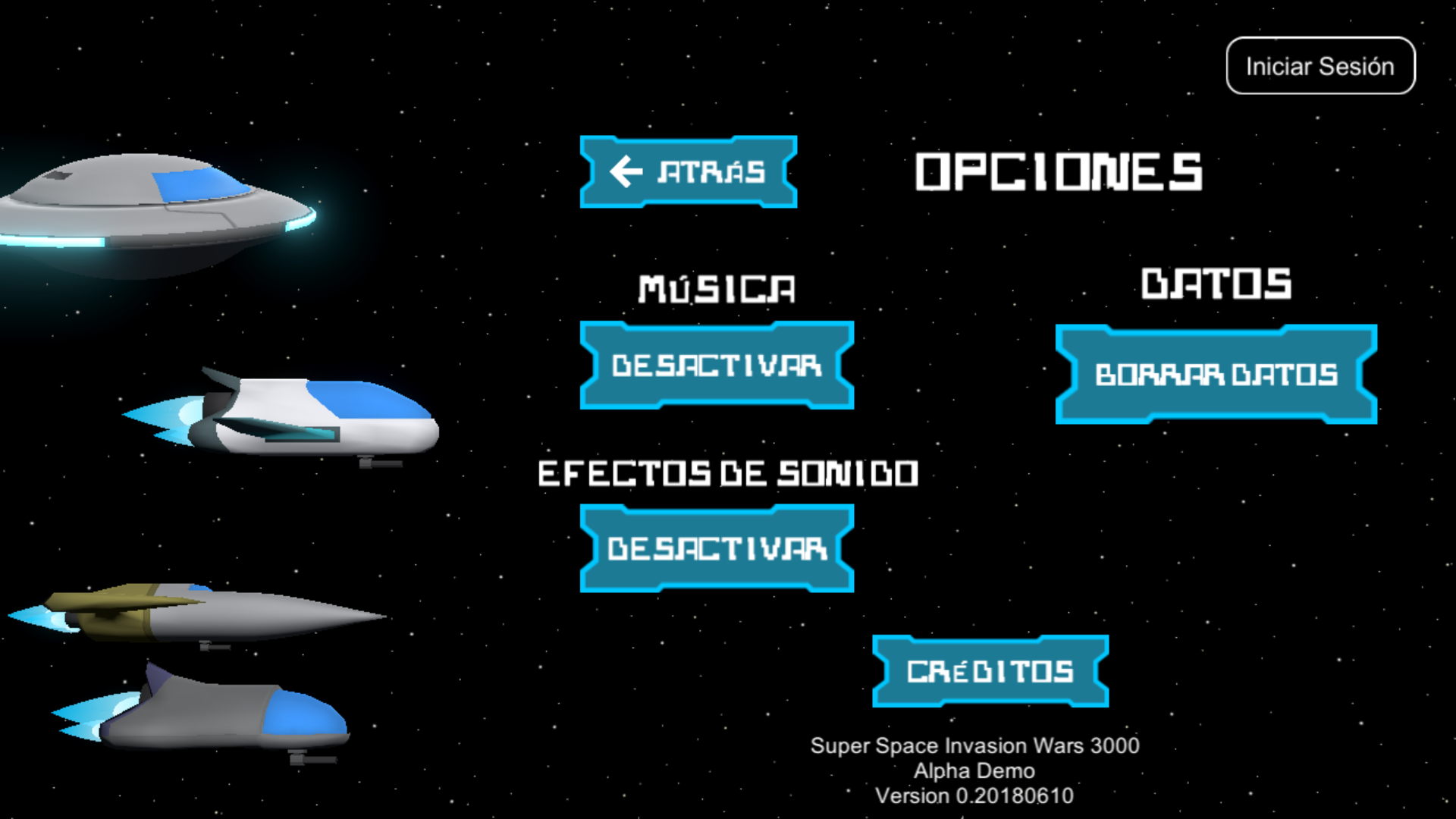


**2.4 Capturas del juego**

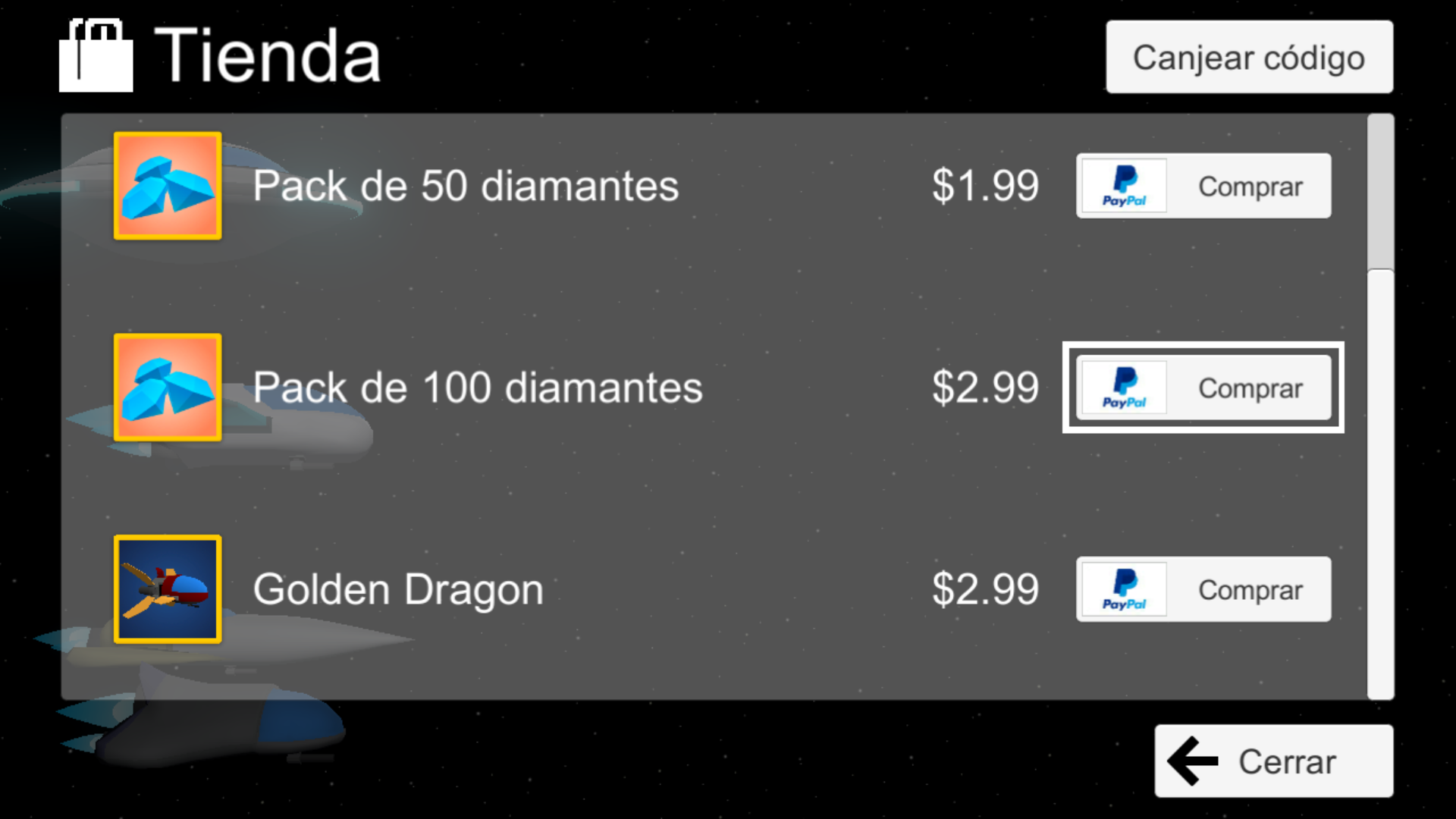
Pantalla inicial



Pantalla de opciones



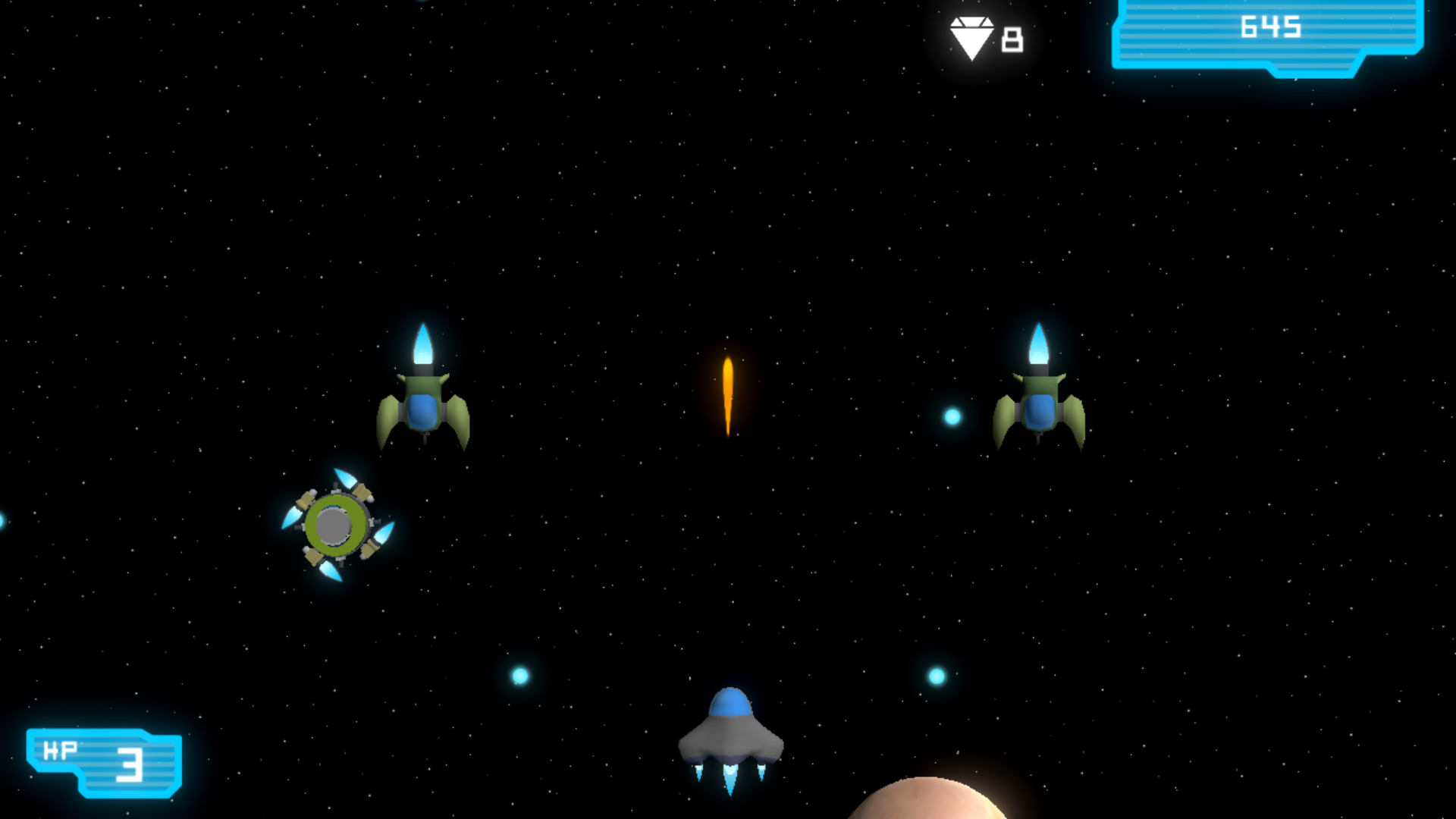
Pantalla de tienda



Pantalla escoge una nave



Pantalla general del juego



Pantalla de iniciar sesión

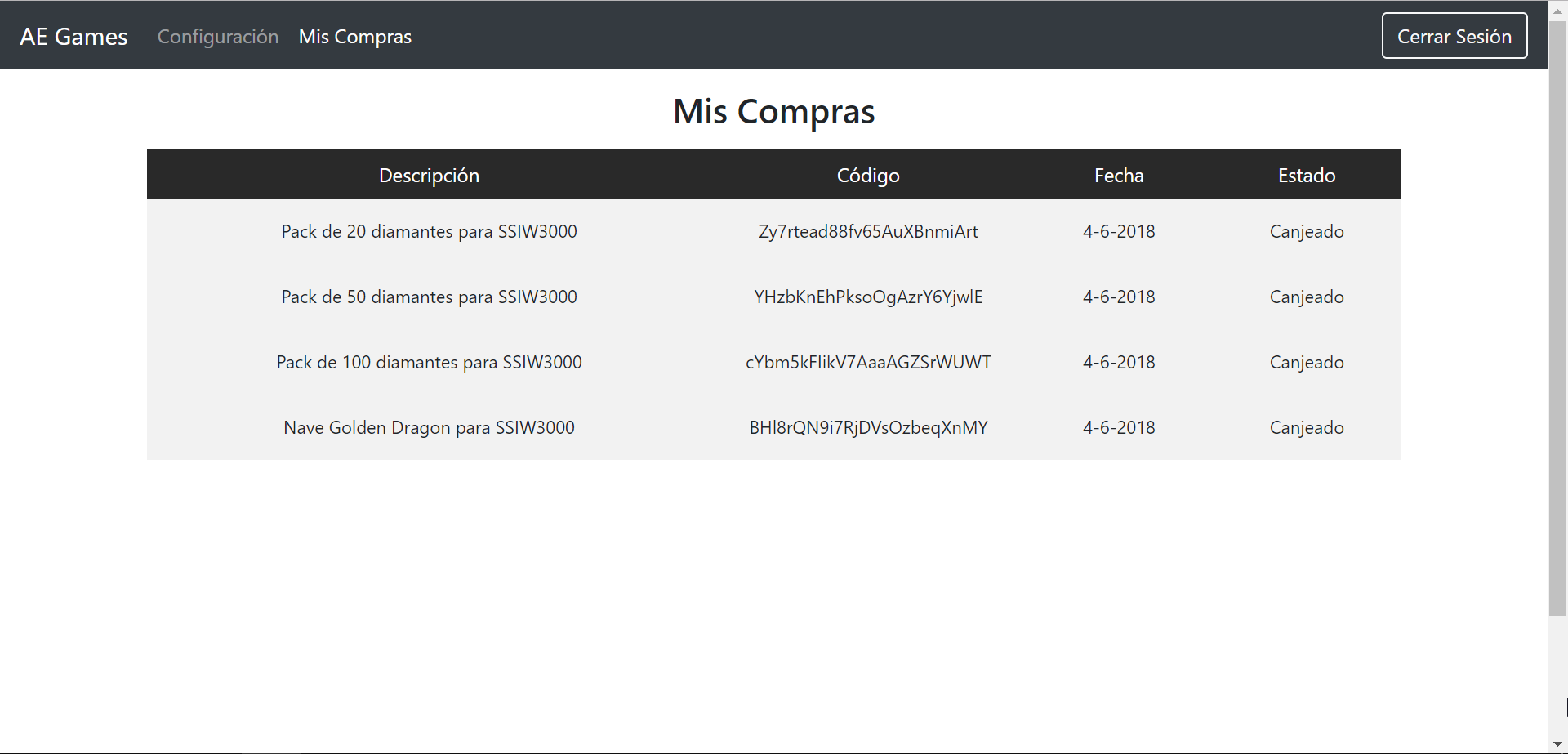


Pantalla de canjear código

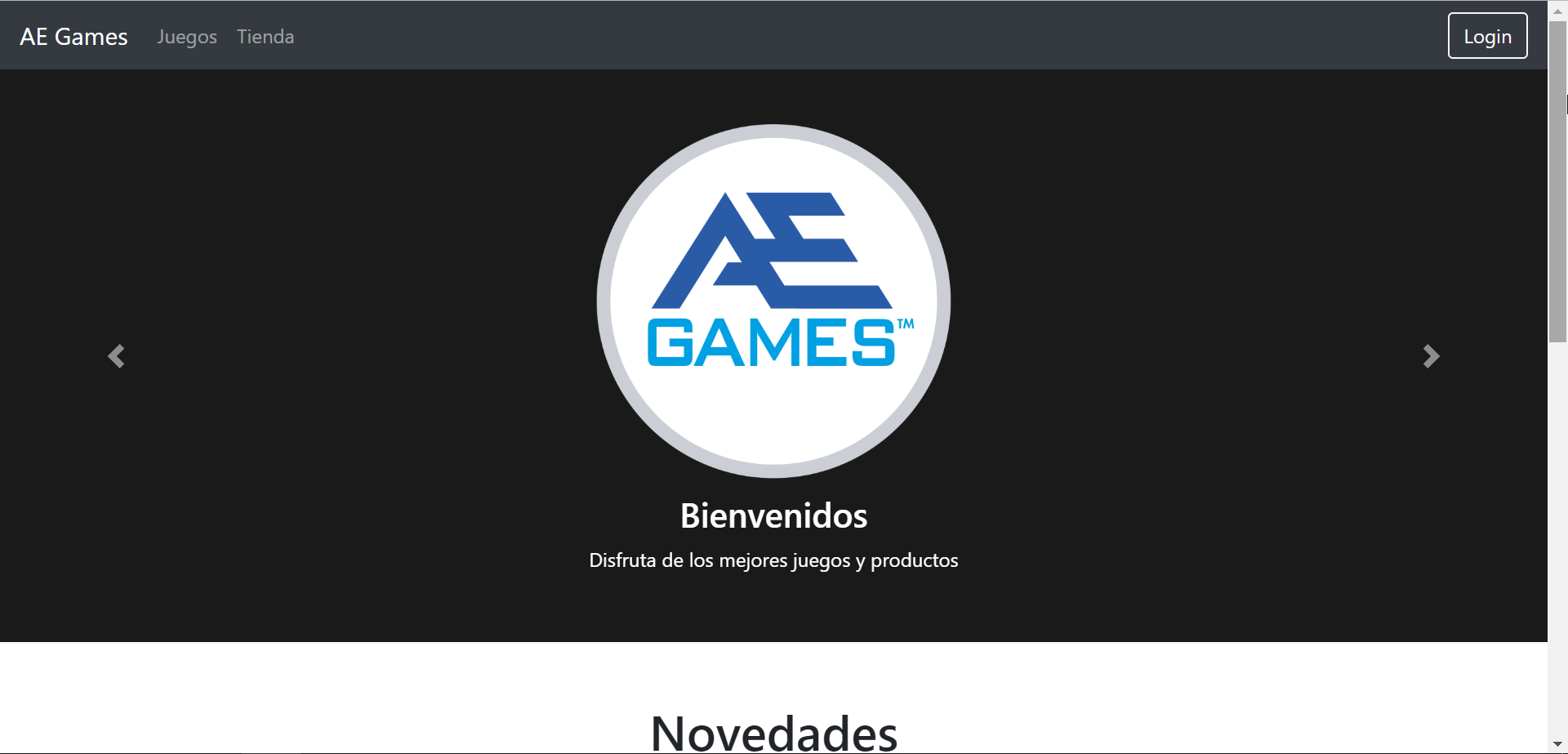


Capturas de la página web

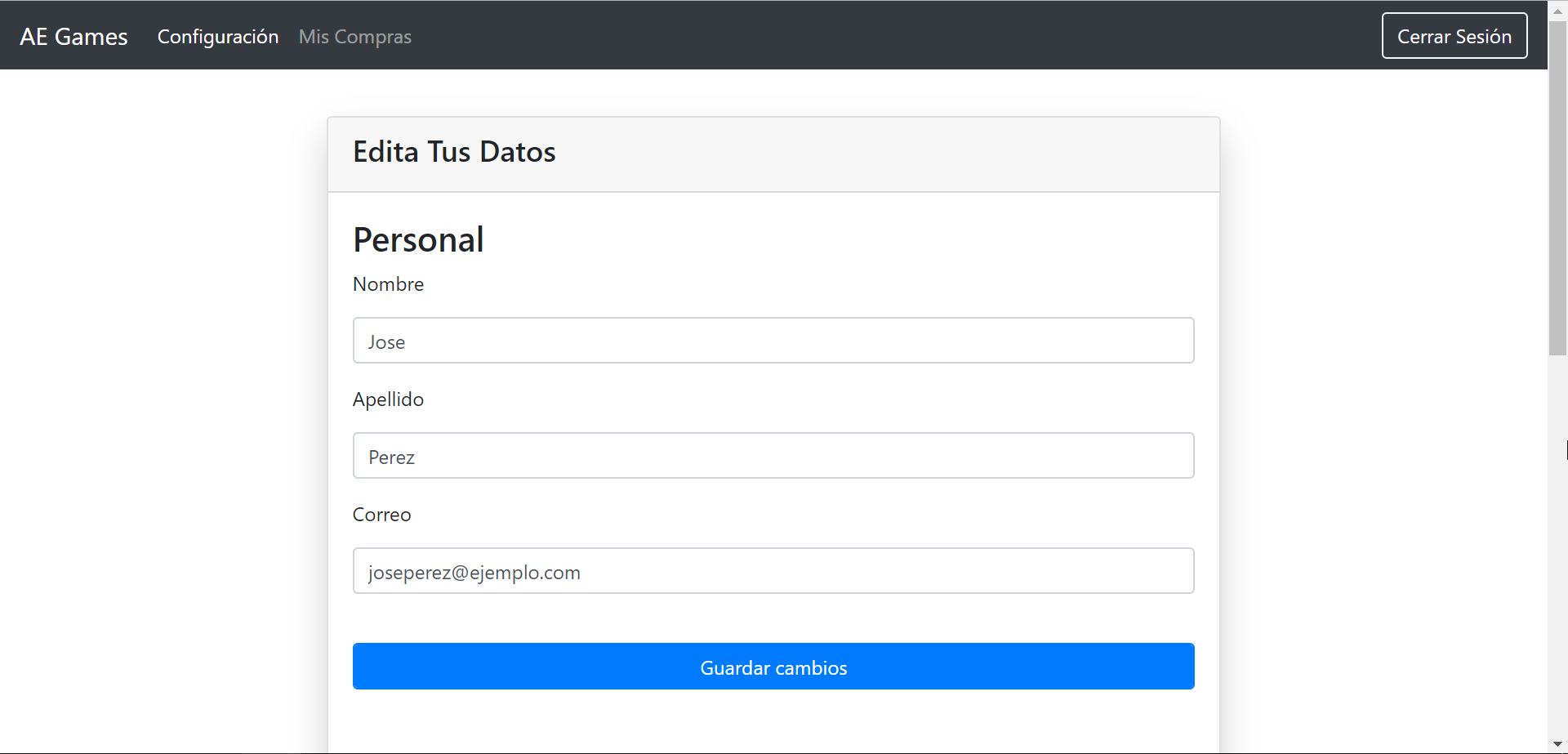
Página de mis compras



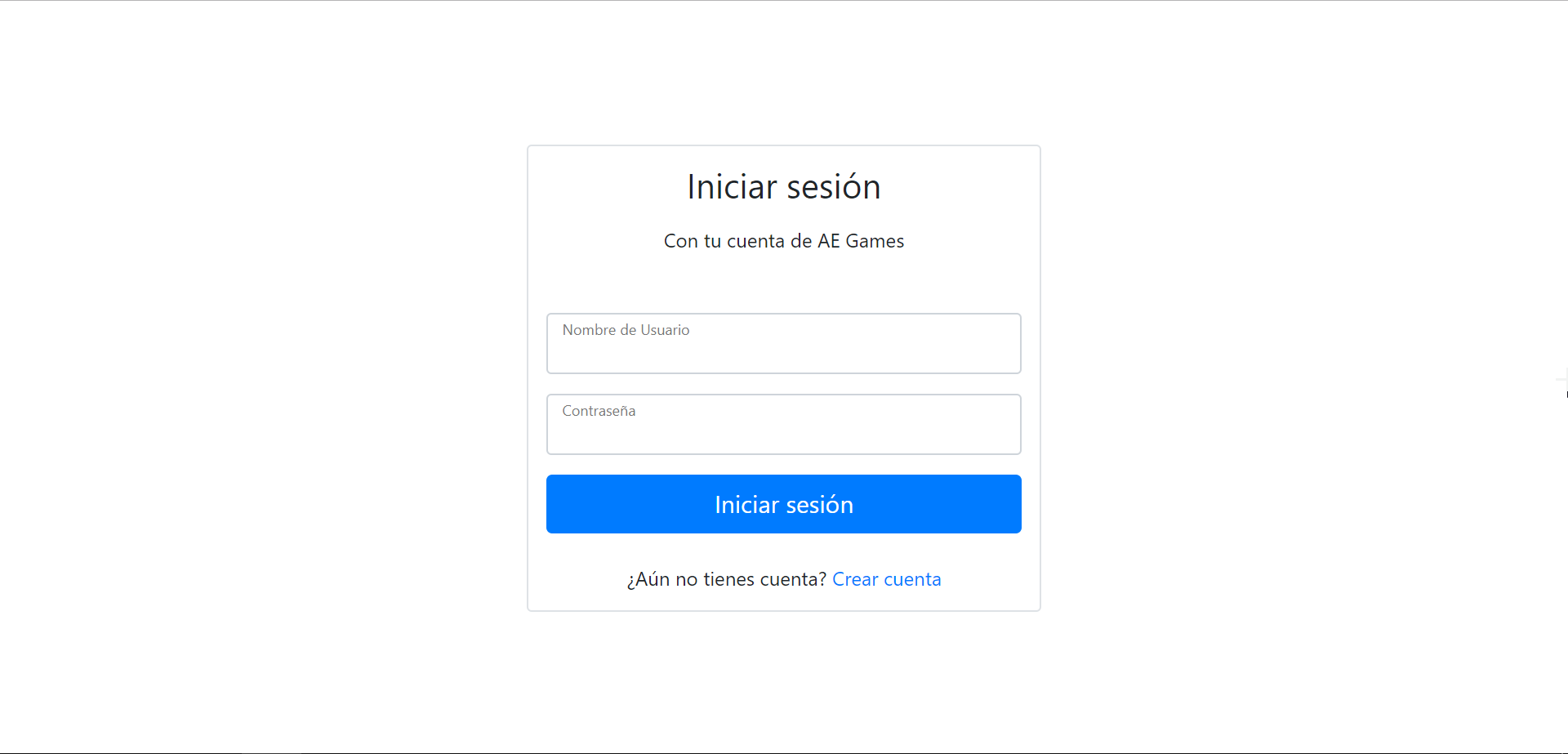
Captura de la página inicial



Captura de la página editar datos



Captura de la página iniciar sesión

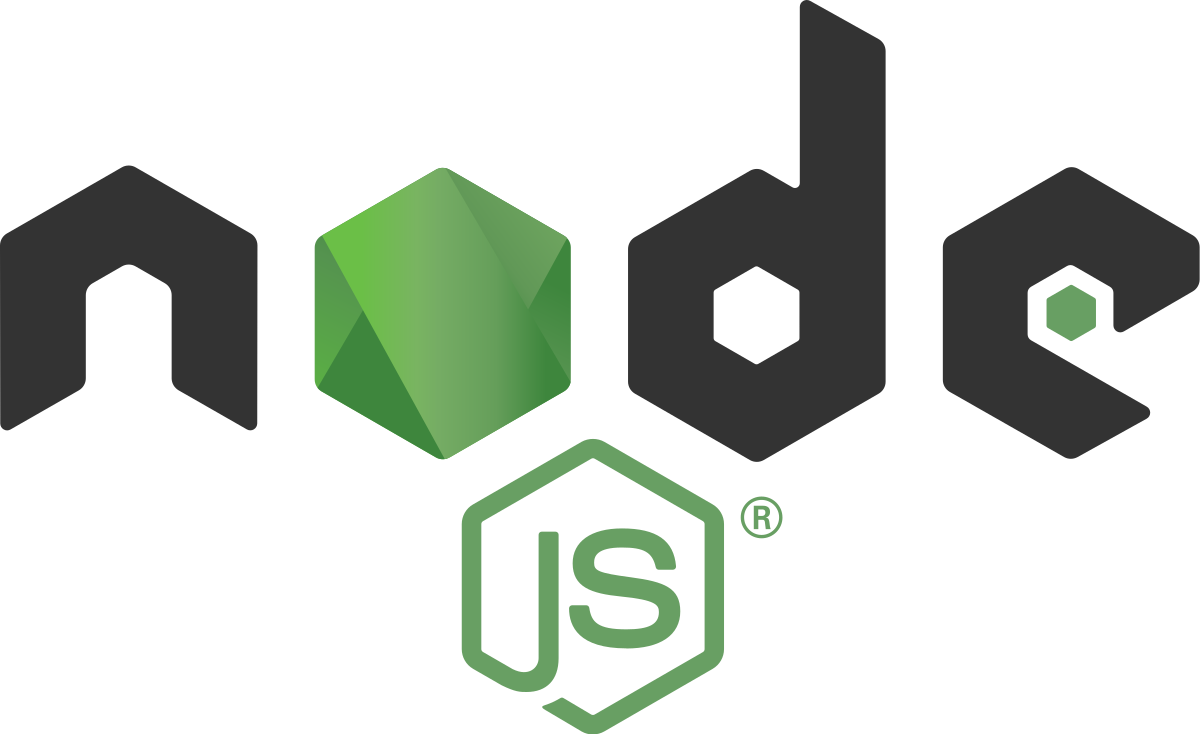


**Anexo**

Bootstrap es un framework desarrollado y liberado por Twitter que tiene como objetivo facilitar el diseño web. Permite crear de forma sencilla webs de diseño adaptable, es decir, que se ajusten a cualquier dispositivo y tamaño de pantalla y siempre se vean igual de bien. Es Open Source o código abierto, por lo que lo podemos usar de forma gratuita y sin restricciones.



Node.js® es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome.



**Conclusiones**

***Héctor Vásquez:***

El desarrollo de este proyecto pone en práctica todos nuestros conocimientos adquiridos alrededor de toda nuestra vida universitaria y nos presentó nuevas tecnologías que formarán parte de nuestros mayores herramientas a la hora de crear proyectos personales o en nuestros empleos.

***Yino Segura:***

Me gusto el desarrollo de este proyecto, ya que más allá de implementar lo aprendido en el curso. Se puso en práctica los conocimientos adquiridos a lo largo de la carrera.

Me resultó interesante la manera en que se desenvuelve ya que fue más real en cambio a otros desarrollados con anterioridad. Muy satisfecho con el proyecto terminado, con mi equipo de desarrollo y especialmente con el profe quien en cada clase inculcó en nosotros no solo su conocimiento sino ese afan de superacion tan caracteristico de el.

***Eloy Gonzalez:***

Este curso puso mis conocimientos a prueba y seguiré aplicando las tecnologías aprendidas para continuar con mis proyectos personales.