**DOCUMENTACIÓN DEL CODIGO FUENTE**

****

**TABLA DE CONTENIDO**

[1. INTRODUCCIÓN 3](#_Toc479695109)

[2. Código Html 4](#_Toc479695110)

[2.1 IndexGame.html 4](#_Toc479695111)

[3. Código JavaScript 4](#_Toc479695112)

[3.1 Phaser.js 4](#_Toc479695113)

[3.2 Game.js 4](#_Toc479695114)

[3.3 Historieta.js 5](#_Toc479695115)

[3.4 Navegación.js 5](#_Toc479695116)

[3.5 Creditos.js 11](#_Toc479695117)

[3.6 SelecciónPersonaje.js 13](#_Toc479695118)

[3.7 SelecciónAvatar.js 14](#_Toc479695119)

[3.8 Primer.js 16](#_Toc479695120)

[3.9 Fin.js 19](#_Toc479695121)

[3.10 InvitarAmigos.js 20](#_Toc479695122)

[3.11 PerfilJugador.js 21](#_Toc479695123)

[3.12 Logros.js 22](#_Toc479695124)

[4. Código Jsp 24](#_Toc479695125)

[4.1 Login.jsp 24](#_Toc479695126)

[4.2 Forgot.jsp 25](#_Toc479695127)

[4.3 Registro.jsp 26](#_Toc479695128)

[5. Código Css 26](#_Toc479695129)

[6. Código Java 26](#_Toc479695130)

# INTRODUCCIÓN

En este documento se encuentra la explicación y documentación del código fuente realizado por el grupo de Frontend.

En este documento se encontrará divido por código en Java, código en Html, código en JavaScript y código en Css.

# Código Html

## IndexGame.html

* Se coloca “Selección avatar y personaje” como nombre para la pestaña **Línea 5.**
* Se enlaza **phaser.js** (Librería de phaser) para poder utilizarla **Línea 6.**
* Se enlazan los scripts de las diferentes pantallas del juego **seleccionavatar.js, historieta.js, créditos.js, selecciónpersonaje.js, navegación.js, primer.js, fin.js** y **game.js Líneas 7-15.**
* Se enlaza **phaser-min.js** (Librería de phaser) para poder utilizarla.
* Se enlaza desde Google Fonts un tipo de letra, específicamente la letra “Roboto” **Línea 16.**
* Se enlaza un documento de estilos llamado **styles.css**, en el cual estarán los estilos que se usarán **Línea 17**

# Código JavaScript

## Phaser.js

* Este archivo contiene toda la librería de **PHASER** que será utilizada para desarrollar el juego.

## Game.js

* Se declara una variable llamada **game** con la cual se utilizará la librería de **phaser**, y con ella se crea una ventana de tamaño 800 (ancho) x 600 (alto) **Línea 1.**
* A la variable **game** se le crean 9 estados, las cuales son **historieta, navegación, primer, créditos, invitarAmigos, perfilJugador, seleccionpersonaje, seleccionavatar** y **fin**, estos corresponden a las diferentes “pantallas” a las que se accedera durante el juego **Líneas 2-8.**
* Se inicia el estado llamado “seleccionavatar” **Línea 9.**

## Historieta.js

* Se crean dos variables llamadas **histo y button**, los cuales serán utilizados para la sección de la historieta **Líneas 1-2.**
* Se centra el estado **historieta** horizontal y verticalmente **Líneas 6-7.**
* Se carga el sprite de la historieta y el del botón con la función **load.spritesheet()**, la cual recibe como parámetros, una cadena con la cual se identificara lo cargado, una cadena que contiene la ruta del sprite a cargar, el ancho de cada frame y el alto de cada frame **Líneas 9-10.**
* Se agrega el sprite de la historieta a la ventana con la función **add.sprite()**, la cual recibe como parámetros, la posición en X, la posición en Y en la cual se ubicará y la cadena con la cual se identificó el sprite que se agregará **Línea 13.**
* Se agrega una animación al sprite que se acaba de agregar con la función **animations.add()**, la cual recibe como parámetros, una cadena que identificará esta animación, los números de frames a agregar a la animación (encerrados en corchetes cuadrados y separados por comas), la velocidad con la cual se cambiara de frame, y un booleano que representará la repetición de esta animación **Línea 14.**
* Se inicia la animación recién agregada con la función **animations.play()**, la cual recibe como parámetro, una cadena con la cual se identificó la animación **Línea 15.**
* Se agrega un botón con la función **add.button()**, la cual recibe como parámetros, la posición en X, la posición en Y, una cadena con la cual se identificó el sprite cargado anteriormente, la función que realizara al ser presionado, el contexto en el que se usara (por lo general es this), el número de frame del estado **Over** (cuando el mouse este sobre el), el número de frame del estado **Out** (cuando no hay ninguna interacción con el botón), el número de frame del estado **Down** (cuando se presione el botón), el número de frame del estado **Up** (cuando se deje de presionar el botón) **Línea 16.**
* Se agrega la función **verMapa** que se ejecutará cuando el botón recién agregado sea presionado, la cual iniciará otro estado del juego llamado **navegación** **Líneas 19-21.**

## Navegación.js

* Se crean 12 variables llamadas **popup, popup2, popup3, popup4, popup5, popup6, popup7, popup8, popup9, popup10, popup11 y popup12**, las cuales representaran los popups (una nueva ventana que aparece) usados en la sección de navegación **Líneas 3-4.**
* Se crean 12 variables llamadas **closeButton**, **closeButton2, closeButton3, closeButton4, closeButton5, closeButton6, closeButton7, closeButton8, closeButton9, closeButton10, closeButton11 y closeButton12** las cuales representaran los botones para cerrar cada popup **Líneas 7-8.**
* Se crea una variable llamada **tween**, la cual se usará para modificar el tamaño de los popups **Línea 9.**
* Se crean 12 variables llamadas **btMundo, btMundo2, btMundo3, btMundo4, btMundo5, btMundo6, btMundo7, btMundo8, btMundo9, btMundo10, btMundo11** y **btMundo12** las cuales representaran los botones para abrir cada uno de los popups **Líneas 11-12.**
* Se carga el sprite del botón de pausa con la función **load.spritesheet()**, la cual recibe como parámetros, una cadena que contiene la ruta del sprite a cargar, el ancho de cada frame y el alto de cada frame **Línea 17.**
* Se cargan las imágenes que se usaran en la sección de navegación, las cuales son el mapa de selección de nivel y los 12 escenarios de los mundos con la función **load.image()**, la cual recibe como parámetros, una cadena con la cual se identificara lo cargado, y una cadena que contiene la ruta de la imagen a cargar **Líneas 21-33.**
* Se cargan los sprites de los botones de los subniveles y de los créditos con la función l**oad.spritesheet()**, la cual recibe como parámetros, una cadena con la cual se identificara lo cargado, una cadena que contiene la ruta del sprite a cargar, el ancho de cada frame y el alto de cada frame **Líneas 35-44.**
* Se carga la imagen que se usara para representar el botón de cerrar los con la función **load.image()**, la cual recibe como parámetros, una cadena con la cual se identificara lo cargado, y una cadena que contiene la ruta de la imagen a cargar **Línea 45.**
* Se agrega el sprite del mapa con la función **add.sprite()**, la cual recibe como parámetros, la posición en X, la posición en Y en la cual se ubicará y la cadena con la cual se identificó el sprite que se agregará **Línea 51.**
* Se agrega el botón que representara la sección de créditos con la función **add.button()**, la cual recibe como parámetros, la posición en X, la posición en Y, una cadena con la cual se identificó el sprite cargado anteriormente, la función que realizara al ser presionado, el contexto en el que se usara (por lo general es this), el número de frame del estado **Over** (cuando el mouse este sobre el), el número de frame del estado **Out** (cuando no hay ninguna interacción con el botón), el número de frame del estado **Down** (cuando se presione el botón), el número de frame del estado **Up** (cuando se deje de presionar el botón) **Línea 52.**
* Se agregan 12 botones que abrirán los popups de cada uno de los 12 mundos con la función **add.button()**, la cual recibe como parámetros, la posición en X, la posición en Y, una cadena con la cual se identificó el sprite cargado anteriormente, la función que realizara al ser presionado, el contexto en el que se usara (por lo general es this), el número de frame del estado **Over** (cuando el mouse este sobre el), el número de frame del estado **Out** (cuando no hay ninguna interacción con el botón), el número de frame del estado **Down** (cuando se presione el botón), el número de frame del estado **Up** (cuando se deje de presionar el botón), también se les reajusta su tamaño con la función **scale.setTo()**, la cual recibe como parámetros, la escala en número de forma horizontal y la escala en número de forma vertical (el valor 1 representa el tamaño normal). Además, se activa el uso de los cursores para poder desplazarse en la ventana **Líneas 54-90.**
* Se agrega la función **verCreditos** que se ejecutará cuando el botón llamado **botónCreditos** se presione, la cual iniciará otro estado del juego llamado **créditos** **Líneas 96-98.**
* Se agrega la función **onMundo1** que se ejecutara cuando el botón llamado **btMundo** sea presionado **(Líneas 100-171).** Dentro de esta función se agrega el sprite del escenario de “Poli Medellín”, escalándolo al 70% de su tamaño original y se activa su interacción **(Líneas 103-106).** Se creandos variables llamadas **pw** y **ph** las cuales representaran la posición en la que se ubicara el botón encargado de cerrar el popup1 **(Líneas 109-110).** Se agrega un botón que se encargara de cerrar el popup1, se escala al 50% de su tamaño original y se agrega al popup **(Líneas 112-116).** Se agregan los 5 botones que representaran los subniveles de cada mundo, se les activa su interactividad dentro del popup, se les activa su uso con los cursores, se agrega cada subnivel al popup, se agrega una función llamada **iniciarnivel1** que se ejecutara cuando el botón del subnivel 1 sea presionado **(Líneas 119-163)**. Y por último se crea una función llamada **closeWindow** que se ejecutara cuando el botón de cerrar popup1 sea presionado **(Línea 165-171).**
* Se agrega la función **onMundo2** que se ejecutara cuando el botón llamado **btMund2** sea presionado **(Líneas 175-234).** Dentro de esta función se agrega el sprite del escenario de “Postgrados”, escalándolo al 70% de su tamaño original y se activa su interacción **(Líneas 177-179).** Se agrega un botón que se encargara de cerrar el popup2, se escala al 50% de su tamaño original y se agrega al popup2 **(Líneas 182-184).** Se agregan los 5 botones que representaran los subniveles de cada mundo, se les activa su interactividad dentro del popup, se les activa su uso con los cursores, se agrega cada subnivel al popup **(Líneas 186-226)**. Y por último se crea una función llamada **closeWindow** que se ejecutara cuando el botón de cerrar popup2 sea presionado **(Línea 228-230).**
* Se agrega la función **onMundo3** que se ejecutara cuando el botón llamado **btMund3** sea presionado **(Líneas 236-296).** Dentro de esta función se agrega el sprite del escenario de “Sede Excelencia”, escalándolo al 70% de su tamaño original y se activa su interacción **(Líneas 239-241).** Se agrega un botón que se encargara de cerrar el popup3, se escala al 50% de su tamaño original y se agrega al popup3 **(Líneas 245-247).** Se agregan los 5 botones que representaran los subniveles de cada mundo, se les activa su interactividad dentro del popup, se les activa su uso con los cursores, se agrega cada subnivel al popup **(Líneas 248-290)**. Y por último se crea una función llamada **closeWindow** que se ejecutara cuando el botón de cerrar popup3 sea presionado **(Línea 292-295).**
* Se agrega la función **onMundo4** que se ejecutara cuando el botón llamado **btMund4** sea presionado **(Líneas 297-356).** Dentro de esta función se agrega el sprite del escenario de “Paradero”, escalándolo al 70% de su tamaño original y se activa su interacción **(Líneas 300-302).** Se agrega un botón que se encargara de cerrar el popup4, se escala al 50% de su tamaño original y se agrega al popup4 **(Líneas 306-308).** Se agregan los 5 botones que representaran los subniveles de cada mundo, se les activa su interactividad dentro del popup, se les activa su uso con los cursores, se agrega cada subnivel al popup **(Líneas 310-350)**. Y por último se crea una función llamada **closeWindow** que se ejecutara cuando el botón de cerrar popup4 sea presionado **(Línea 352-355).**
* Se agrega la función **onMundo5** que se ejecutara cuando el botón llamado **btMund5** sea presionado **(Líneas 357-415).** Dentro de esta función se agrega el sprite del escenario de “Facultad de Mercadeo”, escalándolo al 70% de su tamaño original y se activa su interacción **(Líneas 360-362).** Se agrega un botón que se encargara de cerrar el popup5, se escala al 50% de su tamaño original y se agrega al popup5 **(Líneas 365-367).** Se agregan los 5 botones que representaran los subniveles de cada mundo, se les activa su interactividad dentro del popup, se les activa su uso con los cursores, se agrega cada subnivel al popup **(Líneas 369-409)**. Y por último se crea una función llamada **closeWindow** que se ejecutara cuando el botón de cerrar popup5 sea presionado **(Línea 411-414).**
* Se agrega la función **onMundo6** que se ejecutara cuando el botón llamado **btMund6** sea presionado **(Líneas 406-474).** Dentro de esta función se agrega el sprite del escenario de “Facultad de Ciencias Sociales”, escalándolo al 70% de su tamaño original y se activa su interacción **(Líneas 419-421).** Se agrega un botón que se encargara de cerrar el popup6, se escala al 50% de su tamaño original y se agrega al popup6 **(Líneas 424-426).** Se agregan los 5 botones que representaran los subniveles de cada mundo, se les activa su interactividad dentro del popup, se les activa su uso con los cursores, se agrega cada subnivel al popup **(Líneas 428-468)**. Y por último se crea una función llamada **closeWindow** que se ejecutara cuando el botón de cerrar popup6 sea presionado **(Línea 470-473).**
* Se agrega la función **onMundo7** que se ejecutara cuando el botón llamado **btMund7** sea presionado **(Líneas 475-533).** Dentro de esta función se agrega el sprite del escenario de “Bienestar Universitario”, escalándolo al 70% de su tamaño original y se activa su interacción **(Líneas 478-480).** Se agrega un botón que se encargara de cerrar el popup7, se escala al 50% de su tamaño original y se agrega al popup7 **(Líneas 483-485).** Se agregan los 5 botones que representaran los subniveles de cada mundo, se les activa su interactividad dentro del popup, se les activa su uso con los cursores, se agrega cada subnivel al popup **(Líneas 487-527)**. Y por último se crea una función llamada **closeWindow** que se ejecutara cuando el botón de cerrar popup7 sea presionado **(Línea 529-532).**
* Se agrega la función **onMundo8** que se ejecutara cuando el botón llamado **btMund8** sea presionado **(Líneas 534-593).** Dentro de esta función se agrega el sprite del escenario de “Plazoleta Principal”, escalándolo al 70% de su tamaño original y se activa su interacción **(Líneas 537-539).** Se agrega un botón que se encargara de cerrar el popup8, se escala al 50% de su tamaño original y se agrega al popup8 **(Líneas 543-545).** Se agregan los 5 botones que representaran los subniveles de cada mundo, se les activa su interactividad dentro del popup, se les activa su uso con los cursores, se agrega cada subnivel al popup **(Líneas 547-587)**. Y por último se crea una función llamada **closeWindow** que se ejecutara cuando el botón de cerrar popup8 sea presionado **(Línea 589-592).**
* Se agrega la función **onMundo9** que se ejecutara cuando el botón llamado **btMund9** sea presionado **(Líneas 594-653).** Dentro de esta función se agrega el sprite del escenario de “Facultad de Ciencias Administrativas”, escalándolo al 70% de su tamaño original y se activa su interacción **(Líneas 597-599).** Se agrega un botón que se encargara de cerrar el popup9, se escala al 50% de su tamaño original y se agrega al popup9 **(Líneas 603-605).** Se agregan los 5 botones que representaran los subniveles de cada mundo, se les activa su interactividad dentro del popup, se les activa su uso con los cursores, se agrega cada subnivel al popup **(Líneas 607-647)**. Y por último se crea una función llamada **closeWindow** que se ejecutara cuando el botón de cerrar popup9 sea presionado **(Línea 649-652).**
* Se agrega la función **onMundo10** que se ejecutara cuando el botón llamado **btMund10** sea presionado **(Líneas 654-713).** Dentro de esta función se agrega el sprite del escenario de “Cancha”, escalándolo al 70% de su tamaño original y se activa su interacción **(Líneas 657-659).** Se agrega un botón que se encargara de cerrar el popup10, se escala al 50% de su tamaño original y se agrega al popup10 **(Líneas 663-665).** Se agregan los 5 botones que representaran los subniveles de cada mundo, se les activa su interactividad dentro del popup, se les activa su uso con los cursores, se agrega cada subnivel al popup **(Líneas 667-707)**. Y por último se crea una función llamada **closeWindow** que se ejecutara cuando el botón de cerrar popup10 sea presionado **(Línea 709-712).**
* Se agrega la función **onMundo11** que se ejecutara cuando el botón llamado **btMund11** sea presionado **(Líneas 714-773).** Dentro de esta función se agrega el sprite del escenario de “Facultad de Ingeniería”, escalándolo al 70% de su tamaño original y se activa su interacción **(Líneas 717-719).** Se agrega un botón que se encargara de cerrar el popup11, se escala al 50% de su tamaño original y se agrega al popup11 **(Líneas 723-725).** Se agregan los 5 botones que representaran los subniveles de cada mundo, se les activa su interactividad dentro del popup, se les activa su uso con los cursores, se agrega cada subnivel al popup **(Líneas 727-767)**. Y por último se crea una función llamada **closeWindow** que se ejecutara cuando el botón de cerrar popup11 sea presionado **Línea 769-772.**
* Se agrega la función **onMundo12** que se ejecutara cuando el botón llamado **btMund12** sea presionado **(Líneas 774-833).** Dentro de esta función se agrega el sprite del escenario de “Bloque I”, escalándolo al 70% de su tamaño original y se activa su interacción **(Líneas 777-779).** Se agrega un botón que se encargara de cerrar el popup12, se escala al 50% de su tamaño original y se agrega al popup12 **(Líneas 782-784).** Se agregan los 5 botones que representaran los subniveles de cada mundo, se les activa su interactividad dentro del popup, se les activa su uso con los cursores, se agrega cada subnivel al popup **(Líneas 786-826)**. Y por último se crea una función llamada **closeWindow** que se ejecutara cuando el botón de cerrar popup12 sea presionado **(Línea 828-832).**

## Creditos.js

* Se declaran los botones **botonVolver, botonPacto, botonBuenos, botonMalos, botonPlagios** que se usaran para la sección de créditos **Línea 2.**
* Se declara e inicializa una cadena que tendrá la descripción del pacto de honor **Línea 4.**
* Se declara e inicializa un arreglo de cadenas llamado **descripcionesBuenos** que tendrá las descripciones de cada uno de los personajes buenos **Línea 6.**
* Se declara e inicializa un arreglo de cadenas llamado **descripcionesMalos** que tendrá las descripciones de cada uno de los personajes malos **Línea 16.**
* Se declara e inicializa un arreglo de cadenas llamado **plagios** que tendrá las descripciones de cada uno de los 10 tipos de plagio y su frecuencia con la que ocurren **Línea 29.**
* Se declaran las variables **logoPacto, textoPacto, textoBuenos, textoMalos, textoPlagios** que se usaran para representar los textos de la sección de créditos y el logo **Línea 40.**
* Se centra la ventana horizontal y verticalmente **Líneas 45-46.**
* Se carga la imagen del logo del pacto de honor con la función **load.image()**, la cual recibe como parámetros, una cadena con la cual se identificara lo cargado, y una cadena que contiene la ruta de la imagen a cargar **Línea 48.**
* Se cargan los sprites de los botones que tendrá la sección de créditos, los botones para abrir la descripción de los personajes buenos, los botones para abrir la descripción de los personajes malos y los botones para abrir la descripción de los tipos de plagio con la función l**oad.spritesheet()**, la cual recibe como parámetros, una cadena con la cual se identificara lo cargado, una cadena que contiene la ruta del sprite a cargar, el ancho de cada frame y el alto de cada frame **Líneas 50-88.**
* Se establece un color de fondo para la ventana de phaser el cual es #0060b2 **Línea 92.**
* Se agregan los botones **botonVolver, botonPacto, botonBuenos, botonMalos y botónPlagios** que corresponden a los botones que tendrá la sección de créditos, los botones para abrir la descripción de los personajes buenos, los botones para abrir la descripción de los personajes malos y los botones para abrir la descripción de los tipos de plagio con la función **add.button()**, la cual recibe como parámetros, la posición en X, la posición en Y, una cadena con la cual se identificó el sprite cargado anteriormente, la función que realizara al ser presionado, el contexto en el que se usara (por lo general es this), el número de frame del estado **Over** (cuando el mouse este sobre el), el número de frame del estado **Out** (cuando no hay ninguna interacción con el botón), el número de frame del estado **Down** (cuando se presione el botón), el número de frame del estado **Up** (cuando se deje de presionar el botón) **Líneas 96-137.**
* Se ocultan todos los botones previamente agregados **Líneas 140-174.**
* Se agrega una variable **textoPacto** el cual se utilizara para representar la descripción del pacto de honor, se agrega un texto con la función **add.text()**, la cual recibe como parámetros, la posición en x, la posición en y, el texto a poner, y los estilos que se aplicaran al texto y se oculta **Líneas 177-178.**
* Se agrega una variable **textoBuenos** el cual se utilizara para representar la descripción de cada uno de los personajes buenos, se agrega un texto con la función **add.text()**, la cual recibe como parámetros, la posición en x, la posición en y, el texto a poner, y los estilos que se aplicaran al texto **Línea 181.**
* Se agrega una variable **textoMalos** el cual se utilizara para representar la descripción de cada uno de los personajes malos, se agrega un texto con la función **add.text()**, la cual recibe como parámetros, la posición en x, la posición en y, el texto a poner, y los estilos que se aplicaran al texto **Línea 182.**
* Se agrega una variable **textoPlagios** el cual se utilizara para representar la descripción de cada uno de los tipos de plagio, se agrega un texto con la función **add.text()**, la cual recibe como parámetros, la posición en x, la posición en y, el texto a poner, y los estilos que se aplicaran al texto **Línea 183.**
* Se agrega el logo del pacto de honor con la función **add.image()**, la cual recibe como parámetros la posición en X, la posición en Y y una cadena con la cual se identificara y se oculta **Líneas 186-187.**
* Se agregan 31 funciones las cuales serán ejecutadas cuando sea presionado el botón de algún personaje bueno, o algún personaje malo o algún tipo de plagio, y dentro de cada una de estas se modifica el texto de la variable **textoBuenos, textoMalos o textoPlagios** **Líneas 192-222.**
* Se agrega una función llamada **verPacto** que será ejecutada cuando sea presionado el botón llamado **botonPacto**, dentro de esta función se pone visible el texto de la variable **textoPacto** y la imagen de la variable **logoPacto** **Líneas 227-265.**
* Se agrega una función llamada **verPersonajesBuenos** que será ejecutada cuando sea presionado el botón llamado **botonBuenos**, dentro de esta función se ponen visibles los botones de cada uno de los personajes buenos y la variable llamada **textoBuenos** **Líneas 268-307.**
* Se agrega una función llamada **verPersonajesMalos** que será ejecutada cuando sea presionado el botón llamado **botonMalos**, dentro de esta función se ponen visibles los botones de cada uno de los personajes malos y la variable llamada **textoMalos** **Líneas 310-349.**
* Se agrega una función llamada **verPlagios** que será ejecutada cuando sea presionado el botón llamado **botonPlagios**, dentro de esta función se ponen visibles los botones de cada uno de los tipos de plagio y la variable llamada **textoPlagios** **Líneas 352-391.**
* Se crea una función llamada **volver** la cual se ejecutará cuando el botón llamado **botonVolver** sea presionado, dentro de esta función se iniciará el estado navegación **Líneas 394-396.**

## SelecciónPersonaje.js

* Se declaran las variables a usar en el código, como lo son el número de columnas y filas que tendrá la selección de personajes, el ancho y el alto de cada botón, el espacio entre botones y un arreglo de cadenas que contendrá los nombres de los personajes buenos **Líneas 1-6.**
* Se centra horizontalmente y verticalmente la ventana **Líneas 13-14.**
* Se cargan sprites de los botones de personajes y del botón de iniciar batalla **Líneas 16-25.**
* Se modifica el backgroundColor (Color de fondo) de la ventana por el color #2451A6 (Color azul) **Línea 28.**
* Se agrega un texto a la ventana, el cual es “Selección de personaje”, en una posición y con unos estilos específicos **Línea 29.**
* Se definen unas márgenes para ubicar los botones **Líneas 32-35.**
* Mediante 3 ciclos se agregan a la ventana los botones de los personajes con la función **add.button()**, la cual recibe como parámetros, la posición en X, la posición en Y, una cadena con la cual se identificó el sprite cargado anteriormente, la función que realizara al ser presionado, el contexto en el que se usara (por lo general es this), el número de frame del estado **Over** (cuando el mouse este sobre el), el número de frame del estado **Out** (cuando no hay ninguna interacción con el botón), el número de frame del estado **Down** (cuando se presione el botón), el número de frame del estado **Up** (cuando se deje de presionar el botón) **Líneas 38-47.**
* Se agrega el botón llamado **startButton** con la función **add.button()**, la cual recibe como parámetros, la posición en X, la posición en Y, una cadena con la cual se identificó el sprite cargado anteriormente, la función que realizara al ser presionado, el contexto en el que se usara (por lo general es this), el número de frame del estado **Over** (cuando el mouse este sobre el), el número de frame del estado **Out** (cuando no hay ninguna interacción con el botón), el número de frame del estado **Down** (cuando se presione el botón), el número de frame del estado **Up** (cuando se deje de presionar el botón) y se centra en las posiciones dadas **Líneas 48-49.**
* Se crea una función llamada **verbatalla** que se ejecutará cuando el botón llamado **startButton** sea presionado, dentro de esta función se inicia el estado llamado **primer.**

## SelecciónAvatar.js

* Se definen las variables a usar para la selección de avatar, las cuales corresponden a la velocidad en que se moverá dentro de un scroll horizontal, los nombres de los 9 personajes buenos, las descripciones de cada uno de los 9 personajes buenos, una variable que representará las descripciones de los personajes en cada momento del scroll, una variable que representará los nombres de los personajes en cada momento del scroll y un botón que al ser presionado accederá a la sección de la historieta **Líneas 1-16.**
* Se centra la ventana de horizontal y verticalmente **Líneas 21-22.**
* Se carga una imagen transparente que estará debajo de las imágenes en el scroll **Línea 24.**
* Se cargan las imágenes de los 9 personajes buenos que estarán en un scroll (Agregado más adelante) con la función **add.load.image()**, la cual recibe como parámetros, una cadena con la cual se identificara en el código y la ruta en donde se encuentra la imagen **Líneas 25-33.**
* Se carga un sprite de un botón con 3 frames (cada uno representa un estado del botón) con la función **load.spritesheet()**, la cual recibe como parámetros una cadena que con la cual se identificara al sprite en el código, la ruta del sprite, el ancho de cada frame y el alto de cada frame **Línea 34.**
* Se modifica el **backgroundColor** (Color de fondo) de la ventana por el color #2451A6 (Color azul) **Línea 37.**
* Se agrega un texto a la ventana, el cual es “Selección de avatar”, en una posición y con unos estilos específicos con la función **add.text()**, la cual recibe como parámetros, la posición en x, la posición en y, el texto a poner, y los estilos que se aplicaran al texto **Línea 38.**
* Se agrega un ScrollingMap y se le define que la imagen transparente (previamente cargada) tendrá como ancho, el ancho del scroll y como alto, el alto del scroll **Línea 39.**
* Se agregan las funciones del scroll, las cuales son, que este activado, que se guarde la posición en la que se está dentro del scroll, una velocidad inicial cuando no esté en movimiento, que no haya movimiento vertical, y se le agrega un Rectangle (Limites) **Líneas 39-46.**
* Se agrega imagen por imagen de los personajes mediante un ciclo, el cual agrega la imagen a la ventana con la función **add.image()**, la cual recibe como parámetros la posición en X, la posición en Y y una cadena con la cual se identificara, luego se centra a esa posición, y por último se le agrega al scroll **Líneas 48-52.**
* Cuando el scroll este en movimiento, una variable predefinida como **isBeingDragged** estará en true y abra una velocidad inicial en el scroll, en el caso contrario (En el que el scroll no esté en movimiento), la variable **isBeingDragged** estará en false **Líneas 54-60.**
* Se inicializa una variable llamada **apodo** que agregara un texto a la ventana con la función **add.text()**, la cual recibe como parámetros, la posición en x, la posición en y, el texto a poner, y los estilos que se aplicaran al texto, también se centra en la posiciones dadas. Esta variable se usará para representar el nombre de cada personaje **Líneas 62-63.**
* Se inicializa una variable llamada **description** que agregara un texto con la función **add.text()**, la cual recibe como parámetros, la posición en x, la posición en y, el texto a poner, y los estilos que se aplicaran al texto, también se centra en la posiciones dadas. Esta variable se usará para representar la descripción de personalidad de cada personaje **Líneas 64-65.**
* Se agrega un botón a la ventana con la función add.button(), la cual recibe como parámetros, la posición en X, la posición en Y, una cadena con la cual se identificó el sprite cargado anteriormente, la función que realizara al ser presionado, el contexto en el que se usara (por lo general es this), el número de frame del estado Over (cuando el mouse este sobre el), el número de frame del estado Out (cuando no hay ninguna interacción con el botón), el número de frame del estado Down (cuando se presione el botón), el número de frame del estado Up (cuando se deje de presionar el botón). Este botón se usará para abrir el estado de la historieta **Líneas 66-67.**
* Se crea una función llamada **verH**, la cual será ejecutada cuando el botón llamado **startButton** sea presionado, dentro de esta función se inicia el estado llamado **historieta** **Líneas 70-72.**
* Se declara una variable llamada **zoomed** que representara cuando un elemento del scroll este seleccionada **Línea 74.**
* Mediante dos ciclos aninados, se comprobará cuando una imagen este seleccionada, el primero se encargará de agrandar al 150% la imagen, cuando esta esté seleccionada y pondrá la variable **zoomed** en true, y el segundo, modificara el nombre y la descripción de personalidad cada personaje, dependiendo de cuál este seleccionado **Líneas 75-86.**
* Cuando una imagen ya no está seleccionada, esta cambiara su tamaño al normal **Líneas 87-89.**
* Cuando haya movimiento dentro del scroll, se guardará la posición en una variable predefinida como **savedPosition Líneas 91-93.**
* Se define con que velocidad se moverá dentro del scroll, dependiendo de donde este la posición en x **Líneas 94-120.**

## Primer.js

* Se declaran las variables que se usaran para la realización de la sección de la primera pelea, las cuales son **gallo1**, **oso1, cursores, text, counter, ti, x, y, atqueH1** (Ataque del héroe)**, ataqueV1** (Ataque del villano), **atqueH2** (Ataque especial del héroe), **ataqueV2** (Ataque especial del villano), **ataqueV3** (Ataque tipo plagio del villano), **danoH** (Daño del héroe), **danoV** (Daño del villano), **movH, movV, ataquePlagio, ataquePersonalidadV** y **ataquePersonalidadB Líneas 1-27.**
* Se carga el escenario que se usará de fondo para la batalla con la función **load.image()**, la cual recibe como parámetros, una cadena con la cual se identificara lo cargado, y una cadena que contiene la ruta de la imagen a cargar **Línea 33.**
* Se cargan los sprites del personaje “Felipe Oso” y “Andrés Gallo” con la función **load.spritesheet()**, la cual recibe como parámetros, una cadena con la cual se identificara lo cargado, una cadena que contiene la ruta del sprite a cargar, el ancho de cada frame y el alto de cada frame **Líneas 34-35.**
* Se cargan las imágenes de la cara del “Felipe Oso”, la cara de “Andrés Gallo” y la imagen de pausa con la función **load.image()**, la cual recibe como parámetros, una cadena con la cual se identificara lo cargado, y una cadena que contiene la ruta de la imagen a cargar **Líneas 36-39.**
* Se carga el sprite del poder llamado “Error 404” con la función **load.spritesheet()**, la cual recibe como parámetros, una cadena con la cual se identificara lo cargado, una cadena que contiene la ruta del sprite a cargar, el ancho de cada frame y el alto de cada frame **Línea 40.**
* Se cargan los logos del poder de personalidad de “Felipe Oso” y de “Andrés Gallo” con la función **load.image()**, la cual recibe como parámetros, una cadena con la cual se identificara lo cargado, y una cadena que contiene la ruta de la imagen a cargar **Líneas 41-42.**
* Se agrega físicas al juego (Permiten que haya colisiones y permite establecer límites) **Línea 48.**
* Se agrega el escenario cargado como fondo de la batalla, los sprites de los personaje “Andrés Gallo” y “Felipe Oso” con la función **add.sprite()**, la cual recibe como parámetros, la posición en X, la posición en Y en la cual se ubicará y la cadena con la cual se identificó el sprite que se agregará. También se ajusta el tamaño de “Andrés Gallo” y centra en las posiciones indicadas a los sprites de los dos personajes **Líneas 53-59.**
* Se agregan diferentes animaciones a los personajes “Andrés Gallo” y “Felipe Oso”, las cuales son **quieto, correr, punos, defensa, especial, quieto** con la función **animations.add()**, la cual recibe como parámetros, una cadena que identificará esta animación, los números de frames a agregar a la animación (encerrados en corchetes cuadrados y separados por comas), la velocidad con la cual se cambiara de frame, y un booleano que representará la repetición de esta animación **Líneas 62-66 y 71-75**. También se inicia la animación llamada **quieto** de ambos personajes **Líneas 67 y 76**, además se activan las físicas en estos dos personajes **Líneas 68 y 79** y se limita hasta donde pueden moverse (Los límites son el tamaño de la ventana del juego (800x600)) **Líneas 69 y 80.**
* Se activa el uso de los cursores (se usarán para mover a Andrés Gallo con estos), además se agrega la tecla “Espacio” para ser utilizada también **Líneas 82-83.**
* Se agrega la cara de “Felipe Oso” y la de “Andrés Gallo” a la ventana, además se ajusta su tamaño al 60% de su tamaño original con la función **add.sprite()**, la cual recibe como parámetros, la posición en X, la posición en Y en la cual se ubicará y la cadena con la cual se identificó el sprite que se agregará **Líneas 89-92.**
* Se agrega un botón que se utilizara para pausar el juego con la función **add.button()**, la cual recibe como parámetros, la posición en X, la posición en Y, una cadena con la cual se identificó el sprite cargado anteriormente, la función que realizara al ser presionado, el contexto en el que se usara (por lo general es this), el número de frame del estado **Over** (cuando el mouse este sobre el), el número de frame del estado **Out** (cuando no hay ninguna interacción con el botón), el número de frame del estado **Down** (cuando se presione el botón), el número de frame del estado **Up** (cuando se deje de presionar el botón) **Línea 93.**
* Se agrega una función que se ejecutara cuando el botón de pausa sea soltado con la función **events.onInputUp.add()**, la cual recibe como parámetro una función a realizar, dentro de esta función se agrega una imagen de tamaño 270x170 que representará la ventana que aparecerá cuando el juego este en pausa y se ajusta su tamaño al 150% de su tamaño original **Líneas 96-102.**
* Cuando se presione dentro del juego se realizará una función llamada **unpause,** la cual pausara el juego, removerá el escenario de fondo, pondrá el contador del tiempo en cero, y se iniciara la sección **primer** **Líneas 104-116.**
* Se ajusta el tamaño del botón de pausa al 40% de su tamaño original **Línea 118.**
* Se agrega un texto a la ventana el cual representará el tiempo transcurrido con la función **add.text()**, la cual recibe como parámetros, la posición en x, la posición en y, el texto a poner, y los estilos que se aplicaran al texto **Líneas 119-122.**
* Se instancian objetos de tipo Rectangle los cuales representaran el piso, las barras de cada personaje con el objeto con la función **new Phaser.Rectangle()**, la cual recibe como parámetros, la posición en x, la posición en y, el ancho y el alto **Líneas 124-141.**
* Se agrega una función llamada **render** la cual es utilizada para darle colores a los rectángulos previamente agregados **Líneas 150-167.**
* Se agrega una función llamada **actualizarBarra** la cual recibe como parámetros la barra de vida y el daño causado, dentro de esta función se disminuye la barra de vida de acuerdo al daño recibido, y cuando la barra este en cero, se inicia un estado llamado **fin** con la función **state.start()**, la cual recibe como parámetro el nombre del estado que se iniciará **Líneas 169-176.**
* Se agrega una función llamada **actualizarEspecial,** dentro de esta función se agrega el sprite del ataque tipo plagio de “Andrés Gallo” con la función **add.sprite()**, la cual recibe como parámetros, la posición en X, la posición en Y en la cual se ubicará y la cadena con la cual se identificó el sprite que se agregará **Línea 178.** También se le activan las físicas al poder y se establecen unos límites **Líneas 179-180**. Se agrega una animación llamada **especiall** y se inicia **Líneas 181-182** y después de que se visualice el poder se destruye con la función **destroy()**, esta no recibe parámetros **Línea 183.**
* Se agrega una función llamada **destruirSprite**, la cual recibe como parámetro el sprite a destruir, y dentro de esta se destruye el sprite con la función **destroy()** **Líneas 185-187.**
* Se agrega una función llamada **colision Líneas 193-215**, dentro de esta se desplaza el personaje “Andrés Gallo” 100 pixeles a la izquierda **Línea 194**, se inicia la animación llamada **correr** de “Andrés Gallo” **Línea 195**, se desplaza el personaje “Felipe Oso” 200 pixeles a la derecha **Línea 196**, se cambia el arreglo de posiciones de cada personaje según haya sido su movimiento **Líneas 198-213.**
* Se especifica que sucederá cada vez que se utilice algún curso o la tecla “Espacio”, cuando el cursor de la derecha este presionado el personaje de “Andrés Gallo” se moverá 2 pixeles a la derecha y se iniciara el estado llamado **correr** de este, cuando el cursor de la izquierda este presionado el personaje de “Andrés Gallo” se moverá 2 pixeles a la izquierda y se iniciara el estado llamado **correr** de este, cuando el cursor de abajo este presionado se iniciara el estado llamado **defensa** de este, cuando el cursor de arriba este presionado se iniciara el estado llamado **especial** de “Andrés Gallo”, cuando la tecla de espacio este presionada se iniciara el estado llamado **correr** de “Andrés Gallo” y cuando cualquier otra tecla sea presionada se iniciara el estado llamado **quieto** de “Andrés Gallo” **Líneas 218-239.**
* Se implementa los movimientos de la computadora “Felipe Oso”, iniciando algunos de sus estados dependiendo el valor de vida que tenga **Líneas 242-269.**
* Se definen los tiempos en los que la batalla se acabara y cuando se inicie el estado llamado **fin** **Líneas 274-283.**
* Se agregan las colisiones de “Andrés Gallo” y de “Felipe Oso” con la función **physics.arcade.collide()** **Líneas 285-287.**

## Fin.js

* Se recibe quien fue el ganador de la batalla **Línea 1.**
* Se carga la imagen de un escenario del poli que se usara como fondo cuando haya un ganador en la batalla con la función **load.image()**, la cual recibe como parámetros, una cadena con la cual se identificara lo cargado, y una cadena que contiene la ruta de la imagen a cargar **Línea 5.**
* Se carga el sprite de “Andrés Gallo” con la función **load.spritesheet()**, la cual recibe como parámetros, una cadena con la cual se identificara lo cargado, una cadena que contiene la ruta del sprite a cargar, el ancho de cada frame y el alto de cada frame **Línea 6.**
* Se agrega la imagen del escenario que se utilizara cuando haya un ganador en la batalla a la ventana con la función **add.sprite()**, la cual recibe como parámetros, la posición en X, la posición en Y en la cual se ubicará y la cadena con la cual se identificó el sprite que se agregará **Línea 16.**
* Se agrega el sprite del personaje “Andrés Gallo” con la función **add.sprite()**, la cual recibe como parámetros, la posición en X, la posición en Y en la cual se ubicará y la cadena con la cual se identificó el sprite que se agregará **Línea 17**, seajusta el tamañoal 150% de su tamaño original **Línea 18** y se centra en las posiciones indicadas **Línea 19.**
* Se agregan dos textos a la ventana, uno será “Ganador” y el otro será “Personaje Malo” o “Personaje Bueno” con la función **add.text()**, la cual recibe como parámetros, la posición en x, la posición en y, el texto a poner, y los estilos que se aplicaran al texto **Líneas 21-22.**

## InvitarAmigos.js

* Se crea un botón llamado **botonVolver**, el cual será utilizado en la sección **invitarAmigos** **Línea 1.**
* Se centra la ventana del juego horizontal y verticalmente **Líneas 6-7.**
* Se carga un sprite que se utilizara para representar el botón llamado **botonVolver** con la función **load.spritesheet()**, la cual recibe como parámetros, una cadena con la cual se identificara lo cargado, una cadena que contiene la ruta del sprite a cargar, el ancho de cada frame y el alto de cada frame **Línea 10.**
* Se modifica el backgroundColor (Color de fondo) de la ventana por el color #0060b2 **Línea 14.**
* Se agrega un texto a la ventana, el cual es “Invitar Amigos” con la función **add.text()**, la cual recibe como parámetros, la posición en x, la posición en y, el texto a poner, y los estilos que se aplicaran al texto **Línea 15.**
* Se agrega un botón llamado **botonVolver** a la ventana, el cual se usará para regresar a la sección de **navegación** con la función **add.button()**, la cual recibe como parámetros, la posición en X, la posición en Y, una cadena con la cual se identificó el sprite cargado anteriormente, la función que realizara al ser presionado, el contexto en el que se usara (por lo general es this), el número de frame del estado **Over** (cuando el mouse este sobre el), el número de frame del estado **Out** (cuando no hay ninguna interacción con el botón), el número de frame del estado **Down** (cuando se presione el botón), el número de frame del estado **Up** (cuando se deje de presionar el botón) **Línea 18.**
* Se agrega una función llamada **volver**, que se ejecutara cuando el botón **botonVolver** sea presionado, dentro de esta función se inicia el estado llamado **navegación Líneas 21-23.**

## PerfilJugador.js

* Se centra la ventana del juego horizontal y verticalmente **Líneas 9-10.**
* Se carga la imagen de avatar del jugador con la función **load.image()**, la cual recibe como parámetros, una cadena con la cual se identificara lo cargado, y una cadena que contiene la ruta de la imagen a cargar **Línea 11.**
* Se cargan los sprites de los botones que se usaran en la sección de **perfilJugador** con la función **load.spritesheet()**, la cual recibe como parámetros, una cadena con la cual se identificara lo cargado, una cadena que contiene la ruta del sprite a cargar, el ancho de cada frame y el alto de cada frame **Líneas 15-19.**
* Se modifica el backgroundColor (Color de fondo) de la ventana por el color #2451A6 **Línea 24.**
* Se agrega a la ventana el sprite de la imagen de avatar con la función **add.sprite()**, la cual recibe como parámetros, la posición en X, la posición en Y en la cual se ubicará y la cadena con la cual se identificó el sprite que se agregará **Línea 25.**
* Se agregan botones a la ventana, los cuales son **botonVolver, boton-personaje, boton-jefe, boton-trofe y boton-alfanumerico** con la función **add.button()**, la cual recibe como parámetros, la posición en X, la posición en Y, una cadena con la cual se identificó el sprite cargado anteriormente, la función que realizara al ser presionado, el contexto en el que se usara (por lo general es this), el número de frame del estado **Over** (cuando el mouse este sobre el), el número de frame del estado **Out** (cuando no hay ninguna interacción con el botón), el número de frame del estado **Down** (cuando se presione el botón), el número de frame del estado **Up** (cuando se deje de presionar el botón) **Líneas 27-31.**
* Se agregan diferentes textos a la ventana, los cuales son “Perfil del usuario”, “Nickname:”, “Mundo:” y “Nivel:” con la función **add.text()**, la cual recibe como parámetros, la posición en x, la posición en y, el texto a poner, y los estilos que se aplicaran al texto y se oculta **Líneas 33-36.**
* Se agrega una función llamada **verNavegacion**, la cual será ejecutada al presionar el botón llamado **botonVolver,** dentro de esta función se inicia el estado llamado **navegación** **Líneas 39-41.**

## Logros.js

* Se definen las variables a usar para la selección de avatar, las cuales corresponden a la velocidad en que se moverá dentro de un scroll horizontal, los nombres de los 10 logros, una arreglo que representara si un logro está bloqueado o desbloqueado una variable que representará las descripciones de los personajes en cada momento del scroll, una variable que representará los nombres de los logros en cada momento del scroll, un botón que al ser presionado accederá a la sección **perfilJugador** y un arreglo que tendrá las descripciones de cada uno de los 10 logros **Líneas 3-22.**
* Se centra la ventana horizontal y verticalmente **Líneas 27-28.**
* Se carga una imagen transparente que estará debajo de las imágenes en el scroll **Línea 31.**
* Se cargan las imágenes de los 10 tipos de plagio que estarán en un scroll (Agregado más adelante) con la función **add.load.image()**, la cual recibe como parámetros, una cadena con la cual se identificara en el código y la ruta en donde se encuentra la imagen **Líneas 33-42.**
* Se carga un sprite del botón **botonVolver** con la función **load.spritesheet()**, la cual recibe como parámetros una cadena que con la cual se identificara al sprite en el código, la ruta del sprite, el ancho de cada frame y el alto de cada frame **Línea 43.**
* Se carga un sprite de un botón con 3 frames (cada uno representa un estado del botón) con la función **load.spritesheet()**, la cual recibe como parámetros una cadena que con la cual se identificara al sprite en el código, la ruta del sprite, el ancho de cada frame y el alto de cada frame **Línea 46.**
* Se modifica el **backgroundColor** (Color de fondo) de la ventana por el color #2451A6 (Color azul) **Línea 51.**
* Se agrega un texto a la ventana, el cual es “Logros”, en una posición y con unos estilos específicos con la función **add.text()**, la cual recibe como parámetros, la posición en x, la posición en y, el texto a poner, y los estilos que se aplicaran al texto **Línea 55.**
* Se agrega un ScrollingMap y se le define que la imagen transparente (previamente cargada) tendrá como ancho, el ancho del scroll y como alto, el alto del scroll **Línea 58.**
* Se agregan las funciones del scroll, las cuales son, que este activado, que se guarde la posición en la que se está dentro del scroll, una velocidad inicial cuando no esté en movimiento, que no haya movimiento vertical, y se le agrega un Rectangle (Limites) **Líneas 59-69.**
* Se agrega imagen por imagen de los logros mediante un ciclo, el cual agrega la imagen a la ventana con la función **add.image()**, la cual recibe como parámetros la posición en X, la posición en Y y una cadena con la cual se identificara, luego se centra a esa posición, y por último se le agrega al scroll **Líneas 71-80.**
* Cuando el scroll este en movimiento, una variable predefinida como **isBeingDragged** estará en true y abra una velocidad inicial en el scroll, en el caso contrario (En el que el scroll no esté en movimiento), la variable **isBeingDragged** estará en false **Líneas 83-91.**
* Se inicializa una variable llamada **apodo** que agregara un texto a la ventana con la función **add.text()**, la cual recibe como parámetros, la posición en x, la posición en y, el texto a poner, y los estilos que se aplicaran al texto, también se centra en la posiciones dadas. Esta variable se usará para representar el nombre de tipo de logro **Líneas 62-63.**
* Se inicializa una variable llamada **description** que agregara un texto con la función **add.text()**, la cual recibe como parámetros, la posición en x, la posición en y, el texto a poner, y los estilos que se aplicaran al texto, también se centra en la posiciones dadas. Esta variable se usará para representar la descripción de cada tipo de logro **Líneas 64-65.**
* Se agrega el botón **startButton** a la ventana con la función add.button(), la cual recibe como parámetros, la posición en X, la posición en Y, una cadena con la cual se identificó el sprite cargado anteriormente, la función que realizara al ser presionado, el contexto en el que se usara (por lo general es this), el número de frame del estado Over (cuando el mouse este sobre el), el número de frame del estado Out (cuando no hay ninguna interacción con el botón), el número de frame del estado Down (cuando se presione el botón), el número de frame del estado Up (cuando se deje de presionar el botón). Este botón se usará para abrir el estado **navegacion** **Líneas 105-106.**
* Se agrega el botón **botonVolver** a la ventana con la función add.button(), la cual recibe como parámetros, la posición en X, la posición en Y, una cadena con la cual se identificó el sprite cargado anteriormente, la función que realizara al ser presionado, el contexto en el que se usara (por lo general es this), el número de frame del estado Over (cuando el mouse este sobre el), el número de frame del estado Out (cuando no hay ninguna interacción con el botón), el número de frame del estado Down (cuando se presione el botón), el número de frame del estado Up (cuando se deje de presionar el botón). Este botón se usará para regresar al estado **verPerfil** **Líneas 108.**
* Se crea una función llamada **verPerfil**, la cual será ejecutada cuando el botón llamado **botonVolver** sea presionado, dentro de esta función se inicia el estado llamado **perfilJugador** **Líneas 70-72.**
* Se declara una variable llamada **zoomed** que representara cuando un elemento del scroll este seleccionada **Línea 113.**
* Mediante dos ciclos aninados, se comprobará cuando una imagen dentro del scroll este seleccionada, el primero se encargará de agrandar al 150% la imagen, cuando esta esté seleccionada y pondrá la variable **zoomed** en true, y el segundo, modificara el nombre y la descripción de cada tipo de plagio, dependiendo de cuál este seleccionado. Además, se comprueba si algún logro esta desbloqueado o no **Líneas 121-150.**
* Cuando una imagen ya no está seleccionada, esta cambiara su tamaño al normal **Líneas 151-154.**
* Cuando haya movimiento dentro del scroll, se guardará la posición en una variable predefinida como **savedPosition Líneas 156-159.**
* Se define con que velocidad se moverá dentro del scroll, dependiendo de donde este la posición en x **Líneas 160-187.**

# Código Jsp

## Login.jsp

* Se especifica el tipo de codificación a usar, en este caso se utiliza la codificación UTF-8 **Línea 11.**
* Se coloca “Login Pacto de Honor” como nombre para la pestaña **Línea 12.**
* Se enlaza un script llamado **main.js Línea 14.**
* Se enlaza desde Google Fonts un tipo de letra, específicamente la letra “Roboto” **Línea 15.**
* Se enlaza un CSS (Hoja de Estilos) llamada **Decorator.css**, la cual se usará para darle estilos al archivo **Login.jsp Línea 16.**
* Se crea un formulario con el id **formInicio**, esté recibirá datos con el método POST **Línea 22.**
* Se agrega una imagen dentro del formulario, la cual es el logo del juego “Poli Fighters” **Línea 24.**
* Se agregan 2 inputs al formulario, uno de tipo **text** y otro de tipo **password**, estos tienen los id’s **textusuario** y **textpassword**, se usarán para que los jugadores inicien sesión **Líneas 25-26.**
* Se agrega un link dentro del formulario de texto **Forgot Password**, el cual se re direccionara al archivo **Forgot.jsp Línea 27.**
* Se agrega un link dentro del formulario de texto **Registrarse**, el cual se re direccionara al archivo **Registro.jsp Línea 28.**
* Se agregan 4 botones dentro del formulario, los cuales servirán para diferentes tipos de autenticaciones de los jugadores, el primero tiene id **btnlogin** y servirápara ingresarpor medio de usuario y contraseña, el segundo tiene id **facebook** y servirá para ingresar mediante facebook, el tercero tiene id **google** y servirá para ingresar con la cuenta de Google y el cuarto tiene id **Microsoft** y servirá para ingresar con la cuenta de Microsoft **Líneas 29-33.**
* Se agrega el siguiente texto “Derechos Reservados © Copyright HackDemons 2010-2017” **Línea 33.**

## Forgot.jsp

* Se especifica el tipo de codificación a usar, en este caso se utiliza la codificación UTF-8 **Línea 11.**
* Se coloca “Olvido Clave Pacto de Honor” como nombre para la pestaña **Línea 12.**
* Se enlaza desde Google Fonts un tipo de letra, específicamente la letra “Roboto” **Línea 14.**
* Se enlaza un CSS (Hoja de Estilos) llamada **Decorator.css**, la cual se usará para darle estilos al archivo **Forgot.jsp Línea 15.**
* Se crea un formulario que recibirá datos con el método POST **Línea 20.**
* Se agrega una imagen dentro del formulario, la cual es el logo del juego “Poli Fighters” **Línea 21.**
* Se agrega una etiqueta, la cual contiene un texto que es “Recupera tu clave” **Línea 22.**
* Se agrega un input de tipo **email**, este se usará cuando el jugador olvide su contraseña **Línea 24.**
* Se agregan dos botones dentro del formulario, los cuales tienen como texto **Enviar** y **Cancelar** respectivamente **Líneas 25-26.**
* Se agrega el siguiente texto “Derechos Reservados © Copyright HackDemons 2010-2017” **Línea 29.**

## Registro.jsp

* Se especifica el tipo de codificación a usar, en este caso se utiliza la codificación UTF-8 **Línea 11.**
* Se coloca “Registro Pacto de Honor” como nombre para la pestaña **Línea 12.**
* Se enlaza desde Google Fonts un tipo de letra, específicamente la letra “Roboto” **Línea 14.**
* Se enlaza un CSS (Hoja de Estilos) llamada **Decorator.css**, la cual se usará para darle estilos al archivo **Registro.jsp Línea 15.**
* Se crea un formulario que recibirá datos con el método POST **Línea 20.**
* Dentro de este formulario se agregan 7 inputs, los cuales son de tipo **text, email y password,** estos se usarán para que el usuario ingrese el nombre, el apellido, el email, el país, la ciudad, el usuario, la contraseña **Líneas 23-29.**
* Se agrega un botón el cual enviara los datos ingresados **Línea 30.**
* Se agrega el siguiente texto “Derechos Reservados © Copyright HackDemons 2010-2017” **Línea 29.**

# Código Css

# Código Java