

MANUAL DE RPG

DERROTA AL JEFE BANDIDO

Proyecto javaScript para desarrollo web en entorno cliente

DERROTA AL JEFE BANDIDO

Has sido llamado a filas para defender tu tierra, un infame bandido coordina una banda de maleantes que asola nuestras tierras y debe ser derrotado. Pero lo primero es lo primero, ¿como te llamas?

Buen nombre, aunque necesitarás mas que eso para hacerte fuerte. ¿Que te gustaria ser?

Los guerreros son los mas balanceados y no dependen tanto de la magia.

Los magos son muy poderosos pero al confiar mucho en su magia son fisicamente débiles.

Entiendo, puedes ser util ahi fuera. Toma este equipamiento y gánate un nombre luchando. Sea como sea, derrota a tu objetivo

El juego comienza con una pequeña introducción. En ella, podremos personalizar un poco nuestro personaje tanto su nombre como su clase.

Ser guerrero o ser mago cambia la jugabilidad del juego. De ser guerrero podrás ir cambiando entre ataques mágicos o físicos, pero si eliges mago los ataques físicos serán tan débiles que deberás usar casi solo ataques mágicos.

Empezaremos con un equipamiento inicial que nos ayudará en el comienzo.

Te adentras en un pueblo y encuentras varios lugares de interés:

Tienda

Posada

Bosque

Información

Tablón de anuncios

Este es el menú principal, con él podrás navegar entre las cinco distintas secciones que mencionaremos más adelante, pero como resumen podemos decir que su función es:

Tienda: Comprar nuevo equipamiento.

Posada: Curar a nuestro personaje a cambio de unas monedas.

Bosque: Enfrentarnos a los diferentes enemigos que hay disponibles.

Información: Ver información de nuestro personaje.

Tablón de anuncios: Consultar diferentes misiones y su recompensa.

Te adentras en un pueblo y encuentras varios lugares de interés:

[Tienda](#) [Posada](#) [Bosque](#) [Información](#)
[Tablón de anuncios](#)

Tienes 100 monedas.
Que deseas comprar?

ARMAS

Espada oxidada. Precio: 100 monedas. [Comprar](#)
Espada de soldado. Precio: 300 monedas. [Comprar](#)
Hacha de veterano. Precio: 500 monedas. [Comprar](#)
Espada legendaria. Precio: 1000 monedas. [Comprar](#)

ARMADURAS

Ropajes de aventurero. Precio: 100 monedas. [Comprar](#)
Ropajes reforzados. Precio: 300 monedas. [Comprar](#)
Armadura de placas. Precio: 500 monedas. [Comprar](#)
Armadura legendaria. Precio: 1000 monedas. [Comprar](#)

UTILIDAD

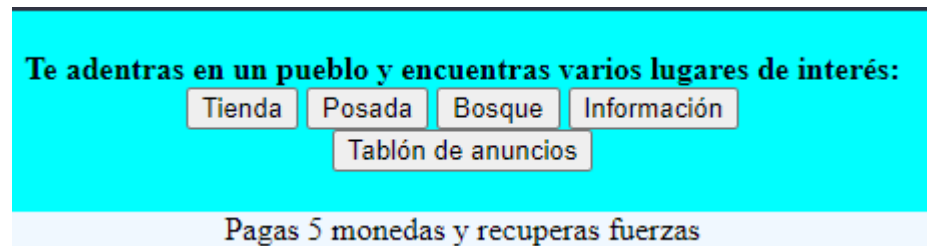
Comprar poción: Precio 15 monedas [Comprar](#)

En el apartado “Tienda”, podemos consultar nuestro dinero y comprar diferentes armas o armaduras. Esto será necesario a lo largo del juego

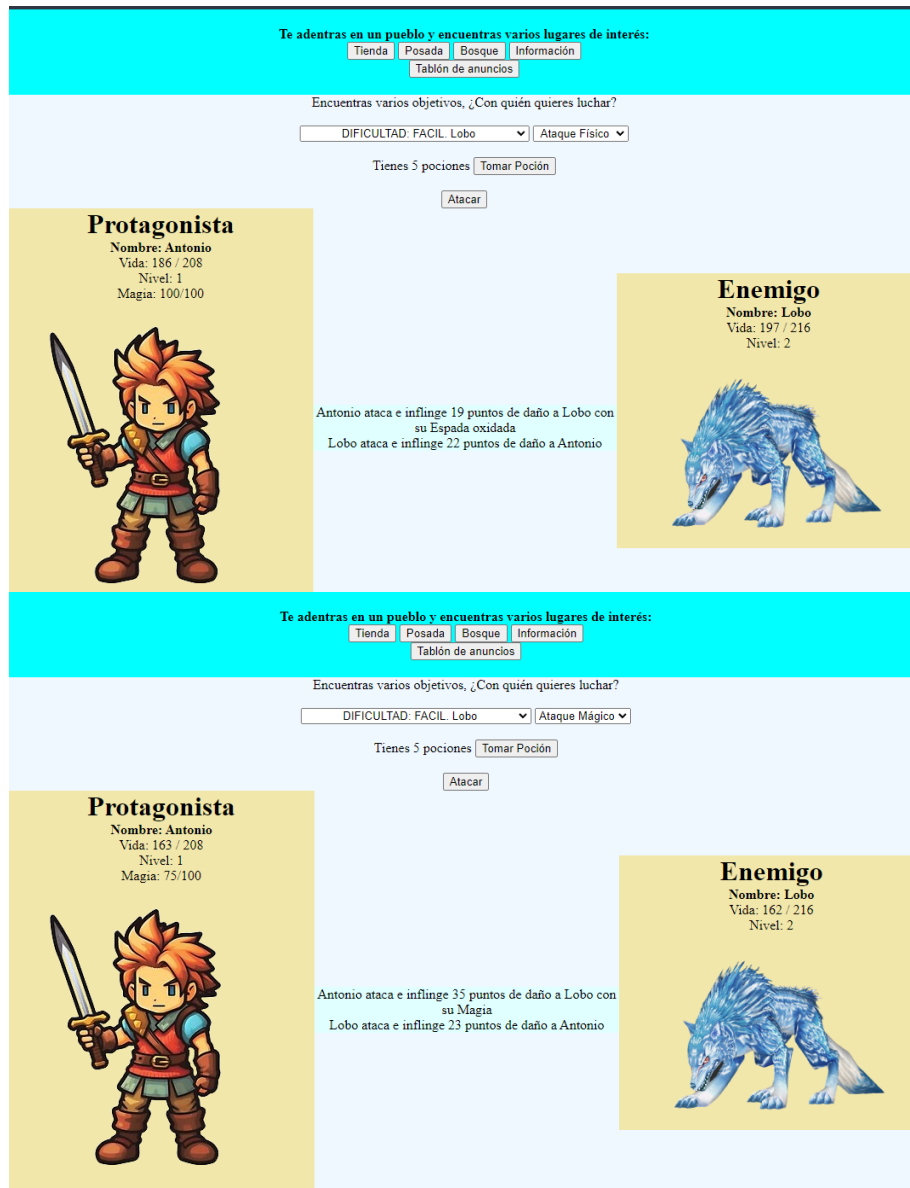
Las **armas** incrementan el daño que produce nuestro personaje, tanto del ataque físico como el mágico.

Las **armaduras** incrementan la vida del personaje.

Las **pociones** regeneran al completo la magia del personaje.



Al clicar en posada, se cobrará al personaje una cantidad y su "Vida" será regenerada.



Antes que nada, hay que explicar que existen dos estadísticas en tu personaje, la **fuerza** y la **vida**.

La **fuerza** se calcula con el nivel y el arma equipada más un multiplicador, y modificará el daño que infringe el personaje

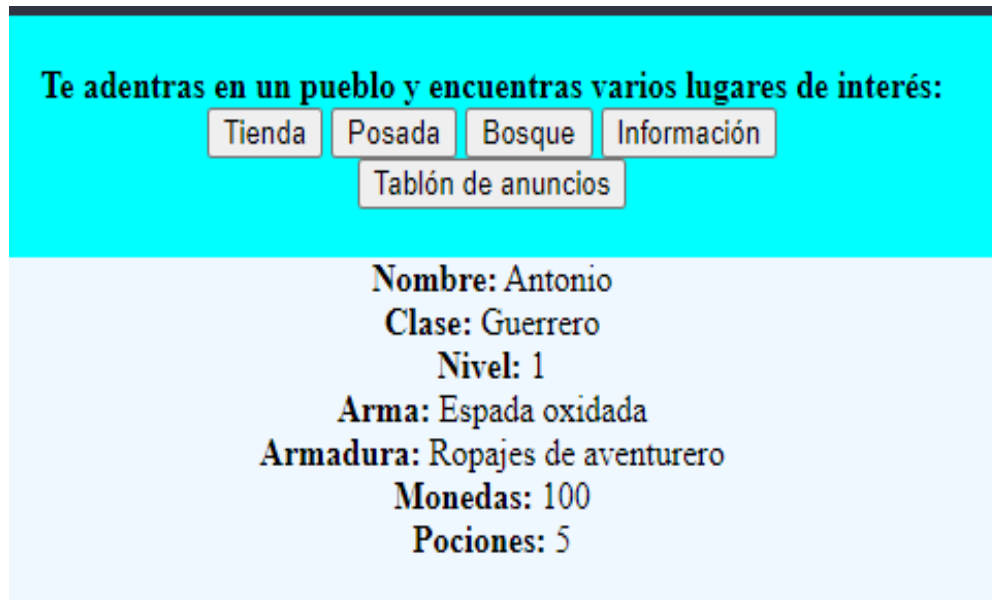
La **vida** se calcula con el nivel y la armadura equipada más un multiplicador. Si tu Vida llega a 0, será **Game Over**.

Existen tres enemigos, los **Lobos**, los **Bandidos** y el **Jefe Bandido**(el objetivo a eliminar para acabar el juego). Se recomienda empezar por los **lobos**.

Dependiendo del ataque y clase usada, tu personaje hará un daño u otro. Por ejemplo, si eres **Guerrero** harás un daño decente en general pero tu magia no será tan poderosa. Si eres **Mago** harás un gran daño mágico pero tu daño físico será muy bajo.

Cabe destacar que los ataques mágicos consumirá **magia**, y la única forma de recuperarla será con pociones. Si se hace un ataque mágico cuando la magia está a 0, el ataque realizará 0 puntos de daño y se perderá el turno.

Hay que valorar cuándo usar ataques físicos y mágicos y aprovechar lo máximo posible la magia.



El apartado información muestra tanto el equipamiento equipado actualmente como la **clase**, **nombre**, **nivel**, **cantidad de monedas** y total de **pociones** disponibles.

Te adentras en un pueblo y encuentras varios lugares de interés:

Tienda

Posada

Bosque

Información

Tablón de anuncios

MISIÓN: Acaba con 7 lobos.

Recompensa: 100 Monedas

Entregar misión

MISIÓN: Acaba con 20 lobos.

Recompensa: 300 Monedas

Entregar misión

MISIÓN: Acaba con 10 Bandidos.

Recompensa: 500 Monedas

Entregar misión

Te adentras en un pueblo y encuentras varios lugares de interés:

Tienda

Posada

Bosque

Información

Tablón de anuncios

MISIÓN COMPLETADA: Has recibido 100 monedas

MISIÓN: Acaba con 20 lobos.

Recompensa: 300 Monedas

Entregar misión

Aun tienes que acabar con 10 lobos mas

MISIÓN: Acaba con 10 Bandidos.

Recompensa: 500 Monedas

Entregar misión

Aun tienes que acabar con 10 bandidos mas

Finalmente, en el apartado Tablón de anuncios, cuando vayamos derrotando enemigos, podremos completar misiones y recibir monedas a cambio para ayudarnos a progresar más rápido. También podremos ver cuantos enemigos nos falta para cada misión o si ya completamos alguna misión

Explicación técnica JS

DERROTA AL JEFE BANDIDO

Proyecto javascript para desarrollo web en entorno cliente

La ejecución del juego empieza al darle a “comenzar aventura”. Una función toma los datos del formulario y construye el objeto tipo personaje llamado “protagonista”. También dibuja el menú en el html y oculta la pantalla del formulario.

```
<button onclick="comenzar()">Comenzar aventura</button>
```

La función sería esta.

```
// Esta función oculta el main inicio, muestra el main principal
// y crea el objeto que usaremos como personaje principal.
function comenzar() {
  let nombre = document.getElementById('nombre').value;
  let clase = document.querySelector('select[name="clase"]').value;
  protagonista = new personaje(
    nombre,
    clase,
    "Espada oxidada",
    "Ropajes de aventurero",
    1
  );
  document.getElementById('inicio').setAttribute('hidden', true);
  document.getElementById('principal').removeAttribute('hidden');
};
```

Lo más importante del código de este juego es la función constructora “**personaje**”. Además de ocupar más de la mitad del código de mi script, es la que se encarga tanto de construir a los objeto personaje, gestionar los combates, curarlos, gestionar las subidas de nivel, las compras, las misiones... Más adelante veremos en profundidad cómo funcionan, pero por ahora solo aclarar que además me apoyo en otras funciones para trasladar la información al html.

Para mostrar la información en el html y poder ver las diferentes secciones como Tienda o Bosque, he implementado diferentes <main>. Cada uno contiene la información de una sección, por lo que tengo seis diferentes(contando que uno de ellos es para mostrar la pantalla de introducción). De inicio, estos <main> tienen el atributo hidden de base para no ser mostrados hasta que sea necesario.

Cuando hago click en el menú, en el html se llama a una función dependiendo del botón pulsado.

```
<div class="barra">
  <p>
    <strong>Te adentras en un pueblo y encuentras varios lugares de interés:</strong>
  </p>
  <button onclick="menu('tienda') ">Tienda</button>
  <button onclick="menu('posada') ">Posada</button>
  <button onclick="menu('bosque') ">Bosque</button>
  <button onclick="menu('informacion') ">Información</button><br>
  <button onclick="menu('tablon') ">Tablón de anuncios</button><br>
</div>
```

Desarrollo web en entorno Cliente 2ªDAW Jesús Cordero Ávila

A su vez, ahora en el código js existe una función menu que dependiendo del parámetro con la que se llama, realizará una de 5 acciones (Una por cada opción disponible en el menú).

Como ejemplo mostraré uno de las opciones disponibles en la función:

```
function menu(opt) {  
  if (opt == "bosque") {  
    enemigoDebil.curar();  
    enemigoMedio.curar();  
    enemigoFuerte.curar();  
    document.getElementById("nPociones").innerHTML = `Tienes ${protagonista.equip["pociones"]} pociones`;   
    document.getElementById("infoPr").innerHTML = "";  
    document.getElementById("infoEn").innerHTML = "";  
    document.getElementById("log").innerHTML = "";  
    limpiar(opt);  
  }  
  else if (opt == "posada") {...  
    ...}  
  ...};
```

Cuando hago click en una sección, normalmente se llama a la función limpiar que oculta el resto de main menos el que quiero ver.

```
const opMenu = ["bosque", "posada", "tienda", "informacion", "tablon"]  
  
// Esta función oculta los elementos que no sean necesarios tomándolos de opMenu.  
function limpiar(opt) {  
  for (var x of opMenu) {  
    if (x != opt) {  
      document.getElementById(x).setAttribute('hidden', true);  
    } else {  
      document.getElementById(opt).removeAttribute('hidden');  
    }  
  }  
}
```

La función constructora personaje es demasiado extensa para ponerla entera aquí, por lo que mostraré únicamente sus atributos y dejaré que los comentarios en el código expliquen el funcionamiento de los métodos.

```
function personaje(nombre, clase, arma, armadura, nivel) {  
  this.nivel = nivel;  
  this.nombre = nombre;  
  this.clase = clase;  
  // esta propiedad recoge el equipo del objeto. Esto sirve para calcular las stats.  
  this.equip = {  
    "arma": arma,  
    "armadura": armadura,  
    "monedas": 100,  
    "pociones": 5  
  };  
  // Esta propiedad recoge las stats del objeto.  
  this.stat = {  
    "fuerza": eval(conf["fuerza"]),  
    "hp": [eval(conf["hp"]), eval(conf["hp"])],  
    "experiencia": 0,  
    "magia": [100, 100]  
  };  
  this.bajas = { "Lobo": 0, "Bandido": 0 };  
}
```

Primero, la función recoge algunos de los valores de los atributos del objeto. Lo más importante son los objetos stat y equip, ya que en ellos gestionan el equipamiento y las stats del objeto.

“stat” se calcula tomando los valores de “equip” y el nivel las estadísticas del personaje, aunque como se puede ver en el código, hace nada con estos y en su lugar usa otro objeto. Esto se debe a que lo que debería ir ahí está escrito arriba del todo en un objeto llamado conf. Desde ahí puedo aumentar o reducir la fórmula de vida, fuerza, costes de armas, armaduras... De esta forma si quiero ajustar los parámetros del juego no necesito buscar en el código, ya que todo está junto.

A continuación muestro el objeto conf.

```
const conf = {
  //Cálculo de fuerza
  "fuerza": "armas[this.equip['arma']] + (this.nivel * 5)",
  //Cálculo de vida
  "hp": "100 + (this.nivel * 8) + (armaduras[this.equip['armadura']] * 10)",
  //Coste posada
  "posada": 5,
  //monedas recibida
  "monedas": "Math.trunc(enemy.stat['fuerza'] / 2)",
  //Experiencia recibida
  "exp": "Math.trunc(50 + enemy.stat['fuerza'])",
  //Aleatoriedad de los daños
  "daño": "Math.trunc(getRandomInt(enemy.stat['fuerza'] / 3))",
  //Coste de las armas
  "cArma": "armas[obj] * 10",
  //Coste de las armaduras
  "cArmadura": "armaduras[obj] * 10"
};
```

Al evaluar las cadenas, puedo insertar ese código donde lo necesite.

Existe un método “combate” que al invocarse sobre un objeto “personaje” y pasándole de parámetro otro objeto “personaje”, crea un combate entre los dos. Se calcula el daño dependiendo de la clase y del tipo de ataque y se reduce la vida al enemigo. Si la vida del enemigo llega a cero, se invoca a otro método llamado combateVictoria que se encarga de dar la experiencia y recompensa en base de la dificultad del enemigo vencido. También tiene en cuenta de si el enemigo vencido es el Jefe Bandido, y de serlo genera la **pantalla de victoria** y Fin del juego. En cambio, si la vida del protagonista es la que llega a cero, se genera la pantalla de **Game Over** y habría que volver a empezar.