

**DOCUMENTO
TRABAJO FINAL
DWE CLIENTE**

DIEGO ALARCÓN ORTIZ

2ºA DAW



MENSAJE IMPORTANTE

**PARA TRUCAR EL JUEGO, VAYA A LA
LÍNEA 309 Y VEA LOS COMENTARIOS**



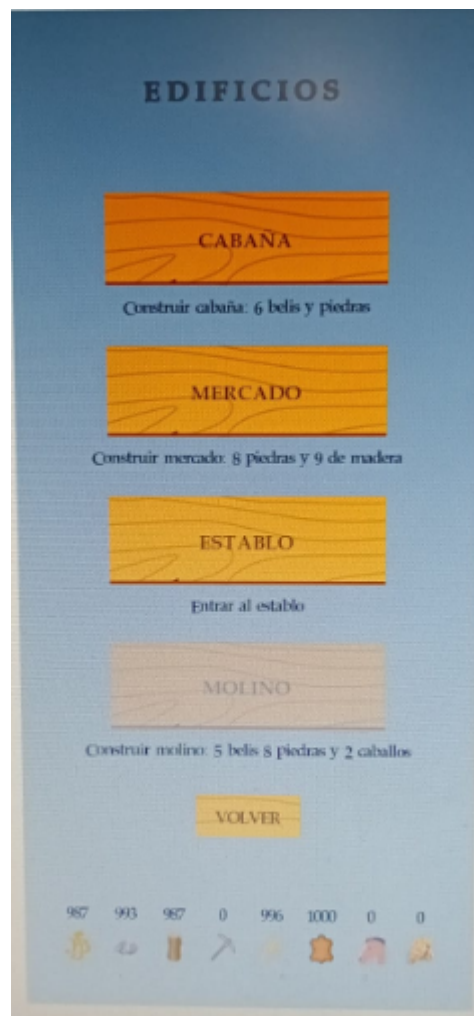
PIRATYCOON

El desarrollo de la aplicación constó de una semana entera de trabajo, en la cual existieron diferentes fases de diseño y creación.

El primer planteamiento que hice fue la implementación correcta del enunciado y las funcionalidades requeridas, quedando un juego que, según lo estipulado por el profesor, funcionaba correctamente.

Sin embargo, era escueto, tenía un diseño pobre y una jugabilidad nula, todo se basaba en las peticiones del ejercicio, por ello, tras terminar toda la funcionalidad del juego, entenderla y saber desarrollarla, pasé a la siguiente fase.

A continuación se adjuntan imágenes lo explicado:



En esta segunda fase, decidí tomarme un tiempo para pensar sobre la interfaz, ya que una vez hecha la funcionalidad, lo que prima en un juego son los elementos visuales y el dinamismo de este.

Para ello empecé a estructurar diferentes plantillas para decidir cómo estructurar toda la información de manera que no quedase recargado y que se entendiese.

La principal idea que mantuve hasta el final fue el hecho de dividir el juego en diferentes secciones donde hacer cosas diferentes en cada una, lo que a su vez me llevó a plantear una pequeña narrativa que siga el progreso del juego, para así poder dotar al usuario de una experiencia menos monótona.

La narrativa es sencilla y únicamente hace que exista un dónde y un por qué de tus acciones, además, está inspirada en la obra japonesa One Piece.

Se basa en un aventurero (el usuario) que desea ser pirata y hacerse a la mar con su propio barco. Vives en una aldea donde necesitas conseguir todos los recursos suficientes para construir diferentes edificaciones que te hagan ser autosuficiente y poder construir el muelle, que es el objetivo final añadido después de la taberna.

Una vez construido tu barco, acaba el juego, pero se han realizado breves avances sobre futuras implementaciones a la hora de poder navegar, esto lo comentaremos más adelante.

Volviendo a las secciones, una vez establecida la narrativa, se distinguen tres secciones diferentes en el juego:

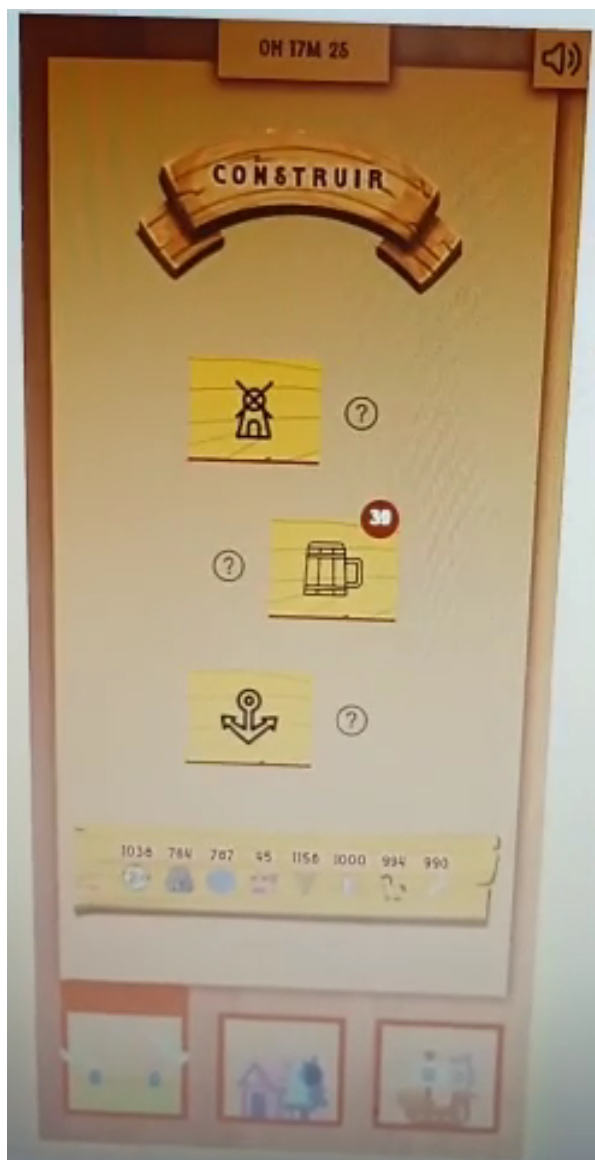
- CONSTRUCCIONES
- ALDEA
- BARCO



De esta forma, todas las funcionalidades quedan mejor estructuradas y se diferencian entre sí.

Además, en este proceso también se mejoró la interfaz, que a lo largo de los días terminó quedando así

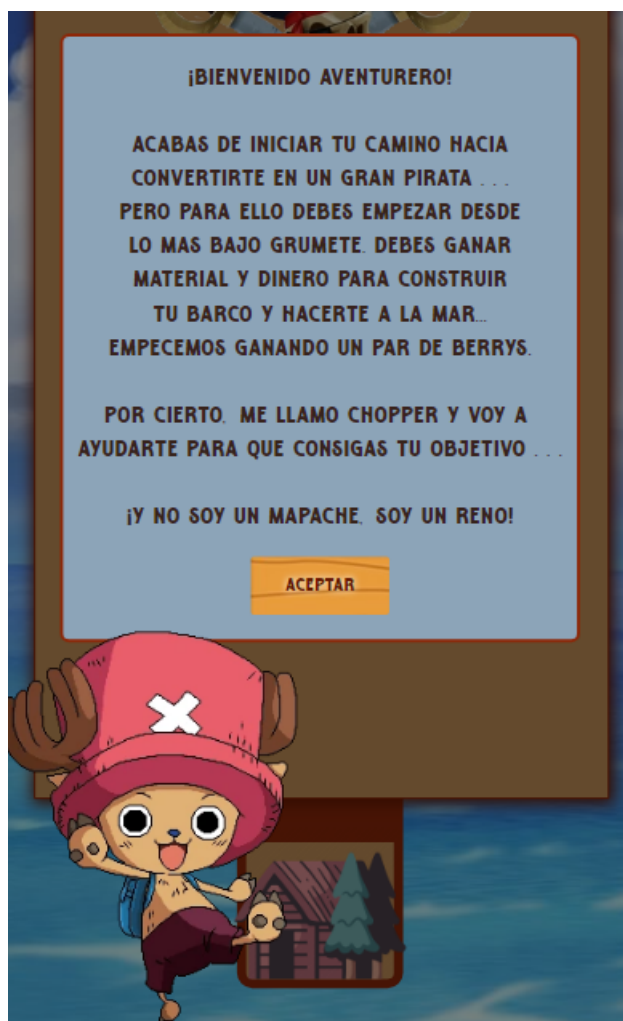
IMAGEN PREVIA AL DISEÑO FINAL, NO ES EL DEFINITIVO



En esta imagen ya podemos ver los aditivos del sonido, el contador de tiempo de juego y los “assets” necesarios para una interfaz aceptable.

Además de esto, se añade al juego un acompañante que te ayuda a proseguir la historia, dando mensajes informativos, mensajes de error y un tutorial inicial, lo que ameniza la experiencia narrativa.

Se adjunta imagen de ejemplo:



En la versión final del juego, la interfaz varía para aumentar la calidad de imagen y evitar un diseño escaso o pobre, esta se puede ver en los archivos adjuntos en la carpeta, donde se puede jugar al juego.

Por último cabe destacar la integración de un mapa donde, como futura implementación, se indica un camino por el que el usuario puede navegar y recorrer islas en las que podrá recolectar diferentes materiales y seguir una narrativa más compleja.

(Véase también en los archivos de la versión final)

Pasando a la parte del desarrollo enfocada al código, las tecnologías usadas han sido HTML, CSS y JavaScript, sin el uso de ningún framework o librería.

Primeramente desarrollé todo el HTML en el fichero index.html correspondiente para así facilitar visualmente la estructura, la cual fue pasada a JS posteriormente para optimizar el peso y el rendimiento del juego.

Para ello realicé una función manualmente donde con una sola línea de código podría crear nodos, unirlos y asignarle diferentes atributos.

```
function crearElemento(elemento, clase, id, src, texto, padre) {  
  let elem = document.createElement(elemento)  
  if (clase !== "") elem.setAttribute("class", clase)  
  if (id !== "") elem.setAttribute("id", id)  
  if (src !== "") elem.setAttribute("src", src)  
  if (texto !== "") elem.appendChild(document.createTextNode(texto))  
  document.querySelector(padre).appendChild(elem)  
}
```

Por otra parte, en cuanto a la estructura seguida para el avance del juego, tuve en cuenta dos factores principales:

- Debían declararse los recursos con los que el jugador empieza e ir modificándose según las acciones
- Declarar el estado en el que el usuario se encuentra, es decir, qué edificios tiene contruidos y qué puede o no puede hacer.

Para ello creé las variables de los recursos y variables booleanas de "edificioExiste".

En caso de que un edificio no exista, no se van a poder hacer diferentes acciones, que son independientes de cada construcción. A su vez, el tener o no tener los suficientes recursos para realizar alguna acción también implica un cambio en las posibilidades de juego.

Otras dos funciones fundamentales para esto que se acaba de explicar son:

- actualizarRecursos()
- comprobarRecursos()

Estas funciones permiten que siempre que se realice una acción, se ejecuten ambas, lo que permite actualizar el número de recursos que posee el usuario y comprobar de nuevo qué puede y qué no puede hacer según los mismos.

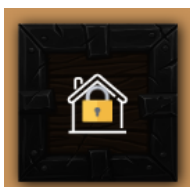
(Véanse ambas desde la línea 326 a la 413)

En cuanto a la construcciones, cada una tiene su propia función “construirEdificio” que tendrá un condicional según si está o no construido cuando el usuario pulse en su botón, realizando así acciones diferentes cada vez que se pulse, pudiendo estas ser:

- Entrar al edificio para realizar acciones dentro de él
- Si no es un edificio con acciones dentro de él, informar sobre que ya se ha construido
- Informar sobre la construcción del mismo si se han tenido los recursos suficientes
- Informar sobre la falta de recursos o progreso para su construcción

Por otra parte, según la situación en la que nos encontremos (De entre las mencionadas), el botón tendrá un estado u otro

Edificio bloqueado por progreso



Edificio disponible para construir, pero falta de materiales



Edificio disponible para construir



Edificio ya construido



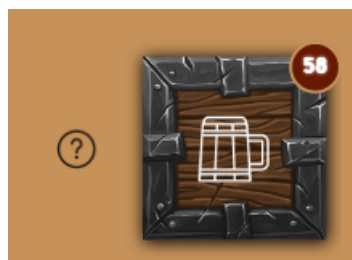
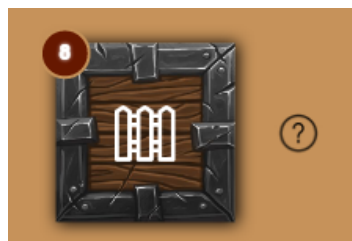
Además en la sección de construcción se ha aplicado un DIV scrollable para así dinamizar y estructurar correctamente la navegación por los edificios (Véase en el juego).

Cada botón tiene al lado un icono de info que permite al usuario poder saber qué acción conlleva pulsar cada botón del juego, por supuesto siendo esta información dada por el acompañante mencionado anteriormente.



Los mensajes de información emergentes son siempre el mismo DIV, que se oculta y se muestra con un texto diferente según la acción realizada, siendo esto igual con todos los elementos del juego que van apareciendo y ocultándose.

En cuanto a algunas mejoras realizadas, se ha añadido en el mercado un contador de cantidad para la compra, agilizando así que si el usuario quiere comprar 5 unidades de madera, no tenga que comprar 5 veces. Además también se ha añadido un contador a los edificios que aportan recursos cada cierto tiempo.





Por último, en cuanto al guardado del progreso, este se guarda de forma automática en localStorage, en cada cambio que se hace en el juego, se aplica un cambio en el localStorage. De esta forma, al darle al botón de cargar partida, todos los progresos se aplicarán para así poder seguir con el juego.

FIN