

Documentación del “Reto 1”

Luis Daniel Barragués
Imanol Luis
Unai Puelles
Jon Maneiro

Índice

Introducción	3
Planificación y desarrollo	3
Documentación del código	4
index.js	4
Variables:	4
Funciones:	4
index_jquery.js	4
Variables:	4
Funciones:	5
automatico.js	5
Variables:	5
Funciones:	5
sub_mm.js	5
Variables:	5
Funciones:	6
sub_manual.js	6
Variables:	6
Funciones:	6
estadisticas.js	7
Variables:	7
Funciones:	7
Guía de estilo	8
Estructura de la página	8
Estructura de la página en la versión móvil	9
Paleta de colores	9
Estructura de carpetas	10
Fuentes	10
Elementos gráficos y logotipos	11
Imágenes	11
Propuestas y futuro	11
Anexo	12

Introducción

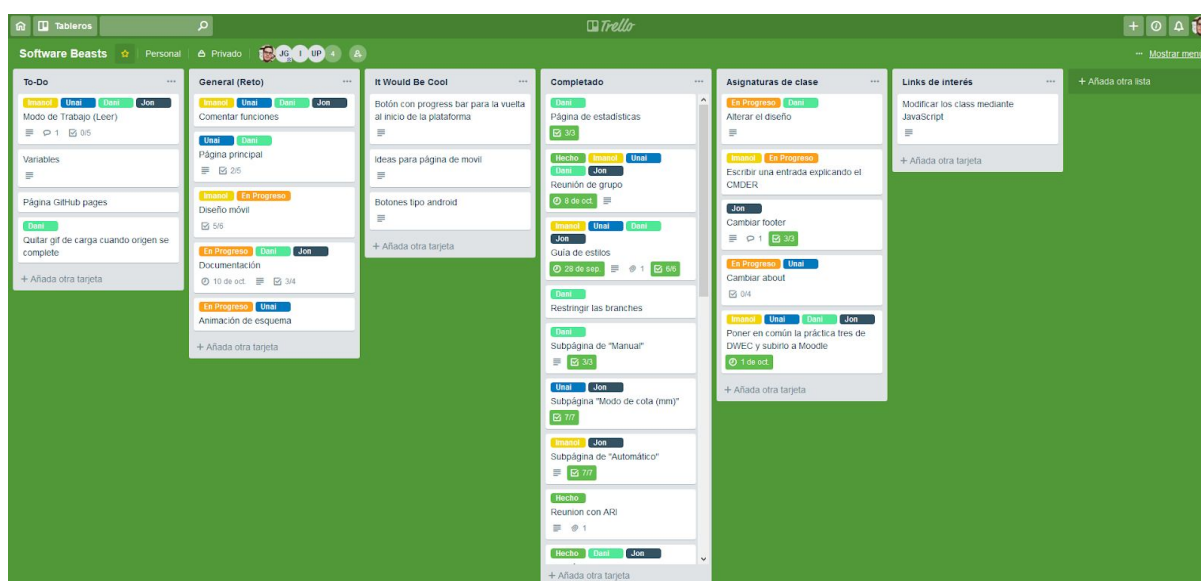
Somos la empresa Software Beasts que se dedica al desarrollo de aplicaciones web, y hemos comenzado trabajando para una empresa local que se dedica a la comercialización de latas de conservas. El primer reto que se nos propone es realizar una aplicación web para llevar el control total de un autómata que tiene controlar las distintas fases de producción. En este documento se detalla el desarrollo, los conceptos y el diseño que hemos seguido.

Planificación y desarrollo

Para poder funcionar como un equipo ordenadamente, llegamos a un consenso en las herramientas que íbamos a utilizar, los roles que íbamos a desempeñar y cómo íbamos a distribuir el trabajo.

Las herramientas decididas son:

- Trello, para llevar un historial de lo realizado y lo que queda por hacer.
- GitHub, para almacenar el código y la documentación referente al reto.
- Team Drives, para elaborar documentos de forma común.



Nuestro tablero de Trello

A la hora de planificar, decidimos que la mejor forma era ir usando Trello, creando tarjetas con fechas de expiración cuando se nos ocurriesen nuevas ideas que implementar, manteniendo un historial de lo que estaba en proceso, completado o aún sin realizar.

Durante el desarrollo de la aplicación tuvimos diferentes conflictos de opiniones y de gustos, pero tras votaciones y propuestas los solventamos sin problemas.

Todos nos hemos encontrado a gusto, en confianza y animados mientras hemos ido evolucionando la aplicación, descubriendo nuevas cosas juntos e intentando ayudarnos entre todos.

Documentación del código

index.js

Variables:

arrayVelocidad,contador,tiempo.

Funciones:

leerVarPos() :

Esta función lee las 8 variables de posición para tenerlas guardadas.

CambiarModo(modos) :

Mantiene la selección del modo en el nav y guarda el modo en el que se queda.

avanzar() :

Mueve el indicador de posición a la posición actual.

GuardarDatos() :

Guarda datos necesarios para las estadísticas.

window.onbeforeunload : (Línea 18)

Vacía el almacenamiento local.

setInterval : (Línea 90)

Lee la posición actual y la velocidad actual y las muestra cada medio segundo, además de mover el indicador de posición actual.

index_jquery.js

Variables:

modoActivo.

Funciones:

comprobarUso() :

Comprueba el estado de la variable modoActivo.

Devuelve un booleano.

volverOrigen() :

Esta función envía una señal true, y después de un segundo envía false, cambia la variable ON_ORIGEN.

RESET() :

Esta función envía una señal true, y después de un segundo envía false, cambia la variable RESET.

automatico.js

Variables:

automaticoCheck.

Funciones:

enviarDatosx() :

Envía los datos de los inputs a la base de datos.

alternarAutomatico() :

Alterna los botones entre activados y desactivados dependiendo si el modo está activo, además activa y desactiva el modo.

enviarSennalx(nomVar,val) :

Recibe el nombre de la variable y el valor que hay que asignarle(Que siempre será true).

Envía true a la variable a la variable correspondiente y después de un segundo le envía false.

sub_mm.js

Variables:

mmCheck.

Funciones:

enviarDatos() :

Envía los datos de las posiciones y de la velocidad introducidos en los inputs.

alternarMM() :

Activa y desactiva los botones dependiendo de si el modo está activado y además alterna el modo entre activado y desactivado.

enviarSennalxx(nomVar,val) :

Recibe el nombre de la variable y el valor que hay que asignarle(Que siempre será true).

Envía true a la variable correspondiente y después de un segundo le envía false.

sub_manual.js

Variables:

manualCheck,adelanteCheck,atrasCheck.

Funciones:

alternarManual() :

Activa y desactiva los botones dependiendo de si el modo está activado y además alterna el modo entre activado y desactivado.

enviarSennal(nomVar,currentId,otherId) :

Recibe el nombre de la variable y dos Id, la del botón que activa la función y la del botón contrario.

Envía un true a la variable correspondiente y deja el botón pulsado, si se clica de nuevo, envía false.

alternBotn(currentId,otherId) :

Recibe el id del botón que se activa y del botón contrario.

Alterna el estado de los botones entre activado y desactivado, impidiendo que se active uno si el otro ya está activado.

estadisticas.js

Variables:

tiempototal , velocidad , reparto , c, ctx, velMedia, arrayVelocidad.

Funciones:

No existen funciones como tal, todo se ejecuta nada más cargar la página.

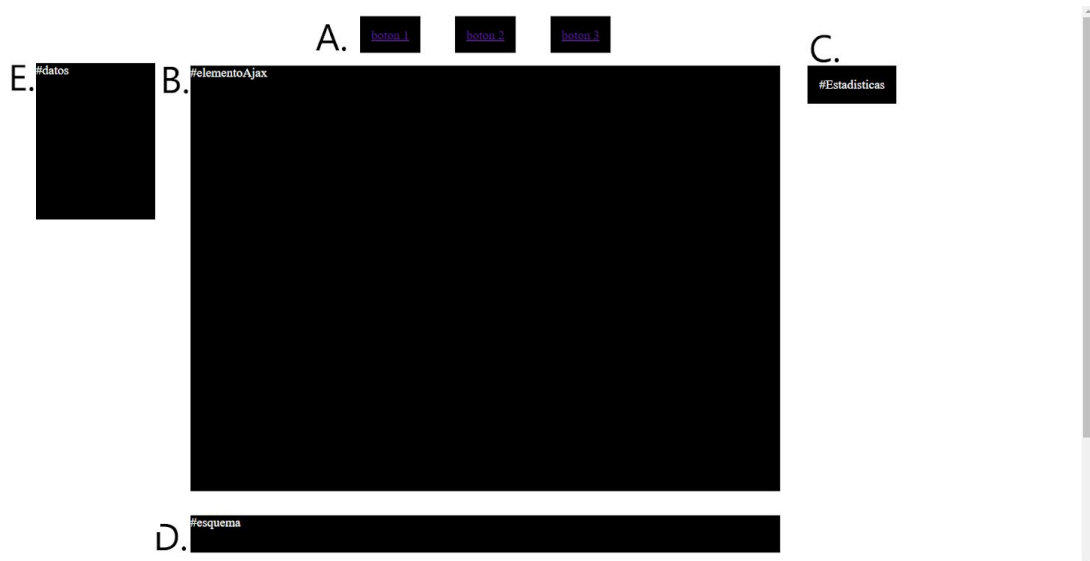
Podemos distinguir diferentes trozos de código que hacen diferentes funciones:

- La primera línea escribe en el documento el modo del que se muestran las estadísticas.
- Luego vemos la creación del gráfico, que usa la etiqueta “canvas” de HTML5 para dibujar un gráfico que depende del array de velocidad que se ha generado durante la ejecución del modo.
- Más adelante nos encontramos 3 “IF” que imprimen las diferentes estadísticas en la página, dependiendo del modo que sea.
- Por último, se rellena el historial de posiciones, mostrando cuántas veces ha estado el robot en cada posición.

Guía de estilo

Estructura de la página

La estructura de la página principal estará dispuesta de la siguiente forma, dando prioridad a los elementos más importantes (los que se usarán para controlar el robot):



Todos los elementos, excepto el footer, están separados 30px de los elementos de arriba. El la barra del footer tiene 90px de separación de los elementos de arriba.

La barra de navegación (A.) está alineada al centro con un margen entre elementos de 20px.

El div (B.) que cargará páginas tiene una anchura de 50% de la página y una altura de 550px.

El botón de estadísticas (C.) tiene una posición absoluta y está separado 76.5% del margen izquierdo.

El esquema en tiempo real (D.) tiene de ancho del 50% de la página y una altura de 50px.

El recuadro de datos (E.) tiene una separación del borde izquierdo de 13% y una anchura del 10%.

El botón origen se encuentra en el margen izquierdo del esquema en tiempo real y tiene una separación del borde izquierdo de 13% y una anchura del 10%.

El botón reset se encuentra en el margen derecho del esquema en tiempo real y tiene una separación del borde derecho de 13% y una anchura del 10%.

El footer está separado por una línea que tiene 90px de margen con el elemento superior y está compuesto por iconos e información de contacto.

Estructura de la página en la versión móvil

La estructura de la página principal en la versión móvil estará dispuesta igual que en la versión de escritorio con los siguientes cambios:

El cuerpo de la página tiene una anchura del 90% y todos los elementos están con una anchura del 100%.

La barra de navegación está alineada al centro con cada elemento con una anchura del 100% de la página.

El botón de estadísticas tiene una posición estática y está encima del div que controla el autómata.

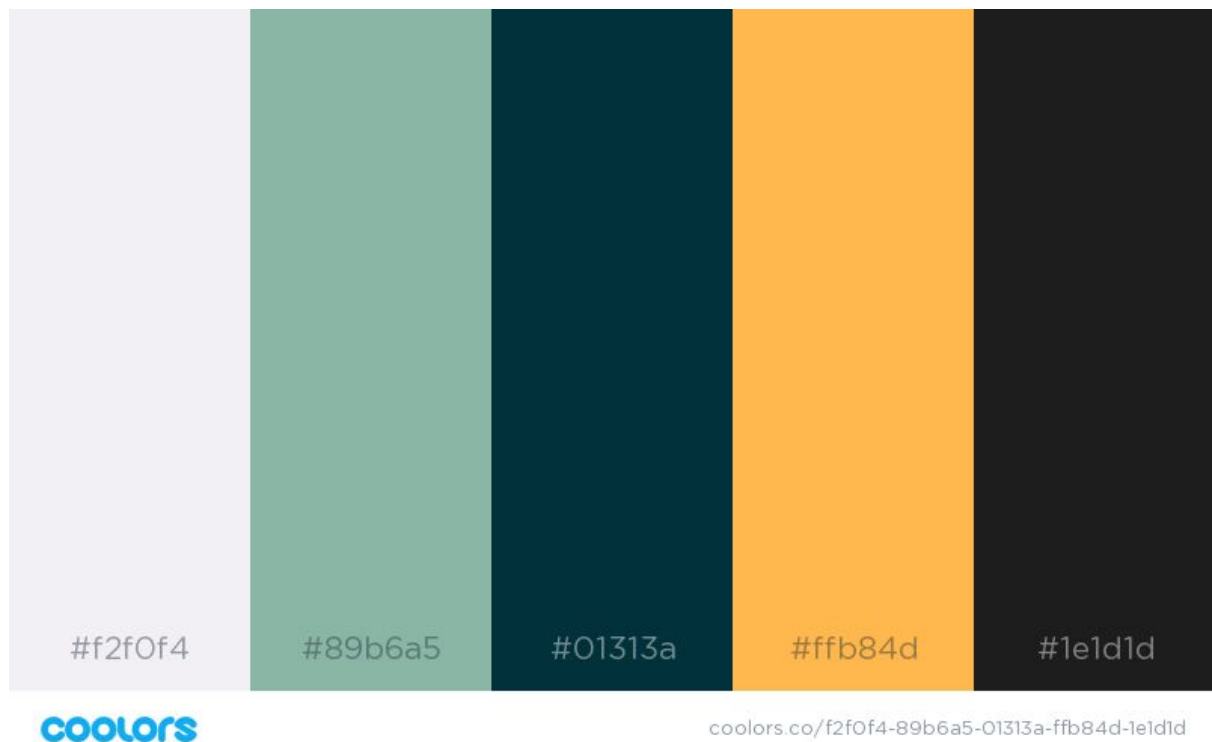
El botón de reset tiene una posición estática y está debajo del div que controla el autómata.

El botón de origen tiene una posición estática y está debajo del botón de reset.

El recuadro de datos y el esquema en tiempo real están ocultos.

Paleta de colores

El diseño de colores que va a utilizar la página es el siguiente. Consisten en 5 diferentes colores en los que nos encontramos diferentes tonos de azul, negro, blanco y un color de resalte naranja.



Para el fondo de la página utilizaremos el color #f2f0f4

El color del fondo del nav y del footer será #89B6A5

Para los botones y la selección del nav utilizaremos #01313a

Para la selección de botones se utilizará #ffb84d

Los colores de respaldo para efectos serán el restante #1e1d1d

El color de las letras que queden dentro de los botones será #f2f0f4

Estructura de carpetas

Está dividida de la siguiente forma:

- Web
 - pages
 - media
 - leerVar
 - css
 - js

“pages” contiene las subpáginas y páginas de la web, es decir, todo HTML que no sea el index será guardado en esta carpeta.

“media” contiene las imágenes y logotipos usados en la web.

“leerVar” contiene las páginas que contienen las variables para leerlas en la web.

“css” contiene todo el css de las páginas, subpáginas y página principal.

“js” contiene todo el JavaScript de las páginas, subpáginas y página principal.

Fuentes

Para el texto que se muestre en la página, se utilizará la fuente Verdana, y como fuente de respaldo se utilizará Calibri, en caso de que las anteriores fallen, se utilizará serif como último recurso.

the quick brown fox
jumps over the lazy dog.
THE QUICK BROWN FOX
JUMPS OVER THE LAZY DOG.

Verdana

The quick brown fox jumped over *the lazy dog.*

Calibri

Elementos gráficos y logotipos

En la página web, incorporaremos distintos elementos gráficos:

- Los botones para seleccionar el modo de controlar el autómata que se visualizará en la cabecera de la página.
- El botón para visualizar las estadísticas que mostraremos en el lateral derecho en ordenadores y en la parte inferior en dispositivos móviles.
- Dentro de cada modo, los botones grandes para controlar el autómata.
- El gráfico para identificar el paso donde se encuentra el autómata que se visualizará tras los botones de control del autómata.

El logotipo será incorporado en el pie de página.

Imágenes

Las únicas imágenes que habrá en la página serán las que se utilicen para los links que habrá en el pie de página.

El tamaño que tendrán esas imágenes será de : 50x50px

Propuestas y futuro

Todos los integrantes de la empresa hemos echado en falta una explicación más detallada del entorno de máquina virtual que se nos ha dado. Una explicación de las aplicaciones que se nos daban o una aclaración sobre cómo usar cada cosa. Hemos conseguido aprender a duras penas como usarla por nuestra cuenta, pero ese tiempo “perdido” en ello podría haber sido mejor aprovechado si hubiésemos tenido una explicación antes de ponernos a usarlo.

Afortunadamente, hemos logrado que el autómata se mueva sin problemas desde el entorno web.

Anexo

Hemos usado una librería online para usar un icono de carga en el botón origen.
La librería tiene la siguiente dirección:

[https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.
css](https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css)