



Una silla de tres patas

# Control Automata

## Documentación

Equipo 4

Fecha de publicación 10 de octubre del 2018

Versión 1.0

# Contenido

|   |           |
|---|-----------|
| <b>Introducción</b>                                 | <b>3</b>  |
| Sobre nosotros                                      | 3         |
| El reto   | 4         |
| Logros del reto                                     | 4         |
| <b>Inicio del proyecto</b>                          | <b>5</b>  |
| Planteamiento y Reunión ARI                         | 5         |
| Reunión del equipo y Diseño inicial                 | 5         |
| Herramientas  | 8         |
| <b>Desarrollo</b>                                   | <b>9</b>  |
| Diseño y desarrollo de la base del HTML             | 9         |
| Investigación del Simulador y TIA Portal de Siemens | 9         |
| Desarrollo JavaScript y JQuery                      | 12        |
| Revisión del diseño inicial                         | 12        |
| Desarrollo HTML y CSS                               | 13        |
| Detalle de las funcionalidades                      | 14        |
| <b>Propuestas para el futuro</b>                    | <b>15</b> |
| Tareas pendientes                                   | 15        |
| Mejoras futuras                                     | 15        |
| <b>Bibliografía</b>                                 | <b>15</b> |
| Documentar el proyecto                              | 15        |
| Información sobre las herramientas                  | 15        |

# Introducción

## Sobre nosotros

Somos Aitor Bringas Martínez, Jon Xu Jin y Sebastián Zawisza Abreu, miembros del equipo 4 del grupo 2GA-DAW de Egibide Arriaga.

Mediante contrato de equipo, formalizamos el equipo 4 y con denominación TRIPOD.

Documentamos el proceso completo en este documento del desarrollo de la aplicación web que maneja un autómata desarrollado por el equipo de ARI, para así dejar constancia de los procesos, detalles de las funciones, el uso de las herramientas y de los ensayos que nos ha permitido el desarrollo y el testeo de esta aplicación.

Nuestro repositorio Git:

<https://github.com/tripod-egibide/reto1>

También pueden visitarnos en:

<https://tripod-egibide.github.io/>



# TRIPOD

## El Reto

Nuestro reto es el siguiente:

Nuestro cliente dispone de una línea de producción en la que una cinta transportadora transporta el producto (latas de conserva) por distintas etapas. Las latas avanzan por la cinta y pasan por las siguientes fases: primero se apilan en grupos de 8, luego se empaquetan y se etiquetan para su posterior distribución. Para que os hagáis una idea, podéis visualizar el video que muestra uno de los pasos que realiza la máquina que acaban de adquirir para realizar estas tareas.

<https://www.youtube.com/watch?v=ubgwI3IOwEI>

El cliente tiene como objetivo llevar el control de esta máquina transportadora mediante una aplicación web, y es por ello que nos han contratado. La página deberá mostrar las distintas fases en las que se encuentran las latas. Un operario de la fábrica visualizará en todo momento la máquina y también el estado de la cinta transportadora en la página web, de forma que si algo falla en alguna de las etapas, podrá parar la máquina y llevarla al paso adecuado mediante la página desarrollada.

Se desea poder consultar ciertos resúmenes y estadísticas sobre como van las cosas durante la jornada.

## Logros del reto

- Desarrollo de las actividades propuestas en clase para adquirir los conocimientos necesarios para el desarrollo del reto.
- Uso de software de simulación.(PLCSim)
- Utilización del software de programación de un autómata. (Tia Portal)
- Desarrollo de una aplicación web cliente (HTML, CSS y Javascript)
- Valorar la importancia de la actividad emprendedora y la innovación.
- Ser capaz de analizar y expresar ideas de negocio y de iniciar su puesta en marcha.

# Inicio del proyecto

## Planteamiento y Reunión ARI

Realizamos una reunión de los miembros del equipo para desarrollar las dudas y preguntas posibles para la implementación de nuestro proyecto con el autómata de ARI.

Realizamos una reunión con el equipo de ARI y nos explicaron las funcionalidades que va a tener su autómata, y posteriormente planteamos nuestras dudas, consultas y preguntas desarrolladas y surgidas durante la entrevista con los integrantes del equipo de ARI.

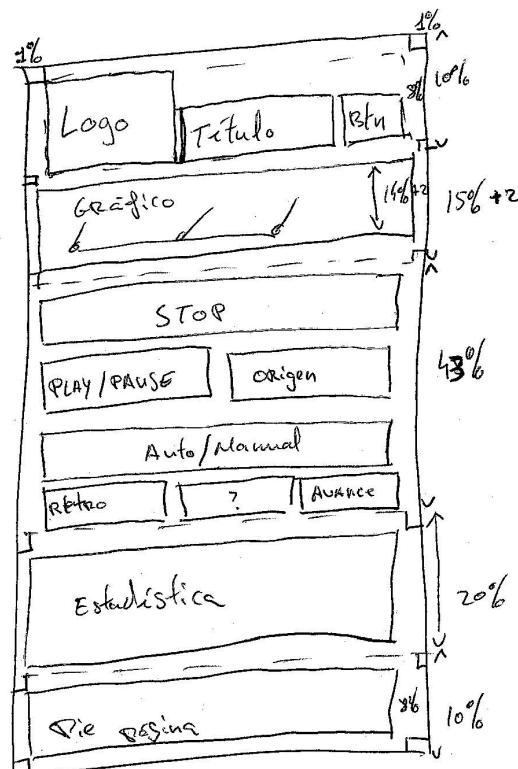
## Reunión del equipo y Diseño inicial

Tras la reunión con ARI con las dudas y preguntas resueltas, realizamos una reunión de los miembros del Equipo Tripod para por fin empezar a plantear cómo sería nuestra aplicación Web y diseñar un cuaderno de equipo para plasmar la planificación.

Tras una intensa lluvia de ideas sobre el diseño de las interfaces, optamos por una combinación de 3 ideas y plasmarlo en 1.

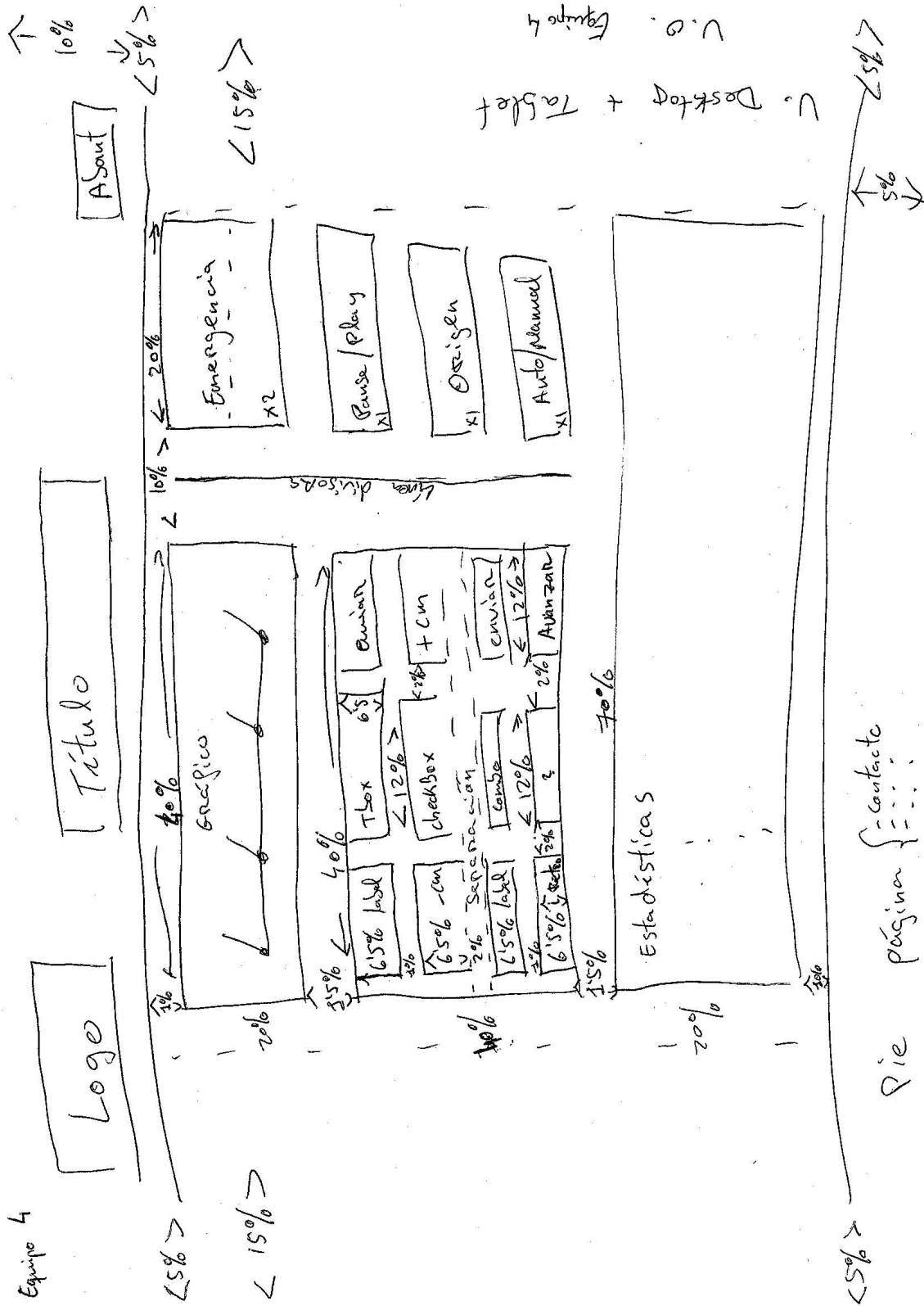
Realizamos varios Sketch intentando combinar las 3 ideas y el resultado fue lo siguiente:

Equipo 4



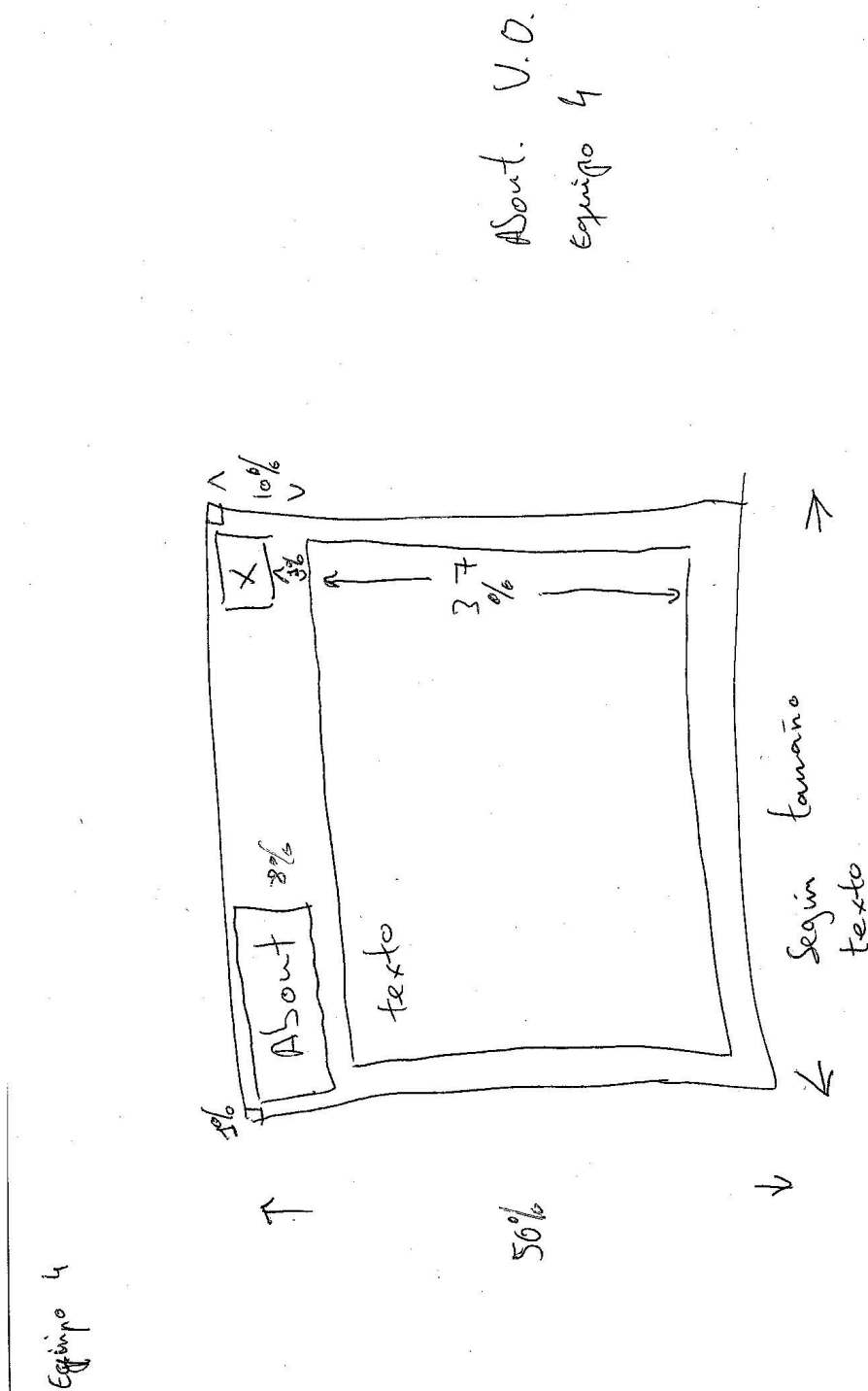
Versión SmartPhone y Tablet

Versión Móvil V.0



Versión común en todos los navegadores:

El about será una ventana flotante



Quedándonos todos conformes con el Sketch y llevarlo al desarrollo.

## Herramientas

Por decisión en conjunto de todos los miembros del equipo 4, se ha decidido emplear las siguientes herramientas para el diseño, desarrollo y ensayos de la aplicación web de control del autómata antes de su implementación al entorno de uso real:

| Herramientas empleadas en el proyecto |  |
|---------------------------------------|--|
| Papel y boli                          | Diseño inicial de los Sketch (se ha empleado boli por falta de lápiz).   |
| Trello                                | Planificar las tareas  |
| Hoja de seguimiento                   | Seguimiento de las tareas y las asistencias de los miembros.   |
| Atom                                  | Aplicación para el desarrollo de los códigos de JavaScript de manera colaborativa y en vivo, permitiendo editar en caliente todos los desarrolladores. |
| WebStorm                              | Aplicación para el desarrollo de los HTML y CSS.   |
| GitHub                                | Plataforma para el control de versiones de los códigos diseñados.  |
| NotePad++                             | Editor de textos y códigos en formato reducido, para corregir códigos sencillos para pruebas.  |
| Google Docs                           | Para el diseño de todas las documentaciones.   |
| Firefox<br>Google Chrome              | Navegador predeterminado para el uso y ensayo durante todo el proyecto.  |
| TIA Portal                            | Aplicación del autómata, que gestiona las funcionalidades y programación del autómata.   |
| Simulador Autómata Siemens            | Un simulador que simula el autómata como si estuviera físicamente.   |
| Pack iconos                           | Obtenidos en Material.io para el decorado de la aplicación web.  |
| Manual Animaciones                    | Manual detallado del W3School para el desarrollo de las animaciones.   |
| VirtualBox                            | Entorno virtual para las pruebas con una ova obtenido del Moodle.  |
| Jquery-3.3.1.min.js                   | Librería JQuery para el funcionamiento de las comunicaciones con el autómata.  |
| Google Fonts                          | Fuentes personalizadas del aplicativo.   |



# Desarrollo

## Diseño y desarrollo de la base del HTML

Iniciando esta segunda semana, empezamos a desarrollar la parte fija de nuestra aplicación web, una parte que no se modificará en ninguna de las páginas futuras, permitiendo mantener una estética fija de la cabecera y el pie de la página.

De la siguiente manera podemos añadir nuevas páginas sin necesidad de recargar la principal:

El Código debe estar dentro del <body>

```
<script type="text/javascript">

    $(()=> => $("#contenido").load("./main.html"));

</script>
```

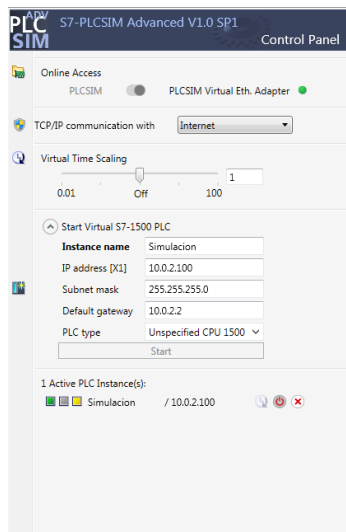
## Investigación del Simulador y TIA Portal de Siemens

Realizamos investigaciones sobre el manejo del simulador y TIA Portal que viene incluida en la .ova de la página de Moodle, ya que es imprescindible poseer conocimientos básicos/avanzados para poder realizar simulaciones y comprobar el correcto funcionamiento de los comandos que hemos diseñado y de su integración.

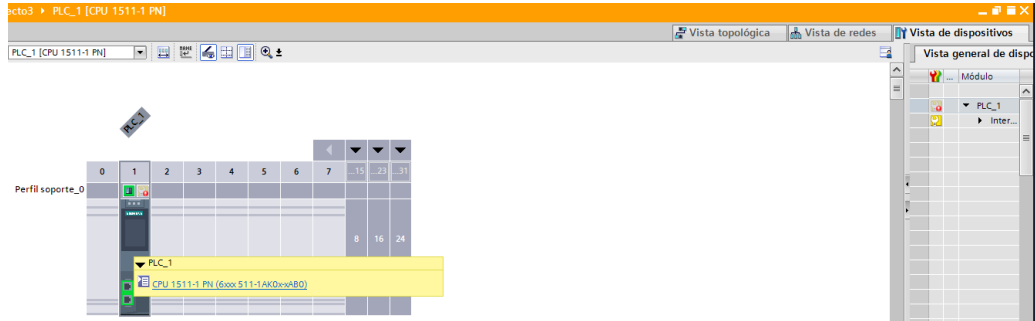
Pasamos varios días hasta entender el funcionamiento y otras opciones que posee dicha aplicación, así como la carga de nuestra aplicación en el simulador, el traspaso de la base de datos de un proyecto a otro, insertar nuevas variables, crear conexión entre aplicación TIA Portal y web, interpretar los errores y avisos que marca en el historial para solventarlos posteriormente.

Así como función que más utilizamos fue la de carga del aplicativo al simulador, aquí dejo una explicación sencilla y breve de cómo cargarlo al simulador:

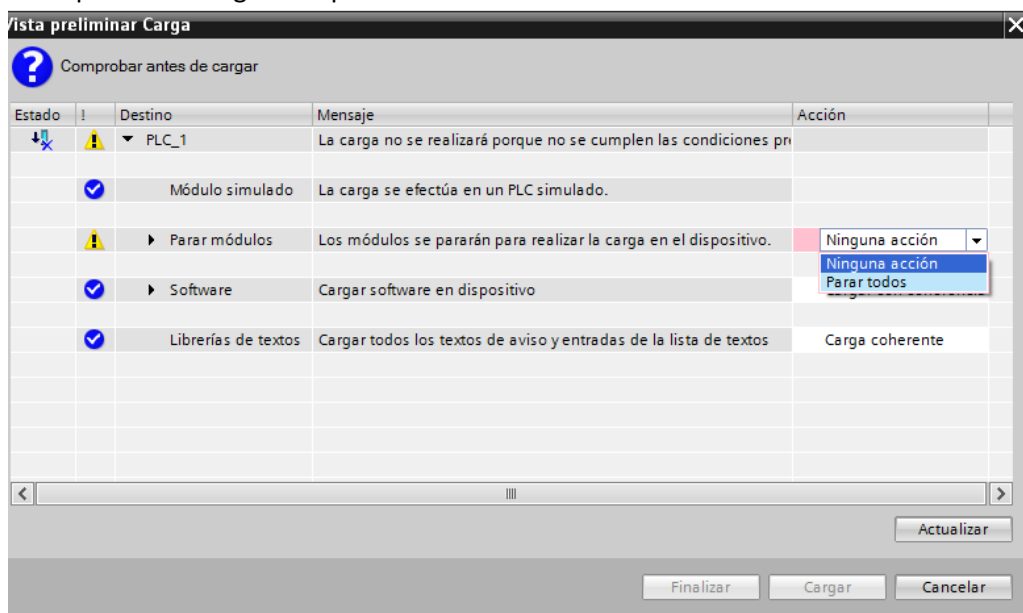
1. Abrir el emulador y aplicativo TIA Portal desde el escritorio de la maquina virtual(S7-PLCSIM Advanced V1.0 y TIA Portal V14) .
2. Hacer click derecho al ícono de notificaciones PLCSIM e introducir los datos como muestra en la imagen:



3. Abrir el proyecto desde el TIA Portal y posteriormente en Abrir la Vista del Proyecto para cargar el proyecto en pantalla.
4. En el área del proyecto (lateral izquierdo de la aplicación), seleccionar Dispositivo y redes > Doble click en el PLC\_1 > Doble click en la ranura del PLC que queremos manejar, y se abrirá las propiedades del PLC en la parte baja centro, como muestra en la imagen:

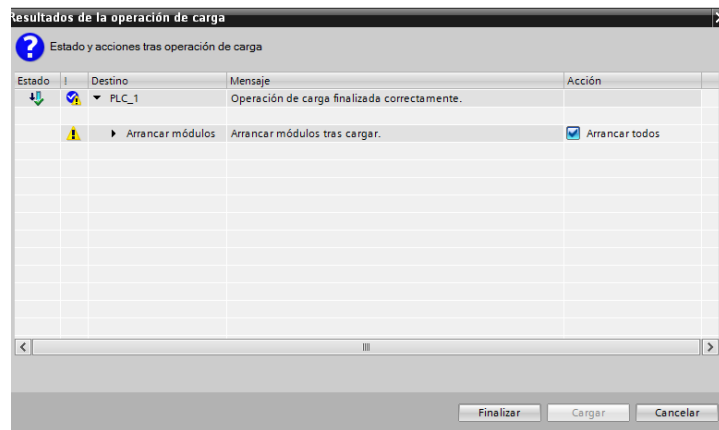


5. Seleccionamos la ruta del Directorio HTML, viene siendo la carpeta del proyecto HTML.
6. Seleccionamos Página de inicio HTML, viene siendo el index.html del proyecto ubicado en la misma carpeta del proyecto.
7. pulsamos en borrar bloques para vaciar los datos de la web de usuario que tiene TIA Portal.
8. Pulsamos en Generar bloques para que genere los bloques/códigos necesarios para el servidor web de TIA Portal.
9. Pulsamos derecho en el icono del PLC que anteriormente dimos doble click para que nos abra la pestaña de opciones.
10. Seleccionar “compilar” y seleccionar “Software (compilar todo)”.
11. Nuevamente el paso 9 y ahora en “Cargar en dispositivo” y seleccionar “Software (cargar todo)”.
12. Nos aparecerá la siguiente pantalla:



Damos doble click en “Ninguna acción” y seleccionamos “Parar todo”.

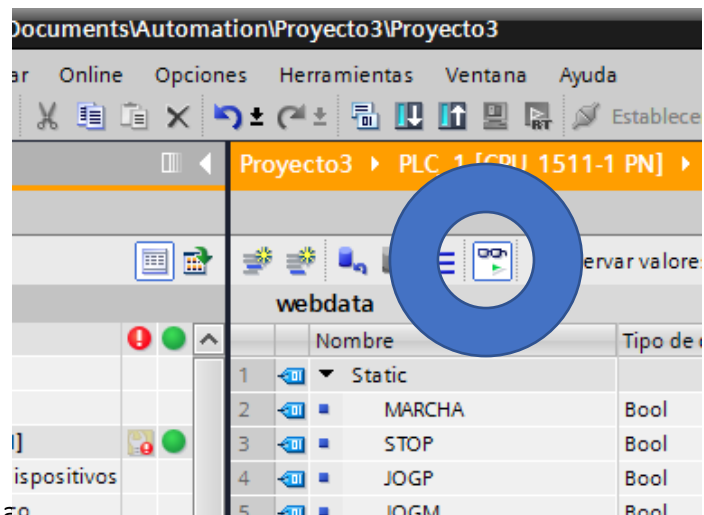
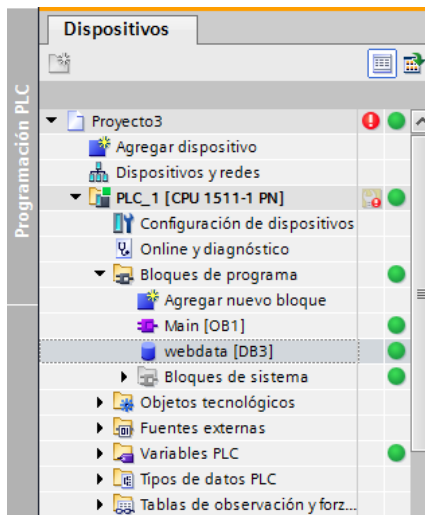
13. Pulsamos en “Cargar”, esperamos, seleccionamos “Arrancar todos” y pulsamos “Finalizar”.



14. Seleccionar del proyecto la base que queremos manejar(foto 1) y pulsamos sobre las gafas(foto 2) para cargar todo y arrancar todo.

Foto 1

Foto 2



16. Acceder a “Páginas de usuario” desde el menú lateral izquierdo y seleccionar “Página de inicio de la aplicación Autómata”.

Todo listo, se mostrará la página que se haya subido al simulador.

## Desarrollo JavaScript y JQuery

Empezamos a desarrollar los códigos en JavaScript y JQuery.

Así de primeras, tuvimos que acudir a San Google, para buscar información de cómo comunicarnos con JQuery, y pudimos dar con los siguientes códigos para enviar y leer información del simulador:

Enviar datos:

```
$( $.ajax({
    type: "POST",
    data: '"webdata".' + variable + ' = ' + valor
}));
```

Leer datos:

```
$.get("htm/TargetPosition.htm", function(result) {
    targetPosition = result.toString();
    $("#tar_pos").text(targetPosition);
});
```

Llamando desde una función y que lea desde otro html las variables que se desee

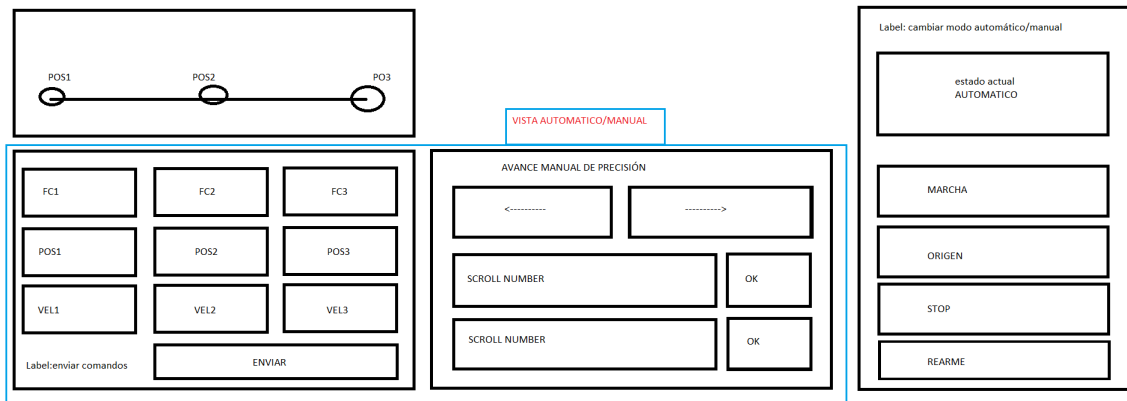
Lógicamente, hay más códigos, pero ya básicamente son de lógica para el manejo de las demás funciones, así como códigos claves, son los arriba indicados.

Aquí hemos plasmado una pequeñas muestras de cómo enviamos y obtenemos los datos, para comprender mejor y ver el código completo, recomendamos realizar una lectura del código fuente de la aplicación web, ya que está comentada y detallada de mejor manera.

## Revisión del diseño inicial

Realizar cambios durante un proyecto es algo bastante habitual si dependemos de otros equipos u organizaciones, en nuestro caso, hemos tenido que modificar parte del proyecto por los cambios que han realizado el equipo de ARI.

A continuación, adjuntamos el nuevo Sketch, adaptado a las nuevas variables que nos han facilitado:



## Desarrollo de HTML y CSS

El diseño de un html base como hemos indicado anteriormente, nos permite anidar otros html, como es en nuestro caso, main.html, en ello hemos reflejado la mayor parte de los códigos de la web del proyecto para llamar las funciones del JavaScript.

Básicamente, aquí declaramos todos los campos y botones necesarios sin importar apenas cómo distribuirlos de manera gráfica.

A continuación, adjuntamos una muestra del código html:

```
<script type="text/javascript" src="js/botones.js"></script>
<div class="row">
  <div class="col66">
    <div class="animacion">
      <div class="col100">
        <button onclick="go()">Go</button>
        <button onclick="stop()">Stop</button>
        <button onclick="fase12()">fase12</button>
        <button onclick="fase23()">fase23</button>
        <button onclick="fase31()">fase31</button>
        <div class="b" id="goAnimacion">o</div>
      </div>
    </div>
  </div>
</div>
```

Como pueden ver, únicamente se ha incluido el fichero JavaScript para poder llamar a las funciones y las declaraciones de los botones, sin necesidad de más códigos.

Dependiendo de lo que queramos mandar al simulador, necesitamos declarar las siguientes líneas:

```
<!-- AWP_In_Variable Name="webdata".JOGP' -->  
<!-- AWP_In_Variable Name="webdata".JOGM' -->  
<!-- AWP_In_Variable Name="webdata".REARME' -->  
<!-- AWP_In_Variable Name="webdata".InicioOrigen' -->
```

Entrecomillas debemos poner el nombre de la BD del PLC y después un punto, seguido de la variable que deseemos manejar desde nuestra web.

Se debe declarar, en nuestro caso, en el index.html, donde es la página padre.

## Detalle de las funcionalidades

Es habitual documentar durante el proceso del desarrollo de una aplicación.

Así como otros proyectos, pueden encontrar más información sobre las funcionalidades explorando nuestros proyectos.

```
//variables de lectura  
  
let targetPosition, currentPosition, alarm1, auto;  
  
//variables calculadas  
  
let posPorcentage, contadorCiclos = 0,  
ultimaAlarmaTiempo = null;
```

En este caso, todo el proyecto se puede explorar empleando en NotePad++, que lo pueden descargar desde el siguiente enlace:

<https://notepad-plus-plus.org/>

# Propuestas para el futuro

## Tareas pendientes

Al haber un plazo muy corto y problemas con el equipo de ARI, hubo parte de las funcionalidades que no se ha podido mejorar/corregir, tales como:

El entorno gráfico: la certeza de la flecha que señala la barra de progresos.

Los íconos: Una mejor adaptación de los iconos.

Estadísticas: Falta las florituras.

## Mejoras para el futuro

Así como plan de futuro, se podría corregir/implementar las tareas pendientes y además añadir nuevas funcionalidades, tales como:

Autenticación al acceder a la web.

Una BBDD para los datos de estadísticas.

Agregar accesibilidad para personas con discapacidad.

La versión final de la aplicación web que se ha entregado por el fin de plazos es funcional a nivel del enunciado, por lo que pensamos que cumple correctamente lo que nos exigen.

# Bibliografía

## Fuentes consultadas

Para el desarrollo de la aplicación de control del autómatas vía web, se ha consultados las siguientes fuentes:

Manejo del TIA Portal:

Equipo ARI

[https://w5.siemens.com/spain/web/es/industry/automatizacion/simatic/tia-portal/tia\\_portal/pages/tia-portal.aspx](https://w5.siemens.com/spain/web/es/industry/automatizacion/simatic/tia-portal/tia_portal/pages/tia-portal.aspx)

<https://www.youtube.com/watch?v=xFAYA0Q-N9A>

HTML y CSS:

<https://www.w3schools.com/>

JavaScript y JQuery:

<https://developer.mozilla.org/es/docs/Web>

Gracias a las aportaciones de dichas fuentes, pudimos llevar a cabo el desarrollo del proyecto reto 1, por lo que agradecemos de corazón a las fuentes y personas que nos han ayudado.

## Información sobre las herramientas

En este apartado, detallamos un poco de para qué se usan y dónde descargarlos.

Trello:

<https://trello.com>

Es un software de administración de proyectos con interfaz web para organizar proyectos, permitiéndonos registrar actividades del día a día, así como establecer fechas, etc

Hoja de seguimiento:

Básicamente es una hoja, física o virtual, para registrar las personas integrantes de un proyecto y así llevar un control de asistencia y puntualidad de los miembros.

Atom:

<https://atom.io/>

Es un editor de códigos de fuente de código abierto con soporte multiplataforma e integrado con Git para el control de versiones. Nos permiten trabajar en vivo con varios



desarrolladores a la vez, permitiéndonos así desarrollar y revisar a la vez, uno desarrolla y otro revisa u optimiza el código.

WebStorm:

<https://www.jetbrains.com/webstorm/>

Es un software diseñado para el desarrollo de páginas webs en formatos html, php, etc, incluyendo el css y muchos más. Al ser un software diseñado para el diseño de webs, integra asistentes para las correcciones y sugerencias códigos, permitiéndonos reducir el tiempo de desarrollo y errores frecuentes de escritura de códigos.

GitHub:

<https://github.com/>

Es una plataforma online y gratuito que nos permite gestionar el control de versiones de las aplicaciones durante el desarrollo, así como crear ramas para trabajar con casi un sinfín de usuarios en un mismo proyecto.

NotePad++:

<https://notepad-plus-plus.org/>

Es una aplicación de código abierto de edición de códigos muy básico, con posibilidad de integrar pluggins que nos permiten facilitar la edición, comparación, etc. de códigos. En nuestro caso, lo hemos empleado para pequeñas correcciones, comparaciones y pruebas.

Google Docs:

<https://google.es/>

Ya conocemos esta funcionalidad de la gran empresa del internet. Su uso es completamente gratuito con previo registro, nos permiten crear documentos Word, Excel, presentaciones, etc. De manera fácil y cómo no, de manera cooperativa.

Firefox y Chrome browser:

<https://www.mozilla.org/es-ES/firefox/new/>

<https://www.google.com/chrome/>

Navegadores libres, sin costes y muy utilizados en todo el mundo. Hemos empleado estos 2 navegadores por ser los más utilizados, tanto en ordenadores, Tablet y SmartPhones, y así asegurarnos que nuestra aplicación sea compatible con navegadores diseñados con framework web de los más usados.

Iconos:

<https://material.io/>

Es una de las páginas con la más amplia en iconos para el diseño de webs, dándonos así un toque más atractiva a nuestras webs.

VirtualBox:

<https://www.virtualbox.org/>

Es un software para la creación de máquinas virtuales, donde nos permite simular un sistema operativo en un entorno seguro para realizar pruebas de aplicaciones en desarrollo sin arriesgar nuestro sistema de trabajo o invertir en nuevas máquinas.

JQuery:

<https://jquery.com/download/>

Donde podemos descargarnos las librerías necesarias para el manejo de códigos en JQuery. En nuestro caso, hemos empleado la versión 3.3.1.

Google Fonts:

<https://fonts.google.com/>

Aquí podemos descargar fuentes de manera gratuita por si no nos gustase las fuentes existentes dentro de las herramientas de desarrollo que empleamos.

FIN



TRIPOD