Contenido

[1. Autores del trabajo, planificación y entrega 2](#_Toc448254544)

[1.1 Autores 2](#_Toc448254545)

[1.2 Planificación 2](#_Toc448254546)

[1.3 Entrega 2](#_Toc448254547)

[2. Requisitos del prototipo a implementar 3](#_Toc448254548)

[2.1 Requisitos funcionales 3](#_Toc448254549)

[2.2 Otros requisitos 3](#_Toc448254550)

[3. Criterios de comparación en la implementación 4](#_Toc448254551)

[3.1 Criterio 1: Nombre del criterio 4](#_Toc448254552)

[3.2 Criterio 2: Nombre del criterio 4](#_Toc448254553)

[3.N Criterio N: Nombre del criterio 4](#_Toc448254554)

[4. Proyecto de implementación de un prototipo del sistema utilizando la tecnología A 5](#_Toc448254555)

[4.1 Documentación de diseño 5](#_Toc448254556)

[4.2 Documentación de construcción 5](#_Toc448254557)

[4.3 Documentación de pruebas 5](#_Toc448254558)

[4.4 Documentación de instalación 5](#_Toc448254559)

[4.5 Manual de usuario 5](#_Toc448254560)

[5. Proyecto de implementación de un prototipo del sistema utilizando la tecnología B 6](#_Toc448254561)

[5.1 Documentación de diseño 6](#_Toc448254562)

[5.2 Documentación de construcción 6](#_Toc448254563)

[5.3 Documentación de pruebas 6](#_Toc448254564)

[5.4 Documentación de instalación 6](#_Toc448254565)

[5.5 Manual de usuario 6](#_Toc448254566)

[6. Comparación de las dos implementaciones 7](#_Toc448254567)

[6.1 Evaluación de los criterios en la implementación usando la tecnología A 7](#_Toc448254568)

[6.2 Evaluación de los criterios en la implementación usando la tecnología B 7](#_Toc448254569)

[7. Comparación de la implementación de las tecnologías 8](#_Toc448254570)

[8. Conclusiones 10](#_Toc448254571)

# 1. Autores del trabajo, planificación y entrega

## 1.1 Autores

Carlos Espejo Martínez

Julio Alberto Jiménez de la Torre

Álvaro Herrera Morales

Álvaro Pérez Lietor

Pedro Lanza Arroyo

## 1.2 Planificación

<https://app.ganttpro.com/shared/token/4fa02de26af2e5729ccd6b898bb73fcc77538ab7078e52794ff5f4a800c41f79>

## 1.3 Entrega

## <https://github.com/CarlosEspejoM/TG3-del-grupo-T5>

# 2. Requisitos del prototipo a implementar

**2.1 Requisitos funcionales**

|  |  |
| --- | --- |
| **REQ.** | **DESCRIPCIÓN** |
| RF01 | Empezar a contar. |
| RF02 | Stop |
| RF03 | Lap (cuenta tiempo por vuelta) |
| RF04 | Reset |

**2.2 Otros requisitos**

|  |  |
| --- | --- |
| **REQ.** | **DESCRIPCIÓN** |
| R01 | Nos informa del tiempo por vuelta |

**3. Criterios de comparación en la implementación**

**3.1 Criterio 1: Tiempo de instalación**

Nombre del criterio: Tiempo de instalación de ambos sistemas.

Descripción: Minutos y segundos exactos que dura la instalación en una máquina de similares características.

Tipo de valor: Numérico (segundos).

**3.2 Criterio 2: Base de Datos**

Nombre del criterio: Facilidad de aprendizaje de base de datos

Descripción: Horas/minutos necesarias para el aprendizaje necesario para realizar el despliegue de la base de datos de la aplicación.

Tipo de valor: Numérico (Horas/minutos).

**3.3 Criterio 3: Líneas de código**

Nombre del criterio: Líneas de código

Descripción: Líneas de código usando programación optimizada que tiene la aplicación en ambos sistemas

Tipo de valor: Numérico (unidades).

**3.4 Criterio 4: Facilidad de aprendizaje**

Nombre del criterio: Facilidad de aprendizaje

Descripción: Horas/minutos necesarias para el aprendizaje necesario para realizar el despliegue de la aplicación

Tipo de valor: Numérico (Horas/minutos).

# 4. Proyecto de implementación de un prototipo del sistema utilizando la tecnología Angular.

## 4.1 Documentación de diseño

Nuestra aplicación es un cronometro, la cual tiene tres funciones, empezar a contar, parar de contar y LAP, que te guarda un parcial de un tiempo que tengas cronometrado.

Diagrama de interfaz:

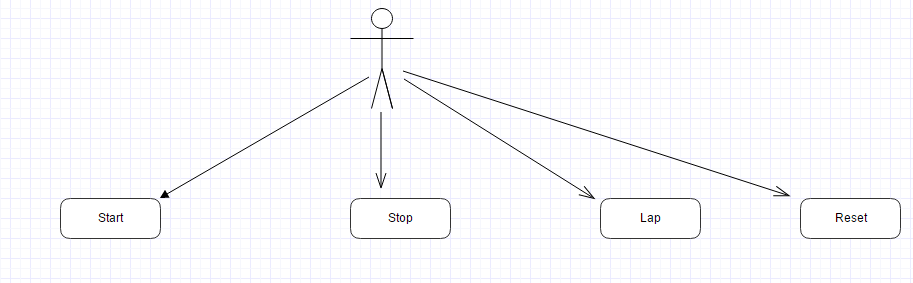
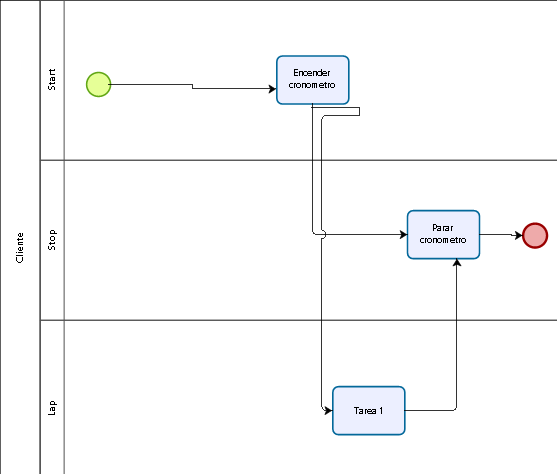


Diagrama BPMN:



## 4.2 Documentación de construcción

Para la construcción del prototipo en Angular, se ha trabajado alrededor de 8 archivos, siendo los principales cronometro ts, cronometro html, app.module.ts y app.componenet.ts.

**Cronometros.ts:**

import { Component } from '@angular/core';

import { Reloj } from './classes/reloj';

import { RelojFactory } from './classes/relojFactory';

@Component({

selector: 'cronometro',

templateUrl: './cronometro.html',

})

export class Cronometro{

public horas: number = 0;

public minutos: number = 0;

public segundos: number = 0;

public collection:Array<Reloj> = [];

public contador: any;

private relojFactory: RelojFactory;

constructor(relojFactory:RelojFactory){

this.relojFactory = relojFactory;

}

start(){

this.contador = setInterval(()=>{

this.segundos += 1;

if(this.segundos == 60){

this.segundos = 0;

this.minutos += 1;

if(this.minutos == 60){

this.minutos = 0;

this.horas += 1;

if(this.horas == 24){

this.reset();

}

}

}

}, 1)

}

reset(){

this.segundos = 0;

this.minutos = 0;

this.horas = 0;

}

lap(){

let obj = this.relojFactory.nuevoReloj(this.horas, this.minutos, this.segundos);

this.collection.push(obj);

}

stop(){

clearInterval(this.contador);

}

}

**Cronometro.html**

<div>

<table style="border:1px;">

<th>Horas</th><th>Minutos</th><th>Segundos</th>

<tr>

<td>{{horas}}</td><td>{{minutos}}</td><td>{{segundos}}</td>

</tr>

<br><br>

<tr>

<td><p \*ngFor="let cole of collection">{{cole.horas}}</p></td>

<td><p \*ngFor="let cole of collection">{{cole.minutos}}</p></td>

<td><p \*ngFor="let cole of collection">{{cole.segundos}}</p></td>

</tr>

</table>

</div>

<br><br>

<div>

<button (click)="start()">START</button>

<button (click)="reset()">RESET</button>

<button (click)="lap()">LAP</button>

<button (click)="stop()">STOP</button>

</div>

**app.module:**

import { BrowserModule } from '@angular/platform-browser';

import { NgModule } from '@angular/core';

import { FormsModule } from '@angular/forms';

import { HttpModule } from '@angular/http';

import { Cronometro } from './cronometro';

import { RelojFactory } from './classes/relojFactory';

import { AppComponent } from './app.component';

@NgModule({

declarations: [

AppComponent,

Cronometro,

],

imports: [

BrowserModule,

FormsModule,

HttpModule

],

providers: [ RelojFactory ],

bootstrap: [AppComponent]

})

export class AppModule { }

## 4.3 Casos de prueba fallidos.

Tuvimos problemas al principio, para inicializar el proyecto con sus correspondientes archivos, más tarde, profundizando en la materia, conseguimos utilizar la estructura que utiliza google para angular y solventamos los problemas

## 4.4 Documentación de instalación

- Primero es necesario instalar el programa Atom (Versión 1.16.0 64 bytes)

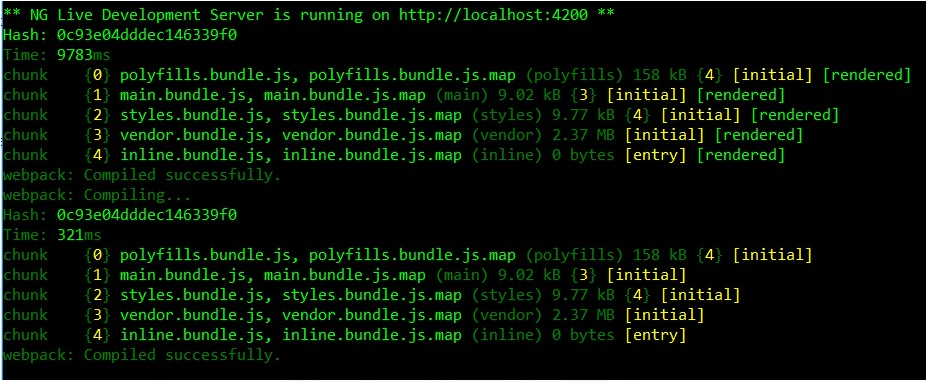
Para conseguir montar toda la estructura del proyecto, tuvimos que utilizar un constructor de paquetes para javascript, llamado npm.js



## 4.5 Manual de usuario

Para arrancar el cronometro debemos posicionaros en la carpeta principal del proyecto (concretamente donde se encuentra el archivo package.json), desde la terminal. A continuación, utilizamos el comando npm install para bajarnos las dependencias del proyecto (librerías). Solamente se debe ejecutar que vamos a utilizar el proyecto.

Para finalizar debemos ejecutar el comando npm start y acceder a la dirección que nos indica la terminal desde el navegador, en nuestro caso <http://localhost:4200>.



---> Start, que empieza el cronómetro

---> Stop, que detiene el cronómetro

---> Lap, cuenta tiempo por vuelta

---> Reset, que resetea el cronómetro a 00:00:00

# 5. Proyecto de implementación de un prototipo del sistema utilizando la tecnología B

## 5.1 Documentación de diseño

Diagrama interfaz:

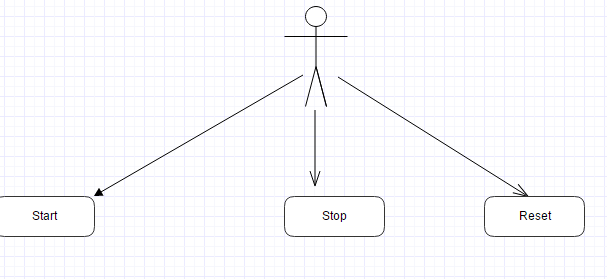
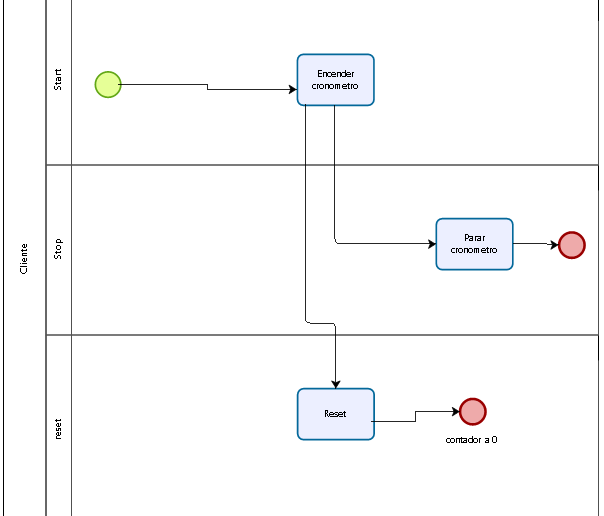


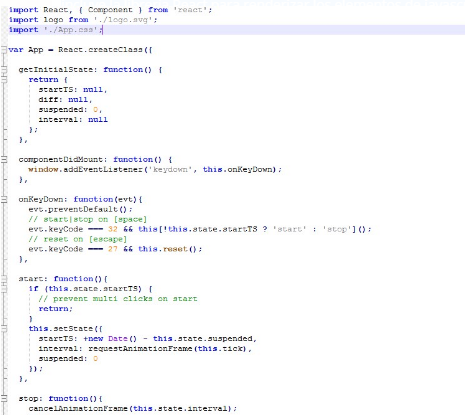
Diagrama BPMN:



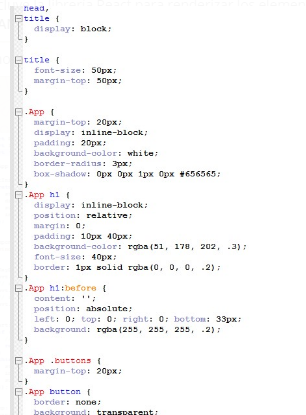
## 5.2 Documentación de construcción

Para la construcción del prototipo en React, se ha trabajado en 3 archivos principalmente: App.js, App.css e index.html.

En el App.js se encuentra el cuerpo de nuestra aplicación, es decir, las clases y los elementos que van a aparecer en nuestro cronómetro (Chrono, Botones, métodos..).



En el App.css se encuentra el código para dar formato a los elementos del App.js, como color, tamaño, fuente etc



Por último, en el index.html se encuentra el código html desde el cual se ejecutará nuestra aplicación, así como una llamada al método Render que incluye la librería React para renderizar los elementos de javascript



## 5.3 Documentación de pruebas

Casos de prueba fallidos.

-Hemos probado a llamar al metodo Render desde el index pero no funciona, no renderiza el cronómetro

-Hemos probado a llamar al método RenderDOM de react desde el index, pero tampoco renderiza el cronómetro

-Durante la creación del código, numerosos errores porque no teníamos las librerias de React importadas de manera correcta en el mismo.

Esto es lo único que hemos conseguido debido a que no somos capaces de solucionar el fallo del render:



## 5.4 Documentación de instalación

- Primero es necesario instalar el programa nodejs (Versión 4 o superior)

- Una vez instalado, desde la ventana de comandos se ejecutará el comando "npm install -g create-react-app", que instalará una librería para crear un proyecto nuevo en React de manera automática

- Después, desde la ventana de comandos se ejecutará el comando "create-react-app cronometro" que hará uso de la librería instalada anteriormente para crear nuestra aplicación, con nombre "cronometro".

- Una vez hecho esto, lo restante es la creación del código. Realizados estos pasos se creará en C:/Users/Usuario/cronometro una carpeta con el proyecto, y podremos trabajar sobre los mismos. Si se quiere coger nuestro proyecto, simplemente hay que descargar los archivos de nuestro GitHub y sustituir los de la carpeta mencionada anteriormente.

## 5.5 Manual de usuario

Para poder ejecutar el proyecto hay que tener instalado todo lo redactado en el punto 5.4, y después realizar los siguientes pasos:

- Abrir la ventana de comandos y acceder al directorio en el que se encuentra el proyecto (Por defecto, con escribir "cd cronometro" será suficiente)

- Desde la ventana de comandos, ejecutar "npm start". Esto abrirá una ventana en el navegador con nuestro proyecto.

- En caso de que lo hiciéramos funcionar, sería un cronómetro con 3 funcionalidades:

---> Start, que empieza el cronómetro

---> Stop, que detiene el cronómetro

---> Reset, que resetea el cronómetro a 00:00:00

# 6. Comparación de las dos implementaciones

## 6.1 Evaluación de los criterios en la implementación usando la tecnología Angular

| **CRITERIO** | **EVALUACIÓN** |
| --- | --- |
| Tiempo de instalación | 20 minutos |
| Base de datos | No requiere |
| Líneas de código | 5000 aprox |
| Facilidad de aprendizaje | 20 horas |

## 6.2 Evaluación de los criterios en la implementación usando la tecnología React

| **CRITERIO** | **EVALUACIÓN** |
| --- | --- |
| Tiempo de instalación | 25 minutos |
| Base de datos | No requiere |
| Líneas de código | 4800 aprox |
| Facilidad de aprendizaje | 14horas |

# 7. Comparación de la implementación de las tecnologías

| **CRITERIOS** | **TECNOLOGÍA angular** | **TECNOLOGÍA react** | **COMENTARIOS** |
| --- | --- | --- | --- |
| Tiempo de instalación | 20min | 25 min | Angular requiere de más tiempo. |
| Base de datos | No requiere | No requiere |  |
| Líneas de código | 5000 aprox | 4800 aprox | Como se puede observar en las react utiliza menos líneas de código y es más eficiente que angular. |
| Facilidad de aprendizaje | 20 horas | 14 horas | Nos resulta más fácil react. |

# 8. Conclusiones

Tras la realización de ambos prototipos del cronometros se llega a la conclusión de que ambos sistemas son ciertamente potentes y útiles, pero se han encontrado ciertas desventajas y ventajas que serán comentadas a continuación.

**AngularJS** se presenta como un framework total. Un framework donde podemos hacer todo lo necesario en una aplicación front compleja. Podemos incluir ciertas extensiones al framework, pero por lo general, el core de AngularJS nos va a permitir realizar un porcentaje altísimo de aplicaciones sin tener que incluir grandes cambios.

**ReactJS** en cambio se define como una librería. Una librería que tiene como fin un propósito muy definido: ReactJS solo quiere encargarse de gestionar la vista de tu aplicación. Si te acercas a esta librería, te cansarás de que Facebook te diga que si nuestra aplicación está diseñada como un MVC, ReactJS se encargará