

# TG2 – E4

Álvaro García Bayo

## Contenido

|  |                                      |
|--|--------------------------------------|
| 1. Autores del trabajo, planificación y entrega.....                   | 3                                    |
| 1.1 Autores.....   | 3                                    |
| 1.2 Planificación .....  | 3                                    |
| 1.3 Entrega.....   | 3                                    |
| 2. Descripción de los Frameworks a implementar .....                   | 4                                    |
| 2.1 Descripción de Angular .....                                       | 4                                    |
| 2.2 Descripción de React .....   | 4                                    |
| 3. Criterios de comparación (por categoría) .....                      | 5                                    |
| 3.1 Categoría 1: General.....  | 5                                    |
| 3.1.1 Criterio 1: Framework o librería.....                            | 5                                    |
| 3.1.2 Criterio 2: Tipo de Arquitectura .....                           | 5                                    |
| 3.1.3 Criterio 3: Plantillas y Temas.....                              | 5                                    |
| 3.2 Categoría 2: Rendimiento.....                                      | <b>¡Error! Marcador no definido.</b> |
| 3.2.1 Criterio 4: Lenguaje de Desarrollo.....                          | 6                                    |
| 3.2.2 Criterio 5: Herramientas de Desarrollo .....                     | 6                                    |
| 3.3 Categoría 3: Aprendizaje .....                                     | 6                                    |
| 3.3.1 Criterio 6: Curva de Aprendizaje .....                           | 6                                    |
| 3.4 Categoría 4: Utilidades .....                                      | 6                                    |
| 3.4.1 Criterio 7: Robustez .....                                       | 6                                    |
| 3.4.2 Criterio 8: Expresividad .....                                   | 6                                    |
| 3.4.3 Criterio 9: Entorno de Trabajo .....                             | 6                                    |
| 3.4.4 Criterio 10: Soporte.....  | 6                                    |
| 4. Evaluación de los criterios por tecnología .....                    | 7                                    |
| 4.1 Evaluación de los criterios para la implementación de Angular..... | 7                                    |
| 4.2 Evaluación de los criterios para la implementación de React .....  | 8                                    |
| 5. Comparación de las tecnologías: Angular vs React .....              | 9-10                                 |
| 6. Recomendaciones de elección según situaciones .....                 | 11                                   |
| 6.1 Situación 1: Crear una gran aplicación .....                       | 11                                   |
| 6.1.1 Descripción de la situación .....                                | 11                                   |
| 6.1.2 Recomendación de tecnología a utilizar.....                      | 11                                   |
| 6.2 Situación 2: Crear una aplicación con poco tráfico .....           | 12                                   |
| 6.2.1 Descripción de la situación .....                                | 12                                   |
| 6.2.2 Recomendación de tecnología a utilizar.....                      | 12                                   |
| 7. Anexo .....   | 12                                   |

# 1. Autores del trabajo, planificación y entrega

## 1.1 Autor

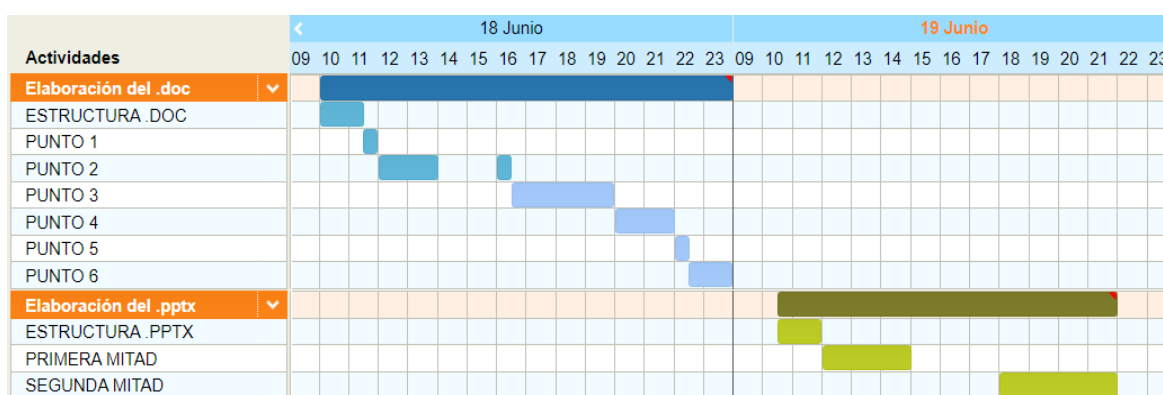
Álvaro García Bayo

## 1.2 Planificación

El diagrama de Gantt se ha realizado a través de la web Toms Planner, dedicada a la creación online de diagramas para la gestión de grupos de trabajo.:

<https://plan.tomsplanner.es/>

En este caso ha habido un solo participante. Se ha realizado la totalidad de la práctica en un total de 21h como se puede observar en el diagrama.



## 1.3 Entrega

Enlace (URL) al repositorio en GitHub donde está alojado el trabajo.

<https://github.com/alvaroogb/TG2-Alvaro-Garcia-Bayo>

En dicho repositorio se encuentran los siguientes archivos en la rama máster:

- Trabajo terminado: del trabajo terminado con el nombre TG2\_final.pdf
- Presentación del trabajo: TG2\_final.pptx
- Anexo: Documento Excel con todas las tablas creadas: TG2\_final.xlsx

Además, por motivos que así lo requieren se ha enviado un correo al profesor de la asignatura con los tres ficheros adjuntos.

## 2. Descripción de los Frameworks a comparar

En este trabajo se compararán dos de los Frameworks de Java Script más usados en el ámbito tecnológico. Se trata de Angular y React.

### 2.1 Descripción de Angular



Via: <https://dotnettutorials.net/wp-content/uploads/2018/08/c-users-pranaya-pictures-angular-png.png>

Angular es un framework de javascript de código abierto, mantenido por google, que se utiliza para crear y mantener aplicaciones web de una sola página. Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de Modelo Vista Controlador (MVC), en un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles.

Está construido en torno a la creencia de que la programación declarativa es la que debe utilizarse para generar interfaces de usuario y enlazar componentes de software, mientras que la programación imperativa es excelente para expresar la lógica de negocio

Su página web:

<https://angularjs.org/>

### 2.2 Descripción de React



Via: [https://cdn-images-1.medium.com/max/1600/1\\*3SVfBkNZl2f-sspiq59xcw.png](https://cdn-images-1.medium.com/max/1600/1*3SVfBkNZl2f-sspiq59xcw.png)

React (también llamada React.js o ReactJS) es una biblioteca Javascript de código abierto diseñada para crear interfaces de usuario con el objetivo de facilitar el desarrollo de aplicaciones en una sola página. Es mantenido por Facebook y la comunidad de software libre, han participado en el proyecto más de mil desarrolladores diferentes.

Esta librería intenta ayudar a los desarrolladores a construir aplicaciones que usan datos que cambian todo el tiempo. Su objetivo es ser sencillo, declarativo y fácil de combinar. Sólo maneja la interfaz de usuario en una aplicación.

Su página web:

<https://es.reactjs.org/>

### 3. Criterios de comparación en la implementación

Los criterios se organizarán en 3 categorías:

- Categoría 1: General
- Categoría 2: Rendimiento
- Categoría 3: Aprendizaje
- Categoría 4: Utilidades

#### 3.1 Categoría 1: General

##### 3.1.1 Criterio 1: Framework o librería

Nombre del criterio: Framework o librería.

Descripción: Se describe si el tipo de tecnología es un framework o una librería.

Tipo de valor: Texto libre.

##### 3.1.2 Criterio 2: Tipo de Arquitectura

Nombre del criterio: Tipo de Arquitectura.

Descripción: Se describe si el tipo de tecnología es un framework o una librería.

Tipo de valor: Texto libre.

##### 3.1.3 Criterio 3: Plantillas y Temas

Nombre del criterio: Plantillas y Temas

Descripción: Plantillas y temas disponibles para cada tecnología.

Tipo de valor: Texto Libre

#### 3.2 Categoría 2: Rendimiento

##### 3.2.1 Criterio 4: Lenguaje de Desarrollo

Nombre del criterio: Lenguaje de Desarrollo

Descripción: Tipo de lenguaje de programación que emplean.

Tipo de valor: Texto libre.

### **3.2.2 Criterio 5: Herramientas de Desarrollo**

Nombre del criterio: Herramientas de Desarrollo.

Descripción: Distintos tipos de herramientas necesarias para su uso.

Tipo de valor: Texto libre.

## **3.3 Categoría 3: Aprendizaje**

### **3.3.1 Criterio 6: Curva de Aprendizaje**

Nombre del criterio: Curva de Aprendizaje

Descripción: Dificultad para dominar ambas tecnologías.

Tipo de valor: Escala (1-5)

## **3.4 Categoría 4: Utilidades**

### **3.4.1 Criterio 7: Expresividad**

Nombre del criterio: Expresividad

Descripción: Facilidad para mantener el código de la aplicación.

Tipo de valor: Escala (1-5)

### **3.4.2 Criterio 8: Robustez**

Nombre del criterio: Robustez

Descripción: Capacidad de hacer aplicaciones a gran escala.

Tipo de valor: Booleano (SI/No)

### **3.4.3 Criterio 9: Entorno de Trabajo**

Nombre del criterio: Entornos de Trabajo

Descripción: Entornos necesarios para usar estas aplicaciones.

Tipo de valor: Texto Libre

### **3.4.4 Criterio 10: Soporte**

Nombre del criterio: Soporte

Descripción: Empresa que da soporte a cada tecnología.

Tipo de valor: Texto Libre

## 4. Evaluación de los criterios por tecnología

### 4.1 Evaluación de los criterios para Angular:

En la siguiente tabla se muestra de forma resumida la evaluación de esta tecnología:

| CATEGORÍA                | CRITERIO                               | EVALUACIÓN                    |
|--------------------------|--|-------------------------------|
| Categoría 1: General     | Criterio 1: Framework o Librería       | Framework                     |
|                          | Criterio 2: Tipo de Arquitectura       | Arquitectura básica MVC       |
|                          | Criterio 3: Plantillas y Temas         | HTML (Traslada de JS a HTML)  |
| Categoría 2: Rendimiento | Criterio 4: Lenguaje de Desarrollo     | Type Script                   |
|                          | Criterio 5: Herramientas de Desarrollo | Type Script                   |
| Categoría 3: Aprendizaje | Criterio 6: Curva de Aprendizaje       | (4-5)                         |
| Categoría 4: Utilidades  | Criterio 7: Robustez                   | SI                            |
|                          | Criterio 8: Expresividad               | (2-5)                         |
|                          | Criterio 9: Entorno de Trabajo         | Complejo, muchas herramientas |
|                          | Criterio 10: Soporte                   | Google                        |

- Criterio 1: Framework o librería: Angular es un Framework, es decir es una solución todo en uno que dispone de todas las herramientas necesarias para llevar a cabo una aplicación.

- Criterio 2: Tipo de Arquitectura: Angular maneja una arquitectura básica MVC (Modelo Vista Controlador), disponemos de Componentes para la vista, Enrutador para la capa de control y servicios para la capa de backend. El paradigma usado es orientado a componentes.

- Criterio 3: Plantillas y Temas: Angular centra sus plantillas en HTML, es decir escribimos cierta lógica en el HTML, trasladando javascript a HTML, esto implica que se mantiene un Html y javascript por componente lo cual da más claridad.

- Criterio 4: Lenguaje de Desarrollo: Angular utiliza TypeScript, que ofrece extensiones de lenguaje que están diseñados para hacer escritura en JavaScript más fácil, y asocia la información de tipo con entidades de JavaScript para hacer cumplir la comprobación de tipos y mejorar el flujo de trabajo de desarrollo. (La sintaxis de este lenguaje es muy parecida a java o .net).

- Criterio 5: Herramientas de Desarrollo: En Angular todos los frameworks se crean obligatoriamente usando Typescript, un lenguaje que mejora considerablemente la experiencia de desarrollo de aplicaciones JavaScript como las de Angular.

- Criterio 6: Curva de Aprendizaje: Angular es una tecnología que cuesta más tiempo dominar con respecto a otras.

- Criterio 7: Expresividad: Angular dispone del inyector de dependencias o el data binding, Éstos pueden ser difíciles de comprender y convertirse en un problema cuando se trata de encontrar un bug.

- Criterio 8: Robustez: Angular lo usa la propia Google además de: HBO, Nike, Sony, General Motors etc. Es un framework concebido para esta clase de tareas, aplicaciones de gran escala y bases de código grandes.

- Criterio 9: Entorno de Trabajo: Angular es un entorno de trabajo muy pesado ya que está pensado para desarrollar aplicaciones con muy alto tráfico.

- Criterio 10: Soporte: Angular es soportado por Google.

## 4.2 Evaluación de los criterios para React:

En la siguiente tabla se muestra de forma resumida la evaluación de esta tecnología:

| CATEGORÍA                | CRITERIO                               | EVALUACIÓN                        |
|--------------------------|--|-----------------------------------|
| Categoría 1: General     | Criterio 1: Framework o Librería       | Librería                          |
|                          | Criterio 2: Tipo de Arquitectura       | Arquitectura Flux (Similar a MVC) |
|                          | Criterio 3: Plantillas y Temas         | Javascript (Traslada HTML a JS)   |
| Categoría 2: Rendimiento | Criterio 4: Lenguaje de Desarrollo     | JSX                               |
|                          | Criterio 5: Herramientas de Desarrollo | React Developer Tools             |
| Categoría 3: Aprendizaje | Criterio 6: Curva de Aprendizaje       | (2-5)                             |
| Categoría 4: Utilidades  | Criterio 7: Robustez                   | SI                                |
|                          | Criterio 8: Expresividad               | (4-5)                             |
|                          | Criterio 9: Entorno de Trabajo         | Sencillo, pocas herramientas      |
|                          | Criterio 10: Soporte                   | FaceBook                          |

- Criterio 1: Framework o librería: React es una librería que solo se encarga de la vista, lo que implica que el resto de herramientas para hacer una aplicación hay que definirlas. Esto logra una mayor flexibilidad, ya que al poder definir todos los componentes por separado se pueden cambiar librerías que se desarrollen en un futuro y mejoren la aplicación sin grandes alteraciones.

- Criterio 2: Tipo de Arquitectura: React maneja una arquitectura Llamada Flux, que es similar a MVC ya que también contiene su modelo, vista y controladores pero está pensada en un flujo de datos unidireccional. Los datos viajan desde la vista por medio de acciones y llegan a un Store desde el cual se actualizará la vista de nuevo.

- Criterio 3: Plantillas y Temas: En React toda la lógica y vista permanece en javascript, es decir, se traslada HTML a Javascript. Esto genera un código más centralizado. La detección de errores se genera en la compilación de la plantilla, aportando información acerca del error y la línea que provoca el error.

- Criterio 4: Lenguaje de Desarrollo: React utiliza JSX que permite incrustar etiquetas XML/HTML en el archivo de JavaScript, esto implica que JSX es una extensión de sintaxis para JavaScript. Se tiene que usar un compilador como Babel, que recoge nuestro código JSX y lo compila para generar JavaScript que los navegadores puedan entender.

- Criterio 5: Herramientas de Desarrollo: React tiene herramientas como las React Developer Tools, una extensión para el navegador que, precisamente, extiende el soporte para depurar aplicaciones creadas con React. Añadir que como las React Developer Tools, existen similares para Angular, es Facebook mismo quien crea esta herramienta, no como en el caso de Angular que no es Google quien las desarrolla.

- Criterio 6: Curva de Aprendizaje: El mismo equipo que desarrolla el framework trabaja en las librerías más populares por lo que aprender esta tecnología es mucho más fácil.



- Criterio 7: Expresividad: React se distingue por ser un “framework” mucho más expresivo, ya que tiene componentes funcionales y no hay data binding porque el flujo de información es muy claro y detectar un posible error más fácil.

- Criterio 8: Robustez: React lo usa Facebook además de Instagram, Whatsapp, Netflix...

- Criterio 9: Entorno de Trabajo: Para trabajar con React se deben usar algunas herramientas como JSX. Es muy simple.

- Criterio 10: Soporte: React es soportado por FaceBook

## 5. Comparación de las tecnologías: Angular vs React

En este apartado se puede observar una tabla-resumen que compara a ambas tecnologías. Los comentarios están cortados en otra tabla distinta:

| CRITERIO                           | Angular  | React   |
|------------------------------------|--|---|
| Criterio 1: Framework o Librería   | Es un Framework  | Es una Librería   |
| Criterio 2: Tipo de Arquitectura   | Usa Arquitectura básica MVC (Modelo Vista Controlador)                   | Usa Arquitectura Flux (Similar a MVC)   |
| Criterio 3: Plantillas y Temas     | Angular centra sus plantillas en HTML                                    | En React toda la lógica y vista permanece en javascript                             |
| Criterio 4: Lenguaje de Desarrollo | Usa obligatoriamente Type Script   | Usa JSX y/u opcionalmente Type Script   |
| Criterio 6: Curva de Aprendizaje   | Curva de Prendizaje Alta   | Curva de Aprendizaje Baja   |
| Criterio 7: Robustez               | Creada para hacer aplicaciones de gran escala y bases de código grandes. | Creada para hacer aplicaciones de gran escala y facilidad para hacer pequeñas apps. |
| Criterio 8: Expresividad           | Versiones obsoletas incompatibles con nuevas versiones                   | Versiones antiguas más facil de actualizar a las nuevas                             |
| Criterio 9: Entorno de Trabajo     | Framework muy potente con muchas herramientas                            | Sencillo, pocas herramientas con posibilidad de añadir más                          |
| Criterio 10: Soporte               | Creada por Google  | Creada por FaceBook   |

| COMENTARIOS   |
|---|
| Angular es un Framework (Todas las herramientas posibles) frente a React que es una librería y por lo tanto más sencilla desde un principio |
| Angular usa arquitectura básica MVC, paradigma usado orientado a componentes, React utiliza Flux, una arquitectura parecida                 |
| Angular traslada de JS a HTML mientras que React traslada el lenguaje HTML a JS   |
| En Angular se usa obligatoriamente Type Script mientras que en React cabe la posibilidad der elegir entre JSX o Type Script                 |
| Para un usuario sin contacto con estas tecnologías es más difícil iniciarse en Angular que en React   |
| Ámbas son tecnologías creadas para hacer grandes proyectos pero React es más eficiente a la hora de hacer un proyecto pequeño               |
| React tiene una mejor comunicación entre las diferentes versiones que tiene. No es el caso de Angular                                       |
| Al ser un Framework, Angular dispone de multitud de herramientas base, React es más sencillo desde un principio                             |
| Angular fue creado y es soportado por Google mientras que en el caso de React es Facebook quen creó y lo soporta                            |

Además de los comentarios se explica detalladamente las diferencias entre ambas tecnologías:

**Angular** es un framework robusto, incluye todo lo necesario para empezar una aplicación pero con poca flexibilidad, su curva de aprendizaje en general no es tan corta y puede ser un poco confuso al principio si no se tiene vista de lenguajes como java y .net por la sintaxis que maneja, Dispone de un buen soporte gracias a sus creadores (Google) y la comunidad en general.

**React** es una librería ligera y flexible, pero deja la tarea de encontrar todas las partes (librerías) necesarias para generar aplicaciones. Esto en algún momento podría ser un inconveniente si se desprecian las librerías complementarias usadas, aunque en general es raro este caso. Debido a que su lenguaje es una extensión de javascript generalmente la curva de aprendizaje no es tan alta. Mantiene un buen soporte por FaceBook y por la comunidad en general.

Llegados a este punto podríamos haber llegado a la pregunta: **¿Entonces cuál de estas tecnologías es mejor?**.

Realmente nos será imposible hallar una respuesta correcta basándonos solo en las características de cada una de ellas; la respuesta correcta es, considerando las ventajas y desventajas de cada librería, se debería usar la que más se adecúe al proyecto que queremos llevar a cabo.

En el siguiente punto veremos dos situaciones distintas en las que debemos decantarnos por una de estas tecnologías, valorando por qué es mejor elegir una u otra.

La tabla completa se puede encontrar en el documento xlx adjuntado.

## 6. Recomendaciones

En este último punto se plantearán dos situaciones ficticias para resolver según los criterios establecidos cuál de estas dos tecnologías sería la más adecuada.

### 6.1 Situación 1: Proyecto ECO Work: la aplicación de la Práctica E2

#### 6.1.1 Descripción de la Situación

ECO Work es una empresa emergente que tiene todo listo para iniciar los primeros pasos de desarrollo y lanzamiento de su producto/servicio.

Se desea crear una aplicación SPA usando un framework de Java Script.

Se espera llegar a muchas personas por lo tanto el tráfico va a ser elevado.

#### 6.1.2 Recomendación de tecnología a utilizar

Tanto React como Angular son buenas opciones para lograr el propósito de cualquiera de las dos situaciones que veremos, es decir, realmente ambas situaciones se podrían llevar a cabo con cualquiera de las dos tecnologías a comparar.

La elección entre una y otra depende de cada usuario y de la envergadura del proyecto.

En la siguiente tabla se observan los criterios que a mi parecer se deben comparar a la hora de elegir una tecnología frente a otra en esta situación. Las ventajas y desventajas que se muestran en cada tecnología son relativas ya que dependen del criterio personal de cada uno.

| CRITERIOS RELEVANTES PARA LA DECSIÓN | VENTAJAS ÁNGULAR                                       | VENTAJAS REACT   |
|--------------------------------------|--|--|
| Framework o Librería                 | Framework  | Es una Librería  |
| Tipo de Arquitectura                 | MVC (Modelo Vista Controlador)                         | Flux   |
| Lenguaje de Desarrollo               | Type Script (Obligatorio)                              | JSX y opcionalmente Type Script                            |
| Curva de Aprendizaje                 | Curva de Prendizaje Alta                               | Curva de Aprendizaje Baja                                  |
| Robustez                             | Creada para hacer aplicaciones de gran escala          | Facilidad para hacer pequeñas apps.                        |
| Expresividad                         | Versiones obsoletas incompatibles con nuevas versiones | Versiones antiguas más facil de actualizar a las nuevas    |
| Entorno de Trabajo                   | Framework muy potente con muchas herramientas          | Sencillo, pocas herramientas con posibilidad de añadir más |

En mi opinión, en este caso usaría Angular debido a que es un Framework y no una librería, lo que le da más robustez. Utiliza una arquitectura MVC, mucho más definida que Flux en React.

Un punto muy importante es que se usa obligatoriamente un mismo lenguaje de desarrollo, Typescript; si bien esto podría ser una desventaja en otro proyecto, pero en este caso, al ser una aplicación de gran escala la empresa trabajará con un mismo lenguaje lo que es una gran ventaja para un desarrollo más ágil.

Angular está concebido para el desarrollo de aplicaciones a gran escala, además posee multitud de herramientas, siendo más “absurdo” crear una pequeña aplicación con esta tecnología ya que no se aprovecharía todo su potencial.

## 6.2 Situación 2: App para una comunidad de vecinos.

### 6.2.1 Descripción de la situación

Se desea crear una aplicación multiplataforma para gestionar las cuentas, actas, informes, etc de una comunidad de vecinos.

En la comunidad son 100 personas y no se espera llegar a nadie más.

### 6.2.2 Recomendación de tecnología a utilizar

Como se venía mencionando en la anterior situación, ambas tecnologías son viables para crear cualquiera de estos proyectos. En este caso es un proyecto mucho más sencillo por lo que prima la facilidad de uso, de aprendizaje y sencillez del framework antes que su potencia.

Dicho esto, se muestra la tabla comparativa con respecto a esta situación:

| CRITERIOS RELEVANTES PARA LA DECSIÓN | VENTAJAS ÁNGULAR                                       | VENTAJAS REACT   |
|--------------------------------------|--|--|
| Framework o Librería                 | Framework  | Es una Librería  |
| Tipo de Arquitectura                 | MVC (Modelo Vista Controlador)                         | Flux   |
| Lenguaje de Desarrollo               | Type Script (Obligatorio)                              | JSX y opcionalmente Type Script                            |
| Curva de Aprendizaje                 | Curva de Prendizaje Alta                               | Curva de Aprendizaje Baja                                  |
| Robustez                             | Creada para hacer aplicaciones de gran escala          | Facilidad para hacer pequeñas apps.                        |
| Expresividad                         | Versiones obsoletas incompatibles con nuevas versiones | Versiones antiguas más facil de actualizar a las nuevas    |
| Entorno de Trabajo                   | Framework muy potente con muchas herramientas          | Sencillo, pocas herramientas con posibilidad de añadir más |

React es una librería, no un framework por lo tanto, como se mencionó en el punto 4.2 (Criterio 1) se logra una mayor flexibilidad, ya que al poder definir todos los componentes por separado se pueden cambiar librerías que se desarrollen en un futuro y mejoren la aplicación sin grandes alteraciones.

Respecto al lenguaje de desarrollo, en este caso no es importante que todos los miembros de la empresa usen el mismo. Al ser un proyecto pequeño es una gran ventaja el poder decidir qué tipo de lenguaje de desarrollo usar y no solo ceñirnos a Typescript.

Generalmente este tipo de aplicaciones se llevan a cabo por pequeñas empresas o incluso por una sola persona. Esto beneficia a React debido a que tiene una curva de aprendizaje bastante baja.

Por último, React fue concebida para el desarrollo de grandes plataformas, pero no olvidemos que al ser una librería es mucho más simple que Angular, usando además menos herramientas necesarias para arrancar una proyecto. Esto nos da más facilidades para hacer pequeñas app's como la de esta situación.

Todo esto no quiere decir que debemos usar React para pequeños proyectos y Angular para proyectos más grandes. En cualquier caso, es más eficiente usar React para un proyecto pequeño que Angular. Por este motivo, para esta situación usaría React.

## 7 Anexo

Se adjunta junto a los documentos.doc y pptx una hoja Excel en la que se han realizado todas las tablas.