

**PROCEDIMIENTO PARA UTILIZAR EL FRAMEWORK ANGULAR EN LA  
CREACIÓN DE APLICACIONES WEB MODERNAS SPA.**

**Autor**

**DANNY BISMARCK CASTRO ISIDRO**

**UNIVERSIDAD DE PAMPLONA**

**FACULTAD DE INGENIERÍAS Y ARQUITECTURA**

**DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA, SISTEMAS Y  
TELECOMUNICACIONES**

**INGENIERÍA DE SISTEMAS**

**PAMPLONA, COLOMBIA**

**2018**

**PROCEDIMIENTO PARA UTILIZAR EL FRAMEWORK ANGULAR EN LA  
CREACIÓN DE APLICACIONES WEB MODERNAS SPA.**

**Autor**

**DANNY BISMARCK CASTRO ISIDRO**

**TRABAJO PRESENTADO PARA OPTAR POR EL TITULO  
DE INGENIERO DE SISTEMAS**

**Director**

**EDGAR ALEXIS ALBORNOZ ESPINEL**

**Magister en Ciencias de la Computación**

**UNIVERSIDAD DE PAMPLONA**

**FACULTAD DE INGENIERÍAS Y ARQUITECTURA**

**DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA, SISTEMAS Y  
TELECOMUNICACIONES**

**INGENIERÍA DE SISTEMAS**

**PAMPLONA, COLOMBIA**

**2018**

**Procedimiento para utilizar el framework Angular en la creación de aplicaciones web modernas SPA.**

## **RESUMEN**

Cuando un desarrollador de software se da a la tarea de incursionar en un nuevo framework, se enfrenta a una curva de aprendizaje elevada, ya que se hace necesario conocer sus características, instalación, sintaxis, requerimientos y demás actividades necesarias para el desarrollo, invirtiendo mucho tiempo en ello ya que es necesario buscar en distintas fuentes. Por ello surge la necesidad de generar un procedimiento en el cual la curva de aprendizaje disminuya notablemente agilizando el proceso de desarrollo enfocándose solo en las tareas que realmente conduzcan al desarrollo del aplicativo.

La propuesta en este proyecto es diseñar un procedimiento en el cual el lector pueda aprender a utilizar el framework Angular en la creación de aplicaciones web modernas del tipo SPA, abordando las etapas más importantes y necesarias para el desarrollo, en este framework.

## **ABSTRACT**

When a software developer takes on the task of entering a new framework, he faces a high learning curve, since it is necessary to know its characteristics, installation, syntax, requirements and other activities necessary for development, investing a lot of time in it since it is necessary to search in different sources. Therefore, there is a need to generate a procedure in which the learning curve decreases significantly by streamlining the development process focusing only on the tasks that really lead to the development of the application.

The proposal in this project is to design a procedure in which the reader can learn to use the Angular framework in the creation of modern web applications of the SPA type, addressing the most important and necessary stages for development, in this framework.

## TABLA DE CONTENIDO

### Contenido

1. INTRODUCCIÓN.....	1
1.1 PLANTEAMIENTO DEL PROBLEMA.....	1
1.2 JUSTIFICACIÓN .....	2
1.3 INDICADORES DE USO DE ANGULAR .....	2
1.3.1 USO DE ANGULAR .....	2
1.3.2 SITIOS WEB QUE USAN ANGULAR .....	4
1.4 OBJETIVOS.....	7
1.4.1 OBJETIVO GENERAL.....	7
1.4.2 OBJETIVOS ESPECIFICOS.....	7
1.5 ACOTACIONES .....	7
2 MARCO TEÓRICO Y ESTADO DEL ARTE.....	8
2.1 MARCO TEORICO.....	8
2.1.1 FRAMEWORK.....	8
2.1.2 SINGLE PAGE APPLICATION .....	9
2.1.3 ANGULAR .....	10
2.1.4 FIREBASE.....	12
2.1.5 TYPESCRIPT .....	13
2.1.6 TIPOS DE DESARROLLADORES.....	14
2.2 ESTADO DEL ARTE.....	16
2.2.1 TESIS RELACIONADAS.....	16
2.2.2 ANÁLISIS .....	25
3. DISEÑO Y VALIDACIÓN DEL PROCEDIMIENTO PARA UTILIZAR EL FRAMEWORK ANGULAR EN LA CREACIÓN DE APLICACIONES WEB SPA .....	27
3.1 PROCEDIMIENTO ANGULARDASPA.....	27

3.1.1	INSTALACIÓN Y CONFIGURACIÓN DE ANGULAR.....	29
3.1.2	CONCEPTUALIZACIÓN DEL PROYECTO .....	35
3.1.3	IDENTIFICACIÓN DE REQUERIMIENTOS .....	38
3.1.4	CREACIÓN DE STORYBOARD .....	40
3.1.5	INSTALACIÓN DE DEPENDENCIAS .....	46
3.1.6	CREACIÓN DE COMPONENTES .....	50
3.1.7	CREACIÓN DE RUTAS.....	57
3.1.8	CONFIGURACIÓN FIREBASE .....	61
3.1.9	GESTIÓN DE AUTORIZACIONES .....	70
3.1.10	REALIZACIÓN CRUD.....	76
3.1.11	PRUEBAS DE ACEPTACIÓN.....	82
3.1.12	ENTREGA DE PRODUCTO .....	84
3.2	VALIDACIÓN DEL PROCEDIMIENTO CON DESARROLLO DE UN PROTOTIPO	84
3.3	VALIDACIÓN DEL PROCEDIMIENTO POR EXPERTOS .....	95
3.3.1	EXPERTOS .....	96
3.3.2	CUESTIONARIO .....	96
3.3.3	RESULTADOS DE VALIDACIÓN .....	98
3.3.4	VALIDEZ DEL PROCEDIMIENTO .....	101
4.	CONCLUSIONES, RECOMENDACIONES Y TRABAJOS FUTUROS.....	102
4.1	CONCLUSIONES.....	102
4.2	RECOMENDACIONES .....	103
4.3	TRABAJOS FUTUROS .....	103
5.	BIBLIOGRAFIA.....	104
6.	ANEXOS .....	106
6.1	DIRECTIVAS EN ANGULAR.....	106
6.2	JUICIO DE EXPERTOS .....	107

## LISTAS DE TABLAS

Tabla 1 Herramientas Firebase.....	13
Tabla 2 Comparativo estado del arte .....	26
Tabla 3 Características de viabilidad de Angular .....	38
Tabla 4 Dependencias más utilizadas.....	48
Tabla 5 Data Binding .....	56
Tabla 6 Evaluación viabilidad proyecto .....	86
Tabla 7 Directivas en Angular .....	106

## LISTA DE ILUSTRACIONES

Figura 1 Pagina Vmware .....	5
Figura 2 Pagina Freelancer .....	5
Figura 3 Pagina Royal Caribbean .....	6
Figura 4 Pagina NBA .....	6
Figura 5 Procedimiento AngularDASPA.....	28
Figura 6 Node.js ( <a href="https://nodejs.org/es/">https://nodejs.org/es/</a> ) .....	30
Figura 7 Visual Studio Code .....	31
Figura 8 Angular CLIC .....	31
Figura 9 Estructura de archivos .....	33
Figura 10 Proyecto Angular .....	35
Figura 11 StoryBoard Cliente.....	41
Figura 12 StoryBoard Desarrollador – Dependencias .....	43
Figura 13 StoryBoard Desarrollador – Componentes.....	44
Figura 14 StoryBoard Desarrollador - Datos .....	45
Figura 15 Formato StoryBoard desarrollador .....	46
Figura 16 Página principal NPM .....	47
Figura 17 Incluir dependencias Angular .....	49
Figura 18 Estructura de un componente .....	51
Figura 19 Identificación de componentes.....	52
Figura 20 Creación de un componente .....	53
Figura 21 HTML de un componente.....	53
Figura 22 CSS de un componente .....	54
Figura 23 TypeScript de un componente .....	54
Figura 24 Inyección de un componente .....	55
Figura 25 Estructura de Routing en Angular .....	58
Figura 26 Rutas en Angular .....	58
Figura 27 Imports de Routing.....	59
Figura 28 Router-Outlet en Angular .....	59
Figura 29 Enviar parámetro en Routing. ....	60
Figura 30 Capturar parámetro en Routing.....	60
Figura 31 Creación de enlace con Angular .....	61

Figura 32 Pagina Firebase.....	62
Figura 33 Proyecto Firebase.....	63
Figura 34 Crear un proyecto en Firebase .....	63
Figura 35 Añadir un proyecto en Firebase .....	64
Figura 36 Configuración Firebase .....	64
Figura 37 Añadir Configuración a Angular .....	65
Figura 38 Incluir Firebase en module.ts .....	65
Figura 39 Inyección Firebase en module.ts.....	66
Figura 40 Gestión de usuarios Firebase .....	66
Figura 41 Método de inicio de sesión Firebase.....	67
Figura 42 Selección base de datos Firebase .....	68
Figura 43 Reglas de seguridad Firebase .....	68
Figura 44 Panel control base de datos Firebase.....	69
Figura 45 Permisos base de datos Firebase.....	69
Figura 46 Creación del service .....	70
Figura 47 Pruebas en Angular. Tomado de: < <a href="https://angular.io/guide/testing">https://angular.io/guide/testing</a> > .....	83
Figura 48 StoryBoard cliente - Bitacora trabajo social.....	88
Figura 49 StoryBoard dependencias.....	89
Figura 50 StoryBoard Componentes 1 .....	89
Figura 51 StoryBoard Componentes 2.....	90
Figura 52 StoryBoard Datos 1.....	90
Figura 53 StoryBoard Datos 2.....	90
Figura 54 Servicios utilizados de Firebase.....	92
Figura 55 Métodos inicio de sesión.....	93
Figura 56 Estructura base de datos Firebase .....	94
Figura 57 Carta aceptación bitácora trabajo social .....	95

## **1. INTRODUCCIÓN**

El desarrollo de software es un campo que cuenta con infinidad de Frameworks que le aportan al desarrollador una base para iniciar un proyecto en menos tiempo, más seguridad, más funcionalidades, y demás características propias de cada framework, pero aporta ventajas solo si se tiene conocimiento previo del uso y configuración del mismo, en caso contrario aumenta la curva de aprendizaje para el desarrollador, por ello en este trabajo se propone un procedimiento para utilizar el framework Angular en la creación de aplicaciones web modernas del tipo SPA.

### **1.1 PLANTEAMIENTO DEL PROBLEMA**

Actualmente el desarrollo web es un universo el cual cuenta con una infinidad de herramientas que le facilitan el trabajo al desarrollador pero estas a su vez evolucionan extremadamente rápido, por lo que se hace necesario estar actualizado de las innovaciones y tendencias tecnológicas.

Los framework son el punto de partida para muchos desarrolladores ya que estos facilitan y agilizan el trabajo, logrando así organización en su código, en archivos, proporcionan más seguridad y a su vez reducen notablemente las líneas de código, atendiendo aspectos más importantes a la hora del desarrollo como el rendimiento, usabilidad, escalabilidad y modularidad. Sin embargo cuando los desarrolladores van a incursionar en la utilización de algún framework no existen procedimientos que den pautas para entender los diferentes pasos que se deben de seguir, es allí donde se presenta la dificultad ya que es necesario buscar información de distintas fuentes para comenzar a entender cómo funciona el framework que se va a utilizar y desarrollar en él.

## **1.2 JUSTIFICACIÓN**

Hoy en día tanto empresas como personas independientes que se dedican al desarrollo de software, cuando se dan a la tarea de trabajar en un nuevo proyecto toman la decisión de programar mediante la ayuda de un Framework el cual les aporta grandes beneficios en la aplicación a desarrollar. Para los desarrolladores web, lo más importante es crear una interfaz que sea atractiva, fácil de usar y que sea rápida a la hora de visualizar información es por ello que el framework Angular es una excelente opción de aplicaciones web del tipo Single Page Application, proporcionándole al usuario una experiencia más fluida, ya que los códigos necesarios se cargan una sola vez y luego en ningún otro momento volverá a solicitar estos archivos para su visualización.

A pesar de que en internet se encuentra una gran más de información acerca de Angular, esta información varia de una fuente a otra, es de pago o se encuentra incompleta para lo que se requiere, es de allí donde surge la necesidad de contar con un procedimiento en el cual se encuentren plasmados los pasos y pautas necesarias para desarrollar una aplicación web moderna del tipo SPA con el framework Angular.

## **1.3 INDICADORES DE USO DE ANGULAR**

### **1.3.1 USO DE ANGULAR**

Actualmente se ha reflejado un considerable aumento en el uso de los framework JavaScript. Cualquier desarrollador en algún momento va a utilizar JavaScript en algún proyecto, por ello se hace necesario adquirir conocimientos del mismo debido a que se es muy utilizado y preferido por empresas o personas independientes dedicadas al desarrollo de software.

“La popularidad de JavaScript sigue creciendo. Recientemente JavaScript ha sido designado como uno de los mejores lenguajes de programación para aprender en 2017 por IBM. Se usa tanto en el lado del cliente como en el lado servidor y ayuda a diseñar interfaces espectaculares, enriquecer aplicaciones web con multitud de funciones, editar páginas web en tiempo real y más cosas, extendiendo su influencia al servidor e incluso a dispositivos embebidos.”<sup>1</sup>

Un punto a favor de los framework JavaScript es el desarrollo rápido de aplicaciones web, gracias a sus herramientas y facilidades de desarrollo, aumento en la seguridad de las aplicaciones, costos ya que estos framework son gratuitos, mantenimiento más fácil ya que tienen una arquitectura bien definida y fácil de entender.

Según el portal de cursos online para programadores **Campus MVP** sitúan a Angular en el 1 de la lista en un estudio de los mejores Framework JavaScript en 2017 y destacando las siguientes ventajas y puntos a favor.

- “Desarrollo rápido.
- Enlazado de datos en ambas direcciones.
- Te fuerza a usar buenas prácticas y una serie de arquitecturas, por lo que es fácil que cualquiera retome tu trabajo y también colaborar.
- Facilidad de testing.

Angular es probablemente la mejor opción para empresas o para entornos de programación estrictos, con altos estándares en cuanto a la legibilidad del código.”

2

---

<sup>1</sup> Campus MVP. Los 5 mejores Framework de JavaScript en 2017. [En Línea]  
<<https://www.campusmvp.es/recursos/post/los-5-mejores-frameworks-de-javascript-en-2017.aspx>>  
[Citado el 03 Marzo de 2018]

<sup>2</sup> Campus MVP. Los 5 mejores Framework de JavaScript en 2017. [En Línea]  
<<https://www.campusmvp.es/recursos/post/los-5-mejores-frameworks-de-javascript-en-2017.aspx>>  
[Citado el 03 Abril de 2018]

Otro estudio realizado por parte del portal **Open Webinars** de los 6 mejores Framework JavaScript, en la cual destacan:

“Si la idea es crear sitios web de una sola página con elementos interactivos, no hay una opción mejor que Angular. La vinculación bidireccional es la característica innovadora más intuitiva de Angular que permite la actualización en tiempo real de un sitio web desde diferentes dispositivos. Cualquier vista se cambia de forma simultánea en la aplicación móvil y el sitio web. Angular cuenta además con el mayor apoyo comunitario activo en todo el mundo.”<sup>3</sup>

Así como estos 2 portales de cursos online de desarrollo web en otros sitios web también se encuentran con grandes ventajas y herramientas que ofrece el Framework Angular.

### 1.3.2 SITIOS WEB QUE USAN ANGULAR

Actualmente son muchas las personas o empresas dedicadas al desarrollo de software que utilizan Angular en sus proyectos, algunos de estos proyectos desarrollados con Angular se pueden encontrar en los sitios web:

- <http://angularexpo.com/>
- <https://www.madewithangular.com/categories/angular>

A continuación algunos de los proyectos desarrollados con Angular más conocidos se pueden apreciar en las figuras 1, 2, 3 y 4.

---

<sup>3</sup> Open Webinars. Los 6 mejores Frameworks JavaScript. [En línea]  
<<https://openwebinars.net/blog/los-6-mejores-frameworks-javascript/>> [Citado el 03 Abril del 2018]

# VMware Open Source Software

Angular 4.3.6



Figura 1 Página VMware

# Freelancer

Angular 5.1.0

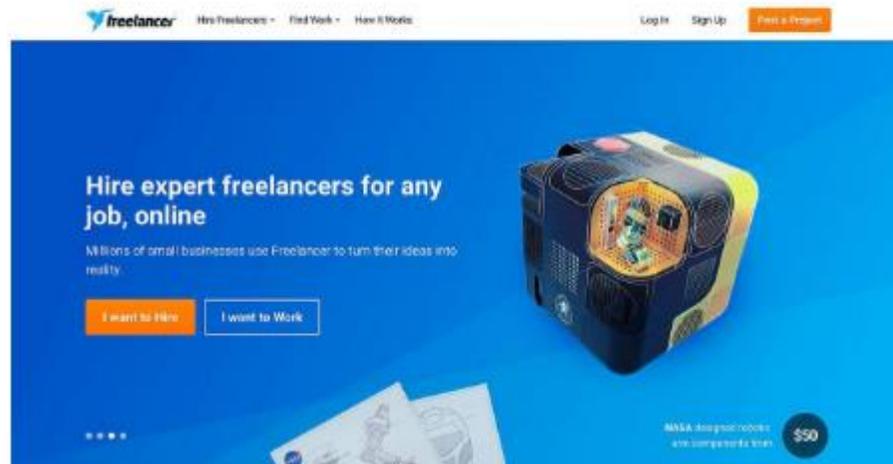


Figura 2 Página Freelancer

# Royal Caribbean

Angular 4.3.1

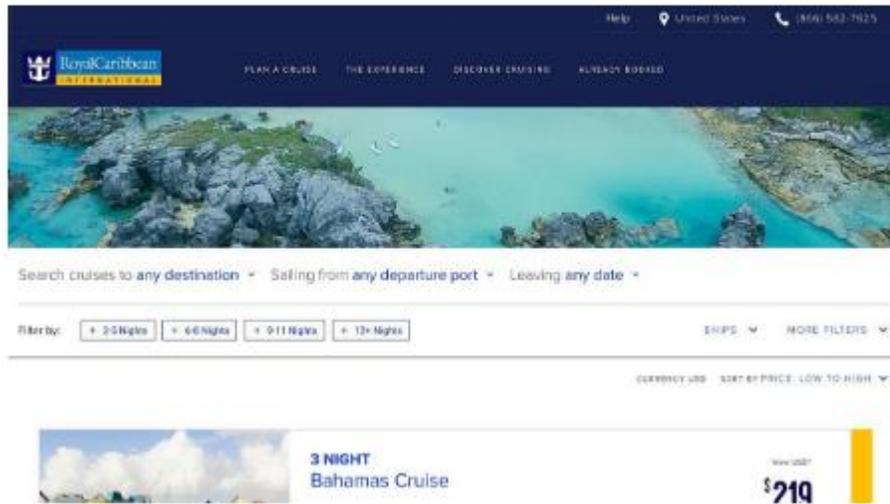


Figura 3 Página Royal Caribbean

# NBA.com

Angular 4.2.4



Figura 4 Página NBA

## **1.4 OBJETIVOS**

### **1.4.1 OBJETIVO GENERAL**

Diseño de un procedimiento para utilizar el framework Angular en la creación de aplicaciones web modernas SPA.

### **1.4.2 OBJETIVOS ESPECIFICOS**

- Realizar un estudio del estado del arte de los procedimientos existentes para la implementación de los diferentes framework.
- Diseñar el procedimiento para la utilización del framework Angular en el desarrollo de aplicaciones web modernas de tipo SPA.
- Validar el procedimiento mediante el desarrollo de una aplicación web del tipo SPA con el uso del framework Angular.

## **1.5 ACOTACIONES**

El procedimiento se diseñó para el framework Angular. La validación del procedimiento fue con el desarrollo de un prototipo de aplicativo web “Bitácora de trabajo social dirigido por Edgar Alexis Albornoz. Así mismo se suministró este procedimiento a dos expertos los cuales aportaron sus observaciones, evaluando la calidad de dicho procedimiento.

## 2 MARCO TEÓRICO Y ESTADO DEL ARTE

En el presente capítulo se presentarán los conceptos más importantes para entrar en contexto, también se realizará un estudio del estado del arte, analizando un conjunto de tesis con proyectos similares, para posteriormente realizar un cuadro comparativo donde se resaltarán las características de vital importancia para nutrir el desarrollo del procedimiento.

### 2.1 MARCO TEORICO

Una aplicación de una sola página o SPA. Es una aplicación en la cual todas las funcionalidades se ejecutan en una única página, cargando toda la lógica una vez al momento de cargar la página o de forma dinámica en el momento en el que la requiera, eliminando así los tiempos de carga y generando la mayoría de procesamiento por parte del cliente, esto con el fin de darle al cliente una aplicación web más fluida y más rápida similar a una aplicación de escritorio.

#### 2.1.1 FRAMEWORK

“En general, un marco de trabajo, o framework, es una estructura real o conceptual destinada a servir de soporte o guía para la construcción de algo que expande la estructura en algo útil.”<sup>4</sup>

“En los sistemas informáticos, un framework es a menudo una estructura en capas que indica qué tipo de programas pueden o deben ser construidos y cómo se interrelacionan. Algunos marcos de trabajo de sistemas informáticos también

---

<sup>4</sup> Red de TechTarget. Framework. [En Línea]  
<<http://searchdatacenter.techtarget.com/es/definicion/Framework>> [Citado el 03 de Abril del 2018].

incluyen programas reales, especifican interfaces de programación u ofrecen herramientas de programación para usar los marcos. Un framework puede servir para un conjunto de funciones dentro de un sistema y cómo se interrelacionan; las capas de un sistema operativo; las capas de un subsistema de aplicación; cómo debería normalizarse la comunicación en algún nivel de una red; etcétera. Un marco de trabajo es generalmente más completo que un protocolo y más prescriptivo que una estructura.”<sup>5</sup>

Algunas de las principales razones por las cuales se hace uso de los Frameworks en el desarrollo de software son: escribir menos líneas de código o código repetitivo, organización en el código, realizar cosas de manera más fácil rápida que solos costaría mucho tiempo y esfuerzo, logrando un desarrollo más ágil y rápido.

### **2.1.2 SINGLE PAGE APPLICATION**

Single Page Application (SPA), Aplicación de página única, significa que cabe en una sola página, con la finalidad de que su acceso sea más fluido y rápido simulando una aplicación de escritorio siendo más rápida la visualización de la información. La navegación en una aplicación web de una sola página se realiza casi que de manera inmediata al dar clic.

“En pocas palabras, SPA son las siglas de Single Page Application. Es un tipo de aplicación web donde todas las pantallas las muestra en la misma página, sin recargar el navegador.

Técnicamente, una SPA es un sitio donde existe un único punto de entrada, generalmente el archivo index.html. En la aplicación no hay ningún otro archivo HTML al que se pueda acceder de manera separada y que nos muestre un

---

<sup>5</sup> Red de TechTarget. Framework. [En Línea]  
<<http://searchdatacenter.techtarget.com/es/definicion/Framework>> [Citado el 03 de Abril del 2018].

contenido o parte de la aplicación, toda la acción se produce dentro del mismo index.html.”<sup>6</sup>

### 2.1.3 ANGULAR

“Angular (comúnmente llamado "Angular 2+" o "Angular 2"), es un framework para aplicaciones web desarrollado en TypeScript, 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. La biblioteca lee el HTML que contiene atributos de las etiquetas personalizadas adicionales, entonces obedece a las directivas de los atributos personalizados, y une las piezas de entrada o salida de la página a un modelo representado por las variables estándar de JavaScript. Los valores de las variables de JavaScript se pueden configurar manualmente, o ser recuperados de recursos JSONestáticos o dinámicos.

Angular es la evolución de AngularJS aunque incompatible con su predecesor.”<sup>7</sup>

#### **Características:**

- Velocidad y rendimiento:

Generación de código: Angular convierte tus plantillas en código altamente optimizado para las máquinas virtuales de JavaScript de hoy en día, ofreciéndote

---

<sup>6</sup> Desarrollo web. Que es una SPA. [En línea]. < <https://desarrolloweb.com/articulos/que-es-una-spa.html> > [citado el 3 de marzo de 2018].

<sup>7</sup> Wikipedia. Angular (framework), Características. [En línea] <[https://es.wikipedia.org/wiki/Angular\\_\(framework\)](https://es.wikipedia.org/wiki/Angular_(framework))> [citado el 3 de marzo de 2018].

todas las ventajas del código escrito a mano con la productividad de un framework.

Universal: Ejecuta la primera vista de tu aplicación en node.js, .NET, PHP, y otros servidores para renderizado de forma casi instantánea obteniendo solo HTML y CSS. También abre posibilidades para la optimización del SEO del sitio.

División del código: Las aplicaciones de Angular se cargan rápidamente gracias al nuevo enrutador de componentes. Éste ofrece una división automática de códigos para que los usuarios sólo carguen el código necesario para procesar la vista que solicitan.

- Productividad

Plantillas: Permite crear rápidamente vistas de interfaz de usuario con una sintaxis de plantilla simple y potente.

Angular CLI: Las herramientas de línea de comandos permiten empezar a desarrollar rápidamente, añadir componentes y realizar test, así como pre visualizar de forma instantánea la aplicación.

IDEs: Obtén sugerencias de código inteligente, detección de errores y otros comentarios en la mayoría de los editores populares e IDEs.

- Historia completa del desarrollo

Testing: Utiliza Karma para realizar pruebas unitarias, y Protractor para realizar pruebas end-to-end de forma rápida y estable.

Animación: Permite crear animaciones complejas y de alto rendimiento con muy poco código a través de la intuitiva API de Angular.

Accesibilidad: Posee características para crear aplicaciones accesibles con los componentes disponibles para ARIA.”<sup>8</sup>

## 2.1.4 FIREBASE

Firebase es una plataforma perteneciente a Google, con la finalidad de desarrollar aplicaciones web y móviles de alta calidad. En este procedimiento se realizará la parte de backend con esta plataforma. Firebase cuenta con un “conjunto de herramientas orientadas a la creación de aplicaciones de alta calidad, al crecimiento de los usuarios y a ganar más dinero. Personalmente describen la plataforma como una suite de diferentes aplicaciones que nos harán más fácil el desarrollo de nuestra aplicación.”<sup>9</sup>

Entre los servicios más destacados se encuentran:

HERRAMIENTAS	DESCRIPCION
Base de datos en tiempo real	Permite la administración de los datos en tiempo real, datos almacenados en formato JSON.
Autenticación	Servicio el cual hace más fácil un inicio de sesión, y gestión de la aplicación siendo muy fácil configurar métodos con Google, Facebook, Twitter, entre otros.
Almacenamiento de archivos	Servicio excelente a la hora de tener un aplicativo en el que se requiera guardar archivos de todo tipo.
Servicio de hosting	Servicio de hosting gratuito por parte de Firebase, simplemente realizando el build de la aplicación se podrá subir.
Laboratorio de test	Permite probar los test de la aplicación en el entorno deseado, solo para android.
Informes sobre fallos	Con esta herramienta se pueden detectar los errores en las aplicaciones usadas, y visualiza como eliminarlos.

---

<sup>8</sup> Wikipedia. Angular (framework), Características. [En línea]

<[https://es.wikipedia.org/wiki/Angular\\_\(framework\)](https://es.wikipedia.org/wiki/Angular_(framework))> [citado el 3 de marzo de 2018].

<sup>9</sup> Open Webinars. ¿Qué es Firebase de Google? [En línea] <<https://openwebinars.net/blog/que-es-firebase-de-google/>>

Monitoreo de rendimiento	Permite visualizar el rendimiento de la aplicación y de igual manera el tiempo de los usuarios activos en la aplicación.
Notificaciones.	Con esta herramienta se puede enviar notificaciones a los usuarios registrados en el aplicativo y programar envío de notificaciones.
Configuración remota.	Da la posibilidad de personalizar el comportamiento y apariencia del aplicativo.

*Tabla 1 Herramientas Firebase*

### **2.1.5 TYPESCRIPT**

TypeScript es un lenguaje de programación de alto nivel de código abierto que fue desarrollado por Microsoft. Este lenguaje de programación es un súper conjunto de JavaScript.

“TypeScript es un lenguaje de programación de alto nivel que implementa muchos de los mecanismos más habituales de la programación orientada a objetos, pudiendo extraer grandes beneficios que serán especialmente deseables en aplicaciones grandes, capaces de escalar correctamente durante todo su tiempo de mantenimiento.

La característica fundamental de TypeScript es que compila en JavaScript nativo, por lo que se puede usar en todo proyecto donde se esté usando JavaScript. Dicho con otras palabras, cuando se usa TypeScript en algún momento se realiza su compilación, convirtiendo su código a JavaScript común. El navegador, o cualquier otra plataforma donde se ejecuta JavaScript, nunca llegarán a enterarse que el código original estaba escrito en TypeScript, porque lo único que llegará a ejecutar es el JavaScript resultante de la compilación.

En resumen, TypeScript es lo que se conoce como un "superset" de JavaScript, aportando herramientas avanzadas para la programación que traen grandes beneficios a los proyectos."<sup>10</sup>

## 2.1.6 TIPOS DE DESARROLLADORES

- **DESARROLLADOR FRONT-END**

Este tipo de desarrollador es el encargado de elaborar la parte visual de una aplicación web, por lo que su trabajo se encuentra en el navegador o lado del cliente, estos obligatoriamente deben conocer estas 3 herramientas principales: HTML, CSS y JavaScript. "Son los encargados de traducir el diseño de las páginas en código para construir la interfaz de los sitios. Este actor es una de las piezas fundamentales en un proyecto digital, encargado de traducir las definiciones de diseño y estilo visual realizadas en etapas previas a código HTML semántico, tomando la batuta en la construcción final de las interfaces."<sup>11</sup>

"En general se asocia a los desarrolladores front-end con los principios de diseño y de estructura de páginas. Sin embargo, un desarrollador web va más allá que un diseñador. Obviamente tiene que tener en cuenta la usabilidad y la legibilidad de la página o de la aplicación web, pero como buen programador es consciente de que su trabajo se ejecutará en el lado Cliente, en la mayoría de los casos, en el navegador. Y la información no se almacena en el lado Cliente."<sup>12</sup>

---

<sup>10</sup> Desarrollo web. Introducción a TypeScript. [En línea].

<<https://desarrolloweb.com/articulos/introduccion-a-typescript.html>> [citado el 3 de marzo de 2018].

<sup>11</sup> IDA. ¿Qué esperamos de un desarrollador front-end? [En línea]

<<https://www.ida.cl/blog/desarrollo/desarrollador-front-end/>> [citado el 3 de marzo de 2018].

<sup>12</sup> Campus MVP. Desarrollador web: Front-end, back-end y full stack. ¿Quién es quién? [En línea]

<<https://www.campusmvp.es/recursos/post/Desarrollador-web-Front-end-back-end-y-full-stack-Quien-es-quien.aspx>> [citado el 3 de marzo de 2018].

- **DESARROLLADOR BACK-END**

Este es el encargado de desarrollar la programación de los distintos componentes los cuales está conformado el sitio web.

“El desarrollador back-end trabaja del lado Servidor, detrás del escenario, permitiendo con su trabajo que el usuario disfrute de su experiencia. Sin él, el desarrollo llevado a cabo por su anterior compañero no se sostendría.”<sup>13</sup>

Este tipo de desarrollador tiene que tener conocimientos en algunas de las siguientes tecnologías: Java, ASP.NET, PHP, Node.js, Ruby. Pero también debe de tener conocimientos en bases de datos las cuales se pueda almacenar la información e interactuar con la programación de los distintos componentes, algunos de los motores de base de datos más comunes son: PostgreSQL, Mysql, Oracle, Mongó entre muchos más.

- **DESARROLLADOR FULL STACK**

Este tipo de desarrollador es la unión de los dos anteriormente mencionados, debe tener claro y familiarizarse con los conceptos front-end y back-end ya que este tipo de desarrollador se encarga de todo el desarrollo del proyecto de principio a fin, desde el montaje de los servidores hasta la maquetación de la aplicación web con CSS.

---

<sup>13</sup> Campus MVP. Desarrollador web: Front-end, back-end y full stack. ¿Quién es quién? [En línea] <<https://www.campusmvp.es/recursos/post/Desarrollador-web-Front-end-back-end-y-full-stack-Quien-es-quien.aspx>> [citado el 3 de marzo de 2018].

## **2.2 ESTADO DEL ARTE**

En este capítulo se presentarán las tesis más similares en ámbitos local, nacional, e internacional, visualizando un resumen de dicha tesis para posteriormente estudiarlas y analizarlas, sacando resultados que posteriormente van a nutrir el procedimiento propuesto.

### **2.2.1 TESIS RELACIONADAS**

#### **1. TESIS DE GRADO: “INTEGRACIÓN DE SINGLE-PAGE APPLICATION EN LIFERAY”. AUTOR: JOSÉ DE LA FUENTE MACIÁ. BARCELONA 2016.**

En la universidad politécnica de la ciudad de Cataluña, Barcelona, para la obtención al título de Ingeniero Informático, proyecto en el cual se realiza un estudio acerca de las páginas web del tipo SPA, mostrando sus características, uso, rendimiento, requisitos, protocolos entre otros, describiendo la funcionalidad de una web SPA, para proceder a conocer el framework AngularJS con el cual se realiza la tarea de desarrollar una web Single Page Application, mostrando así las grandes ventajas que ofrecen frente a una página web convencional y posteriormente integrar la web resultante en el CMS Liferay.

#### **Especificación:**

En este proyecto utilizaron la metodología ágil SCRUM, ya que es más flexible y se adapta fácil a los cambios, ya que en la realización de la integración se encuentran cambios constantes. Para el control de versiones se utilizó Git. Las dos tareas más importantes que definieron en este proyecto fueron el desarrollo de la aplicación con AngularJS y la integración de la aplicación AngularJS en Liferay.

## **Desarrollo Aplicación AngularJS:**

- **Requisitos Funcionales:** En el proyecto primero definieron los requisitos funcionales los cuales son los encargados de establecer los comportamientos del sistema, siguiendo una metodología SCRUM, el cual utiliza las historias de usuario para definir las funcionalidades del sistema en este caso de carreras de fórmula uno.
- **Requisitos no funcionales.**
- **Diseño arquitectónico:** Seguido definieron el diseño arquitectónico en el cual describe la aplicación web está conformada por una web SPA en AngularJS, la cual se comunica a una API externa mediante HTTP, recibiendo datos JSON e integrarla a Liferay.
- **Patrones de diseño:** Los patrones de diseño que utilizaron fue MVC para AngularJS, y si se habla de los servicios de AngularJS se utiliza el patrón de diseño Singleton.
- **Protocolos:** Seguido en el proyecto especifican los protocolos empleados REST y RESTFUL API.
- **Capa de presentación:** Como no debe faltar se realiza un bosquejo de la capa de presentación mostrando las vistas a implementar con sus respectivas funcionalidades.
- **Capa de dominio:** Por medio de la metodología SCRUM la especificación del modelo conceptual a medida que el proyecto avanza, este fue incrementando y perfeccionando en cada Spring, acá también describieron el modelo de comportamiento por medio de diagramas de secuencia.

## **Integración Liferay:**

Acá presentan una introducción de Liferay, sus características, arquitectura, roles, jerarquía, así como una descripción de las tecnologías que utiliza Liferay. Para posteriormente proceder a la implementación.

## **2. TESIS DE GRADO: “DESARROLLO DE LA CAPA DE PRESENTACIÓN DE UNA PLATAFORMA DE GESTIÓN DE FORMAS DE PAGO Y CÁLCULO DE CONDICIONES EN LA GESTIÓN DE CRÉDITOS AL CONSUMO UTILIZANDO ANGULAR 2” AUTOR: VEGA L DAVID. MADRID JUNIO 2017**

En la universidad politécnica de la ciudad de Madrid, España, para la obtención del título de Ingeniero Informático, realizaron este trabajo de grado en el cual la principal tarea fue desarrollar la capa de presentación de una plataforma que gestiona créditos al consumo. En este trabajo el estudiante describe el desarrollo Front-End en Angular para implementar en la plataforma de créditos, también una comparación de otros Framework Front-End que existen, el manejo de Bootstrap para los estilos, el uso de Firebase como motor de base de datos, presenta también el modelo de datos utilizado y por último presenta una galería con la capa de presentación funcionando. Tomando como conclusión la gran experiencia adquirida en el desarrollo de la capa de presentación con el uso de tan potente Framework como lo es Angular.

### **Especificación:**

El estudiante en primera instancia realiza una breve descripción de los créditos al consumo y la gestión de formas de pago, también una presentación del framework a utilizar Angular 2, comparándolo con otros framework y librerías también utilizadas actualmente. En el proyecto describe también otras tecnologías utilizadas para su desarrollo de la capa de presentación como lo es: Bootstrap y Firebase.

### **Desarrollo:**

- **Modelo de datos:** En esta sección explican el motor de base de datos localizada en el servidor de Firebase, la cual recibe como entrada una estructura en JSON con la información necesaria, describen también las tablas empleadas.
- **Diseño web del cliente:** Este diseño web y la implementación fue con Angular y Bootstrap, con motor de base de datos en Firebase, realizando peticiones

entre sí. En esta misma sección describen paso a paso las vistas o funcionalidades implementadas.

### **3. TESIS DE GRADO: “DESARROLLO DE UN PORTAL WEB UTILIZANDO EL FRAMEWORK LARAVEL Y BOOTSTRAP PARA LA DIFUSIÓN TURÍSTICA DE LA HOSTERÍA FABRICIO’S EN LA PARROQUIA DE AMBUQUI” AUTOR: IRMA PATRICIA COLIMBA HUERTAS. ECUADOR 2016.**

En la universidad técnica del norte por medio de la obtención al título de ingeniería en sistemas computacionales existe una tesis en la cual desarrollaron un portal web utilizando el Framework Laravel y Bootstrap para la difusión turística de la hostería para una parroquia, en Ibarra, Ecuador. Proyecto en el cual con el fin de incentivar el turismo en la parroquia de Ambuqui con la ayuda del Framework Laravel junto a Bootstrap, siguiendo una metodología XP para sistematizar los procesos de la hostería Fabricio. Allí se desarrollan módulos como: Facturación, administrador de contenidos, inventarios, Reservaciones en línea, así como recepción. En este trabajo se realiza un estudio preliminar de población, de las herramientas a utilizar, de la metodología propuesta, para proceder a crear historias de usuario, base de datos y comenzar a programar en Laravel. Dando como resultado el portal web requerido.

#### **Especificación:**

En este proyecto principalmente se enfocaron en conocer detalladamente los portales web, las herramientas de desarrollo a utilizar, la integración de Laravel y Bootstrap y posteriormente la metodología de desarrollo a utilizar que en este caso fue XP (Programación Extrema), por lo que vieron necesario la rápida culminación del proyecto, desarrollando en periodos cortos. Para proceder al desarrollo.

#### **Desarrollo portal web:**

- Planificación del proyecto: En esta etapa del proyecto llevaron a cabo por

medio de una metodología XP, la asignación de roles, creación de historias de usuario, velocidad del proyecto, presupuesto, así como los recursos.

- Diseño del sistema: Acá definieron la arquitectura basado en MVC, en el proyecto utilizaron Postgresql, Laravel y Bootstrap.
- Módulos del sistema: Breve descripción de los módulos a implementar.
- Asignación de Historias de usuario.
- Estimación del esfuerzo: Planificación del tiempo a emplear en el desarrollo.
- Fase de diseño: Realizaron la creación del modelo de datos a utilizar con los respectivos casos de uso y sus descripciones.
- Fase de iteraciones: Por medio de iteraciones fueron construyendo las historias de usuario.
- Fase de pruebas: Realizaron las respectivas pruebas de aceptación a cada una de las funcionalidades condensadas en las historias de usuario.

**4. TESIS DE GRADO: “DESARROLLO E IMPLEMENTACIÓN DEL MÓDULO DE CONTABILIDAD, CARTERA EN EL SISTEMA DE ADMINISTRACIÓN ELECTRÓNICA (SAEL) BASADO EN EL FRAMEWORK JBOSS SEAM PARA LA EMPRESA TOVAR COLOMBIA S.A”. AUTOR: JESÚS ALBERTO LOZANO TAMAYO, JOSÉ LUIS LOZANO TAMAYO. OCAÑA 2012.**

En la universidad Francisco de Paula Santander en Ocaña, Norte de Santander, previo a la obtención al título de ingeniero de sistemas, realizaron este proyecto en el cual se enfocaron en la metodología ágil XP y apoyándose en el Framework Jboss Seam, realizando primero la búsqueda del tipo de investigación, seguido seleccionaron una población para hallar la muestra, realizaron la recolección de datos y utilizando técnicas de procesamiento y análisis de información procedieron a realizar las tareas de desarrollo como: planeación, historias de usuario, plan de iteración, diseño, diagramas de secuencia, la codificación y por ultimo las pruebas para llevar a cabo la implementación del módulo de contabilidad para la empresa

PROVAR Colombia S.A. mejorando notablemente los procesos realizados diariamente y así obtener un mejor desempeño de la empresa.

### **Especificación:**

Para este proyecto utilizaron también la metodología XP, especificando el funcionamiento, seleccionando la población y la muestra para posteriormente analizar la información y seleccionar las herramientas tecnológicas con las cuales procedieron a trabajar.

### **Desarrollo del módulo de contabilidad:**

- Planeación: Aquí describieron la metodología XP la cual utilizaron en su proyecto.
- Historias de usuario: De igual manera describieron cada una de las historias de usuario necesarias para el proyecto.
- Plan de iteración: Describieron las iteraciones de acuerdo a la primera planeación.
- Diseño: Elaboraron las tarjetas CRC, en las cuales está la descripción de las iteraciones entre clases, describiendo así cada uno de los módulos a desarrollar.
- Diagramas de secuencia: Luego para entender mejor las historias de usuario.
- Codificación: Allí llevaron a cabo la programación por parejas, describieron la arquitectura por capas para luego implementarlas y por último realizaron las pruebas necesarias: una prueba de aceptación y otras encuestas evaluando la calidad del desarrollo del módulo de contabilidad.

## **5. TESIS DE GRADO: “FRAMEWORK PARA EL DESARROLLO DE SOFTWARE EN ENTORNOS ACADÉMICOS”. AUTOR: RAFAEL DAVID RINCÓN BERMÚDEZ. MEDELLÍN 2012**

En la universidad EAFIT de Medellín, para optar al título de magister en ingeniería,

realizaron un proyecto en el cual la principal tarea era desarrollar un Framework conocido como Software EDUP para el desarrollo de software en entornos académicos. Framework el cual buscaba ser del tipo metodológico con herramientas, plantillas, instructivos, políticas, métodos útiles para los desarrollos formales en materias relacionadas con la ingeniería de software. Este Framework lo construyeron con base en estos pilares como lo son: RUP, CMMI Y PmBok. El cual está adaptado para proyectos académicos ofreciendo una guía para cada uno de los roles implicados en el ciclo de vida del desarrollo de software. Es útil para que los estudiantes comprendan y aprendan las actividades a realizar por los distintos roles en un proyecto. (Jorge Hernán Abad Londoño, 2012).

### **Especificación:**

Este proyecto de maestría se centra en la construcción de un Framework, comenzando con un marco teórico, para conocer las bases con las cuales se fundamenta el Framework, RUP, CMMI y PmBok. También fundamentado con el desarrollo ágil de software. En cuanto a la metodología de trabajo, primero eligieron EPF composer para la publicación del framework, plantearon los procesos y proyectos propuestos, definieron la estructura dinámica del framework, algunos ejemplos propuestos, escenarios de uso, por supuesto una validación del framework y una vista de los componentes y licencia del framework. También se aclaró el valor agregado del framework frente a propuestas ya existentes.

### **6. TESIS DE GRADO: “ESTUDIO DEL FRAMEWORK ANGULAR.JS COMO HERRAMIENTA PARA LA IMPLEMENTACIÓN DEL PATRÓN MVC DEL LADO DEL CLIENTE”. AUTOR: DOYLER EMMANUEL RUBIO VILLAMIZAR. PAMPLONA, JUNIO 2015.**

En el ámbito local se encuentra la Universidad de Pamplona un estudio de los inicios de Angular conocido como Angular.js su predecesor por Doyler Emmanuel Rubio Villamizar, titulado Estudio del Framework Angular.js como herramienta para la

implementación del patrón MVC del lado del cliente. Proyecto en el cual el estudiante realiza un estudio bibliográfico de la herramienta Angular.js, de la arquitectura MVC para posteriormente con estos conocimientos proponer un modelo MVC utilizando la tecnología web service e integrando así mismo el Framework Angular.js creando 2 funcionalidades para una empresa de enseñanza de carros de la ciudad de Pamplona. (Doyler Emmanuel Rubio Villamizar, Universidad de Pamplona, 2015).

### **Especificación:**

En primera instancia en este proyecto realizaron un estudio de los conceptos preliminares, una introducción a la programación web, navegadores web, conceptos generales de: HTML5, CSS3, bases de datos, JSON, AJAX, patrones de diseño MVC, MVP, MVVM, también sobre servicios web. Seguido de ello el estudiante realizo un estudio de AngularJS: consultando las herramientas necesarias para el desarrollo, estrategias de organización en su código, puesta en marcha del proyecto, las directivas, importación de bootstrap a AngularJS, formularios, algo sobre los proyectos en AngularJS.

### **Modelo MVC propuesto:**

Acá primero describe el modelo MVC funcionalidades de Login y Gestión de usuarios, también la estructura de un proyecto en AngularJS, así mismo se indican los enlaces para la inclusión de formularios y conexión a AngularJS. Se muestra una capa de control del formulario en AngularJS y la capa del modelo en PHP de los mismos.

Como segundo paso el estudiante reestructura la aplicación a un enfoque MVC generando un directorio que contenga la funcionalidad de gestión de usuarios, en la cual el modelo la vista y el control estén en el mismo lugar, allí describen el enrutamiento, la interacción entre AngularJS y PHP por medio del enrutamiento y los archivos JSON.

**7. TESIS DE GRADO: “PROCEDIMIENTO PARA LA APLICACIÓN DE CONCEPTOS DEL DESARROLLO DIRIGIDO POR MODELOS, MEDIANTE EL USO DEL FRAMEWORK SPRING”. AUTOR: FRANCISCO JAVIER GERALDINO. PAMPLONA, 2017.**

En la Universidad de Pamplona existe un trabajo de grado el cual consiste en un procedimiento para la aplicación de conceptos de desarrollo dirigido por modelos mediante el uso del Framework Spring, donde se realiza un estudio bibliográfico correspondiente de dicho Framework y sobre el desarrollo dirigido por modelos para posteriormente el estudiante definir los pasos necesarios para el desarrollo dirigido por modelos en la construcción de operaciones web básicas: insertar, buscar, actualizar y eliminar validando con ello el procedimiento propuesto. (Francisco Javier Geraldino Fernández, Universidad de Pamplona, 2017).

**Especificación:**

En este proyecto realizaron primero un estudio acerca de modelos, metamodelos, orígenes de MDD, el ciclo de vida de desarrollo dirigido por modelos, metodologías relacionadas con el desarrollo dirigido por modelos, modelo vista controlador, y por último el estado del arte, para proceder a proponer el procedimiento y luego validarlo.

**Procedimiento del desarrollo dirigido por modelos:**

- Descripción y especificación del sistema: Parte en la que describe los alcances y limitaciones del sistema para reconocer las funcionalidades.
- Crear modelos independientes de plataforma: Ya con las funcionalidades clara se utilizó diagramas de clase, casos de uso y storyboard para aclarar la panorámica del sistema.
- Convertir modelos independientes a dependientes: Etapa en la que los diagramas de clases pasan a ser modelos de entidad-relación.
- Crear modelos sujetos a la plataforma: Por medio del diagrama entidad – relación, generaron el código SQL de la base de datos, así mismo los

diagramas de secuencia definen las interacciones entre las capas definidas dentro del sistema.

- Convertir modelos sujetos a plataforma en código: Convirtieron cada entidad en una tabla y con el uso de la herramienta Spring Roo generaron los códigos que construyen el funcionamiento.
- Generación de código: Haciendo uso de las herramientas generaron el código necesario.

### 2.2.2 ANÁLISIS

TESIS	FRAMEWORK	METODOLOGIA	PASOS DEL DESARROLLO	ENFOQUE DEL FRAMEWORK
1	Liferay.	SCRUM	<ol style="list-style-type: none"> <li>1. Requisitos Funcionales.</li> <li>2. Requisitos no Funcionales.</li> <li>3. Diseño arquitectónico.</li> <li>4. Patrones de diseño.</li> <li>5. Protocolos.</li> <li>6. Capa presentación.</li> <li>7. Capa dominio.</li> </ol>	Realizo la integración a modo general.
2	Angular 2.	Ninguna	<ol style="list-style-type: none"> <li>1. Modelo datos.</li> <li>2. Diseño web cliente.</li> </ol>	Gestión de formas de pago y créditos al consumo.
3	Laravel y Bootstrap	XP	<ol style="list-style-type: none"> <li>1. Planificación.</li> <li>2. Diseño sistema.</li> <li>3. Módulos.</li> <li>4. Historias usuario.</li> <li>5. Estimación esfuerzo.</li> <li>6. Fase diseño.</li> </ol>	Difusión turística de la hostería para una parroquia

			<ul style="list-style-type: none"> <li>7. Fase iteraciones.</li> <li>8. Fase pruebas.</li> </ul>	
4	Jboss Seam	XP	<ul style="list-style-type: none"> <li>1. Planeación.</li> <li>2. Historias usuario.</li> <li>3. Plan iteración.</li> <li>4. Diseño.</li> <li>5. Diagramas secuencia.</li> <li>6. Codificación.</li> </ul>	Contabilidad y Cartera en la empresa Tovar, Colombia SA.
5	Desarrollo Framework desde 0, titulado EDUP	Ninguna	No se requiere.	Desarrollo de software en entornos académicos
6	AngularJS.	Ninguna	No se especificó.	Patrón MVC del lado del Cliente.
7	Spring.	Dirigido por modelos (MDD)	<ul style="list-style-type: none"> <li>1. Descripción y especificación del sistema.</li> <li>2. Crear modelos independientes de plataforma.</li> <li>3. Convertir modelos independientes a dependientes.</li> <li>4. Crear modelos sujetos a la plataforma.</li> <li>5. Convertir modelos sujetos a plataforma en código.</li> <li>6. Generación de código.</li> </ul>	Aplicación de conceptos del desarrollo dirigido por modelos

*Tabla 2 Comparativo estado del arte*

Como se puede observar en las tesis mencionadas, estas a excepción de la integración del Single Page Application en Liferay, se enfocan en necesidades

específicas, temas relacionados a lo académico, contable o una empresa y no profundizan en la utilización de dicho Framework, logrando que las personas lo puedan aplicar a sus desarrollos orientado a la temática de su interés.

El procedimiento propuesto está orientado a conocer el framework Angular y poder desarrollar aplicaciones web SPA con el mismo, disminuyendo la curva de aprendizaje.

### **3. DISEÑO Y VALIDACIÓN DEL PROCEDIMIENTO PARA UTILIZAR EL FRAMEWORK ANGULAR EN LA CREACIÓN DE APLICACIONES WEB SPA**

En este capítulo se diseñará un procedimiento para lograr desarrollar aplicaciones web SPA minimizando el tiempo de aprendizaje con el uso de Angular. El Procedimiento se compone de pasos que disminuirán en gran parte el tiempo de aprendizaje invertido, enfocándose solo en tareas puntuales e importantes, sin desviarse del camino en pasos que no conllevan al desarrollo del aplicativo web SPA. Posteriormente, se realizará la validación respectiva del procedimiento, verificando que cumpla con las expectativas, para ello se desarrollará un aplicativo web SPA usando el procedimiento, adicionalmente se anexa el concepto de 2 profesionales, que estudiarán y evaluarán la calidad del procedimiento propuesto, aportando sus observaciones.

#### **3.1 PROCEDIMIENTO ANGULARDASPA**

El procedimiento AngularDASPA (Angular Desarrollo Ágil SPA) se compone de varias etapas que deben abordarse para lograr tener un desarrollo ordenado y el desarrollador pueda avanzar en los diferentes apartados necesarios, iniciando con la conceptualización del proyecto, obtención de requerimientos, diseño del

Storyboard, selección de las dependencias necesarias, manejo de componentes, routing, configuración de firebase el cual proporciona las herramientas de base de datos, autenticación de usuarios y operaciones CRUD.

## PROCEDIMIENTO ANGULARDASPA

Danny Bismarck Castro I |

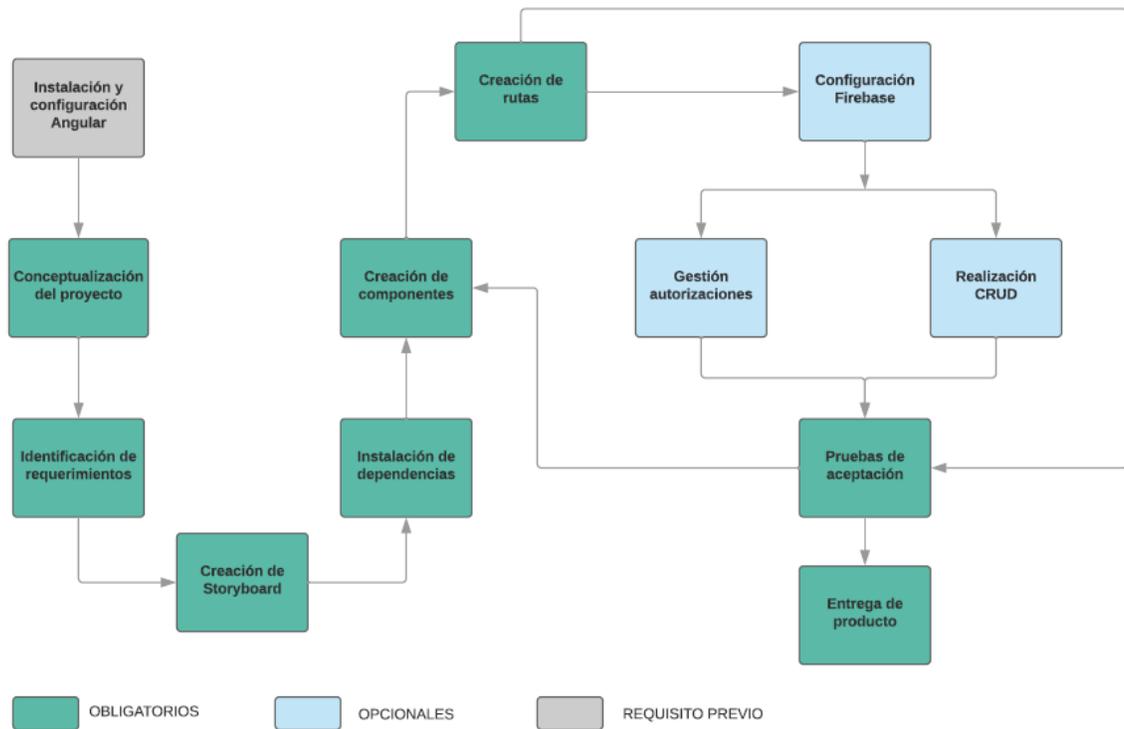


Figura 5 Procedimiento AngularDASPA

Para comprender el desarrollo del procedimiento se procederá a abordar los diferentes pasos que se consideran necesarios en el aprendizaje del framework Angular (ver figura 5) para el desarrollo de una aplicación web del tipo SPA.

### 3.1.1 INSTALACIÓN Y CONFIGURACIÓN DE ANGULAR

Un paso muy importante para trabajar en Angular, así como en cualquier otro Framework orientado al desarrollo de aplicaciones, es la instalación de dicha herramienta junto con sus requisitos previos, este paso es imprescindible para comenzar a trabajar en Angular, pero a su vez no se considera estrictamente un paso a seguir dentro de la lista de este procedimiento ya que la persona interesada puede que tenga una instalación previa de Angular y sus requisitos, siendo innecesario cumplir con este paso.

#### A. Instalación Node.JS

Angular es una herramienta de JavaScript, por ellos requiere de **Node.js**, entorno en tiempo de ejecución, para la capa del servidor, el cual permite ejecutar JavaScript, lenguaje que utiliza Angular y es necesario para compilar y ejecutar los proyectos allí desarrollados, es por ello que se requiere la instalación previa para poder utilizar Angular. Para descargarlo utilice la página oficial ver figura 6:

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

14

La instalación es sencilla y solo basta con seguir el proceso aceptando la configuración sugerida hasta completarla.

---

<sup>14</sup> La siguiente convención será utilizada para especificar: Notas, Código y direcciones de páginas.



Figura 6 Node.js (<https://nodejs.org/es/>)

## B. Instalación IDE

Otra de las herramientas importantes para los desarrolladores cuando inician un proyecto es la elección de un editor de código en el cual se va a escribir la aplicación, por lo general la mayoría utiliza el editor de su preferencia para todos sus desarrollos sin importar el lenguaje, pero dependiendo del proyecto que se esté desarrollando, y sus herramientas es aconsejable usar un editor que se adapte fácilmente a las necesidades del proyecto, es por ello que en este procedimiento se sugiere utilizar **Visual Studio Code**, un editor de código abierto, desarrollado por Microsoft, el cual cuenta con soporte de TypeScript, con ello se evita la realización de configuraciones adicionales y se va a trabajar de manera más cómoda y rápida. Esta herramienta se puede descargar desde la página oficial:

<https://code.visualstudio.com/>

La instalación es sencilla y solo basta con seguir el proceso aceptando la configuración sugerida hasta completarla.

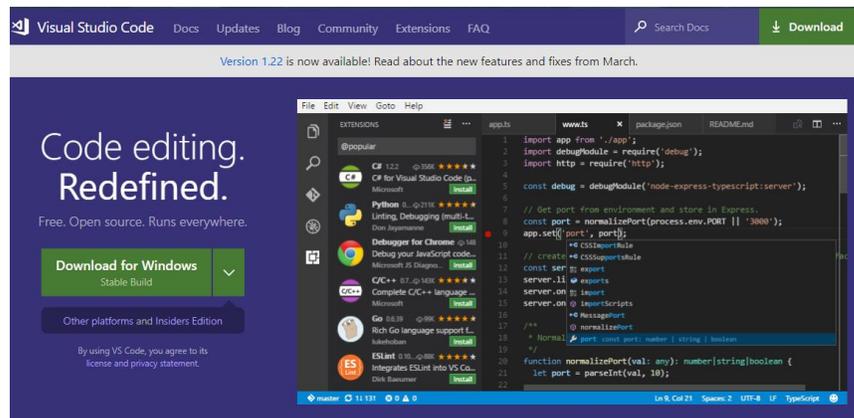


Figura 7 Visual Studio Code

### C. Instalación Angular CLI

Ya instalado Node.js se puede hacer uso del instalador de paquetes npm, con ella se puede instalar Angular Cli, una herramienta de línea de comando para generar proyectos listos para empezar, con ventajas como: desarrollar más rápido, añadir componentes y visualizar de forma instantánea la aplicación. Angular Cli se instala con el siguiente comando.

```
npm install -g @angular/cli
```

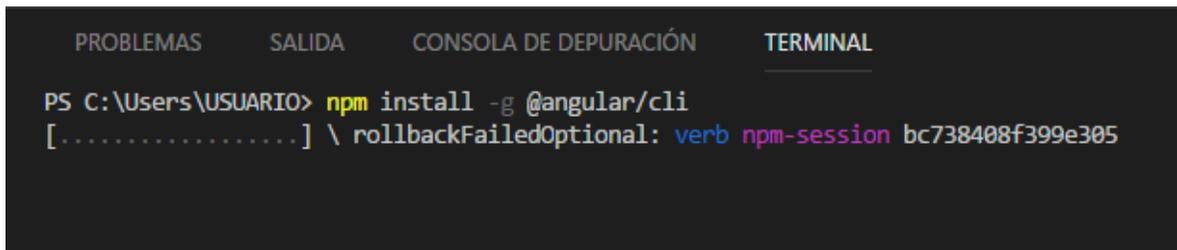


Figura 8 Angular CLIC

Terminada esta instalación el equipo está configurado para comenzar a generar un nuevo proyecto y comenzar a desarrollar en él.

#### D. Validación de instalación:

Para validar que la instalación fue exitosa se sugiere crear un proyecto en Angular básico y ejecutarlo. Se debe crear un proyecto el cual debe contar con características básicas, agregando la dependencia Routing para la navegación, también es muy importante conocer la estructura interna del árbol de directorios del proyecto.

Para la creación de un proyecto en Angular primero es necesario iniciar Visual Studio Code y presionando Ctrl + ñ, se va a abrir la consola, para situarse en el directorio en el cual va a quedar guardado el proyecto, estando en dicha ruta se escribirá el comando:

```
ng new NombreDelProyecto
```

Pero para poder generar el archivo de Routing para la navegación entre vistas se requiere agregar el parámetro `-routing`, quedando de la siguiente manera.

```
ng new NombreDelProyecto --routing
```

Seguido es procederá a situarse en la carpeta raíz del proyecto.

```
cd NombreDelProyecto
```

Allí se va a visualizar la estructura del proyecto en Angular, a continuación, una breve explicación de los archivos y directorios más importantes.

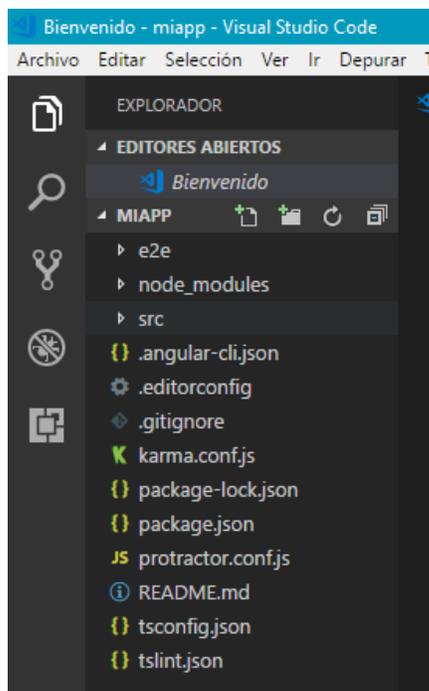


Figura 9 Estructura de archivos

De esta estructura se puede resaltar lo siguiente:

**Carpeta e2e:** Es la que contiene los archivos para la realización de pruebas.

**Carpeta node\_modules:** Donde se encuentran todas las dependencias de Angular, incluyendo las que se van instalando por medio de npm, es decir allí se encuentra todo el código de las librerías o componentes.

**Carpeta src:** Esta carpeta es muy importante ya que es donde se va a trabajar el código del proyecto, siendo más específicos en la carpeta **app**.

**Archivo .angular-cli.json:** Se encuentran las configuraciones del CLI de Angular.

**Archivo .package.json:** Se definen las dependencias, librerías con que se va desarrollando el proyecto por medio de npm.

**Archivo Index.html:** Cualquier sitio web tiene un archivo index, en Angular se puede encontrar en **src/app/app.component.html** ya que en Angular se trabaja mediante componentes.

**Archivo Styles.css:** Para los estilos globales del proyecto se puede guardar en **src/app/style.css**.

**Archivo Favicon.ico:** Este es el icono de Angular por defecto, el cual se puede reemplazar por el que más sea de agrado.

**Carpeta Assets:** Allí se van a guardar las imágenes, logos, videos y demás archivos multimedia que el proyecto requiera.

### **E. Ejecutar proyecto de Angular:**

Se iniciará el servidor Node JS de desarrollo con el cual se va a ejecutar la aplicación en el navegador, con él se visualizarán los cambios de manera inmediata, estando situados en la carpeta raíz del proyecto se escribirá el siguiente comando:

```
ng serve
```

Se procederá a acceder a la siguiente dirección para visualizar la aplicación:

```
http://localhost:4200/
```

Si por algún inconveniente o se requiere usar un puerto diferente se puede hacer con el comando:

```
ng serve -port NumeroDelPuerto
```

Al ingresar a la dirección de la aplicación web de Angular por primera vez saldrá lo siguiente, para detener el servidor, se presiona **Ctrl + c**.



*Figura 10 Proyecto Angular*

### **3.1.2 CONCEPTUALIZACIÓN DEL PROYECTO**

La conceptualización del proyecto es uno de los pasos más importantes, ya que es donde se va a ordenar y construir una idea general de lo que se quiere obtener como producto, adicional se analiza si el Framework se adapta para el desarrollo de la aplicación web del tipo SPA, en las características del proyecto definido con anterioridad. Las principales características con las que cuenta, ventajas, desventajas y hacia que tipos de proyectos está orientado, esto con el fin de determinar si los propósitos del aplicativo web a desarrollar se cumplirán en su totalidad utilizando como herramienta de desarrollo el Framework Angular.

Primero, se van a definir los alcances del proyecto y ver si en realidad Angular se ajusta a las necesidades antes de proceder a escribir líneas de código, ya que se puede llegar al punto donde las características que ofrece Angular estén limitadas

a cierto tipo de requerimientos tanto funcionales como no funcionales, perdiendo así el tiempo invertido en el desarrollo sin antes tener una visión clara de lo que se espera obtener y verificando si en realidad el framework Angular es la mejor opción.

### **A. Entrevista con el cliente.**

La primera tarea que se debe realizar en el desarrollo de software, es plantear de manera clara y precisa lo que el cliente espera obtener como resultado, los proyectos deben ser realistas desde el inicio y un desarrollador no se puede comprometer a hacer lo que está fuera de su alcance. Es por ello que primero se debe realizar una entrevista con el cliente, con preguntas generales acerca de las expectativas que tiene el cliente con el producto software a desarrollar, para llegar a un acuerdo entre las dos partes (cliente y desarrollador) antes de proceder a desarrollar el aplicativo, con el fin de evitar que en el transcurso del desarrollo, el cliente solicite nuevas funcionalidades o requisitos.

En la entrevista con el cliente se deben plantear preguntas iniciando desde lo más básico acerca de lo que será el aplicativo, preguntas enfocadas a los procesos que se manejan dentro de la empresa o persona que requieren el servicio, esto con el fin de ir planteando una idea general de lo que será el aplicativo a desarrollar y con ello poder elegir las herramientas de desarrollo que mejor se adapten a las necesidades. Algunas preguntas generales podrían ser las siguientes:

¿El aplicativo a desarrollar está orientado a Escritorio, Web, Movil, etc?

¿Cuántas personas y que roles de usuario tendrán contacto con el aplicativo?

¿Qué tareas debe realizar el aplicativo?

¿Cómo debería de ser el aplicativo?

¿Cuáles son los procesos básicos a manejar?

¿Qué datos de entrada y salida tiene cada proceso?

Las preguntas van de acuerdo al enfoque del proyecto, y según sean las respuestas del clientes las preguntas van surgiendo, con el fin generar una idea de los requerimientos del cliente para posteriormente en una próxima reunión obtener información más clara y específica.

## **B. Análisis de viabilidad del uso de Angular**

A continuación se va a presentar una descripción de las características de Angular, para posteriormente mediante un sistema de puntos evaluar si el proyecto se ajusta a las necesidades.

Angular es un framework desarrollado y mantenido por Google, ampliamente utilizado en el desarrollo del Front-end de plataformas web de alto rendimiento, en el cual se puede hacer un desarrollo full stack. Trabajando el Front-End con Angular y el Back-end con Firebase. Framework JavaScript destinado a desarrollar aplicaciones web SPA (Single Page Application), los tipos de aplicaciones web que se pueden llegar a desarrollar con Angular son:

Aplicaciones web dinámicas.

Aplicaciones web animadas.

Tiendas virtuales.

Portales web.

Gestores de contenido.

La tabla 3 fue diseñada para determinar, qué tan viable es angular en el desarrollo del aplicativo, al cual se le está realizando el estudio para el desarrollo.

Se recopilaron las principales características con las que cuenta angular dando un factor positivo a la calificación, y unas características con las que no cuenta que aportarían un factor negativo, el desarrollador califica la importancia en un rango de

1 a 3 al totalizar el puntaje si es superior a 15 se considera viable el desarrollo en angular, de lo contrario se recomienda explorar otros framework de desarrollo.

<b>CARACTERISTICA</b>	<b>SI</b>	<b>NO</b>	<b>EN ESCALA DE 1 AL 3 siendo 1 Poco importante, 2 Importante, 3 Muy importante QUE TAN IMPORTANTE ES EN SU PROYECTO LA CARACTERÍSTICA</b>
HTML dinámico	+		
Manejo de formularios y eventos.	+		
Manejo de servicios e inyección de dependencias	+		
Modelos y vistas	+		
Protección de rutas.	+		
Data Binding	+		
Directivas	+		
Enrutamiento	+		
Cliente HTTP	+		
Patrón MVC	+		
Óptimo rendimiento en aplicaciones de gran tamaño		-	
CL Debugger		-	
Aplicativo de escritorio o Android		-	
<b>TOTAL DE PUNTOS</b>			

*Tabla 3 Características de viabilidad de Angular*

### 3.1.3 IDENTIFICACIÓN DE REQUERIMIENTOS

La identificación de los requerimientos es una pieza fundamental en el desarrollo de software, del éxito de esta etapa y la correcta identificación de los requerimientos, tanto funcionales como no funcionales depende el futuro del proyecto. Por ello es

importante obtener de manera clara y precisa estos requerimientos que servirán como marco de convenio entre lo que espera el cliente y el compromiso del programador en lo que va a desarrollar.

En esta etapa se procede a identificar los requerimientos funcionales y no funcionales, aplicando diferentes instrumentos como cuestionarios, entrevistas, lluvia de ideas, etc., para posteriormente analizar esta información y de allí extraer los requerimientos funcionales y no funcionales.

- Entrevistas: Esta técnica es muy utilizada cuando se necesita recolectar opiniones, críticas o descripciones de las actividades realizadas, es fundamental la sinceridad y conocer esta información de la manera más detallada posible para analizarla y obtener de manera clara los requerimientos.
- Lluvia de ideas: Esta técnica es utilizada cuando se quieren identificar nuevas necesidades que existen en el ambiente de trabajo, obteniendo así los servicios que no se identifican de manera clara.
- Cuestionarios: Por medio de un cuestionario se realizan preguntas claras y precisas al cliente o personas que harán uso del aplicativo web, seguido analizar el cuestionario para encontrar los requisitos a partir de las respuestas suministradas.

Después de que se obtiene la información se procede a analizar sacando de allí los requerimientos funcionales y no funcionales que se pueden diferenciar de la siguiente manera.

- Requerimientos funcionales: Estos son los servicios o funcionalidades a los cuales el cliente podrá acceder. Estas funcionalidades describen las interacciones entre el sistema y el usuario o sistema externo.
- Requerimientos no funcionales: Estos son las características del sistema que no se refieren directamente a una funcionalidad en el sistema. Estos

describen aspectos visibles sin intervenir de forma directa en las funcionalidades del sistema.

### **3.1.4 CREACIÓN DE STORYBOARD**

En esta etapa, con la obtención de requerimientos lista y clara, se procederá a crear dos StoryBoard, uno para el cliente, que se encuentra conformado por una serie de imágenes que representaran la interfaz de usuario donde se muestra la navegación entre las diferentes vistas, formularios, botones, etc., y otro StoryBoard para el desarrollador, que le servirá de guía, orientándolo en la estructura de las diferentes etapas del procedimiento AngularDASPA.

#### **A. StoryBoard cliente**

Una de las maneras más recomendables y agradables para que el cliente tenga una visión clara de lo que será el proyecto, y de igual manera obtener una revisión temprana de los requerimientos del proyecto, es con la ayuda de un StoryBoard (Guion gráfico). La finalidad del StoryBoard es que el cliente pueda visualizar un boceto de lo que será el proyecto de manera temprana, logrando así detectar requerimientos adicionales, corrección de errores o aclaración de requerimientos. Luego de que el cliente de una aceptación del StoryBoard, con ello ya se puede tener plena confianza en comenzar con el desarrollo del proyecto ya que este será el contrato entre las dos partes y no generara desacuerdos en etapas intermedias o finales del proyecto.

Con los requerimientos funcionales y no funcionales, se procede a realizar un prototipado de la aplicación web que el cliente espera obtener, una serie de imágenes que describen el funcionamiento de la aplicación, con ello en poco tiempo el cliente podrá dar un visto bueno del aplicativo que se va a desarrollar, sin esperar

a que las funcionalidades se vayan desarrollando con el tiempo y luego se presenten inconvenientes en la aceptación por parte del cliente.

A pesar de que en internet existe una infinidad de herramientas para realizar un StoryBoard, se puede utilizar la que se desee o ya se conozca, para destacar una herramienta gratuita e interesante para crear la maquetación de la aplicación web es **Pencil Project**, un aplicativo gratuito en el cual de manera intuitiva se puede crear el boceto para la aceptación por parte del cliente del aplicativo que se espera desarrollar.

A continuación se observa un ejemplo de StoryBoard.

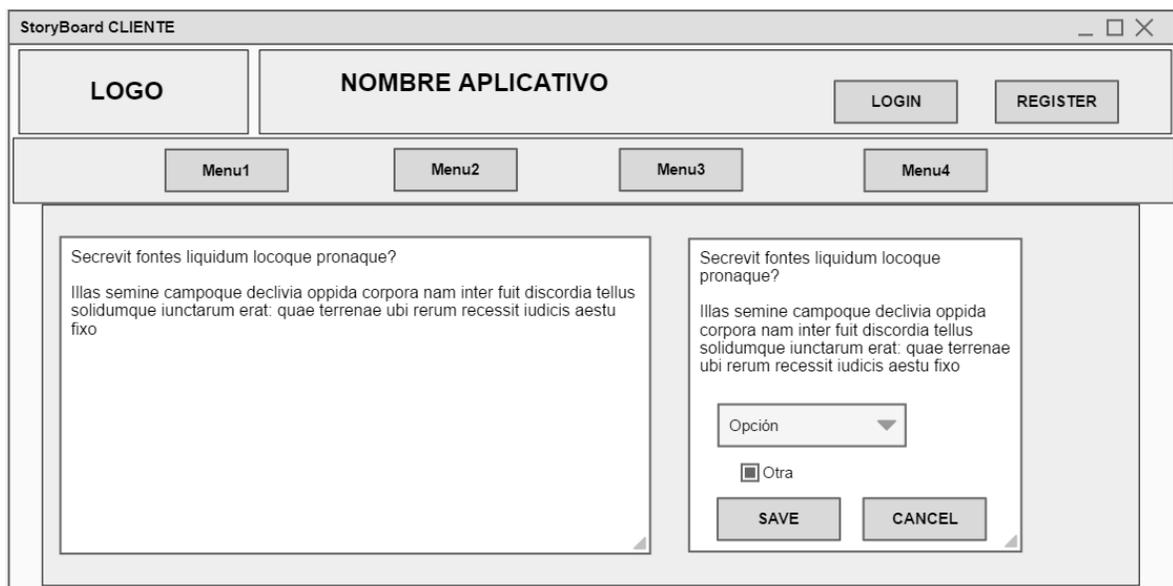


Figura 11 StoryBoard Cliente

## B. StoryBoard Desarrollador

Los desarrolladores que incursionan en el mundo de Angular, se enfrentan a la dificultad de entender cuáles son los diferentes componentes, dependencias y necesidades de almacenamiento de información, se necesita un grado de experiencia en este framework para identificar todas estas partes, por esta dificultad

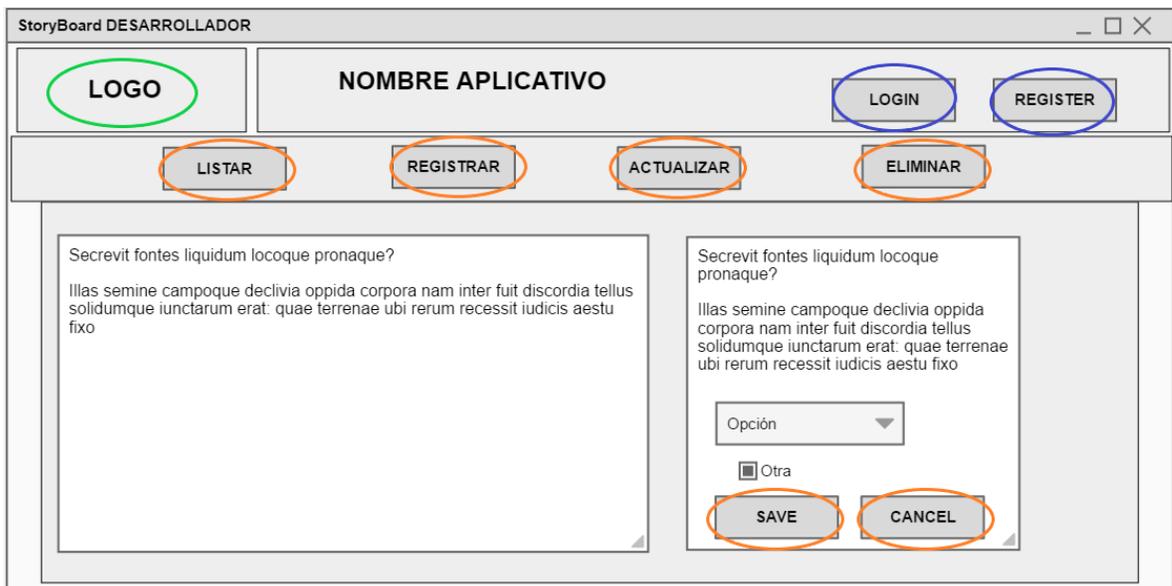
se ideó un StoryBoard enfocado al desarrollador que facilitara la identificación de componentes, dependencias y almacenamiento de datos.

Con un StoryBoard de cliente y su aceptación, se puede proceder a generar un nuevo StoryBoard enfocado al desarrollador, donde se identifiquen las piezas fundamentales que contribuirán a un correcto proceso de desarrollo, evitando posibles inconvenientes en etapas posteriores a causa de una inexacta identificación de requisitos previos al desarrollo.

El desarrollador requiere de una visión clara acerca de las herramientas tecnológicas necesarias en su proyecto y su uso, para darle cumplimiento a los requerimientos del proyecto, se propone la identificación de un StoryBoard para el desarrollador en donde se identificaran aspectos importantes como: dependencias, componentes y datos (persistentes o no persistentes).

### Storyboard Dependencias

La identificación de dependencias en el StoryBoard del desarrollador se ve reflejada de manera intuitiva gracias a la GUI que se definió en el Storyboard del cliente, identificando las diferentes secciones y lo que se quiere lograr con ellas. Por ejemplo:



Para toda la interfaz en general si requiere agilizar y desarrollar de una manera más profesional los estilos del aplicativo web se recomienda usar las librerías: bootstrap y JQuery.

**Circulo azul:** Son operaciones en la cuales se requiere la gestión de autorizaciones, que se puede realizar con la dependencia Firebase.

**Circulo naranja:** Estas operaciones CRUD, requieren de un almacenamiento de datos persistente, por ello requieren la dependencia Firebase.

**Circulo verde:** Para la parte de iconos se recomienda usar la dependencia: Font awesome, que cuenta con infinidad de iconos.

Las demás dependencias surgirán de los requerimientos específicos de cada proyecto, en la etapa instalación de dependencias se describe de manera más detallada su búsqueda.

### **Storyboard Componentes**

Angular es orientado a componentes es por ello que con el StoryBoard se identificaran de manera rápida y sencilla, ya que un componente es el encargado de realizar una función en específico dentro del aplicativo, estos se van a identificar de la siguiente manera.



Figura 13 StoryBoard Desarrollador – Componentes

**Círculo rojo oscuro:** Los componentes son los encargados de controlar una parte de la aplicación y en una interfaz GUI se pueden identificar por los que se accionan por eventos generados por el usuario adicional con las secciones que componen la interfaz.

**Cuadrado naranja:** Los componentes en Angular controlan una parte de la aplicación, por ello se requieren componentes independientes para las secciones de la interfaz GUI seleccionados.

En una aplicación web con Angular existirán tantos componentes conforme sea la complejidad de la aplicación, se explicara de manera más detallada en el apartado de componentes.

### Storyboard Datos

En la parte de los datos, se tendrá que definir si la información necesaria en el aplicativo web, se requiere almacenar de forma persistente o no. En caso de requerir el almacenamiento de información persistente se debe proceder a identificar que campos son importantes de guardar en la base de datos, una forma de reconocer

estos campos es por medio de los formularios presentes en el StoryBoard. Que se puede observar de la siguiente manera.

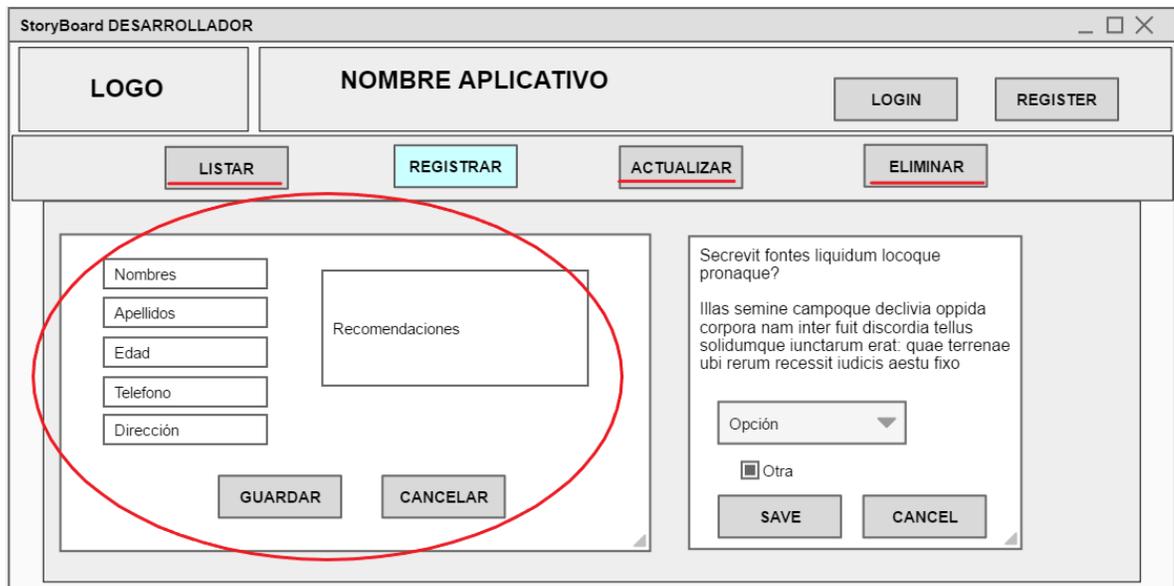


Figura 14 StoryBoard Desarrollador - Datos

**Círculo y línea rojo oscuro:** Los datos de almacenamiento persistentes se identificarán gracias a los formularios presentes en el StoryBoard, con ello ya se puede reconocer fácilmente los campos principales a almacenar, los cuales son primordiales para la base de datos.

De la base de datos se explicara más detalladamente en la sección de configuración de firebase.

A continuación se presenta un formato para el StoryBoard de desarrollador, el cual se sugiere a raíz de la complejidad de algunos proyectos, donde llegarán a existirán múltiples StoryBoard de desarrollador, por ello se hace necesario organizarlos por niveles e indicando la versión del mismo ya que se pueden generar modificaciones, allí también se realizaran anotaciones importantes y observaciones para un correcto análisis previo al desarrollo.

FORMATO DE STORYBOARD			
Nombre:	Nivel:	Versión:	Responsable:
NOTAS			NOTAS
-			
-			
-			
-			
-			
-			
Observaciones:			
Firma:	Fecha:		

Figura 15 Formato StoryBoard desarrollador

### 3.1.5 INSTALACIÓN DE DEPENDENCIAS

Una vez listos los requerimientos funcionales y no funcionales junto con el StoryBoard de desarrollador identificando las dependencias necesarias, conociendo que el Framework Angular se ajusta a las necesidades del proyecto, a su vez teniendo lista una instalación de Angular y conociendo lo básico de su estructura de archivos, se requiere instalar las dependencias necesarias para el proyecto. Cuando se dispone a desarrollar una aplicación web independientemente del framework utilizado, siempre se va a necesitar paquetes adicionales para suplir con los requerimientos de la aplicación, bibliotecas como jQuery, Bootstrap, React y por supuesto Angular. Todas estas dependencias están hechas para hacerle la vida más fácil a los desarrolladores, reutilizando código ya existente y no tener que reinventar lo que ya está hecho, evitando invertir tiempo creando desde cero el requerimiento solicitado.

Al comienzo en la sección de instalación y configuración de Angular, la primera tarea era realizar la instalación de Node.js, el entorno de ejecución de JavaScript, al instalarlo, automáticamente se podrá hacer uso de **NPM**, el manejador de paquetes por defecto para Node.js, considerado como el mayor ecosistema de bibliotecas de código abierto del mundo, y es mediante el cual se podrá instalar las dependencias necesarias para el proyecto a desarrollar. NPM cuenta con una infinidad de bibliotecas de código abierto dispuestas a utilizar, las cuales se pueden encontrar en el sitio oficial:

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

Allí se puede realizar la búsqueda de los paquetes que se requieran en el buscador.

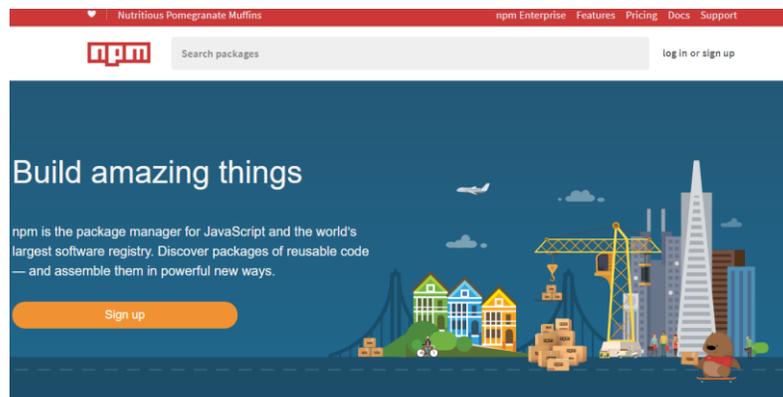


Figura 16 Página principal NPM

En el apartado de Docs (Documentación) de npm, se encontrara información más específica acerca de los paquetes npm, resolviendo preguntas tales como:

¿Cómo buscar y seleccionar paquetes?

<https://docs.npmjs.com/getting-started/searching-for-packages>

¿Cómo instalar paquetes de manera local?

En la tabla 4 se va a presentar los paquetes que más se utilizan en un aplicativo web SPA, con los requerimientos básicos.

<b>Nombre</b>	<b>Uso</b>	<b>Sintaxis</b>
Bootstrap	Marco para trabajar el front-end, fácil e intuitivo para un desarrollo web más rápido y fácil.	npm install bootstrap
JQuery	Biblioteca JavaScript rápida, con una gran cantidad de funciones para el Front-end.	npm install jquery
Font awesome	Paquete con 675 iconos pictográficos listos para utilizar.	npm i font-awesome
Firebase	Conjunto de herramientas para desarrolladores. (Base de datos, Autenticación, Hosting...)	npm i firebase
Angular fire2	Biblioteca oficial de firebase y angular.	npm i angularfire2

*Tabla 4 Dependencias más utilizadas*

### **A. Instalación dependencia**

Mediante el instalador de paquete npm se podrá instalar las dependencias necesarias para el proyecto ya sea si se va a incluir bootstrap, firebase, jquery, fuentes, o cualquier otra dependencia necesaria para el proyecto.

```
npm install NombreDelPaquete --save
```

Se indica `--save` luego del nombre del paquete para que se incluya en el `package.json` del proyecto.

De la siguiente manera se podrá instalar varios paquetes separados por espacio:

```
npm install Paquete1 Paquete2 Paquete3 --save
```

## B. Desinstalar dependencia

Para desinstalar es similar simplemente se cambia install por uninstall, indicándole de igual manera --save para eliminar el registro en el archivo.

```
npm uninstall NombreDelPaquete --save
```

## C. Ejemplo con Bootstrap y Firebase

En un ejemplo con bootstrap y firebase que son unas de las más utilizadas con Angular. Solo basta con separar por medio de espacios las dependencias a instalar.

```
npm install bootstrap jquery angularfire2 firebase
```

Luego de instalar las dependencias necesarias se requiere incluir las dependencias relacionadas al estilo en **styles** y los js en **scripts** en el archivo: **.angular-cli.json**

```
{ } .angular-cli.json ●
21 |   "styles": [
22 |     "../node_modules/bootstrap/dist/css/bootstrap.css",
23 |     "styles.css"
24 |   ],
25 |   "scripts": [
26 |     "../node_modules/jquery/dist/jquery.js",
27 |     "../node_modules/bootstrap/dist/js/bootstrap.js"
28 |   ],
```

Figura 17 Incluir dependencias Angular

### 3.1.6 CREACIÓN DE COMPONENTES

Con la ayuda del StoryBoard de desarrollador en la parte de identificación de componentes se realizará la etapa de componentes más fácil, donde se empezará a escribir el código de la aplicación en la que se está trabajando, el componente estará conformado por el código HTML, CSS Y Javascript de la aplicación, código que se escribirá dependiendo de las necesidades del aplicativo.

El Framework Angular está orientado a componentes, siendo una de sus principales características, los cuales aportan grandes, por ello es de vital importancia conocer su uso y funcionamiento para poder desarrollar aplicaciones web con Angular. Cada componente dentro de Angular es el encargado de controlar una parte de la aplicación, es decir, cada componente tiene asignada una función en específico que realiza de manera independiente, logrando así poder reutilizar estos componentes en otros desarrollos, de esta manera una aplicación web SPA con Angular estará conformada por un árbol de componentes. Al crear un componente en Angular este va a generar 3 archivos: HTML, CSS Y TS, los cuales se comunican entre sí para generar la funcionalidad requerida.

Gracias al CLI de Angular se pueden generar los componentes de forma automática el cual se encuentra conformado por bloques de código, estructura de un componente:

```
TS app.component.ts x
1 |import { Component } from '@angular/core'; 1
2
3 @Component({
4   selector: 'app-root',
5   templateUrl: './app.component.html', 2
6   styleUrls: ['./app.component.css']
7 })
8 export class AppComponent { 3
9   title = 'app';
10 }
11
```

Figura 18 Estructura de un componente

En el punto 1 se describe el import del componente, para este ejemplo el núcleo (Core) de Angular.

En el punto 2 es un objeto de metadatos el cual consta de 3 definiciones: Selector: Nombre por el cual se invoca para llamar el componente, define en que etiqueta html es en la cual renderizara el componente. Template: define el código html, allí se indica la ruta del archivo template o plantilla. Referencia de estilo: Se define los estilos CSS, y aquí se indica la ruta del archivo.

En el punto 3 es en la cual se va a definir la lógica del componente, por defecto se define solo el nombre.

### A. Identificación

Antes de proceder a crear los componentes primero con ayuda del StoryBoard de desarrollador, se va a proceder a identificar que componentes se requieren para el aplicativo web.

Un componente se encarga de controlar una parte de la aplicación web, es por ello que cada una de las secciones del StoryBoard va a corresponder a un componente en Angular.



Figura 19 Identificación de componentes

En el ejemplo anterior se necesitaría generar los siguientes componentes: Home, Cabecera, Navbar, Login, Create, Edit, Update y View.

Por cada funcionalidad o por cada vista se procederá a generar un componente en Angular.

## B. Creación de los componentes

Para la creación de los componentes se va a utilizar Angular CLI, generando estos componentes de manera automática por medio del comando:

```
ng generate component NombreDelComponente
```

Para abreviar se puede escribir de la siguiente manera:

```
ng g c NombreDelComponente
```

Siempre es recomendable guardar los componentes dentro de una carpeta dedicada solo a componentes, por ejemplo para la vista de contacto:

ng g c componentes/contacto

```
└─ componentes
  └─ contacto
     # contacto.component.css
     <> contacto.component.html
     TS contacto.component.spec.ts
     TS contacto.component.ts
```

Figura 20 Creación de un componente

Este es el resultado de la creación de un componente, genera archivos muy importantes los cuales juntos van a ser los encargados del funcionamiento de una parte de la aplicación web. Donde se utilizara HTML, CSS Y TypeScript para programar el componente y así encargarse de una parte la aplicación web que se esté desarrollando.

## HTML

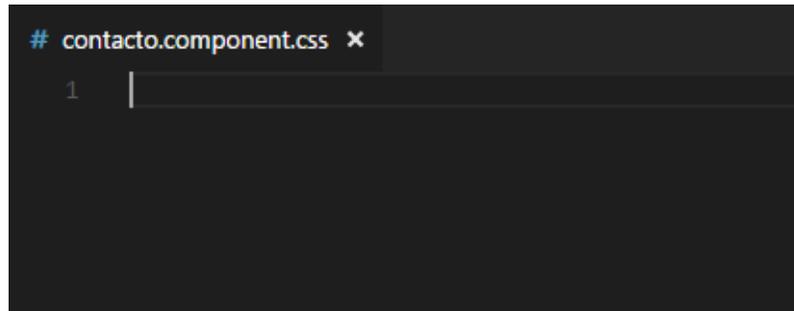
En este archivo se procederá a escribir todo el código HTML necesario para darle forma al sitio web, utilizando lo que requiera el componente, o lo que se desee visualizar en el mismo. Lo que se conoce como maquetación.

```
<> contacto.component.html x
1  <p>
2  |   contacto works!
3  </p>
4  |
```

Figura 21 HTML de un componente

## CSS

En el archivo CSS, se escribirán los estilos necesarios a utilizar solamente en el componente que se esté trabajando.



```
# contacto.component.css x
1 |
```

Figura 22 CSS de un componente

## TYPESCRIPT

El siguiente archivo es TypeScript, el cual se ha mencionado anteriormente allí se escribirán los métodos que sean necesarios para darle la funcionalidad al componente.



```
TS contacto.component.ts x
1 | import { Component, OnInit } from '@angular/core';
2 |
3 | @Component({
4 |   selector: 'app-contacto',
5 |   templateUrl: './contacto.component.html',
6 |   styleUrls: ['./contacto.component.css']
7 | })
8 | export class ContactoComponent implements OnInit {
9 |
10 |   constructor() { }
11 |
12 |   ngOnInit() {
13 |   }
14 |
15 | }
```

Figura 23 TypeScript de un componente

### C. Incluir componente en la página principal

Luego de trabajar en los componentes es importante conocer la manera de llamarlos y poder visualizarlos en el aplicativo web.

Para que se incluya un componente en la página principal, sin necesidad de asignarle una ruta y que se va a visualizar siempre en las páginas, es necesario incluir estos componentes en el archivo: **app.component.html**, el cual se modificará dejando solo la sentencia `<router-outlet></router-outlet>` que es el encargado de inyectar el contenido de la ruta que se llame. Un ejemplo de ello es un componente para la barra de navegación que por lo general siempre en todas las vistas tiene que aparecer, para la asignación se realiza de la siguiente manera en la figura 24.

```
app.component.html ●
14 <div class="container">
15   <h3>Ejemplo de Inclusion de un Componente</h3>
16   <hr>
17   <app-navegacion></app-navegacion>
18
19 </div>
20
21 <router-outlet></router-outlet>
22
```

Figura 24 Inyección de un componente

Con ello ya se puede ir trabajando en el código del aplicativo web, en lo que se refiere al código HTML, CSS y lógica necesaria, para irle dando forma al desarrollo del aplicativo web.

Gracias a `<router-outlet></router-outlet>` se podrá acceder a rutas que tendrán un componente asignado, esto se realizará mediante Routing, tema que se abordará en el siguiente paso.

## D. Uso de Data Binding

El Data Binding o enlace de datos, es la manera de crear la comunicación entre el código Typescript de un componente y el código HTML del template. Enlaces que se pueden crear de distintas maneras como se visualiza en la tabla 5:

TIPO	NOMBRE	USO	SINTAXIS
One Way	Interpolación	Incluir dentro del HTML, variable Typescript	<code>{{ propiedad/objeto/expresión }}</code>
One Way	Property Binding	Incluir valores de la fuente de datos a la vista.	<code>[ atributodeelementoHTML ] = “expresión “</code>
One Way	Event Binding	Enlace de la vista a la fuente de datos.	<code>evento=“nombreMetodo()”;</code>
Two-Way Binding		Enlaces en ambos sentidos	<code>[( directiva )] = “nombredelapropiedad”</code>

Tabla 5 Data Binding

## E. Uso de Directivas

Si en el proyecto se presenta la necesidad de crear, formatear e interaccionar con elementos HTML en el DOM, se puede proceder a hacer uso de **DIRECTIVAS**, siendo la función principal: crear o modificar un comportamiento nuevo al ser llamadas. Ver Anexo 6.1.

## F. Uso de Pipes

Los pipes en Angular son elementos de código que permiten modificar la salida de los datos, algunos de ellos son: Pipe date, Uppercase y Lowercase, Decimal, Currency, i18nSelect, de igual manera se pueden crear pipes con el requerimiento deseado. En caso de requerir alguno de ellos en la aplicación web se puede ver el uso y manejo de ellos en la documentación de Angular:

<https://angular.io/guide/pipes>

### 3.1.7 CREACIÓN DE RUTAS

Terminados los componentes se requiere de un sistema de navegación mediante el cual se puedan acceder a estos componentes gracias a dirección por ello este paso se considera fundamental ya que una aplicación web por básica que sea, estará conformada por dos o más vistas, las cuales se identificarán por una dirección para poder acceder a ellas y ver su contenido, con ello se hace necesario la utilización navegación entre las diferentes vistas que conforman el aplicativo, realizar redirecciones, pasar parámetros y así mismo proteger rutas que no se pueden dejar como públicas, ya que se comprometería la seguridad de la información. Para poder estructurar la navegación entre los distintos componentes se requiere de un sistema de rutas en el cual se pueda declarar y asignar contenido a ellas, en Angular esto se puede realizar mediante su característica conocida como Routing.

Angular Routing, es el encargado de permitir la navegación de una vista a otra, decidir cuándo se muestra un componente u otro, el cual tendrá una ruta asignada.

Para poder hacer uso de las rutas se va a dirigir al archivo que se creó al inicio del proyecto al indicarle la dependencia --routing, archivo que se encontrara en:

`src/app/app-routing.module.ts,`

```
TS app-routing.module.ts
1 import { NgModule } from '@angular/core';
2 import { Routes, RouterModule } from '@angular/router';
3
4 const routes: Routes = [
5
6 ];
7
8 @NgModule({
9   imports: [RouterModule.forRoot(routes)],
10  exports: [RouterModule]
11 })
12 export class AppRoutingModule { }
13
```

Figura 25 Estructura de Routing en Angular

Del archivo se puede destacar lo siguiente:

En el **Punto 1** se realiza la importación del módulo de Router que se instaló cuando se creó el proyecto.

En el **Punto 2** se declaran todas las rutas que se van a utilizar por medio de un Array.

### A. Declaración de Rutas

Para la declaración de las rutas, se va a necesitar declararlas en el array: Routes, en el cual se le asigna un nombre de la ruta y el componente a mostrar en dicha ruta, de la siguiente manera.

```
{path: 'NombreRuta', component: NombreComponente}
```

Se pueden ingresar las rutas que se requieran separados por comas, un ejemplo más claro es el siguiente.

```
const routes: Routes = [
  {path: 'inicio', component: InicioComponent},
  {path: 'contacto', component: ContactoComponent}
];
```

Figura 26 Rutas en Angular

Si no se genera la importación por sí sola de los componentes es necesario realizar la importación de los componentes, en los **imports** de manera manual.

```
TS app-routing.module.ts x
1  import { NgModule } from '@angular/core';
2  import { Routes, RouterModule } from '@angular/router';
3  import { InicioComponent } from '../inicio/inicio.component';
4  import { ContactoComponent } from '../contacto/contacto.component';
5
6
7  const routes: Routes = [
8    {path: 'inicio', component: InicioComponent},
9    {path: 'contacto', component: ContactoComponent}
10 ];
11
```

Figura 27 Imports de Routing

Con ello ya se podrá visualizar el contenido de estos dos componentes en la dirección:

```
http://localhost:4200/inicio
http://localhost:4200/contacto
```

Es importante tener `<router-outlet></router-outlet>` en **app.component** para que funcionen las rutas.

```
<> app.component.html ●
14 | <div class="container">
15 |   <router-outlet></router-outlet>
16 | </div>
```

Figura 28 Router-Outlet en Angular

## B. Pasar parámetros

Si se requiere pasar parámetros a través de estas rutas se puede realizar de la siguiente manera:

```
{path: 'NombreRuta/:Parametro', component: NombreComponente}
```

```
const routes: Routes = [  
  {path: 'inicio', component: InicioComponent},  
  {path: 'contacto', component: ContactoComponent},  
  {path: 'persona/:id', component: PersonaComponent}  
];
```

Figura 29 Enviar parámetro en Routing.

Seguido en el componente persona se tendrá que configurar el archivo TypeScript para recibir el parámetro. En este caso sería el archivo persona.component.ts.

Al realizar la importación de [Router](#), [ActivatedRoute](#), [Params](#) desde Angular Router. Inicializándola las variables en el constructor y con la funcionalidad [snapshot.params](#) capturar el id. De la siguiente manera.

```
TS persona.component.ts x  
1 import { Component, OnInit } from '@angular/core';  
2 import { Router, ActivatedRoute, Params } from '@angular/router';  
3  
4 @Component({  
5   selector: 'app-persona',  
6   templateUrl: './persona.component.html',  
7   styleUrls: ['./persona.component.css']  
8 })  
9 export class PersonaComponent implements OnInit {  
10   idPersona: String;  
11   constructor(  
12     private route: ActivatedRoute,  
13     private router: Router  
14   ) { }  
15  
16   ngOnInit() {  
17     this.idPersona = this.route.snapshot.params['id'];  
18   }  
19 }
```

Figura 30 Capturar parámetro en Routing

### C. Creación de un enlace a una Ruta

En código HTML se acostumbra a realizar los enlaces mediante la etiqueta href, en angular las cosas cambian un poco, acá se realizará mediante RouterLink. Esto se incluirá en los archivos HTML del componente que sea necesario.

```
<a routerLink="contacto"> Contacto </a>
```

Un ejemplo del componente de barra de navegación sería el siguiente.

```
<> navegacion.component.html ●
1  <div id="container">
2
3    <a class="nav-link" routerLink="/">Inicio
4      <span class="sr-only">(current)</span>
5    </a>
6
7    <a class="nav-link" routerLink="/contacto">Contacto
8      <span class="sr-only">(current)</span>
9    </a>
10
11 </div>
```

Figura 31 Creación de enlace con Angular

De esta manera ya se puede acceder a las rutas creadas anteriormente mediante estos enlaces que se han creado.

### 3.1.8 CONFIGURACIÓN FIREBASE

La etapa de configuración de Firebase se requiere cuando en el StoryBoard del desarrollador se detecta el almacenamiento de datos persistentes, en la configuración de Firebase se realizará la configuración para autenticación de usuarios y manejo de base de datos NOSQL.

Cuando se está desarrollando una aplicación web, en la gran mayoría de los casos se requiere hacer uso de una base de datos, en la cual se pueda almacenar información importante para la empresa o el proyecto que se esté desarrollando. De esta manera se debe asegurar que la información se encuentre almacenada en un lugar seguro que no dé lugar a pérdidas o inconsistencias en la misma.

Es por ello que es importante realizar una conexión a un servidor de una base de datos, la cual permita almacenar la información de forma persistente, y así interactuar con la información realizando tareas de crear, editar, eliminar y actualizar, esto se puede llevar a cabo gracias a los métodos HTTP que utiliza Angular.

Firebase es una plataforma bajo el respaldo de Google, alojada en la nube, que cuenta con un conjunto de herramientas para desarrolladores, facilitando la creación de APPS de una manera más rápida y con una alta calidad. Una de estas herramientas es una base de datos en tiempo real la cual se utilizara para el desarrollo. También se puede realizar autenticación de usuarios, servicio de hosting gratuito, entre muchas herramientas interesantes que valen la pena probar. Para comenzar se ira a la página oficial de Firebase: <https://firebase.google.com/>



Figura 32 Pagina Firebase

Se requiere tener una cuenta de Google para poder acceder a los servicios que ofrece, teniendo una cuenta Google, se inicia sesión y luego clic en ir a la consola.

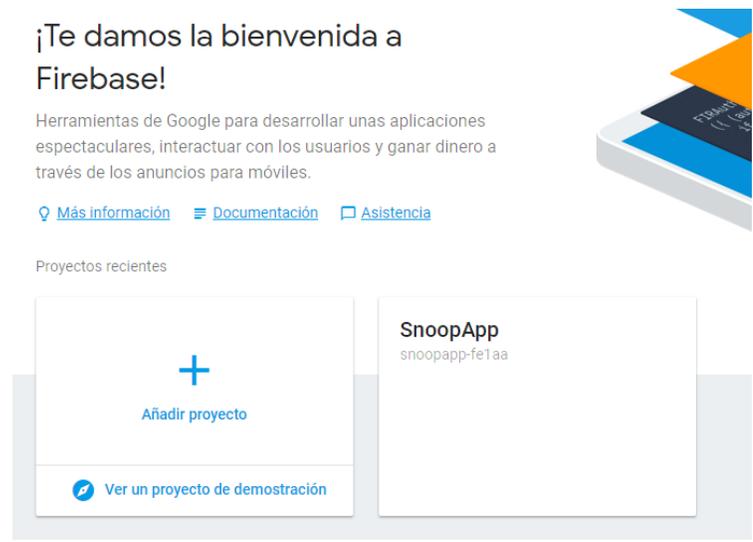


Figura 33 Proyecto Firebase

Estando situados en la consola se va a proceder a añadir un proyecto nuevo.

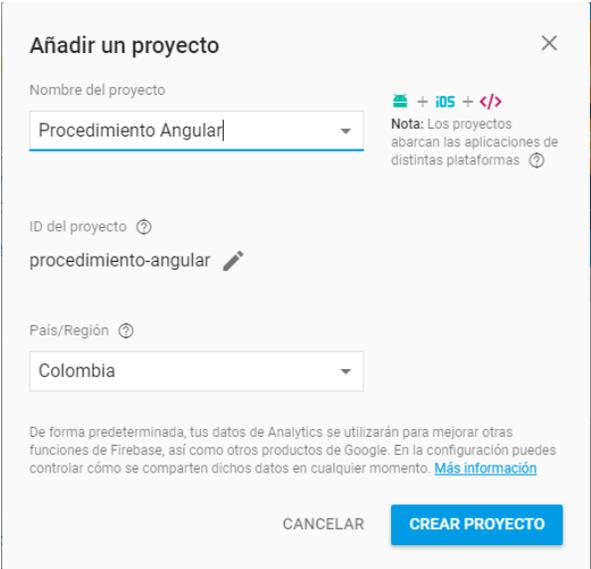


Figura 34 Crear un proyecto en Firebase

Allí se va a asignar un nombre y la región a la cual se pertenece, o donde se esté desarrollando el proyecto. Cuando termine, se selecciona continuar y saldrá la pantalla con todos los productos que se pueden utilizar.

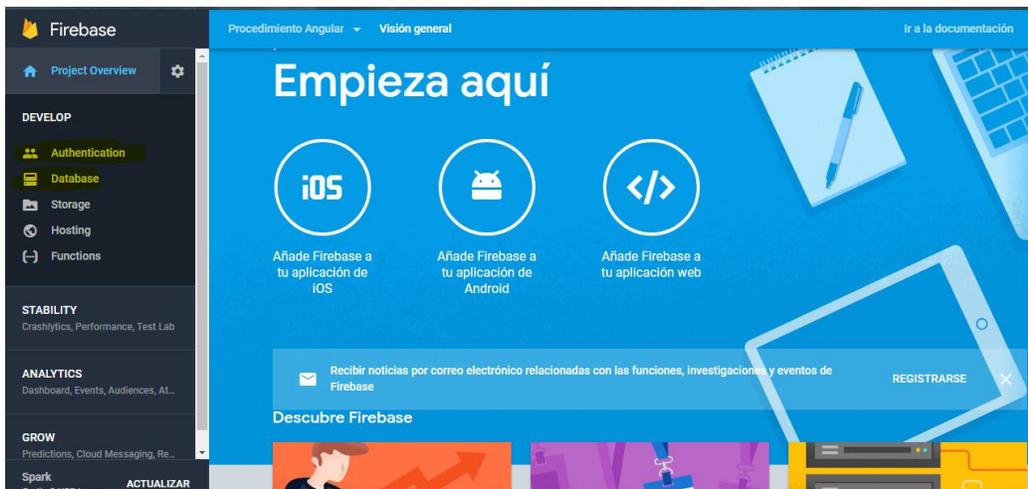


Figura 35 Añadir un proyecto en Firebase

Se especificarán dos herramientas importantes para el procedimiento, la primera es el apartado **Authentication**, herramienta en la que se puede gestionar las autorizaciones de inicio de sesión del aplicativo web. Seguido se utilizará también **Database**, base de datos en tiempo real alojada en la nube, en la cual se va a guardar la información del aplicativo web.

### A. Añadir Firebase a la Aplicación web.

Para conectar la aplicación con la base de datos Firebase, estando situados en la página principal, se dará clic en el apartado que dice Añadir Firebase a la aplicación web.

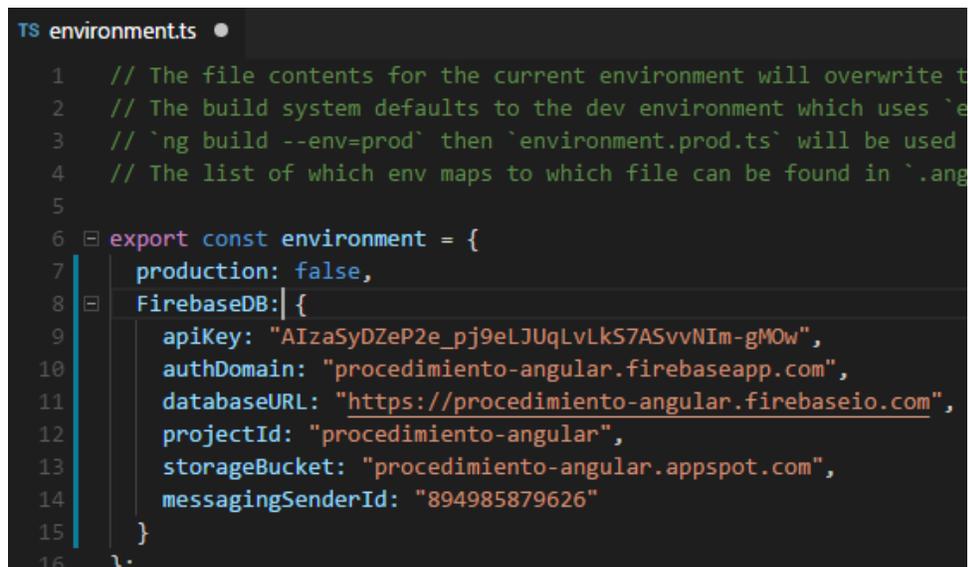


Figura 36 Configuración Firebase

Se procederá a copiar los datos seleccionados en amarillo, los cuales se van a incluir en el aplicativo web de Angular. Dentro del proyecto se va a dirigir al archivo:

```
src/environments/environment.ts
```

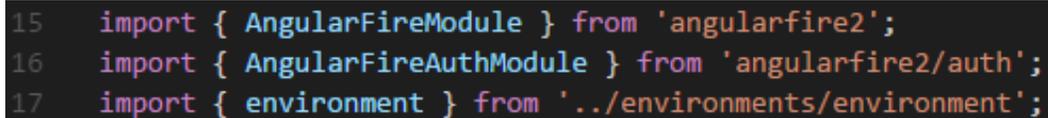
Quedando de la siguiente manera:



```
TS environment.ts ●
1 // The file contents for the current environment will overwrite t
2 // The build system defaults to the dev environment which uses `e
3 // `ng build --env=prod` then `environment.prod.ts` will be used
4 // The list of which env maps to which file can be found in `ang
5
6 export const environment = {
7   production: false,
8   FirebaseDB: {
9     apiKey: "AIzaSyDZeP2e_pj9eLJUqLvLkS7ASvvNIm-gMOW",
10    authDomain: "procedimiento-angular.firebaseio.com",
11    databaseURL: "https://procedimiento-angular.firebaseio.com",
12    projectId: "procedimiento-angular",
13    storageBucket: "procedimiento-angular.appspot.com",
14    messagingSenderId: "894985879626"
15  }
16 };
```

Figura 37 Añadir Configuración a Angular

Lo siguiente se tendrá que incluir en el archivo **app.module.ts** para poder utilizar la base de datos. En el cual en la sección de import, se incluirá lo siguiente.



```
15 import { AngularFireModule } from 'angularfire2';
16 import { AngularFireAuthModule } from 'angularfire2/auth';
17 import { environment } from '../environments/environment';
```

Figura 38 Incluir Firebase en module.ts

Más abajo del archivo se tendrán que inyectar en el array de Imports, de la siguiente manera.

```

imports: [
  BrowserModule,
  FormsModule,
  AppRoutingModule,
  AngularFireAuthModule,
  AngularFireModule.initializeApp(environment.firebaseConfig),
  FlashMessagesModule
],

```

Figura 39 Inyección Firebase en module.ts

Con esto ya quedaría configurada la conexión a Firebase desde Angular.

En este paso se van a dividir por dos caminos uno para la gestión de las autorizaciones, en donde se usara Authentication y otro para realizar un CRUD de tareas utilizando la herramienta Database.

## B. Authentication

Este paso se realizará si en el aplicativo se va a un sistema de gestión de usuarios, para ello se va a seleccionar en el panel izquierdo la opción Authentication donde se observará la siguiente pantalla, donde existen muchas opciones con respecto al manejo de usuarios con Firebase.



Figura 40 Gestión de usuarios Firebase

Allí se va a seleccionar configurar el método de inicio de sesión, y allí se seleccionan los métodos de inicio de sesión que se requieran.

Authentication	
USUARIOS	MÉTODO DE INICIO DE SESIÓN
Proveedores de inicio de sesión	
Proveedor	Estado
✉ Correo electrónico/contraseña	Inhabilitado
☎ Teléfono	Inhabilitado
🌐 Google	Inhabilitado
🎮 Play Juegos	Inhabilitado
📘 Facebook	Inhabilitado
🐦 Twitter	Inhabilitado
🐙 GitHub	Inhabilitado
👤 Anónimo	Inhabilitado

Figura 41 Método de inicio de sesión Firebase

Como recomendación el método de correo y Google, no requieren configuración por ser de la misma plataforma.

### C. Database

Si en los requerimientos es necesario almacenar información de forma persistente en el aplicativo web, se procederá a utilizar la herramienta de base de datos NOSQL de Firebase. Primero dirigirse al apartado Database, en el panel izquierdo y aparecerán las dos opciones para Base de datos que ofrece Firebase, muy similares por cierto.

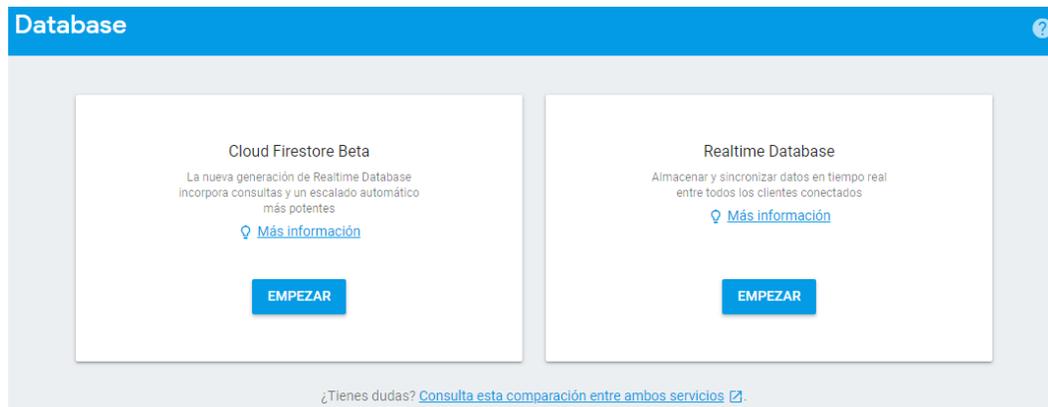


Figura 42 Selección base de datos Firebase

Se elige la opción que más se adapte al proyecto, aunque en realidad son similares, en este caso se seleccionara Cloud Firestore, la nueva generación con mejoras incorporadas, para grandes proyecto usar la versión más estable Realtime Database.



Figura 43 Reglas de seguridad Firebase

Seguido al finalizar la configuración ya estará lista la base de datos en Firebase.

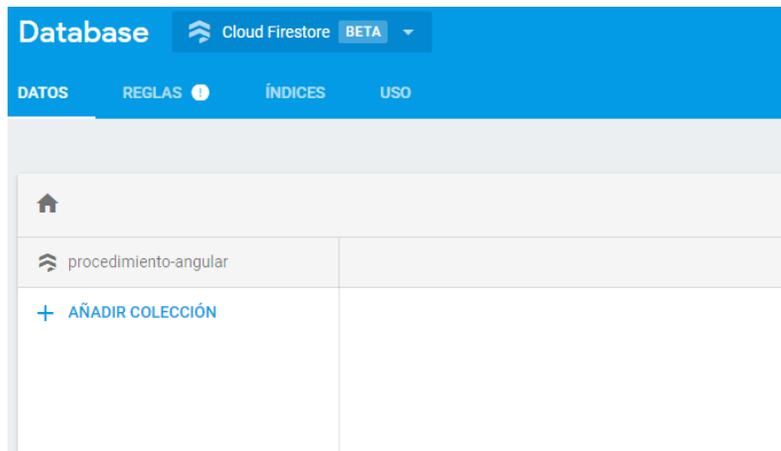


Figura 44 Panel control base de datos Firebase.

Algo muy importante en la sección de reglas es configurar que los usuarios con acceso a la base de datos, puedan leer y escribir en ella, por ello es importante asegurarse que los permisos de lectura y escritura se encuentran habilitados, como en el siguiente pantallazo.

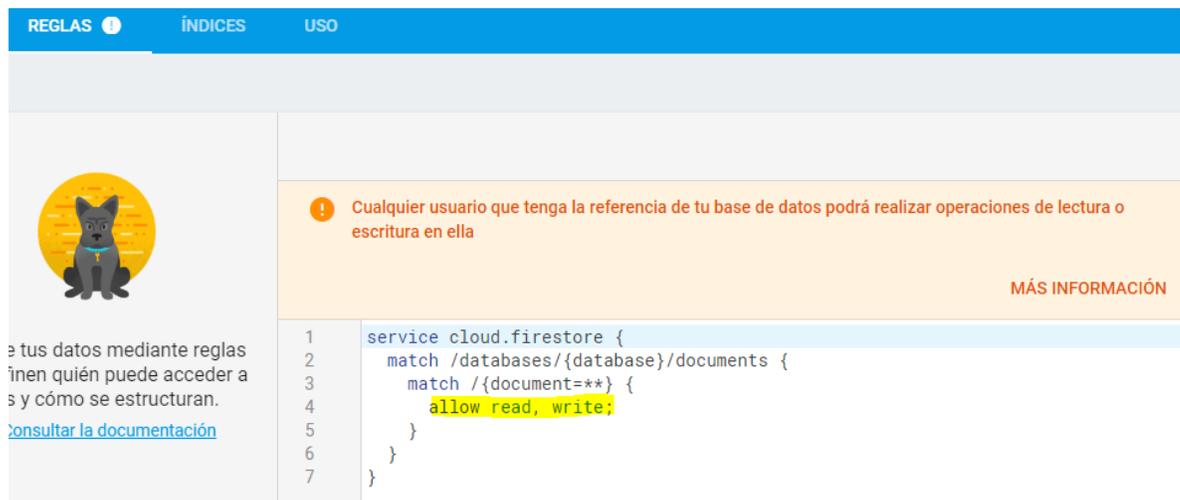


Figura 45 Permisos base de datos Firebase

### 3.1.9 GESTIÓN DE AUTORIZACIONES

Se realizará la creación de un servicio el cual contenga los métodos para gestionar el inicio de sesión y protección de rutas.

#### A. Creación del Servicio:

Una de las características más importantes de Angular es la creación de un Service, encargado de compartir información con otros componentes.

Se procederá a crear una carpeta llamada servicios y con el siguiente comando se creara el servicio: **s** que corresponde a la abreviatura de servicie.

```
ng g s services/auth
```

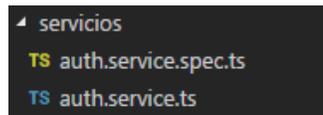


Figura 46 Creación del service

Generando el archivo spec, para pruebas y **auth.service.ts** donde se programaran los métodos necesarios, en este caso métodos para gestión de usuarios.

Lo primero es realizar las importaciones necesarias del Angular Fire, la herramienta de Map de la librería Rxjs, seguido inyectar en la clase constructor, el Auth de AngularFire. Quedando de la siguiente manera:

```
import { AngularFireAuth } from 'angularfire2/auth';  
import * as firebase from 'firebase/app';  
import 'rxjs/add/operator/map'
```

Seguido de ello en el constructor es necesario declarar el Auth de Angular Fire.

```
constructor(  
  public afAuth: AngularFireAuth  
) { }
```

Con ello ya se pueden crear los métodos necesarios para las autorizaciones. Debajo de la clase constructor, se va a observar como sería el inicio de sesión con Google, Correo, registro de usuario, conocer quien esta logado, y cerrar sesión, a manera de ejemplo.

### **Registrar un usuario:**

```
RegistrarUsuario(email: string, pass: string){  
  return new Promise ((resolve, reject) =>{  
  
    this.afAuth.auth.createUserWithEmailAndPassword(email,pass)  
      .then( userData => resolve (userData),  
        err => reject (err));  
  });  
}
```

### **Iniciar sesión correo electrónico:**

```
loginEmail(email: string, pass: string){  
  return new Promise ((resolve, reject) =>{  
  
    this.afAuth.auth.signInWithEmailAndPassword(email,pass)  
      .then( userData => resolve (userData),  
        err => reject (err));  
  });  
}
```

### Iniciar sesión Google:

```
loginGoogle(){
    return this.afAuth.auth.signInWithPopup(
    new firebase.auth.GoogleAuthProvider());
}
```

### Iniciar sesión Facebook:

```
loginFacebook(){
    return this.afAuth.auth.signInWithPopup(
    new firebase.auth.FacebookAuthProvider());
}
```

### Obtener usuario:

```
getAuth(){
    return this.afAuth.authState.map (auth =>
    auth);
}
```

### Cerrar sesión:

```
CerrarSesion() {
    return this.afAuth.auth.signOut();
}
```

Siempre que se cree un servicio se debe importar al archivo **app.module.ts** para que Angular lo reconozca.

```
import { AuthService } from './servicios/auth.service';
```

Seguido se debe asignar en el Array providers.

```
providers: [AuthService]
```

### **B. Llamar el servicio dentro de un Componente:**

Para realizar el llamado del servicio que se ha creado anteriormente, se procederá a ubicarse en el componente que se necesita realizar el llamado, y en el archivo **ts**, se va a realizar la invocación del servicio de la siguiente manera.

Se requiere llamarlo desde el Import:

```
import { AuthService } from './servicios/auth.service';
```

Inicializarlo en el constructor:

```
constructor(  
  public authservice: AuthService  
) { }
```

Y con ello ya se podrán utilizar los métodos creados en el servicio y hacer uso de ellos de manera local en cada componente que se requiera. De la siguiente manera:

```
this.authservice. NombreDelMetodo
```

### **C. Llamar un método desde el HTML del componente:**

Para llamar el método en el correspondiente componente donde lo necesitemos, este será en el archivo HTML del componente de la siguiente manera:

```
(click)="NombreDelMetodo"
```

Un pequeño ejemplo:

```
<a class="nav-link" href="#" (click)="LoginFacebook()">Salir</a>
```

Con esto al seleccionar esta etiqueta se llamara el método del componente, el cual está utilizando un servicio.

#### D. Protección de rutas

La protección de rutas es un aspecto muy importante para la seguridad del aplicativo web, los Guards (Protectores de rutas) son los encargados de informarle al enrutador (Routing), si debe o no permitir la navegación a la ruta a la que el usuario está intentando acceder. Para la creación de un Guard, se realiza con el siguiente comando:

```
ng g g guards/NombreDelGuard
```

La propiedad más usada, para la protección de rutas es **CanActive**, la cual vigila si un determinado usuario puede acceder o no a una ruta.

Para implementar dicho método se hace necesario la importación de los siguientes métodos:

```
import { CanActivate, ActivatedRouteSnapshot, RouterStateSnapshot, Router } from '@angular/router';
import { Observable } from 'rxjs/Observable';
import 'rxjs/add/operator/do';
import 'rxjs/add/operator/map';
import 'rxjs/add/operator/take';

import { AngularFireAuth } from 'angularfire2/auth';
import { AuthService } from '../servicios/auth.service';
```

Inicializar un constructor con los siguientes métodos:

```

constructor(
  private router: Router,
  private afAuth: AngularFireAuth,
  private authService: AuthService
) { }

```

El método **CanActivate** quedaría de la siguiente manera para validar que un usuario pueda acceder a una ruta.

```

canActivate(
  next: ActivatedRouteSnapshot,
  state: RouterStateSnapshot): Observable<boolean> |
  Promise<boolean> | boolean {
  return this.authService.afAuth.authState
    .take(1)
    .map(authState => !! authState)
    .do( UsuarioRegistrado => {
      if(!UsuarioRegistrado){
        this.router.navigate(['/PaginaDeInicio']);
      }
    })
  }

```

En donde en el if de UsuarioRegistrado si no ha iniciado sesión, lo redirija a la página de inicio.

Y así como en los servicios es necesario incluir los Guard en el archivo **app.module.ts**,

```

import { AuthGuard } from './guards/auth.guard';

```

De igual manera se agrega al array providers, donde anteriormente se había asignado AuthService.

```

providers: [AuthService, AuthGuard]

```

Ahora en el archivo donde se encuentran las rutas, **app-routing.module.ts** se realizara la importación del Guard.

```
import { AuthGuard } from './guards/auth.guard';
```

Para posteriormente asignarlo a la ruta privada, o solo para usuarios registrados. De la siguiente manera.

```
{path: 'PaginaPrivada', component: ComponentePaginaPrivada,  
canActivate: [AuthGuard]},
```

De esta manera la ruta que se le asigne el CanActivate solo la podrán visualizar los usuarios que hayan iniciado sesión.

### 3.1.10 REALIZACIÓN CRUD

Se realizará la creación de una interface o model, en la cual se van a definir propiedades necesarias a la hora de interactuar con la base de datos de firebase, seguido creando el servicio para poder crear, ver, editar, eliminar y actualizar los datos.

#### A. Creación de una interfaz

La creación de un modelo o interfaz es importante cuando se va a interactuar con datos los cuales se van a almacenar de manera persistente, en un modelo se van a indicar los campos que se requieren junto con el tipo de datos.

Para la creación de un modelo con el Angular CLI, es con el siguiente comando.

```
ng g i NombreInterfaz
```

Un ejemplo de un modelo para los datos de una Persona sería el siguiente.

```
ng g i models/persona
```

Para tener organizado el proyecto es recomendable guardarlo en una carpeta llamada modelos. Allí la estructura sería la siguiente.

```
export interface Persona {
  public id: number;
  public nombre: string;
  public apellidos: string;
  public ciudad: string;
  constructor(id: number, nombre: string,
    apellidos: string, ciudad: string) {
    this.id = id;
    this.nombre = nombre;
    this.apellidos = apellidos;
    this.ciudad = ciudad;
  }
}
```

## B. Creación del Servicio:

Se procederá a crear la carpeta llamada servicios y con el siguiente comando se creará el servicio: **s** que corresponde a la abreviatura de service el cual contendrá los métodos del CRUD.

```
ng g s services/NombreServicio
```

Luego de crear el servicio se hace necesaria la importación de la interfaz que se acaba de crear, junto con los módulos necesarios para los métodos que se van a utilizar. Son los siguientes:

```
import { NombreInterfaz } from
'../CarpetaInterfaz/NombreInterfaz';
import { AngularFirestore, AngularFirestoreColletion,
AngularFirestoreDocument } from 'angularfire2/firestore';
import { Observable } from 'rxjs/Observable';
```

Ahora se requiere la creación de las propiedades con ayuda de las colecciones y documentos de AngularFireStore de la siguiente manera:

```
NombreServicioColletion:
AngularFirestoreColletion<NombreInterfaz>;
    NombreServicioDoc:
AngularFirestoreDocument<NombreInterfaz>;
    NombreServicio: Observable<NombreInterfaz[]>;
    NombreServicio: Observable<NombreInterfaz>;
```

De igual manera es necesario inyectar en el constructor el AngularFirestore, e incluir la colección dentro de dicha propiedad para cuando arranque la aplicación no genere errores.

```
constructor(
    private NombreCons: AngularFirestore ) {
    this.NombreServicioColletion=this.NombreC('NombreServicio', ref => ref);
}
```

Ya con ello se puede proceder a programar los métodos necesarios, como crear, visualizar, editar, eliminar y actualizar. A continuación se describirá el código de cada uno de los métodos.

### Agregar información:

```
NewDatos(NuevoDato: NombreInterfaz){
    This.NombreServicioColletion.add(Nuevo);
}
```

### Recuperar información:

```
getAllDatos():Observable< NombreInterfaz []>{
    this.datos =
    this.NombreServicioColletion.snapshotChanges()
    .map(changes => {
        return changes.map(action => {
            const data = action.payload.doc.data() as
NombreInterfaz;
            data.id = action.payload.doc.id;
            return data;
        });
    });
    return this.datos;
}
```

### Buscar un dato:

```
getOneDato(idBuscar: string){
    this.NombreServicioDoc = this.NombreCons.doc<
NombreInterfaz >(`datos/${idDato}`);
    this.dato =
    this.NombreServicioDoc.snapshotChanges().map(action=>{
        if(action.payload.exists === false){
            return null;
        }else{
            const data = action.payload.data() as
NombreInterfaz;
            data.id = action.payload.id;
            return data;
        }
    });
    return this.dato;}
}
```

### Actualizar información:

```
updateDato(actualizarDato: NombreInterfaz){
  this.NombreServicioDoc = this.
NombreCons.doc(`datos/${dato.id}`);
  this.NombreServicioDoc.update(dato);
}
```

### Eliminar información:

```
deleteDato(eliminarDato: NombreInterfaz){
  this.NombreServicioDoc =
this.afs.doc(`datos/${dato.id}`);
  this.NombreServicioDoc.delete();
}
```

Al igual que en la autenticación, siempre que se cree un servicio se debe importar al archivo **app.module.ts** para que Angular lo reconozca.

```
import { NombreServicio } from './servicios/NombreServicio;
```

Seguido se debe asignar en el Array providers.

```
providers: [NombreServicio]
```

Lo siguiente es configurar el componente donde se va a hacer uso del servicio como de la interfaz.

### C. Llamar el Modelo y Servicio dentro del Componente:

Para realizar el llamado del modelo junto con el servicio que se ha creado anteriormente, se procederá a ubicarse en el componente que se necesita realizar el llamado, y en el archivo **ts**, se va a realizar la invocación del servicio de la siguiente manera.

Se requiere primero llamarlos mediante Import:

```
import { NombreInterface } from '../..//NombreCarpeta/NombreInterface;  
import { NombreServicio } from '../..//NombreCarpeta/ NombreServicio;
```

Junto con los demás imports necesarios.

Inicializar el servicio en el constructor y crear las mismas propiedades del interfaz.

```
constructor(  
    public NombreServicio: NombreServicio  
) { }
```

Y con ello ya se podrá utilizar los métodos creados en el servicio, utilizando la interfaz creada anteriormente y programar los métodos necesarios de manera local en cada componente que se requiera, ya sea guardar, editar, visualizar, buscar y eliminar. De la siguiente manera podremos acceder a un método del servicio:

```
this.NombreServicio.NombreDelMetodo
```

#### **D. Llamar un método desde el HTML del componente:**

Para llamar el método en el correspondiente componente donde se requiera, este será en el archivo HTML, y será de la siguiente manera:

```
(click)="NombreDelMetodo"  
(ngSubmit)="NombreDelMetodo"
```

Un pequeño ejemplo:

```
<a class="nav-link" href="#" (click)=" NombreDelMetodo ()">Salir</a>  
<form (ngSubmit)="GuardarDatos()">
```

Con esto al seleccionar esta etiqueta se llamara el método del componente, el cual está utilizando un servicio.

### **3.1.11 PRUEBAS DE ACEPTACIÓN**

Un punto importante en la culminación de un proyecto de desarrollo de software, es verificar que el producto de software se encuentre funcionando de manera correcta, antes de realizar la entrega al cliente de la aplicación web que solicito, se debe revisar minuciosamente cada uno de los detalles contemplados tanto en los requisitos como en el StoryBoard, tarea fundamental por parte del programador para asegurar la calidad del desarrollo y así obtener la aprobación por parte del cliente.

Es por ello que luego de terminada la programación necesaria del aplicativo web SPA, se hace necesario la realización de pruebas a cada uno de los componentes creados con anterioridad, ya sea si se ha creado un componente para el menú de navegación, un componente de login, de registrar, actualizar entre otros más., es importante asegurarse que cumplan a cabalidad con los requisitos y lo pactado en etapas previas al desarrollo. En caso de cualquier anomalía presentada en estas pruebas se requiere de dirigirse al paso donde se realizó la creación y codificación de componentes y comenzar a verificar su funcionamiento, seguido por las rutas, configuración de firebase, service, guards o interface, hasta encontrar el error que se presenta y corregirlo. Una vez terminada la corrección de errores en tal caso que se presente, se renueva la realización de una prueba al respectivo componente donde se encontró el error para darle el visto bueno y dar por finalizada la realización de las pruebas de manera exitosa.

Para la realización de pruebas en angular se puede realizar mediante el marco de pruebas de Jazmine, el cual se puede instalar de manera sencilla en la consola mediante el comando:

```
ng test
```

Este comando crea la aplicación en modo vigilancia, y se ejecuta el corredor de pruebas karma.

Mostrando en consola el registro con las pruebas que se realizaron, también se abre en el navegador el resultado de la prueba Jasmine HTML report.



Figura 47 Pruebas en Angular. Tomado de: <<https://angular.io/guide/testing>>

Para la configuración de las pruebas se puede realizar editando el siguiente archivo:

```
karma.conf.js
```

Para los test se manejarían en el siguiente archivo:

```
test.ts
```

En el apartado de documentación de angular se puede obtener más información acerca de pruebas específicas a cada uno de los componentes, dependencias, servicios, al DOM, entre otras. También existen pruebas predeterminadas por el cli de Angular que se pueden ver detalladamente en su página oficial:

<https://angular.io/guide/testing>

Una vez pasada la etapa de pruebas ya sea de manera manual o mediante las herramientas de Angular para la realización de testing, se procede a la siguiente y última etapa del procedimiento.

### **3.1.12 ENTREGA DE PRODUCTO**

Por último y ya para finalizar el procedimiento de la utilización del framework Angular en la creación de aplicaciones web SPA, se procede a realizar la entrega al cliente, mostrando lo pactado en los requisitos, junto con el StoryBoard el cual se aprobó con anterioridad al desarrollo. En esta entrega se revisa lo que estaba acordado en el proyecto junto con el producto, el desarrollo de la aplicación web, de manera que el cliente vea que se cumplió con lo pactado en el proyecto y no se presenten inconformidades en lo que se solicitó.

## **3.2 VALIDACIÓN DEL PROCEDIMIENTO CON DESARROLLO DE UN PROTOTIPO**

En la validación del procedimiento se desarrollará un prototipo de aplicativo web sencillo, con mínimo 3 funcionalidades, en el cual se abordaran los pasos propuestos en el procedimiento AngularDASPA, que se describirán a continuación.

### **A. INSTALACIÓN Y CONFIGURACIÓN ANGULAR**

Este paso no fue necesario ya que el autor ya contaba con la instalación de Angular en su equipo.

## B. CONCEPTUALIZACIÓN DEL PROYECTO

El prototipo a desarrollar será para apoyar el control de prácticas de los tutores que desarrollan el trabajo social del proyecto “Desarrollo del pensamiento computacional en estudiantes de colegios” dirigido por el docente Edgar Albornoz.

La primera etapa fue aplicar una encuesta donde se aterrizará la idea central del proyecto, adicionalmente se obtendrá información para conocer la necesidad que el aplicativo debe resolver, con ayuda de la información suministrada resolver el cuestionario de viabilidad de uso de Angular, esperando obtener un resultado favorable para continuar con el desarrollo del proyecto usando Angular.

Las preguntas que se plantearon en una primera entrevista fueron las siguientes:

1. ¿Cuántas personas tendrán contacto con el aplicativo, y que función desempeñan?

RTA: Conjunto de tutores (estudiantes Ing. sistemas) y el director (docente)

Tutores: fecha, duración, horario, actividades, guía o evidencias y tareas.

Director: Revisa las actividades registradas.

2. ¿Qué necesidad quiere que resuelva el desarrollo del aplicativo?

Se quiere resolver el control de las diferentes prácticas que se desarrollan con los estudiantes llevando el control una bitácora de desarrollo.

3. ¿Qué información considera importante manejar en el aplicativo?

Actividades, tareas, guías o evidencias, horarios, fecha y duración de la actividad.

4. ¿Desea que su aplicativo sea: Web, de Escritorio o Móvil?

Aplicativo web.

Seguido se realiza el cuestionario para evaluar la viabilidad de usar el Framework Angular en el desarrollo del proyecto.

CARACTERÍSTICA	SI	NO	EN ESCALA DE 1 AL 3 siendo 1 Poco importante, 2 Importante, 3 Muy importante QUE TAN IMPORTANTE ES EN SU PROYECTO LA CARACTERÍSTICA
HTML dinámico	+		1
Manejo de formularios y eventos.	+		3
Manejo de servicios e inyección de dependencias	+		2
Modelos y vistas	+		3
Protección de rutas.	+		2
Data Binding	+		1
Directivas	+		1
Enrutamiento	+		3
Cliente HTTP	+		3
Patrón MVC	+		2
Óptimo rendimiento en aplicaciones de gran tamaño		-	1
CL Debugger		-	1
Aplicativo de escritorio o Android		-	1
<b>TOTAL DE PUNTOS</b>			<b>24</b>

*Tabla 6 Evaluación viabilidad proyecto*

Obteniendo un total de 24 puntos, donde el mínimo requerido para usar Angular era de 15 puntos.

### **C. IDENTIFICACIÓN DE REQUERIMIENTOS**

Luego de obtener un resultado favorable en el cuestionario de uso de viabilidad de Angular, se programa otra reunión para la obtención de requerimientos funcionales y no funcionales, en esta etapa se procede a realizar otra entrevista más a fondo con el fin de recolectar información, que resuelva las dudas teniendo claros los

requerimientos y continuar con el siguiente paso del procedimiento. En esta etapa se plantearon las siguientes preguntas:

1. La bitácora de desarrollo de clases en el trabajo social debe controlar:

Actividades: SI Tareas: SI Guía o evidencia: SI

Duración: SI Fecha: SI Notas: NO

Otra: Participantes y Grupo.

2. Que información considera pertinente para controlar en:

Tutores: Datos básicos

Director: Datos básicos.

Actividades: Descripción, Tiempo duración, fecha, evidencia, participantes, tareas, grupo.

3. Todos pueden visualizar las actividades de la bitácora.

Si: X No: \_\_\_ Quienes: \_\_\_

4. Quien o quienes pueden crear, editar, actualizar y eliminar.

Actividades: El que la crea.

Tareas: El que la crea.

Guías o evidencias: El que la crea

Duración: El que la crea

Fecha: El que la crea

Mediante las cuales se infieren los requerimientos funcionales y no funcionales del aplicativo web a desarrollar.

## D. CREACIÓN DE STORYBOARD

En esta etapa se realizará la creación del StoryBoard del cliente y del desarrollador, que se plantearon de la siguiente manera.

### Storyboard cliente

Con base en las entrevistas y requerimientos capturados el desarrollador ya tiene una visión clara acerca de lo que quiere el cliente y se procede a diseñar en papel las diferentes interfaces graficas de usuario para presentarlas al cliente. Para el proyecto se presenta el ejemplo de la página principal.



Figura 48 StoryBoard cliente - Bitacora trabajo social

En reunión con el cliente este acepto la propuesta de GUI y se superó la etapa pasando a la siguiente parte del StoryBoard de desarrollador.

## Storyboard desarrollador

Una vez listo el StoryBoard de cliente completo y aceptado, se procede a identificar las piezas fundamentales del StoryBoard de desarrollador. Como se puede visualizar en las figuras 49 a 53:

### Dependencias:

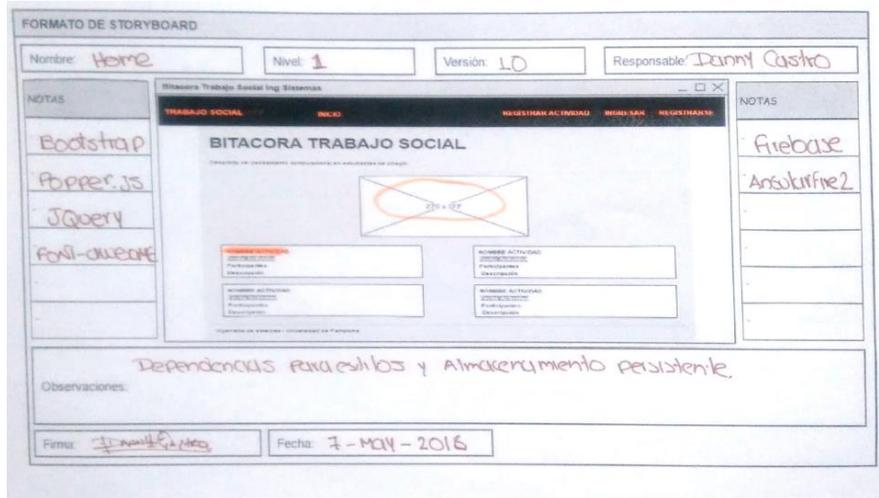


Figura 49 StoryBoard dependencias

### Componentes:

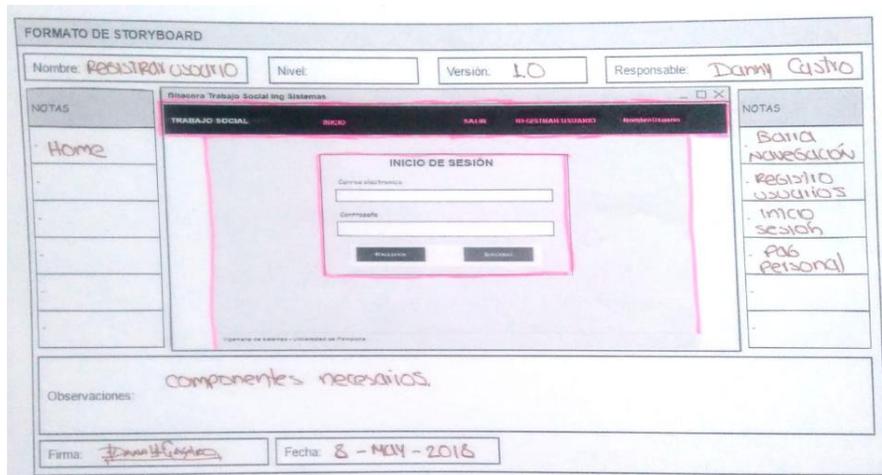


Figura 50 StoryBoard Componentes 1

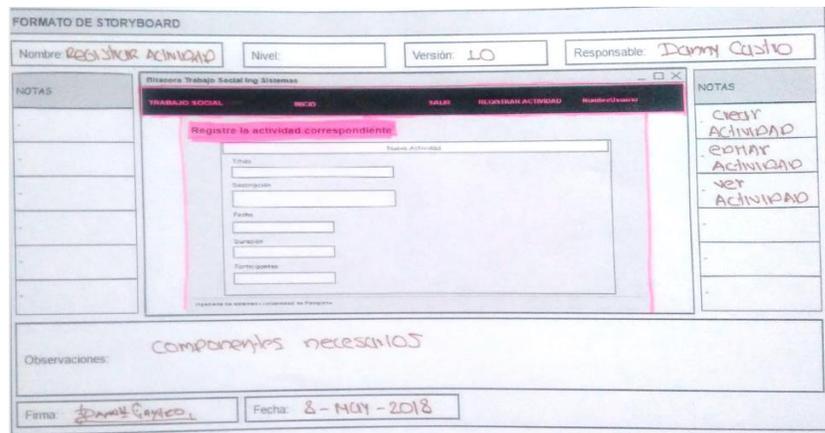


Figura 51 StoryBoard Componentes 2

**Datos:**

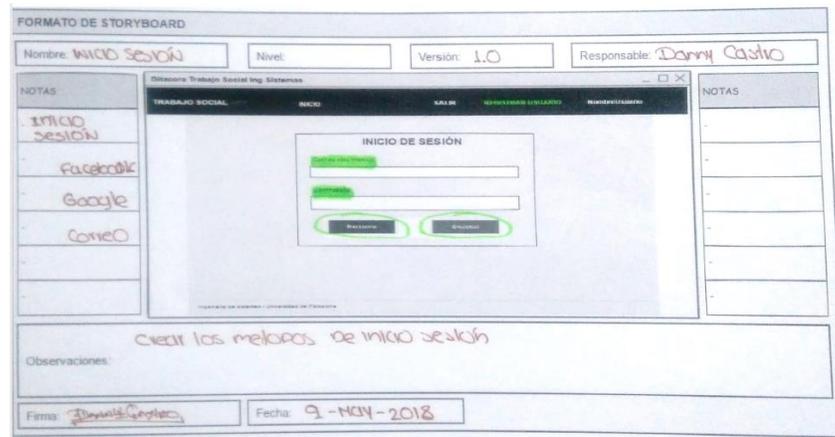


Figura 52 StoryBoard Datos 1

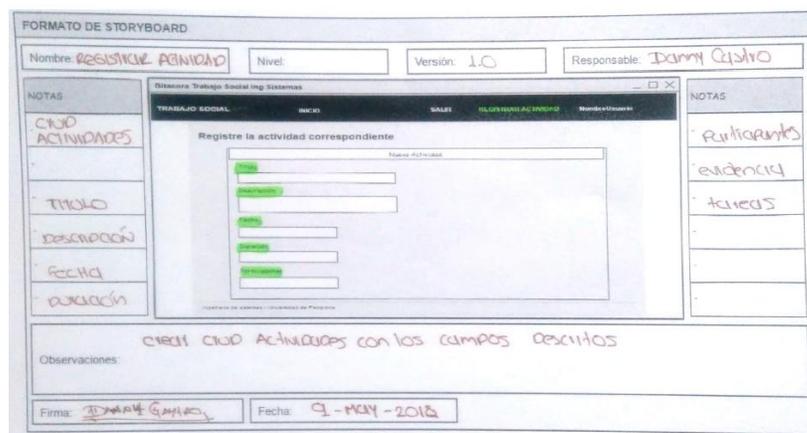


Figura 53 StoryBoard Datos 2

## **E. INSTALACIÓN DE DEPENDENCIAS**

En esta etapa se requirió la ayuda del StoryBoard de desarrollador, en la sección dependencias, de las cuales se identificaron necesarias para la instalación las siguientes:

- Bootstrap.
- Popper.js
- JQuery.
- Font-awesome.
- Firebase.
- AngularFire2.

## **F. CREACIÓN DE COMPONENTES**

En la etapa de componentes, junto con la ayuda del StoryBoard de desarrollador, en la sección de componentes se identificaron y crearon los siguientes componentes.

Agregar actividad, editar actividad, ver actividad, inicio sesión, navegación, pagina no encontrada, pagina inicio, pagina registro, página personal.

## **G. CREACIÓN DE RUTAS**

En la etapa de creación de rutas, se procede a la creación de las vistas necesarias en el aplicativo siendo las siguientes:

- Home.
- Add-actividad.
- Editar-actividad.
- Ver-actividad.
- Inicio-sesión.

- Registrar-usuario.
- Personal.

## H. CONFIGURACIÓN FIREBASE

En esta etapa se realizó la configuración de Firebase para el inicio de sesión, junto con el manejo de base de datos, ya que en los requerimientos se identificó que se requería el almacenamiento de datos de manera persistente y se necesitaba un inicio de sesión, junto con protección de rutas y de información.

En este caso se configura para la gestión de autorizaciones con el servicio de autenticación de usuario de Firebase. Junto con el servicio de base de datos NOSQL, alojada en la nube y también se hace necesario activar el servicio de Storage de Firebase para la gestión de archivos.

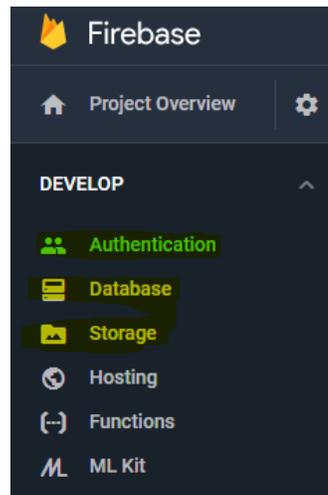


Figura 54 Servicios utilizados de Firebase

## I. GESTIÓN DE AUTORIZACIONES

La gestión de autorización para este proyecto se realizó un inicio de sesión mediante 3 métodos que fueron los siguientes:

Proveedores de inicio de sesión	
Proveedor	Estado
 Correo electrónico/contraseña	Habilitada
 Teléfono	Inhabilitado
 Google	Habilitada
 Play Juegos	Inhabilitado
 Facebook	Habilitada
 Twitter	Inhabilitado
 GitHub	Inhabilitado
 Anónimo	Inhabilitado

Figura 55 Métodos inicio de sesión

Se habilita el inicio de sesión con estos tres métodos al ser los más utilizados, ya que la mayoría de los futuros usuarios del aplicativo cuentan con Facebook o un correo electrónico, y con ello se hace más fácil el contacto con ellos.

## J. REALIZACIÓN CRUD

En la realización de CRUD para el proyecto de Bitácora de Trabajo social, fue necesario implementar para los siguientes ítems:

- Registro de usuarios por correo electrónico, Gmail o Facebook.
- Administración de actividades.

Allí se realiza la creación, actualización y visualización de actividades, las cual se guarda como una colección actividades las cuales contienen un documento con los campos necesarios. La figura 56 describe la estructura:

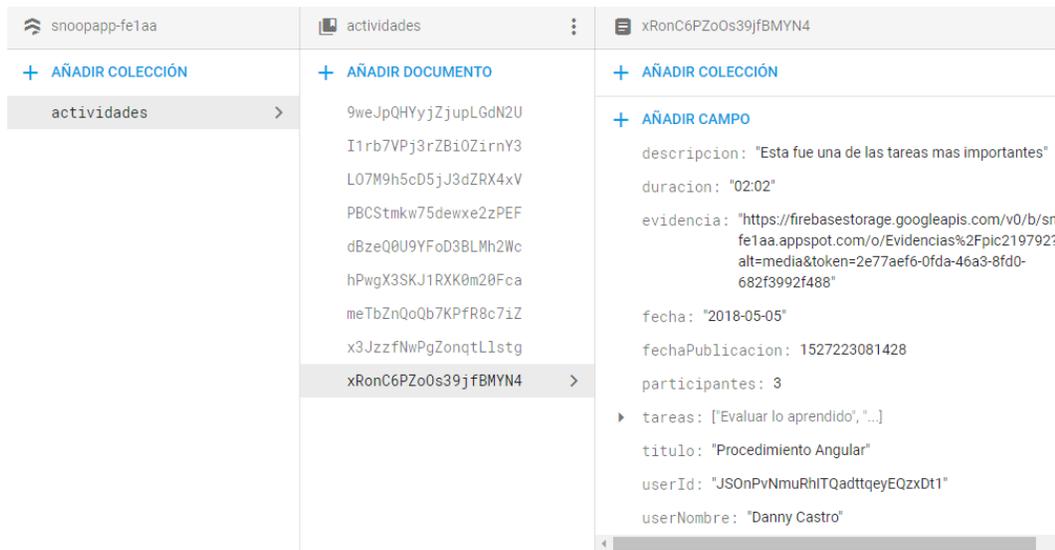


Figura 56 Estructura base de datos Firebase

## K. PRUEBAS DE ACEPTACIÓN

Las pruebas de aceptación en este proyecto dedicado a los estudiantes de trabajo social de ingeniería de sistemas, fueron realizadas de manera manual, verificando el correcto funcionamiento del inicio de sesión, ingresar, editar y eliminar actividades. De tal manera, que cumpliera con las expectativas del cliente que en este caso particular es el directo de trabajo social, el profesor Edgar Albornoz. No fueron realizadas de manera exhaustiva ya que en las acotaciones del proyecto se indicó que el desarrollo sería un prototipo y el uso de este se limitará a una mínima cantidad de estudiantes solo con fines educativos.

## L. ENTREGA DE PRODUCTO

Luego de verificar y corregir los errores encontrados se procede a la entrega del aplicativo web al director de trabajo social, Edgar Albornoz de manera digital y un despliegue a producción con ayuda de la herramienta Firebase.

Pamplona 31 de mayo 2018

Señor

Edgar Albornoz  
Coordinador trabajo social "Desarrollo del pensamiento computacional aplicado en estudiantes de colegio"  
L. C.

Cordial saludo,

En cumplimiento del objetivo de trabajo de grado se realiza la entrega del aplicativo para el control de actividades que desarrollan los estudiantes de trabajo social del programa de ingeniería de sistemas, en el proyecto "Desarrollo del pensamiento computacional aplicado en estudiantes de colegio".

Cumpliendo con los requerimientos identificados al inicio del trabajo de grado, para hacer uso del aplicativo se puede acceder mediante la dirección: <https://trabajosocialup-f15aa.firebaseio.com/> y el código del mismo se encuentra alojado en GitHub en la dirección: <https://github.com/Bismarck21/SnoopApp>

Anexo: Manual de uso y configuración del aplicativo

Att:   
Danny Bismarck Castro

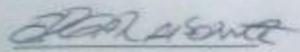
  
Prof. Edgar Albornoz

Figura 57 Carta aceptación bitácora trabajo social

### 3.3 VALIDACIÓN DEL PROCEDIMIENTO POR EXPERTOS

Este método de validación va a permitir que dos expertos validen el procedimiento propuesto denominado AngularDASPA (Angular Desarrollo Ágil SPA). Por ello se les envió a los expertos una copia del procedimiento para poder leerlo y analizarlo detalladamente, y posteriormente diligenciar con su criterio experto las diferentes preguntas del cuestionario, verificando la calidad en cada una de las etapas allí plasmadas, con el fin de obtener una medida de aceptación del procedimiento.

### 3.3.1 EXPERTOS

NOMBRE	TITULO	EXPERIENCIA
Jorge Erickson Trujillo Flórez	Ing. de Sistemas	Desarrollador en múltiples lenguajes de programación con la empresa Dominio Estudio, (Bogotá). Actualmente desarrollador independiente en múltiples campos.
Doyler Emmanuel Rubio Villamizar	Ing. de Sistemas	Desarrollo tesis de grado en Angular.js

### 3.3.2 CUESTIONARIO

FECHA					
NOMBRE DEL EXPERTO					
TITULO DEL EXPERTO					
CORREO DEL EXPERTO					
PARAMETRO	Inadecuado	Poco Adecuado	Adecuado	Bastante Adecuado	Muy Adecuado
VALOR	1	2	3	4	5

CRITERIOS	PARAMETROS				
ANALISIS	1	2	3	4	5
El procedimiento contempla la configuración previa para el uso del framework Angular.					
El procedimiento presenta pautas para obtener información importante e inferir de manera adecuada los requerimientos.					
Dentro del procedimiento se conceptualiza al lector, si es viable usar Angular en un determinado proyecto.					
DISEÑO					
El procedimiento describe la utilización de un Storyboard para la aceptación por parte del cliente.					
Se emplea el uso de un Storyboard para el desarrollador, mediante el cual se infieran aspectos importantes para el correcto proceso de desarrollo.					
El procedimiento cuenta con un diseño previo de los requerimientos para el desarrollo, presentado de manera clara y entendible.					
DESARROLLO					
En el procedimiento se describe la instalación de dependencias para un proyecto en general de manera clara.					
Los componentes necesarios para un proyecto en Angular se logran identificar y desarrollar de manera clara con las indicaciones sugeridas.					
La creación de rutas necesarias y envío de parámetros se entiende y logra trabajar de manera clara.					
En el procedimiento se presenta de manera clara el uso de Firebase para la gestión de autorizaciones y base de datos.					
Es capaz de crear un inicio de sesión y registro de usuarios con las indicaciones sugeridas de gestión de autorizaciones con Angular y Firebase.					
El procedimiento describe la protección de rutas, para el manejo de la seguridad del aplicativo.					

Dentro del procedimiento se describe la realización de operaciones CRUD con el manejo de Angular y Firebase.					
<b>PRUEBAS Y ENTREGA</b>					
El procedimiento tiene en cuenta el manejo de pruebas antes de la entrega del producto al usuario final.					
El procedimiento enfatiza en regresar a una etapa previa en caso de alguna anomalía presentada en las pruebas.					
La entrega del producto final es tomada en cuenta dentro del procedimiento.					

## RESULTADO DE LA EVALUACIÓN

Al finalizar el cuestionario el experto deberá sumar los puntos obtenidos en cada una de las preguntas y clasificar este resultado en la siguiente tabla, determinando el nivel de aceptación del procedimiento.

CRITERIO	RANGO	PUNTAJE OBTENIDO
EXCELENTE	80 – 65	
MUY BUENO	64 – 49	
BUENO	48 – 33	
REGULAR	32 – 17	
MALO	16 – 0	

## OBSERVACIONES Y RECOMENDACIONES

---



---



---



---



---



---



---



---

**FIRMA DEL EXPERTO**

### 3.3.3 RESULTADOS DE VALIDACIÓN

Los resultados de la validación del procedimiento AngularDASPA, se presenta con el nombre del experto, el puntaje, el criterio y sus respectivas observaciones. Resultados en los cuales se evidenciaron recomendaciones importantes que se llevaron a cabo en el procedimiento para aumentar la calidad del mismo.

Para ver detalladamente los resultados de cada uno de los expertos: Ver anexo 6.2.

Nº	NOMBRE DEL EXPERTO	PUNTAJE	CRITERIO	OBSERVACIONES
1	DOYLER EMMANUEL RUBIO VILLAMIZAR	62	MUY BUENO	<p>Dar a conocer al lector en que consiste el manejo de la tecnología Firebase como soporte al backend dentro del procedimiento. Utilizar otro tipo de estrategia de documentación para reemplazar el Story board propuesto con el fin de ser mucho más ágiles a la hora de mantener, recuperar y depurar diseños.</p> <p>Presentar bloques de instrucciones de ejemplo siempre formateados y escritos dentro de un editor de código para facilitar su legibilidad y entendimiento. En algunos apartados se observan pantallazos del IDE que son mucho más amigables, que aquellos escritos directamente en el procesador de texto.</p> <p>Es de gran importancia dar a conocer el soporte teórico sobre el cual se establece la métrica para la calificación del framework, considero que puede ser comparado contra otros framework existentes y que pueden aportar valor al desarrollo del procedimiento. Explicando los componentes considero que se genera una ambigüedad al determinar lo que realmente considera un elemento funcional en la aplicación. Un componente no se debe limitar simplemente a desarrollar elementos del frontend como la cabecera o un menú de navegación, puesto que a estos se debe vincular el manejo de servicios, routing, lógica y paso de parámetros al API de manejo de datos del firebase.</p> <p>Es importante exponer de igual manera que ventajas supone el uso de Typescript respecto al javascript pues el lector debe entender que, aunque son lenguajes similares de programación se comportan con</p>

				<p>ciertas diferencias que pueden llegar a influir en el desarrollo del procedimiento.</p> <p>El termino SPA (Single Page Application) es utilizado en todo el documento, pero no se expone ni se explica su fundamento teorico.</p> <p>Finalmente, en el apartado de validaciones y aceptación del producto se debe orientar sobre el tipo de pruebas que se puedan llevar a cabo sobre el código, y además los procedimientos que formalmente se deben tener en cuenta como documentos y actas que garanticen un producto de calidad.</p>
2	JORGE ERICKSON TRUJILLO FLOREZ	47	BUENO	<p>Se evidencia el proceso de instalación solo para plataformas Windows, no se contempla plataformas MAC y tampoco GNU/Linux, tampoco existe un procedimiento de instalación por consola.</p> <p>El StoryBoard del desarrollador debe incluir un poco más específico las explicaciones sobre eventos, comportamientos y acciones.</p> <p>Al procedimiento de desarrollo le hace falta hablar sobre, el manejo de building, compresión y optimización que muchos Frameworks de este tipo tienen en este tipo de aplicaciones. Además se debe realizar un acercamiento a las directivas más frecuentemente utilizadas, durante el desarrollo de una aplicación para que sirva como referencia a personas que no tengan conocimiento de las mismas.</p> <p>Incluir un aparte que hable concretamente de temas de seguridad y como asegurar la aplicación de diferentes XSS, CSRF, XSSI.</p> <p>La etapa de pruebas está muy básica, no se evidencia formato de recolección de información de bugs y tampoco seguimiento. Debe incluirse una estrategia para la recolección de ajustes por parte del cliente. También sería bueno mirar el tema de cómo se escriben pruebas para angular y como se puede aplicar al proyecto.</p> <p>En cuanto a presentación se debe mejorar la diagramación de la guía resaltando el código y las notas importantes, con el fin que si se desea consultar posteriormente sea fácil encontrar la referencia.</p>

Por tanto, las observaciones de los 2 expertos fueron de vital importancia para el mejoramiento del procedimiento AngularDASPA.

Citando las observaciones del experto N° 1, donde presenta que, no se explica el manejo de la tecnología firebase, el termino SPA, lenguaje Typescript, y también el soporte teórico que fundamenta el uso de angular, estos son apartados que se definieron en el capítulo 2, marco teórico y estado del arte. La observación de utilizar otro tipo de estrategia de documentación para reemplazar el StoryBoard propuesto, se mejoró la redacción para dar a entender la importancia de esta propuesta de StoryBoard para desarrollador. Los bloques de instrucciones o comandos, se incluyeron en una caja de texto con fondo gris para mejorar su legibilidad como se propuso. En donde se comenta que un componente no se debe limitar a desarrollar elementos del frontend, puesto que se deben vincular a servicios, routing, lógica, envío de parámetros, y manejo de datos en firebase, en esta sección se pretendía solo identificar los elementos fundamentales para abordar en las etapas siguientes. Por ultimo en la sección de pruebas se abordó el manejo de pruebas con Angular y se hizo la referencia correspondiente a la documentación de pruebas ya que son complejas de escribir y distintas para cada uno de los proyectos.

Citando las observaciones del experto N° 2, para el proceso de instalación que no se realizó para las plataformas MAC y tampoco GNU/Linux, fue porque la instalación es una etapa previa al procedimiento, y este no se considera un paso a seguir estrictamente dentro del procedimiento sino un paso previo de manera opcional. En cuanto a las características de viabilidad del framework estas se encuentran presentes en el capítulo 2, marco teórico y estado del arte. En cuanto al StoryBoard del desarrollador debe ser más específico, se mejoró la redacción para entender mejor esta etapa. En cuanto al manejo de templates, procesadores CSS, building, pruebas de seguridad (XSS, CSRF, XSSI), no se ve necesario profundizar tanto ya que la idea central del procedimiento es cortar la curva de aprendizaje y el manejo de estas herramientas se encuentra en un nivel más avanzado y se emplearía más tiempo investigando algo que de pronto no llegue a utilizar en el desarrollo del

aplicativo, pero si se realizó una tabla con las directivas más utilizadas en Angular como anexo, una tabla con los elementos más usados de Data binding, junto con la referencia a las pipes que posiblemente se puedan necesitar en algún tipo de aplicativo . La etapa de pruebas se complementó con una introducción al manejo de ellas con Angular y se referencio a la documentación para una investigación más a fondo si el lector así lo quiere. En cuanto a la presentación del código se mejoró, resaltándolo en una caja de texto con fondo gris para una mejor legibilidad.

### **3.3.4 VALIDEZ DEL PROCEDIMIENTO**

El procedimiento AngularDASPA, con la valoración de expertos en el desarrollo con Angular, al final este trabajo se culminó arrojando un resultado promedio de **MUY BUENO**, con los criterios allí definidos en cada una de las etapas, obteniendo una media estadística de **54.5** entre los valores de los dos expertos.

En conclusión el procedimiento AngularDASPA es válido, con criterio muy bueno, ya que disminuye de manera significativa la curva de aprendizaje para nuevos desarrolladores enfocándose en las tareas primordiales para la construcción de un aplicativo web del tipo SPA con el framework Angular, sin llegar a profundizar en aspectos avanzados como temas de seguridad, manejadores de CSS, desarrollo de pruebas para los componentes. Por ello el procedimiento queda abierto a futuras modificaciones a aquellos que deseen investigar más acerca de este novedoso Framework.

## 4. CONCLUSIONES, RECOMENDACIONES Y TRABAJOS FUTUROS

### 4.1 CONCLUSIONES

Se determinó en la revisión bibliográfica del estado del arte, que en un estudio, procedimiento o metodología de un aplicativo para un determinado fin, las etapas en el proceso de desarrollo varían una de la otra, de acuerdo al tipo de metodología utilizada, tanto de requerimientos y tecnología a utilizada, por ello es conveniente tener en cuenta estos aspectos y las características del framework Angular al momento de definir las etapas del procedimiento para desarrollar aplicaciones web SPA con el framework Angular.

Se establecieron una serie de etapas en el procedimiento AngularDASPA, gracias al análisis previo del estado del arte, funcionalidades y alcances del framework Angular. Etapas en las cuales se plasman los conceptos y pasos más importantes que debe abordar un desarrollador para disminuir la curva de aprendizaje, enfocándose solo en las etapas necesarias para la culminación de un proyecto en el menor tiempo posible, con los conceptos necesarios para el correcto aprendizaje del framework Angular desarrollando un aplicativo web SPA.

Se verifica la validez del procedimiento AngularDASPA, mediante el desarrollo de un prototipo de aplicativo web orientado al trabajo social por parte de los estudiantes de ingeniería de sistemas, desarrollo del pensamiento computacional en estudiantes de colegio. Aplicativo con funcionalidades CRUD e inicio de sesión, en el cual se cumplieron satisfactoriamente cada una de las etapas propuestas en el procedimiento.

Se verifica de igual manera la validez del procedimiento AngularDASPA, mediante el juicio de expertos en el desarrollo con el framework Angular, obteniendo observaciones y/o recomendaciones, que permitieron identificar la calidad del procedimiento propuesto y así mismo se logró mejorar la calidad de este.

## **4.2 RECOMENDACIONES**

Se recomienda a los interesados en profundizar en el framework Angular visitar la página oficial (<https://angular.io/docs>), donde encontrarán toda la documentación necesaria, ejemplos, blog de noticias para estar pendientes de las actualizaciones, versiones y demás.

Angular permite desarrollar en múltiples estilos y existen diversas librerías que se integran para lograr resultados óptimos, se puede llegar a profundizar en temas como la protección del sitio contra ataques, desarrollo de pruebas de cada uno de los componentes de Angular, para lograr aplicaciones más robustas y complejas.

## **4.3 TRABAJOS FUTUROS**

Se recomienda como trabajos futuros integrar al procedimiento AngularDASPA, el desarrollo de aplicativos móvil, ya que es frecuentemente utilizado gracias a su rendimiento e infinidad de herramientas con las que cuenta y en este trabajo se encuentra enfocado al desarrollo de aplicaciones web SPA.

## 5. BIBLIOGRAFIA

Campus MVP. Los 5 mejores Framework de JavaScript en 2017. [En Línea] <<https://www.campusmvp.es/recursos/post/los-5-mejores-frameworks-de-javascript-en-2017.aspx>> [Citado el 03 Marzo de 2018]

Open Webinars. Los 6 mejores Frameworks JavaScript. [En línea] <<https://openwebinars.net/blog/los-6-mejores-frameworks-javascript/>> [Citado el 03 Abril del 2018]

Made with Angular. Proyecto hechos con Angular. [En Línea] <<https://www.madewithangular.com/categories/angular>>

Red de TechTarget. Framework. [En Línea] <<http://searchdatacenter.techtarget.com/es/definicion/Framework>> [Citado el 03 de Abril del 2018].

Desarrollo web. Que es una SPA. [En línea]. <<https://desarrolloweb.com/articulos/que-es-una-spa.html>> [citado el 3 de marzo de 2018].

Wikipedia. Angular (framework), Características. [En línea] <[https://es.wikipedia.org/wiki/Angular\\_\(framework\)](https://es.wikipedia.org/wiki/Angular_(framework))> [citado el 3 de marzo de 2018].

Desarrollo web. Introducción a TypeScript. [En línea]. <<https://desarrolloweb.com/articulos/introduccion-a-typescript.html>> [citado el 3 de marzo de 2018].

IDA. ¿Qué esperamos de un desarrollador front-end? [En línea] <<https://www.ida.cl/blog/desarrollo/desarrollador-front-end/>> [citado el 3 de marzo de 2018].

Campus MVP. Desarrollador web: Front-end, back-end y full stack. ¿Quién es quién? [En línea] <<https://www.campusmvp.es/recursos/post/Desarrollador-web->

Front-end-back-end-y-full-stack-Quien-es-quien.aspx> [citado el 3 de marzo de 2018].

Open webinars. ¿Qué es firebase de Google? [En Línea]. <  
<https://openwebinars.net/blog/que-es-firebase-de-google/>>

Google. Técnicas para identificar requisitos funcionales y no funcionales. [En Línea]  
<<https://sites.google.com/site/metodologiareq/capitulo-ii/tecnicas-para-identificar-requisitos-funcionales-y-no-funcionales>>

Angular. Documentación de Angular. [En Línea] < <https://angular.io/docs>>

Pedro Jiménez Castela. Angular 4 desde Cero. La guía práctica y completa para convertirte en desarrollador Angular Full stack. Madrid: Sapiens IT, Junio de 2017.

## 6. ANEXOS

### 6.1 DIRECTIVAS EN ANGULAR

Las directivas en Angular se usan para cambiar el comportamiento de la aplicación en determinado momento, directivas las cuales se clasifican en 3 tipos: componentes, atributos y estructurales. Ver tabla 7.

NOMBRE	TIPO	DESCRIPCION	SINTAXIS
Directiva ngIf	Estructural	Permite visualizar o no un elemento.	*ngIf="expresión/propiedad/metodo"
Directiva ngStyle	Atributo	Establece de manera dinámica el estilo de un elemento.	[ngStyle]="{ expresión/propiedad/método }"
Directiva ngClass	Atributo	Establece de manera dinámica las clases de un elemento.	[ngClass]="{ expresión/propiedad/método }"
Directiva ngFor	Estructural	Realiza iteraciones.	*ngFor="let objeto/propiedad of objetos/propiedades"
Directiva NgSwitch	Estructural	Presenta opciones en cada caso.	*ngSwitchCase="expresión/propiedad/me todo"

Tabla 7 Directivas en Angular

## 6.2 JUICIO DE EXPERTOS

- JORGE ERICKSON TRUJILLO FLOREZ

### CUESTIONARIO DE EVALUACION DEL PROCEDIMIENTO ANGULARDASPA

FECHA		22 de Mayo de 2018						
NOMBRE DEL EXPERTO		Jorge Erickson Trujillo Florez						
TITULO DEL EXPERTO		Ingeniero de Sistemas						
CORREO DEL EXPERTO		joertru@gmail.com						
PARAMETRO	Inadecuado	Poco Adecuado	Adecuado	Bastante Adecuado	Muy Adecuado			
VALOR	1	2	3	4	5			
CRITERIOS				PARAMETROS				
ANALISIS				1	2	3	4	5
El procedimiento contempla la configuración previa para el uso del framework Angular.						X		
El procedimiento presenta pautas para obtener información importante e inferir de manera adecuada los requerimientos.							X	
Dentro del procedimiento se conceptualiza al lector, si es viable usar Angular en un determinado proyecto.						X		
DISEÑO								
El procedimiento describe la utilización de un Storyboard para la aceptación por parte del cliente.							X	
Se emplea el uso de un Storyboard para el desarrollador, mediante el cual se infieran aspectos importantes para el correcto proceso de desarrollo.					X			
El procedimiento cuenta con un diseño previo de los requerimientos para el desarrollo, presentado de manera clara y entendible.						X		
DESARROLLO								
En el procedimiento se describe la instalación de dependencias para un proyecto en general de manera clara.						X		
Los componentes necesarios para un proyecto en Angular se logran identificar y desarrollar de manera clara con las indicaciones sugeridas.						X		
La creación de rutas necesarias y envío de parámetros se entiende y logra trabajar de manera clara.						X		
En el procedimiento se presenta de manera clara el uso de Firebase para la gestión de autorizaciones y base de datos.							X	
Es capaz de crear un inicio de sesión y registro de usuarios con las indicaciones sugeridas de gestión de autorizaciones con Angular y Firebase.						X		
El procedimiento describe la protección de rutas, para el manejo de la seguridad del aplicativo.						X		
Dentro del procedimiento se describe la realización de operaciones CRUD con el manejo de Angular y Firebase.						X		
PRUEBAS Y ENTREGA								
El procedimiento tiene en cuenta el manejo de pruebas antes de la entrega del producto al usuario final.					X			
El procedimiento enfatiza en regresar a una etapa previa en caso de alguna anomalía presentada en las pruebas.					X			
La entrega del producto final es tomada en cuenta dentro del procedimiento.					X			

## RESULTADO DE LA EVALUACION

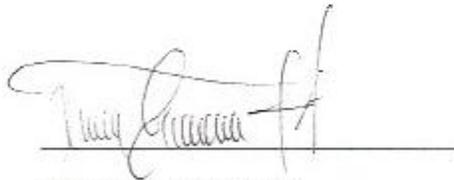
Al finalizar el cuestionario el experto deberá sumar los puntos obtenidos en cada una de las preguntas y clasificar este resultado en la siguiente tabla, determinando el nivel de aceptación del procedimiento.

CRITERIO	RANGO	PUNTAJE OBTENIDO
EXCELENTE	80 – 65	
MUY BUENO	64 – 49	
BUENO	48 – 33	47
REGULAR	32 – 17	
MALO	16 – 0	

## OBSERVACIONES Y RECOMENDACIONES

- Se evidencia el proceso de instalación solo para plataformas Windows, no se contempla plataformas MAC y tampoco GNU/Linux, tampoco existe un procedimiento de instalación por consola.
- En características de viabilidad del framework, se podrían agregar mas características como: Manejadores de paquetes, Curva de aprendizaje, Apoyo de la comunidad, Facilidad para conseguir desarrolladores, Sistema de actualización de módulos, seguridad.
- El StoryBoard del desarrollador debe incluir un poco más específico las explicaciones sobre eventos, comportamientos y acciones.
- Al procedimiento de desarrollo le hace falta hablar sobre, el manejo de templates, el manejo de procesadores de CSS, el manejo de building, compresión y optimización que muchos frameworks de este tipo tienen en este tipo de aplicaciones. Además se debe realizar un acercamiento a las directivas más frecuentemente utilizadas, durante el desarrollo de una aplicación para que sirva como referencia a personas que no tenga conocimiento de las mismas.
- Incluir un aparte que hable concretamente de temas de seguridad y como asegurar la aplicación de diferentes XSS, CSRF, XSSI.

- La etapa de pruebas está muy básica, no se evidencia formato de recolección de información de bugs y tampoco seguimiento. Debe incluirse una estrategia para la recolección de ajustes por parte del cliente. También sería bueno mirar el tema de como se escriben pruebas para Angular y como se puede aplicar al proyecto.
- En cuanto a presentación se debe mejorar la diagramación de la guía resaltando el código y las notas importantes, con el fin que si se desea consultar posteriormente sea fácil encontrar la referencia.



FIRMA DEL EXPERTO

*Jorge Erickson Trujillo Flórez*  
INGENIERO DE SISTEMAS  
joertru@gmail.com  
317 4910155

- DOYLER EMMANUEL RUBIO VILLAMIZAR

**CUESTIONARIO DE EVALUACION DEL PROCEDIMIENTO ANGULARDASPA**

FECHA		16/05/2018			
NOMBRE DEL EXPERTO		DOYLER EMMANUEL RUBIO VILLAMIZAR			
TITULO DEL EXPERTO		INGENIERO DE SISTEMAS			
CORREO DEL EXPERTO		derubio1@gmail.com			
PARAMETRO	Inadecuado	Poco Adecuado	Adecuado	Bastante Adecuado	Muy Adecuado
VALOR	1	2	3	4	5

CRITERIOS	PARAMETROS				
	1	2	3	4	5
<b>ANALISIS</b>					
El procedimiento contempla la configuración previa para el uso del framework Angular.					x
El procedimiento presenta pautas para obtener información importante e inferir de manera adecuada los requerimientos.				x	
Dentro del procedimiento se conceptualiza al lector, si es viable usar Angular en un determinado proyecto.			x		
<b>DISEÑO</b>					
El procedimiento describe la utilización de un Storyboard para la aceptación por parte del cliente.					x
Se emplea el uso de un Storyboard para el desarrollador, mediante el cual se infieran aspectos importantes para el correcto proceso de desarrollo.		x			
El procedimiento cuenta con un diseño previo de los requerimientos para el desarrollo, presentado de manera clara y entendible.		x			
<b>DESARROLLO</b>					
En el procedimiento se describe la instalación de dependencias para un proyecto en general de manera clara.					x
Los componentes necesarios para un proyecto en Angular se logran identificar y desarrollar de manera clara con las indicaciones sugeridas.				x	
La creación de rutas necesarias y envío de parámetros se entiende y logra trabajar de manera clara.				x	
En el procedimiento se presenta de manera clara el uso de Firebase para la gestión de autorizaciones y base de datos.				x	
Es capaz de crear un inicio de sesión y registro de usuarios con las indicaciones sugeridas de gestión de autorizaciones con Angular y Firebase.				x	
El procedimiento describe la protección de rutas, para el manejo de la seguridad del aplicativo.			x		
Dentro del procedimiento se describe la realización de operaciones CRUD con el manejo de Angular y Firebase.				x	
<b>PRUEBAS Y ENTREGA</b>					

El procedimiento tiene en cuenta el manejo de pruebas antes de la entrega del producto al usuario final.			x	
El procedimiento enfatiza en regresar a una etapa previa en caso de alguna anomalía presentada en las pruebas.				x
La entrega del producto final es tomada en cuenta dentro del procedimiento.				x

## RESULTADO DE LA EVALUACION

Al finalizar el cuestionario el experto deberá sumar los puntos obtenidos en cada una de las preguntas y clasificar este resultado en la siguiente tabla, determinando el nivel de aceptación del procedimiento.

CRITERIO	RANGO	PUNTAJE OBTENIDO
EXCELENTE	80 – 65	
MUY BUENO	64 – 49	62 PUNTOS
BUENO	48 – 33	
REGULAR	32 – 17	
MALO	16 – 0	

## OBSERVACIONES Y RECOMENDACIONES

### Se recomienda:

Dar a conocer al lector en que consiste el manejo de la tecnología firebase como soporte al backend dentro del procedimiento.

Utilizar otro tipo de estrategia de documentación para reemplazar el story board propuesto con el fin de ser mucho más ágiles a la hora de mantener, recuperar y depurar diseños.

Presentar los bloques de instrucciones de ejemplo siempre formateados y escritos dentro de un editor de código para facilitar su legibilidad y entendimiento. En algunos apartados se observan pantallazos del IDE que son mucho más "amigables" que aquellos escritos directamente en el procesador de texto.

**Observaciones:**

Es de gran importancia dar a conocer el soporte teórico sobre el cual se establece la métrica para la calificación del framework, considero que puede ser comparado contra otros frameworks existentes y que pueden aportar valor al desarrollo del procedimiento.

Explicando los componentes considero que se genera una ambigüedad al determinar lo que realmente se considera un elemento funcional en la aplicación. Un componente no se debe limitar simplemente a desarrollar elementos del frontend como la cabecera o un menú de navegación, puesto que a estos se debe vincular el manejo de servicios, routing, lógica y paso de parámetros al API de manejo de datos del firebase.

Es importante exponer de igual manera que ventajas supone el uso del Typescript respecto al Javascript pues el lector debe entender que, aunque son lenguajes similares de programación se comportan con ciertas diferencias que pueden llegar a influir en el desarrollo del procedimiento.

El término SPA (Single Page Application) es utilizado en todo el documento, pero no se expone ni explica su fundamento teórico.

Finalmente, en el apartado de validaciones y aceptación del producto se debe orientar sobre el tipo de pruebas que se pueden llevar a cabo sobre el código, y además los procedimientos que formalmente se deben tener en cuenta como documentos y actas que garanticen un producto de software de calidad.



**FIRMA DEL EXPERTO**