UNIVERSIDAD PRIVADA FRANZ TAMAYO

FACULTAD DE INGENIERIA

CARRERA DE INGENIERIA DE SISTEMAS

DEFENSA HITO 3 TAREA FINAL

ESTUDIANTE: Gabriela Valencia Lazarte

ASIGNATURA: PDM

CARRERA: Ing en Sistemas

Paralelo: PDM(1)

DOCENTE: Lic. William R. Barra Paredes

FECHA: 11/05/2020

GITHUB: LINK de la carpeta Hito3 de GIT

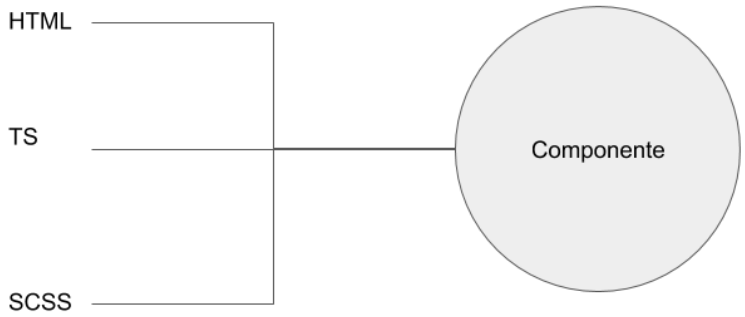
Cochabamba - Bolivia

PARTE TEORICA:

1. **Defina que es un componente en Angular y muestre un ejemplo.**

   Un componente en Angular es una combinación de un archivo html con un ts y algunas veces scss para crear un elemento con características propias tanto de comportamiento como de apariencia que se puede mostrar en un navegador.

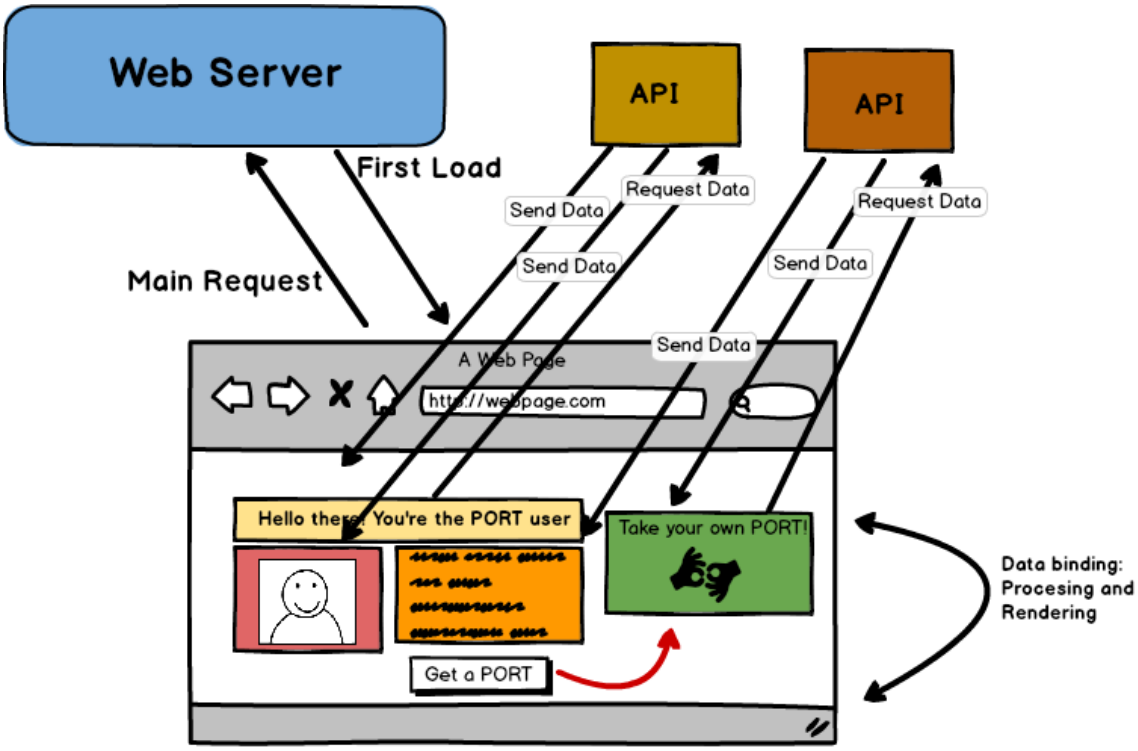
   Un componente es un elemento que compone a un todo y si lo llevamos a un punto de vista anatómico del ser humano, este puede ser un brazo, una pierna, etc.



   Una aplicación en Angular está compuesta por varios componentes:

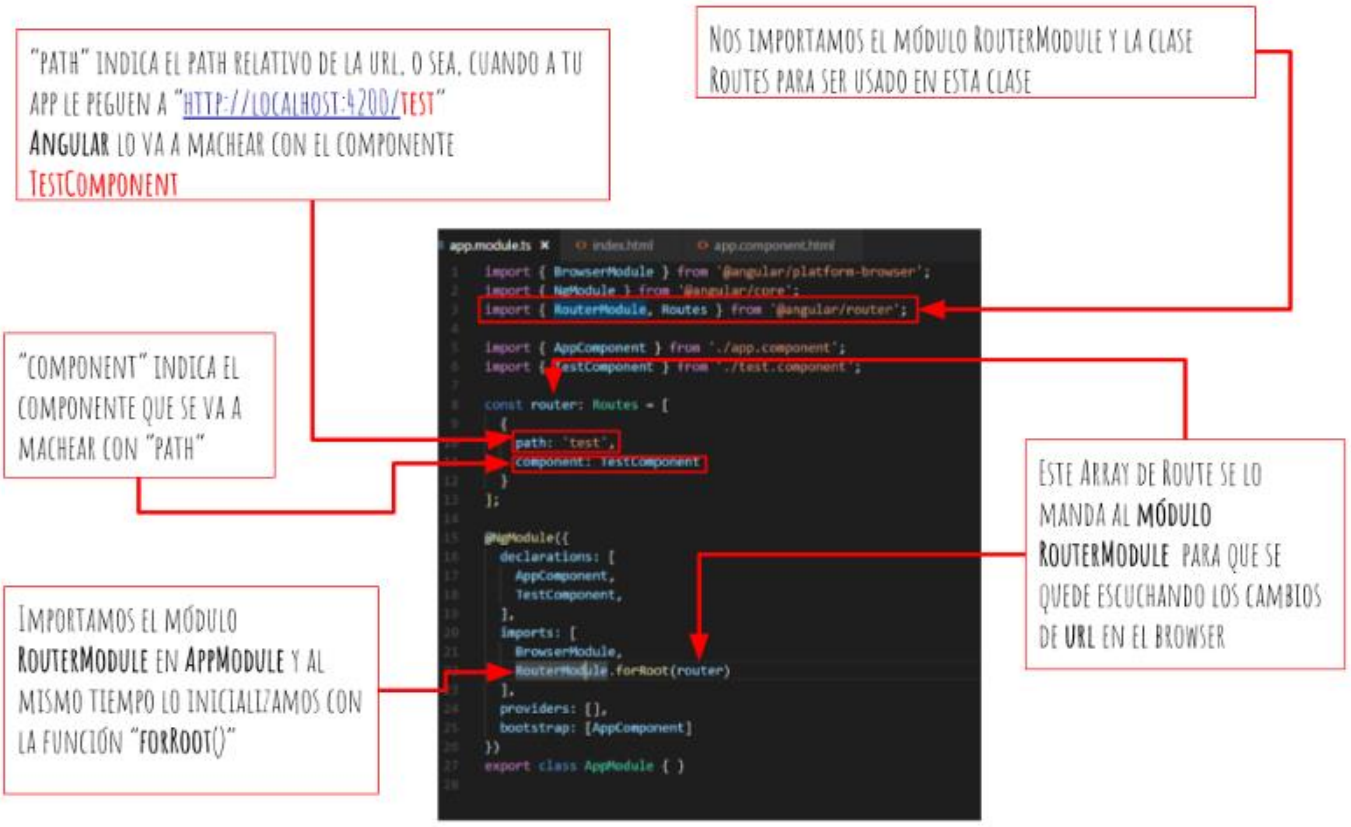
2. **Explique cómo se realiza la navegación entre screens en Angular**

   Lo que hacen ahora las aplicaciones Web Modernas es cargar una sola página y después, todas las otras páginas se refrescan usando Javascript; pero sólo index.html es una página completa, el resto de las páginas son sólo porciones de HTML que su código Javascript va cambiando dinámicamente. Esto hacía que la aplicación del browser funcionara mucho más rápida, ya que una de las cosas más lentas de una aplicación web es el intercambio de información entre el browser y el Server. Angular se encarga de cambiar las páginas que tiene cargadas en el browser.



Single Page Application - SPA Rendering Model (Idealized)

Lo que se hace es «actualizar» solo un «cachito» de una página, en lugar de la página completa, como se hacía antiguamente; el cachito que le indiquemos va cambiando, pero el resto de la página sigue igual, sin modificación. Para eso se usa el tag <router-outlet>**.** Angular tiene un mecanismo que escucha los cambios de URL del browser y, dependiendo de la configuración, va a cambiar ese <router-outlet> por el Template correspondiente a la URL



3. Que significa IaaS, PaaS y SaaS .

Software as Service (SaaS):

Básicamente se trata de cualquier servicio basado en la web. Tenemos ejemplos claros como el Webmail de Gmail, los CRM onlines. En este tipo de servicios nosotros accedemos normalmente a través del navegador sin atender al software. Todo el desarrollo, mantenimiento, actualizaciones, copias de seguridad es responsabilidad del proveedor.

En este caso tenemos poco control, nosotros nos situamos en la parte más arriba de la capa del servicio. Si el servicio se cae es responsabilidad de proveedor hacer que vuelva a funcionar.

Infraestructure as Service (IaaS):

En este caso con IaaS tendremos mucho más control que con PaaS, aunque a cambio de eso tendremos que encargarnos de la gestión de infraestructura El ejemplo perfecto es el proporcionado por Amazon Web Service (AWS) que no provee una serie de servicios como EC2 que nos permite manejar maquinas virtuales en la nube o S3 para usar como almacenamiento. Nosotros podemos elegir

qué tipo de instancias queremos usar LInux o Windows, así como la capacidad de memoria o procesador de cada una de nuestras maquinas. El hardware para nosotros es transparente, todo lo que manejamos es de forma virtual.

La principal diferencia es que nosotros nos encargamos de escalar nuestras aplicaciones según nuestras necesidades, además de preparar todo el entorno en las maquinas (aunque existen imágenes de instancias preparadas con las configuraciones más comunes).

Además de AWS nos encontramos ejemplos como Rackspace Cloud o vCloud de VMWare.

Plataform as Service (PaaS):

Es el punto donde los desarrolladores empezamos a tocar y desarrollar nuestras propias aplicaciones que se ejecutan en la nube. En este caso nuestra única preocupación es la construcción de nuestra aplicación, ya que la infraestructura nos la da la plataforma.

Es un modelo que reduce bastante la complejidad a la hora de desplegar y mantener aplicaciones ya que las soluciones PaaS gestionan automáticamente la escalabilidad usando más recursos si fuera necesario. Los desarrolladores aun así tienen que preocuparse de que sus aplicaciones estén lo mejor optimizadas posibles para consumir menos recursos posibles Pero todo ello sin entrar al nivel de máquinas.

Ejemplos populares son Google App Engine que permite desarrollar aplicaciones en Java o Python desplegándolas en la infraestructura que provee Google, cosa que también hace Heroku con Rails y Django.

Para los desarrolladores que ignoran la infraestructura que deben montar y sólo quieren preocuparse de escribir software, esta es la alternativa a seguir.

4. Que es Firebase, Firestore y explique a que se refiere cuando se habla de Baas.

Firebase:

Provee una API para guardar y sincronizar datos en la nube en tiempo real.

Sus características fundamentales están divididas en varios grupos, las cuales podemos agrupar en:

- Analíticas: Provee una solución gratuita para tener todo tipo de medidas para gestionarlo todo desde un único panel.
- Desarrollo: Permite construir mejores apps, permitiendo delegar determinadas operaciones en Firebase, para poder ahorrar tiempo, evitar bugs y obtener un aceptable nivel de calidad. Entre sus características destacan el almacenamiento, testeo, configuración remota, mensajería en la nube o autenticación, entre otras.
- Crecimiento: Permite gestionar los usuarios de las aplicaciones, pudiendo además captar nuevos. Para ello dispondremos de funcionalidades como las de invitaciones, indexación o notificaciones.
- Monetización: Permite ganar dinero gracias a AdMob.

Firestore:

Cloud Firestore es una base de datos flexible y escalable para la programación en servidores, dispositivos móviles y la Web desde Firebase y Google Cloud Platform. Al igual que Firebase Realtime Database, mantiene tus datos sincronizados entre apps cliente a través de agentes de escucha en tiempo real y ofrece asistencia sin conexión para dispositivos móviles y la Web, por lo que puedes compilar apps con capacidad de respuesta que funcionan sin importar la latencia de la red ni la conectividad a Internet. Cloud Firestore también ofrece una integración sin interrupciones con otros productos de Firebase y Google Cloud Platform, incluido Cloud Functions.

Backend as Service (BaaS):

La arquitectura BaaS es una de las más recientes arquitecturas cloud que han repuntado en los últimos años, la cual consiste en olvidarnos por completo del concepto de servidores y aplicaciones, incluso, hasta casi nos podríamos olvidar de las bases de datos, pues BaaS nos ofrece una nueva forma de crear todo el BackEnd de nuestras aplicaciones basadas en Cloud Functions las cuales son por lo general funciones escritas en JavaScript y que luego BaaS las expone como servicios, de tal forma que en lugar de tener una aplicación con cientos de objetos y procedimientos, tenemos una serie de Cloud functions, las cuales viven exclusivamente en la nube.

5. Defina o explique Angular es lo mismo que React. Si son distintos liste cuales son las diferencias.

Framework o Librería:

Angular es un Framework, es decir es una solución todo en uno que dispone de todas las herramientas necesarias para llevar a cabo una aplicación, lo que implica que todo tengas que hacerlo "a su manera" pero te despreocupas de tener que buscar la forma de implementar diversas funcionalidades.

React es una librería que solo se encarga de la vista, lo que implica que el resto de las herramientas para hacer una aplicación hay que definir cuales usar, pero esto logra una mayor flexibilidad, ya que al poder definir todos los componentes por separado se pueden cambiar librerías que se desarrollen en un futuro y mejoren la aplicación sin grandes alteraciones.

Arquitectura:

Angular maneja una arquitectura básica MVC, disponemos de Componentes para la vista , Enrutador para la capa de control y servicios para la capa de backend. El paradigma usado es orientado a componentes.

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

Lenguaje:

Angular utiliza TypeScript que es un superconjunto y utiliza un transpiler para compilar el archivo .ts a un archivo .js normal. TypeScript ofrece extensiones de lenguaje que están diseñados para hacer escritura en JavaScript más fácil, y asocia la información de tipo con entidades de JavaScript para hacer cumplir la comprobación de tipos y mejorar el flujo de trabajo de desarrollo.

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

Angular centra sus plantillas en HTML, es decir escribimos cierta lógica en el HTML, trasladando javascript a HTML, esto implica que se mantiene un Html y javascript por componente lo cual da mas claridad de las cosas pero la detección de errores en una plantilla se produce en tiempo de ejecución, aportando además una información poco determinante para encontrar el error

React toda la lógica y vista permanece en javascript, es decir, se traslada HTML a Javascript. Esto genera código mas centralizado, pero pueden ser archivos

bastantes grandes por componentes, la detección de errores se genera en la compilación de la plantilla, aportando información acerca del error y la línea que provoca el error.

En cuanto a los temas, ambos tienen una gran variedad de temas disponibles y muy parecidos en look and feel y de hecho Bootstrap con Material Design puede ser implementado en ambos.