	Oficina de transferencia de resultados de investigación
	Vicerrectoría de investigaciones de la Universidad del Valle
	Estructura de código para aplicaciones web en desarrollo en el marco de los proyectos de sistematización de procesos y gestión de laboratorios en ejecución por la vicerrectoría de investigaciones de la Universidad del Valle
	Elaborado por Julian Felipe Rodríguez Muñoz

La estructura de código de las aplicaciones de los proyectos en desarrollo para la OTRI de la Universidad del Valle, contempla la utilización de repositorios en [Github](#) para preservar el código en línea, gestionar las versiones de las aplicaciones y utilizar su sistema de seguimiento de corrección de errores, solicitud y desarrollo de funcionalidades.

Los repositorios deben contar con todo el código necesario para desplegar la aplicación en cualquier servidor, es por ello que se propone mantener en el código tanto de la aplicación cliente como de las aplicaciones del lado del servidor pertenecientes al mismo proyecto en un mismo repositorio. Para ello se propone conservar una estructura como indica la ilustración 1. A continuación se describe la estructura del código fuente de las aplicaciones web, propuesta para la utilización en los repositorios:

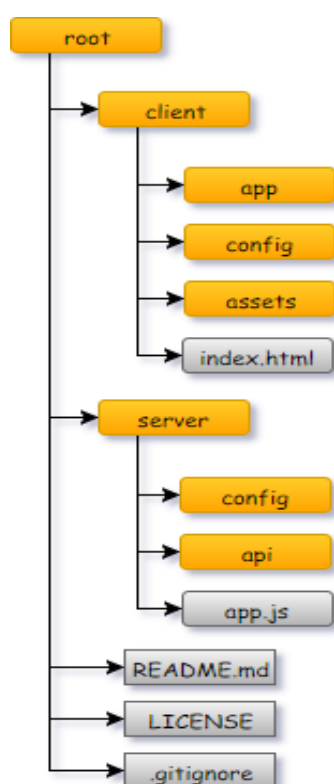


Ilustración 1: Estructura de repositorios de aplicaciones de la Universidad del Valle


Carpeta raíz o estructura de repositorio:

Como se dijo con anterioridad, se desea mantener todo el código de la aplicación en un mismo sitio, es por ello que la estructura propuesta para el repositorio (representado en la ilustración 1 como carpeta raíz o “root”) incluye una carpeta para la aplicación cliente, una carpeta para la aplicación del lado del servidor, una archivo de especificación de licencia bajo la cual fue desarrollado el software del proyecto, un archivo de configuración del git (sistema de control de versiones) y un archivo “readme” en el cual se describe el proyecto y especifica cómo hacer un despliegue de la aplicación.

Código aplicación cliente:

El código perteneciente a la aplicación cliente contempla un directorio para una aplicación desarrollada en [Angular](#), acompañado por un directorio de configuración y uno más para mantener el contenido estático que utilizará la aplicación (como logos, imágenes y reglas de estilo por ejemplo).

El directorio de configuración debe albergar archivos que definan variables de configuración para el uso de la aplicación, esto es por ejemplo, las llaves de acceso a servicios de terceros y el idioma predeterminado de la aplicación. Por otro lado el directorio de “assets” tiene una estructura definida por tipo de archivos (como indica la ilustración 2) que permitirá organizar el contenido estático de la plantilla principal de la aplicación.

	Oficina de transferencia de resultados de investigación
	Vicerrectoría de investigaciones de la Universidad del Valle
	Estructura de código para aplicaciones web en desarrollo en el marco de los proyectos de sistematización de procesos y gestión de laboratorios en ejecución por la vicerrectoría de investigaciones de la Universidad del Valle
	Elaborado por Julian Felipe Rodríguez Muñoz

El directorio “app” contiene la aplicación desarrollada haciendo uso del framework [Angular](#), separada por módulos de funcionalidades que requiera la aplicación en el marco del proyecto en desarrollo. Cada módulo tiene un número de componentes que en conjunto construyen la funcionalidad requerida para dicho módulo. La aplicación como tal conserva una estructura simple de dos directorios, una de código compartido y otro perteneciente los componentes únicos de la aplicación.

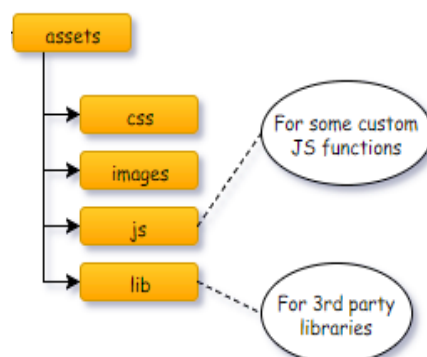


Ilustración 2: Estructura de directorio “assets” para código de aplicación cliente

El directorio compartido alberga componentes de la aplicación cuya función es utilizada por múltiples módulos de la aplicación (como la componente de autenticación, roles y permisos de un usuario por ejemplo). Del mismo modo se plantea propone la utilización de este directorio para albergar las plantillas visuales de la aplicación y componentes reutilizables de la interfaz a través de los diferentes módulos de funcionalidad de la aplicación.

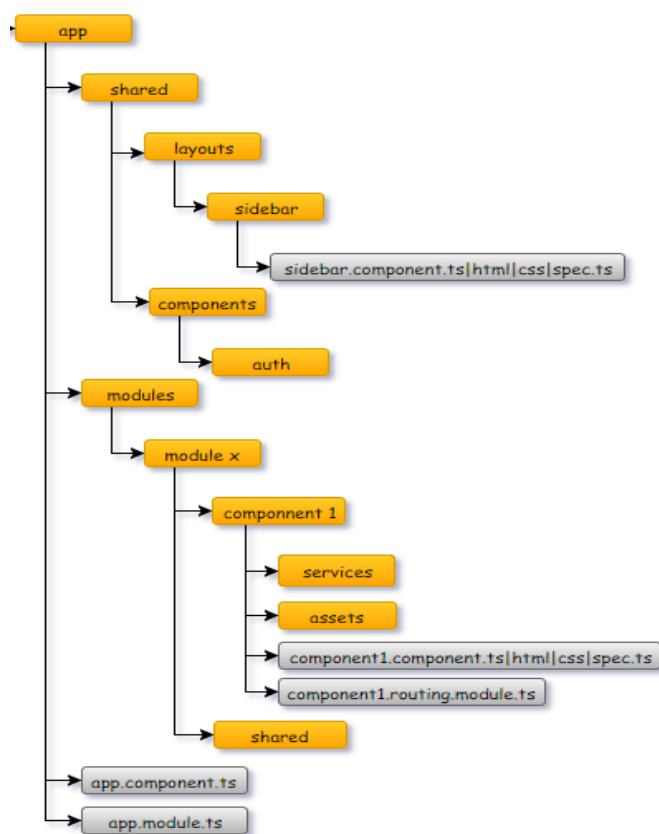



Ilustración 3: Estructura de la aplicación Angular

Por lo anterior, el directorio compartido o “shared” contempla dos directorios en su interior, uno para los componentes compartidos y otro para las plantillas compartidas en la aplicación (tal y como se indica en la ilustración 3).

Adicionalmente y como se dijo con anterioridad, la estructura contempla el desarrollo por módulos de la aplicación, en donde cada módulo está compuesto por uno o más componentes. Cada módulo puede o no contar con un directorio compartido de que albergará funciones de ayuda reutilizables en varios componentes, estas puede ser desde funciones propias de procesamiento hasta servicios (conexiones con aplicaciones externas).

Finalmente, cada componente perteneciente a un módulo debe incluir sus propias servicios y assets (si hace uso de ellos).

Estructura de código para aplicaciones web en desarrollo en el marco de los proyectos de sistematización de procesos y gestión de laboratorios en ejecución por la vicerrectoría de investigaciones de la Universidad del Valle.

	Oficina de transferencia de resultados de investigación
	Vicerrectoría de investigaciones de la Universidad del Valle
	Estructura de código para aplicaciones web en desarrollo en el marco de los proyectos de sistematización de procesos y gestión de laboratorios en ejecución por la vicerrectoría de investigaciones de la Universidad del Valle
	Elaborado por Julian Felipe Rodríguez Muñoz

Para conservar la estructura de la aplicación propuesta a través del desarrollo de los módulos y sus respectivos componentes, se hace necesario explicar el flujo de información al interior de los componentes de la aplicación. Tres factores se hacen necesarios para el flujo de la información al interior de un componente perteneciente a un módulo. La función de estos factores se explica a continuación:

Servicio: Se encarga de sustraer o enviar información requerida o enviada por el controlador desde o hacia sistemas externos.

Controlador: Se encarga de procesar la información proveída por sistemas externos o la interfaz de usuario, según sea el caso.

Vista: Se encarga de desplegar la información procesada por el controlador, así mismo provee información ingresada o modificada por el usuario para que sea nuevamente procesada por el controlador.

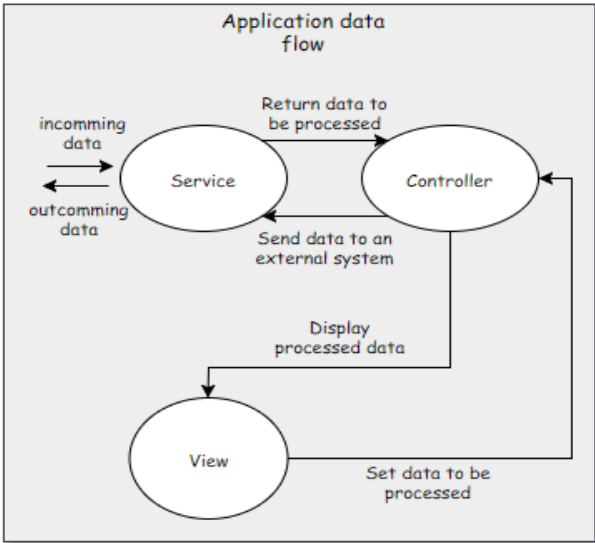


Ilustración 4: Flujo de información al interior de los componentes de la aplicación

Finalmente en la ilustración 4 se puede observar el flujo de la información a través de los factores de los componentes.

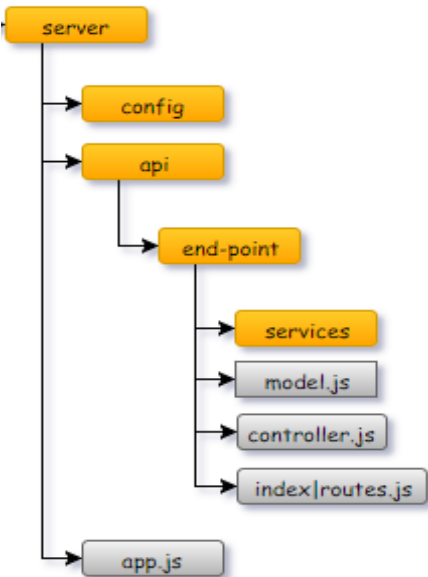



Ilustración 5: Estructura de código de la aplicación del lado del servidor

Código de aplicación de servidor:

Siguiendo la misma línea de estructura del código de la aplicación cliente, el código del lado del servidor contiene principalmente dos directorios, uno de configuración (que cumple la misma función que en el código de la aplicación cliente) y otro de la aplicación de interfaz de comunicación marcado como “api” en la ilustración 5. En dicho directorio se propone utilizar una estructura por punto de acceso o recurso, es decir, por cada

El directorio de la interfaz de programación contendrá tantos directorios como recursos o puntos de acceso se considere y en ellos se conserva una estructura que también respeta el flujo de la información propuesto para la aplicación cliente. Cada punto de acceso o recurso cuenta con una definición de rutas o URLs para acceder a la información a través de los

	Oficina de transferencia de resultados de investigación
	Vicerrectoría de investigaciones de la Universidad del Valle
	Estructura de código para aplicaciones web en desarrollo en el marco de los proyectos de sistematización de procesos y gestión de laboratorios en ejecución por la vicerrectoría de investigaciones de la Universidad del Valle
	Elaborado por Julian Felipe Rodríguez Muñoz

diferentes métodos que ofrece el protocolo HTTP, así como un controlador que define un comportamiento para cada una de las rutas establecidas. Cada recurso contará, de ser necesario, con un directorio de servicios para la adquisición o emisión de información, así como de un modelo de base de datos.

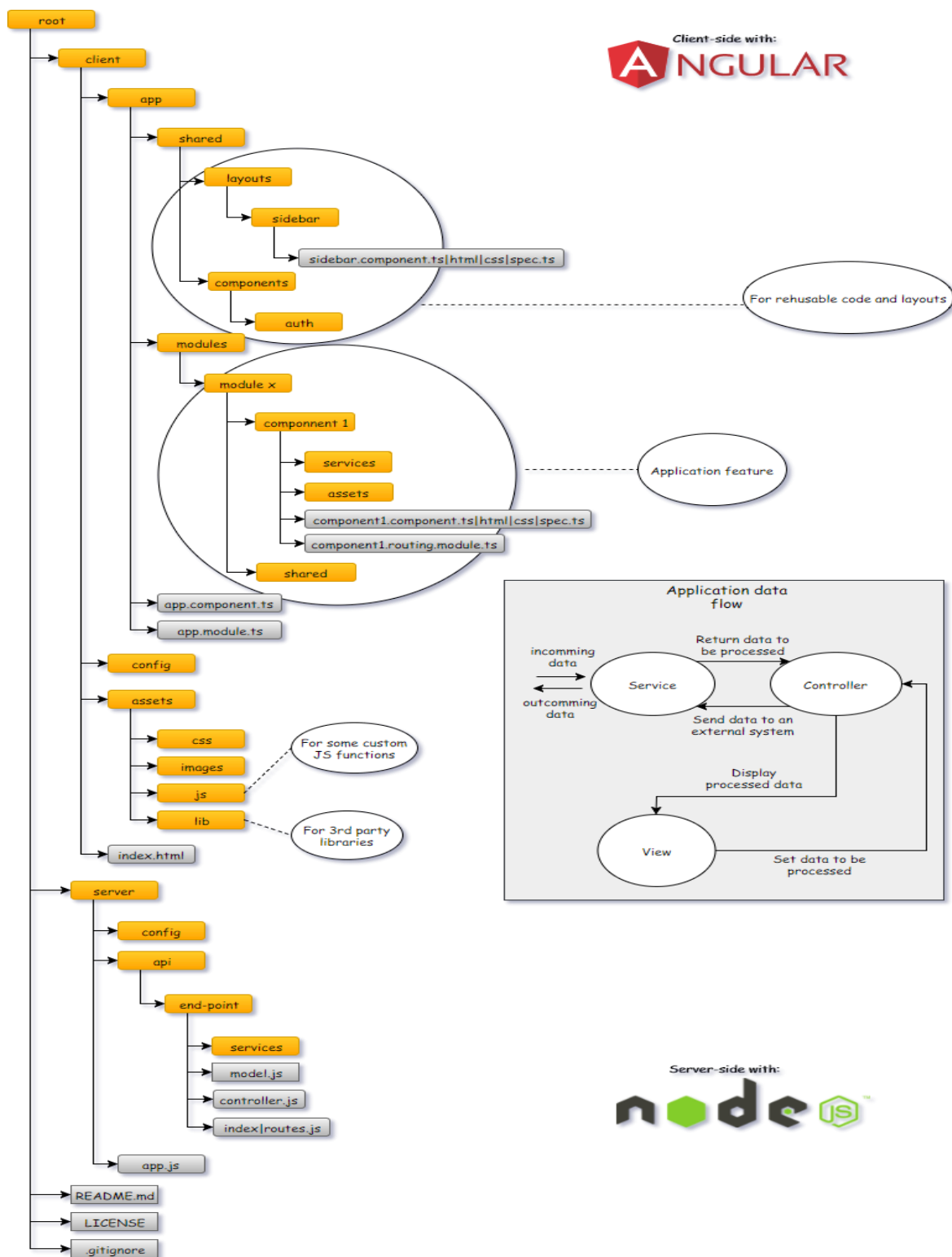


Ilustración 6: Estructura de código completa

Estructura de código para aplicaciones web en desarrollo en el marco de los proyectos de sistematización de procesos y gestión de laboratorios en ejecución por la vicerrectoría de investigaciones de la Universidad del Valle.