# Responder a las siguientes consignas

1. ¿Qué es un componente? ¿De qué manera realizaría el intercambio de información entre 2 componentes?

Un componente es básicamente una parte de la UI o interfaz que verá el usuario al acceder a cada uno de los componentes disponibles en la aplicación. Cada uno de estos puede tener una combinación de código HTML, hojas de estilo propias, un módulo de inicialización con las dependencias necesarias y una sección de código que hará a la lógica propia de este componente.

1. ¿Cómo gestionaría el diseño responsivo de un componente?

Particularmente opero los aspectos responsivos de un componente o de los componentes de la aplicación usando la integración de Bootrstap en la aplicación.

1. Nombre 3 directivas estructurales. Ejemplifique el uso de 1

Las directivas estruccturales son los que conocemos como estructuras de control, solo que en Angular estas directivas permiten alterar o tomar datos de las etiquetas HTML del dom de la página.

Ejemplo

<div \*ngIf="unCargo.estado; then thenBlockDelete else elseBlockDelete"></div>

<ng-template #thenBlockDelete>

<a class="btn btn-danger" href="#" [routerLink]="['/cargo/delete/', unCargo.id]" role="button"><i class="fas fa-trash-alt"></i></a>

</ng-template>

<ng-template #elseBlockDelete>

<a class="btn btn-danger disabled" href="#" [routerLink]="['/cargo/delete/', unCargo.id]" role="button"><i class="fas fa-trash-alt"></i></a>

</ng-template>

Este ejemplo en particular es el que usé dentro del proyecto para mostrar un botón habilitado o no dependiendo del estado del objeto que se observa.

1. ¿Cómo protegería el acceso a las rutas de un componente?

En Angular podemos usar lo que se conocen como Guards (guardias o guardianes). Particularmente en este proyecto no los he incorporado.

1. ¿Qué es un observable? De un ejemplo de definición y uso.

Un observable es un tipo que permite su uso dentro de aplicaciones reactivas con la posibilidad de reaccionar ante eventos de un “subject”. Particularmente comprenden el conjunto de datos o el dato que estemos esperando luego de hacer una llamada a una dirección, y con un observable podemos capturar estos datos y manipularlos el UI de la Aplicación.

1. Diferencia entre constructor y OnInit.

El constructor es un método más de los que pueden estar provistos en la clase y es lo que se ejecuta al momento de crearse una instancia de una clase especifica. En cambio, el onInit, es un método que se ejecuta una vez se ha ejecutado el constructor, y se usa para establecer en la “clase” los datos o las acciones que requerimos antes que el usuario tenga cualquier interacción con el componente.