

EXAMEN UNIDAD 3 – DESARROLLO FRONTEND

NOMBRE DEL ESTUDIANTE: Velazquez Nava Jose Alberto 191190060DOCENTE: ING. AQUINO SEGURA ROLDAN

1 - ¿Cuáles son los 3 elementos que lleva el decorador component? (1 pts)

selector: styleUrl: templateUrl:

2 - ¿Cuál es el comando para poder generar un componente en angular? (1 pts)

ng g c newComponent

3 - Si tenemos una propiedad en TS "*public verdad : boolean = true;*" y en la sección de html tenemos un div con un h1 que dice hola mundo, como podemos mostrar el div si la propiedad verdad es verdadera y que no muestre el div si la propiedad verdad es false. Escribe el código solo de la sección html. (2 pts)

4 - Crea un objeto que tenga id, nombre y rfc como propiedades, a su vez agrega al menos 3 elementos o registros con esas propiedades, una vez terminado el objeto muestra su información en una tabla html. (1 pts)

5 - Crear una interfaz con los atributos nombre, paterno y materno. (1 pts)

6 - Tipar y crear un objeto, agregar al menos 2 registros con la interfaz creada en la pregunta 5. (1 pts)

7 - ¿Para poder utilizar una page o un main-page la agregamos en la sección de export o import? (1 pts)

8 - Escribe la estructura de carpetas que usaríamos en un proyecto real, según lo aprendido. (1 pts)

9 - Si tengo un componente llamado app-listado1 y quiero declararlo en el html de mi modulo principal con un input llamado personaje, ¿Cómo escribo esa declaración? (1 pts)

10 - Se requiere crear dos componentes uno llamado padre y otro hijo, el componente hijo desea enviar al componente padre el mensaje hola, declara el código y la explicación de cómo podemos llevar a cabo esto. (5 pts)

```
3: <div *ngIf = "verdad">
    <h1> hola mundo </h1>
</div>
```

4: // Ts.

```
public inventarios: any[] [
    {
        id: 0,
        nombre: "Jose A",
        rfc: "VENA 010416 H11W7"
    },
    {
        id: 1,
        nombre: "Mitzi G",
        rfc: "MGBP 980113 M1179"
    },
    {
        id: 2,
        nombre: "Aline 11",
        rfc: "VENA 050428 M1LA9"
    }
]
```

// html.

```
<table>
  <tbody>
    <tr *ngFor = "let inventario of inventarios">
      <td> {{ inventario.id }} </td>
      <td> {{ inventario.nombre }} </td>
      <td> {{ inventario.rfc }} </td>
    </tr>
  </tbody>
</table>
```

5: export interface Datos {

nombre: string;

paterno: string;

materno: string;

}

6: public infoPersonal: Datos[] = [

{ nombre: "Jose A", paterno: "Velazquez",
materno: "Nava" },

{ nombre: "Mitzi G", paterno: "Benitez",
materno: "Pérez" },

];

7:

exports

8: modulo

├ components

├ interfaces

├ pages

└ modulo.modules.ts

9:

< app-listado1 [personaje] = "personajes" >

< /app-listado1 >

10.

Componente padre

↙ componente hijo

// ts.

```
public obtenerMensaje (mensaje:string)
: void {
```

```
    alert (mensaje);
```

```
}
```

// html

<app-componente-hijo>

{ mensaje } = "obtenerMensaje (evento)"

>

</app-componente-hijo>

Explicación:

el componente hijo

con el metodo emitMensaje

en el doc html creamos un boton agregando el evento click para el metodo creado.

Posteriormente en el componente padre creamos el metodo para recibir el dato y en html declaramos el componente hijo con el input (mensaje). llamando el metodo creado y agregando como parametro \$event que se recibe del

// ts.

```
//decorador Output
```

```
@Output()
```

```
// evento
```

```
public mensaje: EventEmitter<string>
```

```
= new EventEmitter();
```

```
public emitMensaje(): void {
```

```
    this.mensaje.emit("hola");
```

// html

<button (click) = "emitMensaje()"> send

</button>

usar el decorador output y el EventEmitter,

podemos enviar la información a cualquier componente

un boton agregando el evento click para el

metodo creado.

Posteriormente en el componente padre creamos el metodo para recibir el dato y en html declaramos el componente hijo con el input (mensaje).

llamando el metodo creado y agregando como parametro \$event que se recibe del