

# EXAMEN TEORICO ANGULAR

1.- ¿Cómo mostrarías en un párrafo de un componente el valor de una propiedad de su clase llamada “edad”?

**Con la Interpolación.**

`<p> La edad es de: {{ edad }} </p>`

2.- Un componente padre muestra en su vista un hijo “cabecera” y otro hijo “item” tantas veces como elementos haya en un array de su clase asociada llamada “items”. ¿Cómo sería el código de esa vista?

```
<div>
  <app-cabecera></app-cabecera>
  <div *ngFor="let item of items">
    <app-item [data]="item"></app-item>
  </div>
</div>
```

3.- Explica de forma detenida como se produce el arranque de una aplicación Angular.

**Una vez que haces el `ng serve -o`, se inicializa el `AppModule`, en este es donde está definido todos los componentes, servicios... Carga el componente raíz, que angular lo va a encontrar dentro del `AppModule`. Lo encuentra y ejecuta una serie de acciones. Después de ejecutarse eso examina el html del componente y lo muestra en el navegador.**

4.- Si en la vista de un componente aparece el siguiente código `<hijo>Texto</hijo>`, ¿qué características de Angular estoy usando? ¿Cuál es su utilidad?

**Estas utilizando la característica de componentes hijos.**

**La utilidad de los componentes hijos es dividir la interfaz de usuario en partes más pequeñas y reutilizables, lo que facilita la construcción, mantenimiento y escalabilidad de la aplicación.**

5.-Escribe el código del controlador de un componente hijo que envíe a su padre , en forma de Output, el numero de veces que se ha pulsado un botón. Escribe la linea de código que necesita el padre para acceder a esos datos.

```
export class Hijo {  
  contadorClicks: number = 0;  
  @Output() contadorActualizado = new EventEmitter<number>();  
  
  incrementarContador() {  
    this.contadorClicks++;  
    this.contadorActualizado.emit(this.contadorClicks);  
  }  
}
```

La línea de código que necesita el padre es:

```
<app-child (contadorActualizado)="actualizarContador($event)"></app-child>
```

6.-Crea una directiva en Angular llamada “MiDirectiva” que cambie el color de fondo del elemento al que se le aplica, utilizando un color personalizado proporcionado como parametro, y proporciona un ejemplo de uso en un componente.

```
export class MiDirectivaDirective implements OnInit {  
  @Input('miDirectiva') colorPersonalizado: string;  
  constructor(private el: ElementRef, private renderer: Renderer2) {}  
  ngOnInit() {  
    this.renderer.setStyle(this.el.nativeElement, 'background-color', this.colorPersonalizado);  
  }  
}
```

Ejemplo de uso en un componente:

```
<div miDirectiva="lightblue">
```

Este es un ejemplo de componente que utiliza la directiva MiDirectiva.

```
</div>
```

7.- Implementa un formulario model-driven en Angular con campos “Nombre” (cadena de texto requerida) y “Edad” (numero mayor a 0).

```
export class Formulario implements OnInit{  
    formulario: FormGroup;  
    constructor(private fb: FormBuilder) {}  
    ngOnInit(): void {  
        this.formulario = this.fb.group({  
            nombre: ['', Validators.required],  
            edad: [null, Validators.min(1)]  
        });  
    }  
}
```

8.- Crea un servicio “Contador” con un método que devuelva un observable donde recogeremos el numero de veces que se ha llamado a otro método del servicio llamado “contar”. Pon un ejemplo de uso de este servicio.

```
export class ContadorService {  
    private contador: number = 0;  
    private contadorSubject = new Subject<number>();  
  
    constructor() {}  
  
    contar(): void {  
        this.contador++;  
        this.contadorSubject.next(this.contador);  
    }  
  
    obtenerContador(): Observable<number> {  
        return this.contadorSubject.asObservable();  
    }  
}
```

Ejemplo de uso:

```
export class EjemploComponenteComponent implements OnInit {  
  contadorActual: number;  
  
  constructor(private contadorService: ContadorService) { }  
  
  ngOnInit(): void {  
    this.contadorService.obtenerContador().subscribe(contador => {  
      this.contadorActual = contador;  
    });  
  }  
  
  incrementarContador() {  
    this.contadorService.contar();  
  }  
}
```

9.- En Angular, ¿cómo se capturan y utilizan los parámetros de una ruta en un componente?

En primer lugar defines las rutas en el app-routing.ts :

```
const routes: Routes = [  
  { path: 'producto/:id', component: DetalleProductoComponent }  
];
```

Y luego se utilizan así en un componente:

```
import { ActivatedRoute } from '@angular/router';  
export class DetalleProductoComponent implements OnInit {  
  productId: string;  
  constructor(private route: ActivatedRoute) { }  
  ngOnInit(): void {  
    this.route.params.subscribe(params => {  
      this.productId = params['id'];  
    });  
  }  
}
```