

Historia de Angular(.js)



- Miško Hevery creó 'GetAngular' en 2009 como side project.
- Trabajaba en una herramienta llamada Google Feedback.
- Hizo una apuesta con su manager de reescribirlo en 2 semanas.
- Tardó 3 pero pasó de 17.000 líneas a solo 1.500.
- Miško junto a su manager (Brad Green) convirtieron 'GetAngular' en Angular.js.
- Vamos, un refactor en toda regla.

Angular Renaissance



Angular



Modern Angular

Jesús Rodríguez Rodríguez



Standalone

```
@NgModule({
  declarations: [
    AppComponent,
    HeaderComponent,
    FooterComponent,
    HomeComponent,
    FormatPipe,
  ],
  imports: [BrowserModule, RouterModule, SharedModule],
  providers: [UserService],
  bootstrap: [AppComponent],
})
export class AppModule {}
```

Standalone

```
@Injectable({  
  providedIn: 'root'  
})  
export class UserService {  
  // Código  
}
```

```
@Component({  
  standalone: true,  
  imports: [NxWelcomeComponent, RouterOutlet],  
  selector: 'app-root',  
  templateUrl: './app.component.html',  
  styleUrls: ['./app.component.scss'],  
})  
export class AppComponent {  
  title = 'modules';  
}
```

```
export const appConfig: ApplicationConfig = {  
  providers: [  
    provideZoneChangeDetection({ eventCoalescing: true }),  
    provideRouter(appRoutes),  
  ],  
};
```

Standalone - v19

```
@Component({
  selector: 'app-root',
  imports: [RouterOutlet],
  template: '<router-outlet></router-outlet>',
  changeDetection: ChangeDetectionStrategy.OnPush,
  encapsulation: ViewEncapsulation.None
})
export class AppComponent {
  title = 'Hola DotNetMalaga';
}
```

Inject()

```
export class MyComponent {  
  constructor(private dataService: DataService) {}  
  
  changeLanguage() {  
    this.dataService.currentLanguage.set('fr');  
  }  
}
```

Inject()

```
export class MyComponent {  
  #dataService = inject(DataService);  
  
  changeLanguage() {  
    this.#dataService.currentLanguage.set('fr');  
  }  
}
```


Inject()

```
import { inject } from '@angular/core';  
import { DataService } from '../data.service';  
  
export const injectCurrentLanguage = () => inject(DataService).currentLanguage;
```

```
export class MyComponent {  
  currentLanguage = injectCurrentLanguage();  
  
  changeLanguage() {  
    | this.currentLanguage.set('fr');  
  }  
}
```

Control de Flujo - @if

```
<ng-container *ngIf="user() as user; else noUser">
  <h1>{{ user.name }}</h1>
  <p>{{ user.company }}</p>
</ng-container>

<ng-template #noUser>
  <p>No user</p>
</ng-template>
```

```
@if (user(); as user) {
  <h1>{{ user.name }}</h1>
  <p>{{ user.company }}</p>
} @else {
  <p>No user</p>
}
```

Control de Flujo - @else-if

```
@if (a > b) {  
|   {{a}} is greater than {{b}}  
} @else if (b > a) {  
|   {{a}} is less than {{b}}  
} @else {  
|   {{a}} is equal to {{b}}  
}
```

Control de Flujo - @for

```
@for (framework of frameworks(); track framework.id) {  
  <p>{{ framework.name }}</p>  
} @empty {  
  <p>No frameworks web(mentira!)</p>  
}
```

Control de Flujo - @switch

```
@switch (user().role) {  
    @case (role.ADMIN) {  
        <dotnet-admin />  
    }  
    @case (role.USER) {  
        <dotnet-user />  
    }  
}
```

Control de Flujo - @let

```
<ng-container *ngIf="products$ | async as products">
  <ul>
    <li *ngFor="let product of products">
      {{ product.name }} - {{ product.price }}
    </li>
  </ul>
</ng-container>
```

Control de Flujo - @let

```
@let products = products$ | async;

@if (products) {
<ul>
  @for (product of products; track product) {
    <li>{{ product.name }} - {{ product.price }}</li>
  }
</ul>
} @else {
  <p>No products</p>
}
```

Control de Flujo - @defer

- Permite cargar componentes de manera lazy
- Puede retrasar los componentes que no están visible en pantalla
- Muestra "placeholders" mientras no está cargada esa porción de template
- Cargan bajo ciertos eventos:
 - Idle
 - Viewport
 - Interaction
 - Hover
 - Immediate
 - Timer

Control de Flujo - @defer

```
@defer {  
  <componente-lento-de-la-muerte />  
}
```

Control de Flujo - @defer

Initial chunk files	Names	Raw size	Estimated transfer size
chunk-WUPHH373.js	-	106.13 kB	32.08 kB
polyfills-FFHMD2TL.js	polyfills	34.52 kB	11.28 kB
main-QDOXTWRA.js	main	17.75 kB	5.69 kB
styles-5INURTS0.css	styles	0 bytes	0 bytes
	Initial total	158.40 kB	49.05 kB
Lazy chunk files	Names	Raw size	Estimated transfer size
chunk-MRRY4GLD.js	lento-component	351 bytes	351 bytes

Control de Flujo - @defer

```
@defer (on idle) {  
  <componente-lento-de-la-muerte />  
}
```

Control de Flujo - @defer

```
@defer (on viewport) {  
  <componente-lento-de-la-muerte />  
}  
  
@placeholder {  
  <div>Cargando...</div>  
}
```

Control de Flujo - @defer

```
@defer (on interaction) {  
  <componente-lento-de-la-muerte />  
}  
  
@loading (after 100ms;minimum 1s) {  
  <div>Lo que se ve mientras se procesa</div>  
}  
  
@placeholder {  
  <div>Lo que se ve mientras no se procesa</div>  
}
```

Control de Flujo - @defer

```
<h1 #cabecera>Esta cabecera tiene un huevo de pascua</h1>
  @defer (on interaction(cabecera)) {
    <componente-lento-de-la-muerte />
  }
,
```

Simplificación de código

```
@Injectable({
  providedIn: 'root',
})
export class AuthGuard implements CanActivate {
  canActivate(
    route: ActivatedRouteSnapshot,
    state: RouterStateSnapshot
  ): boolean | UrlTree | Observable<boolean | UrlTree> | Promise<boolean | UrlTree> {
    // Example logic: Check if the user is authenticated
    return !!localStorage.getItem('token');
  }
}
```

Simplificación de código

```
import { CanActivateFn } from '@angular/router';

export const authGuard: CanActivateFn = (route, state) => {
  // Example logic: Check if the user is authenticated
  return !!localStorage.getItem('token');
};
```


Simplificación de código

```
@Injectable()
export class AuthInterceptor implements HttpInterceptor {
  intercept(req: HttpRequest<any>, next: HttpHandler): Observable<HttpEvent<any>> {
    // Example logic: Add an Authorization header
    const authToken = localStorage.getItem('token');
    if (authToken) {
      const clonedRequest = req.clone({
        headers: {
          Authorization: `Bearer ${authToken}`,
        },
      });
      return next.handle(clonedRequest);
    }
    return next.handle(req);
  }
}
```

Simplificación de código

```
export const authInterceptor: HttpInterceptorFn = (req, next) => {  
  // Example logic: Add an Authorization header  
  const authToken = localStorage.getItem('token');  
  if (authToken) {  
    const clonedRequest = req.clone({  
      setHeaders: {  
        Authorization: `Bearer ${authToken}`,  
      },  
    });  
    return next(clonedRequest);  
  }  
  return next(req);  
};
```

Zone - Zoneless



Signals

```
@Component ({
  selector: 'app-root',
  template: `
    <h1>{{title()}}</h1>
  `,
  changeDetection: ChangeDetectionStrategy.OnPush,
  encapsulation: ViewEncapsulation.None
})
export class AppComponent {
  title = signal('Hola DotNetMalaga') // -> WritableSignal<string>
}
```

Signals

```
@Component({
  selector: 'app-root',
  template: `
    <h1>{{title()}}</h1>
    <button (click)="title.set('Algo ha cambiado')">Cambiar título</button>
  `,
  changeDetection: ChangeDetectionStrategy.OnPush,
  encapsulation: ViewEncapsulation.None
})
export class AppComponent {
  title = signal('Hola DotNetMalaga') // -> WritableSignal<string>
}
```

Signals - Computed

```
@Component({
  selector: 'app-root',
  template: `
    <p>{{nombreCompleto()}}</p>
  `,
  changeDetection: ChangeDetectionStrategy.OnPush,
  encapsulation: ViewEncapsulation.None
})
export class AppComponent {
  nombre = signal('Jesús');
  apellido = signal('Rodriguez');

  nombreCompleto = computed(() => `${this.nombre()} ${this.apellido()}`);
}
```

Signals - Computed

```
export class AppComponent {  
  pedidos = signal([  
    { fecha: new Date('2024-03-01'), producto: 'Portatil' },  
    { fecha: new Date('2024-03-15'), producto: 'Monitor' },  
    { fecha: new Date('2024-03-20'), producto: 'Teclado' }  
  ]);  
  
  ultimoPedido = computed(() => {  
    const listaPedidos = this.pedidos();  
    if (listaPedidos.length === 0) return null;  
  
    return listaPedidos.reduce((pedidoMasReciente, pedidoActual) =>  
      pedidoActual.fecha > pedidoMasReciente.fecha ? pedidoActual : pedidoMasReciente  
    );  
  });  
}
```


Signals - linkedSignal

```
@Component({
  selector: 'app-root',
  imports: [NgFor],
  template: `
    <select (change)="changeShippingMethod($event)">
      <option *ngFor="let method of shippingMethods()" [value]="method">{{method}}</option>
    </select>
    <p>{{selectedMethod()}}</p>
  `,
  changeDetection: ChangeDetectionStrategy.OnPush,
  encapsulation: ViewEncapsulation.None
})
export class AppComponent {
  shippingMethods = signal(['DHL', 'FedEx', 'UPS']);

  selectedMethod = linkedSignal(() => {
    return this.shippingMethods()[0];
  });

  changeShippingMethod(event: Event) {
    this.selectedMethod.set((event.target as HTMLSelectElement).value);
  }
}
```


Signals - Effect

```
export class AppComponent {  
  shippingMethods = signal(['DHL', 'FedEx', 'UPS']);  
  
  selectedMethod = linkedSignal(() => {  
    return this.shippingMethods()[0];  
  });  
  
  changeShippingMethod(event: Event) {  
    this.selectedMethod.set((event.target as HTMLSelectElement).value);  
  }  
  
  constructor() {  
    effect(() => {  
      window.localStorage.setItem('shippingMethod', this.selectedMethod());  
    });  
  }  
}
```

Signals - Inputs

```
export class MyComponent {  
  
  name = input.required<string>();  
  selected = output<boolean>();  
  
  twoway = model<string>();  
  
}
```

¿Qué pasa con RXJS? ¿Lo reemplaza
Signals?


NO

Resource

```
pokemonResult = input.required<PokemonListResult>({ alias: 'pokemon' });  
loaded = output<void>();  
  
pokemonResource = resource({  
  request: this.pokemonResult,  
  loader: ({ request: pokemon, abortSignal }) =>  
    | fetch(pokemon.url, { signal: abortSignal }).then(res => res.json()),  
  });
```

Resource

```
this.pokemonResource.
```

-  asReadOnly (method) WritableResource<{ name: stri...
-  destroy
-  error
-  hasValue
-  isLoading
-  reload
-  set
-  status
-  update
-  value

¿Qué más?

- nxResource
- toSignal
- toObservable

```
busqueda = signal('');  
  
busqueda$ = toObservable(this.busqueda).pipe(  
  | debounceTime(500),  
  | distinctUntilChanged()  
);
```

¿Futuro?

- Adios Karma (por fin)
- Formularios con Signals
- Componentes con Signals
- Router con Signals
- ¿Nueva sintaxis para componentes?

DEMO





EL EVENTO SOBRE

TECNOLOGÍAS
**CLOUD, WEB
Y DATA**

