Análise e desenvolvimento de sistemas

Adilson fábio furtado da Silva

Guilherme Duarte Faria

Douglas Cacimeiro

Tainah

**Trabalho sobre Angular JS**

São Paulo

2025

Adilson fábio furtado da Silva

Guilherme Duarte Faria

Douglas Cacimeiro

Tainah

trabalho sobre AngularJS

Trabalho apresentado à Universidade Anhanguera, como requisito parcial à aprovação no Terceiro semestre do curso de Análise e Desenvolvimento de Sistema.

São Paulo

2025

SUMÁRIO

[INTRODUÇÃO 3](#_Toc46509815)

[DESENVOLVIMENTO 4](#_Toc46509816)

[CONSIDERAÇÕES FINAIS 8](#_Toc46509817)

[REFERÊNCIAS 9](#_Toc46509818)

# INTRODUÇÃO

Esta pesquisa tem como objetivo analisar o framework Angular JS, desde o seu nascimento, como foi iniciado o projeto e para que foi iniciado, os seus princípios inovadores até seu impacto duradouro no universo do desenvolvimento web. Tentamos explorar a arquitetura única do Angular JS, sua importância para a popularização das chamadas “Single-Pages Aplications” (SAPs), as razões que levaram a uma evolução para o Angular.

# DESENVOLVIMENTO

1. **Contexto histórico**

No final dos anos 2000, o ambiente em torno do desenvolvimento web na parte de front-end se encontrava em um estado de transição. As aplicações web estavam em um momento de complexidade, pois via-se um aumento exponencial nas aplicações, mas, ao mesmo tempo, as ferramentas disponíveis ainda eram, na sua maioria, bibliotecas que eram manipuladas por dom, como é o caso do jQuery. Por esse motivo, os desenvolvedores web, ficavam mais tempo escrevendo códigos boilerplate para uma sincronização da interface do usuário (UI) com os dados pressupostos, algo que mostrava um processo mais propenso a erros e de difícil manutenção.

Observando esse contexto, foi que Misko Hevery, engenheiro de software que trabalhava na Google, deu inicio a um projeto framewoek chamado GetAngular. O objetivo por traz do desenvolvimento desse projeto era uma simplificação no desenvolvimento web para designers, o que permitiria que eles utilizassem a sintaxe do HTML para declarar o que a aplicação deveria fazer e não como ela deveria fazer. Por traz dessa ideia, existia o desejo de uma extensão do vocabilário do HTML por meio de um sistema de diretivas o que cruaria uma experiencia mais declarativa. Esse projeto ganhou uma guinada maior internamente quando Hevery e mais um colega começam a reescrever uma aplicação interna da Google. Originalmente essa aplicação era composta de 17 mil linhas de código, depois da reescrita, esse mesmo código ficou com apenas 1.500 linhas usando se4u novo framework. Essa efeito na mudança do código chamou a atenção da empresa. Assim a Google decidiu adotar, patrocinar 3e revatizar o projeto como Angular JS. O lançamento de sua primeira versão estável foi em 2012.

1. **Angular JS como marco no Desenvolvimento Web**

Esse novo projeto que chega como uma revolução, não foi somente mais biblioteca, mas sim um framework de opinião, já que ele veio para introduzir um novo paradigma na construção de SPAs. A importância desse novo framework não se limita a um conceito somente , mas em vários conceitos fundamentais, a saber:

* Two-Way Data Bilding – Essa talvez seja a característica mais famosa desse projeto. É um mecanismo que criava uma sincronização automática e bidirecional entre o Model (modelo) e o View (visualização). Assim que esses dados no Model mudavam, a interface do usuário (UI) era atualizada automaticamente, e o inverso também é uma realidade.

Com isso tiveram a eliminação da necessidade de manipular o DOM manualmente para mostrar mudanças de estado. Desse modo, houve uma redução drástica na quantidade de código e uma potencialização na produção.

* Injeção de dependência (DI) – O projeto Angular JS trouxe consigo um padrão de design de software, já consolidado no back-end, para o front-end. O seu sistema de DI estava intrínseco, isso permitia a criação de componentes desacoplados, testáveis e reutilizáveis. Esses serviços já comportam uma lógica de negócios, podendo ser facilmente injetados em controladores, filtros e outras partes da aplicação, o que facilita seus testes unitários.
* **Arquitetura MVC/MVVM**

O Angular JS impulsionou uma separação de concerns mais clara. O modelo era representado pelos dados, enquanto o View era o DOM (tendo seu aumento pelas diretivas) e o Controller era quem media a interação entre eles. Grande parte de Desenvolvedores defendem que a arquitetura do Angular JS tem uma semelhança bem mais voltada ao model-view-ViewModel (MVVM) onde o escopo atua como ViewModel.

* **Diretivas** – Aqui podemos dizer que está o coração do projeto. Algumas diretivas como ng-model, ng-repeat, ng-if e ng-click, davam permissão aos desenvolvedores para eles adicionarem comportamentos dinâmicos ao HTMLde uma forma mais intuitiva. Foi essa possibilidade de criação de diretivas customizadas que permitiu a criação de componentes web que podem ser reutilizados, isso antes da Era dos web Components standards. Podemos dizer assim que essas diretivas foi um divisor de águas no mundo da TI.
* **Testabilidade** - A idealização do design do framework, junto com sua DI e o desacoplamento, foi algo pensando em testes. Algumas ferramentas como o Karma (test runner) e jasmine (framework de teste) foram integradas ao seu ambiente, o que tornou o test-Drivin Development (TDD) uma prática viável e incentivavel para quem trabalha com front-end.

1. **Os Desafios e a necessidade de mudança para o Angular JS**

Apesar do sucesso que o projeto trouxe ao ser lançado, o Angular em sua primeira versão começou a mostrar sua limitações no decorrer da evolução da web e com isso as aplicações vão se tornando mais complexas. As problemáticas mais relevantes foram:

* Problema de desempenho – Apesar da conveniência da two-way data binding, ela também poderia se tornar um transtorno no desempenho. Cada uma das ligações ({{ }} ou ng-model) criava um $watch no sistema de digest cycle. Ter uma aplicação com muitos watchers deixavam a pesma aplicação muito lenta, já que o ciclo precisava verificar constantemente todas as expressões a fim de detectar essas mudanças.
* A complexidade da curva de aprendizagem - para os novatos que estavam aprendendo esse framework, os conceitos como $scope, prototypal, inheritance, e a API complexa de diretivas. Isso foi ocasionado pela falta de uma estrutura mais rígida levando a um código desorganizado em projetos mais robustos.
* Advento da web Moderna – Com o surgimento de novas tecnologias como a Web Components e uma crescente adoção de módulos ES6 só mostraram que a arquitetura do Angular JS ainda não estava preparada para os desafios futuros. A Web estava evoluindo em direção a uma abordagem baseadas em componentes.

Por conta destes fatores, a Google foi levada a tomar uma decisão radical, a saber, ao invés de tomar o caminho de iterar sobre o Angular JS, o caminho tomado foi de uma reescrita completa. Assim , foi anunciado em 2014 no ng-conf, o Angular (versão 2, e depois o 4, 5, etc.) era um framework novo, estruturado em TypeScript e focado em performance, mobilidade e escalabilidade.

Como principais mudanças no Angular 2.0 temo o TypeScript, que oferece tipagem estática, interfaces, decoradores e um melhor tooling, o que torna o código mais previsível e manipulável. Temos também a arquitetura baseada em componentes, que substitui completamente controladores e escopos. Tudo é um componente e promove um melhor encapsulamento e reutilização. Temos ainda zonas e change detection melhoradas, que é um mecanismo de detecção de mudanças mais eficientes, o que reduz consideravelmente o número de verificações necessárias.

Outra mudança é a compilação AOT (ahead-of-time) que faz com que o código seja compilado durante a build, resultando em aplicações mais leves e iniciam em uma velocidade muito mais rápido no navegador. Por fim , temos como mudança o Angular CLI, que uma ferramenta de linha de comando poderosa que faz a automação na criação de projetos, gerações de componentes, testes e deploy.

Com o tempo o Angular JS começou a ser considerado como modo de suporte a longo prazo em meados de julho de 2018, e foi descontinuado oficialmente em 31 de dezembro de 2021.

# CONSIDERAÇÕES FINAIS

Apesar de ser um framework descontinuado, o Angular JS tem um legado imensurável. Esse projeto foi um catalizador que demonstrou e apresentou ao mundo o quanto a aplicação web é complexa e rica e que poderiam ser construídas principalmente em front-end, para uma democratização no desenvolvimento SPAs. A maioria dos padrões e conceitos que o Angular JS popularizou, como injeções de dependência , testes unitário, binding declarativo, se tornaram como que um requisito básico para qualquer framework moderno.

A descontinuação do Angular, não é um sinal de fracasso do projeto, mas sim devolução. Essa transição para o Angular representa uma maturidade da plataforma web. Para aqueles desenvolvedores que iniciaram suas carreiras no caminho do Angular JS, significa muto entender sua história para uma melhor compreensão da evolução do desenvolvimento front-end. Em tomadas de decisões referentes a novos projetos, as principais escolhas são pelos frameworks Angular, React ou Vue.JS, porém a herança intelectual que essas ttecnologias tempara com o Angular JS é incontestável. No universo do desenvolvimento web, é inegável que ele foi um dos frameworks mais importantes e mais influentes de sua geração.

## REFERÊNCIAS

FREEMAN, Adan. *Pro Angular JS*: Build powerful and Dynamic Web Apps. Editora. Springer Nature BV. 1° ed. Indianapolis, IN, EUA. 2014.

SHAMEL, Brad; GREEN, Jeff; SESIA, Adan. *AngularJS: Up and Running: Enhanced productivity with Structured Web Apps.* Editora. O’Railly Media. O’Rail, USA. 2014.

LIPPMAN, Cory. *AngularJS.* Editora. Novatec. 2015.

Hevery, Misko. *AngularJS: The bad Parts. 2014 . Acesso em:* [**http://larseidnes.com/2014/11/05/angularjs-the-bad-parts/**](http://larseidnes.com/2014/11/05/angularjs-the-bad-parts/)

**Atualizando o AngularJS para o Angular. Acessoe em:** [**Angular - Upgrading from AngularJS to Angular**](https://v17.angular.io/guide/upgrade)**.**

**Grzegor. Angular vs AngularJS — Why you should migrate from AngularJS to Angular. Acessoe em:** [**Angular vs AngularJS — Why you should migrate from AngularJS to Angular - House of Angular**](https://houseofangular.io/angular-vs-angularjs-why-you-should-migrate-from-angularjs-to-angular/)