

# Relatório de NativeScript

**Bruno Moiteiro, Lucas Lara**

Curso de Bacharelado em Sistemas de Informação – Universidade Federal da Bahia  
(UFBA) – Campus de Ondina  
40170-120 – Salvador – BA– Brasil

bruno.moiteiro@gmail.com, lucas.casul@gmail.com

**Abstract.** *This paper describes features, functionality, and major use of the NativeScript tool. It explains the hybrid development process for iOS and Android, without the need of WEB interfaces, through Javascript, Angular 2 or TypeScript with real-time feedback in the emulator or the configured device.*

**Resumo.** *Esse artigo descreve características, funcionalidades e principais uso da ferramenta NativeScript. Explica o processo de desenvolvimento híbrido para iOS e Android, sem a necessidade de interfaces WEB, através de Javascript, Angular 2 ou TypeScript com feedbacks em tempo real no emulador ou no dispositivo configurado.*

## 1. Introdução

Com a necessidade de implementar soluções para os principais sistemas operacionais de dispositivos móveis, manter compatibilidade com versões anteriores dos próprios SOs e ter um desempenho razoável em todas as plataformas, a equipe desenvolvedora de softwares teria um processo cada vez mais longo de trabalho para criar novos aplicativos ou manter suas aplicações já existente. O objetivo do NativeScript é justamente remover essa responsabilidade do desenvolvedor e deixá-lo mais livre para codificação das novas ideias.

O NativeScript é um framework de código aberto escrito em NodeJs e funciona nas principais plataformas (Windows, Linux e Mac). Ele dá suporte para Angular 2 da Google, TypeScript e Javascript. Utiliza XML como linguagem de marcação e CSS para a estilização das telas. É possível também utilizar módulos do NodeJs com uma enorme variedades de pacotes e plugins criados pela comunidade através do gerenciador npm.

## 2. Características

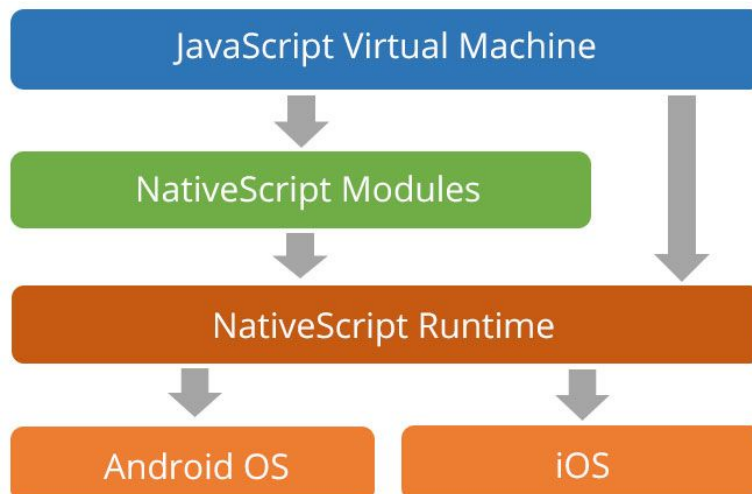
NativeScript é uma aplicação de execução em tempo real que utiliza Javascript Virtual Machine para criar aplicações nativas. Especificamente, utiliza V8 para Android e JavaScriptCore para iOS na execução dos aplicativos. Há planos para o suporte de aplicações Windows. Foi desenvolvido por Telerik e agora está sendo mantida pela Progress após a aquisição da Telerik pela Progress. Está sob a licença Apache 2.0 e o usuário pode fazer o que desejar sendo requerido apenas incluir os avisos necessários [VanToll 2015].

Dentre as principais características, o NativeScript fornece a associação em

duas vias dos objetos em Javascript com os componentes nativos de UI e a estilização desses componentes através do CSS. Outro mecanismo importante é que o desenvolvedor tem acesso direto às APIs nativas da plataforma [VanToll 2015].

## 2.1. Arquitetura

A arquitetura é dividida em três partes. A máquina virtual, onde o código em javascript é interpretado. Todo o código da aplicação é executado em tempo real e toda modificação feita pelo desenvolvedor pode ser visualizada imediatamente na tela do emulador ou dispositivo. A próxima camada é onde ficam os módulos do NativeScript. Quando o desenvolvedor escreve algum trecho de código, a máquina virtual busca nos módulos o que seria o correspondente para cada sistema operacional e cria os códigos adicionais necessários para o funcionamento e acesso de APIs para cada dispositivo. Esse processo é chamado de “bridge”. A terceira camada fica nos dispositivos. Cada aparelho possui um motor próprio para executar Javascript. No Android OS é usado V8 e no iOS, JavaScriptCore [Atanasov 2014].



**Figura 1.Arquitetura do NativeScript**

Quando o código em Javascript é interpretado pelo NativeScript Runtime, o código correspondente dessa execução é enviado para JVM e executado nativamente pelo dispositivo em Java, no caso do Android. O mesmo processo ocorre do JVM para o NativeScript Runtime. A resposta enviada pelo JVM é traduzida para JavaScript para que a aplicação possa interpretar [Atanasov 2014].

Esse processo pode ser visto analisando o trecho de código da figura 2.

```
var file = new java.io.File(path)
```

**Figura 2. Trecho de código em JavaScript**

- A máquina virtual do javascript avalia o código.
- O NativeScript determina qual módulo deverá ser associado.
- O serviço de Marshaling irá converter a string path em javascript para um objeto do tipo java.lang.String.
- O resultado da chamada da função java.io.File será um objeto customizado em

JavaScript que servirá como um proxy. Quando essa função for chamada, ela irá delegar ao que seria nativo do sistema operacional do dispositivo.

### **3. Ambiente de Desenvolvimento**

O NativeScript necessita apenas de NodeJs para o funcionamento básico. A própria instalação da ferramenta pode ser feita executando `npm install -g nativescript` e para a codificação o desenvolvedor pode utilizar qualquer editor de texto, preferencialmente que reconheça a linguagem javascript para dar suporte ao esquema de cores e autocompletar, pois não há a necessidade de uma IDE.

Por não haver uma IDE, não existe funcionalidade de drag-and-drop e nem pré-visualização no editor de texto. Todo o template deve ser criado utilizando marcações em XML definidas pelo NativeScript e a estilização é livre e feita em CSS. Por se tratar de um código que está sendo interpretado instantaneamente a cada modificação do código pelo usuário, os resultados podem ser vistos imediatamente na tela do emulador, recurso denominado de “livesync”. Isso reduz significativamente o tempo no que tange à buildings requeridos por IDEs.

Embora não haver uma IDE integrada, os criadores recomendam a utilização do editor de texto Visual Studio Code, pois foi desenvolvido plugins para o editor que facilitam o desenvolvimento. O NativeScript fornece outras funcionalidades externas que ajudam no desenvolvimento, a exemplo de NativeScript Image Builder. Site que auxilia desenvolvedores a gerar imagens em todas as resoluções e já estão prontas para serem inseridas no código.

#### **3.1. Emulador**

Os emuladores não são instalados por padrão utilizando o npm. Fica a cargo do desenvolvedor baixar os emuladores e fazer os devidos ajustes e/ou configurar dispositivos reais para os testes no momento do desenvolvimento. Entretanto os criadores do framework criaram ferramentas a parte que auxiliam nas configurações para Android e iOS. Ainda assim para executar a emulação do ambiente iOS é necessário estar na plataforma da Apple.

Quando o usuário estiver com todos os emuladores configurados, é possível trocar de dispositivo emulado utilizando um comando no terminal. Dessa forma, todas as modificações podem ser testadas em um tempo bastante hábil em ambas plataformas.

### **4. Suporte e Sensores**

Aplicações modernas estão utilizando cada vez mais sensores para melhorar a experiência do usuário nas tarefas do dia-a-dia, fornecendo mais eficiência, confiabilidade e segurança [Pelegrí-Llopart; 2017].

NativeScript é particularmente bem adaptado para este cenário. Funciona bem no iOS e no Android e fornece acesso a qualquer recurso da plataforma, incluindo novas bibliotecas criadas para acessar os sensores. NativeScript expõe todas as bibliotecas no nível JavaScript e podem ser encapsuladas em um plugin para uma abstração mais limpa e disponibilizada em repositórios. Isso significa que os desenvolvedores de JavaScript e CSS podem escrever suas aplicações sem a necessidade de conhecer os

meandros dos dispositivos apenas baixando esses códigos [Pelegrí-Llopart; 2017].

## **5. Utilização**

O NativeScript possui uma camada a mais em comparação a um app nativo mas o seu desempenho é relativamente semelhante. Como não há um browser emulando a aplicação tudo está bem próximo do que outras aplicações estariam. Desse modo é possível assegurar que qualquer aplicação menos robusta pode ser executada com o mínimo de disparidade no desempenho.

O site do NativeScript possui algumas aplicações de exemplos, como aplicativos com login, registro de informações etc. É também possível baixar o aplicativo “Examples NativeScript” na loja PlayStore do Google que demonstra diversos usos do framework. Aplicações simples como Reddit mobile, uma aplicação da própria Reddit que provê aos usuários uma forma mais rápida e leve de acessar seus conteúdos, poderia ser implementada com o NativeScript. A aplicação é um sistema de tópicos onde os usuários discutem os mais variados assuntos.

Provavelmente não seria indicado para soluções que necessitem muita segurança na informação do usuário. Não por motivos técnicos mas porque é necessário que o desenvolvedor sabia exatamente o que está acontecendo na aplicação para evitar brechas de segurança. Aplicações de bancos, por exemplo, não seria recomendável à utilização dessa ferramenta.

## Referências

VanToll, TJ. “How NativeScript Works”. Disponível:  
<http://developer.telerik.com/featured/nativescript-works>. Acesso: março/2017

Cross-Platform Native Apps Code Samples | NativeScript. Disponível:  
<https://www.nativescript.org/app-samples-with-code>. Acesso: março/2017

Atanasov, Georgi. “NativeScript - a Technical Overview”. Disponível:  
<http://developer.telerik.com/featured/nativescript-a-technical-overview>. Acesso:  
março/2017

Pelegri-Llopart, Eduardo. “NativeScript, Sensors and Modern Applications”  
Disponível:  
<http://www.telerik.com/blogs/nativescript-sensors-and-modern-applications>. Acesso:  
março/2017