



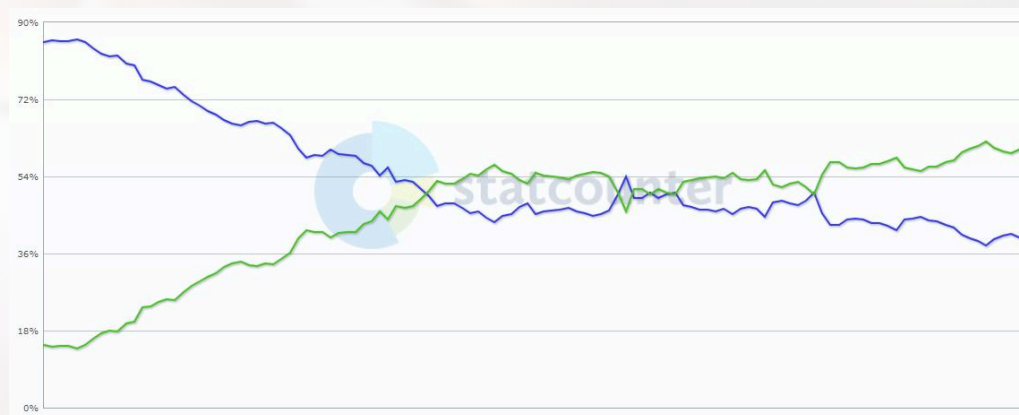
# **GUIA SOBRE TESTES AUTOMATIZADOS EM APLICATIVOS MOBILE**

Conheça essa prática e saiba porque você deve realizar

Faz anos que a **utilização de celulares é parte das nossas vidas, inclusive são mais utilizados que computadores** na atualidade. O comparativo dos últimos 10 anos é surpreendente. Em dezembro de 2012, somente 15% das pessoas usavam dispositivos móveis, contra 85% que utilizavam computadores. Hoje este gráfico é totalmente diferente, de 15% **passamos para quase 62% na utilização de celulares**, enquanto os desktops caíram para menos de 39%.

## EVOLUÇÃO DOS ACESSOS EM DISPOSITIVOS MOBILE:

	DEZ 2012	DEZ 2022
	15%	62% ▲
	85%	39% ▼



Fonte: <https://gs.statcounter.com/platform-market-share/desktop-mobile/worldwide/#monthly-201212-202212>

# O QUE VOCÊ ENCONTRARÁ NESTE E-BOOK:

01

Diferença entre  
automação  
**WEB e MOBILE**

Página 04

02

Aplicativos  
Nativos, Híbridos  
e PWA

Página 06

03

Ferramentas e  
linguagens para  
automação mobile

Página 08

04

Execução local  
e Device Farms

Página 12

05

Limitações

Página 14

Sobre o autor:



**Mateus Freitas**

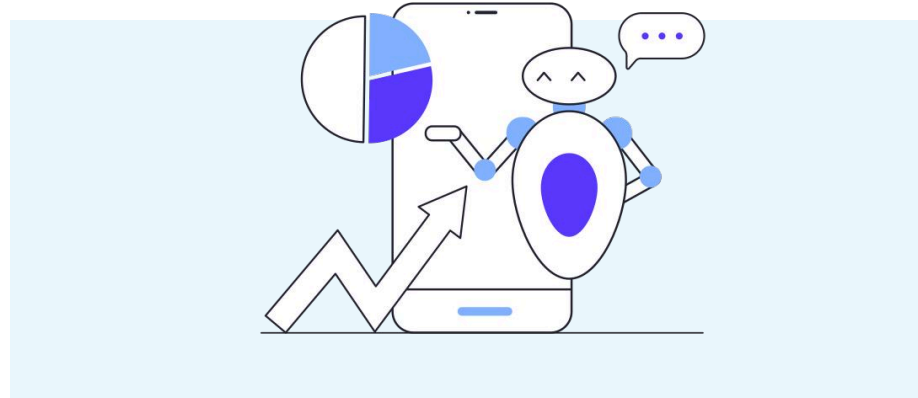
Engenheiro de teste de software





## 01

## Diferença entre automação WEB e MOBILE

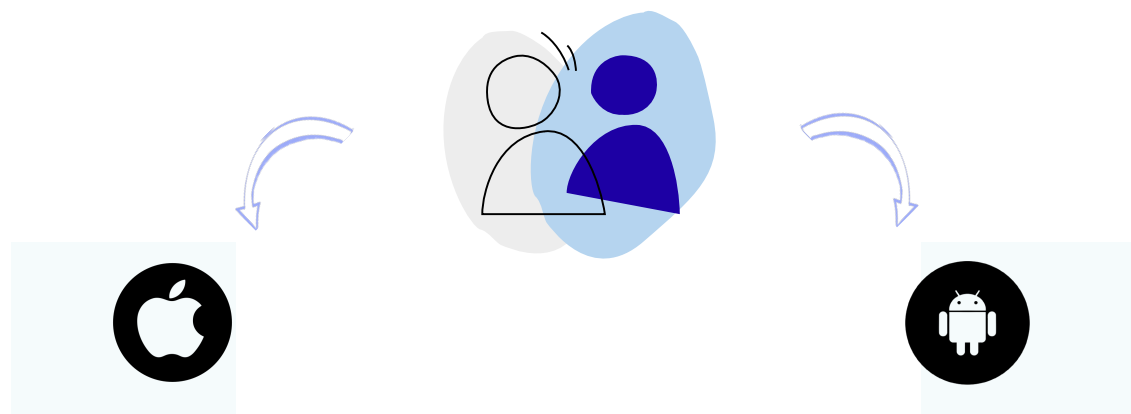


Como comentado anteriormente, **é possível realizar a automação dos testes para aplicativos mobiles**, e esta prática está em alta no mercado. Em alguns aspectos as automações WEB e MOBILE são bem parecidas, podemos realizar os mesmos testes (E2E, Carga, Regressão, etc.), e **algumas vezes usando até os mesmos frameworks e linguagens de programação** (Selenium, Java, Junit, etc.). Porém, existem diversas diferenças entre a execução da automação para aplicações mobile e WEB.

**Quando tratamos de sistemas WEB, estamos acessando o site através de um navegador**, como, por exemplo, o Chrome, Firefox ou Safari, **que está instalado em um computador** com um sistema operacional qualquer. O navegador ficará responsável por realizar a comunicação com o servidor que hospeda o website, e exibir o conteúdo para o usuário, independente do navegador instalado. Deste modo, **sempre teremos os mesmos elementos, indiferente do SO (Sistema Operacional)**, bem como sua execução.

Agora que entendemos a diferença entre uma aplicação WEB e um aplicativo MOBILE, também temos que entender que existem **diferenças entre os devices da população.**

Hoje existem 2 grandes sistemas operacionais quando falamos de dispositivos móveis: **Android e iOS**. Conforme o GS Stat Counter, **cerca de 72% dos celulares do mundo rodam Android**, contra quase **27% que rodam iOS**. Por serem dois sistemas diferentes, os aplicativos são construídos de maneiras diferentes também.



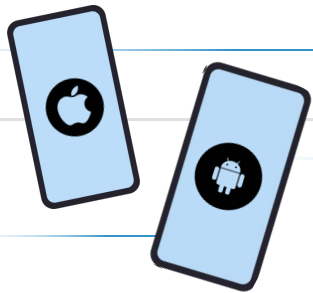
Quando pegamos para analisar os aplicativos de iOS, eles abrangem uma quantidade menor de dispositivos e versões do SO, principalmente por ser um sistema operacional de código fechado, onde é a Apple quem dita quais versões do SO existem e quais os devices que podem rodá-las.

Já o Android é mais abrangente, por ser um sistema open-source (de livre utilização) usado por várias empresas. Além disso, é comum realizarem uma personalização no sistema, dando a sua própria “marca” ao Android, como por exemplo, a Samsung, que utiliza o OneUI, a Xiaomi com o MIUI e muitas outras, o que também impactará em como o aplicativo irá se comportar.

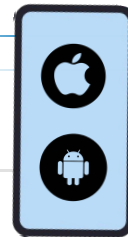
## 02

### Aplicativos Nativos, Híbridos e PWA

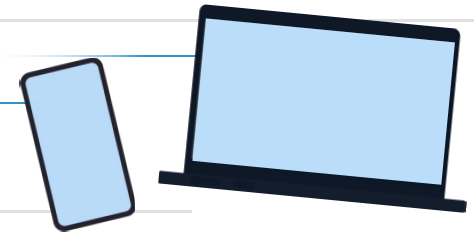
Devido a essas diferenças, os aplicativos mobiles são divididos em 3 categorias:  
**Nativos, Híbridos e PWA (Progressive Web Application).**



**Aplicativos Nativos** são desenvolvidos especialmente para um sistema operacional, com uma linguagem de programação específica. As vantagens deste desenvolvimento é que **os aplicativos ficam mais rápidos e performáticos**. Porém, caso queira que este aplicativo esteja disponível tanto para Android quanto para iOS, então será necessário manter dois códigos separados, tendo assim um trabalho dobrado para a equipe de desenvolvimento.



**Aplicações Híbridas** são uma mistura do desenvolvimento nativo e WEB. Com essa abordagem **é possível ter somente um código para as duas plataformas mobiles**, e usando as linguagens de programação para WEB, como HTML, CSS, JavaScript, TypeScript entre outras. Essa abordagem consegue unir o melhor dos dois mundos, e mesmo assim conseguindo criar uma aplicação performática.



Por último, temos as **Web-App, também conhecidas como PWA**. Nesta abordagem, diferente dos aplicativos híbridos e nativos, **o usuário não instala um aplicativo através das lojas do SO** (AppStore ou Play Store), ele acessa um link que direciona para uma aplicação WEB (website), através do navegador do celular. Apesar de parecer que estamos usando um aplicativo, é o navegador programado para exibir desta maneira responsiva.



# 03

## Ferramentas e linguagens para automação mobile



Assim como na automação WEB, existem diversos frameworks e linguagens de programação que podemos usar para escrever os scripts para a versão mobile. Algumas ferramentas são específicas para um sistema operacional, e outras funcionam em ambos. A seguir, citaremos **algumas ferramentas que estão em alta** no mercado, mas claro, existem muitas outras.

Uma das principais ferramentas do mercado é o **Appium**.

Com ele conseguimos realizar automação tanto para iOS quanto para Android, sendo possível rodar em devices físicos e emulados. Além disso, utiliza o WebDriver do Selenium, fazendo assim com que possua suporte para diversas linguagens de programação, como Java, JavaScript, Python, Ruby e muitas outras

O projeto do Appium **é open-source** e possui uma filosofia muito interessante, que foca em automatizar qualquer aplicativo mobile, de qualquer linguagem de programação, utilizando qualquer ferramenta de teste, tornando-o assim muito versátil e útil em qualquer ambiente.



Quando falamos de testes em dispositivos iOS, um dos principais nomes que aparecem é o **XCUITest**.

```
func checkText() {  
    let obj = XCUIApplication()  
    obj.otherElements.containing(.image, identifier:"Automacao").element.tap()  
    obj.buttons["Saiba mais"].tap()  
    XCTAssert(obj.staticTexts["Testes de iOS com XCUITest"].exists)  
}
```

Exemplo de código de automação do XCUITest



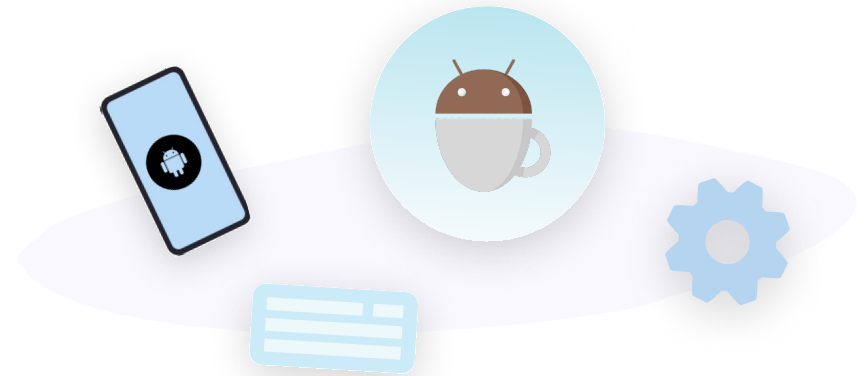
Esta é uma ferramenta integrada ao Xcode da Apple, e com ela é possível rodar testes de interface gráfica em aplicativos iOS. Por ser uma ferramenta mais integrada com o desenvolvimento, somente suporta Objective-C e Swift como linguagens de programação. Uma das vantagens desta ferramenta é que **o time de desenvolvimento e testes ficam mais próximos**, visto que usam a mesma linguagem para a automação e para o desenvolvimento, gerando então mais confiança para os próprios desenvolvedores criarem seus scripts de automação. Sem contar que, por ser uma ferramenta nativa, os testes são mais performáticos e confiáveis.

Semelhante ao iOS, o Android também tem uma ferramenta própria de testes, chamada Espresso.

A sua ideia é bem parecida com a do XCTest, ter uma ferramenta integrada com o desenvolvimento, que suporta somente as linguagens usadas para desenvolver os aplicativos (Java e Kotlin). Com essa abordagem, os testes automatizados de UI (interface do usuário) ficam mais próximos do time de desenvolvimento, e por ser uma **ferramenta mais fácil de aprender**, os próprios desenvolvedores podem se aventurar e criar suas automações.

```
@Test
fun greeterSaysHello() {
    onView(withId(R.id.name_field)).perform(typeText("Mateus"))
    onView(withId(R.id.greet_button)).perform(click())
    onView(withText("Olá Mateus!")).check(matches(isDisplayed()))
}
```

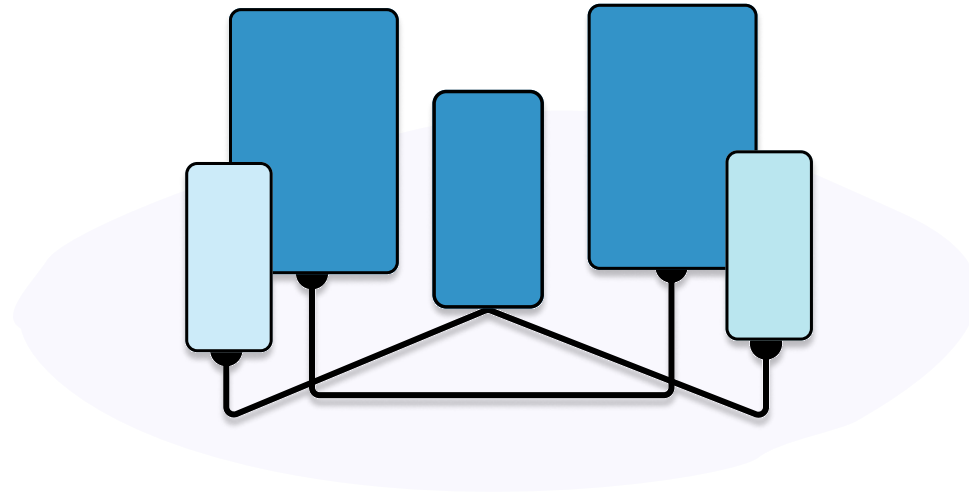
Exemplo de código de automação do Espresso





# 04

## Execução local e Device Farms



Até aqui conseguimos entender as diferenças de um aplicativo web e mobile, tipos de aplicações e ferramentas de automação, mas ainda não falamos **onde essa execução é realizada**.

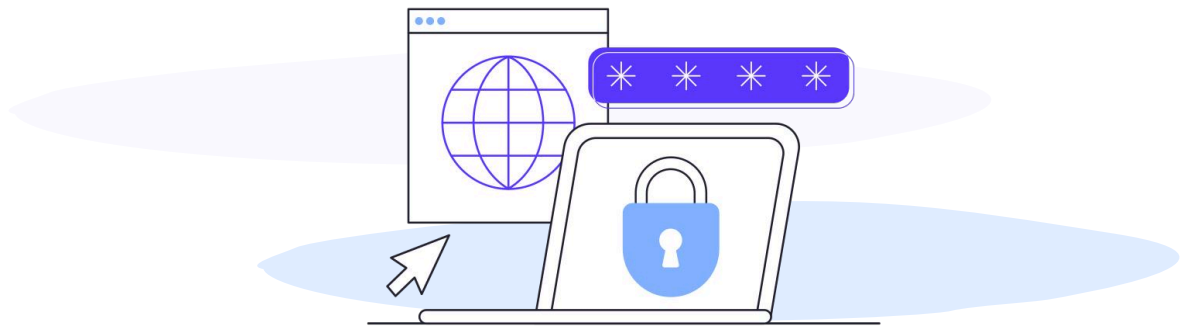
Como comentado anteriormente, existem diversas versões de dispositivos que rodam os sistemas operacionais mobiles. Quando falamos de iOS, temos aproximadamente 20 a 25 versões, porém, quando o assunto é Android, **essa quantidade é exorbitante**, visto que muitas empresas usam esse sistema operacional (Samsung, Xiaomi, Motorola, entre outras).

Quando testamos nossos aplicativos, temos que garantir que **funcionem em todas as versões atuais dos SO's**, mas fazer isso com devices físicos é inviável, levando os testadores a usarem emuladores, rodando com as versões desejadas de cada SO.

Entretanto, manter dezenas de emuladores atualizados e rodando é bastante trabalhoso, sem contar na quantidade de processamento requerido. Pensando nessas dificuldades, começaram a surgir as **Device Farms**, um sistema (normalmente pago), que disponibiliza diversas versões de celulares e sistemas operacionais sobre demanda, para que o time de desenvolvimento consiga testar suas aplicações de **forma rápida e fácil, sem se preocupar com as manutenções** dos devices e emuladores.

Dentre essas empresas voltadas às Device Farms, temos algumas que se destacam, como BrowserStack e AWS Device Farm. Seus serviços podem variar, mas de modo geral são vendidos na forma de pacote, no qual o usuário paga para acessar um ou diversos devices, e pode ser cobrado por tempo de execução dos dispositivos.

Com o aumento da quantidade de dispositivos móveis e da quantidade de aplicativos, um ponto que chama atenção é a **segurança**.



Por isso, as fabricantes disponibilizam algumas soluções nativas para que os desenvolvedores apliquem nos aplicativos, como biometria e FaceID. Além dessas, também possuímos os 2FA (autenticação de dois fatores) e captcha. **Nestes casos, a automação não é aplicada**, e caso o aplicativo testado possua uma dessas etapas, o correto é solicitar que sejam desativadas para que a automação prossiga, visto que o intuito dessas tecnologias é justamente não serem burladas.

05

Limitações



E a sua empresa já trabalha com testes automatizados? Nossos especialistas estão preparados para propor soluções que melhor se adequem as necessidades do cliente no que diz respeito a testes automatizados.

**Fale gratuitamente com um consultor clicando aqui!**



(51) 9 9233-8382



[atendimento@testingcompany.com.br](mailto:atendimento@testingcompany.com.br)



[Matriz | Pedro Adams Filho, 5857 - Centro, NH - RS](#)

