

Automação de Testes Funcionais em Aplicativos Android para Smart PoS: Um Relato de Experiência

Emmanuel Mendes Alves¹

Centro de Ciências Aplicadas e Educação – Departamento de Ciências Exatas (DCX)
Universidade Federal da Paraíba (UFPB) – Campus IV – Rio Tinto – PB – Brasil

(emmanuel.mendes)@dcx.ufpb.br

Abstract. *The present work focuses on describing an experience taken in a functional test automation process for Smart Point of Sales (PoS) devices that use the Android Platform. This work came from a demand in a company that develops software for these devices. In this process, searches were carried out for tools that allow this automation. In addition, we have built and analyzed new tests on these tools and adapted existing manual test cases that were already used in the company. In this work, after the analyzes, we have presented the main lessons learned.*

Resumo. *O presente trabalho tem como foco descrever a experiência obtida em um processo de automação de testes funcionais em dispositivo Smart PoS (Point of Sales) na plataforma Android. Este trabalho surgiu da demanda de uma empresa que desenvolve softwares para esses tipos de dispositivos. Nesse processo foram realizadas buscas de ferramentas que permitem essa automatização. Além disso, foram construídos e analisados testes nestas ferramentas e foram adaptados casos de testes manuais já existentes na empresa. Neste trabalho, depois das análises, foram apresentadas as principais lições aprendidas.*

1. Introdução

No processo de desenvolvimento de software, garantir a qualidade de um software é necessário para encontrar pontos que não atendam aos requisitos do sistema, de forma que possam ser corrigidos antes que este entre em produção. Conforme destaca Sommerville (2019), uma forma de procurar produzir software com qualidade é por meio de testes, os quais visam mostrar que um programa faz o que foi destinado a fazer e descobrir defeitos antes que ele seja colocado em uso. Tais testes podem ser manuais ou automatizados.

Este trabalho em questão tem como objetivo apresentar a experiência obtida no processo de automação de alguns testes em dispositivos Android para Smart PoS em uma empresa de tecnologia. A empresa em questão fornece sistemas de transação financeira e alguns destes executam em maquinas de cartão com o sistema Android que são chamadas de Smart PoS.

¹ Trabalho de conclusão de curso, sob orientação da professora Ayla Débora Dantas de Souza Rebouças e do professor Rodrigo Rebouças de Almeida submetido ao Curso de Licenciatura em Ciência da Computação do Centro de Ciências Aplicadas e Educação (CCAEC) da Universidade Federal da Paraíba, como parte dos requisitos necessários para obtenção do grau de LICENCIADO EM CIÊNCIA DA COMPUTAÇÃO.

No processo de automação, foi realizada a busca de ferramentas que permitissem a automação de testes nos aplicativos fornecidos. A ideia proposta era automatizar testes funcionais que são realizados de forma manual pela empresa e executados novamente a cada nova atualização do sistema. Estes testes são realizados na interface (tela do usuário) e por meio deles se busca validar se os requisitos funcionais estão sendo atendidos nas novas versões. Isso demanda muito tempo e esforço da equipe responsável pelos testes para certificação dos aplicativos na empresa.

Este pedido foi uma demanda da própria empresa, tendo em vista que o esforço para testar a interface do usuário é muito grande, tomando muito tempo da equipe de analistas de testes que tem de executar roteiros de teste a cada atualização do sistema fornecido pela empresa.

Foram encontradas diversas ferramentas que permitem automação de testes desse tipo, tendo sido selecionadas para análise as seguintes: Espresso, Ranorex Studio, e Test Project. Foram levados em consideração os seguintes aspectos nessa busca: o funcionamento da ferramenta em dispositivos Android, a viabilidade de uso da ferramenta para automatizar testes manuais existentes e para se integrar com o Smart PoS.

O processo de análise das ferramentas foi feito com colaboração de profissionais da empresa e utilizando um Smart PoS modelo N910, da marca NewLand, que foi fornecido pela empresa. Após análises iniciais, percebeu-se que apenas duas ferramentas funcionavam com o Smart PoS utilizado: o Espresso e o TestProject.

Em reuniões semanais com a equipe responsável pela automação de testes, foram apresentados os resultados encontrados e decidiu-se focar na ferramenta TestProject, considerando a natureza dos testes a serem automatizados. Neste trabalho será relatada esta experiência visando apoiar outros profissionais responsáveis por automação de testes em sistemas Android e serão reportadas as principais lições aprendidas no processo.

As demais seções deste artigo estão organizadas conforme descrito a seguir: a Seção 2 apresenta a fundamentação teórica; a Seção 3 apresenta a metodologia utilizada no trabalho; a Seção 4 apresenta os principais resultados das análises feitas; a Seção 5 apresenta as lições aprendidas; e, por fim, a Seção 6 apresenta as conclusões e propostas de trabalhos futuros.

2. Fundamentação Teórica e Trabalhos Relacionados

Serão apresentados nesta seção alguns conceitos e uma fundamentação teórica geral sobre testes, especialmente em terminais PoS e sobre automação de testes.

2.1 Testes de Software

Como destaca Pressman e Maxim (2021, p.372) o software é testado para revelar erros cometidos inadvertidamente quando ele foi projetado e construído. Em geral há vários tipos, níveis e abordagens de testes. Por exemplo, podemos ter testes do tipo caixa preta e caixa branca. Um teste caixa-preta examina alguns aspectos fundamentais de um sistema, com pouca preocupação em relação à estrutura lógica interna do software. Já o teste caixa branca é um teste onde se necessita ter acesso ao código sendo testado [Pressman e Maxim 2021].

Testes caixa-preta são também chamados de testes funcionais, conforme destaca Lamounier (2021). Sua estrutura é definida de acordo com os requisitos funcionais do software sem levar em conta detalhes de sua construção, como módulos, classes, código fonte.

Testes de caixa preta são, na maioria das vezes, utilizados para verificar a perspectiva do usuário, realizando a verificação e correção do sistema com base nos seus requisitos funcionais, que, na maioria das vezes, são capazes de resolver erros de interface, comportamentos, erros de iniciação ou término de um sistema, como, por exemplo, um sistema que aceita apenas números inteiros positivos em seus campos [Lamounier 2021].

Neste trabalho houve um foco em testes caixa preta, realizados na interface do usuário e focando apenas na saída que o sistema apresenta nos terminais onde são realizados os testes.

Para realizar um teste funcional na interface do sistema, normalmente, o analista de teste segue um roteiro de testes planejado de acordo com os requisitos do sistema e considerando as necessidades do cliente final. Esses roteiros são constituídos por casos de teste que exercitam cenários de uso do sistema por usuários finais e verificam se os requisitos funcionais estão sendo atendidos.

Ao criar um caso de teste, o analista de testes deve exercitar situações que possam expor possíveis defeitos e verificar se o sistema está atendendo todos os requisitos funcionais necessários. Outro ponto importante sobre o caso de teste, é que ele segue uma lógica estrutural em seu desenvolvimento. A estruturação de um caso de teste se divide em uma especificação das entradas, resultados previstos e condições de execução para um item a testar [De Pádua Filho 2019]. Portanto, quando o analista cria um roteiro de testes, ele leva em consideração essa estrutura e segue esse processo na criação e desenvolvimento dos casos de teste.

Pode-se também dizer que este trabalho está relacionado ao contexto de testes de aceitação, que são testes que visam determinar se um sistema satisfaz seus critérios de aceitação, checando o comportamento do sistema com base nos requisitos dos clientes [IEEE 2014]. Também se pode considerar relação com a metodologia de testes chamada de *End-to-End Testing*, que segundo o *Software Testing Help* (2022) visa testar o fluxo de uma aplicação do início ao fim e tem como propósito simular um cenário real do usuário para validar o sistema sob teste, e seus componentes em relação à sua integração e a integridade dos dados.

2.2 Testes em Terminais Smart PoS

O PoS, conforme destaca Santos e Mello (2019), é um dispositivo que permite por meio de uma interface interativa realizar transações financeiras. O PoS (*Point of Sales*) que significa ponto de venda, permite que comerciantes realizem transações financeiras por meio de cartões. Esse dispositivo fica junto ao comerciante e é a interface responsável pela comunicação do cliente com os diversos processos responsáveis por realizar esse tipo de transação. Um exemplo desse processo está sendo ilustrado na Figura 1.

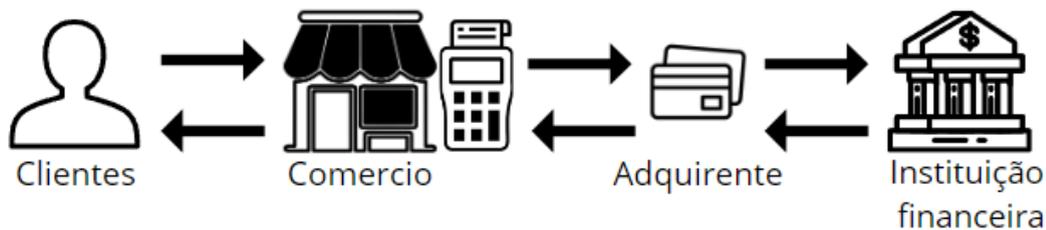


Figura 1: Processo ponta a ponta da transação financeira eletrônica
Fonte: O autor, adaptado de Rolim e Rebouças (2021)

O processo decorre de forma que em uma ponta está o cliente e o estabelecimento, sendo o estabelecimento o portador do Smart PoS. O cliente tem um cartão, sendo esse cartão homologado com uma bandeira, que pode ser, por exemplo, Visa, MasterCard, Elo, etc. O comerciante informa um valor para o equipamento, o cliente passa o cartão, o terminal pega essas informações, envia para a adquirente e conversa com o banco do cliente e o do comerciante, para que o valor seja retirado de sua conta e adicionado na conta do banco do comerciante.

Portanto, para que essas transações sejam realizadas de forma segura, esses dispositivos, que podem ter um modelo mais convencional ou podem ser do tipo Smart PoS, precisam estar homologados por algum órgão de segurança da informação, tendo em vista que os mesmos lidam com dados sensíveis dos usuários e uma falha de processamento poderia levar a transtornos de vazamento de informações do usuário.

Como mostra Santos e Mello (2019)

O POS é um dispositivo embarcado para processamento de pagamentos com cartão do tipo chip-and-PIN. Trata-se de um tipo de pagamento especificado pelas normas EMV (EMVCo, 2018), que são normas de pagamento criadas pela Europay®, MasterCard® e Visa®. Numa transação chip-and-PIN, a autenticação é feita por meio de um identificador de 4 ou 6 dígitos (PIN - Personal Identification Number) e é necessária uma troca de mensagens entre o dispositivo POS e o chip do cartão de crédito. A norma ISO/IEC 7816-3 (ISO, 2006) define o protocolo de comunicação entre chip e leitor de cartão, sendo as mensagens trocadas por comandos chamados de APDU (*Application Protocol Data Units*).

No caso do Smart PoS utilizado neste trabalho, o sistema que nele executa é o Android. Assim como o PoS convencional, o Smart PoS também segue normas de segurança a fim de validar e dar segurança no processamento de informações no momento da transação. Nesse caso, a interação é totalmente pela tela do dispositivo onde o usuário está com o equipamento e realiza tudo por meio de toque na tela (*touch screen*), conforme ilustrado pela Figura 2. Ele se diferencia do PoS convencional onde normalmente a interação é por meio de botões.



Figura 2: Imagem ilustrativa de um Smart PoS da marca NewLand, imagem do site:

<https://www.globasure.net/what-you-should-know-about-servpoint-newland-n910-pos-terminal/>

Neste caso, os testes realizados no Smart PoS também se diferenciam dos testes em PoS convencional, tendo em vista que diversos casos de testes são relacionados na interface do dispositivo. Também se deve destacar que como a tecnologia de um equipamento é diferente da do outro, os sistemas produzidos para eles também se diferem, fazendo com que muitos casos de teste não sejam reutilizados nos dois tipos de equipamentos.

No processo de criação de casos de testes para terminais do tipo Smart PoS, o processo de criação dos casos segue os requisitos funcionais do software. Existe um documento com esses requisitos solicitados e a execução desses casos teste é de forma manual com interação do analista da equipe no equipamento em que o sistema está instalado e configurado, como o Smart PoS da NewLand N910.

No Smart PoS, a grande maioria dos testes na empresa em que se insere este trabalho são realizados neste tipo de equipamento de forma manual. Esse processo feito por analistas de testes da empresa responsáveis pela certificação é realizado conforme descrito a seguir. Quando sai uma nova versão de atualização do sistema do equipamento, a equipe de teste valida se essa nova versão atende os requisitos necessários. Caso não atenda, essa não conformidade é relatada para a equipe de desenvolvimento, destacando o que foi observado para que possam ser feitas as correções necessárias.

Além dos testes na interface, nesse tipo de equipamento são realizados alguns testes de informações que transitam no processo dessas transações, que também são testadas e verificadas pelos analistas de testes. Esse tipo de teste em particular é feito a fim de validar se as informações estão corretas, tanto as que são enviadas pelo terminal para o servidor quanto as que são enviadas pelo servidor que está se comunicando com o terminal.

Na empresa onde foi realizado este trabalho, esse processo é realizado da seguinte forma, como mostra, Rolim e Rebouças (2021)

Após realizar o processo do pagamento, o PoS gera arquivos que vão conter bits responsáveis por informar os dados da transação que são enviados da adquirente para o terminal. Faz-se necessário realizar testes de bits para verificar se os dados da transação, como número do cartão, data, hora e valor, entre outras informações, estão de acordo com a transação realizada [...]

2.3 Automação de Testes Funcionais

O processo de automação de um teste é a transformação de um teste antes praticado de forma manual, por quem está testando, para um teste feito pelo computador. Desta forma, o teste automatizado é um teste onde a entrada está programada e, ao ser executado, o sistema sob teste é executado e com base nas respostas obtidas são mostrados os seus resultados (falha ou sucesso) [De Pádua Filho 2019].

Portanto, automatizar um teste contribui na melhoria do processo de desenvolvimento de software. Na empresa em questão, existe uma equipe dedicada a realizar testes de forma manual, chamada de equipe de certificação, que tem o propósito de validar os requisitos dos sistemas. Essa equipe tem um prazo para realizar os testes necessários e a automação poderia apoiá-la bastante.

Como demonstra Gonçalves, Barreto e Zenker (2019, p.187)

Automatizar testes é uma forma de exercitar e verificar o software desenvolvido sem que haja participação humana. A integração das ferramentas de automatização de testes nos processos de desenvolvimento de software contribui para que o ciclo de vida do projeto seja reduzido, assim como o seu custo [...]

Dito isso, automatizar esses testes realizados de forma manual para que sejam realizados de forma automatizada contribui para que a certificação do software consiga ser feita num prazo mais curto, sem a necessidade de tantas pessoas para realizar os testes e evitando erros no processo de verificação. No caso da empresa, existem alguns casos de testes realizados que demandam bastante tempo da equipe de analistas e que podem ser automatizados.

Para isso, torna-se necessária a busca por ferramentas que possibilitem automatizar testes realizados na interface do usuário e que funcionem em dispositivos Android, em especial Smart PoS.

Dessa forma, foram realizadas buscas por algumas plataformas que permitissem atender essa demanda da empresa. Foram levantadas algumas como o Espresso (<https://developer.android.com/training/testing/espresso>), Ranorex Studio (<https://www.ranorex.com/>) e TestProject (<https://testproject.io/>). O Espresso, que é uma ferramenta da Google, foi apresentado pela empresa para o autor, já o Ranorex Studio e o TestProject foram apresentadas no decorrer das buscas realizadas por ferramentas.

3. Metodologia

Inicialmente, a empresa apresentou uma demanda de automação dos testes realizados na interface do usuário com o uso de alguma ferramenta que permitisse realizar esse tipo de teste em Android. Foi então dado início ao processo de busca de ferramentas para apoiar a automação. Com a busca por materiais publicados em artigos na internet, foram encontradas algumas opções para analisar.

Após essa busca realizada, foram feitas algumas apresentações para os líderes da equipe de certificação e o gerente de desenvolvimento, onde foram debatidas algumas dessas ferramentas encontradas e discutidos alguns pontos positivos e negativos. Após essas apresentações, foram debatidos os próximos passos a serem seguidos na busca e análise dessas ferramentas.

Em paralelo às buscas dessas ferramentas, foram analisados alguns casos de teste já criados pela equipe de analistas de testes da empresa e que eram executados de forma manual. Essa análise foi feita a fim de verificar a possibilidade de automação dos casos de teste, tendo em vista que alguns testes demandam interações que necessitavam de seres humanos, como a verificação sobre se o brilho ou cor da tela estavam agradáveis à vista.

Após essa análise, foi dado início ao processo de automatização de casos de teste nas ferramentas escolhidas. Para isso, tomou-se como base os casos de teste manuais e foram criados os casos de teste automatizados observando os requisitos funcionais dos sistemas sendo testados e que são observados na certificação desses softwares.

Como forma de avaliar as ferramentas escolhidas, era feita uma comparação de alguns aspectos de cada uma e das demandas de teste da empresa. Alguns desses aspectos estão relacionados ao funcionamento em dispositivos Android, opção de reuso dos testes criados e também a algumas funcionalidades como relatórios, *Record and Replay*, plugins e se é possível seu funcionamento em Smart PoS.

No caso das três ferramentas levadas em consideração nesses momentos, algumas atendiam a quase todos os requisitos necessários, como por exemplo o Espresso Google e o TestProject. Já as outras não atenderam um dos principais requisitos, que foi o funcionamento em dispositivos do tipo Smart PoS, que foi o caso do Ranorex Studio. Dessa forma, foram realizados estudos e testes nessas ferramentas a fim de levantar esses pontos e realizar comparações entre as mesmas. No Quadro 1 serão apresentados alguns pontos levantados para escolha da ferramenta ideal:

Pontos Avaliados
Funciona em Android
Tem funções, como Record and Replay, relatórios, etc.
Permite o uso em Smart PoS
Permite o compartilhamento do projeto com diversas pessoas
Permite automatizar testes funcionais

Quadro 1: pontos a serem levados em consideração na análise das ferramentas

Fonte: O autor

Após análises e reuniões, decidiu-se em conjunto que a melhor ferramenta a ser analisada de forma mais intensiva era o TestProject, e deu-se então continuidade aos estudos nesse sistema. Considerando os pontos do Quadro 1, a única ferramenta que atendeu a todos foi o TestProject. Já no caso das demais ferramentas, não foram escolhidas pois não atendiam a todos os requisitos, como por exemplo, o Espresso, que dependia do acesso ao código fonte do aplicativo.

Em paralelo às buscas e análises das ferramentas, foi dado início ao processo de documentação das experiências obtidas no decorrer do processo de automatização de testes junto à empresa. Além disso, foram criadas apresentações para servir na escrita

deste texto, como forma de apresentar essas experiências tidas neste processo e para apresentações na empresa em questão visando validar as análises feitas.

4. Resultados

Esta seção visa apresentar os resultados encontrados nas buscas e análises das ferramentas de automatização de testes. Esses resultados estão distribuídos em alguns tópicos, sendo o primeiro uma análise dos casos de testes manuais que já haviam sido criados na empresa. Além disso, serão apresentados alguns resultados das análises das ferramentas Espresso, Ranorex e TestProject e alguns detalhes sobre estas ferramentas.

4.1 Análise dos casos de testes manuais para terminais Smart PoS

Por se tratar de uma empresa no ramo de transação financeira eletrônica, os seus sistemas têm relação direta com os processos necessários para realizar essas transações. Dessa forma, os casos de teste criados testam se os requisitos funcionais deste processo estão sendo atendidos. Além disso, esses testes foram criados e são realizados atualmente de forma manual.

Durante a análise, foram verificadas algumas particularidades que limitam a automatização. Um exemplo é que em alguns testes é necessário o uso de cartão de crédito ou débito que após ser inserido no dispositivo demanda uma resposta a ser verificada. Nesses casos, é preciso que o analista insira algum cartão desses para que a análise seja feita. Isso limita os testes que podem ser automatizados já que só foi possível automatizar os que não demandam essa inserção.

Foi conversado com a equipe se seria possível via programação direta nos aplicativos mudar isso. Isso seria uma tentativa de fazer com que fosse possível driblar esse problema. Porém, foi retornado que não seria possível mudar nos testes o modo de entrada no Smart PoS em que se demandava que fosse inserido um cartão.

Mesmo assim, foi possível adaptar alguns dos testes, como por exemplo, alguns testes de funcionalidades do sistema e testes de transação nos quais se digitava dados do cartão no modo de digitação manual. Alguns destes testes verificavam a presença de elementos esperados em determinadas telas ou eram testes de impressão de comprovantes, dentre outros. Esses testes foram debatidos por meio de reunião da equipe de automação de testes da empresa, tendo em vista que a experiência dos que estavam presentes há mais tempo na empresa contribuiria nesse processo.

A seguir, no Quadro 2, é apresentado um exemplo de um caso de teste manual que pôde ser automatizado. Seu código não será exposto por questões de sigilo da empresa.

Caso de teste	Passos	Esperado
Realizar uma transação com número de cartão inexistente	- Iniciar uma transação - Inserir um número de cartão errado	A transação não deve ser aprovada e deve ser apresentada uma mensagem de erro “Cartão inválido”

Tabela 2: Exemplo de caso de teste automatizado

Fonte: O autor

4.1.1 Análise da Ferramenta Espresso Google

O Espresso é uma API oferecida pelo Google para uso em testes do tipo caixa preta. É uma ferramenta gratuita e seu uso pode ser feito diretamente através da IDE Android Studio, por meio da qual se pode usar a API. Para seu uso, é necessário ter acesso direto ao código fonte da aplicação. Há de destacar que mesmo com a necessidade de acesso ao código fonte da aplicação, para desenvolver os scripts dos testes, não é necessário verificar o código fonte. Com uso do recurso *Record and Replay* a ferramenta interpreta os elementos e cria os testes automaticamente.

A plataforma permite seu uso por meio de código de programação nas linguagens Java ou Kotlin. Além disso, é possível o uso do recurso de *Record and Replay*, sendo essa função disponibilizada na própria IDE Android Studio. Como simulação do dispositivo, a plataforma utiliza o espelhamento da tela do dispositivo, no caso o Smart PoS. A interação por meio do *Record and Replay* é feita por esse espelhamento da tela do dispositivo. Já a escrita dos códigos é feita por meio dos códigos de programação Java ou Kotlin.

Os testes nesta plataforma foram realizados em um aplicativo desenvolvido apenas para fins de análise e que realiza cálculo de escolhas de combustível (etanol ou gasolina), baseado em um cálculo matemático. Esse aplicativo é bem simples e apresenta somente uma tela. Seu código fonte está disponível em [<https://github.com/EmmanuelMendes/AplicativoEscolhaCertaCombustiveis>]. A interação possível do usuário é extremamente limitada. Nas reuniões realizadas com outras pessoas da equipe de certificação e em a equipe de desenvolvimento de software da empresa, foi relatado que foram realizados testes no código fonte utilizando essa ferramenta na aplicação a qual o autor deste trabalho não tinha acesso.

Dessa forma, com esses testes realizados utilizando a ferramenta, foi relatado que a ferramenta não apresentou o funcionamento esperado ao utilizar a API, por esse motivo a mesma foi descartada. Não se sabe ao certo o motivo do seu mau funcionamento nos aplicativos da empresa, existe um pensamento de que isso tem relação com a versão do sistema operacional do Smart PoS ser antiga. Porém, nos testes realizados no aplicativo acima citado, instalado no dispositivo, o mesmo apresentou o funcionamento esperado.

4.1.2 Análise da Ferramenta Ranorex

O Ranorex é uma ferramenta que permite automatizar testes, sendo um desses testes o teste funcional de interface do usuário. A plataforma em questão, que tem um custo para seu uso, permite a integração com diversas outras ferramentas, permitindo o uso de diversas outras funções no processo de testes. A plataforma funciona localmente no computador, tendo que ser feito download do Ranorex Studio e instalado localmente na máquina do analista, não permitindo um compartilhamento instantâneo com outras pessoas.

Alguns pontos positivos da plataforma são que ela permite realizar uma série de processos, como criação de variáveis, para armazenamento de alguma informação no momento de testes. Além disso, a plataforma permite integração com o Jira e Git,

permite o uso de base de dados em SQL, ou CSV e Excel. Outro ponto importante é que a plataforma permite que ao final seja possível ter acesso a um relatório demonstrativo dos testes realizados.

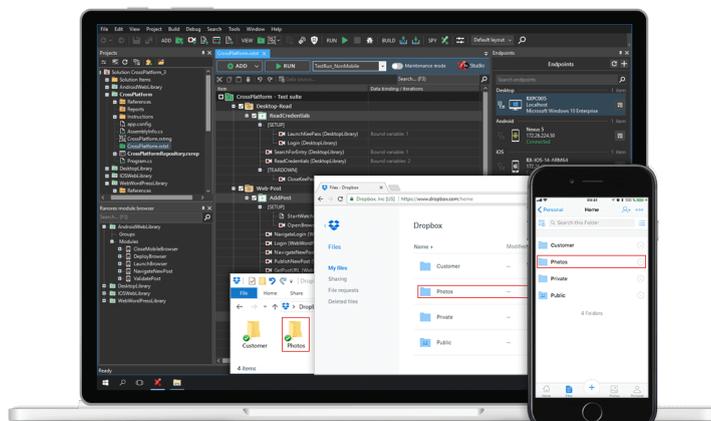


Figura 3: Exemplo da ferramenta Ranorex
Imagem do Site:
<https://www.ranorex.com/test-automation-framework/>

Os pontos negativos dos contratos na plataforma estão relacionados basicamente ao seu bloqueio no uso em Smart PoS, fazendo com que não fosse possível realizar os testes na plataforma, de fato. Além disso, foi tentado realizar testes em um aplicativo desenvolvido neste trabalho para as análises (aplicativo de análise de opções de abastecimento de combustíveis). Esse aplicativo foi instalado em celular e mesmo assim a plataforma não permitiu que fossem realizados testes, sendo esse aplicativo o de escolha de combustíveis, o mesmo que foi utilizado nas demais ferramentas.

Alguns pontos a serem destacados nesse processo serão discutidos a seguir. Primeiro, o Ranorex, para seu funcionamento em dispositivos móveis, precisa de um aplicativo da própria plataforma, responsável por realizar o processo de captura do aplicativo no dispositivo e envio para a ferramenta instalada no computador. Foi realizada uma série de tentativas para que fosse possível executar esse aplicativo no Smart PoS, mas esse dispositivo tem alguns bloqueios, por conta da segurança. Além disso, pelo fato da versão do Android ser uma versão antiga, não permitiu que esse aplicativo da plataforma executasse, mesmo com sua instalação, o que impossibilitou os testes no dispositivo na ferramenta em questão.

4.1.3 Análise da Ferramenta TestProject

O TestProject é uma ferramenta de automação de testes funcionais de interface do usuário, sendo esta online, diferentemente do Espresso e do Ranorex Studio que demandam instalação e configuração local da máquina onde o teste é executado. O fato de ser online possibilita o compartilhamento com diversos outros analistas de testes. Além disso, esses outros analistas podem criar outros testes ou realizar manutenções nos já criados.

Para oferecer esse serviço, o TestProject utiliza várias outras ferramentas e tecnologias, como o Selenium (<https://testproject.io/web-test-recorder/>) para testes web e o Appium (<https://testproject.io/mobile-test-recorder/>), para testes em dispositivos

móveis, com sistema operacional Android. No caso em questão, para desenvolver testes em Smart PoS que têm o sistema Android embarcado no dispositivo, foi utilizada a tecnologia do Appium. A criação e execução dos testes foi feita diretamente no TestProject, deixando a cargo da ferramenta o controle de outras tecnologias por ela utilizadas.

Alguns pontos positivos do TestProject são que ele possibilita a integração com diversas outras ferramentas, como o Jira, Git, Jenkins, etc. Além disso, é possível criar uma variedade de testes. A ferramenta permite a criação de casos de testes em dispositivos Android e testes de interfaces Web. A Figura 4 ilustra a interface principal da ferramenta.

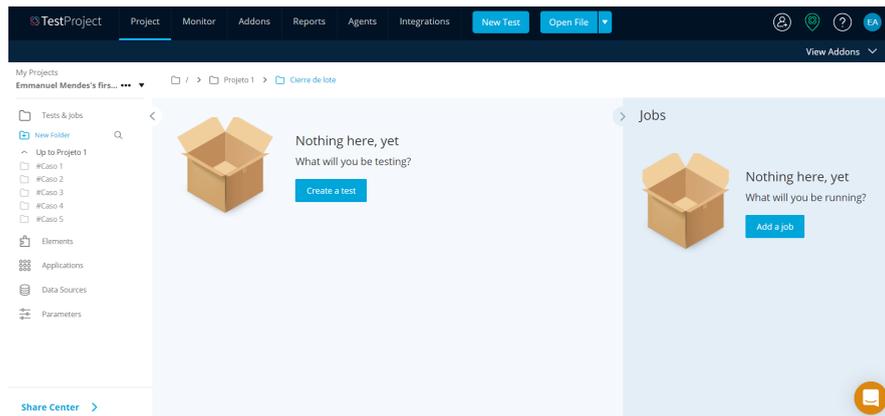


Figura 4: Tela principal TestProject
Fonte: Autor

Um dos pontos positivos sobre o TestProject é que permite o compartilhamento de projetos com diversos integrantes que podem interagir na plataforma simultaneamente. Outro ponto positivo é que apresenta funcionamento em Smart PoS, modelo NewLand N910, com sistema operacional Android 6.0. Além disso, é possível a criação e execução de uma série de testes na interface. Outro ponto positivo é que a ferramenta também possibilita realizar testes automatizados em páginas WEB, sendo esta uma demanda futura da empresa. Na imagem abaixo é possível ver as diferentes plataformas de testes que a ferramenta permite: a Mobile, a Web e a Code. Na função code é possível criar funções de testes, em algumas linguagens de programação e com isso importa na ferramenta, para uso nos testes na ferramenta.

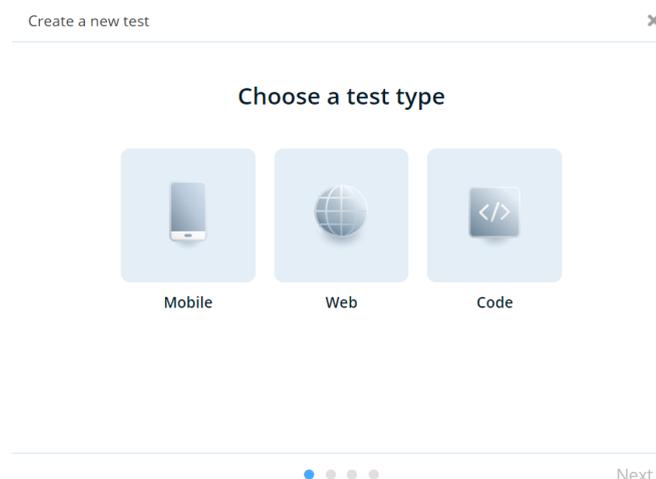


Figura 5: Tela de escolha do plataforma de teste
Fonte: Autor

Por fim, na ferramenta é possível a execução de vários testes em série ou paralelo, sendo essa função chamada de *Jobs* (Trabalhos). Essa função permite a execução de diversos testes e ao final é possível ter acesso a um relatório descritivo de todos os testes e os testes que passaram e que falharam durante a execução. É possível criar *Jobs* para testes Android, IOS ou Web.

Inicialmente, foi realizada uma configuração da ferramenta, para que a mesma funcione no dispositivo em questão. Para isso, foi parametrizado o aplicativo a ser testado, ou seja, foi definido um aplicativo padrão sendo esse que executa a transações que seria executado nos testes em questão. Esse aplicativo em questão, era o que seria testado na empresa. Com isso, a cada nova versão seria possível realizar os mesmos testes. Por esse motivo, foi realizado esse procedimento. O exemplo do código em formato Appium está disponível no link do git (https://github.com/EmmanuelMendes/Projeto_Automacao_testProject).

Após isso, foi realizada uma série de outras configurações de organização dos testes em questão. Como foram automatizados cerca de trinta casos de testes, foi necessário organizar esses testes. Para isso, a ferramenta conta com a criação de pastas de organização dos casos de testes criados. Abaixo, segue um exemplo de como ficaram organizadas as pastas, como demonstra a Figura 6.

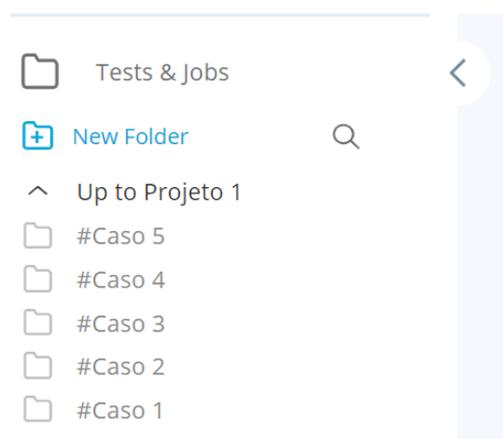


Figura 6: Estrutura de organização das pastas
Fonte: Autor

Dentro dessas pastas eram criados os testes referentes ao caso de teste em questão. Por questões de sigilo comercial da empresa, existem informações sensíveis dos sistema testados, e por esse motivo será apresentada a ferramenta tomando como base um aplicativo desenvolvido pelo autor, onde foram realizados testes nesse aplicativo.

Na imagem da Figura 7 vemos os passos desse teste. Primeiro se realiza o clique na caixa de texto e é inserido um valor (1.50). Depois, é repetido o processo em outra caixa de texto com o valor 5.50. Após inserir os valores, clica-se no botão calcular e é criada uma validação para ver se o resultado esperado, que é o texto dizendo que a melhor escolha é álcool, nesse caso, foi o obtido.



Figura 7: Sequência de códigos para realizar testes
Fonte: Autor

Na parte direita da tela, na tela com cor cinza e roxa está o aplicativo sendo executado no Smart PoS. À esquerda da tela está a tela de edição e gravação dos testes criados. Essa criação pode ser feita de forma manual, clicando em cada elemento da tela e adicionando o teste manualmente ou de forma automática, utilizando o *Record*, em que a cada clique na tela é gravada essa interação com os elementos da tela do aplicativo, como demonstra a figura 8.

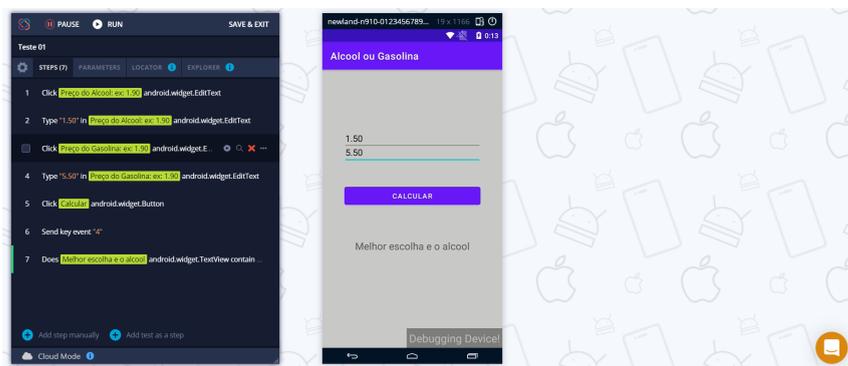


Figura 8: Tela de criação de testes no aplicativo
Fonte: Autor

Com as buscas e análises na ferramenta e com os testes realizados nos aplicativos da empresa em questão e em aplicativos criados pelo autor, foi possível criar testes automatizados, que além de executar testes e validar os resultados esperados com a mesma ferramenta, permitiram o acesso a relatórios que demonstraram resultados esperados desses testes.

Desta forma, a ferramenta em questão cobre o que era esperado, possibilitando a criação de uma série de testes automatizados. Foi possível também validar esses testes de diferentes formas, como por exemplo, validar cores esperadas em telas específicas que o usuário tem acesso ou até mesmo criar condições, onde em alguns cenários em que não se tem o resultado esperado é possível tomar outros caminhos, a fim de chegar de fato a esse resultado.

Há de se destacar que com a ferramenta em questão foi possível automatizar uma série de testes, que antes eram realizados de forma manual. Com as buscas e análises na ferramenta, chegou-se à conclusão que a ferramenta atendia ao que era esperado. Como consequência, após discussão, a empresa decidiu iniciar o processo de automatização de alguns testes que demandam muito tempo da equipe para a ferramenta em questão.

4.5 Análise Comparativa do Espresso x Ranorex x TestProject

No processo de buscas e análises relatados no decorrer do texto, foram levantados alguns pontos importantes, sobre essas três ferramentas. Para a escolha, como dito na metodologia, concentrou-se em alguns pontos a serem levantados em diversas ferramentas, a fim de chegar a um consenso da que melhor se encaixava na demanda da empresa.

Portanto, há de se destacar que nesse processo de análise das ferramentas em questão existem alguns pontos essenciais e outros que poderiam ser relevados. Portanto, os principais pontos para a escolha seriam, ferramentas que possibilitassem executar testes em dispositivos do tipo Smart PoS, como o N910, com sistema operacional Android, que não tivessem a necessidade de acesso ao código fonte da aplicação e que fosse permitido o compartilhamento do projeto com diversos outros integrantes da equipe.

Dessa forma, a primeira ferramenta analisada foi o Espresso, que é um framework do Google, que permite realizar a criação de testes por meio de escrita de códigos de programação ou utilizando o método do *Record and Replay*. Porém, para uso da ferramenta é necessário ter acesso ao código fonte da aplicação em questão para realizar os testes. Em reuniões com a equipe responsável pela automação na empresa, foi destacado que isso não seria possível, impossibilitando o uso da ferramenta.

Além disso, nos testes no Ranorex é necessário um aplicativo instalado no dispositivo, para que a ferramenta no computador consiga se comunicar e com isso ser possível criar e executar os testes de interface. Contudo, por conta de bloqueio de segurança e a versão do Android no N910 ser antiga, esse aplicativo não executou e não permitiu que fossem realizados os testes necessários na ferramenta. Tentou-se realizar testes no celular pessoal, mas mesmo assim a ferramenta não apresentou pleno funcionamento, impossibilitando dar continuidade aos testes em questão.

Por último, foi analisada a ferramenta TestProject. Em todo o processo de buscas, como destacado nos tópicos anteriores, essa foi a ferramenta que atendeu a todos os requisitos iniciais nas buscas realizadas. Primeiro, ela permite realizar testes em dispositivos do tipo Smart PoS, modelo N910 com sistema operacional Android, além disso é possível compartilhar projetos criados com diversas pessoas da equipe de automação na empresa, possibilitando que os mesmos criem novos testes, ajustem ou executem os testes já criados. Além disso, é possível ter acesso a relatórios de testes executados, sabendo os testes que foram validados ou não

A seguir, será apresentado um quadro com os pontos iniciais levantados e os resultados obtidos dessas análises. Nesse quadro estão alguns pontos principais como, funciona em Android, permite o uso em Smart PoS, permite automatizar testes de interface do usuário e por último necessita ter acesso ao código fonte da aplicação. Os pontos marcados com “X” na coluna da ferramenta indicam que esse ponto é atendido.

Pontos	TestProject	Ranorex	Espresso
Funciona em Android	X	X	X
Tem funções, como Record and Replay, relatórios, etc.	X	X	X
Permite o uso em Smart PoS	X		X
Permite o compartilhamento do projeto com diversas pessoas	X		
Permite automatizar testes de interface do usuário	X	X	X
Necessita ter acesso ao código fonte da aplicação			X

Quadro 2: Pontos encontrado na análise das plataformas
Fonte: O autor

5. Lições Aprendidas

O trabalho em questão tem como foco relatar algumas experiências tida pelo autor no processo de automatização de testes de interface do usuário em dispositivos Android. Além disso, essa experiência foi em parceria com uma empresa de tecnologia que desenvolveu um sistema que permite realizar transações financeiras eletrônicas, por meio de PoS convencionais e Smart PoS.

Há de se destacar alguns pontos importantes, como a necessidade e interesses de empresas de tecnologia na busca por ferramentas que atestem a qualidade de seus produtos, além disso, que busquem meios que possam contribuir para o melhor desempenho da equipe de certificação e testes. Contudo, esse processo é muito desafiador, tanto para empresa quanto para pessoas externas. Compartilhar lições aprendidas é fundamental para o aprendizado, além da promoção de discussões entre membros da equipe, da análise crítica das ferramentas, entre outros pontos vivenciados ao longo deste trabalho.

Dito isto, há de se destacar que essas experiências, para um aluno formando na área de computação é essencial para o aprendizado e qualificação do mesmo para o mercado de trabalho. Além disso, estar inserido em todas as fases desse processo, mostrou que além dos interesses em ambas as partes nesse processo, os resultados obtidos para empresa demonstram a exitosa experiência para ambas as partes.

6. Conclusões e Trabalhos Futuros

No processo de formação de um discente, ter experiências que contribuem para seu aprendizado e que o qualifiquem para o mercado de trabalho é fundamental. Dito isso, a ideia inicial desse desafio que a empresa propôs em buscar ferramentas que automatizam testes em seus produtos foi um processo rico para ambos os envolvidos.

Desde a primeira reunião, quando foi apresentado o Espresso, para testes iniciais até o ponto final em que foi discutido o TestProject foi uma caminhada que trouxe uma série de aprendizados acerca dos principais processos que uma empresa de tecnologia pode tomar para chegar a um resultado final satisfatório. Há de se destacar que esse processo na empresa ainda está ocorrendo no momento da escrita desse material e que os resultados encontrados são importantes. Isso permitiu que fosse investido mais com pesquisas acerca dessa e de outras ferramentas que possibilitam automatizar testes.

Além disso, é importante destacar que o processo foi tão frutífero que os testes realizados já relataram instabilidades no sistema que dificilmente um analista conseguiria encontrar. Além disso, a velocidade da execução e a quantidade de testes realizados possibilitou que a ferramenta escolhida fosse utilizada para realizar testes em outras interfaces, como Web, que não era o plano inicial. Porém, tendo em vista as possibilidades que essa ferramenta trouxe, foi realizado esse processo de automatização pela equipe.

Em um ponto desse processo, a própria ferramenta encontrou defeitos nos testes realizados, facilitando a busca por possíveis falhas no sistema, permitindo que fosse garantida uma qualidade melhor. Além disso, foi possível ter acesso a relatórios que ajudaram a entender alguns pontos essenciais, como, o tempo que alguns processos levam para serem executados e onde não era esperado esse tempo. Tendo em vista que nesse relatório era possível ver o tempo de execução de um caso de teste realizado de forma manual, ficou claro que esse processo contribuiu para um melhor aproveitamento do tempo dos testes realizados.

Por fim, há de se destacar alguns pontos importantes sobre esse processo. Primeiro, esse foi um passo inicial no desenvolvimento de automação deste tipo de testes na equipe e o aprendizado obtido por todos os membros foi fundamental no decorrer do processo. Há de se destacar também que foi anunciado pela empresa detentora do TestProject que a ferramenta será descontinuada, o que está fazendo com que a equipe procure outras ferramentas para realizar os testes que antes eram realizados no TestProject.

Com isso, a equipe está analisando novas ferramentas que possibilitem os testes automatizados em interface de usuário em dispositivos Android, sendo algumas das possibilidades o Appium ou Katalon Studio. O interessante é que isso vai possibilitar um avanço maior da equipe e a ideia de migração dos testes feitos anteriormente em "*Record and Playback*", tendo em vista que essas ferramentas possibilitam o uso de códigos para automação desses testes.

Referências

- Araruna, João Guilherme Santana (2017). Estratégias para realizar testes funcionais de interface com o usuário: visão de uma equipe de testes 90 f., il. Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Software)—Universidade de Brasília, Brasília, 2017.
- De Pádua Paula Filho, Wilson. (2019) **Engenharia de Software - Produtos - Vol.1**. [Digite o Local da Editora]: Grupo GEN. E-book. ISBN 9788521636724. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788521636724/>. Acesso em: 25 set. 2022.
- Gonçalves, Priscila de F.; Barreto, Jeanine dos S.; Zenker, Aline M.; et al. (2019). **Testes de software e gerência de configuração**. [Digite o Local da Editora]: Grupo A. E-book. ISBN 9788595029361. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788595029361/>. Acesso em: 09 out. 2022.
- Lamounier, Stella Marys D. (2021) Qualidade de software com Clean Code e técnicas de usabilidade. [Digite o Local da Editora]: Editora Saraiva. E-book. ISBN 9786589965565. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9786589965565/>. Acesso em: 14 nov. 2022.
- Pressman, Roger S.; Maxim, Bruce R. (2021) Engenharia de software. [Digite o Local da Editora]: Grupo A. E-book. ISBN 9786558040118. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9786558040118/>. Acesso em: 18 set. 2022.
- Rolim, Chiara; Rebouças, Ayla. (2021) Automating Transaction Testing Involving Payment at PoS (Point of Sale) Terminals: An Experience Report at a Technology Company. *In: SIMPÓSIO BRASILEIRO DE TESTES DE SOFTWARE SISTEMÁTICO E AUTOMATIZADO (SAST)*, 6. , 2021, Joinville. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação. p. 18–25.
- Santos, Almeida; Ferreira, Mello. (2019). Robô para Automação de Testes em Máquinas de Cartão de Crédito.
- Sommerville, Ian. (2019). Engenharia de software, 10º ed. Editora Pearson. Disponível em: <https://plataforma.bvirtual.com.br/Acervo/Publicacao/168127>. Acesso em: 13 nov. 2022
- Software Testing Help (2022) Disponível em: <https://www.softwaretestinghelp.com/what-is-end-to-end-testing/>. Acesso em: 21 nov. 2022.