Descrição de técnicas e ferramentas para a Medição e qualidade de software.

- Teste de Desenvolvimento

O software desenvolvido passou por diversos testes durante sua “confecção”, entre eles testes unitários das funções (verificando se retornavam valores corretos) e simulando valores em faixas (permitidas e não permitidas).

Testes de validação de variáveis:

- e-mail (Valida a entrada de e-mail somente com as condições "@" && ".com")
- CPF (valida somente CPF válido, utilizando os dígitos verificadores)
- senha (valida somente senha que contenha mais que 6 dígitos)
- placa do carro (valida a placa somente nos formatos “aaa-1111” ou “aaa1a11” onde a representa qualquer letra e 1 representa qualquer número)

Testes de botões:

- Salvar - Verifica se o botão salvar envia as informações para o "banco de dados", fazendo uma conferência entre os dados enviados e os dados salvos.
- Voltar - Verifica se ao clicar no botão, o app redireciona para a página inicial/página de login .
- Login - Testa se ao clicar em "login" o app compara dados informados com os dados salvos no "banco de dados" e permite o login de usuários cadastrados ou envia a informação de dados inconsistentes para usuários não cadastrados e/ou usuário que erraram na digitação.
- Cadastre-se - Verifica o redirecionamento para a tela de novo cadastro
- Esqueceu sua senha? - Verifica o redirecionamento para a tela de recuperação de senha
- Menu de opções - Verifica-se a abertura das opções do usuário
- Alterar senha - Verifica se os dois campos (senha e confirme nova senha) são idênticos antes de enviar a alteração para o "banco de dados".
- Adicionar saldo - Verifica-se o redirecionamento para a tela com opções de adicionar saldo na conta
- Adicionar valor - Verifica-se o valor correspondente foi adicionado em seu saldo (valor armazenado no "banco de dados")
- Sair - Verifica se o usuário é redirecionado para a tela de login

- Estratégias e ferramentas de teste

As ferramentas utilizadas foram:

- JUnit: para testes unitários.
- Robolectric: testar o “Maps”, uma vez que ele é uma dependência mais complexa do android.
- Mockito para testar interações específicas entre alguns componentes e sua dependência no app.

- Equipe e infra-estrutura

A equipe de testes foi formada pelos alunos Diego Menecheli e Victor Vilela, sendo o primeiro responsável por testes durante o desenvolvimento e o segundo por testes de release, após a conclusão (mesmo que parcial) do app. Utilizamos ferramentas presentes dentro do "Android Studio" (como por exemplo emulador de android), nossos celulares próprios para rodar a aplicação. Muitas das informações foram obtidas através do site "https://developer.android.com/", que reúne muita informação e dicas sobre o desenvolvimento para a plataforma. Os custos são difíceis de medir, considerando que todas as validações são feitas dentro do código para garantir o seu funcionamento, impossibilitando assim uma aferição do que é desenvolvimento do app e do que é somente teste.

- Execução do Plano de Teste

Teste de Release (Sistema)

Um membro do grupo que não participou da implementação do app testou o mesmo a fim de encontrar erros que persistiram mesmo após os testes feitos durante o desenvolvimento. Alguns erros foram percebidos e serão corrigidos para a entrega final do trabalho prático da disciplina.

Teste de Usuário (Aceitação)

Considerando que o app não será desenvolvido por completo e como depende também de uma movimentação para que seja testado e se tratando de um momento grave da pandemia não foi possível testá-lo em uma situação real de uso.

Porém foram testados a criação de conta, o login, alteração de dados pessoais, adicionar saldo (sem que necessitasse o real pagamento do boleto), visualização do maps em diferentes locais entre outras funções.