

Processo de desenvolvimento de uma app móvel - Calculadora

O processo de desenvolvimento de uma App móvel que funcione como uma calculadora, envolve uma série de etapas consequentes, que vão desde a definição dos requisitos pertencentes àquilo que irá ser a sua função para o utilizador, até à implementação da mesma, e sucessivos testes de funcionamento da mesma, como qualquer outra aplicação ou serviço, que exige sempre os seus testes fictícios e reais antes de ir para live, digamos assim, o que chamamos em contexto de trabalho, Produção/Produtivo.

Todos os produtos, sejam aplicações ou serviços passarão/passam por vários ambientes de testes até à entrega, normalmente 3. Teste (TST), Pré-Produção (PP) e Produção (PRD).

Na perspetiva do utilizador penso que devemos também pensar num desenho da mesma olhando para as tecnologias mais atuais e corretas, projetando claramente uma interface também que seja intuitiva para o utilizador, realizando todo o tipo de testes rigorosos, para garantir, no final, uma experiência positiva para os seus utilizadores. Devemos sempre pensar na ótica de utilizador também para evidenciar uma experiência positiva.

Em primeiro lugar, como acima foi mencionado, devemos preparar os requisitos e funcionalidades básicas da app, neste caso que está direcionada para servir como uma calculadora.

E quais são, atualmente, as funcionalidades básicas de uma calculadora?

- As suas operações aritméticas básicas: adição, subtração, multiplicação e divisão;
- As suas operações mais críticas: calcular percentagens, calcular raízes quadradas, fator exponencial, etc;
- As questões de memória da mesma: isto passa por armazenar a informação, recuperá-la e no fim limpá-la, e até mesmo pensar numa ótica em que o utilizador possa e deva realizar as operações inúmeras vezes;
- Aqui falando numa questão já mais profissional, podemos também pensar naquilo que será os types dos números que a calculadora deverá suportar, como decimais, booleanos, números negativos.

Após todos estes requisitos, ou pré-requisitos iniciais, podemos pensar em funcionalidades de uma calculadora, adicionais às mais básicas/pré-complexas. E aqui, vai de facto, da perspetiva do utilizador, sempre, mas do gosto de cada developer, digamos assim. Pois, pensando numa ótica de utilizador, podemos adicionalmente querer que a calculadora tenha:

- O histórico dos cálculos inicialmente feitos;
- O tema da calculadora poder ser o habitual claro ou um tema escuro;
- Usar pontos, usar vírgulas
- Factoriais
- Sen, Cos

Após estas primeiras funcionalidades que foram pensadas para a criação da mesma, podemos pensar o que vamos usar para o seu desenvolvimento, qual a plataforma, as suas tecnologias, as

ferramentas de desenvolvimento, o design da interface, pensar na perspetiva front-end e back-end da app.

Dentro daquilo que fomos aprendendo nestas primeiras aulas, podemos pensar no ambiente/plataformas que a calculadora deverá operar e isto depende muito dos requisitos iniciais, cliente final, etc. Porque podemos querer que a calculadora funcione num ambiente Windows, Linux, Android, IOS, ou só um deles, dependendo do cliente final e dos requisitos iniciais aliados a isso.

De seguida, temos as ferramentas que vamos utilizar para o seu desenvolvimento, como falamos nas aulas anteriores, o IDE (por exemplo, usa-se muito o Visual Studio Code), e também depende muito da tecnologia de programação que se utilize, podemos fazer uma app que seja uma calculadora e querer programar em Java e utilizarmos o Netbeans ou Eclipse como podemos querer desenvolver uma calculadora em Python e usarmos o VSCode por exemplo, ou o PyCharm, e depois, imagino, todas as componentes, bibliotecas etc para conseguir desenvolver a mesma com sucesso.

Para um controlo do código e suas futuras versões, isto numa perspetiva já mais profissional (confesso que muito provavelmente que é por isso que me lembro de alguns pormenores), poderíamos fazer recurso do GIT para versionamento do código.

Virados então, como sempre, para o utilizador, podemos chegar à parte em que chegamos ao Design de Interface do Utilizador. Aí, confesso que não sou conhecedor em concreto de quais as melhores práticas de Design, quais os melhores princípios, mas posso apontar alguns pontos que podem ser fulcrais/essenciais numa perspetiva de utilizador de uma calculadora, a saber:

- Uma interface clean;
- Uma interface intuitiva;
- Para uma perspetiva de todo o tipo de utilizadores, a parte da acessibilidade, botões intuitivos, grandes e espaçados.

Depois de escolher o design, e os seus requisitos/melhores práticas, tal como o desenvolvimento do código, escolher aqui ferramentas de design para o efeito (que aqui desconheço).

Depois, ao chegar ao desenvolvimento propriamente dito, devemos também ter em nota vários pontos para o correto desenvolvimento da app:

- Estruturar o Projeto:
 - Desenvolver a interface (front-end) : como já fiz na universidade em java, o display, o teclado numérico e o das operações;
 - Pensar como elaborar toda a lógica de uma calculadora/back-end: o início, atualização do display, funções de cada operação, etc;
 - E aquilo que neste momento são as minhas atuais funções profissionais, fazer aqui a ponte de ligação, ou seja, a parte de integração, conectar a interface do utilizador com a lógica dos cálculos.
- Escolher Linguagem de Programação

No fim de todo o desenvolvimento e testes unitários à medida que se vai desenvolvendo, temos os testes e debugs do programador e consequentemente da equipa de testes/QA. E que tipo de testes se costuma fazer ou se deveria fazer:

- Testes Unitários;
- Testes de Design;
- Testes às funcionalidades da calculadora;
- Testes de usabilidade;
- Testes de carga;
- Testes em dispositivos (Telemóveis, Computadores) Reais;
- Testes de Integração;
- Auditorias;
- Testes de bugs.

Após estas questões deveremos fazer um rescaldo de tudo o que foi feito até agora e olhar, numa perspetiva de algo parecido com metodologia SCRUM, por exemplo fazer um refinement: fazer ajustes e/ou melhorias com base naquilo que foram o resultado de todos os testes mencionados anteriormente, exemplo correção de bugs, otimização das funcionalidades, acrescentar algumas se necessário, e arranjar utilizadores para testes/feedbacks reais.

Depois de lançar o produto, finalmente, e de estar em live e ok para utilizar, fazer o devido lançamento, publicidade/marketing, pensar numa perspetiva de futuro. Tal como todas as aplicações e serviços, devem passar por fazer a respetiva monitorização e atualizações da mesma, corrigindo bugs ao longo do tempo, melhorando o design e utilização, adicionar novas funcionalidades, bem como o suporte ao cliente/utilizador.