

Implementação caso de uso 01 - Cadastrar produto

1) Classificar o caso de uso em uma escala de 1 (mais fácil) a 10 (mais difícil) em nível de dificuldade.

Classificação: 3

2) Fazer uma medição objetiva da quantidade de linhas de código produzida ao final do desenvolvimento do caso de uso.

Linhas de código: 181. Desconsiderando imports.

3) Fazer uma medição subjetiva de quais atividades tomaram tempo além do desenvolvimento. Ex: leitura de documentação, correção de erros, integração com APIs, consultas aos stakeholders, alterações na documentação já produzida ou qualquer outra atividade que não seja uma codificação propriamente dita.

Leitura e consulta da documentação do react-native-web, correção de erros visto o conflito entre react native e react native web (certos componentes não funcionam igualmente).

4) Considerando as informações levantadas e a análise das dificuldades apresentadas, fazer ajustes no planejamento das datas adicionando comentários sobre os motivos das reconsiderações.

As dificuldades ficaram em torno dos conflitos entre as diferentes bibliotecas. Apesar do atraso na implementação do protótipo, a implementação deste caso de uso foi feita no tempo certo, não gerando atrasos.

Implementação caso de uso 02 - CRUD Produtos

1) Classificar o caso de uso em uma escala de 1 (mais fácil) a 10 (mais difícil) em nível de dificuldade.

Classificação: 6

2) Fazer uma medição objetiva da quantidade de linhas de código produzida ao final do desenvolvimento do caso de uso.

Linhas de código: 107. Desconsiderando imports.

3) Fazer uma medição subjetiva de quais atividades tomaram tempo além do desenvolvimento. Ex: leitura de documentação, correção de erros, integração com APIs, consultas aos stakeholders, alterações na documentação já produzida ou qualquer outra atividade que não seja uma codificação propriamente dita.

Leitura e consulta da documentação do react-navigation, implementação do react navigation devida a navegação entre telas para criar/visualizar/atualizar/deletar um produto, correções de erros de incompatibilidade entre o react-native-web e react native.

4) Considerando as informações levantadas e a análise das dificuldades apresentadas, fazer ajustes no planejamento das datas adicionando comentários sobre os motivos das reconsiderações.

As dificuldades ficaram em torno dos conflitos entre as diferentes bibliotecas e no tempo de renderização em tempo real, sendo necessário implementar o hook useCallback no lugar do useEffect.

Implementação caso de uso 03 - Permitir que o cliente finalize sua compra

1) Classificar o caso de uso em uma escala de 1 (mais fácil) a 10 (mais difícil) em nível de dificuldade.

Classificação: 4

2) Fazer uma medição objetiva da quantidade de linhas de código produzida ao final do desenvolvimento do caso de uso.

Linhas de código: 66. Desconsiderando imports e navegações.

3) Fazer uma medição subjetiva de quais atividades tomaram tempo além do desenvolvimento. Ex: leitura de documentação, correção de erros, integração com APIs, consultas aos stakeholders, alterações na documentação já produzida ou qualquer outra atividade que não seja uma codificação propriamente dita.

Implementar a biblioteca React Navigation, revisar a modelagem dos dados e implementar toda a navegação, desde a lista de produtos, na tela principal até a tela de finalizar compra.

4) Considerando as informações levantadas e a análise das dificuldades apresentadas, fazer ajustes no planejamento das datas adicionando comentários sobre os motivos das reconsiderações.

As dificuldades ficaram em torno da Implementação da navegação e encaminhamento dos dados pela navegação.

Implementação caso de uso 04 - Permitir que o cliente adicione produtos a sua lista de desejos

1) Classificar o caso de uso em uma escala de 1 (mais fácil) a 10 (mais difícil) em nível de dificuldade.

Classificação: 8

2) Fazer uma medição objetiva da quantidade de linhas de código produzida ao final do desenvolvimento do caso de uso.

Linhas de código: 97. Desconsiderando imports e navegações.

3) Fazer uma medição subjetiva de quais atividades tomaram tempo além do desenvolvimento. Ex: leitura de documentação, correção de erros, integração com APIs, consultas aos stakeholders, alterações na documentação já produzida ou qualquer outra atividade que não seja uma codificação propriamente dita.

Correção dos bugs, implementação do React Navigation e estudo sobre o hook useCallback

4) Considerando as informações levantadas e a análise das dificuldades apresentadas, fazer ajustes no planejamento das datas adicionando comentários sobre os motivos das reconsiderações.

As dificuldades ficaram em torno da Implementação da renderização em tempo real, sendo necessário implementar o hook useCallback

Implementação caso de uso 05 - Cadastrar usuários

1) Classificar o caso de uso em uma escala de 1 (mais fácil) a 10 (mais difícil) em nível de dificuldade.

Classificação: 2

2) Fazer uma medição objetiva da quantidade de linhas de código produzida ao final do desenvolvimento do caso de uso.

Linhas de código: 215. Desconsiderando imports e navegações.

3) Fazer uma medição subjetiva de quais atividades tomaram tempo além do desenvolvimento. Ex: leitura de documentação, correção de erros, integração com APIs, consultas aos stakeholders, alterações na documentação já produzida ou qualquer outra atividade que não seja uma codificação propriamente dita.

Correção dos bugs, implementação do React Navigation, implementação do hook useState e firebaseAuth

4) Considerando as informações levantadas e a análise das dificuldades apresentadas, fazer ajustes no planejamento das datas adicionando comentários sobre os motivos das reconsiderações.

As dificuldades ficaram em torno da Implementação da função firebase.Auth.

Implementação caso de uso 06 - Permitir login

1) Classificar o caso de uso em uma escala de 1 (mais fácil) a 10 (mais difícil) em nível de dificuldade.

Classificação: 2

2) Fazer uma medição objetiva da quantidade de linhas de código produzida ao final do desenvolvimento do caso de uso.

Linhas de código: 86. Desconsiderando imports e navegações.

3) Fazer uma medição subjetiva de quais atividades tomaram tempo além do desenvolvimento. Ex: leitura de documentação, correção de erros, integração com APIs, consultas aos stakeholders, alterações na documentação já produzida ou qualquer outra atividade que não seja uma codificação propriamente dita.

Correção dos bugs, implementação do React Navigation, implementação do hook useState e firebaseAuth. Same da useCase 05

4) Considerando as informações levantadas e a análise das dificuldades apresentadas, fazer ajustes no planejamento das datas adicionando comentários sobre os motivos das reconsiderações.

As dificuldades ficaram em torno da Implementação da função firebase.Auth. Same do useCase 05.