



Universidade Federal do Ceará
Centro de Ciências
Departamento de Computação
Verificação, Validação e Teste de Software (CK0241)

TB01 - Escopo do projeto de V&V

E-commerce

Grimberg Cryzan - 362964

Beatriz Monte - 371781

José João Silva - 408847

Levir César Ribeiro Lemos - 400555

Histórico de versões

Versão	Data	Responsável	Descrição
1.0	07/12/2020	José João	Criação do documento
2.0	08/12/2020	Levir Lemos	Inclusão da Introdução, bibliografia, aplicação
3.0	09/12/2020	José João. Levir, Grimberg, Beatriz	Edição do conteúdo da aplicação do projeto, funcionamento e tecnologias usadas, introdução, requisitos funcionais, requisitos não funcionais, requisitos a serem testados e conclusão.

Sumário

Introdução	3
- A aplicação	3
- Funcionamento	3
- Tecnologias usadas	4
Requisitos Funcionais	5
Requisitos Não Funcionais	6
Funcionalidades a serem testadas	6
Conclusão	8
Bibliografia ou Referência	8

1. Introdução

Durante o desenvolvimento de um projeto de V&V deve-se tomar alguns processos como base para se ter um guia de como proceder em cada etapa do projeto. Como pontapé inicial a apresentação da aplicação escolhida pelos membros da equipe, seus requisitos funcionais, não funcionais e quais desses requisitos que irão ser testados no projeto de V&V.

Sobre a aplicação, foi escolhido um e-commerce denominado E-commerce que possui o objetivo de promover uma interface amigável para uma loja online, tanto na visão do cliente, como na visão do funcionário. A aplicação foi desenvolvida seguindo o padrão MVC de modo que restrinja funcionalidades baseadas em quem está usando a aplicação.

1.1. A aplicação

O E-commerce é uma aplicação open source, desenvolvida para fins educacionais, mas podendo ser estendida para uso comercial. O objetivo é que ele consiga fornecer, através de uma interface amigável, maneiras de gerenciar todos dados básicos presentes em um e-commerce, tais como funcionários, clientes, produtos, categoria de produtos, vendas e dados pessoais.

Todos os códigos fonte do projeto estão disponíveis em um repositório aberto e hospedados no github. O link é: <https://github.com/jjoaosilva/ecommerce>

1.1.1. Funcionamento

Ao usar o aplicativo, o usuário terá acesso à área de login, podendo logar como cliente ou funcionário. Caso não tenha cadastro a plataforma fornece suporte para registro de nova conta.

Imagem acima apresenta o login feito como cliente

Como cliente, após fazer login, é possível ver todos os produtos da loja em suas respectivas categorias, adicionar produtos ao carrinho, gerenciar o carrinho, realizar a compra de um produto, ver todo o seu histórico de compras e alterar os seus dados pessoais.

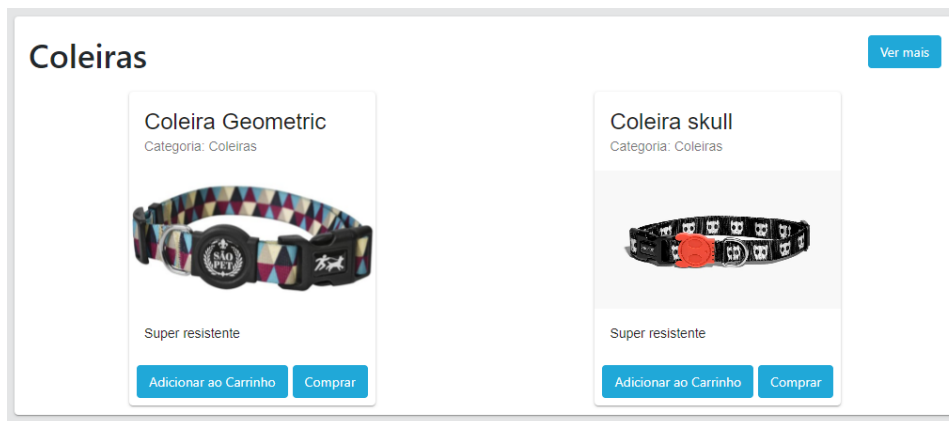


Imagem acima apresenta como o cliente visualiza os produtos de uma categoria

Como funcionário, após fazer login, é possível ver todos, cadastrar, editar e deletar: categorias, produtos, funcionários e clientes.

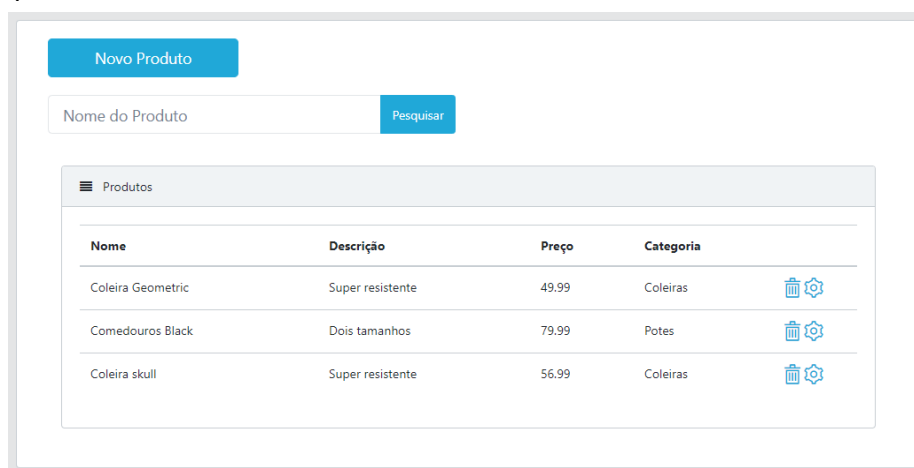


Imagem acima apresenta como o funcionário tem acesso aos produtos e seus respectivas ações. Esse padrão se repete para categorias, funcionários e clientes.

1.1.2. Tecnologias usadas

Todo frontend foi construído com **React JS** e é compatível com versões acima de 16.8.5 do react. o frontend se comunica com uma API (com arquitetura REST) criada com o microframework **Flask** que é codificada em **Python**, funcionando atualmente (09/12/2020) na versão 3.9.0 do Python e na versão 1.1.2 do Flask. Todos os dados são armazenados em um banco **PostgreSQL**.

2. Requisitos Funcionais

Após uma análise do projeto feita a partir do próprio uso da aplicação e leitura de alguns documentos que estão no próprio repositório, podemos notar os seguintes requisitos funcionais:

Código	Descrição do requisito
RF001	Logar como cliente
RF002	Logar como funcionário
RF003	Adicionar itens ao carrinho
RF004	Alterar quantidade de um item do carrinho
RF005	Realizar uma compra
RF006	Ver suas compras passadas
RF007	Alterar dados pessoais
RF008	Criar um produto
RF009	Editar um produto
RF010	Apagar um produto
RF011	Criar uma categoria
RF012	Editar uma categoria
RF013	Apagar uma categoria
RF014	Criar um funcionário
RF015	Editar um funcionário
RF016	Apagar um funcionário
RF017	Criar um cliente
RF018	Editar um cliente
RF019	Apagar um cliente
RF020	Fazer logout

3. Requisitos Não Funcionais

Após uma análise do projeto feita a partir do próprio uso da aplicação e leitura de alguns documentos que estão no próprio repositório, podemos notar os seguintes requisitos não funcionais:

Código	Descrição do requisito
RNF001	Processar pagamentos na própria aplicação
RNF002	Salvar dados do carrinho na cache do browser

4. Funcionalidades a serem testadas

Após a análise de todos os requisitos, decidimos testar os requisitos que influenciam de maneira direta a experiência do usuário durante o uso do e-commerce do ponto de vista do funcionário.

Requisito	Será testada	Não será testada
RF001		X
RF002		X
RF003		X
RF004		X
RF005		X
RF006		X
RF007		X
RF008	X	
RF009	X	
RF010	X	
RF011	X	
RF012	X	
RF013	X	
RF014	X	
RF015	X	
RF016	X	

RF017	X	
RF018	X	
RF019	X	
RF020		X
RNF001		X
RNF002		X

5. Conclusão

Como a aplicação envolve o registro de categorias, produtos, funcionários e clientes e isso é uma das partes principais da aplicação, tendo em vista que sem isso não haveria uma dinamicidade dos dados, optamos por testar os requisitos funcionais relacionados ao registro dessas entidades.

6. Bibliografia ou Referência

João Silva. Github, 2018. E-commerce. Disponível em:
<<https://github.com/jjoaosilva/ecommerce>>. Acesso em: 08, dezembro de 2020.

Core UI. Github, 2018. Coreui-free-react-admin-template. Disponível em:
<<https://github.com/coreui/coreui-free-react-admin-template>>. Acesso em: 08, dezembro de 2020.