

Projeto: *Seleção Oliveira Trust – Calculadora de Pagamentos*

Autor: Mario Henrique Meireles de Aquino
Data: 29/01/2022

Sumário

01- Considerações iniciais

02- Requisitos funcionais

F1 Calcular pagamentos

F2 Consultar cotações

F3 Envio de cotações por e-mail para o usuário.

F4 Armazenar histórico de cotações do usuário.

F5 Ver histórico cotações

F6 Editar as taxas aplicadas

03- Casos de uso

04- Pacote

05- Tempo de desenvolvimento

01- Considerações iniciais

As características do sistema

O presente projeto foi desenvolvido com Laravel na versão 6.

O pacote de autenticação é do próprio **Laravel**.

A interface é desenvolvida usando os Frameworks **Jquery** e **Bootstrap**, com a interface baseada no **CoreUi**.

A estrutura básica do projeto usa OO com PHP.

O cliente usado para consumir os endpoints externos, foi desenvolvido em um pacote na pasta “packages” e importado no Laravel a partir de um repositório local.

Os endpoints AJAX para o cliente, mantendo toda a regra de negócios no servidor para a captura do processamento da conversão.

O SQLite foi SGBD escolhido por ser mais facilmente portátil.

A análise

A análise feita aqui é para mapear os requisitos básicos e delimitar o que deve ser entregue (Casos de uso).

O Documento de requisitos funcionais é retirado do livro “Análise e design orientados a objetos para sistemas de informação”, 2ª e 3ª edições. A metodologia usada pelo autor é o *RUP (Rational Unified Process)*, com desenvolvimento guiado por ciclos evolutivos (*sprints*) e casos de uso, e gestão de requisitos.

A “classificação de requisitos suplementares” são baseados na **ISO/IEC 25010:2011** (livro citado, 3ª Edição, tabela 3.1, pag. 57-62). A saber: **Categoria**, a categoria a que o Requisito Não Funcional se enquadra.

As dicotomias **Obrigatório/ Desejável** e **Transitório/Permanente** (livro citado, 3ª Edição, tabela 3.1, pag. 55-57), são classificações de *Requisitos não Funcionais*. **Obrigatório** implica em implementação; **Desejável**, pode ser implementado de acordo com as demandas do projeto e que não tenha impacto sobre as entregas. **Transitório** e **Permanente** são definidos por conveniência no projeto/cliente e indicam se um requisito pode ou não mudar.

Os casos de uso, devido ao pequeno escopo a ser implementado, são apenas enumerados para indicar as funcionalidades a serem entregues e orientar o desenvolvimento.

Autenticação

A autenticação possui separação entre área administrativa e área do cliente. As estruturas das interfaces são reusadas.

02- Requisitos funcionais

F1 Calcular pagamentos				Oculto (X)
Descrição: O sistema deve calcular taxas e impostos				
Requisitos Não funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF1.1 Cálculo	<p>Variáveis:</p> <p>Moeda de origem: BRL, USD, BTC,...</p> <p>Moeda de destino: BRL, USD, BTC,...</p> <p>Valor da compra: > R\$ 1.000,00 e < R\$ 100.000,00; moeda BRL;</p> <p>Formas de pagamento: compra no boleto, taxa de 1,45%; compra no cartão de crédito, taxa 7,63;</p> <p>Taxa de conversão: Valor da compra < R\$3000 é 2%; Valor da compra > R\$300000 é 1%;</p> <p>Valor comprado em "Moeda de destino": (total de taxas aplicadas ao valor de compra - o valor total de conversão) * taxa de câmbio.</p>			
NF1.2 Tela da calculadora	<p>O usuário pode escolher opções de moeda estrangeira para valores de compra > R\$1.000,00 e < R\$100.000,00.</p> <p>O valor deve ser exibido na Moeda de destino.</p>	Interface		

F3 Envio de cotações por e-mail para o usuário.				Oculto ()
Descrição: O usuário e-mail com cotações atualizadas.				
Requisitos Não funcionais				

Nome	Restrição	Categoria	Desejável	Permanente
NF3.1 Envio	O usuário logado dispara o envio de e-mails.	Segurança		
NF3.2 Período entre envios.	Os e-mails devem ser disparados a cada consulta.	Interface		

F4 Armazenar histórico de cotações do usuário				Oculto (X)
Descrição: O sistema deve armazenar as cotações feitas por um cliente.				
Requisitos Não funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF4.1 Dados	Os dados armazenados são: <ul style="list-style-type: none"> • usuário • cotação 	Persistência		X

F5 Ver histórico cotações				Oculto ()
Descrição: O sistema deve mostrar ao cliente o histórico das suas cotações.				
Requisitos Não funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF5.1 Acesso	O acesso é permitido apenas ao usuário logado.	Segurança		
NF5.2 Acesso cliente	O cliente pode ver apenas seus dados	Segurança		
NF5.3 Filtros do cliente	Filtros: - intervalo de datas, - moeda de origem - intervalo de valores de compra	Interface		
NF5.4 Dados da cotação exibidos ao cliente	<u>Dados da cotação</u> - Valor de compra - Moeda de origem - Moeda de destino - Cotação do dia - Forma de pagamento - Taxa da forma de pagamento - Taxa de conversão	Interface		
NF5.4 Acesso administrador	O administrador pode acessar os dados de todos os clientes.	Segurança		
NF5.5 Filtros do administrador	O administrador pode filtrar por cliente, além dos filtros disponíveis para o cliente.			

F6 Editar as taxas aplicadas				Oculto ()
Descrição: O sistema deve permitir ao administrador editar as taxas aplicadas.				
Requisitos Não funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF6.1 Acesso.	Somente administrador logado	Segurança		
NF6.2 Validade	As taxas alteradas devem possuir um período de validade.	Persistência		

03- Casos de uso

C01 - Usuário consulta cotação

C02 - Usuário vê histórico das suas cotações

C03 - Administrador altera taxas de um cálculo

04- Pacote

mario/client-exchange-guzzle: Cliente HTTP para consultas às cotações, em packages.

05- Tempo de desenvolvimento

Documentação: 3h

Codificação: 16h