

Universidade do planalto catarinense

Curso de sistemas de informação

Ricardo Taruhn Duarte

INtrodução a arquitetura de redes

Lages (SC)

2020

Sumário

1 levantamento de requisitos 3

2 Requisitos de usuário: 4

3 diagrama de classe 5

4 como se faz a instalação de github 6

1. levantamento de requisitos

O sistema de lista de compra tem como função objetiva, desenvolver uma solução em controle de gastos para manter a média de compras mensais, obter uma lista para ter o controle de dados, com ela ter uma média em quanto é gastado em cada mês e fazer uma valor estimado em o quanto e o que pode ser gastado, sendo abaixo do média ou acima, deve haver um relatório de todos os produtos comprados no mês.

Requisitos funcionais:

* Adicionar uma lista de compras.
* Adicionar um produto.
* Cadastrar produtos requisitos de: nome, a unidade de compra do produto, quantidade mensal, quantidade comprada, e o preço estimado dele.
* Editar os valores dos produtos e da quantidade comprada no mês, modificar o nome do produto, tendo a possibilidade de ter sido comprado com uma marca diferente.
* Listar o valor estimado do mês, a quantidade comprada, e o que foi comprado no mês
* Deletar produtos que não precisam está mais na lista por inúmeros fatores e manter os dados mais limpos e apresentar o necessário nas tabelas.
* Calcular o valor total da compra dos produtos.

Requisitos não funcionais:

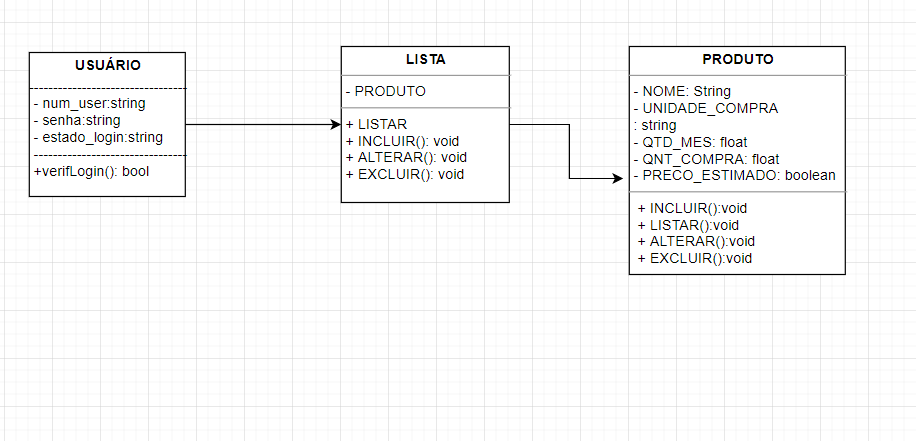
* O software deve documentar e fazer backup de cada mês feito pela lista de compras.
* O software precisa ser operado via computador ou notebook.

1. Requisitos de usuário:

|  |  |  |
| --- | --- | --- |
| CÓDIGO | DESCRIÇÃO | PRIORIDADE |
| RF1 | **Condicionar uma sistematização de adicionar uma lista de compras.** | **Obrigatório** |
| RF1.1 | **A lista de compras deve adicionar o cadastro de um produto;** | **Obrigatório** |
| RF1.1.1 | **Os dados do cadastro do produto devem ser:** | **Obrigatório** |
| RF1.1.1.1 | **Nome do produto, unidade de compra, quantidade no mês, quantidade de compra, preço estimado.** | **Obrigatório** |
| RF2 | **A execução do programa deve ser operada em um computador** | **Obrigatório** |
| RF2.1 | **Quando o cliente liga a aplicação e não existe uma lista armazenada: permitir a criação de uma lista** | **Obrigatório** |
| RF2.1 | **Permitir o status atual da lista de compras Gerenciamento dos produtos** | **Obrigatório** |
| RF2.1.1 | **Listar o relatório da lista de compras feita no mês** | **Obrigatório** |
| RF2.1.2 | **Listar todos os produtos adicionados no mês** | **Obrigatório** |
| RF2.1.1.1 | **Permitir edição dos valores dos preços dos produtos em geral** | **Obrigatório** |
| RF2.1.1.2 | **Permitir edição da unidade de valor utilizada do produto** | **Obrigatório** |
|  | **Permitir edição da quantidade limite dos produtos em geral** | **Obrigatório** |
|  | **Deletar produtos que não vão agregar no relatório da lista do mês** | **Obrigatório** |
|  | **Gerar o valor deve ser calculado na soma de todos os produtos listados** | **Obrigatório** |
|  | **Gerar o relatório da lista de compras feita no mês** | **Obrigatório** |

1. diagrama de classe

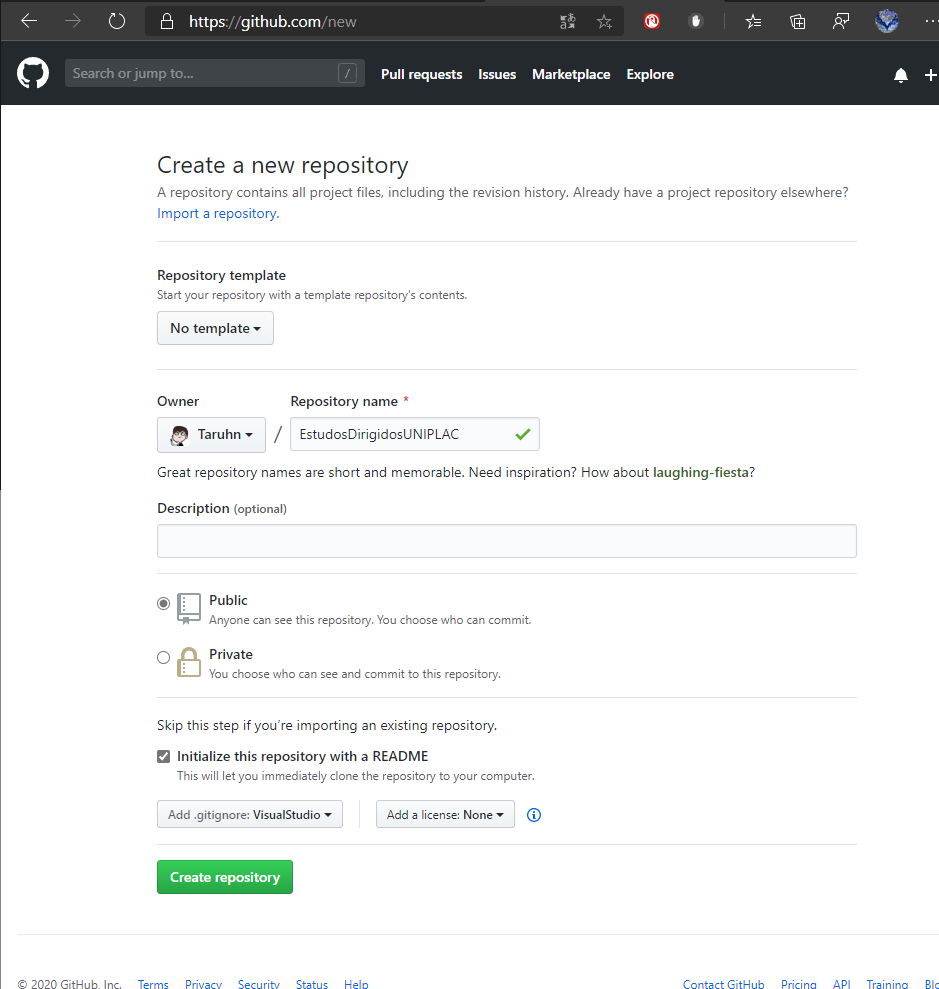
Nesse diagrama de classe para a lista de compras foi feito com os seguintes requisitos:

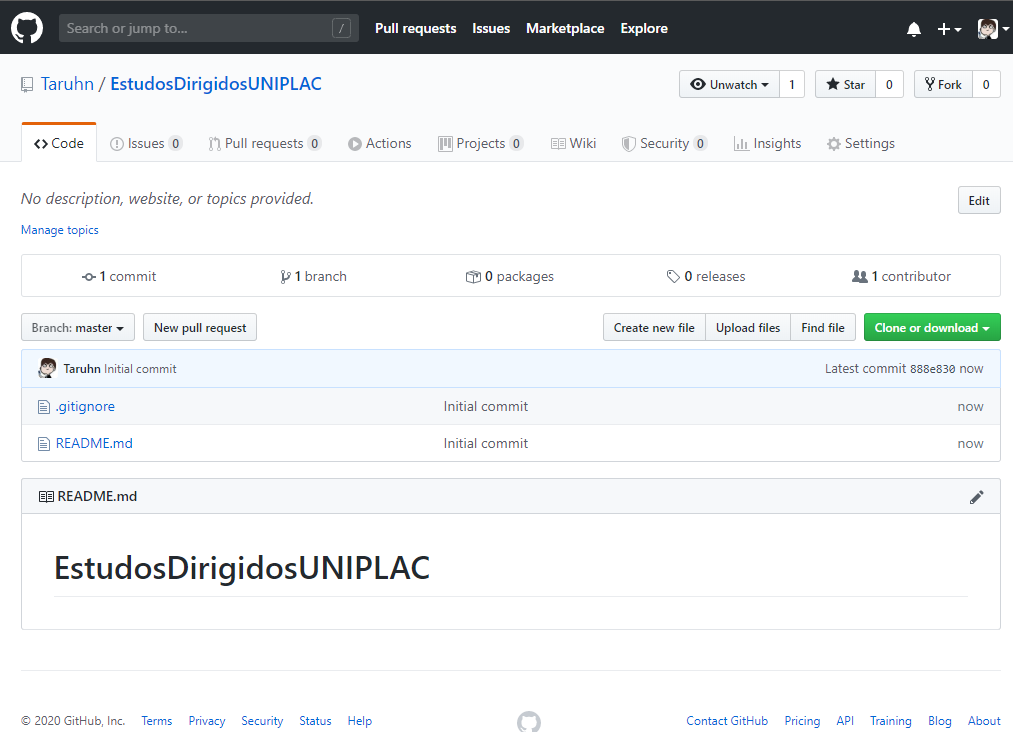


* Usuário: Nele é necessário um número de usuário e a senha para entrar na aplicação de lista e apresentar o estado do login, para verificar ele mostra uma booleana se está online ou não
* Lista: Dentro da lista existe o produto, nesse produto ele tem as funções que um produto tem que fazer, que é listar a lista de compras, incluir, algum dado da lista, alterar para fazer mudanças e atualizar ela, e excluir que nada mais é de deletar alguma coisa dentro da lista.
* Produto: para descrever um produto dentro da lista, ele deve ter o nome, o tipo de unidade da compra, a quantidade a ser comprada durante o mês, a quantidade atual de compras feitas e o preço estimado do valor do produto, e nele tem as funções todas as funções que a Lista apresenta.

1. como se faz a instalação de github

Primeiramente eu no meu GitHub: Taruhn, e entrei no https://github.com/new, para criar o meu repositório, o nome do repositório se chamará, “EstudosDirigidosUNIPLAC”, e em seguida eu tenho opções, nas quais descrevem se eu desejo que esse depositório fique privado ou público, nesse caso eu irei escolher privado porque esse repositório ele tem a importância de ser só eu e o professor de ter esse privilégio de acessar ele.



Em seguida eu colocarei dentro do repositório um *readme* para ver ser ele foi criado com sucesso. Feito assim, o repositório terá sido criado com esse caminho [https://github.com/Taruhn/EstudosDirigidosUNIPLAC.](https://github.com/Taruhn/EstudosDirigidosUNIPLAC.)

Depois de criado, precisa criar uma pasta que possa fazer o repositório dos arquivos que estão no GitHub para o computador, para isso é necessário fazer um Clone dos documentos usando o caminho que o repositório foi criado, sendo ele o <https://github.com/Taruhn/EstudosDirigidosUNIPLAC.git>, em seguida pegar esse link e colocar no terminal do computador e fazer a criação.

Dentro do terminal de controle, para colocar os documentos, é necessário criar uma pasta dedicada para colocar os documentos para não comprometer os arquivos e serão só de uso para esse bem.



Para ser colocado na pasta desejada, em primeiro precisa colocar o caminho do diretório dento do terminal de controle, no caso de estar dentro do no meu computador pessoal ficou assim, C:\Users\ricar\Desktop\Documents\EstudosDirigidosUNIPLAC, essa é a localização que ele ficam, em seguida é posto o comando “git clone”, e o link do repositório do git, com isso ele clona todos os arquivos do repositório dentro do git e coloca dentro da pasta dedicada.

Para colocar arquivos dentro do repositório, é necessário está na posição do diretório, e usar o comando “**git add**”, esse comando adiciona as mudanças do repositório que foram adicionadas dentro dela, depois de executado esse passo, para confirmar se houve alterações ou não se usa o comando “git status”, nele mostra o que está dentro do repositório, se houver alguma alteração ele vai mostrar a cor do nome do documento em vermelho.

Em seguida se usa o comando “**git commit -m “mensagem explicando o que eu estou fazendo”** ”, que significa uma confirmação do que está mandando para o servidor, a partir disso, caso queria adicionar algo mais tem que fazer outra mudança fazendo esse processo.

E para finalizar para mandar o documento para o repositório, se usa o comando “git push” que nada mais é que ele mandar o commit localmente da maquina para o servidor (repositório), para confirmar a transferência do arquivo ele pede o e-mail e a senha do criador do repositório para confirmar a transferência, e com isso o documento estará dentro do servidor.