Rodrigo Bevilacqua Marcondes1; Adailson de Almeida dos Santos2; Andre Vinicius Vieira Mildenberg2

*1 Docente do curso de Sistemas de Informação do Centro Universitário UNIFACEAR;*

*2 Graduando(a) do curso de Sistemas de Informação do Centro Universitário UNIFACEAR*

**RESUMO**

*Realizamos o desenvolvimento de um site de gestão de compras chamado SimpleBuy com o back-end programado em python utilizando o framework Django, para realizar o planejamento do desenvolvimento do sistema utilizamos diagramas UML, como o diagrama de casos de uso, diagrama entidade relacionamento e diagrama de classes.*

*O sistema consiste em um site onde empresas poderão fazer toda sua gestão de compras por meio dele, desde à cotação das compras até o registro das notas fiscais mantendo um histórico de todas as compras.*

*O foco do sistema é atingir as pequenas e medias empresas que muitas vezes não conseguem arcar com os altos custos dos softwares mais populares do mercado.*

*Palavras chave: compras, python, site.*

***ABSTRACT***

*We developed a purchasing management site called SimpleBuy with the back-end programmed in python using the Django framework, to carry out the system development planning we used UML diagrams, such as the use case diagram, entity relationship diagram and diagram of classes.*

*The system consists of a website where companies can manage all their purchases through it, from the quotation of purchases to the registration of invoices, keeping a history of all purchases.*

*The focus of the system is to reach small and medium companies that often cannot afford the high costs of the most popular software on the market.*

*Key Words: shopping, python, website.*

**1. INTRODUÇÃO**

Hodiernamente, no setor de software de compras empresarias existem duas grandes problemáticas, que são a grande complexidade de utilização que os softwares mais famosos do mercado possuem, além dos altos custos que os mesmos cobram de seus clientes.

Dessa forma resolvemos criar o *SimpleBuy* um sistema web com o intuito de ser uma solução com um melhor custo benefício que os concorrentes além de oferecer uma maior simplicidade na sua utilização afim de facilitar seu uso no cotidiano. O nosso software foi criado para suprir a carência que pequenas e medias empresas tem de ter um sistema especifico para compras onde por meio dele elas possam se organizar melhor no processo de compras. Dessa forma podendo conquistar uma maior economia financeira e uma melhor gestão de tempo.

**2. DESENVOLVIMENTO**

Para este projeto nós desenvolvemos um software para compras empresarias onde o front-end foi feito em HTML enquanto o back-end foi desenvolvido em linguagem Phyton com o framework django com base nos princípios de orientação a objetos (POO). Ademais, para nos auxiliares no desenvolvimento nos utilizamos o editor Visual Studio Code e a IDE Pycharm.

Além disso, no processo de desenvolvimento nós utilizamos diagramas baseados na UML para definirmos de forma mais concreta o que o sistema deverá conter, para desta forma, sermos mais produtivos no desenvolvimento. Os diagramas utilizados foram o diagrama de casos de uso, diagrama de entidade relacionamento (DER) entre outros.

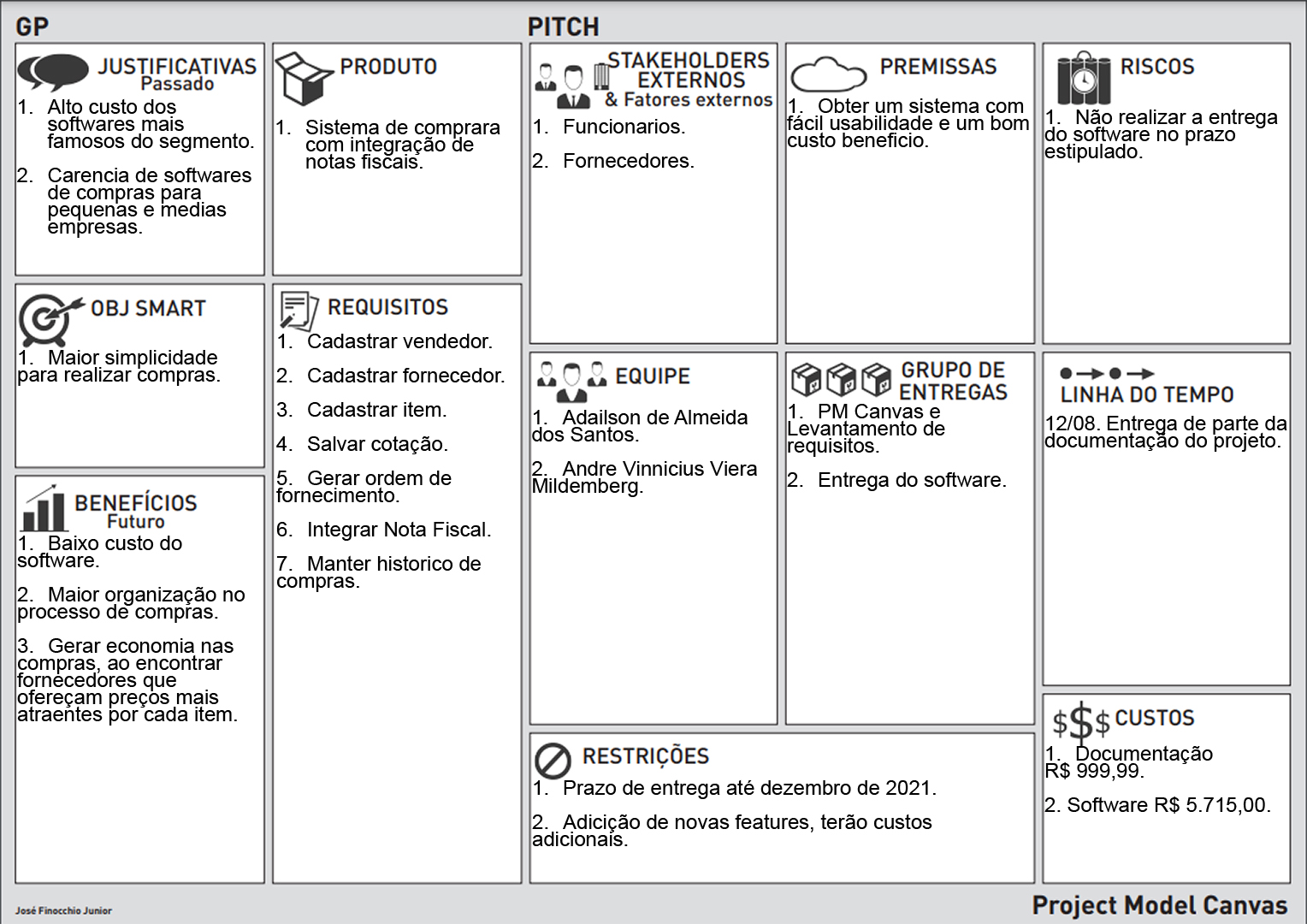
No tópico de desenvolvimento abordaremos no Tópico 2.1 o PMCanvas do projeto, no 2.2 veremos a análise de concorrência feita em um software com intuito parecido ao SimpleBuy. Nos tópicos 2.4 e 2.5 veremos respectivamente os requisitos funcionais e não funcionais do programa. No 2.6 as regras de negócios da aplicação. Enquanto, nos tópicos 2.8 e 2.9 veremos o diagrama de casos de uso e a descrição de casos de uso do sistema. E por fim no tópico 2.10 o diagrama de entidade-relacionamento.

2.1 PMCANVAS

Para organização do projeto foi utilizado o Project Model Canvas (PMCanvas) que é uma metodologia de gerenciamento de projetos focada na agilidade e flexibilidade podendo ser aplicado no desenvolvimento de diversos tipos de projetos o PMCanvas foi desenvolvido pelo professor José Finnochio.

Segundo OSTERWALDER, o PM Canvas funciona como uma espécie de guia para empreendedores que buscam [revolucionar os seus negócios](https://www.projectbuilder.com.br/blog/o-que-e-gestao-da-mudanca-e-por-que-isso-importa-para-seu-negocio/) a partir da utilização de metodologias inovadoras e que proporcionem uma melhor articulação estratégica.

A seguir na figura 1, o PM Canvas do projeto SimpleBuy:

****

2.2 ANÁLISE DA CONCORRÊNCIA

De acordo com FERRERN, a análise de concorrência, também conhecida como análise de mercado, nada mais é do que o[estudo sobre o que as empresas que são suas concorrentes](https://www.agendor.com.br/blog/como-fazer-um-estudo-de-mercado/) oferecem ao público, sobre como fazem a sua gestão comercial e como são vistas por clientes e potenciais clientes.

Sob tal ótica, observa-se que é fulcral realizar a análise de concorrência para analisar os pontos fortes e fracos das empresas concorrentes e dessa forma conseguir moldar o seu negócio de maneira que o público alvo receba e usufrua de seu produto da melhor forma possível. Dessa forma, o seu negócio poderá ter mais chances de alcançar o sucesso.

Foi realizado a análise de um software, que apresenta funcionalidades similares as do projeto que estamos desenvolvendo, que será apresentado no tópico 2.2.1.

2.2.1 Mercado Eletrônico

Hoje em dia o software mais famoso do seguimento de sistemas de compras é o Mercado Eletrônico, o qual é usado por diversas grandes empresas como: Nestle, Toytota, JBS, Vigor entre outras.

Com base nisso revolvemos fazer a análise de concorrência sobre essa empresa, para que dessa forma consigamos replicar e aprimorar seus pontos positivo e tentar resolver seus pontos negativos.

Logo quando entramos no seu site é perceptível a grande atenção que a empresa dá a ao design e apresentação sobre seu software além de ser muito objetiva na exibição destes dados, de uma forma que facilita a interação com os potenciais clientes.

Em relação aos preços o site deixa explicito os valores em relação aos planos para vendedores que são um plano grátis com recursos bastante limitados, plano básico com o valor de R$ 399,00/mês que contem mais recursos disponíveis e um plano Plus que custa R$ 549,00/mês com todos os recursos disponibilizados pela empresa.

Já em relação aos custos para os compradores o site não exibe de maneira clara, fazendo com que o cliente que deseja o serviço solicite uma analise de sua empresa para saber qual será o custo. A seguir uma relação simplifica dos principais pontos positivos (Tópico 2.2.1.1) e negativos (Tópico 2.2.1.2) da empresa.

2.2.1.1 Pontos Positivos

Foram identificados os seguintes pontos positivos no produto Mercado Eletrônico:

* Possui uma versão grátis para vendedores e uma demo para compradores;
* Bom design e interação no site;
* Disponível em mais de 1 (um) idioma;
* Grande assistência de suporte ao cliente;
* Possuem uma grande base de fornecedores;
* Possui compatibilidade com smartphones e tablets;

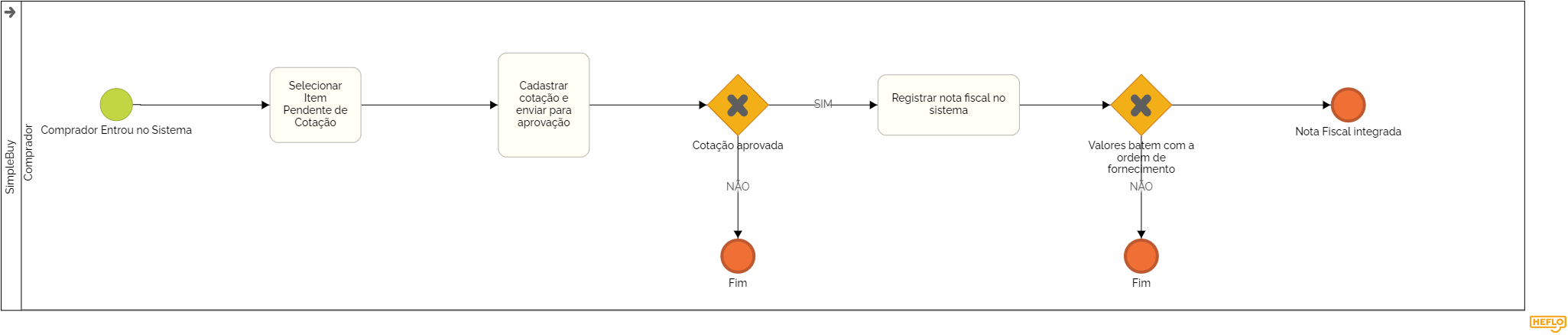
2.2.1.2 Pontos Negativos

Foram identificados os seguintes pontos negativos no produto Mercado Eletrônico:

* Versões mais baratas são limitadas em relação aos recursos que oferecem;
* Não mostra o preço do serviço para compradores;

2.3 MODELO DE PROCESSO DE NEGÓCIO

De acordo com o autor VEYRAT Modelagem de processos de negócios é a representação gráfica do processo e todos os seus passos. Faz parte de uma metodologia chamada Business Process Management – Gerenciamento de Processos de Negócios – e é essencial para o crescimento de uma empresa.



2.4 REQUISITOS FUNCIONAIS

Segundo o autor VENTURA, quando falamos de um Requisito Funcional estamos nos referindo à requisição de uma função que um software deverá atender/realizar. Ou seja, exigência, solicitação, desejo, necessidade, que um software deverá materializar.

É fundamental para qualquer sistema o levantamento de requisitos e sua organização, o levantamento de requisitos na maioria das vezes é feito no início do processo de desenvolvimento, é ele que guia todo o resto do desenvolvimento. Além disso, é importante ressaltar que todo requisito funcional precisa ter ao menos um requisito não funcional relacionado a ele, os requisitos não funcionais serão vistos no item 2.5

|  |  |
| --- | --- |
| Requisitos Funcionais | Descrição |
| RF1 Cadastrar Comprador | O sistema deve permitir o cadastro de compradores |
| RF2 Cadastrar fornecedor | O sistema deve permitir o cadastro de fornecedores com diversas informações como CNPJ, contato entre outros. |
| RF3 Manter Cotação | O sistema deve possibilitar ao comprador salvar as cotações de cada item com base no valor ofertado por cada fornecedor |
| RF4 Gerar Ordem de fornecimento | Após salvar as cotações dos itens do pedido o comprador poderá gerar a ordem de fornecimento formalizando o pedido com varias informações como o saldo do pedido, data de entrega, valor entre outros. |
| RF5 Integrar nota fiscal | O sistema possibilitará a integração de notas fiscais consumindo assim o saldo da ordem de fornecimento dos itens da nota. |
| RF6 Manter histórico de compras | Toda vez que uma ordem de fornecimento for criada o sistema irá salvar as informações dessa ordem. |
| RF7 Cadastrar item | O sistema possibilitará o cadastro de itens com um código e uma descrição. |

2.5 REQUISITOS NÃO FUNCIONAIS

De acordo com o autor VENTURA, Um Requisito Funcional define o que o sistema fará, e o Requisito Não-Funcional define como o sistema fará.

Diante dessa afirmação, observa-se que os requisitos não-funcionais são tão importantes quanto os requisitos funcionais, pois são eles que limitam as funcionalidades e garantem a segurança do sistema.

|  |  |
| --- | --- |
| **Requisito não funcional** | **Descrição** |
| RNF 1.1 Restrição de acesso. | Apenas o perfil do administrador do sistema pode cadastrar compradores |
| RNF 2.1 CNPJ | Todo fornecedor cadastrado deve ter um CNPJ válido. |
| RNF 3.1 Tempo de salvamento | As cotações de todos os fornecedores ficam salvas somente até o comprador fechar a Compra, após isso somente cotação final fica salva definitivamente. |
| RNF 4.1 Tempo limite | A ordem de fornecimento fechará automaticamente 1 mês após a data de entrega do pedido caso o mesmo não tenha sido entregue |
| RNF 5.1 Ordem de fornecimento aberta | Deve haver uma ordem de fornecimento aberta com os itens faturados na nota fiscal |
| RNF 5.2 Valores em conformidade com a ordem de fornecimento | Os valores faturados na nota fiscal devem ser precisamente iguais aos da ordem de fornecimento a qual a nota se refere. |
| RNF 6.1 Imutabilidade | O histórico será imutável não podendo sofrer alterações |
| RNF 7 Código único | O código do item deverá ser único |

2.6 REGRAS DE NEGÓCIO

De acordo com OLIVEIRA CEO da BPM (Business Process Management) define regras de negócio da seguinte forma: Como o próprio termo sugere, são regras que servem para definir ou restringir alguma ação nos processos de sua empresa. São declarações que irão descrever como determinadas operações devem ser realizadas e se há algum limite que precisa ser aplicado.

Em linhas gerais, pode se considerar como regra de negócio uma regra própria da empresa que existiria mesmo se não houvesse o sistema

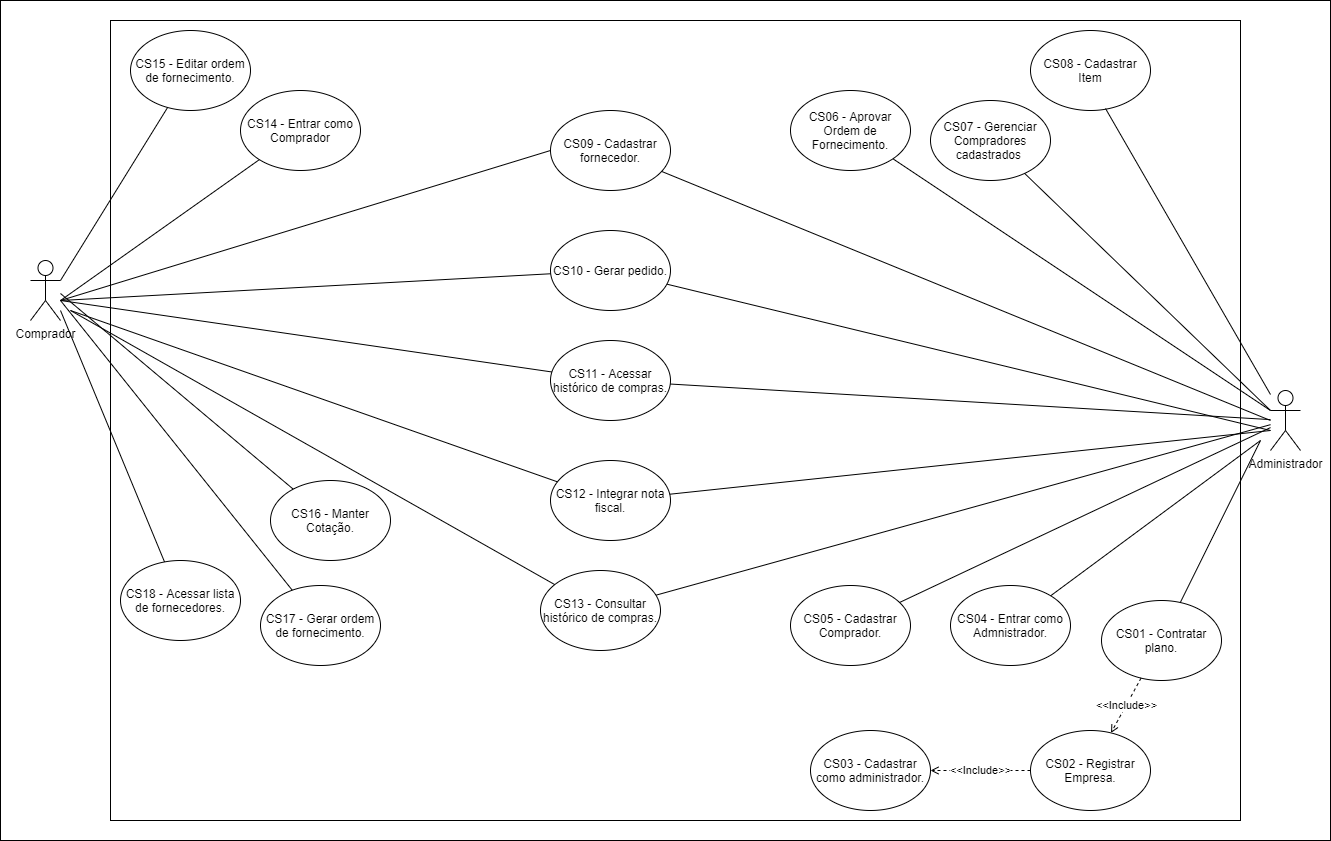
A seguir seguem as regras de negócio do sistema no quadro 4:

|  |
| --- |
| RN1 Todos os fornecedores cadastrados no sistema deverão possuir um CNPJ e o mesmo deve ser valido. |
| RN2 Fornecedores deverão informar o prazo limite de entrega em todas as ordens de fornecimento. |
| RN3 Não serão aceitas notas com valores divergentes das ordens de fornecimento. |
| RN4 O administrador estará pré-cadastrado no sistema. |
| RN5 Cada item do pedido deverá ser negociado com 1 (um) único fornecedor. |

2.7 DIAGRAMA DE CASOS DE USO

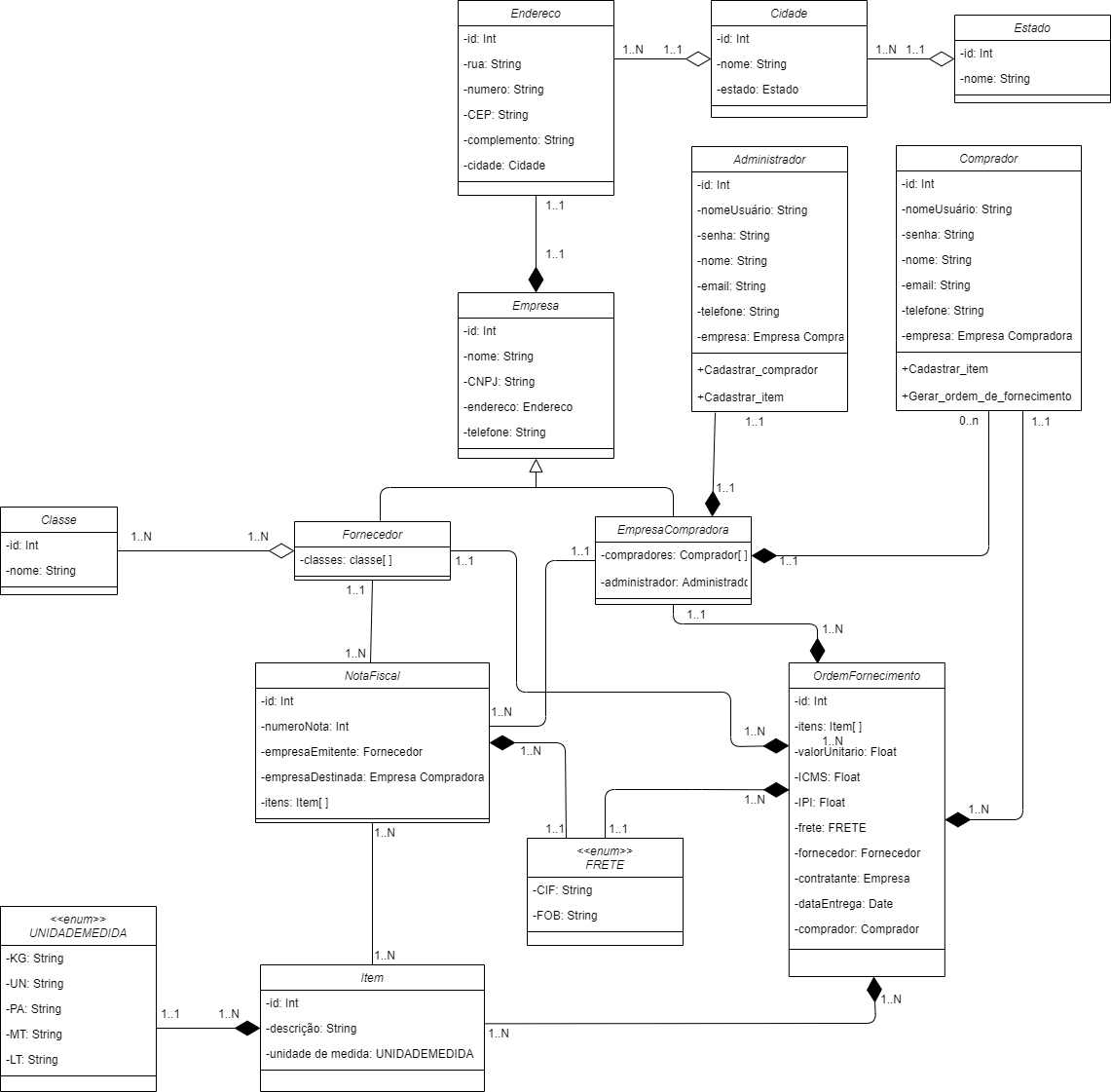
Um diagrama de casos de uso é um diagrama dinâmico ou de comportamento na UML (Linguagem de Modelagem Unificada). O cientista de computação JACOBSON descreve casos de uso da seguinte forma, podemos dizer que um caso de uso é um documento narrativo que descreve a sequência de eventos de um ator que usa um sistema para completar um processo.

Com base nesta descrição desenvolvemos o diagrama de casos de uso do projeto tentando incluir todas os casos de uso do projeto.



2.8 DIAGRAMA DE CLASSES

Segundo BELL no livro Fundamentos básicos de UML: “O propósito do diagrama de classes é mostrar os tipos que estão sendo modelados no sistema e a relações entre eles”. Nessa perspectiva observa-se que é fulcral a elaboração de um diagrama de classes para um sistema afim de documentar corretamente o funcionamento do mesmo e facilitar futuras manutenções no código.

Na figura 4, temos representado a versão completa do diagrama de classes do projeto:

2.9 PROTÓTIPO E DESCRIÇÃO DE CASOS DE USO

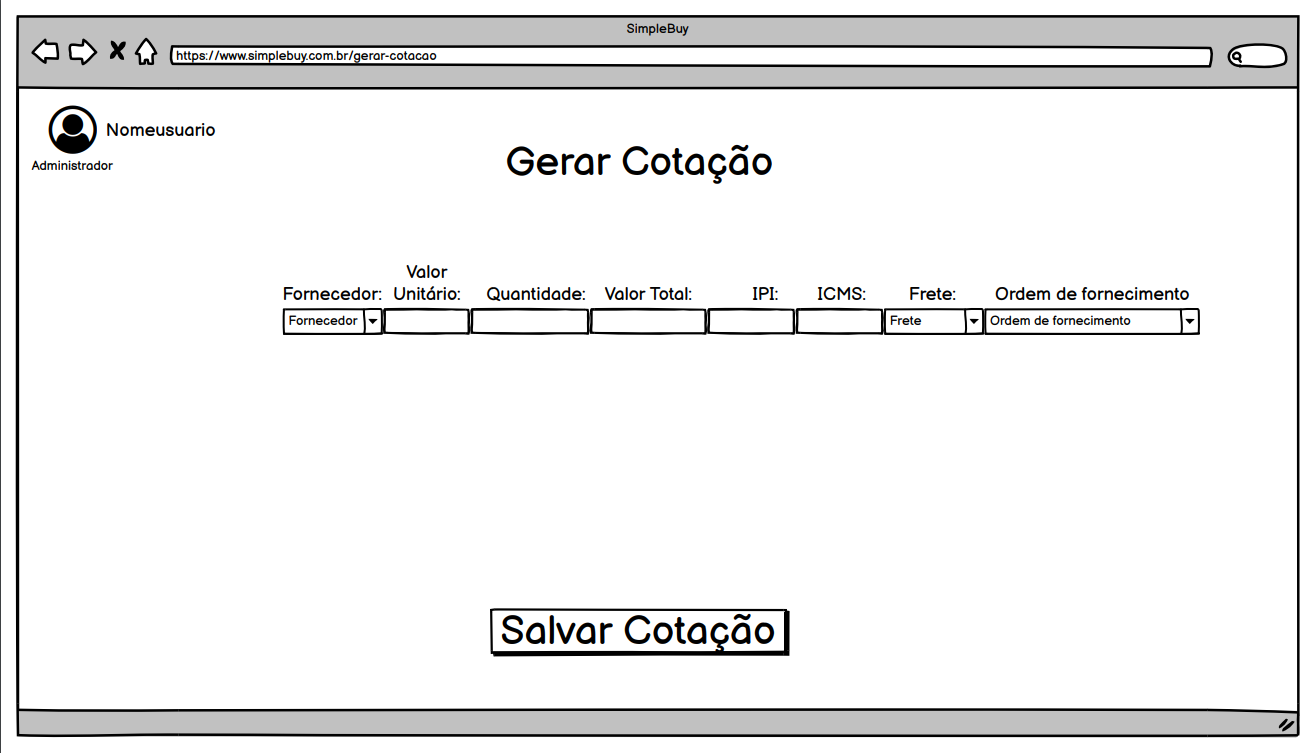
Para o melhor entendimento sobre os casos de uso foi realizado a descrição do caso de uso de ManterCotação, que está representado no quadro 5. Além disso, foi demonstrado o protótipo inicial da tela de Manter Cerveja no item 2.9.1.

|  |  |
| --- | --- |
| Caso de Uso | CS16 – Manter Cotação |
| Referência | Requisito funcional RF03 |
| Descrição | Permitir ao comprador cadastrar, alterar ou excluir uma cotação de um item especifico como seu valor, fornecedor, frete entre outros. |
| Atores | Comprador |
| Pré-Condições | Usuário deve estar logado no sistema e ter o perfil do tipo Comprador. E deve haver um pedido do sistema com o item a ser negociado. |
| Pós – Condições | Cotação salva. |
| Fluxo Básico (Principal) | 1. Comprador seleciona o fornecedor. 2. Comprador informa o valor unitário. 3. Comprador informa a quantidade. 4. Sistema calcula o valor total. 5. Comprador informa o IPI (Imposto sobre Produto Industrializado). 6. Comprador informa o ICMS (Imposto sobre Circulação de Mercadorias e Serviços). 7. Comprador seleciona o tipo do frete (CIF ou FOB). 8. Comprador seleciona uma ordem de fornecimento existente (caso haja). 9. Comprador salva cotação. |
| Restrições/Validações | Campos obrigatórios. |
| Fluxo Alternativo | (8) Não existe ordem de fornecimento aberta com o fornecedor. |
|  | A opção de selecionar a ordem de fornecimento estará bloqueada caso não haja uma aberta com o fornecedor desse item. |

FONTE: Os autores (2021).

2.9.1 PROTÓTIPO DA TELA MANTER COTAÇÃO

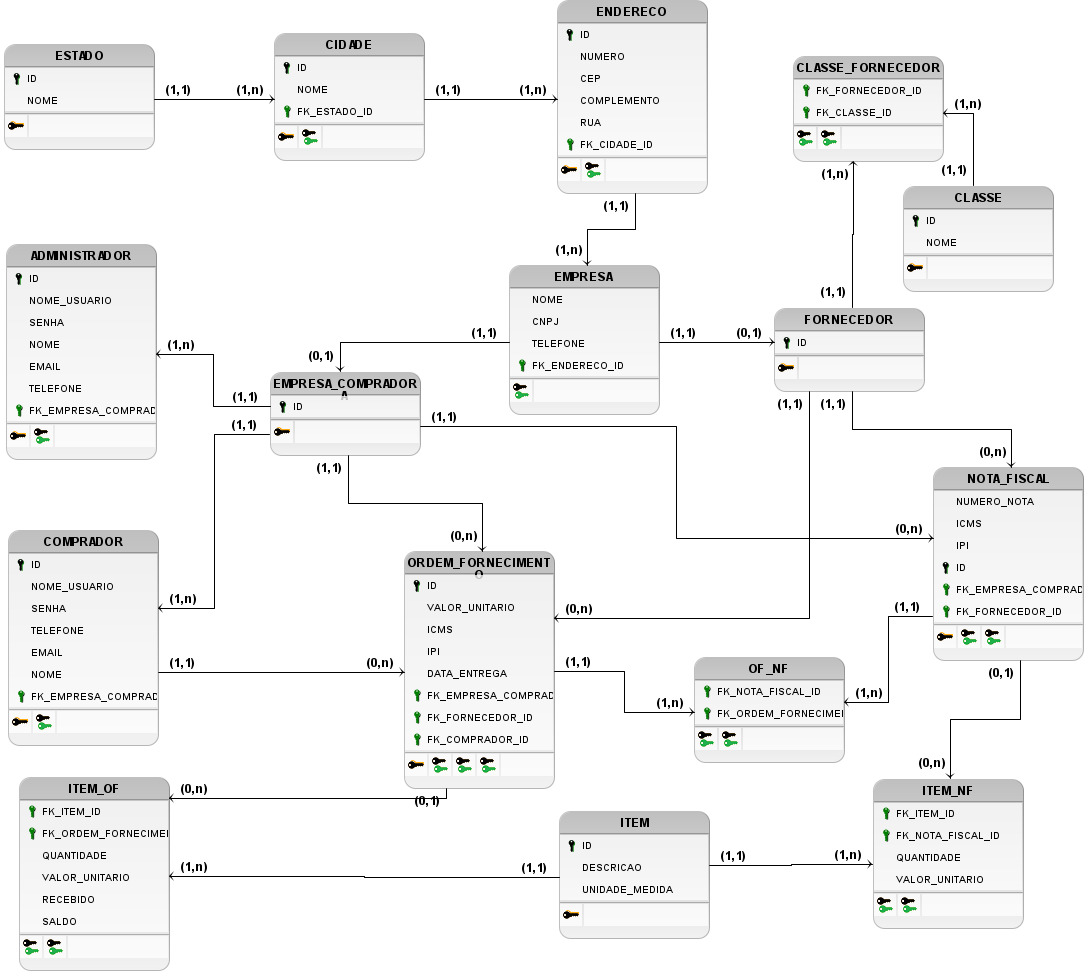
Abaixo na figura 4 está representado o Protótipo inicial da tela de Manter Cerveja, nesta tela o usuário do tipo produtor irá poder cadastrar, alterar ou excluir uma cerveja.



2.10 DIAGRAMA DE ENTIDADE RELACIONAMENTO

O MER (Modelo Entidade-Relacionamento) é uma linguagem que descreve modelos conceituais de banco de dados. Foi criada por cientista da computação Peter Chen em 1976. Por meio dela é possível representar conceitos como entidades e relacionamentos. Em contra partida, originado a partir do MER, o Diagrama Entidade Relacionamento (DER) é a sua representação gráfica e principal ferramenta utilizada nos dias de hoje para representar modelos de banco de dados.

A seguir na figura 5, está representado o Diagrama Entidade Relacionamento do software SimpleBuy.



2.11 TECNOLOGIAS UTILIZADAS

Adicionar aqui as tecnológicas que foram utilizadas no projeto. Nome, versão e breve descrição.

**3. CONCLUSÃO**

Dessa forma, através das metodologias e ferramentas apresentadas nesse artigo, Pycharm para programação do back-end, Vscode para o desenvolvimento do front-end, framework Django para estruturação do servidor, a elaboração dos diagramas de classes e de casos de uso para ter uma visão melhor do escopo do projeto, o uso do padrão de projeto MVC para deixar o código mais desacoplado e facilitar a reutilização. Além de todos os conceitos de orientação a objetos vistos em aula possibilitaram o desenvolvimento e a finalização do projeto com todas as funcionalidades implementadas.

Desta forma o sistema SimpleBuy consegue suprir a demanda que pequenas e medias empresas tem em encontrar softwares para gestão de compra de suprimentos que tenham um preço acessível e facilidade no seu uso. Dispensando assim necessidade de treinamentos especializados para os usuários utilizarem o sistema.

**4. REFERÊNCIAS**

### BELL, DONALD. Fundamentos básicos de UML: O diagrama de classes. 1 ed. IBM Corporation. 19, dezembro de 2016

DASGUPTA**,** SANJOY**.** PAPADIMITRIOU**,** CHRISTOS**,** VAZIRANI**,** UMESH.[***Algorithms***](http://highered.mheducation.com/sites/0073523402/index.html)*.* 1 ed. Editora:  McGraw-Hill Higher Education. 13, setembro de 2006.

GUEDES, MARYLENE. **O que é MVC?** 22, junho de 2020. Disponível em: <https://www.treinaweb.com.br/blog/o-que-e-mvc/>. Acesso em: 11, de novembro de 2020.

### JACOBSON, IVAN: [*Object Oriented Software Engineering: A Use Case Driven Approach*](https://www.amazon.com.br/Object-Oriented-Software-Engineering-Approach/dp/0201544350/ref=sr_1_3?qid=1585572037&refinements=p_27%3AIvar+Jacobson&s=books&sr=1-3). 1 ed. Editora: Addison Wesley. 1, julho de 1992.

### 

### 