Adailson de Almeida dos Santos1; Andre Vinicius Vieira Mildemberg1

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

**RESUMO**

*Realizamos o desenvolvimento de um software mobile de gerenciamento de horas complementares chamado DaHora, o sistema foi desenvolvido inteiramente com android nativo utilizando a linguagem Kotlin.*

*Utilizamos diversas metodologias para planejamento e execução do projeto a fim de tornarmos o desenvolvimento mais objetivo e conseguirmos dessa forma atender todos os requisitos elencados, entre essas metodologias podemos elencar a o levantamento de requisitos funcionais e não funcionais, criação de diagramas de casos de uso, classes e entidade-relacionamento (DER).*

*Palavras chave: Horas, Complementares, Mobile.*

***ABSTRACT***

*We carried out the development of a complementary mobile time management software called DaHora, the system was developed entirely with native android using Kotlin.*

*We use several methodologies for project planning and execution in order to make the development more objective and thus manage to meet the listed requirements, among these methodologies we can choose to collect non-functional requirements, create functional diagrams of use cases, classes and entities -relationship (DER).*

*.*

*Key Words: Hours, Complementary, Mobile.*

**1. INTRODUÇÃO**

Hodiernamente, no setor acadêmico existem duas grandes problemáticas que são a dificuldade no gerenciamento de validação e registro das horas complementares realizado pelos coordenadores além da visualização e acompanhamento dessas horas por parte dos alunos.

Uma vez que a gestão das horas muitas vezes fica centralizada aos coordenadores de curso os quais devem receber comprovantes, validar e registrar as horas de dezenas ou até mesmo centenas de alunos, tal quantidade pode fazer com que aconteçam erros nesse processo.

Ademais, a outra problemática elencada que é a dificuldade de visualização por parte dos alunos é deletéria, uma vez que por ser regulamentada pela lei 9.394/96 ela garante ao aluno o direito de visualização dessas informações, onde o mesmo acaba tendo dificuldades em realizar a administração dessas horas por conta da falta de transparência.

Dessa forma resolvemos criar o *DaHora* um sistema mobile de gestão de horas complementares 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 instituições de ensino tem de ter um sistema especifico e unificado para o gerenciamento de horas complementares de seus alunos onde por meio dele eles possam receber, analisar e aprovar solicitações de horas complementares, além disso os alunos também conseguirão por meio dele ver suas horas de forma clara. Dessa forma iremos tornar o processo de gestão de horas complementares mais eficiente para ambas as partes.

**2. DESENVOLVIMENTO**

Para este projeto nós desenvolvemos um software mobile de gestão de horas complementares desenvolvido em Android nativo utilizando a linguagem kotlin. Ademais, para nos auxiliares no desenvolvimento nos utilizamos a IDE Android Studio.

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), diagrama de classes 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 DaHora. 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:

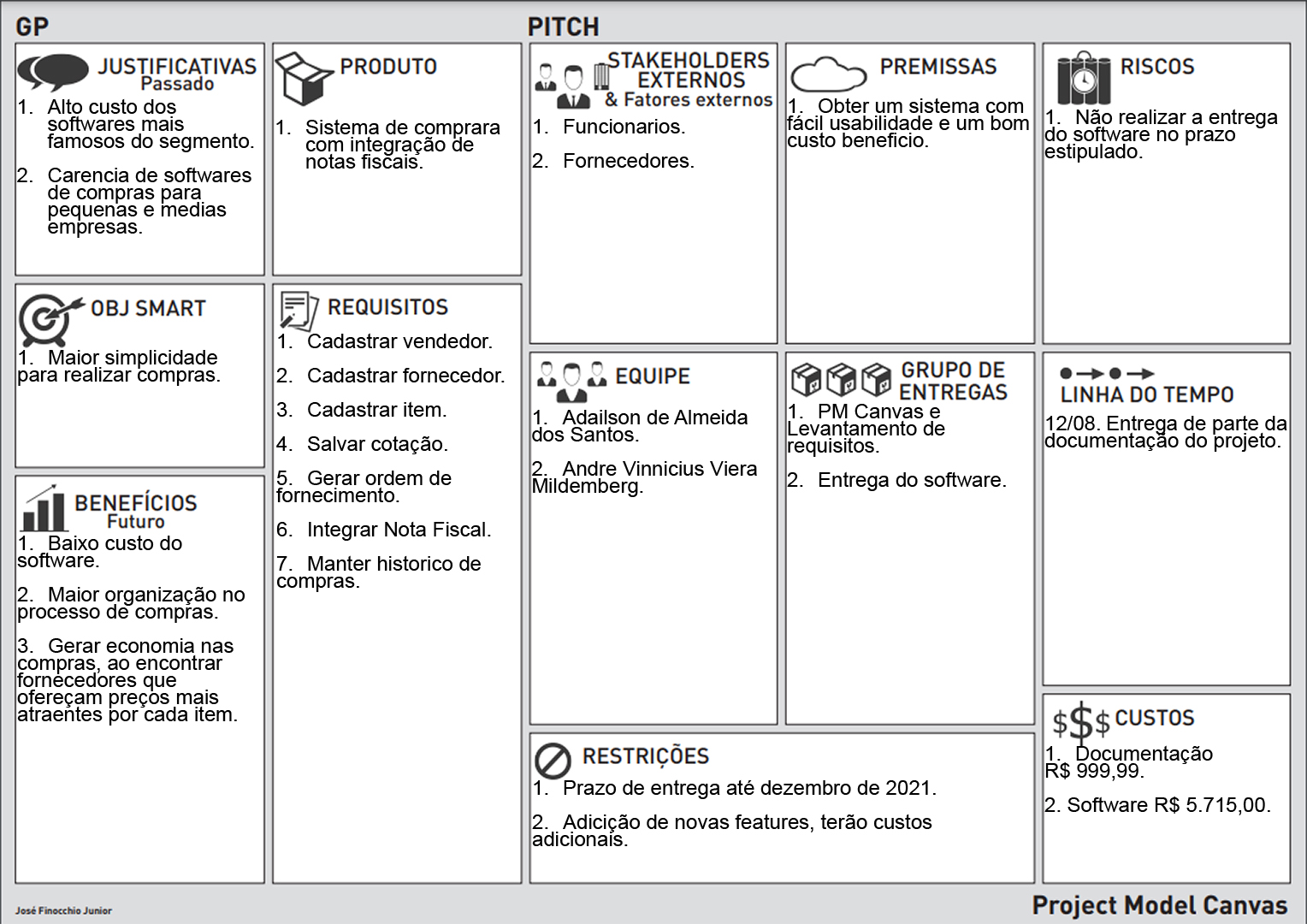
****

FIGURA 1: PM CANVAS  
FONTE: OS AUTORES (2021)

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 e 2.2.2.

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 análise 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;

2.2.2 Cotei

Outro software muito famoso e usado no seguimento de sistemas de compras é o Cotei, o qual é usado por diversas grandes empresas como: Oracle, Amazon, Microsoft, BNDES, Google entre outros.

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, tendo um design agradável e objetivo, além de muitas opções que melhoram a experiencia do usuário.

Em relação aos preços o site não deixa explicito os valores em relação aos planos para vendedores, em relação aos custos para os compradores o site exibe apenas que é a partir de R$ 249,00/mês, pedindo com que o cliente que deseja o serviço solicite uma análise de sua empresa para saber qual será exatamente o custo.

Além de tudo para facilitar a compra o sistema disponibiliza um link para que os fornecedores preencham com as informações sobre o produto desejado, enviando uma tabela posteriormente para o comprador facilitando assim seu trabalho. A seguir uma relação simplifica dos principais pontos positivos (Tópico 2.2.1.1) e negativos (Tópico 2.2.1.2) da empresa.

2.2.2.1 Pontos Positivos

Foram identificados os seguintes pontos positivos no produto Cotei:

* O Cotei recebe as informações dos fornecedores e as integra diretamente ao ERP Mega, garantindo um processo mais seguro e eficiente;
* Bom design e interação no site;
* Grande assistência de suporte ao cliente;
* Possui compatibilidade com smartphones e tablets;

2.2.2.2 Pontos Negativos

Foram identificados os seguintes pontos negativos no produto Mercado Eletrônico:

* Não possuem versão para vendedores;
* Não possui uma própria base de vendedores;

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.

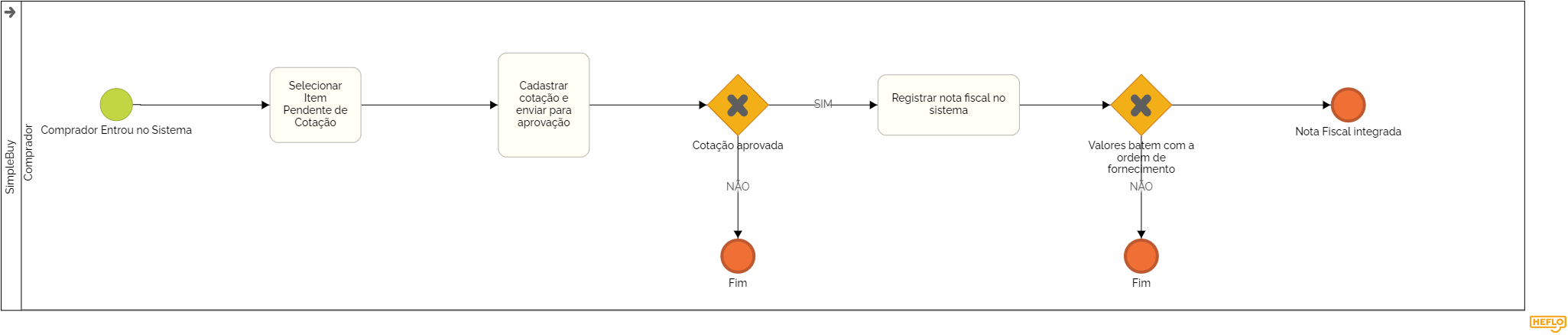


FIGURA 2: MODELO DE PROCESSO DE NEGOCIOS  
FONTE: OS AUTORES (2021)

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 várias 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. |

TABELA 1: REQUISITOS FUNCIONAIS  
FONTE: OS AUTORES (2021)

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 |

TABELA 2: REQUISISTOS NÃO FUNCIONAIS  
FONTE: OS AUTORES (2021)

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 3:

|  |
| --- |
| 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. |

TABELA 3: REGRA DE NEGÓCIOS  
FONTE: OS AUTORES (2021)

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.

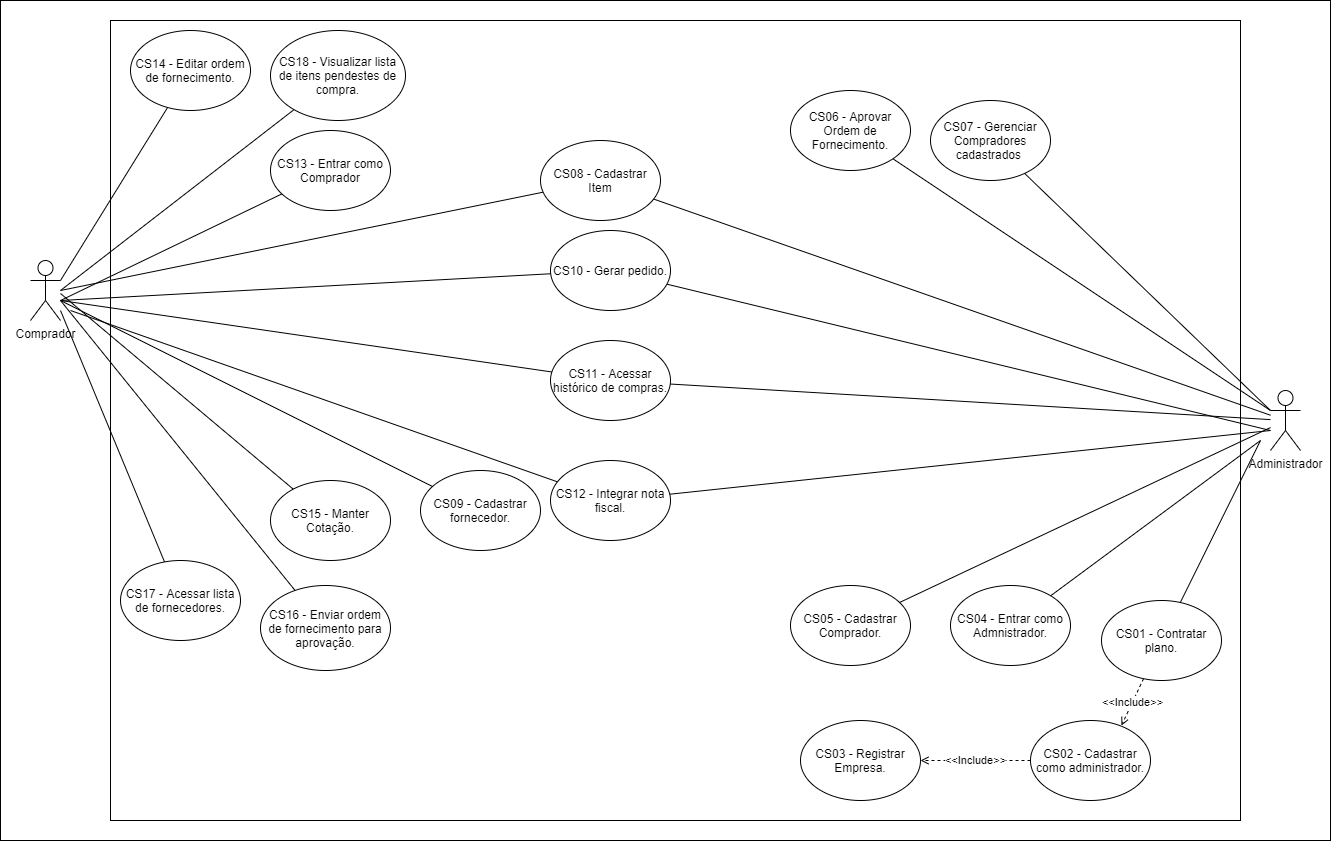


FIGURA 3: CASOS DE USO  
FONTE: OS AUTORES (2021)

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:

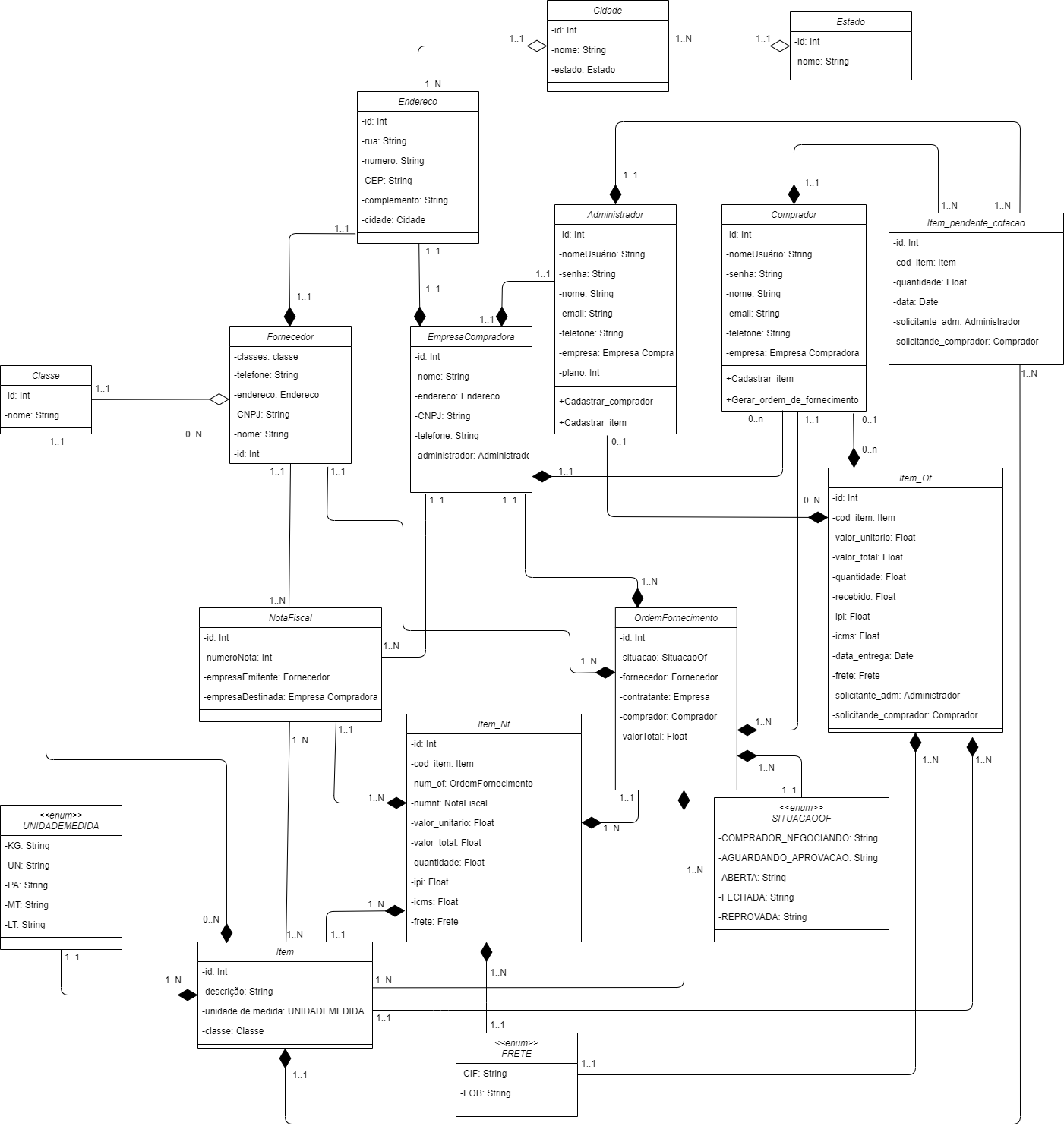


FIGURA 4: DIAGRAMA DE CLASSES  
FONTE: OS AUTORES (2021)

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 4. Além disso, foi demonstrado o protótipo inicial da tela de Manter Cotação 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. |

TABELA 4: CASO DE USO 16  
FONTE: OS AUTORES (2021)

2.9.1 PROTÓTIPO DA TELA MANTER COTAÇÃO

Abaixo na figura 5 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.

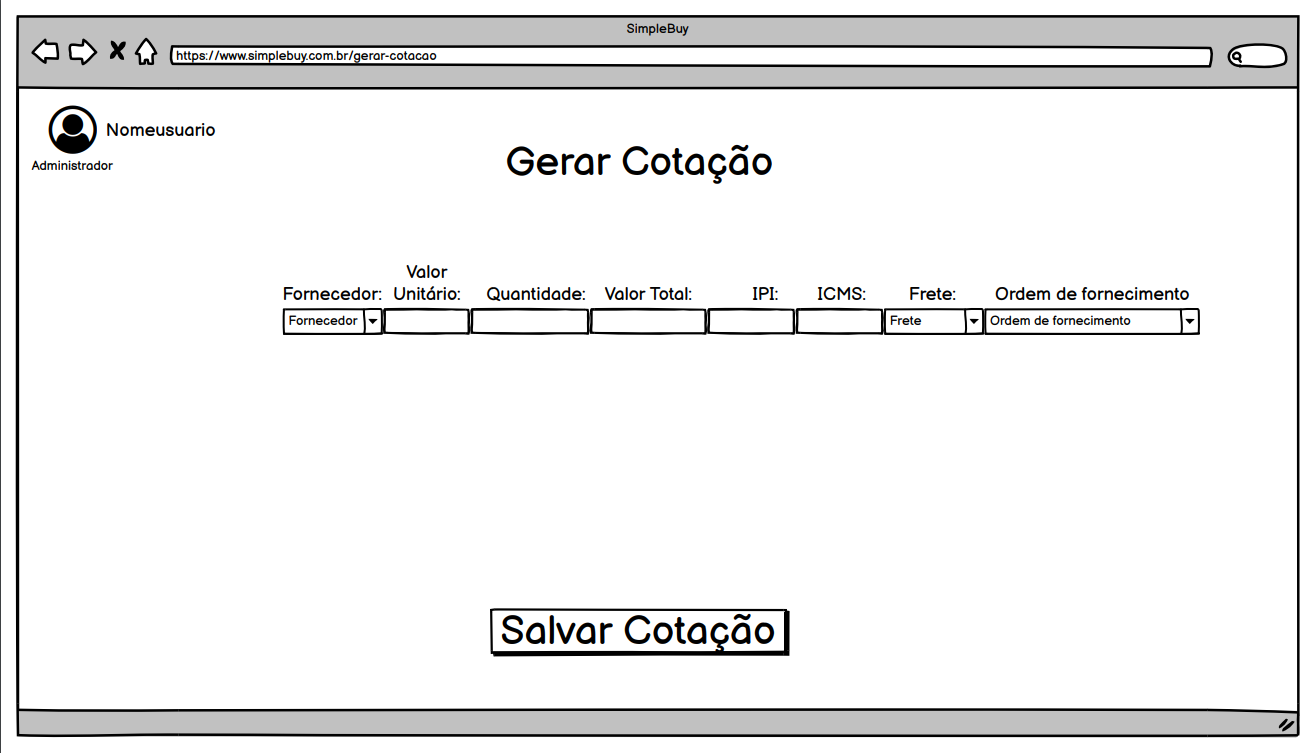


FIGURA 5: PROTÓTIPO DA TELA GERAR COTAÇÃO  
FONTE: OS AUTORES (2021)

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 6, está representado o Diagrama Entidade Relacionamento do software SimpleBuy.

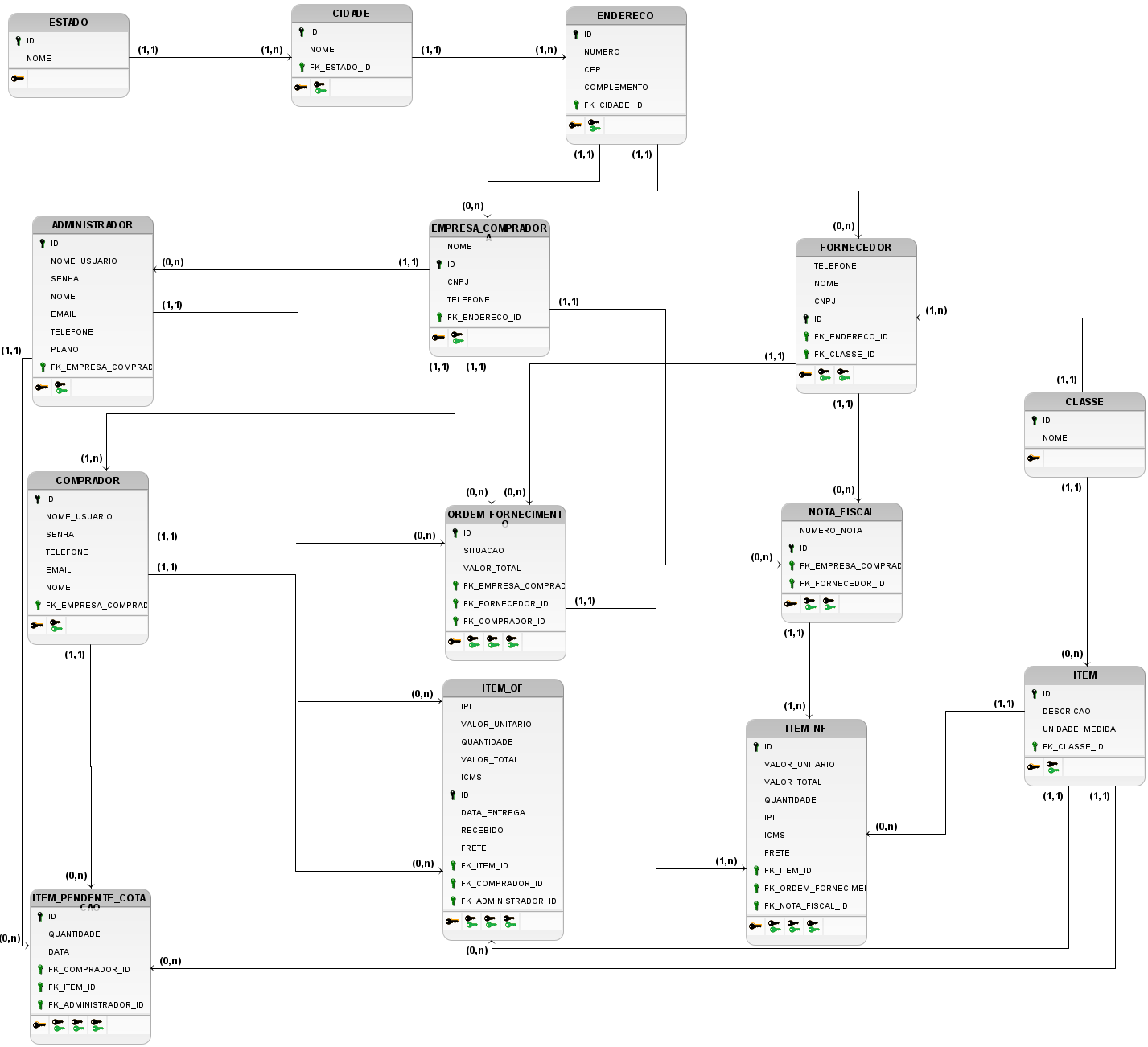


FIGURA 6: DER (DIAGRAMA ENTIDADE RELACIONAMENTO)  
FONTE: OS AUTORES (2021)

2.11 TECNOLOGIAS UTILIZADAS

Para o desenvolvimento do projeto nós fizemos uso de diversas ferramentas a seguir iremos apresenta-las e falar um pouco sobre como o uso delas nos auxiliou no desenvolvimento do projeto.

2.11.1 PyCharm

Para a programação do software nós utilizamos o PyCharm que é uma IDE (Integrated Development Environment ou Ambiente de Desenvolvimento Integrado) que é focada em Python, além que ele fornece análise de código, um depurador gráfico, um testador de unidade integrado, integração com [sistemas de controle de versão](https://en.wikipedia.org/wiki/Revision_control) (VCSes) e suporta desenvolvimento web com [Django](https://en.wikipedia.org/wiki/Django_(web_framework)), bem como [ciência de dados](https://en.wikipedia.org/wiki/Data_science) com [Anaconda](https://en.wikipedia.org/wiki/Anaconda_(Python_distribution)).

2.11.2 Visual Studio Code

É um editor de texto altamente customizável para os mais diversos e variados tipos de linguagens de programação. No desenvolvimento do nosso projeto fizemos uso dessa ferramenta para a criação do front-end da aplicação.

2.11.2 Draw.io

Criado em setembro de 2016, o Draw.io é um software online grátis usado para criar diagramas. Nele nós elaboramos os diagramas de classe e de casos de uso do projeto.

2.11.3 brModelo

No processo de desenvolvimento da modelagem de dados utilizamos o software open source brModelo. Software criado como trabalho de conclusão de curso de especialização de banco de dados pelas universidades UFSC(SC) e UNIVAG (MT), orientadas pelo Prof. Ronaldo dos Santos Mello após se constatar a inexistência de uma ferramenta nacional que pudesse ser utilizada para essa finalidade.

**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.

### 

### 