

CyberStock

1st Natália Maria Rodrigues Queiroz
Engenharia de Software
UnB - FGA
16/0015839

2nd Antonio Rangel Chaves
Engenharia de Software
UnB - FGA
18/0098021

3rd Yago Milagres Passos
Engenharia de Software
UnB - FGA
20/0049879

4th Leonardo Milomes Vitoriano
Engenharia de Software
UnB - FGA
20/1000379

5th Guilherme Meister
Engenharia de Software
UnB - FGA
20/2016382

6th João Paulo Monteiro
Engenharia de Software
UnB - FGA
20/2045141

Abstract—This academic work is a documentation of Cyber Stock software development. Considering, then, the agile development methodology, Scrum, as well as the teacher assistance, it was possible to reach the requirements of a good product.

Index Terms—Academic work, software documentation

I. VISÃO GERAL DO PRODUTO

A. Nome do Produto

Cyber Stock



Fig. 1. Cyber Stock Logo

B. Problemática

É notório analisar que, nos dias atuais, o comércio está se intensificando cada vez mais para acompanhar a demanda exigida pela sociedade consumidora. Nesse contexto, os pequenos, médios e grandes empreendedores necessitam manter um controle e organização clara dos seus produtos em estoque e do setor financeiro, com o objetivo de se manterem no mercado a poderem prosperar. Contudo, essa administração dos produtos e do questão financeira muitas vezes não é tão simples quanto aparenta e não tão viável de colocar todas as informações necessárias, por exemplo, em um caderno, ainda mais que a tecnologia está aqui para nos ajudar.

C. Solução

Tendo em vista a problemática abordada, surgiu assim o **Cyber Stock**. **Cyber Stock** é um software de sistema de gerenciamento de estoque na WEB, no qual o dono do

negócio, gerente, e também os usuários, funcionários, podem cadastrar todos os tipos de produtos e os produtos existentes no estoque. Dessa forma, o gerente possui controle da quantidade em estoque de cada produto e pode comprar ou vender mais produtos, caso seja necessário. Outra funcionalidade do software é acesso à questão financeira, pois nessa parte é possível o gerente verificar o saldo, se está positivo ou negativo. Então, **Cyber Stock** é uma ferramenta que chegou para auxiliar na administração do estoque e do financeiro do seu negócio.

D. Backlog

O Backlog foi realizado utilizando as planilhas do Google Workspace. Ele foi dividido nas seguintes colunas: Épico, Feature, User Story ID, User Story, prioridade e pontuação.

Uma história de usuário (**User Story**) é a representação clara e informal que expressa a necessidade e/ou requisito de um potencial usuário. Também pode ser considerada uma parte de um objetivo final.

A Funcionalidade (**Feature**) é responsável por agrupar um conjunto de histórias de usuário. A Funcionalidade expressa uma função do Produto, da qual contém diversos requisitos funcionais com suas regras e exceções.

O **Épico** (Epic) é uma grande parte do trabalho a ser realizado no Produto. O Épico representa uma parte integral do Produto e deve ser suficiente para ter valor de negócio em sua utilização.

E. Protótipo

A partir da tecnologia Figma, os protótipos de média à alta fidelidade foram realizados. Os protótipos feitos são a base das telas do software **Cyber Stock**. Esses são algumas telas projetadas:

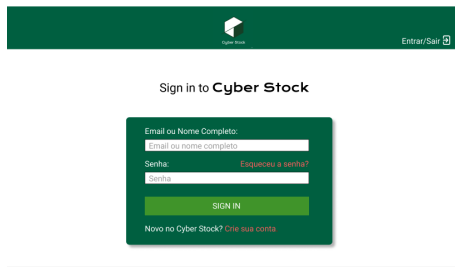


Fig. 2. Protótipo Login

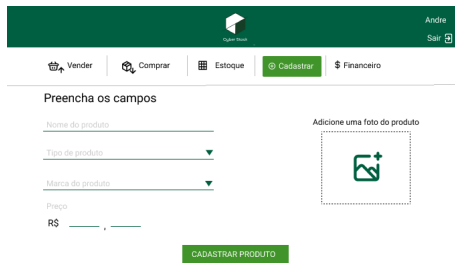


Fig. 3. Protótipo do Cadastro de Produto

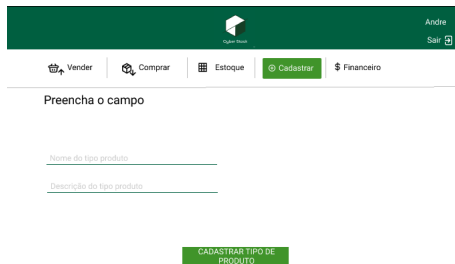


Fig. 4. Protótipo do Cadastro de Tipo de Produto

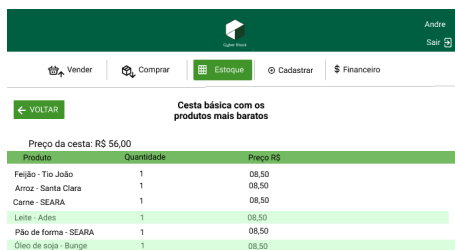


Fig. 5. Protótipo do Estoque

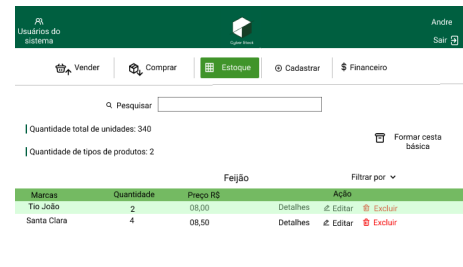


Fig. 6. Protótipo do Estoque

II. ARQUITETURA DO PROJETO

A. Tecnologias Utilizadas

Tecnologia	Função
Python	Linguagem de programação orientada a objetos
Django	Framework de desenvolvimento web (MTV)
SQLite3	Banco de dados
Figma [4]	Fazer protótipos de média à alta fidelidade
Microsoft Teams	Reuniões semanais e comunicação da equipe
Whatsapp	Comunicação da equipe
VS Code	Ambiente de desenvolvimento integrado
Google Workspace	Construção do backlog, cronograma e organização das sprints
Github [5]	Plataforma de hospedagem de código-fonte e controle de versão
Zenhub [6]	Gerenciamento das sprints

B. Representação da Arquitetura (MTV)

A arquitetura do Django é Model Template View.

A definição de **Model** é o arquivo que contém a estrutura lógica do projeto e funciona como um intermediário para manipular dados entre o banco de dados e a View. Dentro do arquivo Model é determinado quais tipos de dados, e como será armazenado dentro do seu banco e, como será exibido quando for requisitado pela View.

O arquivo **View** formata os dados que são vindos do banco através da Model para visualização.

E por último o **Template**, que cuida da parte desta visualização para o usuário final. Ele é como o front-end de sua aplicação. Nesta arquitetura, esta camada fica armazenado os arquivos html, css, javascript extendidos e por conta disso auxilia numa velocidade maior de desenvolvimento e conforme o retorno da aplicação, ele renderiza seus arquivos HTML de sua aplicação no navegador.

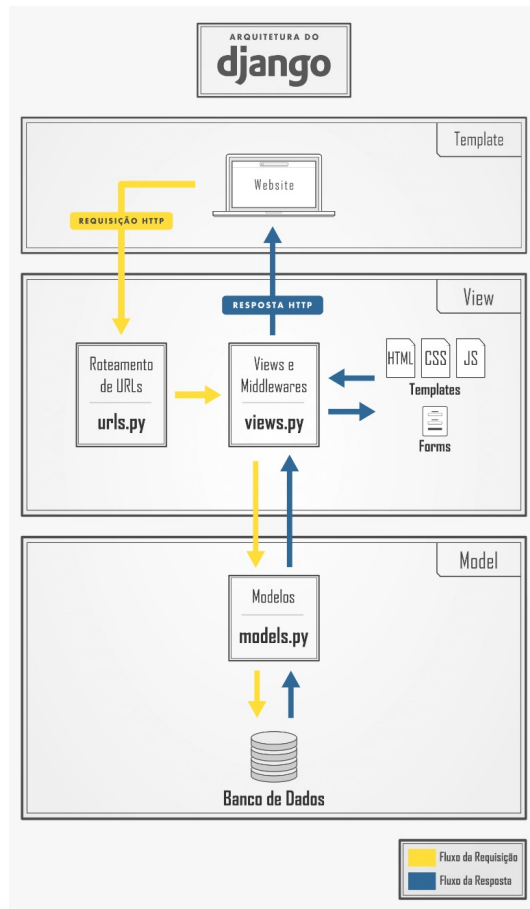


Fig. 7. Arquitetura MTV

III. DESENVOLVIMENTO DO PROJETO

A. Scrum

”**Scrum** é um conjunto de boas práticas empregado no gerenciamento de projetos complexos, em que não se conhece todas as etapas ou necessidades. Focado nos membros da equipe, o Scrum torna os processos mais simples e claros, pois mantém registros visíveis sobre o andamento de todas as etapas.” [7]

No nosso projeto optamos pela rotatividade do **Scrum Master** e do **Project Owner(PO)**, ou seja, a cada duas semanas houve uma mudança nas ”lideranças” do projeto. A função de um Scrum Master é ser responsável por defender o projeto, fornecendo orientação à equipe e ao PO para garantir que todas as práticas sejam seguidas. Ademais, aconteciam reuniões semanais para decidirmos quais histórias fariamos na nossa Sprint da semana. Quando decidido anotávamos em nosso **Zenhub**, que é, de forma crua, um gerenciador de sprints.

Além do mais, ao decidirmos nossas histórias semanais nosso scrum master, se baseando em nosso quadro de conhecimentos, que falaremos mais dela na nossa seção III.C, faz o pareamento semanal.

Sendo assim, uma das práticas utilizadas no nosso Scrum foi o **Planning Poker**. O Planning Poker, também chamado

de Scrum Poker, é uma técnica ágil usada para que a equipe do projeto possa estimar a dificuldade de uma determinada história, essa dificuldade sendo pontuada com base na escala de Fibonacci (0,1,1,2,3,5,8,13,21,...).

B. Sprints

Ao decidirmos as histórias da semana e portuarmos as histórias no Planning Poker, fazíamos uma retrospectiva da semana que se passava. Nessa retrospectiva analisamos e mostramos quais histórias concluímos, quais tivemos dificuldades durante a semana e, se preciso, re pontuamos as histórias.

C. Quadro de Conhecimentos

No final da reunião atualizávamos nossos **quadro de conhecimentos**, que, de forma crua, é uma tabela no Google Sheets com nosso nível de conhecimento em cada área, podendo variar de 1 à 5. Essas áreas são:

- 1) Python;
- 2) Django;
- 3) HTML/CSS;
- 4) JavaScript;
- 5) Git/Github;
- 6) Scrum.

A classificação do conhecimento foi a partir de emojis e cores, na ordem de conheço muito pouco até conheço muito sobre, temos:

- vermelho - sei muito pouco
- laranja - sei pouco
- amarelo - sei mais ou menos
- azul - sei bem
- verde - sei muito bem

Esta foi a evolução do quadro de conhecimento da equipe desde a Sprint 1 até a Sprint 9:

	Python	Django	HTML/CSS	JS	Git/GitHub	Scrum
Antônio	🟡	🟡	🟡	🟡	🟡	🟡
Natália	🟡	🟡	🟡	🟡	🟡	🟡
João Paulo	🟡	🟡	🟡	🟡	🟡	🟡
Leonardo	🟡	🟡	🟡	🟡	🟡	🟡
Yago	🟡	🟡	🟡	🟡	🟡	🟡
Guilherme	🟡	🟡	🟡	🟡	🟡	🟡

Fig. 8. Sprint 1

	Python	Django	HTML/CSS	JS	Git/GitHub	Scrum
Antônio	🟡	🟡	🟡	🟡	🟡	🟡
Natália	🟡	🟡	🟡	🟡	🟡	🟡
João Paulo	🟡	🟡	🟡	🟡	🟡	🟡
Leonardo	🟡	🟡	🟡	🟡	🟡	🟡
Yago	🟡	🟡	🟡	🟡	🟡	🟡
Guilherme	🟡	🟡	🟡	🟡	🟡	🟡

Fig. 9. Sprint 9

Além disso, também tínhamos um quadro onde escrevíamos os pontos positivos, negativos e o que poderíamos melhorar para a próxima semana.

D. Lições aprendidas

Nessa sessão, estão detalhados os relatos dos integrantes do grupo 6, em relação ao que foi aprendido e reciclado na matéria de **Desenvolvimento de Software** e a importância do que foi posto em prática para sua formação acadêmica.

Antônio Rangel, 5º semestre: "Aprendi sobre Django, arquitetura MTV e exercitei a criação de páginas com HTML e CSS."

Guilherme Meister, aluno do 2º semestre: "Tive o meu primeiro contato com outra linguagem que não fosse C, além de ter aprendido bastante sobre Django, Git/GitHub, ZenHub, Figma, e muito dessa minha evolução devo aos meus companheiros de equipe, que foram sempre muito prestativos."

Leonardo Milomes, aluno do 3º semestre: "Foi o meu primeiro contato com uma framework de desenvolvimento para web, Django, assim como a metodologia ágil Scrum. Então tive a oportunidade de entender um pouco sobre a importância de frameworks para um engenheiro de software. Aprendi sobre a arquitetura MTV do Django, páginas web com HTML/CSS e os principais fundamentos do Scrum. Ao longo do projeto, obtive familiaridade com outras ferramentas, tais como o Git/GitHub, ZenHub, Figma e tenho certeza que será de grande utilidade toda a experiência adquirida para minha graduação."

João Paulo, aluno do 2º semestre: "Aprimorei minhas habilidades no front-end (HTML e CSS), tive contato pela primeira vez com uma framework de web development, aprendi os fundamentos do Scrum, Github, Backlog."

Natália Rodrigues, aluna do 10º semestre: "Tive, novamente, contato com o Django, o que me ajudou tanto a relembrar conhecimentos que já tinha quanto a desenvolver novos conhecimentos. Além disso, estar em uma equipe diversificada em termos de competência e poder passar um pouco da minha experiência foi de grande valia para o meu crescimento acadêmico, profissional e pessoal."

Yago Passos, aluno do 3º semestre: "Conheci o Django, aprendi melhor sobre GitHub (como e porquê usar) e desenvolvimento de projeto em grupo (benefícios, dificuldades, scrum...), aprendi mais sobre desenvolvimento front-end (HTML e CSS). Conheci ferramentas novas e usei em outras matérias (Zenhub, figma)"

IV. O SOFTWARE CYBERSTOCK

A. Funcionalidades Implementadas

Quando falamos de controle de estoque, é esperado que o software permita que usuário utilize e desfrute de várias funcionalidades com produtos, como compra e venda de novos produtos, adição de novos tipos de produto e gerenciamento do estoque do lojista. O **Cyber Stock** contém tudo isso e torna fácil o processo de controle de estoque para seus clientes.

Antes de começar a sua rotina de controle de estoque, o usuário consegue **cadastrar** os tipos de produtos que vai comercializar. Tendo um tipo criado, agora ele consegue adicionar os modelos dos produtos na base de dados, de acordo com seu tipo.

Agora que a base de dados possui produtos cadastrados, o usuário pode realizar uma **compra**. Essa funcionalidade permite que o funcionário selecione um produto cadastrado e a quantidade desejada, ao adicionar ao carrinho e confirmar a compra, os produtos serão adicionados ao estoque do usuário.

Na aba de **estoque**, o usuário consegue listar, editar e excluir os tipos de produtos cadastrados. Na listagem de um certo tipo de produto, vemos os produtos cadastrados do tipo escolhido, aqui, é possível visualizar ou editar os detalhes de cada produto, assim como excluir o mesmo.

Por último, o usuário tem a possibilidade de **vender** algum produto que possui quantidade em seu estoque. Basta escolher o produto e a sua quantidade, e confirmar a venda. Caso a quantidade escolhida seja maior que a quantidade em estoque, o programa irá avisar que a venda não foi feita.

B. Telas

As diversas funcionalidades do sistema se tornam acessíveis para o usuário através das nossas telas. A partir das telas de login, chegamos a tela inicial e nela temos a possibilidade de navegar pelas telas das funcionalidades mais específicas. Abaixo estarão algumas telas principais do projeto.

A tela de **Vender**, tem uma tabela que mostra os itens no carrinho, uma ComboBox para selecionar o produto a ser vendido e botões de adicionar e concluir.

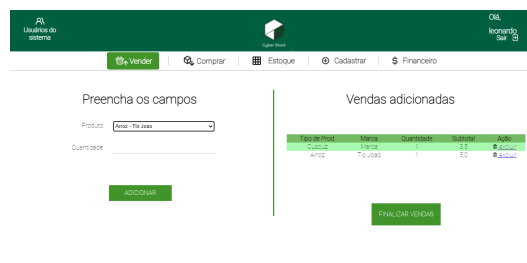


Fig. 10. Tela de Vender

Preencha os campos

Produto:

Quantidade:

Carrinho de compras

Tipo de Produto	Maria	Quantidade	Sustentável	Ação
camiseta	azul	01	R\$ 0,00	<input type="button" value="Remover"/>

A tela de **Estoque** tem uma tabela com todos os tipos de produto cadastrados, bem como os botoes de listar os produtos de cada tipo, editar e excluir.

[illegible]

Por fim, a tela de **Cadastrar**, que se divide em cadastro de produto ou cadastro de tipo de produto.

Fig. 14. Tela de cadastrar produto

Nós, autores do projeto CyberStock, grupo 6 da matéria de Desenvolvimento de Software, 01/2021 da Faculdade do Gama - UnB, devemos agradecimentos ao professor Sergio Antonio Andrade De Freitas pelo acompanhamento do projeto, assim como as aulas ministradas.

- [1] Documentação do Django, disponível em: <https://docs.djangoproject.com/pt-br/3.2/>.
- [2] Django girls, disponível em: <https://djangogirls.org/pt-br/>.
- [3] Curso gratuito de Django na Udemy, disponível em: <https://www.udemy.com/course/python-3-na-web-com-django-basico-intermediario/>
- [4] Figma, site onde podemos fazer protótipos para as nossas páginas, disponível em: <https://www.figma.com/>.
- [5] Github, plataforma de hospedagem de código-fonte e controle de versão, disponível em: <https://github.com/>.
- [6] Zenhub, auxílio na visualização e gerenciamento do projeto, disponível em: <https://www.zenhub.com>.
- [7] <https://fia.com.br/blog/scrum/>
- [8] <https://odonodoproduto.com/epic-feature-and-story-epico-funcionalidade-e-historia/>

Protótipo do projeto: <https://www.figma.com/file/OZh6G6XUhW5EjsxyetaSQU/Prototipo?node-id=41%3A1332>

Acesso à ferramenta: <https://cyber-stock.herokuapp.com/>
 Contas para acesso:

Admin

E-mail: admin@admin.com

Senha: adminadmin

Gerente

E-mail: gerente@gerente.com

Senha: gerger