

TheMarket

Luan D'Amato – 31817051

Lucas Marra Rebello – 31851606

Nasser Bazaan – 31906168

Tarik Vieira Ghazzaoui – 31841082

Sumário

1. Introdução.....	3
2. Descrição do projeto	3
3. Requisitos.....	3
4. Wireframes	4
5. Planejamento.....	5
6. Diagrama de domínio	6
7. Descrição da arquitetura	7
8. Diagrama de pacotes de componentes	8
9. Link do controle de versão	8
10. Link do vídeo	8
11. Breve explicação da Infraestrutura e da Gerenciamento de Configuração	9
13. Passos para execução e utilização.....	10

1. Introdução

O TheMarket é um site que foi criado com o intuito de facilitar a compra/venda de objetos, fazendo com que os compradores e vendedores se conectem através de uma plataforma on-line na qual têm a possibilidade de conversar entre si.

2. Descrição do projeto

Nosso projeto tem como intuito principal, facilitar a compra/venda de objetos através da seguinte ideia: facilitar a comunicação entre o comprador e o vendedor.

Quando o usuário começa a utilizar nossa plataforma ele precisa criar um cadastro com um e-mail válido. Dentro da plataforma o usuário pode anunciar um objeto ou procurar anúncios já existentes.

A plataforma tem um sistema de mensagens permitindo que ele se conecte com outros usuários através de mensagens privadas ou avaliações de outros usuários que já compraram o produto anunciado.

Nós utilizamos pacote socket para a troca de mensagens entre os usuários, o banco de dados MySQL, no front-end utilizamos HTML; CSS; Java Script e para o back-end utilizamos Node JS.

3. Requisitos

- Validar e-mail

Ao fazer o cadastro o sistema enviará um e-mail ao usuário, que deverá confirmar seu e-mail.

- Editar perfil

O usuário poderá alterar as informações de seu perfil .

- Abrir loja

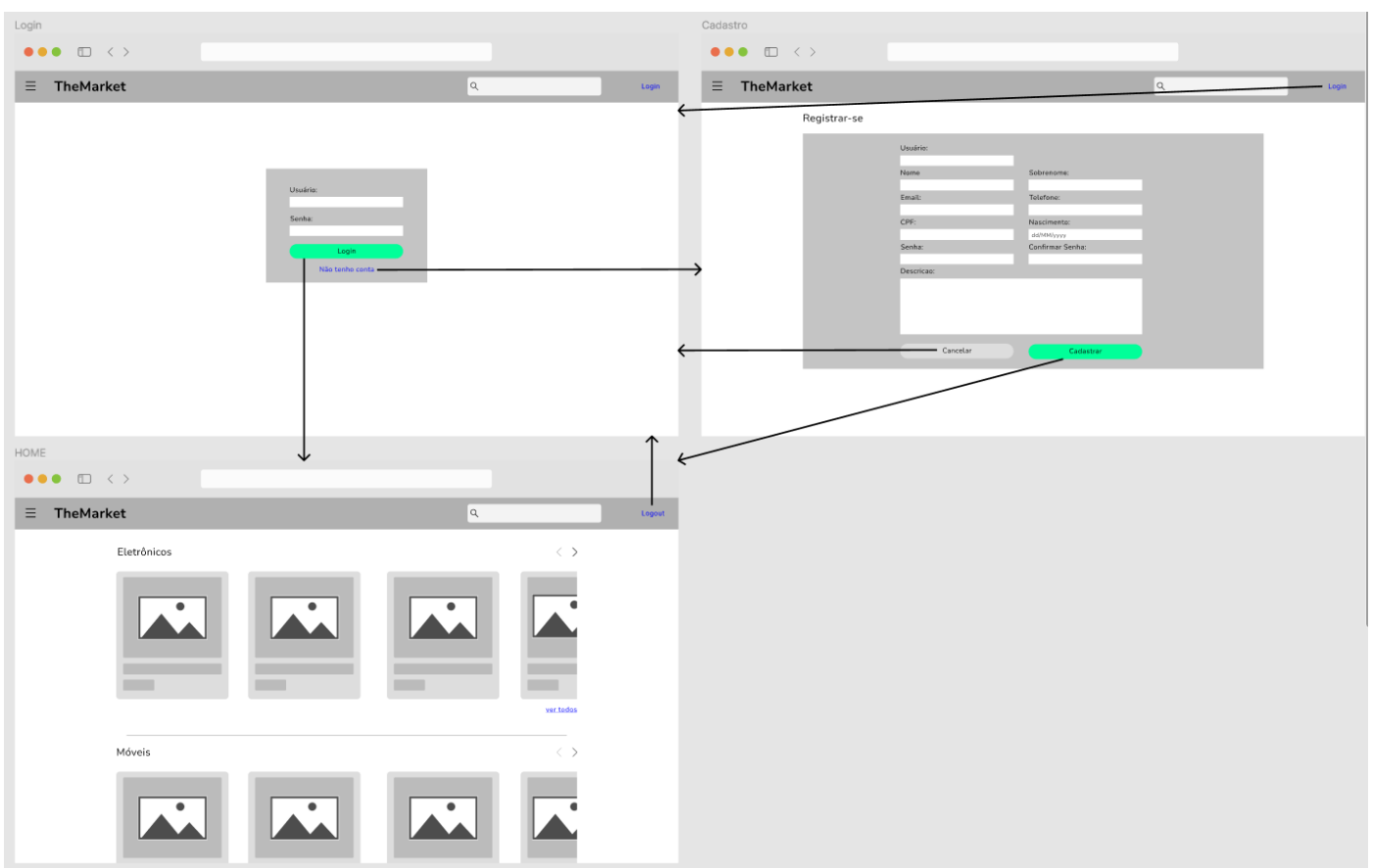
O usuário poderá criar uma loja, caso exista CNPJ regular para ela.

- Anunciar item

Os usuários e lojas poderão anunciar itens para a venda.

- Comprar item
Os usuários poderão navegar pelos itens anunciados no site e comprar os que forem de seu interesse.
- Enviar mensagem
A plataforma conta com um chat para que anunciante e comprador possam se comunicar.
- Avaliar compra
Após receber o item, o comprador poderá avaliar seu pedido.
- Cadastrar forma de pagamento
A plataforma terá opções de pagamentos, sendo possível cadastrá-las.

4. Wireframes



Link para print maior

<https://gyazo.com/8c7bce13b4b0c45a67e3aefc6ceff65d>

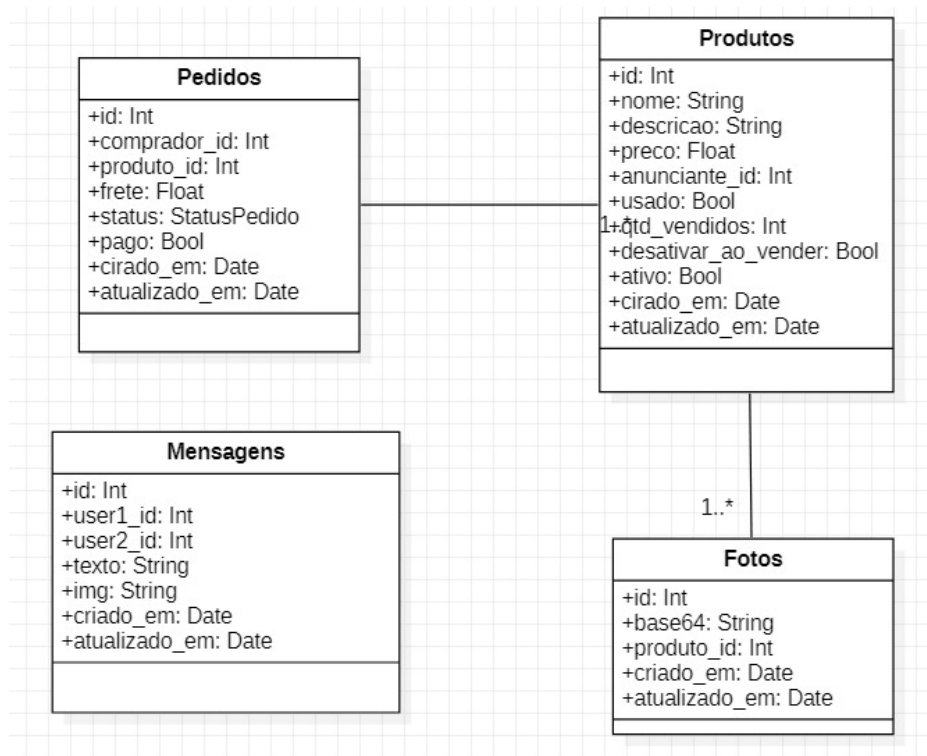
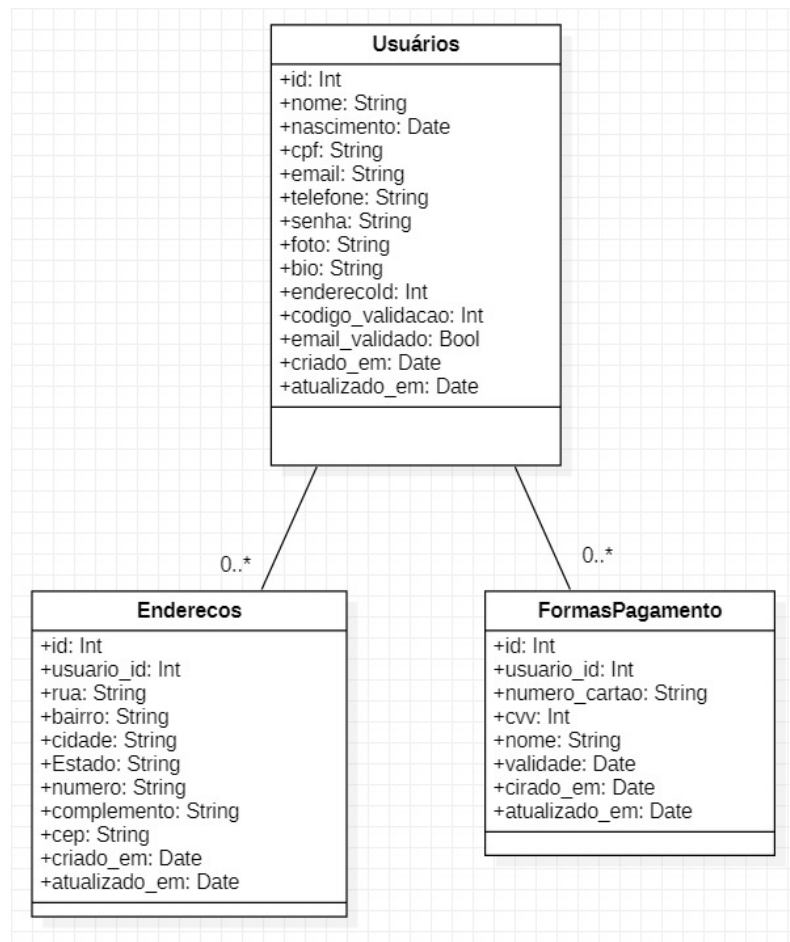
Link do projeto no Figma

<https://www.figma.com/file/XewTOZ61FEHOYJbAFwzLcO/Untitled?node-id=0%3A1>

5. Planejamento

TAREFA	ATRIBUÍDO PARA	PROGRESSO	INÍCIO	TÉRMINO	DEPENDÊNCIA
Título Fase 1		início			
T1 - Tarefa 1	Modelar banco de dados		30/8/21	3/9/21	-
T2 - Tarefa 2	definir cores, temas e estilos de componentes		30/8/21	2/9/21	-
T3 - Tarefa 3	contruir prototipo navegavel da plataforma		2/9/21	5/9/21	T2
E1 - Entregável 1	protptipo			6/9/21	T3
Título Fase 2		implementacao inicial			
Tarefa 1	telas de cadastro		6/9/21	8/9/21	
Tarefa 2	tela de login		8/9/21	8/9/21	
Tarefa 3	home		9/9/21	10/9/21	
Entregável 2	cadastro e login			11/9/21	
Título Fase 3		implementacao de produtos			
Tarefa 1	tela para publicar produtos		13/9/21	14/9/21	
Tarefa 2	tela para visulizar produto		15/9/21	15/9/21	t1
Tarefa 3	tela para editar produto publicado		16/9/21	17/9/21	t2
Tarefa 4	confirmacao de compra de produto		20/9/21	20/9/21	t1
Tarefa 5	testes produtos		21/9/21	23/9/21	t1, t2, t3, t4
Entregável 3				24/9/21	
Título Fase 4		forma de pagamento, perfil comercial e chat			
Tarefa 1	cadastro forma de pagamento		27/9/21	28/9/21	
Tarefa 2	escolher forma de pagamento		29/9/21	29/9/21	t1
Tarefa 3	cadastrar perfil comercial		30/9/21	2/10/21	
Tarefa 4	editar perfil comercail		4/10/21	5/10/21	t3
Tarefa 5	chat		6/10/21	8/10/21	
Entregável 4				9/10/21	

6. Diagrama de domínio



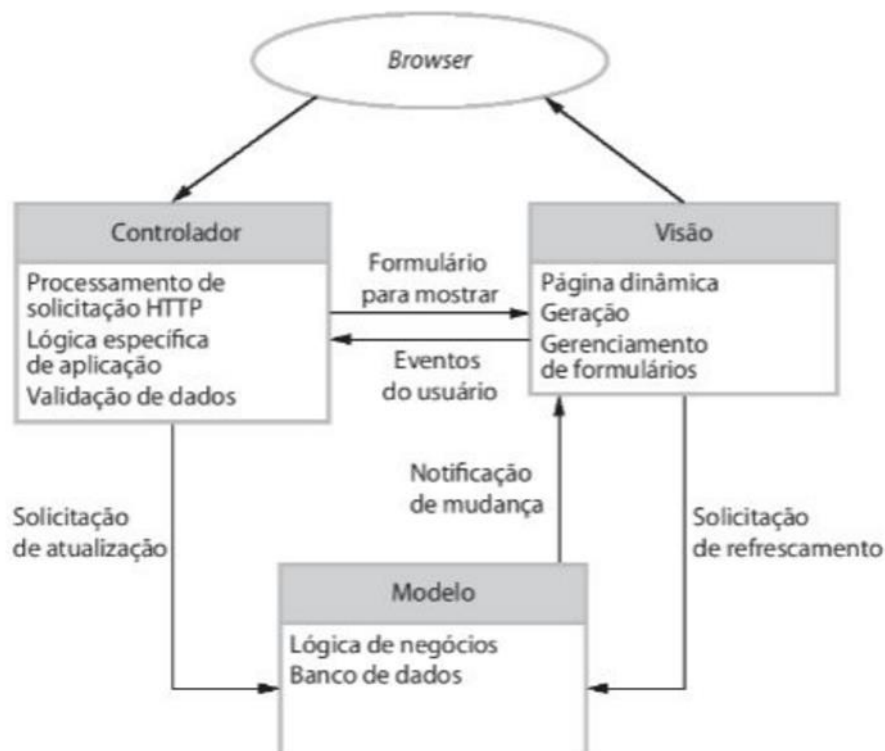
7. Descrição da arquitetura

O padrão de arquitetura escolhido é o MVC (Modelo-Visão-Controlador). Esse padrão de arquitetura foi escolhido pois o MVC é comumente usado em aplicações WEB, que é o caso do nosso software, outro motivo pelo qual esse padrão foi escolhido é que ele separa a apresentação e a interação dos dados no sistema.

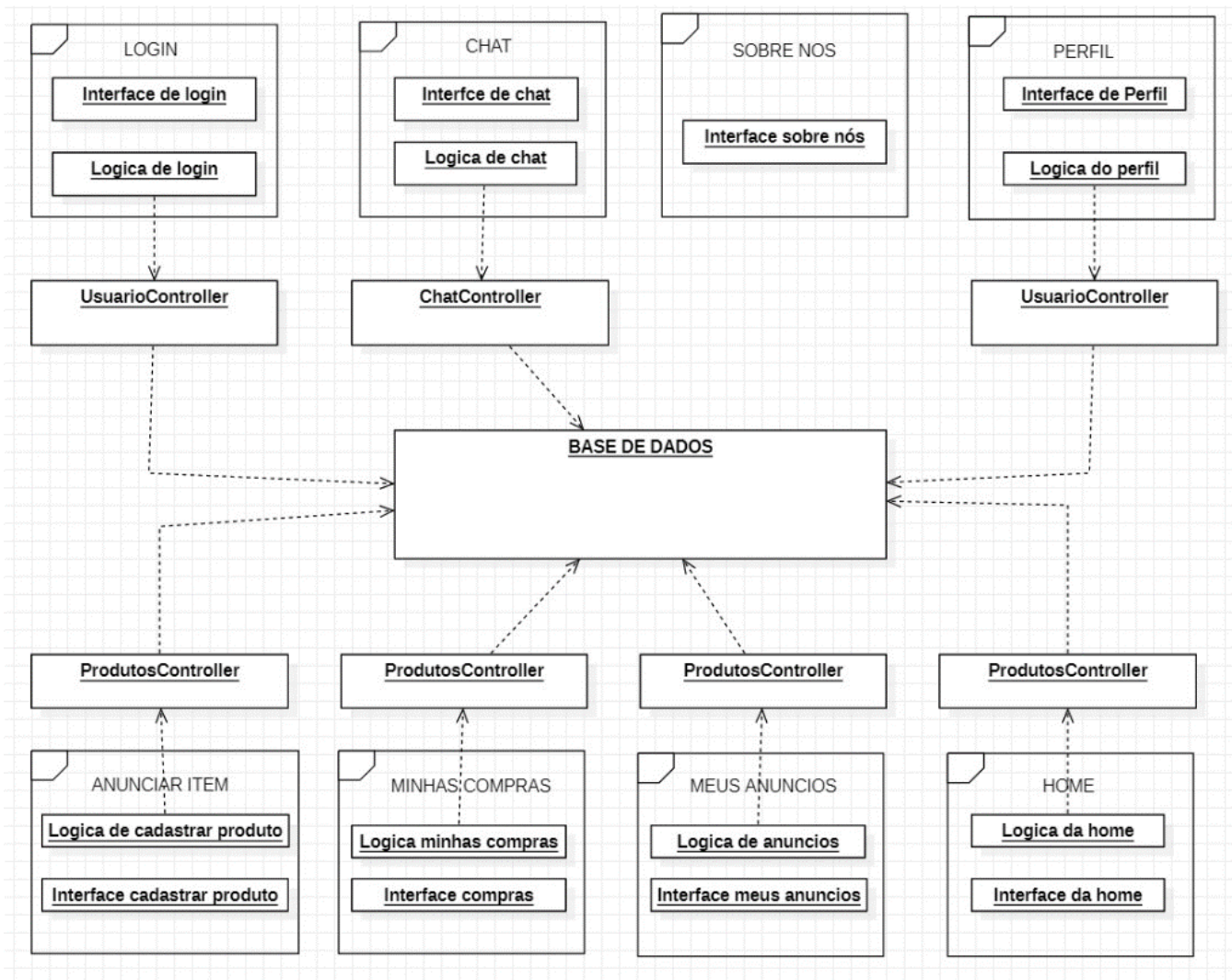
Esse modelo é estruturado em 3 componentes lógicos:

- **Componente modelo:** gerencia os sistemas de dados e as operações associadas a esses dados.
- **Componente visão:** gerencia como os dados são apresentados para o usuário.
- **Componente controlador:** gerencia as interações do usuário como teclas pressionadas e cliques do mouse e passa essas interações para os outros componentes.

Abaixo segue um modelo de aplicação WEB que usa o padrão MVC



8. Diagrama de pacotes de componentes



9. Link do controle de versão

- https://github.com/luandamato/the_market
- <https://the-market-lab.herokuapp.com/>

10. Link do vídeo

- Vídeo 1: Explicação do projeto e planejamento
https://youtu.be/PAOt_9WgVyo
- Vídeo 2: Explicação da arquitetura e da implementação do produto mínimo viável
<https://youtu.be/q5r-gQ-hmeo>

11. Breve explicação da Infraestrutura e da Gerenciamento de Configuração

A infraestrutura está configurada da seguinte forma: 2 VM's que subiram através do Vagrant e após isso o Ansible faz a configuração das máquinas com tudo que é necessário para o funcionamento.

12. Passos para instalação da infraestrutura como código

```
Vagrant.configure("2") do |config|
  config.vm.box = "hashicorp/bionic64"

  config.vm.define :dbserver do |db|
    db.vm.hostname = "db"
    db.vm.network "public_network", ip: "192.168.1.22"

    db.vm.provision "shell",
      inline: "cat /vagrant/configs/id_bionic.pub >>
.ssh/authorized_keys"
  end

  config.vm.define :webserver do |web|
    web.vm.hostname = "web"
    web.vm.network "forwarded port", guest: 80, host: 8081
    web.vm.network "public_network", ip: "192.168.1.24"

    web.vm.provision "shell",
      inline: "cat /vagrant/configs/id_bionic.pub >>
.ssh/authorized_keys"
  end

  config.vm.define :ansible do |ansible|
    ansible.vm.hostname = "ansible"
    ansible.vm.network "public_network", ip: "192.168.1.20"
    ansible.vm.provision "shell",
      inline: "apt-get update && \
        apt-get install -y software-
properties-common && \
        apt-add-repository --yes --update
ppa:ansible/ansible && \
        apt-get install -y ansible"

    ansible.vm.provision "shell",
      inline: "cp /vagrant/configs/id_bionic
/home/vagrant && \
        chmod 600 /home/vagrant/id_bionic && \
        chown vagrant:vagrant
/home/vagrant/id_bionic"

    ansible.vm.provision "shell",
      inline: "ansible-playbook -i
/vagrant/configs/ansible/hosts \
/vagrant/configs/ansible/playbook.yml"
  end
end
```

13. Passos para execução e utilização

Não tivemos êxito na execução e utilização do Vagrant, Ansible e Jenkins.

Segue abaixo prints da execução dos testes

```
Test Suites: 3 passed, 3 total
Tests:       7 passed, 7 total
Snapshots:   0 total
Time:        5.14 s
Ran all test suites.
```

```
PASS __tests__/user.test.js
  ● Console

PASS __tests__/produto.test.js
  ● Console

PASS __tests__/endereco.test.js
  ● Console
```

14. Link do Vídeo

<https://youtu.be/Ec5s6yJ3ucY>