



Colheita Urbana

Por: Ândreo Dias Barros; Txai Mostardeiro Potier; Adão dos Santos Jr.

Professora: Lucinéia Heloisa Thom

Engenharia de software

Descrição do Sistema

- Esse projeto tem por objetivo desenvolver um site para comércio da cooperativa Colheita Urbana, afim de auxiliar na divulgação e nas vendas dos produtos orgânicos por eles produzidos.
- O sistema consta com gerenciamento de usuários e itens, sistema no modelo de carrinho de compras(nesse caso será a cesta de alimentos), sistemas de e pesquisa e exibição dos itens e um mecanismo de pagamento simples. Para possibilitar melhor gerenciamento de vendas aos produtores e facilitar o acesso aos produtos pelos clientes

Diagrama de Casos de uso



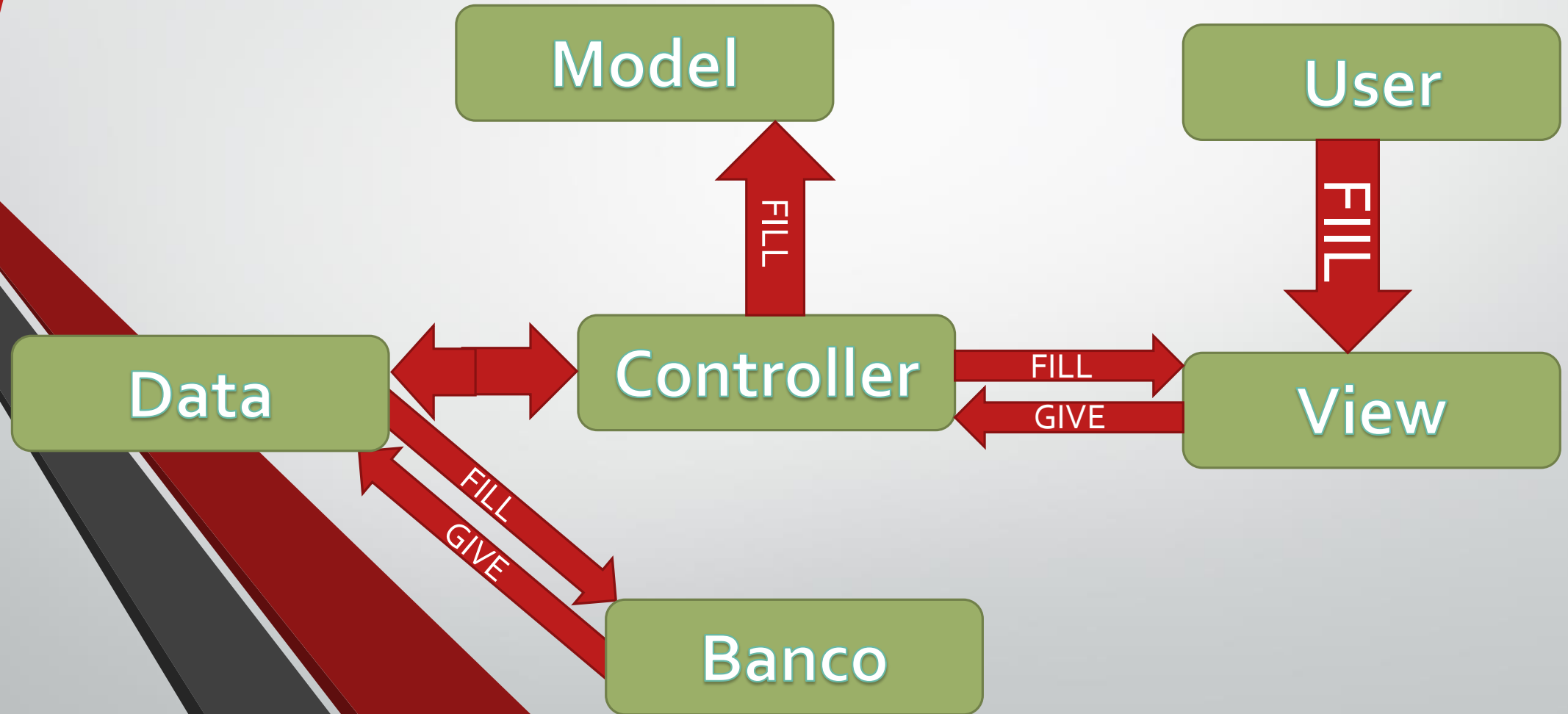
Visão geral dos casos de uso

- Caso de Uso: Adicionar Item
Ator: Administrador
Descrição: Permite ao administrador adicionar itens novos ao estoque.
- Caso de Uso: Mudar senha da conta
Ator: Administrador
Descrição: O administrador pode mudar a senha da própria conta de administrador
- Caso de Uso: Listar itens
Ator: Administrador
Descrição: Permite ao administrador listar todos os itens do estoque para verificação de quantidades.
- Caso de Uso: Remover Itens
Ator: Administrador
Descrição: O administrador pode remover itens que estão disponíveis na lista do estoque.
- Caso de Uso: Editar itens
Ator: Administrador
Descrição: Permite ao administrador editar quaisquer itens que estão presentes no estoque.
- Caso de Uso: Cadastrar cestas
Ator: Administrador
Descrição: O administrador pode cadastrar novos tipos de cestas, a atual lista de cestas.
- Caso de Uso: Listar cestas
Ator: Administrador
Descrição: O administrador pode listar todos os tipos de cestas disponíveis atualmente.
- Caso de Uso: Deletar cestas
Ator: Administrador
Descrição: Permite ao administrador remover tipos de cestas disponíveis no sistema.
- Caso de Uso: Consultar relatórios
Ator: Administrador
Descrição: O administrador pode consultar um relatório, contendo todas as cestas vendidas no site, de todos os usuários.
- Caso de Uso: Consultar usuários
Ator: Administrador
Descrição: Permite ao administrador consultar dados como por exemplo nome, endereço e telefone de todos os usuários cadastrados.
- Caso de Uso: Listar cestas
Ator: Administrador
Descrição: O sistema deve enviar um e-mail ao usuário que solicitou aluguel quando o dono do item aceitou ou recusou seu pedido.
- Caso de Uso: Cadastrar e editar conta
Ator: Usuário comum
Descrição: O usuário comum deve poder se cadastrar no site, e ter poder de alterar os dados cadastrais da própria conta.
- Caso de Uso: Histórico de compra
Ator: Usuário comum
Descrição: Permite ao Usuário comum verificar seu próprio histórico de compras no site.
- Caso de Uso: Restrições de Comida
Ator: Usuário comum
Descrição: Caso o Usuário comum tenha algum tipo de alergia ou restrição nutricional (dieta) é possível restringir a seleção de itens da cesta a apenas os itens da sua dieta.
- Caso de Uso: Selecionar cesta
Ator: Usuário comum
Descrição: O usuário comum pode escolher uma cesta, para futura compra.
- Caso de Uso: Mostrar itens
Ator: Sistema
Descrição: O sistema deve exibir os itens disponíveis para o usuário em uma lista.
- Caso de Uso: Selecionar itens
Ator: Usuário comum
Descrição: Para continuar o processo de compra o Usuário comum deve selecionar itens da lista exibida.
- Caso de Uso: Exibir cestas montadas
Ator: Sistema
Descrição: O sistema deve exibir a cesta montada pelo Usuário comum com os itens por ele escolhidos.
- Caso de Uso: Pagamento
Ator: Usuário comum
Descrição: O usuário efetua a compra, realizando o pagamento feito externamente.

Arquitetura

- Usamos o modelo Model-View-Controller (MVC) com uma classe Extra de acesso ao Database
- Misturou Modelo de camadas com MVC

Model-View-Controle



Exemplo da classe controle

```
<?php
->include("View/RequestView.php");
->include("Data/RequestData.php");
->include("Model/Basket.php");
->include("Model/Item.php");

->class RequestController {
->    ->private $RequestView;
->    ->private $RequestData;

->    ->public function __construct() {
->        ->    ->$this->RequestView = new RequestView();
->        ->    ->$this->RequestData = new RequestData();
->    }

->    ->public function getBasketSelection() {
->        ->    ->$arratBasket = $this->RequestData->getAllBasket();
->        ->    ->$arrayItems = $this->RequestData->getAllItems();

->        ->    ->$returnContent = $this->RequestView->getBasketSelection($arratBasket);
->        ->    ->$returnContent .= $this->RequestView->getItemsSelection($arrayItems);
->        ->    ->return $returnContent;
->    }
->}

?>
```

Requisições Assíncronas

O Sistema utilizará requisições assíncronas para tratar as interações do usuário.

Características:

- + Velocidade
- + Modularidade
- + Segurança
- Possível sobrecarga no servidor
- Requer boa manipulação de sessão

Demais bibliotecas e recursos utilizados

- Banco de dados: MySQL
- Ferramenta de manipulação do Banco de dados: PhpMyAdmin
- Servidor: Apache
- Xampp
- Linguagem de programação principal: PHP
- Linguagem de auxílio UI(FrontEnd) Javascript, biblioteca JQuery
- CSS, HTML5
- Ambiente de desenvolvimento: Notepad++
- Ferramenta de Versionamento: Git(Github servidor de armazenamento dos arquivos)

Desafios e perspectivas

- Um dos grandes desafios do projeto foi a manipulação de banco de dados, já não tínhamos muito conhecimento do assunto.
- Embora a arquitetura tenha sido muito bem escolhida, ela requereu muita atenção na implementação
- No futuro, a falta de um designer na equipe dificultará a implementação de classes de estilização visual