

Relatório

Fundamentos da

Engenharia de Software

Integrantes: Joyce Brum,
Thiago Outeiro,
Gabriel Mattos e
Vitor Millioni

Sumário

1. Time	3
2. Objetivo	4
3. Planejamento Inicial: I-Educar	5
4. Mudanças no planejamento: Merenda	9
4.1. O que?	10
4.2. Por que?	10
4.3. Quem?	10
4.4. Como?	10
4.5. Quando e quanto?	10
5. Desenvolvimento	11
6. Ferramentas	12
7. Conclusão	14
8. Anexo I – Histórias de usuário	15
9. Anexo II – Metáfora do Barco	17
10. Anexo III – Casos de Uso	18
11. Anexo IV – Casos de Teste	34
12. Anexo V – Testes Documentados	35
13. Anexo VI – Reações	39
13.1. Aulas 19 e 21 de Março	39
13.2. Aulas 26 e 28 de Março	39
13.3. Aulas 09 e 11 de Abril	40
13.4. Aulas 14 e 16 de Maio	40
14. Anexo VII – Documentação Sprints	42

1. Time

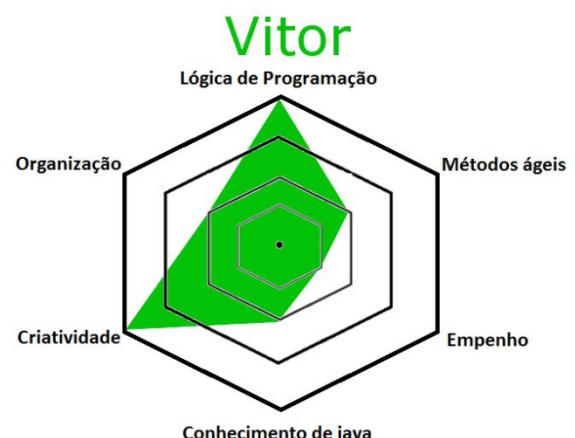
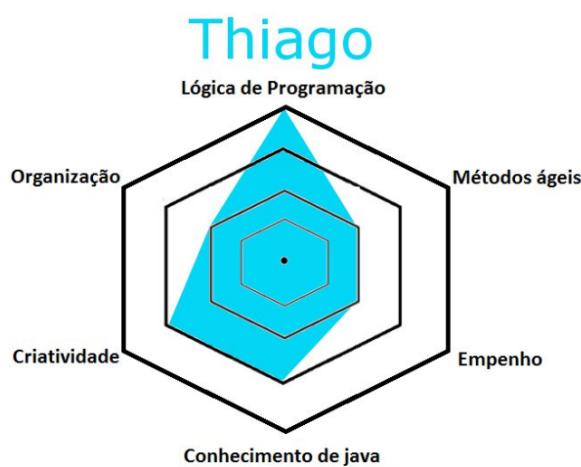
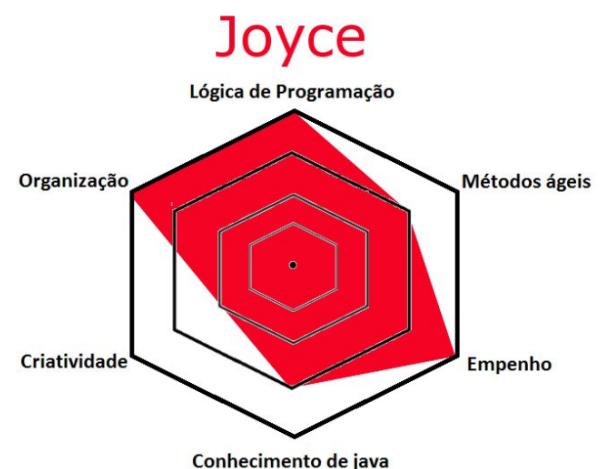
Decidimos dar um trabalho com algum enfoque para cada membro do time. Thiago, Gabriel e Joyce ficaram como desenvolvedores enquanto que Vitor ficou responsável pelo Banco de Dados e pelos testes.

Thiago ficou com um foco maior na interface do projeto, desenvolvendo as telas e planejando o modelo de interface. Gabriel ficou responsável por atualizar a wiki, além de ajudar na programação. A Joyce atuou também como Scrum Master, sendo responsável por coordenar as sprints e acompanhá-las.

Conhecimentos prévios:

Todos os membros do time já possuíam algum conhecimento de java devido à matéria de computação 2, mas algumas bibliotecas necessárias para a aplicação exigiram uma atenção maior de aprendizado.

Sobre os conhecimentos principais para o desenvolvimento do software e aprendizados na disciplina escolhemos os 6 principais: Lógica de programação, Criatividade, Organização, Métodos ágeis, Empenho, Conhecimento de Java.



2. Objetivo

O projeto inicial que nos foi proposto para trabalhar foi o de customização de relatórios para uma plataforma online chamada i-Educar. A fim de nos deixar mais próximos de entender e conseguir corretamente estimar nossos objetivos, requisitos e motivações, fizemos um conjunto de reuniões com um representante do i-Educar e com a prefeitura de Caxias.

Diante dos fatos apresentados, surgiu então uma nova ideia de projeto: Merenda. Tal projeto seria focado a auxiliar a prefeitura de caxias no controle das merendas através de um sistema de geração de relatórios segundo o modelo mostrado.

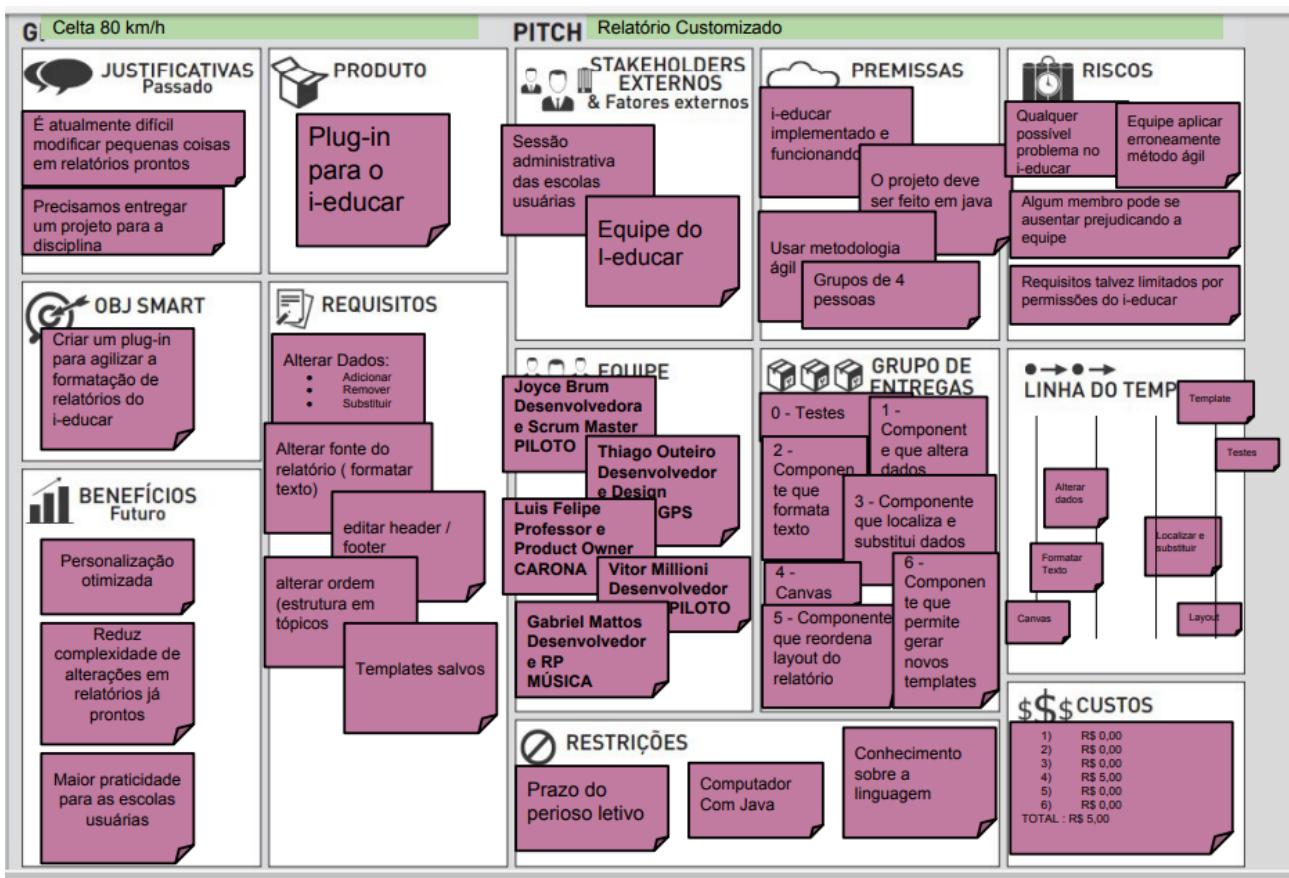
Nosso time escolheu então pela opção da Merenda, mas como boa parte do tempo já havia passado, optamos por não refazer oficialmente o planejamento todo que havia sido feito para a customização de relatórios.

Ao conversar com a representando da prefeitura de caxias sobre os critérios que o software deveria atender, concluiu-se então, que o principal objetivo seria a geração de relatório mensal por escola, com possibilidades de atualização diária dos dados, com os dados da escola, o cardápio do mês e a lista de itens usados no mesmo.

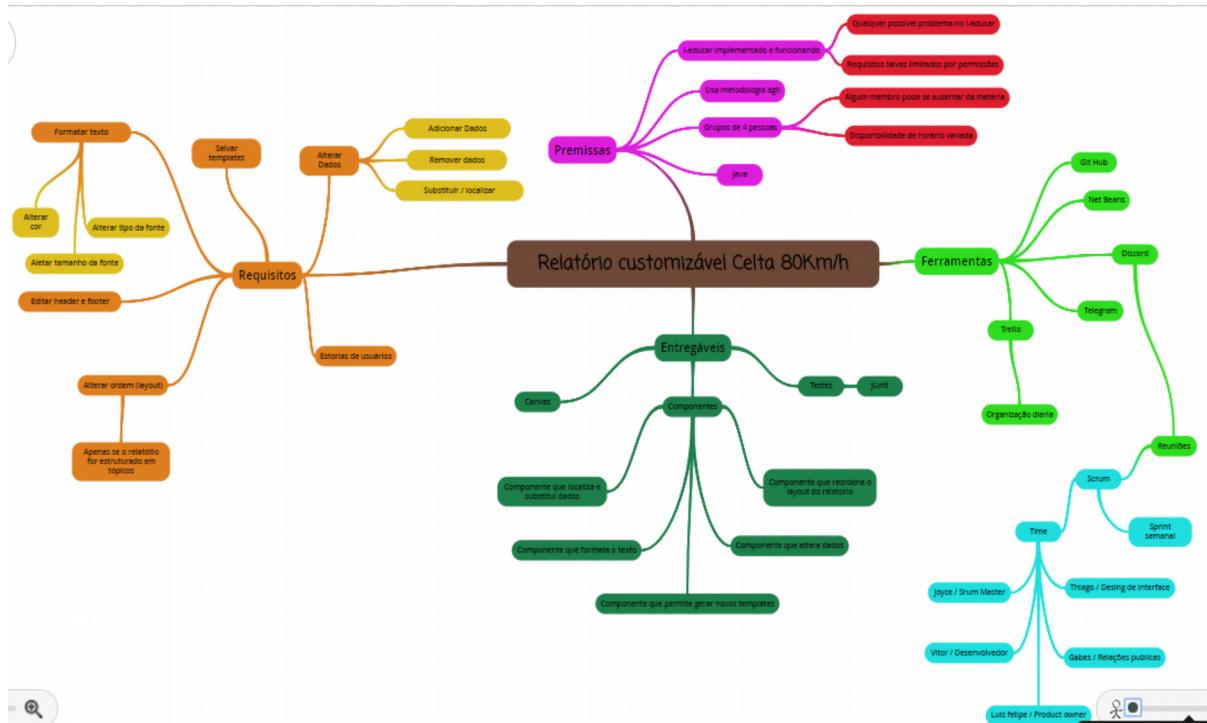
3. Planejamento Inicial: I-Educar

A princípio, o projeto era sobre customizar relatórios para o i-Educar. Para isto, fizemos um canvas a fim de planejar a base do projeto. Neste canvas, respondemos às 6 perguntas fundamentais para o desenvolvimento e planejamento de um projeto: “Por quê?”, “O que?”, “Quem?”, “Como?”, “Quando?” e “Quanto?”, separado em colunas e poucos subtópicos a fim de tornar o processo de idealização e planejamento menos maçante e mais iterativo.

Tendo em vista este objetivo, geramos o canvas abaixo:

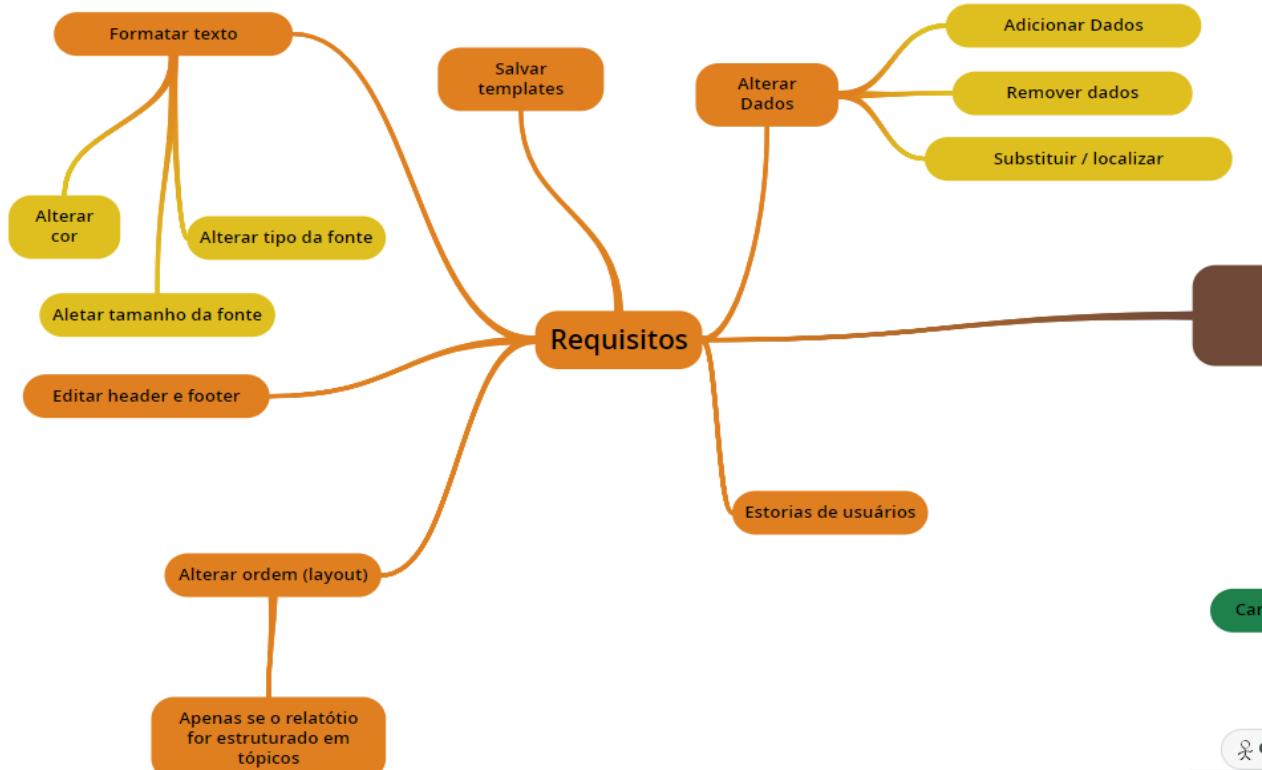


Além do canvas, que mostrava de uma maneira geral e organizada os principais tópicos do planejamento do software, fizemos um mapa mental com o objetivo de mapear de uma maneira mais visual os requisitos, limitações e objetivos ilustrados no canvas acima.

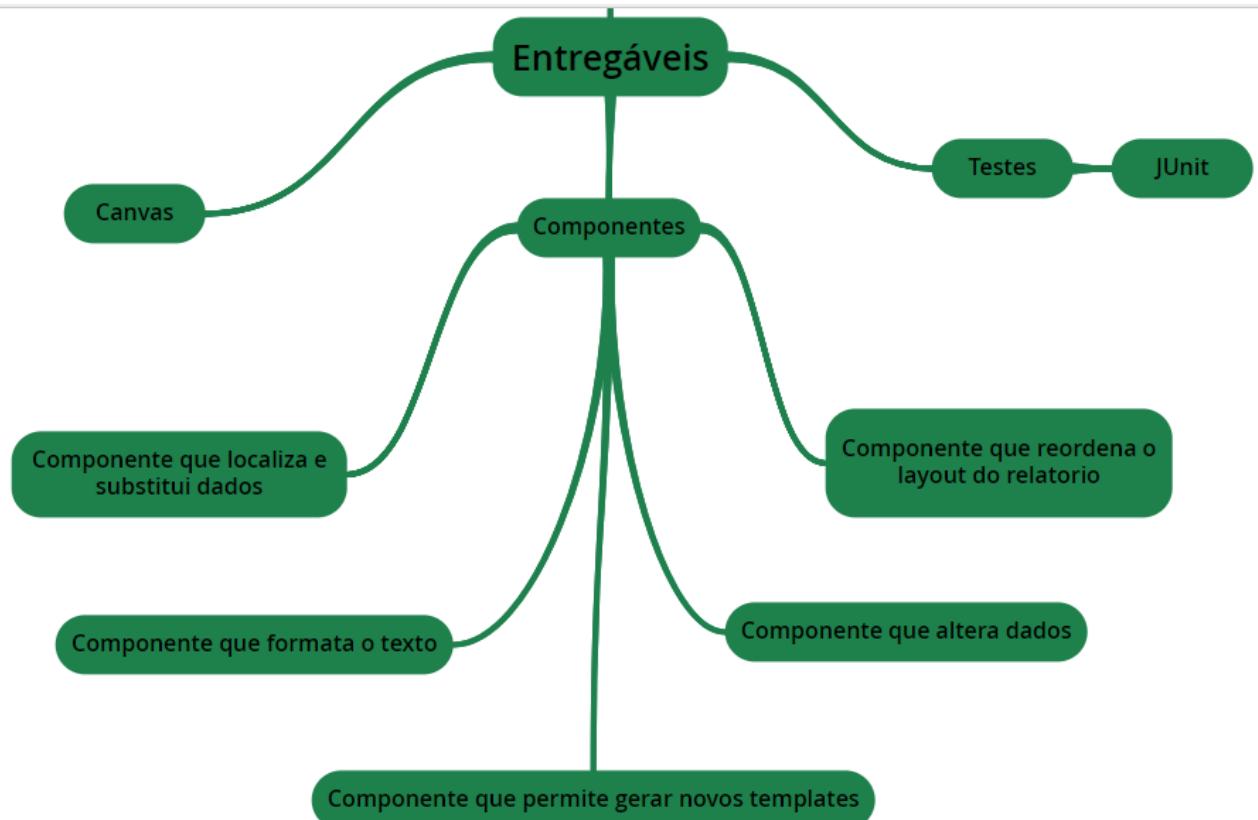


O mesmo possui quatro subárvores principais:

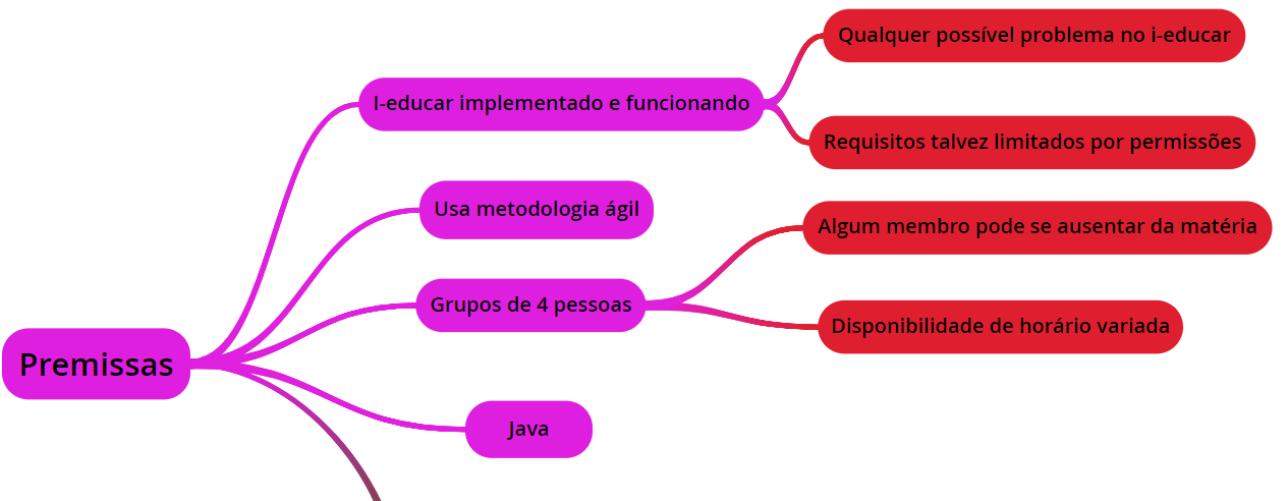
Requisitos:



Entregáveis



Premissas



Ferramentas



4. Mudanças no Planejamento: Merenda

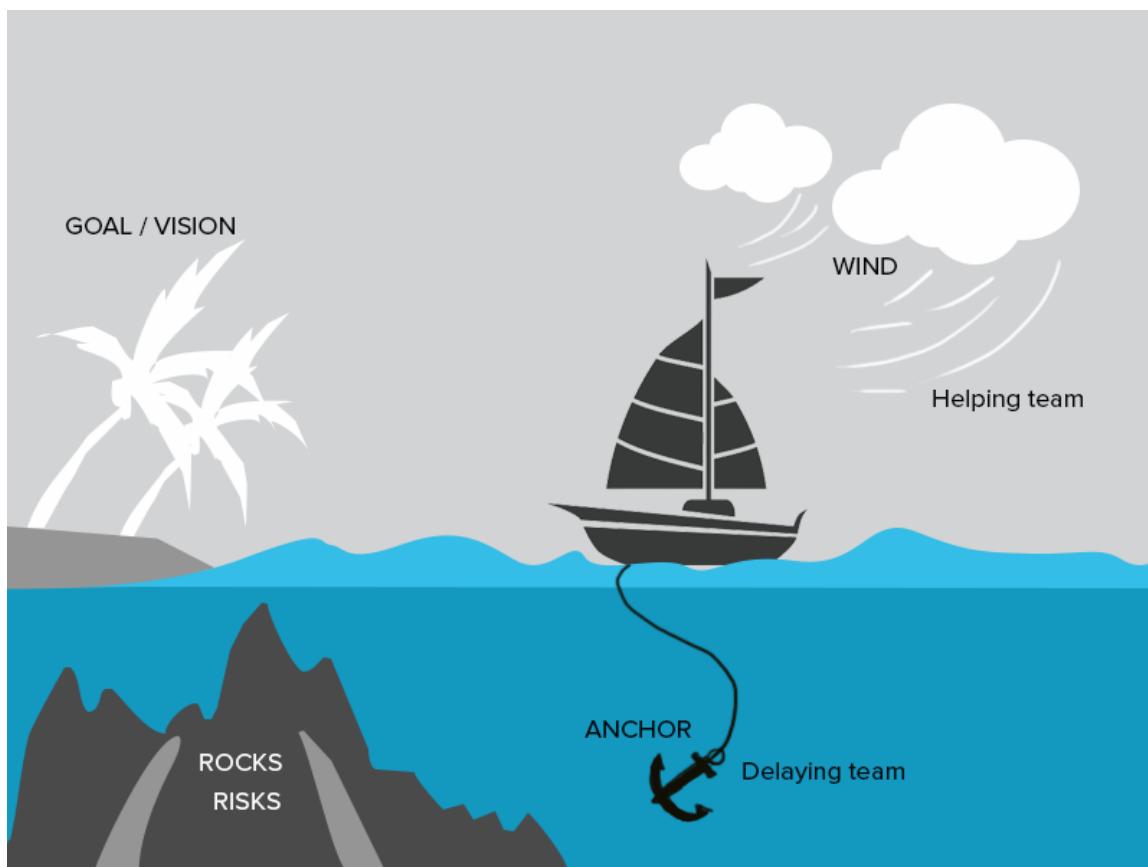
A parte de ferramentas continuou a mesma, entretanto, as demais mudaram bastante. As premissas se mantiveram, apenas a que exigia que o i-Educar estivesse implementado e funcionando não era mais necessária, visto que o novo projeto não possuía nenhum tipo de dependência com esta plataforma.

Os entregáveis e requisitos passaram a ser baseados nas *histórias de usuário* desenvolvidas em conjunto com a prefeitura em caxias e que podem ser vistas com mais detalhes no Anexo I. As histórias tiveram um importante papel no desenvolvimento do software, pois elas representavam de maneira simples e clara qual a funcionalidade desejada do usuário para a aplicação. E como a linguagem era simples, a validação da mesma com o *product owner* era muito mais transparente.

Elas foram a base para as tarefas que escolhíamos e designávamos a cada sprint.

A fim de poder identificar os principais fatores influenciadores em nosso projeto, utilizamos a metáfora do barco associando a cada um dos componentes os aspectos que achávamos que melhor se encaixavam no nosso contexto.

Todas as associações e explicações, sobre a metáfora, do barco podem ser vistas no Anexo II.



O que?

Um software funcional que auxilie no controle de merendas nos colégios da rede pública, seguindo como exemplo a prefeitura de caxias. Tal software deve atender aos requisitos necessários para a geração de relatórios de controle da merenda escolar.

Por que?

Todo o controle de merenda da rede pública é feito via excel, o que torna o lançamento de dados de merenda e até a verificação de tais dados trabalhosos. O software visa facilitar a geração, visualização e controle dos relatórios gerados por escola, de forma a automatizar parte do trabalho e centralizar a informação em um banco de dados.

Quem?

O software vai ser desenvolvido pelos membros do time citados e apresentados no tópico “Time”, além da participação do product owner Luis Felipe Coimbra Costa. Contamos também com a presença e participação da Poliana Guedes, representante intermediária da prefeitura de caxias, auxiliando-nos na criação e validação dos requisitos funcionais do software afim de atender da melhor maneira possível à demanda.

Como?

Utilizamos como especificação de requisitos as histórias de usuário, que contam de maneira resumida e informal uma breve história da aplicação de alguma funcionalidade do software informando o tipo de usuário que utilizaria aquela funcionalidade, a funcionalidade em si e sua motivação. Todas as histórias de usuário produzidas pela equipe podem ser vistas no Anexo I

Quando e Quanto?

A cada semana, escolhemos algumas histórias para trabalhar, a fim de alcançar aquele requisito. Ao final da semana, as tarefas determinadas para a história devem estar completas de forma a que ao final das 11 semanas, todos os requisitos estivessem implementados e testados. O custo do projeto foi de 5 reais, gastos em papeis, post it's e outros objetos de papelarias usados para planejamento do mesmo.

5. Desenvolvimento

Utilizando a técnica de métodos ágeis, realizamos um total de 11 sprints, em que em cada uma delas avaliávamos as tarefas das histórias que cumprimos na última sprint e apresentávamos nossas dificuldades e desenvolvimento no projeto. Em seguida, selecionávamos um conjunto de histórias para trabalhar naquela sprint e de cada uma delas selecionávamos um conjunto de tarefas. Cada tarefa ganhava uma estimativa de acordo com o seu nível de dificuldade.

Usando as estimativas, tentávamos deixar a distribuição de trabalho bem distribuída entre os membros, levando sempre em conta a disponibilidade de cada um.

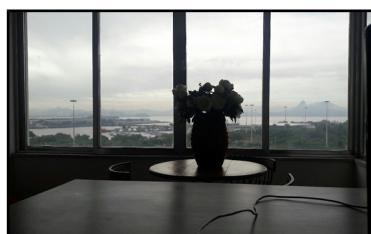


Illustration 1: Pomodoro

Participamos também de uma atividade proposta pelo Luis que se tratava de passar um dia em um ambiente de desenvolvimento de software de uma empresa que trabalha com métodos ágeis. Lá, utilizamos da estratégia Pomodoro para fazer uma sequência de tarefas com tempos predeterminados, a fim de que a sensação de estar cumprido com o programado ser uma constante e nos estimular mais.

Essa atividade foi muito importante para começarmos de vez com a implementação dos testes, em específico os testes unitários. O ambiente agradável do local também foi um forte estímulo para que nos empenhássemos ainda mais em completar as tarefas.

Conseguimos implementar um banco de dados funcional, o que foi muito satisfatório tendo em vista que era uma das dificuldades maiores que previmos para o projeto. A cada sprint, sempre que era possível encontrar a representante de caxias, mostrávamos o que já tínhamos pronto até o momento, pedindo a ela feedback sobre o que poderia melhorar ou revendo algumas histórias para confirmá-las.

Quanto aos testes, os testes realizados e seus resultados estarão anexados no Anexo IV.

6. Ferramentas

Para organização da Sprint e acompanhamento da mesma utilizamos o trello.

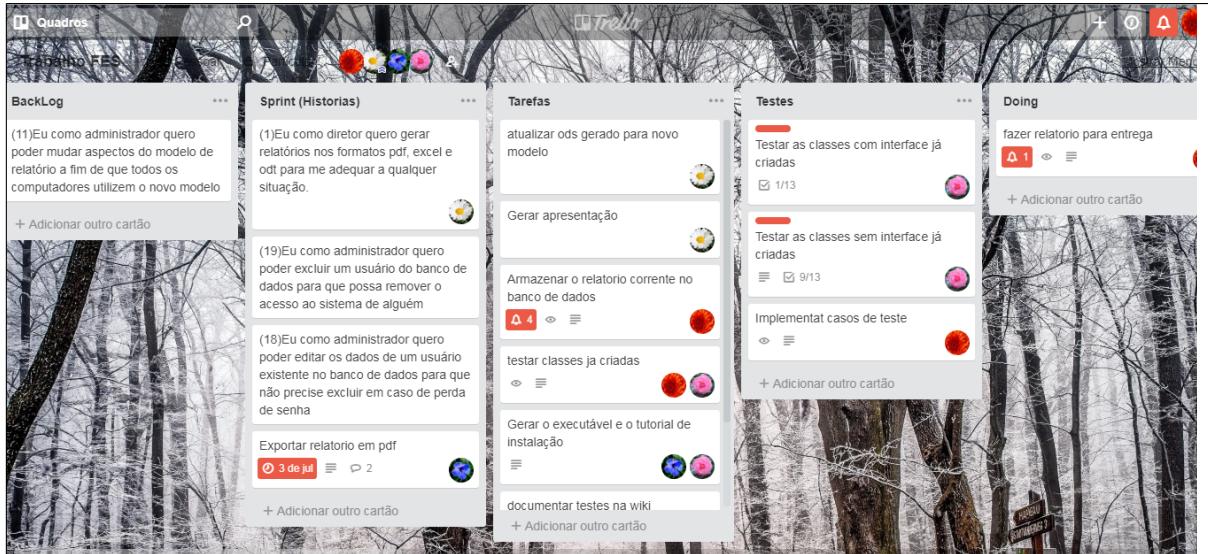
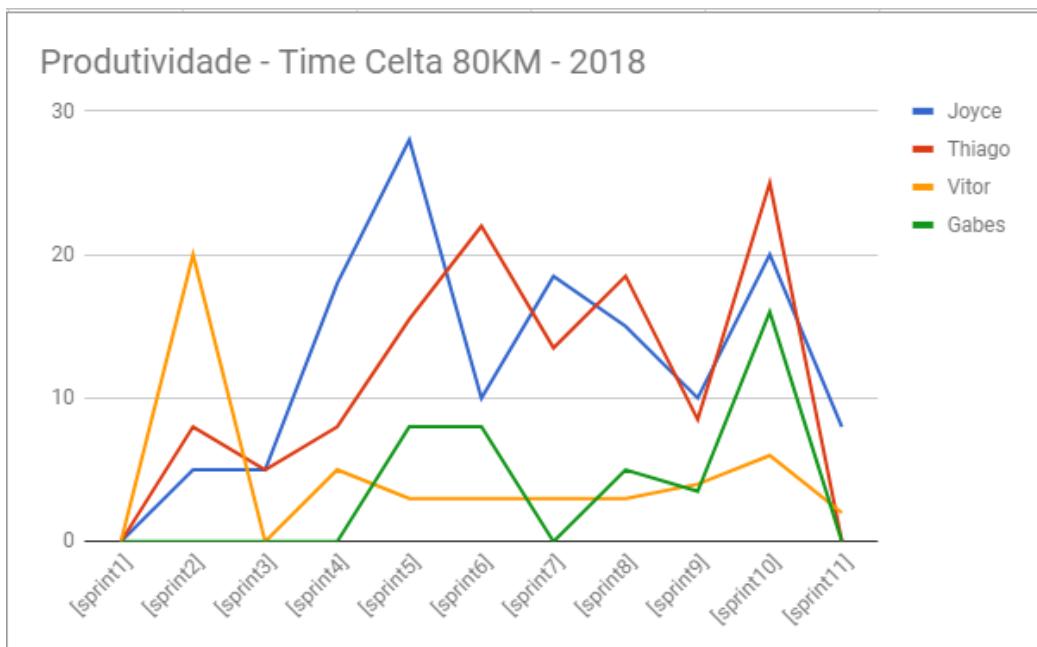


Illustration 2: Trello para Sprint

As sprints eram feita presencialmente às quartas, mas os *daily meeting* era feito pelo telegram, onde compartilhávamos o nosso progresso e dificuldades. Para acompanhar o nosso desenvolvimento individual no decorrer da sprint (da melhor maneira possível visto



que a mesma poderia estar sujeita a erros de estimativa) utilizamos o burndown chart pelo excel online.

7. Conclusão

Com o decorrer do projeto, todos os membros evoluíram seus conhecimentos em relação à distribuição inicial. Sendo colocado na prática, os conhecimentos em métodos ágeis se fizeram entender, todas as dinâmicas, as entregas semanais, as sprints, tudo que antes era apenas um brain storm associado a métodos ágeis, passou a ser parte uma estratégia coesa e coerente na mente de cada um dos membros.

Sobre conhecimento em java, a descoberta de novas API's e bibliotecas e a dificuldade em aprender a usá-las foram fatos constantes no desenvolvimento do software, aumentando consideravelmente o conhecimento dos programadores sobre java.

Vale salientar também que a aplicação de testes na prática foi o fator determinante para que entendêssemos sua importância no processo de engenharia de software. Através dos testes foi possível identificar tanto erros de programação quanto erros de arquitetura de classes, o que permitiu um aprimoramento intenso do código e da própria aplicação.

A presença quase que constante do usuário nas sprints mostrou a importância da participação do cliente no processo da engenharia de software baseada em métodos ágeis, pois com isso foi possível, a todo instante, verificar se de fato o que estava sendo desenvolvido atendia à demanda do cliente e, caso não atendesse, mudar a tempo a fim de evitar maiores prejuízos ao orçamento e ao prazo de entrega.

ANEXO I

HISTÓRIAS DE USUÁRIO

1. Eu como diretor quero gerar relatórios nos formatos pdf, excel e odt para me adequar a qualquer situação.
2. Eu como diretor quero poder gerar o relatório de maneira sequencial para ter mais praticidade.
3. Eu como diretor quero poder lançar atualizações diárias para poder não ter todo o trabalho de uma vez.
4. Eu como diretor quero poder editar dados já lançados para que não tenha que gerar um novo relatório do zero em caso de erro.
5. Eu como diretor quero poder me referenciar apenas ao mês e ano para gerar a página do cardápio para que não tenha que olhar no calendário.
6. Eu como diretor quero cadastrar os dados da minha escola de forma a não ter que ficar repetindo-os em todos os relatórios.
7. Eu como diretor gostaria que cálculos usando dados pré-inseridos fossem feitos pelo sistema para que eu tenha mais praticidade.
8. Eu como diretor gostaria de lançar diretamente os itens usados no cardápio para que não gaste tempo procurando um a um na tabela.
9. Eu como administrador quero poder mudar aspectos do modelo de relatório a fim de que todos os computadores utilizem o novo modelo
10. Eu como leitor quero poder filtrar os relatórios por escola para facilitar minha busca
11. Eu como leitor quero poder filtrar os relatórios por mês de referência para facilitar minha busca
12. Eu como usuário do sistema gostaria de poder ter acesso ao sistema através de um login e senha a fim de que possa trabalhar efetivamente.
13. Eu como administrador quero poder gerar novos perfis a fim de que possa dar acesso a outras pessoas de acordo com seu nível de acesso.
14. Eu como usuário gostaria que todos os arquivos fossem salvos em um banco de dados para centralizar a informação.

15. Eu como usuário quero poder interagir com uma tela de login a fim de que a aplicação fique mais intuitiva e organizada
16. Eu como administrador quero adicionar e modificar permissões por tipo de usuário para que tenha mais liberdade de modificar futuramente.
17. Eu como administrador quero poder editar os dados de um usuário existente no banco de dados para que não precise excluir em caso de perda de senha
18. Eu como administrador quero poder excluir um usuário do banco de dados para que possa remover o acesso ao sistema de alguém

ANEXO II

METÁFORA BARCO

Elementos associados ao celta: Âncora – Freio de mão; Rochas – Buraco na Estrada; Vento – Motor

FREIO DE MÃO

- Nenhum dos membros do time tem conhecimentos de banco de dados.
- Dois dos membros do time estagiam, limitando os horários de reunião.
- Um dos membros tem 6 matérias puxadas.

BURACO NA ESTRADA

- Provas de outras matérias atrapalham o desenvolvimento.
- Não conseguimos implementar o banco de dados.
- Ficar 200% desmotivados

MOTOR

- Gabriel possui conhecimento em interfaces em java.
- Todos os membros possuem bons conhecimentos de lógica de programação e programação orientada à objetos.
- Estamos 200% motivados.
- Comunicação do time é impecável.

OBJETIVO

- Uma aplicação funcional que possibilite o controle de merendas pelas escolas usuárias, com um banco de dados interno.
- Conseguir um bom desempenho na matéria com nosso desenvolvimento individual.

ANEXO III

CASOS DE USO

Caso de uso 1	
1. Nome do caso de uso:	Acesso ao sistema
2. Descrição resumida:	Caso de uso referente ao login do usuário no sistema
3. Ator:	Usuário
4. Atores Secundários:	não há
5. Pré-condição:	Usuário precisa estar cadastrado no sistema
6. Fluxo de eventos	<ul style="list-style-type: none">6.1. Digita o login de acesso6.2. Digita senha de acesso6.3. Clicka em logar
7. Pós-condições:	Sistema deve redirecionar para a tela principal
8. Fluxos Alternativo	<ul style="list-style-type: none">8.1. Usuário pode ter errado seu login<ul style="list-style-type: none">8.1.1. Aparece um pop-up dizendo que o nome de usuário está incorreto.8.2. Usuário errou sua senha

8.2.1. Aparece um pop-up informando que a senha está incorreta.
9. Fluxo de Exceção:
Não há

CASO DE USO 2	
1. Nome do caso de uso:	
Ver Relatório	
2. Descrição resumida:	
Usuário pode ver um relatório filtrado por mes e ou ano em pdf , de acordo com o contexto de escola em que se encontra.	
3. Ator:	
Usuário	
4. Atores Secundários:	
não há	
5. Pré-condição:	
Estar logado no sistema e ter a permissão para ver relatório	
6. Fluxo de eventos	
6.1. Seleciona a opção lista de relatórios	
6.2. Digita no campo “ano” o ano do relatório desejado	
6.3. Digita no campo “mês” o mês do relatório desejado	
6.4. A lista de relatórios irá mostrar apenas o relatório do mês e do ano desejados.	
6.5. Selecione o relatório.	
6.6. Selecione a opção ver relatório	
6.7. Selecione o formato do relatorio (odt, pdf ou ods)	
6.8. Selecione a opção abrir	

Sistema deve abrir o documento (pdf, odt ou ods) mostrando o relatório selecionado
7. Fluxos Alternativo
7.1. Usuário só informou um dos campos (mês ou ano)
7.1.1. Em vez de aparecer apenas um relatório, aparecerá a lista filtrada pelo campo informado.
8. Fluxo de Exceção:
8.1. Não existir relatório encontrado para o filtro informado
8.1.1. A opção ver relatório não será habilitada
8.2. Relatório não encontrado no banco de dados
8.2.1. Informa ao usuário que houve um problema com o banco de dados e que o relatório não está disponível para visualização

CASO DE USO 3
1. Nome do caso de uso:
Editar relatório
2. Descrição resumida:
Usuário poderá editar o conteúdo de relatórios já criados, de qualquer mês ou ano de referência, desde que o mesmo seja da escola a qual ele é responsável.
3. Ator:
Usuário
4. Atores Secundários:
não há
5. Pré-condição:
Estar logado no sistema e ter a permissão para editar relatório
6. Fluxo de eventos
6.1. Seleciona a opção editar relatório

6.2. Escolhe um relatório dentre os mostrados na lista
6.3. Seleciona a opção avançar
6.4. Edita os dados da capa
6.5. Avançar
6.6. Edita os dados do cardápio
6.7. Avançar
6.8. Edita os dados da lista de ingredientes
6.9. Seleciona salvar para salvar as modificações no banco de dados.
7. Pós-condições:
Sistema deve salvar as modificações feitas no relatório selecionado no banco de dados e redirecionar para o menu principal
8. Fluxos Alternativo:
8.1. Seleciona a opção editar relatório
8.2. Filtra os relatórios por mês e/ou ano de referência
8.3. Escolhe um relatório dentre os mostrados na lista
8.4. Seleciona a opção avançar
8.5. Edita os dados da capa
8.6. Avançar
8.7. Edita os dados do cardápio
8.8. Avançar
8.9. Edita os dados da lista de ingredientes
8.10. Seleciona salvar para salvar as modificações no banco de dados
9. Fluxo de Exceção:
9.1. Não existir relatório encontrado para o filtro informado
9.1.1. A opção avançar não será habilitada
9.2. Usuário preenche algum campo incorretamente

9.2.1. Aparecerá um pop-up informando que o campo está inválido e precisa ser verificado.

CASO DE USO 4

1. Nome do caso de uso:

Criar Relatório

2. Descrição resumida:

usuário pode criar um relatório novo para a escola selecionada no contexto.

3. Ator: Usuário

4. Atores Secundários:

não Há

5. Pré-condição:

5.1. Estar logado no sistema e ter a permissão para criar relatório

5.2. Fluxo de eventos

5.2.1. Seleciona a opção criar relatório

5.2.2. Informa o mês de referência

5.2.3. Informa o ano de referência

5.2.4. Preenche os dados da capa

5.2.5. Avança

5.2.6. Preenche os dados do cardápio

5.2.7. Avança

5.2.8. Preenche a lista de itens usados

5.2.9. Seleciona salvar para salvar as modificações no banco de dados.

6. Pós-condições:

Sistema deve inserir o novo relatório no banco de dados no banco de dados e redirecionar para o menu principal.

7. Fluxos Alternativo

7.1. Usuário não informa o mês de referência

7.2. Ao clickar em avançar, aparece um pop-up alertando-o de que o campo está inválido ou não preenchido
7.3. Usuário não informa o ano de referência
7.4. Ao clickar em avançar, aparece um pop-up alertando-o de que o campo está inválido ou não preenchido.
8. Fluxo de Exceção
8.1. Usuário cria um relatório com o mês e ano de referência igual ao de outro já existente no banco de dados para a mesma escola
8.1.1. Aparece um pop-up informando que o relatório já existe e que ele deve ser editado e não criado novamente, impedindo o usuário de prosseguir.

CASO DE USO 5

1. Nome do caso de uso:
Cadastrar escola
2. Descrição resumida:
O usuário pode cadastrar uma nova escola no banco de dados.
3. Ator:
Usuário
4. Atores Secundários:
Não há
5. Pré-condição:
Estar logado no sistema e ter a permissão para cadastrar escola
6. Fluxo de eventos
6.1. Seleciona “Gerenciar Escolas”
6.2. Seleciona a opção com um “+”
6.3. Preenche com os dados dessa nova escola
6.4. Salva

7. Pós-condições:
Sistema deve redirecionar para a tela principal
8. Fluxos Alternativo:
Não há
9. Fluxo de Exceção
9.1. Usuário não tem permissão para cadastrar escola, mas tem permissão de gerenciar
9.1.1. Aparece as demais opções mas a opção com o “+” é desabilitada
9.2. Usuário tenta inserir uma escola já existente no banco de dados
9.2.1. O sistema bloqueia a operação informando que a escola já se encontra cadastrada.

CASO DE USO 6
1. Nome do caso de uso:
Editar Escola
2. Descrição resumida:
O usuário pode editar uma escola do banco de dados.
3. Ator:
Usuário
4. Atores Secundários:
Não há
5. Pré-condição:
Estar logado no sistema e ter a permissão para editar escola
6. Fluxo de eventos
6.1. Seleciona “gerenciar escolas”
6.2. Seleciona uma escola para gerenciar no dropdown

6.3. Selecione a opção editar
6.4. Edita os campos da escola
6.5. Selecione “Salvar”
7. Pós-condições: sistema deve atualizar o banco de dados e redirecionar para a tela principal
8. Fluxos Alternativo
8.1. Cancelar edição
8.1.1. Seleciona “gerenciar escolas”
8.1.2. Seleciona uma escola para gerenciar no dropdown
8.1.3. Selecione a opção editar
8.1.4. Edita os campos da escola
8.1.5. A opção voltar cancela toda edição feita e volta para tela anterior
9. Fluxo de Exceção
9.1. Tentar mudar o inep (unico por escola) por um ja existente no banco
9.2. Edições desprezadas e operação cancelada.
9.3. Volta para tela de escolha de escola.

CASO DE USO 7
1. Nome do caso de uso:
Cadastrar usuário
2. Descrição resumida:
O usuário pode cadastrar um novo usuário no banco de dados
3. Ator:
Usuário
4. Atores Secundários:
não há

5. Pré-condição:
Estar logado no sistema como usuário administrador
6. Fluxo de eventos
6.1. Seleciona “Gerenciar Usuários”
6.2. Seleciona “cadastrar usuário”
6.3. Preencha os campos
6.4. Selecione salvar
7. Pós-condições:
Sistema deve criar o usuário no banco e redirecionar para a tela principal
8. Fluxos Alternativo:
Não há
9. Fluxo de Exceção
9.1. O usuário (username) já existir no banco
9.1.1. pop-up dizendo que este usuário já existe no banco
9.2. Faltar preencher algum campo
9.2.1. pop-up dizendo que tem campos não completos

CASO DE USO 8
1. Nome do caso de uso:
Editar usuário
2. Descrição resumida:
Usuário pode editar os dados de algum usuário
3. Ator:
Usuário
4. Atores Secundários:
5. Pré-condição:

Estar logado no sistema e ter permissão de edição de usuário
6. Fluxo de eventos
6.1. Seleciona “Gerenciar Usuários”
6.2. Seleciona “Editar Usuario”
6.3. Preencha com o nome de usuario
6.4. Seleciona “Pesquisar”
6.5. Edite os dados apresentados
6.6. Seleciona “salvar”
7. Pós-condições: sistema deve atualizar os dados do usuario e redirecionar para a tela principal.
8. Fluxos Alternativo
8.1. Pesquisar por e-mail
8.1.1. Seleciona “Editar Usuario”
8.1.2. Preencha com o e-mail
8.1.3. Seleciona “Pesquisar”
8.1.4. Edite os dados apresentados
8.1.5. Seleciona “salvar”
9. Fluxo de Exceção
9.1. Nome de usuário não existente no banco
9.1.1. Aparece um pop-up informando que não há nenhum usuário com o nome de usuário informado.
9.1.2. permanece na tela para pesquisa de usuário
9.2. Email não existente no banco
9.2.1. Aparece um pop-up informando que não há nenhum usuário com o email informado.
9.2.2. permanece na tela para pesquisa de usuário
9.3. Nome usuário alterado para um já existente no banco

9.3.1. Aparece um pop-up informando que o nome de usuário já existe no banco
9.3.2. permanece na tela para edição.
9.4. Email alterado para um já existente no banco
9.4.1. Aparece um pop-up informando que o email já existe no banco
9.4.2. permanece na tela para edição.

CASO DE USO 9

1. Nome do caso de uso:

Excluir usuário

2. Descrição resumida:

Usuário excluir outro usuário do banco

3. Ator:

Usuário

4. Atores Secundários:

Não há

5. Pré-condição:

Estar logado no sistema e ter permissão de excluir usuário

6. Fluxo de eventos

6.1. Seleciona “Gerenciar Usuários”

6.2. Seleciona “Excluir usuário”

6.3. Preencha com o nome de usuário

6.4. Seleciona “Excluir”

6.5. Selecione Ok no pop up de confirmação

7. Pós-condições:

Sistema deve redirecionar para a tela principal
8. Fluxos Alternativo
8.1. Excluir pelo e-mail
8.1.1. Seleciona “Gerenciar Usuários”
8.1.2. Seleciona “Excluir usuário”
8.1.3. Preencha com o e-mail
8.1.4. Seleciona “Excluir”
8.1.5. Selecione Ok no pop up de confirmação
8.2. Usuário muda de ideia sobre a exclusão
8.2.1. Seleciona “Gerenciar Usuários”
8.2.2. Seleciona “Excluir usuário”
8.2.3. Preencha com o e-mail
8.2.4. Seleciona “Excluir”
8.2.5. Seleciona a opção “cancelar” no pop-up de confirmação
9. Fluxo de Exceção
9.1. Usuário tenta excluir o próprio usuário
9.1.1. Aparece pop-up informando que a operação não é válida
9.1.2. Operação é cancelada
9.1.3. Redireciona para a pesquisa de usuários.

CASO DE USO 10
1. Nome do caso de uso:
Editar relatório corrente
2. Descrição resumida:
Usuário pode de maneira mais rápida editar o último relatório feito pelo mesmo.
3. Ator:

Usuário
4. Atores Secundários:
Não há
5. Pré-condição:
Estar logado no sistema e ter permissão para editar relatorio
6. Fluxo de eventos
6.1. Seleciona a opção “Editar Relatório Corrente”
6.2. Edita os dados da capa
6.3. Avançar
6.4. Edita os dados do cardápio
6.5. Avançar
6.6. Edita os dados da lista de ingredientes
6.7. Seleciona a opção “salvar” para salvar as modificações no banco de dados.
7. Pós-condições:
Sistema deve salvar as modificações feitas no relatório selecionado no banco de dados e redirecionar para o menu principal
8. Fluxos Alternativo:
Não há
9. Fluxo de Exceção:
9.1. Não existir relatório corrente
9.1.1. A opção editar relatório corrente não será habilitada
9.2. Usuário preenche algum campo incorretamente
9.2.1. Aparecerá um pop-up informando que o campo está inválido e precisa ser verificado.

CASO DE USO 11	
1. Nome do caso de uso:	
2. Descrição resumida:	Usuário pode editar as permissões para cada um dos três tipos de usuário: Diretor, Administrador e Leitor
3. Ator:	Usuário
4. Atores Secundários:	Não há
5. Pré-condição:	Estar logado no sistema e ter permissão para editar permissões
6. Fluxo de eventos	<ul style="list-style-type: none">6.1. Selecione a opção “Atualizar permissões”6.2. Selecione o tipo de usuário dentre as três opções no dropDown6.3. Marque as permissões que o mesmo passará a ter6.4. Clique em atualizar6.5. Selecione Ok no pop up de confirmação
7. Pós-condições:	8. Sistema deve redirecionar para a tela principal e atualizar as permissões no banco
9. Fluxos Alternativo	<ul style="list-style-type: none">9.1. Usuário desiste da atualização<ul style="list-style-type: none">9.1.1. Selecione a opção “Atualizar permissões”9.1.2. Selecione o tipo de usuário dentre as três opções no dropDown9.1.3. Marque as permissões que o mesmo passará a ter9.1.4. Clique em atualizar9.1.5. Selecione cancelar no pop up de confirmação

10. Fluxo de Exceção:
Não há

CASO DE USO 12	
1. Nome do caso de uso:	
Deslogar do sistema	
2. Descrição resumida:	
usuário quer sair do seu acesso do sistema e voltar a tela de login	
3. Ator:	
Usuário	
4. Atores Secundários:	
não há	
5. Pré-condição:	
Estar logado no sistema	
6. Fluxo de eventos	
6.1. Selecione a opção “sair”	
7. Pós-condições:	
sistema deve redirecionar para a tela de login	
8. Fluxos Alternativo: não há	
9. Fluxo de Exceção: não há	

CASO DE USO 13	
1. Nome do caso de uso:	
Mudar contexto da escola	
2. Descrição resumida:	

Usuário deseja trocar o contexto da escola na qual está trabalhando para uma outra escola
3. Ator:
Usuário
4. Atores Secundários:
Não há
5. Pré-condição:
Estar logado no sistema
6. Fluxo de eventos
6.1. Selecione a opção “Alterar Escola”
6.2. Selecione a escola desejada
6.3. Selecione trocar
7. Pós-condições:
Sistema deve redirecionar para a tela principal com o contexto atualizado
8. Fluxos Alternativo:
8.1. Desiste da operação
8.1.1. Selecione a opção “Voltar”
9. Fluxo de Exceção:
Não há

ANEXO IV

CASOS DE TESTE

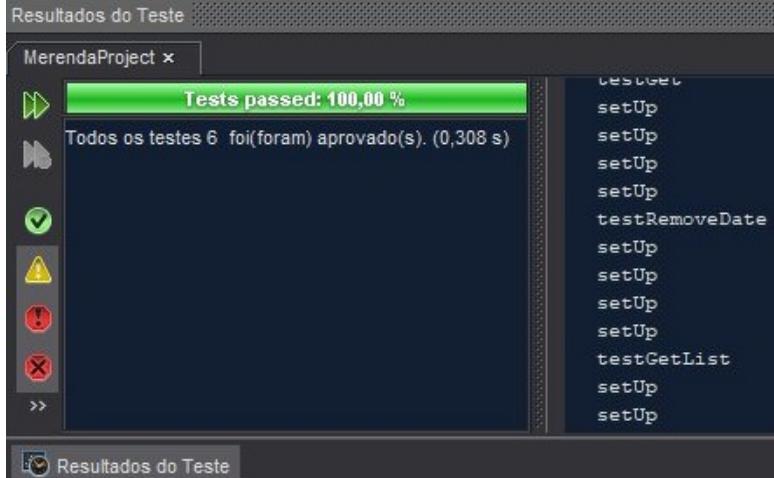
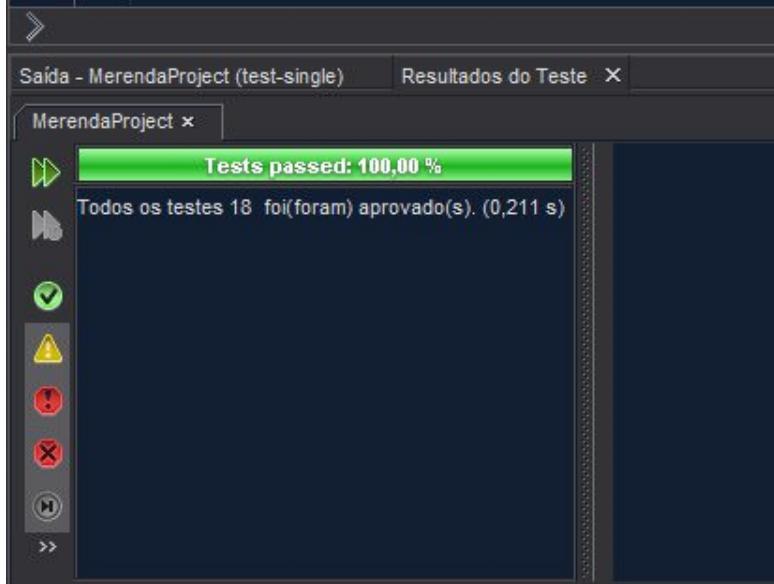
Caso de teste UC001 – CT 001

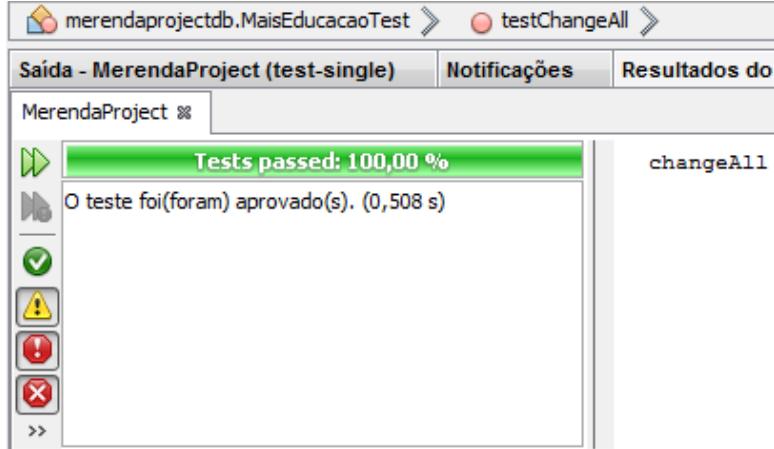
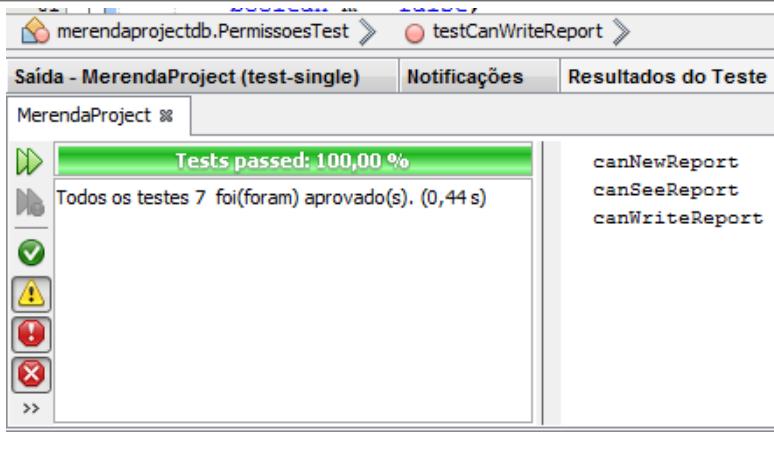
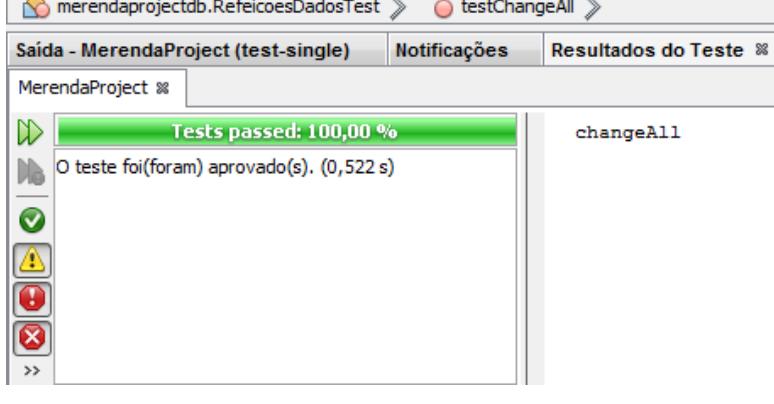
1. **Identificação do cenário:** Fluxo básico
2. **Pré Memória**
 - 2.1. **Massa de dados**
 - 2.1.1. Usuário existente na tabela “usuário” do banco de dados
 - 2.1.2. Senha S1 do usuário
 - 2.2. **Propriedades**
 - 2.2.1. Não há
3. **Pós Memória**
 - 3.1. **Massa de dados:**
 - 3.1.1. Tela principal.usuário = usuário
 - 3.2. **Propriedades:**
 - 3.2.1. Logado no sistema
4. **Entrada e Saída Esperados**

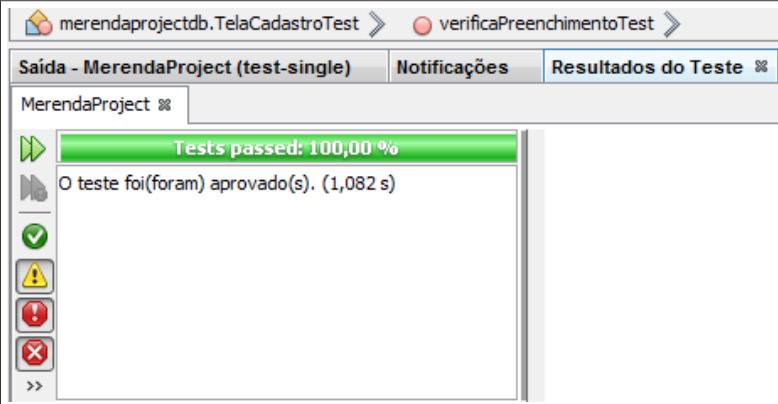
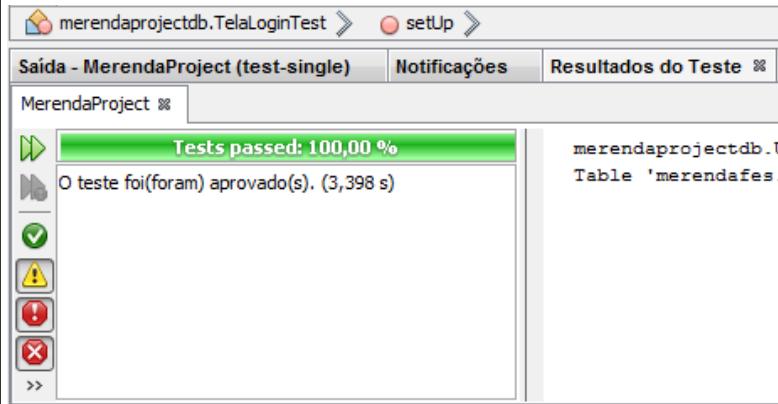
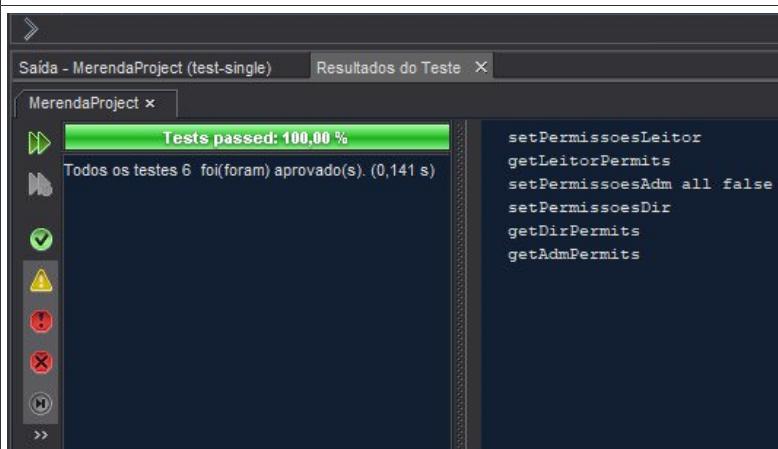
Passo	Descrição	Variáveis			
		Var1	Val1	Var2	Val2
1	Preencher campo usuário	userv	user		
2	Preencher campo senha			senha	S
3	Clicar em “logar”				

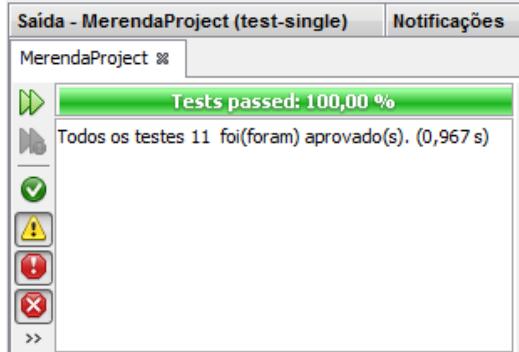
ANEXO V

TESTES DOCUMENTADOS

Tipo de Teste	Classe/Caso de uso	Resultado
Unitário	Calendário	
Unitário	Escola	

Unitário	MaisEducacao	
Unitário	Permissoes	
Unitário	RefeicoesDados	

Unitário	TelaCadastro	
Unitário	TelaLogin	
Unitário	TiposDeUsuario	

			
Unitário	Usuario	<p>Saída - MerendaProject (test-single)</p> <p>MerendaProject</p> <p>Tests passed: 100,00 %</p> <p>Todos os testes 11 foi(foram) aprovado(s). (0,967 s)</p> <p>>></p> <p>getNome getTipo getUser setEscola getEscola setNome setUser setEmail setSenha</p>	

ANEXO VI

REAÇÕES

Aulas 19 e 21 de Março

Metodologia ágil é um modelo de esquematização e execução de projetos de maneira muito mais eficiente e comunicativa do que as clássicas. Uma das maneiras de executar o projeto por esta metodologia é utilizando o scrum, que em reuniões periódicas, normalmente semanais, mas com prazo máximo de um mês (não mais que isso), decide uma sequência de tarefas a serem executadas (sprints) além de estabelecer de maneira mais ampla tudo que deve ser executado pensando em seus prazos de entrega, a fim de cumprir com os requisitos dentro de seus prazos.

Esta maneira de ver o projeto de uma maneira mais geral, estabelecendo metas a longo prazo, é chamada de BackLog e são nas tarefas definidas neste que é decidida e divida a sprint entre todos os desenvolvedores do grupo. Já os requisitos representam todas as características e funcionalidades que um determinado software possui, tais como, ter uma interface amigável, ter uma opção de modificar todas as palavras selecionadas por um outra específica ou um pedido do usuário, como a solicitação de que haja um botão na cor de rosa no canto inferior esquerdo.

Um modo amplamente difundido e conhecido de se aplicar a metodologia ágil no processo de idealização do projeto é utilizando o project model canvas, que tenta responder de maneira colaborativa e organizada em pequenos blocos as perguntas fundamentais do mesmo: Por que, O que, Quem, Como e Quanto.

Aulas 26 e 28 de Março

Duas equipes podem ter como resultado final da elaboração dois canvas completamente diferentes, mesmo que o contexto, as condições base e o projeto seja o mesmo. Isto acontece pois a elaboração bastante subjetiva e depende muito de como cada equipe olha para o projeto como um todo.

Entretanto, existem aspectos fundamentais que não podem deixar de estar presente em todos os canvas, o que poderíamos chamar de “pontos chaves”. Um exemplo é o fato de o product owner sempre aparecer presente na equipe, pois pelo modelo de processo SCRUM o product owner é um importante membro da equipe e deve participar ativamente da idealização do projeto.

Vale a pena comentar a diferença entre Processo, Processo de Software e Modelo de Processo. Um processo é uma sequência de passos com um propósito, enquanto que um processo de software são atividades, métodos, tecnologias e práticas utilizadas para o desenvolvimento de um software. Um Modelo de Processo é a abstração de um determinado processo, de forma a descrevê-lo de uma maneira geral.

Um projeto é uma instância de um processo, pois um determinado processo pode ser aplicado para mais de um projeto, enquanto que um mesmo projeto só pode usar um único processo. Aplicando esta comparação para o projeto atual de desenvolver um plugin para o e-Educar, o processo utilizado seria o SCRUM. Para desenvolver o projeto será utilizada a linguagem java, por isto, durante uma das aulas foi apresentado sua estrutura principal.

Aulas 09 e 11 de Abril

As aulas a qual se referem o texto tratam sobre o escopo do projeto e a sua possível mudança levando em conta os aspectos apresentados pela prefeitura de Caxias sobre sua experiência com o i-report e todos os seus problemas técnicos referente a geração de relatório.

Os problemas mostrados foram, por exemplo: o título que deveria aparecer só na primeira página aparece em todas, caso alguma informação esteja faltando, o sistema não permite substituí-la por um traço, i-educar não permite que o header apareça só na primeira página, etc. A maior dificuldade apresentada pelas representantes da prefeitura de Caxias é ter que passar parâmetros para fazer comparações com tabelas sem necessidade.

O que foi apresentado fez parecer ao nosso grupo que o problema estava muito além simples customização fraca de relatório, sim intrinsecamente ligado a própria estrutura interna do i-report, gerando problemas em relatórios já prontos e customizados. Em seguida, foi discutido que seria feito quanto aos requisitos apresentados pela prefeitura o que gerou um embate entre basicamente duas opções principais: 1) continuar com o projeto de relatórios pro i-report e 2) mudar o projeto para, em vez disso, construir um controle de merenda. Posteriormente, surgiu uma terceira opção que seria fazer um sistema que gerasse relatórios usando um sistema de cadastro já implementado para implementar um banco de dados simples que funcionasse com o mesmo.

Aulas 14 e 16 de Maio

Os testes e validações representam parte importante do processo de engenharia de software. Existem dois tipos de teste: Alfa (realizado no ambiente de desenvolvimento) e Beta (realizado no ambiente do cliente).

Os testes podem ser de unidade (foco nos componentes individuais), de integração (verifica a integração entre os componentes), de validação (verifica o cumprimento dos requisitos) ou de sistema (foco no funcionamento correto da combinação do software com os outros elementos do sistema: hardware, pessoas, dados).

Os testes são feitos com algo chamado de caso de teste, que corresponde a uma dupla de entrada e saída (E,S). Um caso de teste é dado como bem sucedido se para uma entrada E obtivemos uma saída S, e é dado como mal sucedido se para a entrada E houver qualquer saída que não S. O oráculo de teste é quem verifica a validação do caso de uso.

Um caso de teste positivo é quando a entrada deveria corresponder a uma saída válida e um caso de teste negativo a entrada corresponde a um estado ou mensagem de erro.

Existem, basicamente, dois tipos de teste: teste caixa branca e teste caixa preta. Os testes caixa branca são realizados olhando o código e seguindo alguns critérios de validação (como por exemplo casos testes que garantam que todas as linhas de código foram executadas ou que abranjam todos os fluxos de execução) enquanto que os testes caixa preta se atêm apenas às especificações do sistema.

ANEXO VII

DOCUMENTAÇÃO SPRINTS

Sprint 3

SPRINT REVIEW

O que deveria ter sido feito

Tarefas Realizadas na Sprint:

Histórias:

- Eu como diretor quero cadastrar os dados da minha escola de forma a não ter que ficar repetindo-os em todos os relatórios. (Done)

Tarefas:

- Criar interface para cadastrar escola
- Criar interface para excluir escola
- Conseguir fazer modificações no banco de dados no cadastro de escolas
- Eu como diretor quero poder gerar o relatório de maneira sequencial para ter mais praticidade.

Tarefas:

- Criar as três opções de relatório
- Eu como administrador quero poder gerar novos perfis (tipo de conta) a fim de que possa dar acesso a outras pessoas de acordo com seu nível de acesso. (Done)

Tarefas:

- Criar interface de cadastro de usuários

SPRINT PLANNING

Histórias ainda não finalizadas na sprint anterior

- Eu como diretor quero poder lançar os cardápios por dia para que no final do mês não tenha trabalho acumulado

Tarefas:

- Configurar a opção do cardápio
- Mandar para o banco de dados
- Eu como diretor gostaria de lançar diretamente os itens usados no cardápio para que não gaste tempo procurando um a um na tabela.

Tarefas:

- Criar a tela de itens com um dropdown funcional
- Pegar os itens que aparecem no dropdown do banco de dados

- Eu como usuário gostaria que todos os arquivos fossem salvos em um banco de dados para centralizar a informação.

Tarefas:

- Criar uma classe para centralizar a comunicação com o banco
- Conectar todas as partes ate o momento ao banco de dados
- Colocar o banco de dados na amazon
- Eu como diretor quero poder editar dados já lançados para que não tenha que gerar um novo relatório do zero em caso de erro.

Tarefas:

- Criar uma classe que represente um relatorio
- Criar uma interface que mostre uma lista dos relatorios existentes (pegos do banco de dados)

RETROSPECTIVA

Plataforma para fazer o burndown chart muito complicada, a partir dessa sprint usaremos a planilha oferecida pelo professor

Sprint 4

Retrospectiva:

Gabriel tem muitas matérias e essa semana está overjob com trabalhos
Joyce ficou muito doente de quara feira até hoje, mas conseguiu fazer sua parte
Vitor teve complicações técnicas com seu computador mas estudou como fazer
Thiago tava com a Joyce indo em médico mas fez coisas no ultimo dia

Dúvidas

Como fazer testes na interface (Simular interação do usuário com a interface)

Review

- Joyce:

A tela de cardapio agora recebe as entradas do cardapio por dia e armazena num formato de dupla <Date, String> e permite a exclusão de linhas mesmo que ainda precise de melhorias nesta ultima

- Thiago:

A tela de itens de relatório agora consegue receber uma lista pré-estabelecida via código, permitindo que a lista venha direto do banco e adicionar o item selecionado numa tabela com itens selecionados. As funcionalidades de excluir itens dessa tabela e de mandar para o banco de dados ainda precisa ser implementada.

- Vitor:

O mySQL deu problema em seu computador então ele não conseguiu implementar a conexão com o banco de dado como planejada na sprint então o mesmo pesquisou sobre o banco de dados da amazon (que é inviável de usar por só ser grátis por um mês) e estudou sobre como fazer sua parte.

- Gabriel:

Não fez nada por motivos de trabalhos na faculdade.

Sprint Planning

Histórias ainda não finalizadas na sprint anterior

- Eu como diretor gostaria de lançar diretamente os itens usados no cardápio para que não gaste tempo procurando um a um na tabela.

Tarefas:

- Pegar os itens que aparecem no dropdown do banco de dados
- Eu como usuário gostaria que todos os arquivos fossem salvos em um banco de dados para centralizar a informação.

Tarefas:

- Conectar todas as partes ate o momento ao banco de dados
- Conectar todas as partes ao banco de dados
- Eu como diretor quero poder editar dados já lançados para que não tenha que gerar um novo relatório do zero em caso de erro.

Tarefas:

- Criar uma classe que represente um relatorio
- Criar uma interface que mostre uma lista dos relatorios existentes (pegos do banco de dados)
- Criar opção editar relatorio

Sprint 5

Retrospectiva

- Acabou sendo necessário fazer uma redistribuição das tarefas entre os membros devido à maior disponibilidade de alguns membros em comparação com outros.

Review

- Joyce: Criou uma classe que representa um relatório, customizou a opção de editar relatório, criou a classe que representa a capa do relatório.
- Thiago: terminou a tela de seleção de itens do relatório, sincronizou diversas partes do código com a classe BdManager (responsável pela comunicação com o banco de dados) e criou a tela de listagem de relatórios já existentes.
- Vitor: resolveu problemas na conexão com o banco de dados. Planning
- Eu como usuário gostaria que todos os arquivos fossem salvos em um banco de dados para centralizar a informação
 - Conectar todas as partes do código ao BdManager
 - Ligar todas as partes do BdManager ao banco
- Eu como diretor quero poder editar dados já lançados para que não tenha que gerar um novo relatório do zero em caso de erro

- Adaptar tela lista de itens para opção edição
- Adaptar tela cardápio para opção edição
- Configurar a criação de relatórios parar criar uma instancia de Relatório, adiciona-la ao banco e atualizar o relatório corrente
- Eu como diretor quero poder gerar relatórios nos formatos pdf, excel e odt para me adequar a qualquer situação.
 - Criar uma classe que gere pdf
 - Criar uma classe que gere odt
 - Criar uma classe que gere excel
- Eu como diretor gostaria de poder lançar diretamente os itens usados no cardápio para que não gaste tempo procurando um a um na tabela.
 - Permitir pesquisa por nome da escolha de itens do relatório
- Eu como usuário quero poder interagir com uma tela de menu a fim de que a aplicação fique mais intuitiva e organizada
 - Configurar os botões da tela principal para só aparecer as opções permitidas ao usuário
- Eu como administrador quero adicionar e modificar permissões por tipo de usuário para que tenha mais liberdade de modifica-las futuramente.
 - Criar tela para editar permissões de usuário

Sprint TTT

Retrospectiva

- Como a gente esta usando um código auto gerado temos complicações em alguns merges e na própria programação.

Sprint Review

Gabes:

- Criou uma tela de permissão e está bonita porem não funcional.
- Criou uma classe de cada tipo de usuário
- Mudou o sistema de login

Joyce

- Terminou de configurar a opção edições
- Resolveu bugs e tratou exceções.

Thiago

- Fez a geração de arquivos em odt, pdf e ods.
- Fez o jComboBox da lista de itens poder pesquisar por letras porem com um bug a ser tratado ainda.

Vitor

- Fez modificações no banco de dados.

Sprint planning

- (1)Eu como usuário gostaria que todos os arquivos fossem salvos em um banco de dados para centralizar a informação.
- (15)Eu como usuário quero poder interagir com uma tela de menu a fim de que a aplicação fique mais intuitiva e organizada
- (16)Eu como administrador quero adicionar e modificar permissões por tipo de usuário para que tenha mais liberdade de modificar futuramente.
- (10)Eu como leitor quero poder filtrar os relatórios por mês de referência para facilitar minha busca

TTT

- configurar o BD nesse PC
 - configurar os botoes da tela principal para só aparecer as opções permitidas ao usuário
 - conectar todas as partes do código no bdmanager
 - conseguir gerar o relatório nos formatos de texto
 - terminar de colocar as informações na capa relatório
 - resolver bugs

Sprint 7

Retrospectiva:

- Como teve problemas com combustível prejudicou nossa organização da sprint. Sprint Review:

- Thiago e Joyce: Conectaram a maioria das classes ao banco de dados, tornando-o funcional de fato.

- Joyce: Configurou o menu principal para só aparecer as opções restrinidas ao usuário

- Thiago: Configurou telas de confirmação que verificam exceção no código. Sprint Planning:

- Testar as classes já criadas
 - Eu como usuário gostaria que todos os arquivos fossem salvos em um banco de dados para centralizar a informação.
 - Salvar os relatórios no banco de dados
 - Listar relatórios salvos no banco de dados
 - Conseguir representar um Relatorio no banco de dados
 - Eu como leitor quero poder filtrar os relatórios por escola para facilitar minha busca
 - Fazer um filtro por escola que funcione como um Contexto

- Eu como leitor quero poder filtrar os relatórios por mês de referência para facilitar minha busca
 - Implementar uma busca de relatórios por mes de referencia na tela de lista de relatorios
- Eu como diretor quero gerar relatórios nos formatos pdf, excel e odt para me adequar a qualquer situação.
 - Representar um relatorio em ods

Sprint 8

Retrospectiva:

- Joyce e thiago conseguiram cumprir com o que foi designado nessa sprint
- Gabriel teve complicações sérias com as matérias dele por questões de trabalho
- Vitor conseguiu iniciar os testes, mas acabou sendo muito mais complicado do que o esperado.

Sprint Review:

- Thiago e Joyce: conseguiram representar um relatório no banco de dados e implementar o filtro por escola.
- Vitor implementou o teste de duas das classes criadas.

Sprint planning:

- Testar as classes já criadas.
- Eu como leitor quero poder filtrar os relatórios por mês de referência para facilitar minha busca
 - Bug: Consertar código bugado da sprint 7 do gabes
- Eu como diretor quero gerar relatórios nos formatos pdf, excel e odt para me adequar a qualquer situação.
 - gerar um relatório em ods
 - gerar um relatório em odt
- Eu como diretor gostaria que cálculos usando dados pré-inseridos fossem feitos pelo sistema para que eu tenha mais praticidade.
 - Implementar cálculo percentual da capa relatório seguindo modelo do drive
- Melhorias:
 - Implementar a tela com a seleção de mês e de ano verificando a existência do relatório no banco

Sprint 9

Retrospectiva

- Joyce e thiago tiveram complicações pois tinham uma prova complicada para estudar e alguns trabalhos para fazer. Entretanto, pesquisaram intensivamente como utilizar XML para estilizar a formatação do ods, odt e pdf.

- Joyce e Vitor trabalharam nos testes unitários, progredindo um pouco mais, deixando o resto mais simples de dar continuidade

Sprint Review:

- Joyce e Thiago pesquisaram intensamente sobre como usar xml para exportar os arquivos.

- Joyce fez teste de mais 5 classes, corrigindo bugs;

- Vitor fez teste de mais 3 classes, corrigindo bugs, mas pesquisou como fazer os outros a fim de adiantar o trabalho.

- Gabriel reformulou a lógica da implementação do sistema de permissões para ficar menos redundante e mais conciso.

- Thiago inseriu o cálculo que ainda não estava sendo feito sobre os dados pré inseridos.

Sprint Planning:

- Testar classes já criadas (vitor)
- Gerar Relatório em ods (Joyce e Thiago)
- Gerar Relatório em odt (Joyce e Thiago)
- Criar opção de editar usuário (Gabriel)
- Criar opção de excluir usuário (Gabriel)
- Escrever Casos de Teste (Vitor)

Sprint 10

Retrospectiva

- Foi uma semana complicada devido ao fato de que a aula de quarta que usualmente era reservada para a realização e documentação da sprint teve outro objetivo principal. Sendo assim as estimativas das tarefas não ficaram tão fidedignas tendo em vista que nosso time não pôde se reunir fisicamente para fazê-las

Sprint Review

- Thiago implementou o cálculo do percentual aplicado a alguns valores do relatório, seguindo o modelo fornecido.
- Joyce e Thiago conseguiram finalmente exportar o relatório em ods e odt de maneira a conservar os estilos do modelo.
- Gabriel criou a opção de editar usuário e de excluir usuário.
- Vitor fez o testes de mais 3 classes.

Sprint Planning

- Implementar Caso de Teste (Joyce)
- Exportar relatório em pdf (Gabriel)
- Armazenar Relatorio Corrente no banco de dados (Joyce)
- Acertar a tela de lista de itens para corresponder ao modelo do drive (Joyce e Thiago)
- Resolver problema ao armazenar caracteres especiais no banco de dados (Vitor)
- Testar classes já criadas (Vitor)

Sprint 11

Retrospectiva

- Foi um período de provas finais o que complicou a organização do grupo como um todo, entretanto, aproveitamos o tempo para atualizar nossa wiki e resolver alguns problemas do código.

- Decidimos que não gerar em pdf devido ao tempo e complexidade.

Sprint Review

- Joyce atualizou a tela de lista de itens para se adequar ao modelo do drive
- Vitor colocou o exemplo de caso de teste na wiki

Sprint Planning

- Fazer o relatório sobre nossa trajetória, testes realizados, caso de teste e casos de uso (Joyce)
- Cada membro deve mandar um relatorio da sua visão do projeto para a Joyce
- Apresentação de slide (Thiago)
- Terminar a geração de ods e odt para seguir o modelo do drive (Thiago)
- Documentar testes na wiki (Gabriel)
- Documentação das classes (JAVADOCS)
- Terminar o máximo de testes (Vitor)