Programação e Desenvolvimento de Software 2

Trabalho Pratico Final

Alunos:	Matricula:
Hebert do Nascimento Amaral Costa	2016097439
Juliano Fernandes Soares Silva	2016026388
Rafael Lázaro Monteiro	2016435036

Sumário

1. INTRODUÇÃO

O Brasil atingiu em 2015 a marca de 79,9 milhões de toneladas de lixo urbano produzido. Mais de 50% desse lixo pode ser reciclado e uma boa maneira de selecionar esse tipo de resíduo é através da coleta seletiva. O objetivo deste trabalho é desenvolver um sistema de coleta seletiva, com funcionalidades para registrar doadores e coletores, combinar um ponto de encontro definir tipo de resíduo, etc. O trabalho nos permitiu integrar os conhecimentos adquiridos em sala de aula com atividade prática, permitindos-nos desenvolver as práticas de modelagem e programação orientada a objetos.

2. IMPLEMENTAÇÃO

O desenvolvimento pautou-se em etapas bastante conhecidas e consolidadas de desenvolvimento de Softwares, primeiramente desenvolveu-se as User stories que consistem em uma descrição alto nível das funcionalidades que serão encontradas no software pelos usuários durante a utilização do software, em conjunto com as User Stories foram desenvolvidos também os CRCs, que são os cartões de responsabilidade e mostram como as classes interagem entre si, e a responsabilidade de cada uma dentro do software.
Abaixo são mostradas as Users Stories.

Estoria: indice:

-Estoria: Agendamento de Coleta por Doador
-Estoria: Agendamento de Coleta por Receptor

-Estoria: Cadastro de Novo Tipo Residuos
-Estoria: Cadastro de novo Usuario

-Estoria: Computar coleta realizada

-Estoria: Editar cadastro de usuario
-Estoria: Editar Tipo Residuo Cadastrado

-Estoria: Listar/Remover tipo de residuo cadastrado
-Estoria: Listar/Remover usuarios cadastrados

-Estoria: Login de Usuario

-Estoria: Menu principal

-Estoria: Visualizar minhas Coletas agendadas

Estoria: Menu principal

- Como 01 usuario, eu desejo visualizar 01 menu com as opções de uso do sistema
- Neste procedimento eu devo poder
    -Visualizar & selecionar a opção de Fazer Login
    -Visualizar & selecionar a opção de Realizar cadastro
    -Visualizar & selecionar a opção de Sair do sistema (encerrar sessão)

Estoria: Cadastro de novo Usuario

- Como 01 usuario comum, eu desejo realizar o meu próprio cadastro no sistema
    - Neste procedimento eu vou
        - Informar SE meu cadastro sera de PF OU de PJ
            - SE meu cadastro for de PF, devo informar meu CPF
            - SENÃO devo informar meu CNPJ

        - Informar SE serei 01 doador ou 01 receptor de residuos
        - Informar meu nome
        - Informar os tipos de resíduos que eu tenho interesse em doar/receber, dentre a lista de tipos de resíduos cadastrados

Estoria: Login de Usuario

- Como 01 usuario, eu desejo realizar login dentro do sistema
- Neste procedimento eu devo informar o código do meu cadastro
- Caso o codigo informado seja valido, o sistema deve
    - Exibir uma mensagem de confirmação/boas vindas
    - Listar as ações disponíveis para mim

Estoria: Cadastro de Novo Tipo Residuos

- Como 01 usuario ADMINISTRADOR eu desejo cadastrar tipos de resíduo que poderão ser doados/recebidos através do sistema
- Neste procedimento eu devo poder informar
    - 01 codigo de identificacao para o tipo de residuo
    - Nome do tipo de residuo
    - SE ele eh subtipo de algum outro tipo previamente cadastrado
        - CASO seja subtipo, deve selecionar de qual dos tipos já cadastrados o novo tipo será subtipo

    - Forma de armazenamento adequada para o novo tipo/subtipo

Estoria: Listar/Remover tipo de residuo cadastrado

- Como 01 usuario ADMINISTRADOR eu desejo visualizar todos os tipos/subtipos de residuo cadastrados no sistema
    - Neste procedimento eu devo poder
        - Visualizar todos os tipos de residuos disponiveis no momento
        - Informar se deseja editar ou remover algum dos itens da lista
        - CASO selecione a opcao EDITAR
            - Selecionar o item da lista a ser editado

- CASO selecione a opcao REMOVER
    - Selecionar o item da lista a ser removido & concluir sua remocao

Estoria: Editar Tipo Residuo Cadastrado

- Como 01 usuario ADMINISTRADOR eu desejo atualizar o cadastro de 01 tipo de resíduo a ser doado/recebido através do sistema, já cadastrado por mim, anteriormente
- Neste procedimento eu devo poder alterar
    - Nome do tipo de residuo
    - Forma de armazenamento adequada para o tipo/subtipo
    - SE ele eh subtipo de algum outro tipo previamente cadastrado
        - CASO seja subtipo, deve selecionar de qual dos tipos já cadastrados ele deve ser subtipo

Estoria: Editar cadastro de usuario

- Como 01 usuario de qualquer tipo, eu desejo atualizar os dados do meu próprio cadastro
    - Neste procedimento eu poderei alterar
        - SE sou 01 doador ou 01 receptor de resíduos
        - meu nome
        - SE meu cadastro eh de PF OU de PJ
            - SE meu cadastro for de PF, devo informar meu CPF
            - SENÃO devo informar meu CNPJ

        - Informar os tipos de resíduos que eu tenho interesse em doar/receber, dentre a lista de tipos de resíduos cadastrados

Estoria: Listar/Remover usuarios cadastrados

- Como 01 usuario do tipo ADMINISTRADOR, eu desejo listar todos os usuários cadastrados no sistema
- Visualizar todos os usuarios cadastrados no momento
    - Informar se deseja editar ou remover algum dos itens da lista
    - CASO selecione a opcao EDITAR
        - Selecionar o item da lista a ser editado

    - CASO selecione a opcao REMOVER
        - Selecionar o item da lista a ser removido & concluir sua remocao

Estoria: Agendamento de Coleta por Doador

- Como 01 usuario do tipo DOADOR eu desejo agendar 01 doacao de resíduo
- Neste procedimento eu vou
    - Definir um código de identificação do agendamento a ser criado
    - Definir data prevista para a coleta agendada

- Selecionar qual(is) resíduo(s) vou doar, dentre a minha lista de resíduos (resíduos que posso doar)
- Selecionar para qual receptor eu vou fazer a doação
- Selecionar 01 Ponto de Coleta dentre a lista de pontos de coletas disponíveis, no momento

Estoria: Agendamento de Coleta por Receptor

- Como 01 usuario do tipo RECEPTOR eu desejo agendar 01 recolhimento de resíduo
- Neste procedimento eu vou
  - Definir um código de identificação do agendamento a ser criado
  - Definir data prevista para a coleta agendada
  - Selecionar qual(is) resíduo(s) desejo receber, dentre a minha lista de resíduos (resíduos que posso receber)
  - Selecionar de qual doador eu vou receber a doação
  - Selecionar 01 Ponto de Coleta dentre a lista de pontos de coletas disponíveis, no momento

Estoria: Visualizar minhas Coletas agendadas

Estoria: Computar coleta realizada

- Como 01 usuario doador ou receptor, eu desejo listar todos os agendamentos de coleta de resíduos nos quais eu esteja alocado
- Neste procedimento eu devo poder
  - Visualizar lista de agendamentos em que estou alocado
  - Informar se desejo editar, remover ou alterar o status de realização de algum dos itens da lista

- CASO selecione a opcao EDITAR
  - Selecionar o item da lista a ser editado

- CASO selecione a opcao REMOVER
  - Selecionar o item da lista a ser removido & concluir sua remocao

- CASO selecione a opcao ALTERAR STATUS de REALIZACAO
  - Selecionar o item da lista a o status alterado & concluir a alteracao

O código foi desenvolvido conforme o paradigma da orientação à objetos.
Alguns tipos de classes foram criadas. São eles:
Controller: Controladores tem como objetivo gerir a interação do usuário com o sistema. Eles guiam o usuário pela interface via linha de comando para realizar os procedimentos desejados

Dao: As classes do tipo DAO (Data Access Object) encapsulam funções que acessam dados armazenados, estabelecendo o contato entre o código, e o mecanismo de armazenamento. No caso deste projeto, o armazenamento é feito via escrita em arquivos de texto. De forma análoga a um banco de dados, cada arquivo de texto equivale a uma tabela, cada linha do arquivo corresponde a um registro e dentro de cada linha, os conteúdos separados por ‘;’ equivalem as colunas.
Service: o objetivo das classes de serviço é encapsular métodos que manipulam ou realizam algum procedimento relacionado a uma entidade específica.
Model: Os modelos estendem a interface (classe virtual pura) IModel e tem a função de definir as entidades (usuario, tipo de resíduo, agendamento) do sistema (modelo de dados).

A arquitetura do sistema é dividida em módulos que encapsulam as obrigações relativas a cada entidade do sistema. Os recursos da linguagem voltados ao paradigma orientado a objetos favoreceu a construção dessa arquitetura.

O sistema aplica conceitos vistos na disciplina tais como polimorfismo (exemplo: O controlador de usuários “UserController” é ora tratado como controlador genérico “Controller”, ora como um controlador específico). Este mesmo exemplo serve para demonstrar o uso de herança.

Outro conceito evidente em uso nesse projeto é o Encapsulamento. Os módulos separam os arquivos por relação com as entidades. E dentro de cada módulo, classes específicas dividem as tarefas associadas a cada uma dessas entidades.

Todo o sistema implementa tratamento de exceções. Toda interação com usuário é protegida contra entrada de dados inválida, por exemplo (ver mais na seção TESTES).

Funcionalidades Extra:

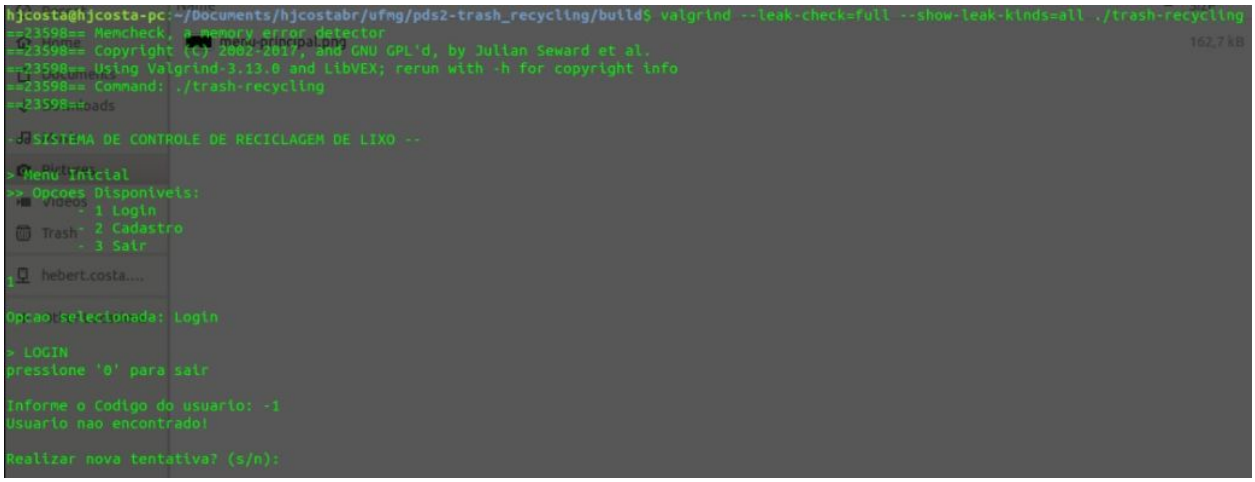
- Login: O sistema permite o cadastro de usuários dos tipos ‘doador’ ou ‘receptor’. Após realizar cadastro o usuário ganha acesso às opções disponíveis realizando login. O login consiste em informar o código de usuário escolhido pelo próprio do momento do seu cadastro. O sistema valida sua existência e notifica em caso do código ser inválido.
- Permissões: O sistema possui um usuário fixo do tipo ‘administrador’. Este usuário pode criar editar e remover pontos de coleta e tipos de

resíduo. Pode ainda remover cadastros de outros usuários. Todas essas opções são invisíveis para os outros usuários.

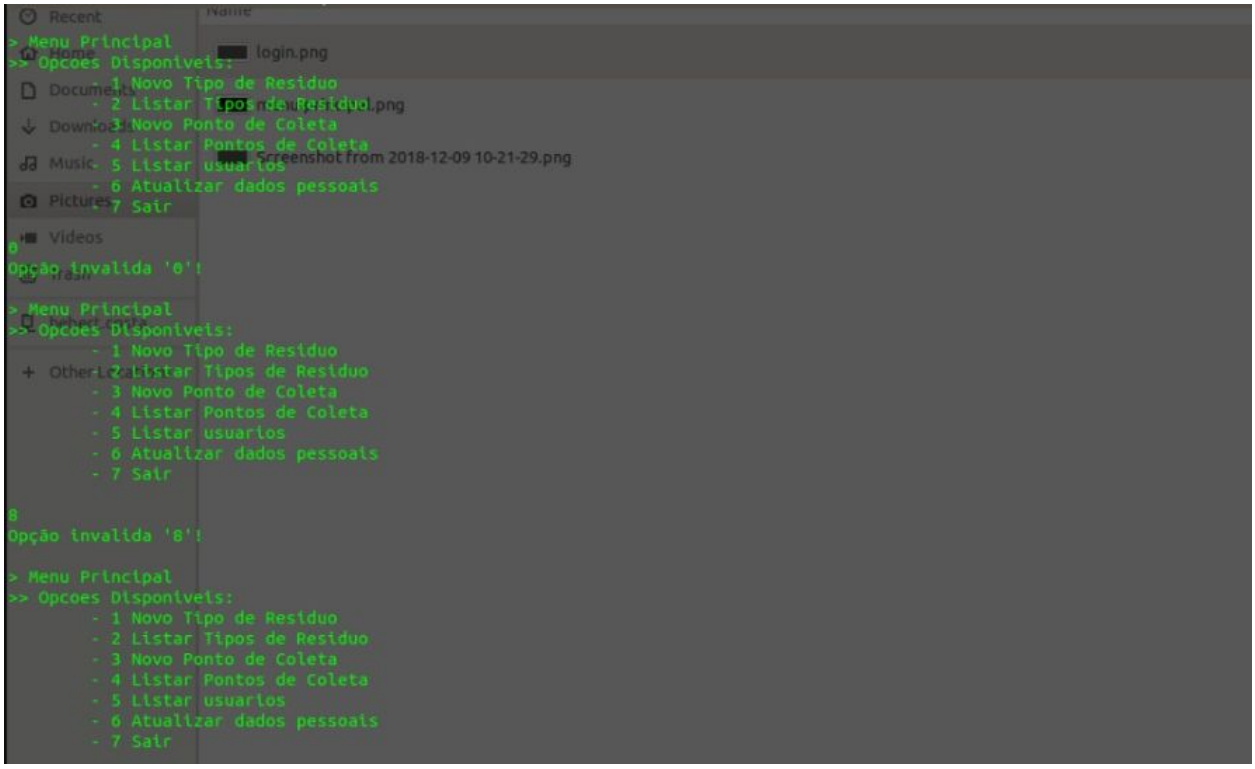
3. TESTES

Os testes são as principais ferramentas de validação das características do software que permitem garantir que este não apresente nenhum comportamento anomalo durante sua execução. Abaixo são mostrados alguns teste realizados no software que foi desenvolvido.

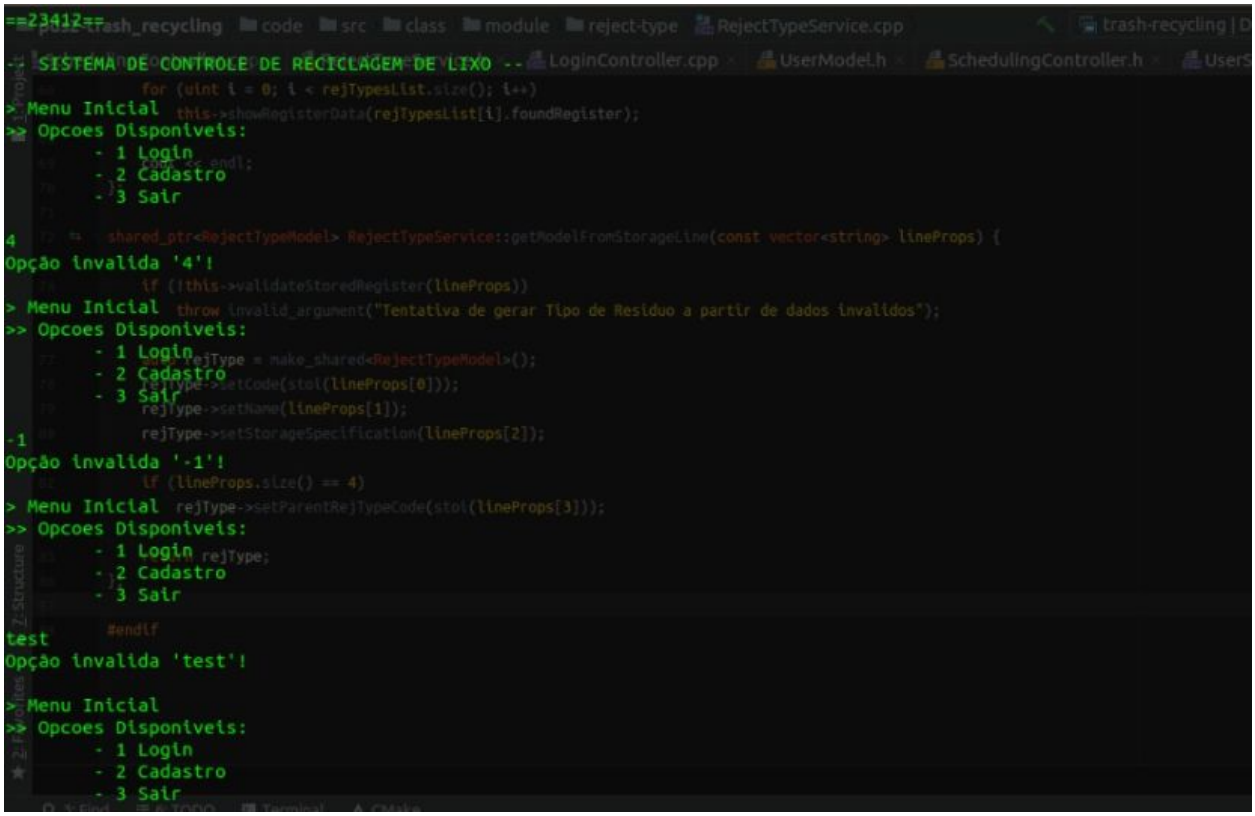
Tela de login quando o usuario insere um codigo invalido



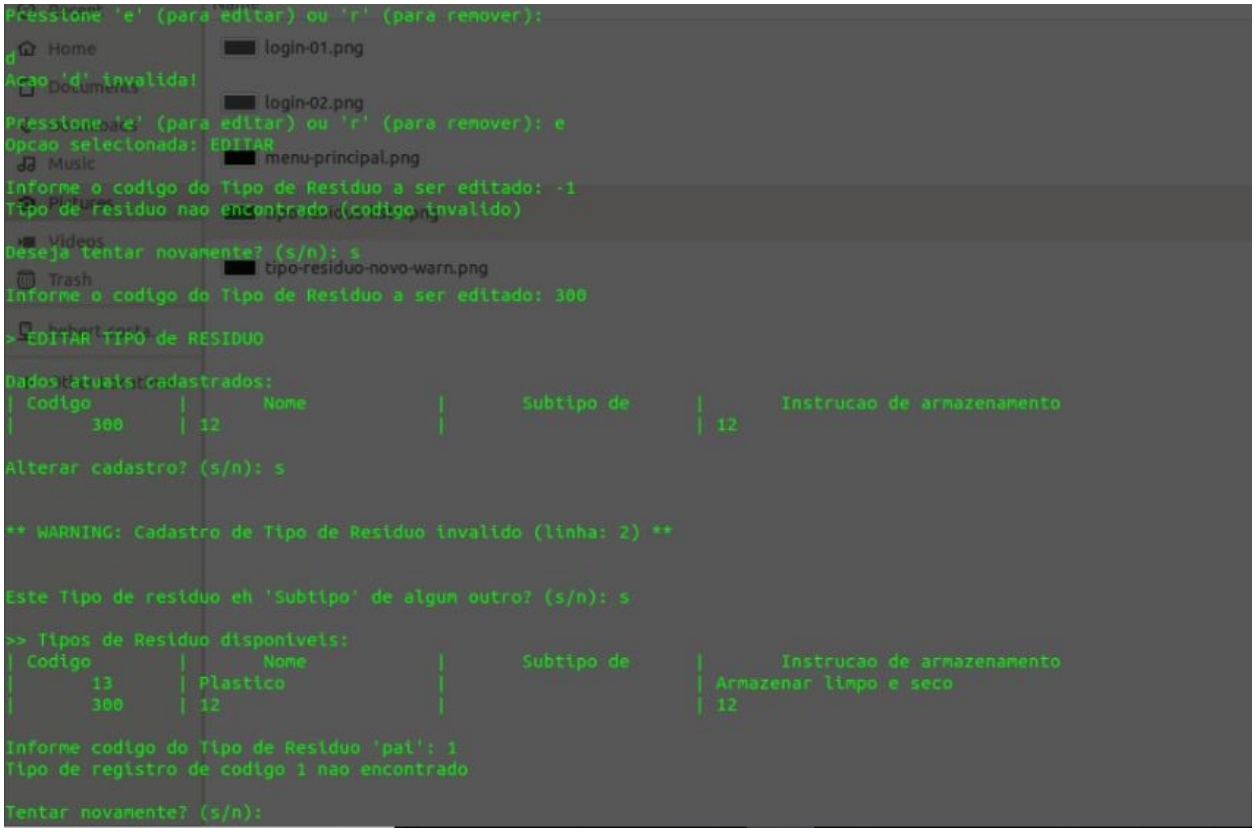
Menu principal do usuario administrador. Como o programa se comporta quando o usuario seleciona opções invalidas.



Menu inicial geral (para qualquer tipo de usuário) .Qual o comportamento do programa quando uma opcao invalida é informada.



Tela de edição de tipo de resíduo. Fica demonstrado a resposta do sistema para qualquer entrada de dados mal formatados. Na parte de baixo podemos ver um ‘warning’ alertando o usuario sobre um registro cadastrado errado no 'banco'


Pressione 'e' (para editar) ou 'r' (para remover):
Acao 'd' invalida!
Pressione 'e' (para editar) ou 'r' (para remover): e
Opcao selecionada: EDITAR
Informe o codigo do Tipo de Residuo a ser editado: -1
Tipo de residuo nao encontrado (codigo invalido)
Deseja tentar novamente? (s/n): s
Informe o codigo do Tipo de Residuo a ser editado: 300
> EDITAR TIPO de RESIDUO
Dados atuais cadastrados:
| Codigo | Nome | Subtipo de | Instrucao de armazenamento
| 300 | 12 | | 12
Alterar cadastro? (s/n): s
** WARNING: Cadastro de Tipo de Residuo invalido (linha: 2) **
Este Tipo de residuo eh 'Subtipo' de algum outro? (s/n): s
>> Tipos de Residuo disponiveis:
| Codigo | Nome | Subtipo de | Instrucao de armazenamento
| 13 | Plastico | | Armazenar limpo e seco
| 300 | 12 | | 12
Informe codigo do Tipo de Residuo 'pai': 1
Tipo de registro de codigo 1 nao encontrado
Tentar novamente? (s/n):
Home
Documents
Music
Videos
Trash
login-01.png
login-02.png
menu-principal.png
tipo-residuo-novo-warn.png

4. CONCLUSÃO

O presente trabalho nos deu a possibilidade de utilizar os conhecimentos adquiridos durante o semestre para a implementação de um programa que tem como objetivo a solução de um problema real enfrentado por uma grande das nações do mundo usando uma ótica realista de implementação.

A realização do projeto permitiu que parte de nós pudesse entender como funciona o processo de confecção de software e também que tivéssemos ideia de como é a vida de um programador que utiliza de seu conhecimentos para resolver os mais diversos problemas.

Além do citado acima nos permitiu contato e utilização das mais modernas e utilizadas técnicas de programação como a Programação Orientada ao Objeto que é um paradigma muito difundido e utilizado de programação.

5. BIBLIOGRAFIA

P.J. Deitel, H.M. Deitel. *“C++ : how to program.”* 8th Edition. ISBN 978-0-13-266236-9.
Cesar, j. *“Programação e Desenvolvimento de Software, Notas de Aula.”* . 2018.