

## Trabalho Prático Final

### 1 Introdução

O Brasil se tornou o quarto maior gerador de resíduos sólidos no mundo. A quantidade de lixo urbano produzida no país em 2015 atingiu 79,9 milhões de toneladas, 1,7% a mais do que no ano anterior<sup>1</sup>. Por causa disso, vivemos hoje em um ambiente onde a natureza é profundamente agredida. Entretanto, será que tudo que descartamos é realmente lixo? Em uma concepção moderna, o lixo se caracteriza por uma massa heterogênea de resíduos sólidos, resultante das atividades humanas, os quais podem ser reciclados e parcialmente utilizados, gerando, entre outros benefícios, proteção à saúde pública e economia de energia e recursos naturais<sup>2</sup>.

Mais de 50% do que chamamos lixo e que formará os chamados “lixões” é composto de materiais que podem ser reutilizados ou reciclados. O lixo é caro, gasta energia, leva tempo para decompor e demanda muito espaço. Mas o lixo só permanecerá um problema se não dermos a ele um tratamento adequado. Você já ouviu falar em coleta seletiva?

Um projeto de coleta seletiva de lixo pode ser uma boa forma de destinar resíduos, como óleos de cozinha, lâmpadas, medicamentos, eletrônicos, pilhas e baterias, adequadamente. E você, como um futuro engenheiro de sistemas, pode contribuir muito neste cenário.

#### 1.1 Objetivo

O nosso objetivo neste trabalho é desenvolver um sistema que auxilie moradores, empresas, condomínios e escolas, na implantação de um projeto de coleta seletiva. Ou seja, dado que existe um usuário com a necessidade de descartar um resíduo de forma adequada, e outro que se propõe a realizar esse descarte (ou reutilização) de forma correta, o seu sistema deve intermediar essa negociação de forma amigável e confiável. O interessante é que além de uma contribuição social este projeto nos permitirá, ao mesmo tempo, exercitar todo o conteúdo visto durante o curso, incluindo desde o processo de desenvolvimento (análise, projeto e implementação), até conceitos e técnicas de programação como modelagem, POO, testes unitários, etc. É importante ressaltar que seu programa deverá ser feito em linguagem C++.

### 2 Especificação das Funcionalidades Gerais Básicas

Basicamente, nosso sistema deverá permitir:

- Cadastro de Usuário: Deve ser implementado um CRUD<sup>3</sup> básico para se criar contas no sistema. Essas contas podem ser de usuários doadores (que pode ser uma pessoa física, ou jurídica, no caso de um condomínio, escola, ou empresa, por exemplo. Esses usuários são responsáveis por reunir os resíduos a serem recolhidos) e receptores ou recolhedores (que também podem ser pessoas físicas, como catadores de papel, e pessoas jurídicas, como associações relacionadas). É importante que doadores e receptores indiquem, a partir de uma lista pré-definida, os tipos de resíduos de interesse *Dica! Este pode ser um bom cenário para aplicarmos o conceito de herança.*
- Cadastro de Resíduos: Deve ser implementado um CRUD básico para cadastro dos tipos de resíduos no sistema (como por exemplo, líquidos e sólidos, e dentre os sólidos nos poderíamos ter metais, vidros, plásticos, etc). Isso é importante pois o usuário só pode doar ou recolher um tipo de resíduo pré-cadastrado no sistema. Além disso, ao informar que tipo de resíduos irá doar futuramente (durante o cadastro), o sistema deverá fornecer ao usuário uma explicação sobre a forma adequada de armazenamento daquele resíduo. Ou seja, caso o usuário deseje juntar óleo de cozinha (um tipo de resíduo líquido) para o descarte adequado, como este produto deve ser armazenado? *Dica! Talvez esse seja novamente um bom momento para aplicarmos o conceito de herança e introduzirmos o conceito de polimorfismo.*
- Cadastro do Ponto de Coleta: o local físico para doação e recepção do resíduo a ser destacado é algo que precisa ser combinado. Este local pode ser o espaço físico onde qualquer um dos usuários envolvidos no processo resida, ou um local qualquer a ser combinado.

<sup>1</sup><https://www.akatu.org.br/noticia/volume-de-lixo-produzido-no-brasil-aumentou-17-em-2015/>

<sup>2</sup><http://www.projutoreciclar.ufv.br>

<sup>3</sup>[https://en.wikipedia.org/wiki/Create,\\_read,\\_update\\_and\\_delete](https://en.wikipedia.org/wiki/Create,_read,_update_and_delete)

- Agendamento/Realização da Coleta: por fim, o sistema deve permitir a realização da coleta do resíduo de fato. Para isso, deve haver uma data, um local, dois usuários envolvidos (doador e receptor) e um ou mais resíduos que serão recolhidos para o descarte automático. Deve haver ainda um identificador booleano que sinalize se a doação/coleta foi ou não realizada de fato.
- Funcionalidade Extra: o grupo poderá ainda propor, caso deseje, alguma funcionalidade não básica, que irá representar o aspecto de inovação do projeto. Como esta etapa não é obrigatória, ela irá compor uma pontuação extra em relação ao trabalho prático aqui proposto.

Como aprendemos em sala de aula, as User Stories<sup>4</sup> são uma forma simples de apresentar os requisitos funcionais desejados para um determinado sistema. São artefatos de desenvolvimento utilizados principalmente em processos baseados em metodologias ágeis. As descrições são intencionalmente genéricas, dando liberdade ao grupo para decidir detalhes da implementação. Logo, o seu grupo deverá identificar e decidir detalhes de implementação de user stories que garantam essas funcionalidades básicas do sistema.

Lembre-se, o objetivo não é apenas escrever um programa funcional, mas desenvolver um sistema confiável, reutilizável e de fácil manutenção e extensão! Logo, tente utilizar os conceitos de orientação a objetos e modularidade vistos em sala de aula. Também serão avaliados critérios como criatividade na solução, assim como a possível implementação de funcionalidades extras.

### 3 Documentação

Um relatório deve ser entregue ao final de forma a deixar claro as decisões tomadas relativas aos casos e detalhes que porventura estejam omissos no enunciado. Adicione também comentários gerais sobre o trabalho e as principais dificuldades encontradas.

Além disso, dentre outras coisas o seu relatório deve conter:

1. Introdução: descrição geral sobre o funcionamento do seu programa;
2. Implementação: descrição sobre a implementação do programa. Devem ser detalhadas as estruturas de dados utilizadas, o funcionamento das principais funções, bem como decisões tomadas. Possível descrição de extras implementados. É importante você informar neste momento como os conceitos de OO foram aplicados no seu sistema;
3. Testes: proponha testes para validar o funcionamento do sistema e apresente aqui os resultados. Pode por exemplo apresentar prints dos menus, das saídas, etc., sempre relacionando com as partes correspondentes do código.
4. Conclusão: comentários gerais sobre o trabalho e as principais dificuldades encontradas em sua implementação.
5. Bibliografia: bibliografia utilizada para o desenvolvimento do trabalho, incluindo sites da Internet se for o caso. Não tem problema olhar códigos na Internet, contanto que você não copie :)

### 4 Comentários Gerais

- Comece a fazer este trabalho logo: o prazo para terminá-lo está tão longe quanto jamais poderá estar! :)
- O programa fonte deve ser claramente comentado;
- Todos os integrantes do grupo devem conhecer a totalidade do código e ter condições de explicar sua função;
- O trabalho deverá ser feito em grupos de no mínimo 3 e no máximo 4 alunos;
- Trabalhos copiados e entregues com atraso serão penalizados;
- Deve ser fornecido com o código um arquivo Makefile com as opções “make” e “make run”;
- A entrega final deverá ser feita utilizando GitHub: cada grupo deve criar um repositório privado no GitHub e me adicionar como colaborador (usuário: julioares.reis). A correção será feita até o último commit antes da data limite para entrega (a ser definida);
- Mais detalhes sobre a entrega poderão ser fornecidos posteriormente.

---

<sup>4</sup>[https://en.wikipedia.org/wiki/User\\_story](https://en.wikipedia.org/wiki/User_story)

## 5 Critérios de Avaliação

### 5.1 Entrega Parcial

- Definição e proposição das Users Stories/Cartões CRC (6 pts);

### 5.2 Entrega Final

- Documentação: clareza, organização, etc (5 pts);
- Funcionamento correto: compila, executa, etc (10 pts);
- Uso correto de boas práticas e dos conceitos de OO (5 pts);
- Testes: proposição de testes, comportamento adequado do sistema dado comportamento inesperado do usuário, etc (4 pts);
- Funcionalidade Extra (até 3 pts).