

UNIVERSIDADE FEDERAL DE MINAS GERAIS
ESCOLA DE ENGENHARIA
CURSO DE GRADUAÇÃO EM ENGENHARIA DE SISTEMAS

Programação e Desenvolvimento de Software II

Trabalho Prático Final

ChatBot

Pedro Herculano Flores

Thiago Luiz Rodarte

Richard Wagner Abras Ribeiro Sobrinho

Ricardo Nascimento de Oliveira

Renato Rodrigues Pereira

2019/2

1. INTRODUÇÃO

O objetivo deste trabalho foi desenvolver um código que possibilitasse facilitar os procedimentos necessários para a doação de sangue. Esses procedimentos envolvem, entre outros, a triagem dentre os candidatos a doadores, daqueles que estão aptos e daqueles que não estão aptos a doar; o agendamento de uma doação e a administração de um cadastro de doadores e pontos de coleta. Para atender a essas necessidades foi desenvolvido um programa onde o candidato a doador consegue facilmente realizar procedimentos de cadastro e triagem clínica. Além disso, é possível para administradores cadastrarem pontos de coleta de sangue.

2. IMPLEMENTAÇÃO

Cadastro:

O cadastro de doadores foi definido como uma classe contendo os seguintes atributos: senha (private); número de contato, nome, endereço e login (protected). O construtor e o destrutor da classe foram definidos como public.

Ponto de Coleta

O ponto de coleta foi definido como uma subclasse do cadastro tendo como atributo o CNPJ.

Doador

Também foi definido como uma subclasse da classe cadastro. Tem como atributo o CPF do doador, que será utilizado como login do doador para o acesso.

Agendamento

Foi definido como uma classe com os seguintes atributos: doador, ponto de coleta, e a data (todos private); Possui gets para todos os seus atributos. Foi criada uma lista de agendamento para armazenar os dados dos agendamentos realizados.

Lista controller, lista_empresa , lista doador e lista agendamento

Foi criado o list controller, como uma classe abstrata, contendo diversos métodos vazios. A partir dele foi criado o lista doador e lista empresa, que servem como um banco de dados para os doadores e pontos de coleta cadastrados. Através dessa classe é possível: criar dados, altera-los, deleta-los e os detalhar.

Com essas listas implementadas, foi criado o lista agendamento, que utiliza os dados incluídos nas listas doador e empresa para marcar o dia da retirada de sangue.

Main

Na main é apresentado o ChatBot, que irá “dialogar” com o usuário final, que é o candidato a doador.

Ao compilar e rodar o programa, a primeira opção dada é a de escolher entre administrador e usuário. Ao escolher a opção de administrador, há uma série de opções relativas à edição de dados. São elas: criar novo local de coleta, alterar dados, deletar local de coleta, deletar dados, mostrar todos os locais de coleta. Em todas elas é solicitado que o administrador insira login e senha.

Ao escolher a opção de usuário, há três opções. A primeira delas é a opção de doador, onde o usuário pode se cadastrar, alterar, deletar ou detalhar seus dados. A segunda opção abre o ChatBot. Nesse caso são feitas várias perguntas ao usuário, que caso dê uma resposta que não esteja de acordo com os parâmetros necessários para a doação é informado de que não preenche os requisitos para doar sangue. Caso atenda a todos os critérios é informado de que aparentemente é elegível para doar sangue e notificado de que deve comparecer ao ponto de coleta mais próximo para se assegurar de que pode doar. A terceira opção disponível para o usuário é o agendamento de doação. Neste caso há três opções: fazer um novo agendamento, deletar um agendamento já marcado ou detalhar os dados.

3. TESTES

Para testar, basta executar todos os arquivos .cpp no prompt de comando e criar um arquivo executável. Ao abrir esse executável, irá aparecer as opções de doador e de admin. Para utilizar as funções do trabalho é preciso criar pontos de coleta e doadores (exceto o chatbot, que é utilizável mesmo sem cadastro). Uma observação importante é

que, ao colocar letras e símbolos no número de telefone, o programa gera um erro, já que a variável referente ao mesmo é um int.

4. CONCLUSÃO

A elaboração desse trabalho proporcionou a utilização de vários conceitos aprendidos durante o curso de PDS II, e aplicar alguns dos conceitos de POO, como por exemplo: classes, classes herdeiras, classes abstratas, listas utilizando o STL, modularização, entre outros. Outro ponto positivo da realização do projeto foi a descoberta de diversos sites qualificados na área de programação em C++, já que para a realização do mesmo foi necessário pesquisa para resolver diversos problemas e para o entendimento de diversas funções.