

# Relatório do projeto final

Artur Amaral

15 de Outubro de 2021

Apresentação:

[https://www.youtube.com/watch?v=Z\\_N8ShZ6GJ4](https://www.youtube.com/watch?v=Z_N8ShZ6GJ4)

## 1. Introdução

Minha proposta de projeto de projeto final para a disciplina é a simulação de um simples sistema de inscrição em disciplinas de uma universidade, neste trabalho nomeado a partir de um nome da própria UFRJ, o SIGA.

O usuário interage com o programa a partir do terminal, executando um binário C++ que fornece um Menu para realizar operações pertinentes ao sistema. Após o término da execução do programa, a persistência dos dados é implementada em um banco de dados MySQL, como parte do requisito realizar uma interface entre duas linguagens de programação do roteiro do trabalho.

## 2. Dependências

O programa foi desenvolvido e testado em um sistema operacional **Ubuntu 20.04** e compilado com a ferramenta **g++ versão 9.3.0** no **padrão C++11** da linguagem. O banco de dados **MySQL** utilizado foi da **versão 8.0.26**. A biblioteca utilizada para realizar a interface entre C++ e MySQL foi o pacote **libmysqlclient-dev**, obtido diretamente dos repositórios remotos de software do Ubuntu. Para instalar esse dependência, basta utilizar a ferramenta “apt”:

```
bash$ sudo apt update
```

```
bash$ sudo apt install libmysqlclient-dev
```

O projeto foi apenas testado nas configurações de ambiente e ferramentas descritas acima, não tendo garantia de funcionar em plena harmonia caso as mesmas sejam diferentes. Como não era o foco do trabalho, não foquei em preparar um ambiente “de produção” para a execução do mesmo, mas também por falta de tempo, pois sei que atualmente é relativamente simples criar um desses usando ferramentas de containerização como Docker.

O programa já pressupõe a existência de um banco de dados chamado “siga” e de um usuário chamado “eel670”, cuja senha é “eel670”. Sendo assim, essas

configurações devem ser realizadas manualmente. Esta tarefa é relativamente simples. Segue o processo:

```
bash$ sudo mysql -u root
~Inserir senha do root do sistema~
mysql> CREATE DATABASE siga;
mysql> CREATE USER 'eel670'@'localhost' IDENTIFIED BY 'eel670';
mysql> FLUSH PRIVILEGES;
mysql> GRANT SELECT ON siga.* TO 'eel670'@'localhost';
mysql> FLUSH PRIVILEGES;
```

### 3. Compilação e execução

O programa conta com um Makefile para compilação e linkedição automática do código fonte. Há quatro comandos possíveis de serem realizados:

```
bash$ make
bash$ make siga    (Este e o acima geram o executável do programa.)
bash$ make clean   (Deleta o executável.)
bash$ make clean-objs (Deleta arquivos objeto.)
```

O executável encontra-se na pasta bin/. Para executá-lo, digite:

```
bash$ ./bin/siga_app
```

### 4. Implementação

O programa têm três classes principais, que são **Aluno**, **Disciplina** e **SIGA**. As classes Aluno e Disciplina basicamente abstraem os atributos relevantes que suas contrapartidas reais possuem e são importantes para o sistema, como nome, DRE, nº de vagas, etc. Mais especificamente a classe Disciplina possui um atributo que é um vetor de "alunos inscritos", que armazena os alunos que garantiram inscrição na mesma. A ideia destas duas classes é servir de modelo para a instanciamento de diversos objetos, cada um representando um aluno ou disciplina únicos, em contrapartida com a classe SIGA, cujo uso presume a instanciamento de um único objeto, que encapsula todos os alunos e disciplinas. Uma leitura detalhada dos arquivos header deve ser suficiente para compreender, mesmo que informalmente, boa parte de seus funcionamentos.

Meu objetivo desde o princípio foi manipular as estruturas de dentro, ao máximo que desse, dentro dos domínios do C++, sendo a interface SQL o mais simples e reduzida possível. Sendo assim, a interação com o banco de dados é realizada em apenas dois momentos no programa: Ao início e ao fim de sua execução, através dos métodos carregarDados() e salvarDados(), da classe SIGA. Existe também métodos de criar tabela e esvaziar tabela, mas o objetivo é mais serem uma ferramenta para utilização do administrador do sistema( no caso, eu).

Ao executar a opção [5] do menu, um pedido de inscrição, encapsulado em uma estrutura própria, é criado e adicionado a um vetor de “pedidos pendentes” da classe SIGA. Ao executar a opção [6], o sistema analisa os pedidos pendentes e inscreve os alunos nas respectivas disciplinas, isto é, insere os objetos Aluno dentro de vetores nos objetos Disciplina. Ao fim da execução do programa, através do Destrutor, uma Query em SQL é realizada e todos os alunos que estão encapsulados em objetos disciplinas são inscritos nas respectivas tabelas que as representam no banco de dados.

## **5. Conclusão**

O sistema implementado fez uso do paradigma da orientação a objetos na medida que me pareceu necessário. Os principais conceitos da disciplina utilizados foram abstração, encapsulamento, sobrecarga de operadores e tratamento de exceção, além de diversos métodos e práticas ensinados ao longo da mesma. Sendo assim, dadas as funcionalidades propostas na primeira parte do trabalho, sinto que a implementação do núcleo do sistema foi satisfatória, embora eu não tenha conseguido testá-lo tanto quanto gostaria. A interface com o banco MySQL também provou-se bastante simples de ser utilizada.