

## 1 Resumo

Criar um programa em C++ utilizando conceitos de orientação à objetos que implemente um sistema de gerenciamento de pessoas, disciplinas e atividades relacionadas a cursos na universidade.

## 2 Detalhes

O sistema deverá ser implementado utilizando os conceitos de orientação a objetos aprendidos em sala de aula e atender, pelo menos, aos requisitos descritos nas seções a seguir:

### 2.1 Telas de cadastro

Deverão ser criadas telas de cadastro para inclusão de registros primários para o sistema.

Deverá haver uma tela de cadastro de disciplinas, uma tela de cadastro de cursos (que deverão ser vinculados a disciplinas e um coordenador) e uma tela de cadastro de pessoas.

A tela de cadastro de pessoas deverá permitir inserção, remoção, alteração, consulta e listagem de pessoas. Nessa tela, deverá ser possível informar os dados pessoais. Nessa tela, deverá haver um campo para indicar o nome de usuário e senha que a pessoa utilizará para acessar (logar) o sistema.

Uma outra tela (ou uma funcionalidade da mesma tela de cadastro) deverá possibilitar a associação da pessoa com os perfis que ela poderá assumir. Uma pessoa poderá assumir os perfis Aluno, Professor, Coordenador, Diretor, Funcionário Administrativo. A partir dos perfis, o sistema deverá apresentar telas compatíveis aos perfis que o usuário logado poderá acessar.

### 2.2 Perfis

Todos os usuários logados terão acesso a informações básicas a respeito de seus acessos e terão uma interface de *chat* para enviar mensagens *off-line* para outros usuários.

Qualquer pessoa pode ser vinculada a qualquer quantidade de perfis.

Os perfis deverão ser escolhidos para cada pessoa cadastrada, de modo que quando a pessoa acessar o sistema, ela deverá ter acessos de acordo com seus perfis. Seguem as restrições básicas a respeito desses acessos.

#### 2.2.1 Funcionário Administrativo

O funcionário administrativo será o único perfil que terá acesso para o cadastro de pessoas, cursos e disciplinas, cabendo-lhe cadastrar as pessoas e lhes atribuir os devidos perfis. A única exceção é o perfil de Coordenador de curso, que será possível de ser definido apenas por um diretor. O perfil de funcionário administrativo gera um salário fixo de valor armazenado no próprio perfil.

#### 2.2.2 Diretor

O diretor terá acesso para definir quais serão os coordenadores dos cursos. O diretor deverá ter acesso a uma tela de listagem de todos os coordenadores de curso e deverá ter uma tela para exibir um resumo de totais de aprovados e reprovados por disciplina/curso. O perfil de Diretor gera um bônus de R\$ 100,00 para cada curso administrado no sistema.

#### 2.2.3 Coordenador

O coordenador terá acesso para criar turmas e trancar matrículas de alunos do curso que ele coordena. O coordenador deverá ter acesso a uma tela contendo uma listagem de todos os professores do curso que coordena. O coordenador deverá ter acesso de leitura às atividades e notas de todos os alunos de cursos que ele coordena. O Coordenador terá um bônus de R\$ 1,00 para cada aluno matriculado no curso que ele coordena.

#### 2.2.4 Professor

O professor terá acesso para criar atividades e atribuir notas aos alunos. O professor deverá ter acesso a uma tela contendo listagem de todos os alunos por turma. Nessa listagem, deverão vir as atividades e notas de cada aluno em cada atividade. O professor tem um salário fixo em R\$ 150 por cada crédito de disciplina ministrada no semestre.

#### 2.2.5 Aluno

O aluno terá acesso para visualizar tudo postado nas turmas em que ele estiver matriculado. O aluno deverá ter acesso às suas notas nas atividades aplicadas. Atividades não executadas deverão aparecer com nota zero.

## 2.3 Atividades

A criação de atividades deverá ser feita por professores em suas turmas, de modo que o professor possa gerenciar de forma clara as atividades de cada turma. Após a criação das atividades, os alunos da turma em questão terão acesso de leitura às atividades dessa turma.

## 2.4 Salários

Todos os perfis de funcionários deverão ter salários. O valor total do contracheque de cada pessoa será dado pela soma de todos os valores recebidos em cima de cada perfil. Como já mencionado, cada perfil de funcionário tem uma forma de calcular o rendimento que aquele perfil lhe garante. O único perfil que não terá salário é o de Aluno. Vale ressaltar que, como uma pessoa pode ter vários perfis, podem haver pessoas com perfil de aluno mas que também sejam funcionários remunerados.

## 2.5 Envio e recebimento de mensagens

Deverá haver uma tela de interação com outros usuários que todos os usuários poderão ter acesso. Nessa tela, deverá haver uma forma de convidar um outro usuário para uma relação de amizade, de modo que, após a aceitação, ambos poderão trocar mensagens.

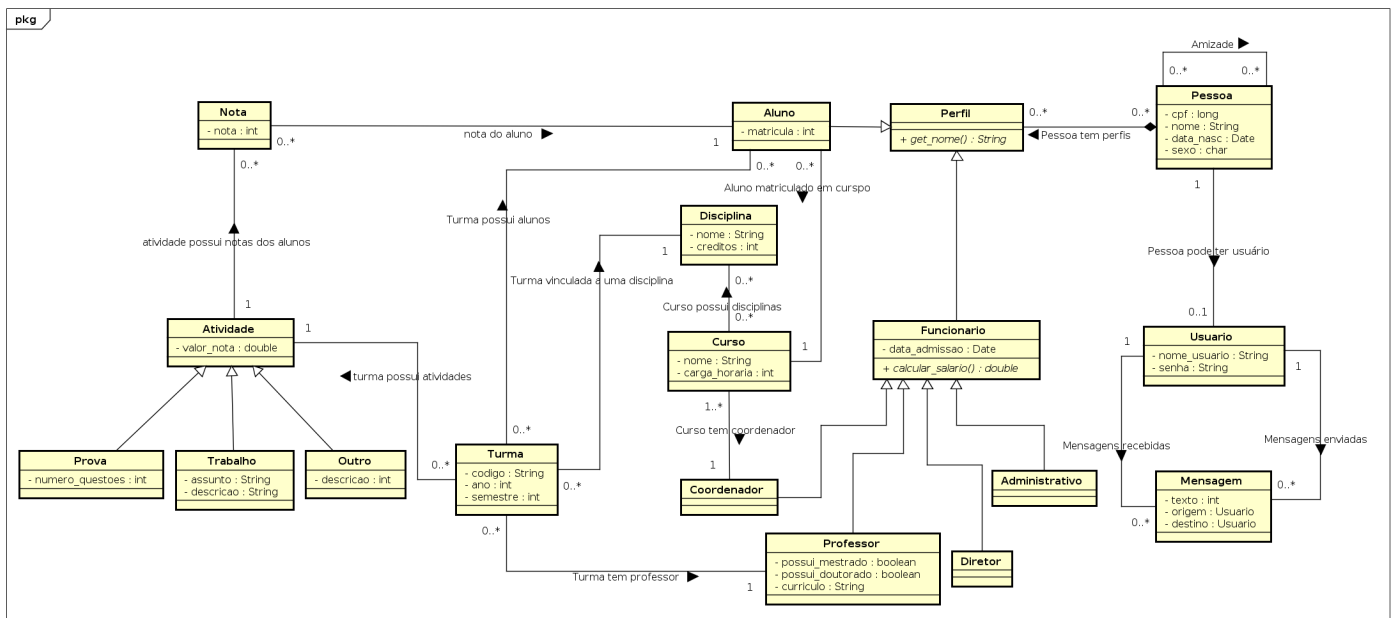
Essa troca de mensagem deverá ocorrer de forma *off-line*. Assim, se um usuário envia uma mensagem para outro, quando o segundo usuário acessar o sistema, ele poderá ver as mensagens recebidas.

# 3 Implementação

O programa deverá ser totalmente escrito em qualquer linguagem de programação orientada a objeto. Caso o aluno tenha interesse, poderá utilizar bibliotecas de interfaces gráficas ricas, como QT, GTK ou OpenGL.

## 3.1 Orientação à objetos

O programa deverá ser desenvolvido utilizando os conceitos de orientação à objetos. Segue um diagrama de classes (incompleto) que pode dar uma ideia de como organizar algumas classes do sistema:



## 3.2 Persistência

O sistema deverá ser capaz de armazenar todas as informações em memória secundária, ou seja, gravar no HD. Isso poderá ser feito com o uso de arquivos ou banco de dados.

Além disso, o sistema deverá fornecer uma ferramenta de inicialização do sistema por meio de um arquivo de importação. Assim, o sistema deverá ser capaz de criar diversos registros (como pessoas, cursos, disciplinas, atividades, etc) por meio de importação de um arquivo.

**O layout desse arquivo de importação ainda será apresentado na próxima versão desse documento.**

## 4 Entrega

O trabalho deverá ser entregue em duas etapas: a primeira é a entrega dos arquivos de código fonte via e-mail; a segunda é a apresentação do trabalho em sala de aula diretamente para o professor.

### 4.1 Código fonte

Os arquivos de código fonte deverão ser compactados em um único arquivo (ZIP, RAR, 7Z, etc.) e enviados até a data estipulada em sala de aula.

### 4.2 Apresentação

Todos os alunos deverão apresentar o trabalho ao professor em sala de aula na data estipulada em sala de aula.

Nenhum trabalho será avaliado após essa data.

Se o aluno quiser apresentar antes da data final, poderá fazê-lo. Basta informar e agendar com o professor.

A apresentação será INDIVIDUAL e irá considerar a capacidade do aluno de explicar todo o código desenvolvido ou partes específicas determinadas pelo professor.

## 5 Avaliação

A avaliação será realizada em duas etapas: uma referente à entrega do programa em si; a outra referente à apresentação.

A entrega vai considerar o respeito ao prazo, o uso de orientação a objetos de forma correta, o uso de persistência no sistema e a criação da ferramenta de importação de dados.

A apresentação vai considerar a explicação do aluno acerca do código desenvolvido e a capacidade de o aluno responder perguntas sobre o código e/ou os algoritmos. O professor poderá ainda solicitar alterações no código que o aluno deverá ser capaz de realizar.