

Documento de Análise de Requisitos - Sistema de Controle Acadêmico

Guilherme Silva Felix
José Cassimiro Toledo Júnior
DECEA/ICEA – UFOP

20 de novembro de 2015

1 Introdução

O sistema de controle acadêmico será um sistema desenvolvido em PHP utilizando a arquitetura MVC e o framework Symfony2. Poderá ser utilizado posteriormente como sistema padrão para professores e alunos do ICEA.

Este sistema tem o intuito de facilitar a gestão de turmas para os professores, bem como melhorar a disponibilidade de notas e frequências para os alunos.

2 Descrição do Problema

Na UFOP (campus João Monlevade) ainda não foi desenvolvido nenhum sistema padrão para controle acadêmico. Atualmente, cada professor utiliza um site qualquer de sua preferência para lançar notas e frequências, como o Google Sites ou a plataforma Moodle, por exemplo, e fazem chamadas através de planilhas. Isso acaba gerando uma ineficiência na hora de propagar as informações aos alunos, além de dificultar a vida do professor.

3 Objetivos

O trabalho tem por objetivo projetar e desenvolver um sistema baseado na web para apoiar a UFOP em sua gestão acadêmica. Neste sentido, o trabalho possui os seguintes objetivos específicos:

- Produzir um sistema padrão para gestão acadêmica;

- Permitir aos alunos acompanhar seu desempenho em disciplinas e no curso em geral, em tempo real;
- Facilitar o trabalho dos professores na gerência da turma, podendo em um mesmo sistema: fazer chamadas, adicionar atividades como provas e trabalhos, bem como suas respectivas datas e notas.

4 Escopo da Aplicação

Este documento especifica todos os requisitos funcionais e não funcionais do sistema além de requisitos futuros. O Projeto visa permitir que os alunos consultem suas notas e faltas de forma rápida e eficiente.

Para estabelecer a prioridade dos requisitos foram adotadas as seguintes denominações:

Tipo	Descrição
Essencial	É o requisito sem o qual o sistema não entra em funcionamento. Requisitos essenciais são requisitos imprescindíveis, que têm que ser implementados impreterivelmente.
Importante	É o requisito sem o qual o sistema entra em funcionamento, mas de forma não satisfatória. Requisitos importantes devem ser implementados, mas, se não forem, o sistema poderá ser implantado e usado mesmo assim.
Desejável	É o requisito que não compromete as funcionalidades básicas do sistema, isto é, o sistema pode funcionar de forma satisfatória sem ele. Requisitos desejáveis são requisitos que podem ser deixados para versões posteriores do sistema, caso não haja tempo hábil para implementá-los na versão que está sendo especificada.

Tabela 1 – Denominações

4.1 Benefícios do Produto

Benefício	Relevância
Agilidade no processo de gerenciamento das frequências.	Essencial
O professor não precisará atualizar o site constantemente.	Essencial
O aluno poderá ver o número de faltas rapidamente a qualquer momento.	Essencial

Tabela 2 – Benefícios do Produto

4.2 Requisitos Funcionais (**Essenciais**)

1. Login no Sistema.
2. Cadastro de disciplinas pelo professor.
3. Marcar frequência dos alunos.
4. Disponibilizar notas de provas.
5. Disponibilizar notas de atividades.
6. Notificar o aluno quando limite de faltas estiver próximo do limite.
7. Cadastro de disciplinas pelo aluno.
8. Acesso ao número de faltas pelo aluno.
9. Acesso às notas de provas pelo aluno.
10. Acesso às notas de trabalhos pelo aluno.

4.3 Requisitos não Funcionais (**Importantes**)

1. **Segurança:** Software deve garantir a segurança dos dados, bem como as permissões de acesso às suas funcionalidades.
2. **Usabilidade:** Procurar desenvolver um sistema fácil de operar e que dispense muitos recursos gráficos. Se possível, adicionar descrições das funções (*hints*) aos botões e configurar teclas de atalho para as funções mais utilizadas.
3. **Confiabilidade:** Determina a capacidade do sistema em lidar com eventos inesperados. Suponha que o professor esteja cadastrando um novo registro, e após inserir todas as informações, ocorre um erro no sistema e o professor acaba perdendo as informações digitadas.

4. **Padrão:** Define a padronização de interface e código utilizada no desenvolvimento do software.
5. **Desempenho:** De nada adianta ter um sistema seguro, interativo e confiável se ele consome muitos recursos do computador e demora pra executar os processamentos.

4.4 Requisitos Futuros (Desejáveis)

1. **Chat:** Desenvolvimento de um chat integrado ao sistema para troca de mensagens rápidas.

5 Descrição do Produto

O sistema de controle acadêmico baseado na Web será implementado em back-end com a linguagem PHP, o framework Symfony2 e arquitetura MVC, através da IDE NetBeans. E front-end com as linguagens HTML5, CSS3 (Bootstrap) e JavaScript. A persistência dos dados utilizará MySQL.

6 Casos de uso

O sistema possui 3 atores: o Aluno, que pode realizar as operações de logar no sistema, cadastrar disciplinas, acessar número de faltas e acessar suas notas de provas e atividades; o Professor, que pode realizar as operações de logar no sistema, cadastrar disciplinas, marcar frequência e disponibilizar notas de provas e atividades; e o Banco de Dados que provê todas as informações necessárias para as atividades dos casos de uso. Ao todo são seis casos de uso em formato curto. Serão apresentados a seguir:

6.1 Logar no Sistema

O usuário deve informar na área de login, o seu nome de usuário, sua senha e selecionar o botão “entrar”. Assim será redirecionado para a página principal do sistema.

6.2 Cadastrar Disciplinas

O sistema exibe um conjunto de menus com as opções disponíveis. O usuário seleciona a opção de cadastro de disciplinas. Então, o sistema direciona o usuário para uma página de cadastro. O usuário preenche o formulário de cadastro e seleciona o botão “cadastrar”. O sistema apresentará as informações dadas pelo usuário para ele conferir se os dados informados estão corretos. Em caso positivo, o usuário deve confirmar o cadastro, caso contrário, voltar para a página de cadastro. Após confirmação, o sistema informa que o cadastro foi realizado com sucesso.

6.3 Marcar frequência

Assim que o usuário seleciona a turma na área de Turmas Cadastradas do sistema, dentre as opções oferecidas pelo sistema está a de marcar as frequências dos alunos matriculados na turma em questão. O usuário clica nesta opção e o sistema disponibiliza o nome de todos os alunos em ordem alfabética e a opção para marcar se o aluno está ou não presente na data referida.

6.4 Disponibilizar notas de provas e trabalhos

Após o usuário selecionar a turma no menu de Turmas Cadastradas, há a opção de disponibilizar as notas de provas e trabalhos. Ao clicar nesta opção, o sistema disponibiliza o nome de todos os alunos em ordem alfabética matriculados na turma. O usuário informa a nota de cada aluno presente na lista. Então, o sistema processa estas alterações e as salva no banco de dados.

6.5 Acessar número de faltas

Dado que o aluno esteja logado e cadastrado nas disciplinas, quando ele clicar em “ver frequência” e escolher a matéria desejada, então o sistema abrirá uma tabela com a frequência detalhada e o total de faltas.

6.6 Acessar notas de provas e trabalhos

Dado que o aluno esteja logado e cadastrado nas disciplinas, quando ele clicar em “notas”, escolher entre notas de provas e notas de atividades e escolher a matéria desejada, então o sistema abrirá uma tabela contendo as provas/atividades realizadas, as datas relacionadas, o valor e a nota.

7 Diagrama de casos de uso

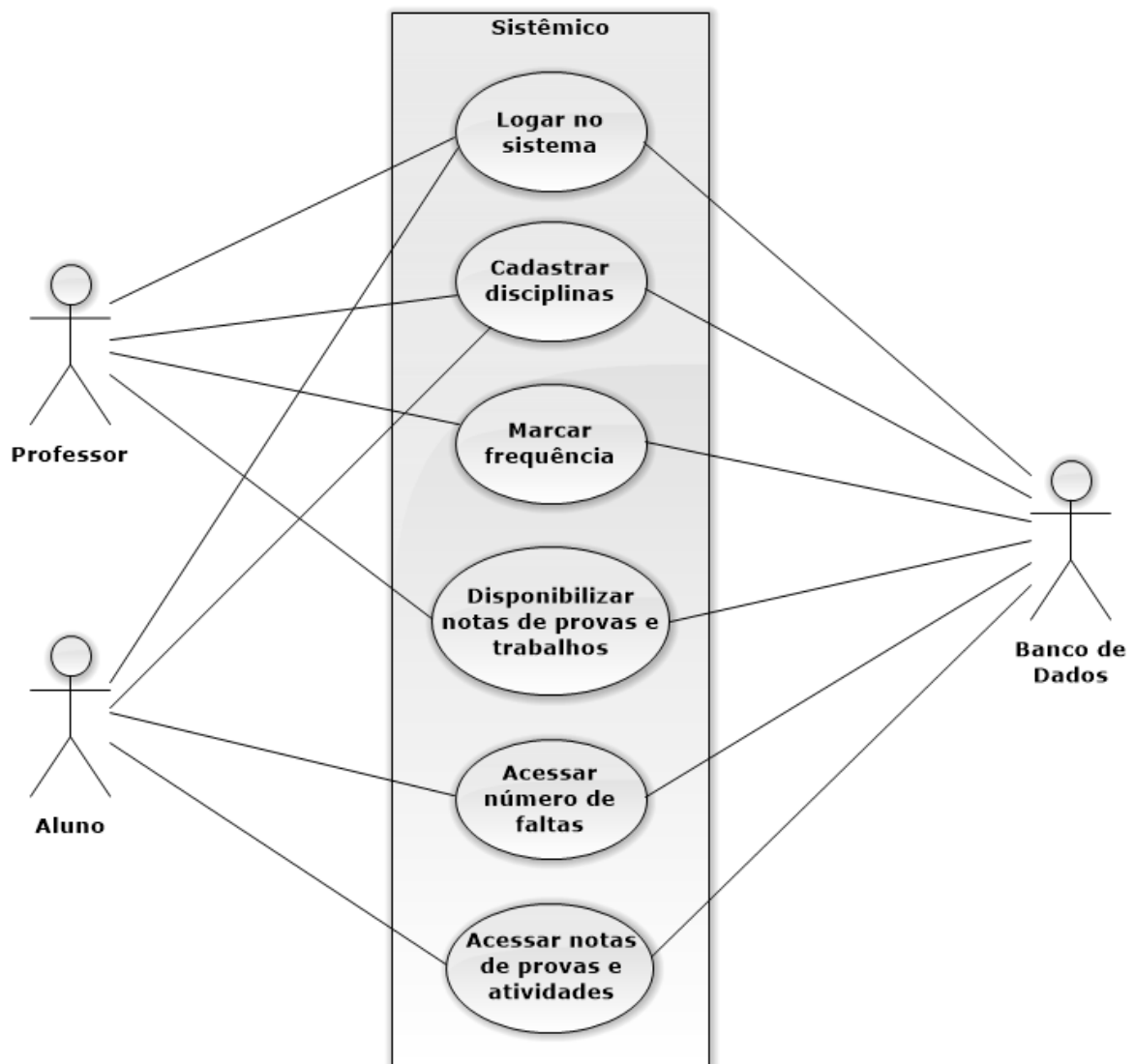


Figura 1 – Diagrama de Casos de Uso

8 Classes

8.1 Descrição

8.1.1 Prova

A classe *Prova* conterá os atributos: descrição, idProva, data, valor e nota. O professor poderá adicionar, alterar e excluir provas.

Essa classe se relaciona com as classes aluno e disciplina, podendo uma prova ser realizada por um aluno e um aluno realizar várias provas e uma disciplina pode conter zero ou mais provas.

8.1.2 Aluno

A classe *Aluno* conterá os atributos: nome, matrícula, período, email e frequência. O aluno pode realizar seu cadastro, fazer login no sistema, consultar notas, consultar frequências, recuperar senha, cadastrar em disciplina e cancelar cadastro em disciplina.

Essa classe se relaciona com disciplina, atividade, curso, turma e prova. Um aluno realiza uma ou várias provas, um ou vários alunos realizam uma ou várias disciplinas, um ou vários alunos fazem uma ou várias atividades, um ou mais alunos fazem um curso e um ou mais alunos são alocados em uma ou várias turmas.

8.1.3 Professor

A classe *Professor* conterá os atributos: nome, idProfessor e email. O professor pode realizar seu cadastro, fazer login no sistema, recuperar senha, fazer chamada, adicionar atividades e provas, cadastrar em disciplina e cancelar cadastro em disciplina.

Essa classe se relaciona com disciplina. Um professor pode lecionar uma ou mais disciplinas.

8.1.4 Disciplina

A classe *Disciplina* conterá os atributos: nome, código, carga horária, prerequisite e departamento. A disciplina pode ser cadastrada, alterada e excluída.

Essa classe se relaciona com professor, aluno, preRequisito, turma, atividade e prova. Uma ou mais disciplinas podem ser lecionadas por um professor, uma ou mais disciplinas podem ser realizadas por uma ou mais alunos, uma ou mais disciplinas possuem um preRequisito, uma ou mais disciplinas estão contidas em uma turma, uma disciplina contém zero ou mais provas, uma disciplina contém zero ou mais atividades.

8.1.5 Atividade

A classe *Atividade* conterá os atributos: nome, id, valor, nota, data de início e data de entrega. A atividade pode ser adicionada, alterada e excluída.

Essa classe se relaciona com disciplina e aluno. Zero ou mais atividades estão contidas em uma disciplina, uma ou mais atividades são realizadas por um ou mais alunos.

8.1.6 Turma

A classe *Turma* conterá os atributos: nome, id. A turma pode ser adicionada, alterada e excluída.

Essa classe se relaciona com aluno e disciplina. Uma ou mais turmas aloca um ou vários alunos, uma ou mais turmas alocam uma ou mais disciplinas.

8.1.7 Curso

A classe *Curso* conterá os atributos: nome, id e departamento. O curso permite realização de cadastro.

Essa classe se relaciona com aluno. Um curso pode ser realizado por um ou mais alunos.

8.1.8 preRequisito

A classe *preRequisito* conterá os atributos: nome e id.

Essa classe se relaciona com disciplina. Um preRequisito é possuído por uma ou várias disciplinas.

8.2 Diagrama de Classes

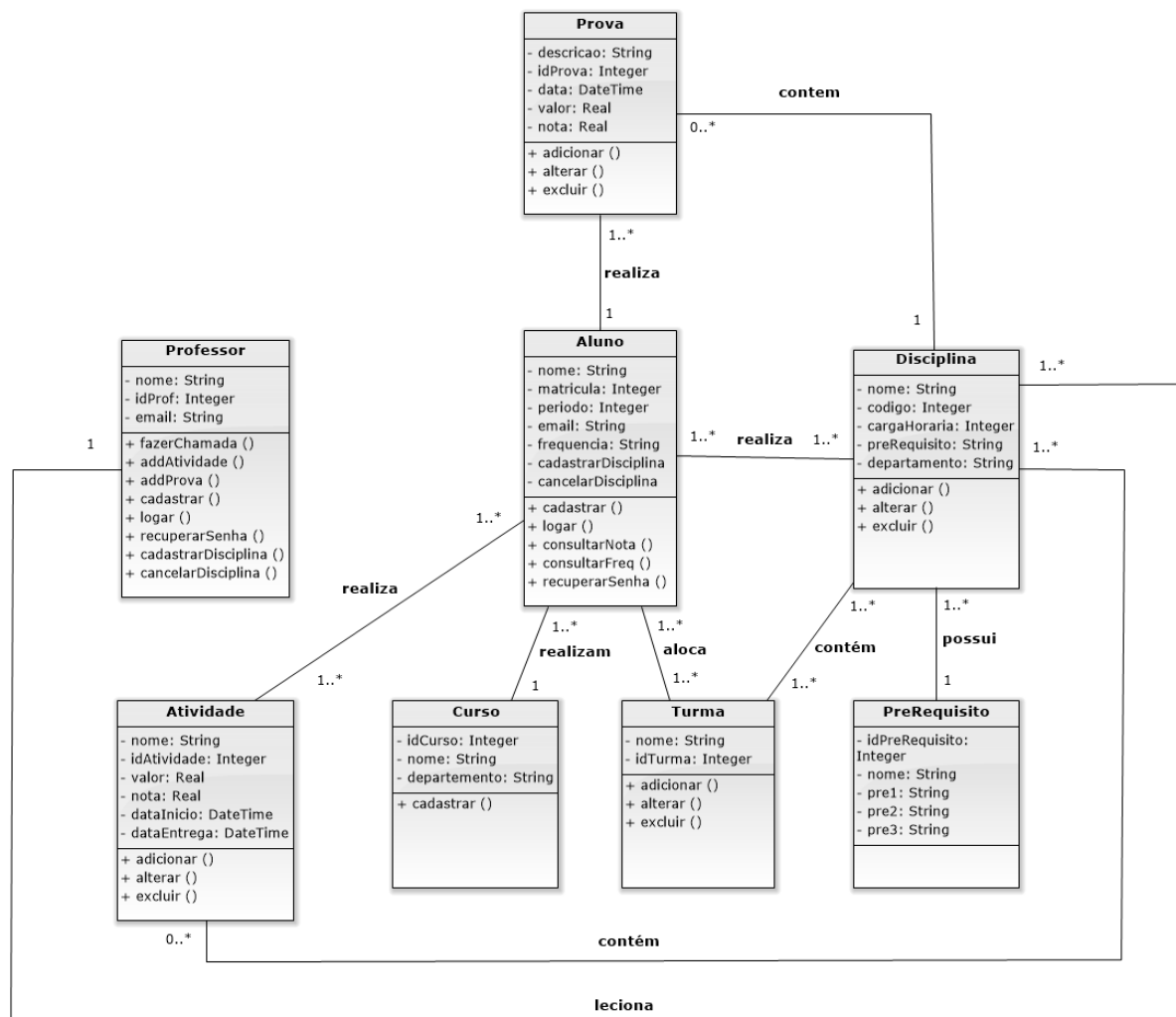


Figura 2 – Diagrama de Classes

9 Banco de Dados (DER)

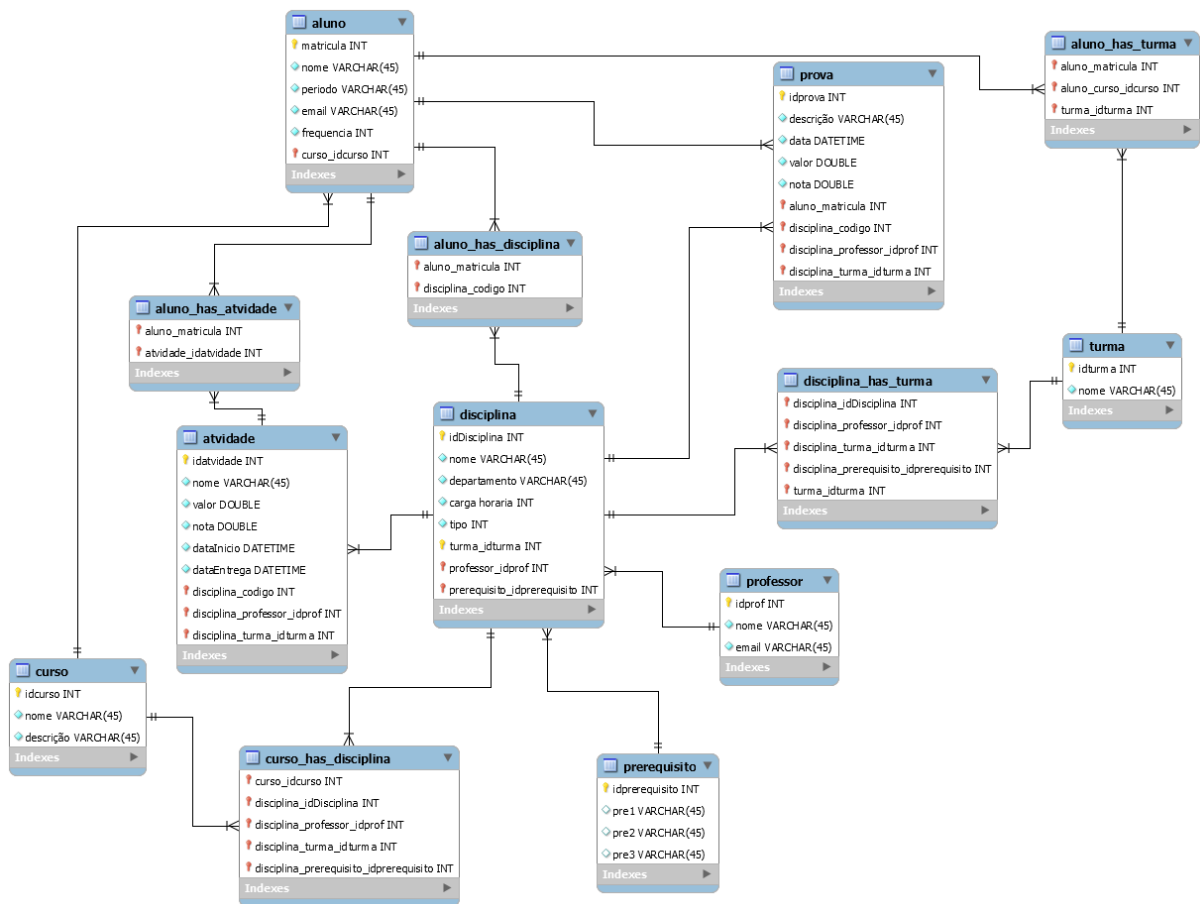


Figura 3 – Diagrama Entidade Relacionamento

10 Protótipos de Tela

O código fonte dos protótipos estão contidos na pasta Protótipos de Tela.

11 Cronograma

O desenvolvimento do sistema será realizado durante todo o semestre letivo 2015/2 e serão realizadas, em cada mês, as seguintes atividades:

- Novembro:
 - criação do documento de análise/protótipo;
 - criação dos diagramas UML.
- Dezembro:

- criação e povoamento do banco de dados.
- Janeiro/Fevereiro/Março:
 - projeto do sistema acadêmico;
 - implementação;
 - validação.

12 Controle de versões

Versão	Data	Autor(es)	Notas de Revisão
0.1	30/10/2015	José	Planejamento inicial do projeto
0.2	03/11/2015	José	Descrição do problema e objetivos
0.3	06/11/2015	José, Guilherme	Introdução, Escopo da aplicação, Descrição do Problema e Cronograma
0.4	08/11/2015	Guilherme	Casos de Uso e Diagrama de Casos de Uso
0.5	12/11/2015	José, Guilherme	Diagrama de Classes e Diagrama Entidade Relacionamento (DER)
0.6	13/11/2015	José, Guilherme	Implementação dos protótipos
0.7	19/11/2015	José, Guilherme	Descrição dos Casos de Uso, Descrição das Classes e Conclusão dos protótipos

Tabela 3 – Controle de Versões

13 Referências

- [1] Site: <http://www.w3schools.com/>
- [2] Site: <http://globocom.github.io/bootstrap/>