

Documento de Análise de Requisitos v2.0 - Sistema de Controle Acadêmico

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

15 de março de 2016

1 Introdução

O sistema de controle acadêmico foi um sistema desenvolvido em PHP utilizando a arquitetura MVC e o *framework* CakePHP. 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 lançar 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, alunos, professores e cursos pelo administrador do sistema.
3. Marcar frequência dos alunos pelo professor.
4. Disponibilizar notas de atividades pelo professor.
5. Acesso ao número de faltas pelo aluno.
6. Acesso às notas das atividades 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.
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 CakePHP e arquitetura MVC, através do editor de texto Sublime Text 2. 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, acessar número de faltas e ver suas notas nas atividades realizadas; o Professor, que pode realizar as operações de logar no sistema, fazer chamada e disponibilizar notas das atividades realizadas pelo aluno; e o Banco de Dados que provê todas as informações necessárias para as atividades dos casos de uso. Ao todo são cinco 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 e-mail, sua senha e selecionar o botão “Entrar”. Assim será redirecionado para a página principal do sistema.

6.2 Fazer Chamada

Assim que o professor seleciona a opção de “Listar Disciplinas” no menu principal do sistema, dentre as opções oferecidas está a de “Fazer Chamada” dos alunos matriculados em uma determinada disciplina. O professor clica nesta opção e o sistema disponibiliza uma lista com o nome de todos os alunos cadastrados na disciplina em questão e a opção para adicionar presença. Ao clicar nesta opção, é exibido um formulário para que o professor marque se o aluno está(1) ou não(0) presente na data referida. Então, o sistema processa estas alterações e as salva no banco de dados.

6.3 Disponibilizar notas de atividades

Assim que o professor seleciona a opção de “Listar Disciplinas” no menu principal do sistema, dentre as opções oferecidas está a de “Adicionar Atividade” para os alunos matriculados em uma determinada disciplina. O professor clica nesta opção e o sistema

disponibiliza uma lista com o nome de todos os alunos cadastrados na disciplina em questão e a opção para adicionar atividade. Ao clicar nesta opção, é exibido um formulário para que o professor informe o nome da atividade, seu valor, nota (caso a atividade já estiver sido feita pelo aluno), a data de entrega e seu tipo. Então, o sistema processa estas alterações e as salva no banco de dados.

6.4 Acessar número de faltas

Dado que o aluno esteja logado e cadastrado nas disciplinas, quando ele clicar em "Ver Frequência", o sistema exibirá uma tabela com o total de faltas em cada disciplina.

6.5 Acessar notas para as atividades

Dado que o aluno esteja logado e cadastrado nas disciplinas, quando ele clicar em "Ver Notas", o sistema exibirá uma tabela contendo as atividades realizadas, o seu valor e as respectivas notas.

7 Diagrama de casos de uso

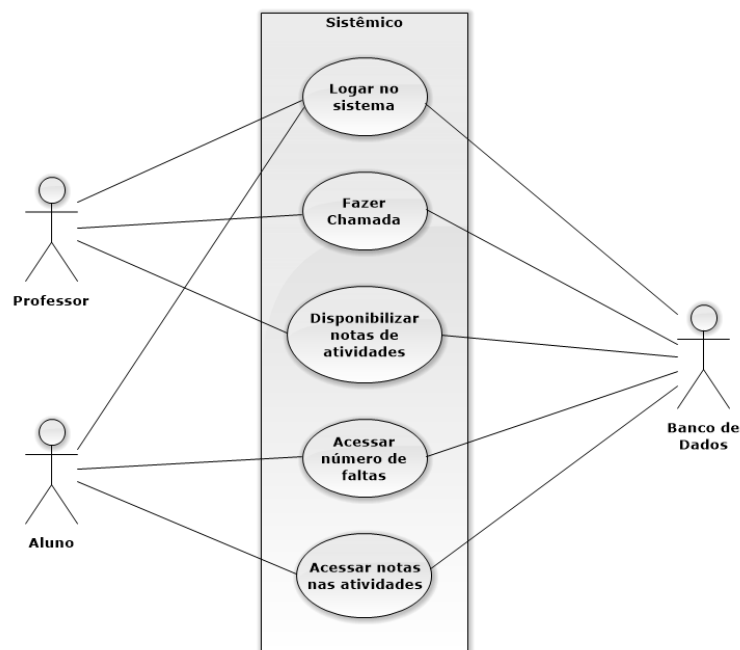


Figura 1 – Diagrama de Casos de Uso

8 Classes

8.1 Descrição

8.1.1 Aluno

A classe *Aluno* conterá os atributos: id, nome, matrícula, período, email e senha. O aluno pode fazer login no sistema, consultar notas e consultar frequências.

Essa classe se relaciona com disciplina, atividade, curso e frequência. 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 aluno possui uma frequência.

8.1.2 Professor

A classe *Professor* conterá os atributos: id, nome, e-mail e senha. O professor pode fazer login no sistema, fazer chamada e adicionar atividades

Essa classe se relaciona com disciplina e frequência. Um professor pode lecionar uma ou mais disciplinas e marca uma ou mais frequências.

8.1.3 Disciplina

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

Essa classe se relaciona com professor, aluno e atividade. Uma ou mais disciplinas podem ser lecionadas por um professor, uma ou mais disciplinas podem ser realizadas por uma ou mais alunos e uma disciplina contém zero ou mais atividades.

8.1.4 Atividade

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

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

8.1.5 Curso

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

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

8.1.6 Tipo

A classe *Tipo* conterá os atributos: id, tipo.

Essa classe se relaciona com atividade. Uma atividade tem um tipo.

8.1.7 Frequência

A classe *Frequência* conterá os atributos: id, data e frequência. A atividade pode ser adicionada.

Essa classe se relaciona com aluno e professor. Um aluno possui uma frequência e uma ou várias frequências são adicionadas por um professor.

8.2 Diagrama de Classes

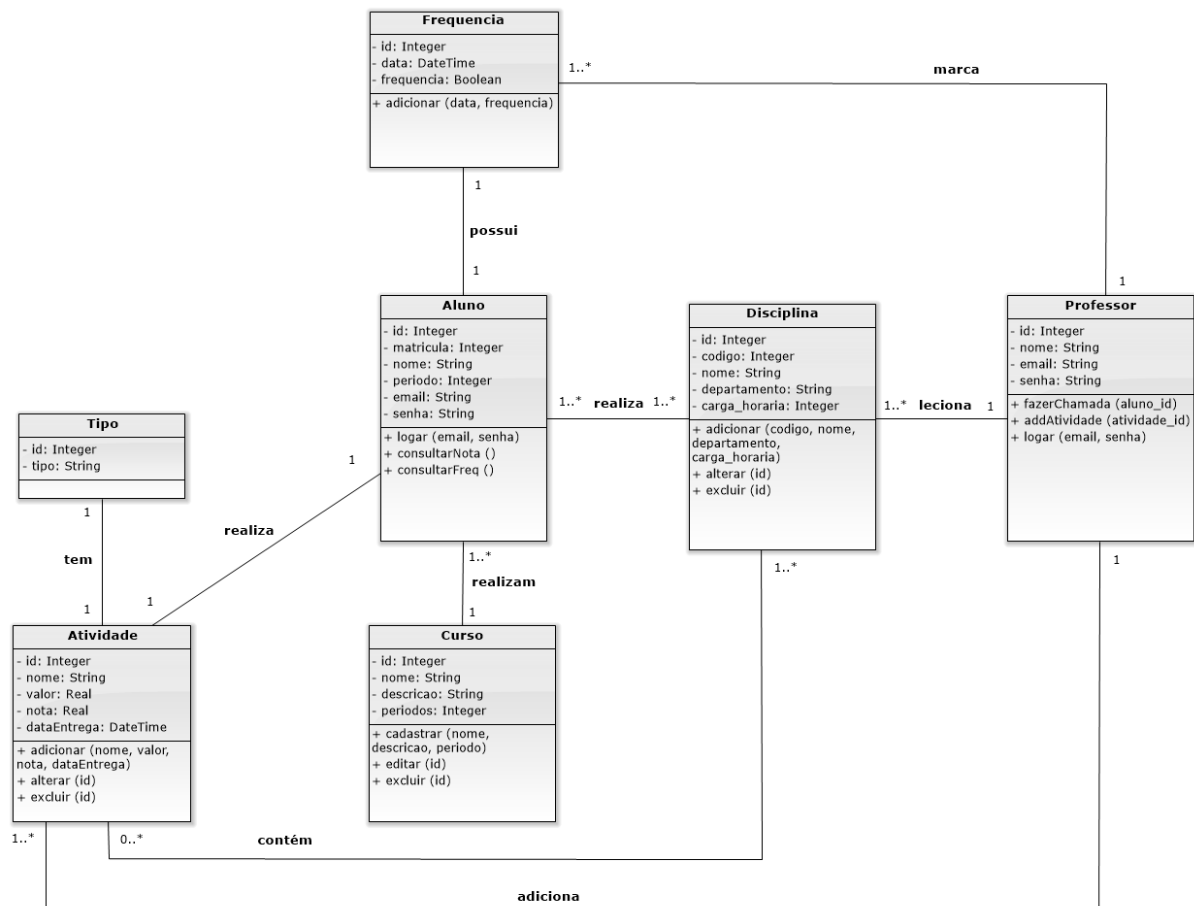


Figura 2 – Diagrama de Classes

9 Banco de Dados (DER)

O Diagrama Entidade-Relacionamento é mostrado na figura a seguir. Pode-se observar que o banco de dados foi reduzido em relação à primeira versão do documento. Ele foi

adequado ao contexto da disciplina, pois estava muito complexo anteriormente.

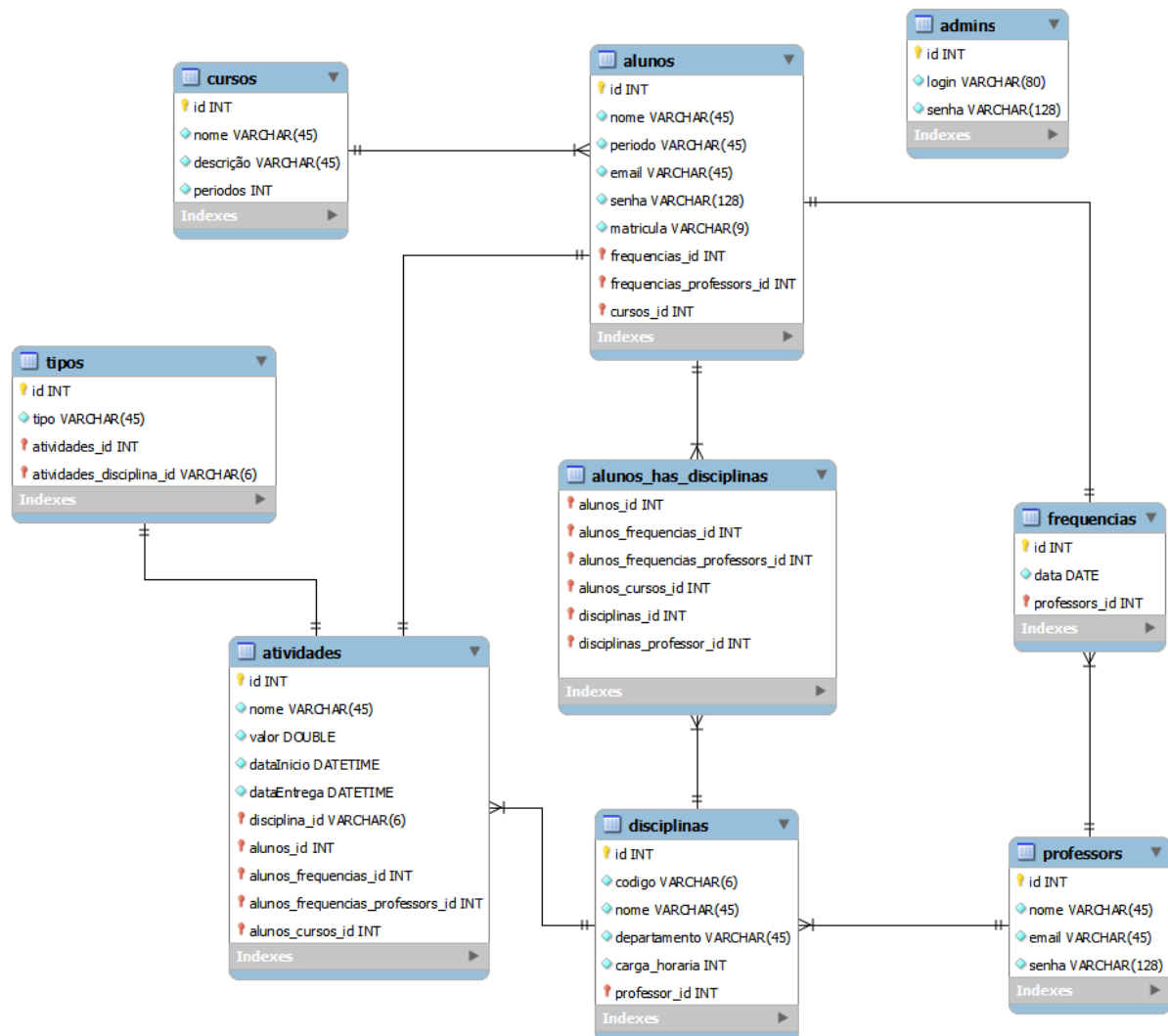


Figura 3 – Diagrama Entidade Relacionamento

10 Telas Finais

As capturas de telas finais do sistema estão contidas dentro da pasta Telas Finais.

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
0.8	09/03/2016	José, Guilherme	Finalização da implementação do sistema
0.9	15/03/2016	José, Guilherme	Atualização do Documento de Análise

Tabela 3 – Controle de Versões

13 Referências

- [1] Site: <http://www.w3schools.com/>
- [2] Site: <http://globocom.github.io/bootstrap/>
- [3] Site: <http://book.cakephp.org/2.0/en/contents.html>