



CENTRO DE INFORMÁTICA - UFPB
MÉTODOS DE PROJETOS DE SOFTWARE

API de Banco de Provas da UFPB

DOCUMENTO DE REQUISITOS

VERSÃO 0.1

João Pessoa, Fevereiro de 2017

EQUIPE DO PROJETO

Diogo Ventura Dantas

11403718

dioogoven@gmail.com

Higor Araújo dos Anjos

11403767

higor.araujo@cc.ci.ufpb.br

Marcos Henrique Alves da Silva

11409511

marcos.alves@cc.ci.ufpb.br

LISTA DE FIGURAS

1	Brainstorming do dia 10 de Fevereiro de 2017	10
2	Diagrama de Casos de Uso	16
3	Diagrama de classes de análise	20

LISTA DE TABELAS

1 Acrônimo e abreviação 7

SUMÁRIO

1	INTRODUÇÃO	7
1.1	Propósito do documento	7
1.2	Acrônimos e abreviações	7
1.3	Visão geral do documento	7
2	DESCRIÇÃO GERAL	8
2.1	Perspectiva do produto	8
2.2	Usuários do software	8
2.3	Suposições e Restrições gerais	8
3	ELICITAÇÃO DOS REQUISITOS	9
3.1	Reuniões de <i>Brainstorming</i>	9
3.2	Análise de Concorrência	9
4	REQUISITOS	10
4.1	Requisitos funcionais (RF)	11
4.1.1	[RF01] Permitir a inserção de prova	11
4.1.2	[RF02] Permitir a edição de prova	11
4.1.3	[RF03] Permitir a deleção de prova	11
4.1.4	[RF04] Permitir a consulta de prova	11
4.1.5	[RF05] Permitir a consulta de provas por departamento	12
4.1.6	[RF06] Permitir a consulta de provas por disciplina	12
4.1.7	[RF07] Permitir a consulta de provas por período	12
4.1.8	[RF08] Permitir a consulta de provas por curso	12
4.1.9	[RF09] Permitir a consulta das últimas provas adicionadas	12
4.1.10	[RF10] Permitir a consulta das últimas provas adicionadas de uma disciplina específica	12
4.2	Requisitos não funcionais (RNF)	13
4.2.1	[RNF01] Armazenamento de Provas	13
4.2.2	[RNF02] Protocolo de Comunicação	13

4.2.3	[RNF03] Ferramentas de programação	13
4.2.4	[RNF04] Formato de submissão de provas	13
4.2.5	[RNF05] Disponibilidade de Acesso	13
4.2.6	[RNF06] Tempo de resposta	13
4.2.7	[RNF07] Número de Usuários	14
4.2.8	[RNF08] Tempo de Manutenção	14
4.2.9	[RNF09] Persistência de dados	14
4.2.10	[RNF10] Versionamento de Código	14
4.2.11	[RNF11] Requisições com autorização	14
4.2.12	[RNF12] Documentação necessária	14
4.2.13	[RNF13] Extração de informações do SIGAA	15
5	CASOS DE USO	16
5.1	Diagrama Geral de Casos de Uso	16
5.2	Descrição de Casos de Uso	16
5.2.1	[UC01] Submeter prova	16
5.2.2	[UC02] Deletar prova	17
5.2.3	[UC03] Editar prova	18
5.2.4	[UC04] Consultar prova	18
6	DIAGRAMA DE CLASSES	20
6.1	DIAGRAMA DE CLASSES DE ANÁLISE	20
	REFERÊNCIAS	20

1 INTRODUÇÃO

1.1 Propósito do documento

Este documento tem por objetivo apresentar a especificação dos requisitos que a *API* de banco de provas da UFPB deve atender em diferentes níveis de detalhamento. Tendo a finalidade de prover serviços de gerenciamento de provas das disciplinas ofertadas na Universidade Federal da Paraíba para o desenvolvimento de aplicações que os utilizem.

1.2 Acrônimos e abreviações

Acrônimo/abreviação	Descrição
RNF	Requisito Não Funcional
RF	Requisito Funcional
UC	Caso de Uso
UFPB	Universidade Federal da Paraíba
API	Application Programming Interface
HTTP	Hypertext Transfer Protocol
URL	Uniform Resource Locator

Tabela 1: Acrônimo e abreviação

1.3 Visão geral do documento

O presente documento de requisitos é organizado como segue:

- **Introdução** - esta seção fornece informações gerais sobre o documento de requisitos, apresentado seu propósito, escopo e também os acrônimos e abreviações necessários para interpretar o documento descrito.
- **Descrição geral** - fornece uma explanação breve sobre o produto de software a ser produzido e suas restrições gerais.
- **Elicitação dos Requisitos** - lista as técnicas utilizadas e resultados obtidos das eliciações de requisitos realizadas.
- **Requisitos** - apresenta os requisitos funcionais e não funcionais mapeados para serem atendidos no desenvolvimento do sistema
- **Descrição de Casos de Uso** - determina os principais casos de uso do sistema.

2 DESCRIÇÃO GERAL

O sistema descrito neste documento tem como objetivo auxiliar desenvolvedores a criar aplicações utilizando dados de provas aplicadas na UFPB. Através da disponibilidade de serviços de gerenciamento de provas, os usuários podem utilizar da base de dados para consultar provas, adicioná-las e criar estatísticas com esses dados dentro de suas aplicações. Com a utilização desta interface, inúmeras aplicações podem ser criadas para facilitar o estudo dos graduandos da UFPB. O acesso a provas anteriores de uma disciplina pode ser considerado um bom exemplo.

2.1 Perspectiva do produto

A UFPB é uma das maiores instituições de ensino do Brasil, com mais de 30 mil alunos ativos e mais de 100 cursos disponíveis[4]. Completar um curso superior em uma universidade pública de qualidade não é uma tarefa simples, o grande número de evasão comprova essa realidade. Fatores como a dificuldade de acesso a informação contribuem de forma incisiva para essa realidade.

De maneira geral, a *API* proposta neste documento tem como principal objetivo disponibilizar diversas provas aplicadas na UFPB de uma forma simples e prática. Através da *API* diversos desenvolvedores poderão consumi-la para criação de aplicativos e ferramentas que auxiliem os alunos durante a sua vida acadêmica.

2.2 Usuários do software

- **Desenvolvedor** - papel de quem utiliza a *API* para a criação de suas aplicações.

2.3 Suposições e Restrições gerais

Para as funcionalidades de todos os usuários é necessário que estes possuam conhecimento dos serviços disponibilizados pelo sistema e como utilizá-los de forma correta. Não há distinção de funcionalidades entre usuários.

O sistema deve ser implantado em um servidor para que possa ser utilizado publicamente na Internet. Devido a isso, a equipe optou por desenvolver o sistema em um ambiente de testes de forma local e posteriormente, em um serviço de nuvem ou em servidores da UFPB. Ainda, por se tratar de um sistema web e por várias aplicações semelhantes funcionarem do mesmo modo, a equipe optou por utilizar a linguagem de programação Javascript e alguns de seus *frameworks*.

3 ELICITAÇÃO DOS REQUISITOS

Durante a fase de elicitação dos requisitos, foram utilizadas primordialmente as técnicas de *brainstorming* e análise de concorrência para a identificação das características desejadas no sistema. Estas técnicas foram utilizadas principalmente a fim de atender o prazo de desenvolvimento do projeto e pelo público alvo a ser alcançado.

Todas as reuniões de *brainstorm* e dados coletados durante essa fase foram registrados em fontes externas à esse documento.

3.1 Reuniões de *Brainstorming*

Foram realizadas reuniões de *brainstorming* a fim de definir o escopo do projeto. Algumas se deram com possíveis usuários do sistema e outros apenas com a equipe do projeto. As reuniões aconteceram tanto virtualmente, através de videoconferência, quanto presencialmente, no Centro de Informática da UFPB. Um exemplo de documento produzido durante uma sessão de *brainstorm* pode ser ilustrado na Figura 1.

3.2 Análise de Concorrência

Não foram encontrados sistemas que exercem o mesmo objetivo do presente projeto. Porém, durante o processo de análise de concorrência algumas aplicações que podem utilizar *APIs* semelhantes foram encontradas, como o *website* Qconcursos[3].

Diferente do escopo deste projeto, mas ainda se tratando de serviços do mesmo segmento, foram avaliados os produtos de empresas como Dropbox[1] e Google Drive [2].

BRAINSTORM DO DIA 10 DE FEVEREIRO

OBJETIVO:

Auxiliar os estudantes no estudo.

COMO:

Criando um banco de provas do centro de informática.

SERVIÇOS QUE UMA APLICAÇÃO PODE DISPONIBILIZAR:

- Envio de provas
- Filtros
- Download de Provas
- Divisao por departamento, centro...
- Pesquisa de provas
- Preview da prova
- Classificar provas
- Report de provas fora do padrão e duplicadas
- (sistema)Disponibilizar um modelo

PROBLEMA DE PROVAS DUPLICADAS:

- Nome de prova seguindo um padrão
- Pesquisar por provas similares no upload

ATRIBUTOS QUE UMA PROVA PODE TER:

- Arquivo PDF (25mb)
- Nome da disciplina
- Centro
- Departamento
- Período
- Número
- Data de inserção

Figura 1: Brainstorming do dia 10 de Fevereiro de 2017

4 REQUISITOS

Para estabelecer a prioridade dos requisitos, foram adotadas as denominações “*essencial*”, “*importante*” e “*desejável*”.

- **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 podem ser deixados para versões posteriores do sistema, caso não haja tempo hábil para implementá-los na versão que está sendo especificada.

4.1 Requisitos funcionais (RF)

4.1.1 [RF01] Permitir a inserção de prova

Descrição: O Sistema deve permitir a inserção de uma prova informando o arquivo (em formato .pdf) e informações de departamento, curso, disciplina e período em suas requisições. Caso o usuário informe campos erroneamente ou inexistentes, o sistema deve informar o erro na resposta da requisição.

Prioridade: Essencial.

4.1.2 [RF02] Permitir a edição de prova

Descrição: O sistema deve permitir a edição de uma prova já cadastrada, informando o identificador da prova e os campos alterados na requisição. O sistema deve informar o sucesso ou insucesso da operação.

Prioridade: Essencial.

4.1.3 [RF03] Permitir a deleção de prova

Descrição: O sistema deve permitir a deleção de uma prova já cadastrada através de um identificador da prova. O sistema deve informar o sucesso ou insucesso da operação.

Prioridade: Essencial.

4.1.4 [RF04] Permitir a consulta de prova

Descrição: O sistema deve permitir a consulta de uma prova presente no sistema retornando todas as informações sobre ela. O sistema deve informar o sucesso ou insucesso da operação.

Prioridade: Essencial.

4.1.5 [RF05] Permitir a consulta de provas por departamento

Descrição: O usuário do sistema pode requisitar uma lista de provas de um departamento específico da Universidade Federal da Paraíba.

Prioridade: Importante.

4.1.6 [RF06] Permitir a consulta de provas por disciplina

Descrição: O sistema deve responder uma lista de provas de uma disciplina específica de acordo com o desejo do usuário.

Prioridade: Importante.

4.1.7 [RF07] Permitir a consulta de provas por período

Descrição: É permitido ao usuário obter uma lista de provas de um período desejado quando especificado.

Prioridade: Importante.

4.1.8 [RF08] Permitir a consulta de provas por curso

Descrição: É permitido ao usuário obter uma lista de provas de um determinado curso quando especificado.

Prioridade: Importante.

4.1.9 [RF09] Permitir a consulta das últimas provas adicionadas

Descrição: O sistema deve fornecer uma lista das 10 últimas provas adicionadas no serviço quando requisitado.

Prioridade: Desejável.

4.1.10 [RF10] Permitir a consulta das últimas provas adicionadas de uma disciplina específica

Descrição: O sistema deve fornecer uma lista das 10 últimas provas de uma disciplina específica quando requisitado.

Prioridade: Desejável.

4.2 Requisitos não funcionais (RNF)

4.2.1 [RNF01] Armazenamento de Provas

Descrição: O sistema deve armazenar e manter as provas submetidas pelos usuários.

Prioridade: Essencial.

4.2.2 [RNF02] Protocolo de Comunicação

Descrição: O sistema deve se comunicar com os usuários através do protocolo de comunicação web HTTP. Utilizando requisições do tipo GET, POST, PUT e DELETE.

Prioridade: Essencial.

4.2.3 [RNF03] Ferramentas de programação

Descrição: O sistema deve ser desenvolvido utilizando um *framework* de código aberto que permita o desenvolvimento ágil.

Prioridade: Essencial.

4.2.4 [RNF04] Formato de submissão de provas

Descrição: As provas submetidas pelos usuários devem ser no formato pdf sem rasuras, respostas ou qualquer outro tipo de escrita manual. Devem seguir o fluxo lógico com perguntas sequenciais e legíveis.

Prioridade: Importante.

4.2.5 [RNF05] Disponibilidade de Acesso

Descrição: O sistema deverá estar disponível para acesso pelo menos em 80% do tempo.

Prioridade: Essencial.

4.2.6 [RNF06] Tempo de resposta

Descrição: Nenhuma resposta de consulta a informações deve demorar mais que 3 segundos para as consultas de listas e 1 segundo para a consulta de uma prova específica.

Prioridade: Essencial.

4.2.7 [RNF07] Número de Usuários

Descrição: O sistema deve suportar até 100 usuários simultâneos.

Prioridade: Essencial.

4.2.8 [RNF08] Tempo de Manutenção

Descrição: Um programador de manutenção com pelo menos 6 meses de experiência no suporte ao produto deverá ser capaz de dar suporte em não mais do que 1 hora de trabalho.

Prioridade: Importante.

4.2.9 [RNF09] Persistência de dados

Descrição: O sistema deverá persistir os dados em um banco de dados de modelo não-relacional, de código aberto e com suporte e documentação amplamente consolidado na comunidade.

Prioridade: Essencial.

4.2.10 [RNF10] Versionamento de Código

Descrição: O sistema deve ser versionado utilizando um sistema de controle descentralizado em repositório de acesso público.

Prioridade: Essencial.

4.2.11 [RNF11] Requisições com autorização

Descrição: As operações de deleção, alteração e inserção devem garantir autorização prévia do sistema através do protocolo padrão para autorização *OAuth 2.0*.

Prioridade: Desejável.

4.2.12 [RNF12] Documentação necessária

Descrição: O sistema deve possuir documentação completa de utilização do sistema. Na documentação, é necessário especificar o formato das requisições, os *endpoints* de cada serviço e exemplos de utilização.

Prioridade: Importante

4.2.13 [RNF13] Extração de informações do SIGAA

Descrição: O sistema deverá extrair as informações necessárias do SIGAA(Sistema Integrado de Gestão de Atividade Acadêmicas) através da API *SIGAA UFPB* [5].

Prioridade: Essencial

5 CASOS DE USO

5.1 Diagrama Geral de Casos de Uso

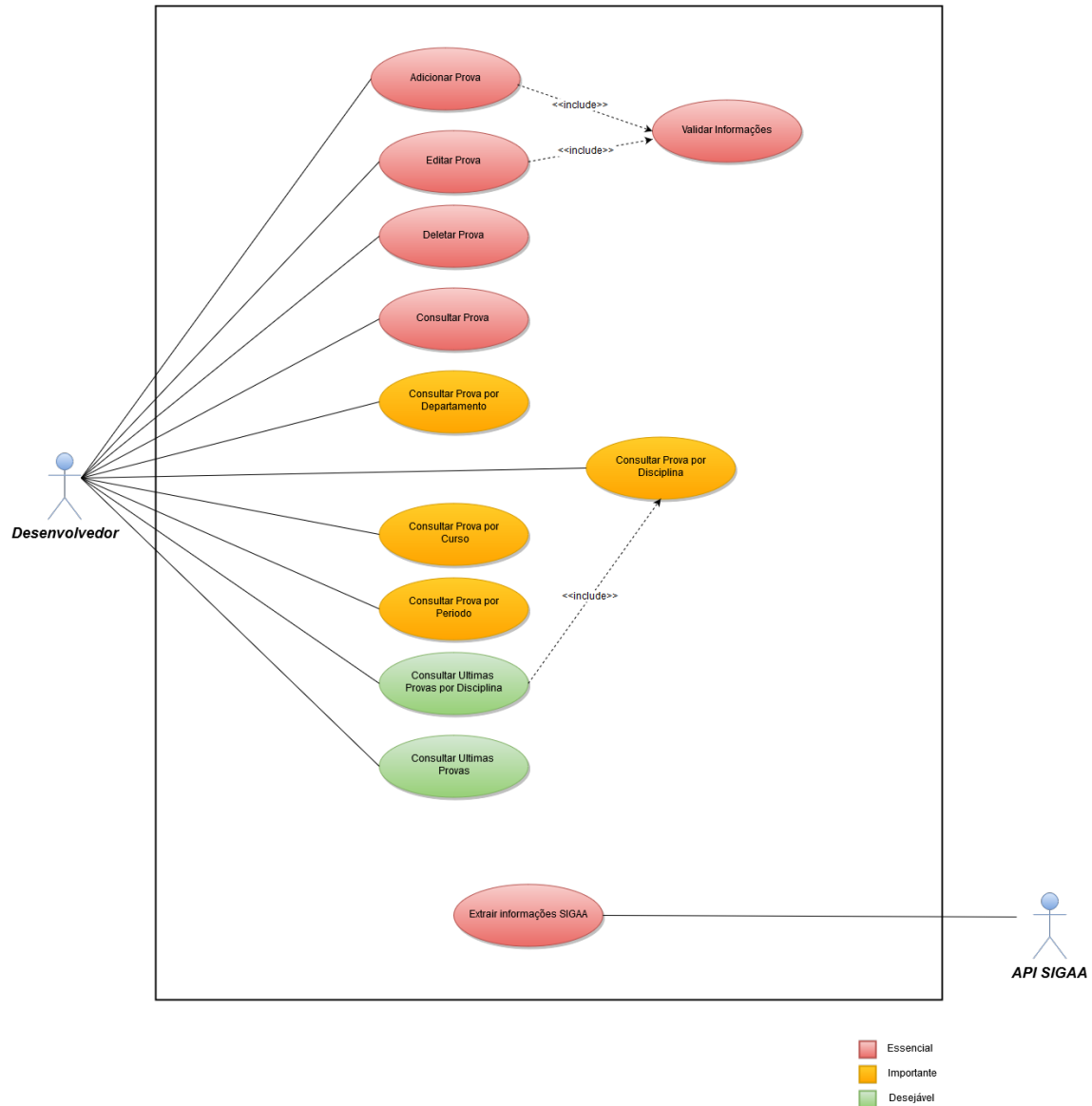


Figura 2: Diagrama de Casos de Uso

5.2 Descrição de Casos de Uso

5.2.1 [UC01] Submeter prova

Identificador:[UC01]

Nome: Submeter Prova

Descrição: Esse caso de uso se dá quando o usuário envia um requisição para a submissão

de uma prova através do protocolo HTTP.

Prioridade: Essencial

Atores: Desenvolvedor

Pré-condições: Não há

Pós-condições: A prova é inserida no banco de dados.

Fluxo Principal:

1. O usuário envia a requisição do tipo POST informando as informações da prova
2. O sistema verifica as condições para a inserção e informa o sucesso ou insucesso

Fluxo Secundário - Informações fora do padrão

1. Caso as informações da prova não correspondam aos padrões especificados uma mensagem de erro deve ser retornada

Fluxo Secundário - Expressão mal-formada

1. Caso a requisição não esteja no padrão utilizado pelo sistema, uma mensagem de erro deve ser retornada

5.2.2 [UC02] Deletar prova

Identificador:[UC02]

Nome: Deletar prova

Descrição: O ator envia uma requisição de deleção para o servidor (DELETE).

Prioridade: Essencial

Atores: Desenvolvedor

Pré-condições: Não há

Pós-condições: A prova é apagada do banco de dados.

Fluxo Principal:

1. O usuário envia a requisição do tipo DELETE informando o identificador da prova
2. O sistema verifica a existência da prova no banco de dados e retorna o sucesso ou insucesso da operação

Fluxo Secundário - Prova Inexistente

1. Caso não exista prova com esse identificador, o sistema responde a requisição com uma mensagem de erro

Fluxo Secundário - Expressão mal-formada

1. Caso a requisição não esteja no padrão utilizado pelo sistema, uma mensagem de erro deve ser retornada

5.2.3 [UC03] Editar prova

Identificador: [UC03]

Nome: Editar prova

Descrição: Esse caso de uso se dá quando o usuário envia uma requisição de alteração de uma prova específica através do protocolo HTTP informando o identificador único da prova

Prioridade: Essencial

Atores: Desenvolvedor

Pré-condições: Não há

Pós-condições: As informações da prova são alteradas no banco de dados

Fluxo Principal:

1. O usuário envia a requisição do tipo PUT informando o identificador da prova pela *URL* e as informações atualizadas no corpo da requisição
2. O sistema consulta o identificador da prova e altera as informações da prova em questão
3. O sistema informa o sucesso ou insucesso da operação

Fluxo Secundário - Prova Inexistente

1. Caso não exista prova com esse identificador, o sistema responde a requisição com uma mensagem de erro

Fluxo Secundário - Expressão mal-formada

1. Caso a requisição não esteja no padrão utilizado pelo sistema, uma mensagem de erro deve ser retornada

5.2.4 [UC04] Consultar prova

Identificador:[UC04]

Nome: Consultar prova

Descrição: Esse caso de uso se dá quando o usuário envia uma requisição para a consulta de uma prova específica através do protocolo HTTP.

Prioridade: Essencial

Atores: Desenvolvedor

Pré-condições: Não há

Pós-condições: As informações específicas dessa provas são retornadas

Fluxo Principal:

1. O usuário envia a requisição do tipo GET informando o identificador da prova
2. O sistema consulta no banco de dados e retorna as informações da prova pesquisada

Fluxo Secundário - Prova Inexistente

1. Caso não exista prova com esse identificador, o sistema responde a requisição com uma mensagem de erro

Fluxo Secundário - Expressão mal-formada

1. Caso a requisição não esteja no padrão utilizado pelo sistema, uma mensagem de erro deve ser retornada

6 DIAGRAMA DE CLASSES

6.1 DIAGRAMA DE CLASSES DE ANÁLISE

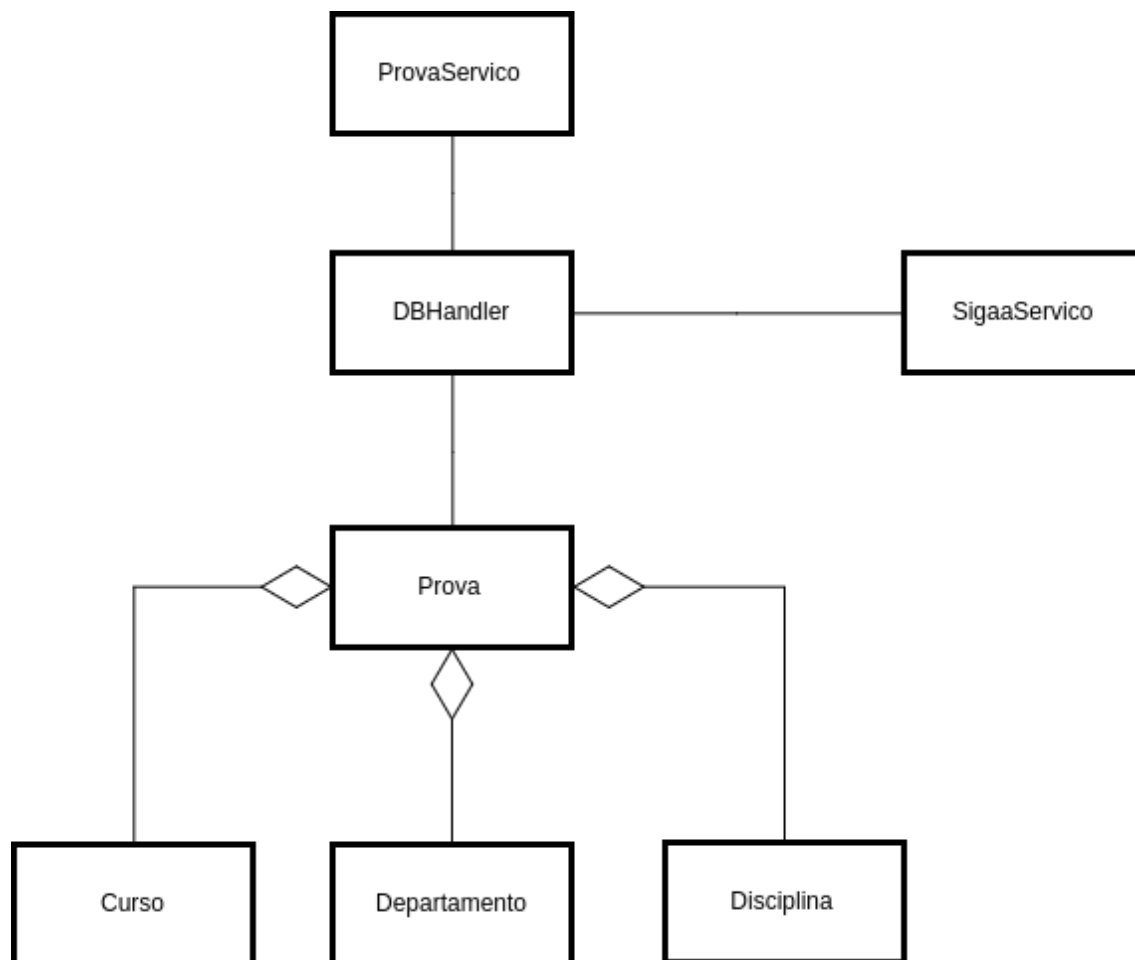


Figura 3: Diagrama de classes de análise

REFERÊNCIAS

- [1] Dropbox. Disponível em: <<https://www.dropbox.com/developers/documentation>>, Acesso em: 20 de Fev. 2017.
- [2] Google drive. Disponível em: <<https://developers.google.com/drive/v3/web/about-sdk>>, Acesso em: 20 de Fev. 2017.
- [3] Qconcursos.com. Disponível em: <<https://www.qconcursos.com>>, Acesso em: 20 de Fev. 2017.
- [4] UFPB. Disponível em: <<http://www.ufpb.br/content/hist%C3%B3rico>>, Acesso em: 27 de Fev. 2017.
- [5] Ygor, CRISPIM David, GUILHERME Fernando, BRITO, Gustavo, SOBRAL. Cursos UFPB API. Disponível em: <https://github.com/fernandobrito/cursos_ufpb>, Acesso em: 27 de Fev. 2017.