

Alexandre Babilone Fonseca

**PDEMT - Programa de Designações da Escola
do Ministério Teocrático**

Vitória, ES

2015

Alexandre Babilone Fonseca

PDEMT - Programa de Designações da Escola do Ministério Teocrático

Monografia apresentada ao Curso de Ciência da Computação do Departamento de Informática da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Bacharel em Ciência da Computação.

Universidade Federal do Espírito Santo – UFES

Centro Tecnológico

Departamento de Informática

Orientador: Prof. Dr. Vítor E. Silva Souza

Vitória, ES

2015

Alexandre Babilone Fonseca

PDEMT - Programa de Designações da Escola do Ministério Teocrático/
Alexandre Babilone Fonseca. – Vitória, ES, 2015-
92 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Vítor E. Silva Souza

Monografia (PG) – Universidade Federal do Espírito Santo – UFES
Centro Tecnológico
Departamento de Informática, 2015.

1. Palavra-chave1. 2. Palavra-chave2. I. Souza, Vítor Estêvão Silva. II.
Universidade Federal do Espírito Santo. IV. PDEMT - Programa de Designações
da Escola do Ministério Teocrático

CDU 02:141:005.7

Alexandre Babilone Fonseca

PDEMT - Programa de Designações da Escola do Ministério Teocrático

Monografia apresentada ao Curso de Ciência da Computação do Departamento de Informática da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Bacharel em Ciência da Computação.

Trabalho aprovado. Vitória, ES, 16 de Maio de 2015:

Prof. Dr. Vítor E. Silva Souza
Orientador

Professor
Convidado 1

Professor
Convidado 2

Vitória, ES
2015

Agradecimentos

À Jeová Deus, o Criador de todas as coisas, por me dar forças para alcançar este objetivo.

Aos meus pais, Gutemberg e Cilene, por simplesmente me darem tudo o que precisei, o que inclui seu tempo, dedicação, amor, carinho e trabalho.

Ao meu orientador Vitor, pelo tempo dedicado, pela disposição em ajudar, pelas dúvidas esclarecidas e pelas idéias transmitidas para que esse trabalho pudesse ser concluído.

À minha namorada, Laísa, por sempre estar ao meu lado, aos amigos da UFES, pelo companherismo demonstrado ao longo desses anos. À todos os professores que contribuíram para minha formação, especialmente à professora Rosane, pelo carinho e por sempre estar pronta pra ajudar seja qual for a situação.

Resumo

Semanalmente, as Testemunhas de Jeová ministram a Escola do Ministério Teocrático. O dirigente desta possui diversas atividades, todas realizadas manualmente e que demandam muito tempo. Para otimizar tais atividades e deixá-las mais agradáveis, foi proposta a criação do PDEMT (Programa de Designações da Escola do Ministério Teocrático).

Dentre suas principais funcionalidades, estão a designação e caracterização de discursos para os participantes habilitados, geração de relatórios para apoiar as decisões do dirigente e a integração do sistema com o Google Calendar, para facilitar a divulgação das designações aos participantes. Para a construção do sistema, foram colocadas em prática as disciplinas aprendidas no decorrer do curso, tais como Engenharia de Software, Engenharia de Requisitos, Banco de Dados, Projeto de Sistema de Software e Programação Orientada a Objetos. O processo de construção do PDEMT passou pelas etapas de levantamento de requisitos, especificação de requisitos, definição da arquitetura do sistema, implementação e testes.

Palavras-chaves: Google Calendar, Agendamento, Testemunhas de Jeová, Escola do Ministério Teocrático, Desenvolvimento Java Desktop.

Lista de ilustrações

Figura 1 – O Modelo em Cascata	22
Figura 2 – Atividades de modelagem e artefatos produzidos	23
Figura 3 – Representação da Service Layer, ou Camada de Serviço em português	24
Figura 4 – Diagrama de Casos de Uso	29
Figura 5 – Diagrama de Classes	30
Figura 6 – Arquitetura de Software do PDEMT	34
Figura 7 – Componente de Domínio do Problema - CDP	39
Figura 8 – Componente de Gerência de Tarefas dos casos de uso mais básicos e designação de discursos	40
Figura 9 – Componente de Gerência de Tarefas dos casos de uso referentes a relatórios e consultas	41
Figura 10 – Arquitetura da CIU	42
Figura 11 – Arquitetura da CGD	43
Figura 12 – A Camada de Gerência de Dados	44
Figura 13 – Modelo Entidade-Relacionamento	44
Figura 14 – Tela principal do PDEMT	45
Figura 15 – Tela de Cadastro de Aluno	45
Figura 16 – Tela de Gerenciamento de Alunos	45
Figura 17 – Primeiro passo para Designar Discursos - selecionar a opção Designar no menu Discursos	46
Figura 18 – Segundo passo para Designar Discursos - escolher um período	46
Figura 19 – Terceiro passo para Designar Discursos - escolher oradores, cenas, oratórias e ajudantes para os discursos	46
Figura 20 – Opção que permite habilitar ou não o uso da marcação de eventos na agenda do Google	46
Figura 21 – Agenda pessoal de um aluno após a inserção de uma designação	47
Figura 22 – Detalhamento da designação na agenda do Google	47
Figura 23 – Trecho de um relatório feito para o dirigente da EMT. Esse relatório contém todas as características de todas as designações feitas durante o período escolhido para a geração do relatório. Para cada discurso, é mostrado quem será o orador, seu tema, fonte de matéria, ajudante, característica de oratória observada e a data.	48
Figura 24 – Trecho de um relatório feito para ser fixado no Quadro de Anúncios da Congregação. Esse relatório mostra a programação feita para as semanas do período selecionado pelo usuário, contendo o tipo de discurso e o nome de quem o proferirá.	48

Figura 25 – Exemplo de 2 relatórios individuais, a serem entregues aos participantes que farão os discursos, contendo todas as informações que o aluno necessita para realizar o discurso. 49

Lista de tabelas

Tabela 1 – Atores do PDEMT	28
Tabela 2 – Mapeamento entre Classes do CGT e Casos de Uso	39
Tabela 3 – Objetivos do projeto e sua situação até o término da monografia	51

Lista de abreviaturas e siglas

UML	Unified Modeling Language
EMT	Escola do Ministério Teocrático
PDEMT	Programa de Designações da Escola do Ministério Teocrático
API	Application Programming Interface

Sumário

1	INTRODUÇÃO	17
1.1	Objetivos	17
1.2	Metodologia	18
1.3	Organização do Texto	18
2	REFERENCIAL TEÓRICO	21
2.1	Ciclo de Vida	21
2.2	Engenharia de Requisitos	22
2.3	Padrões Utilizados na Implementação	24
2.3.1	Camada de Serviço	24
2.3.2	DAO	25
2.4	APIs	25
3	ANÁLISE E ESPECIFICAÇÃO DE REQUISITOS	27
3.1	Descrição do Escopo	27
3.2	Modelo de Casos de Uso	28
3.3	Diagrama de Classes	30
4	PROJETO E IMPLEMENTAÇÃO	33
4.1	Arquitetura de Software	33
4.2	API's Utilizadas	34
4.2.1	Google Calendar	34
4.2.2	JDBC	36
4.2.3	iText-pdf	38
4.3	Detalhamento dos Componentes da Arquitetura	38
4.3.1	Camada de Lógica de Negócio (CLN)	38
4.3.2	Camada de Interface com o Usuário (CIU)	39
4.3.3	Camada de Gerência de Dados (CGD)	40
4.4	O PDEMT	40
5	CONSIDERAÇÕES FINAIS	51
5.1	Conclusões	51
5.2	Trabalhos Futuros	52
	REFERÊNCIAS	55

1 Introdução

Todas as semanas, as Testemunhas de Jeová¹ ministram a Escola do Ministério Teocrático (EMT) em seus Salões do Reino. Cabe ao Dirigente da EMT realizar as designações de discursos para cada participante, levando em conta uma série de características, tanto do discurso quanto do participante (JEOVá, 2008).

Caracterizar manualmente todas as designações leva tempo, pois é preciso verificar para cada participante se as cenas e as características de oratória escolhidas para suas designações estarão variando ou se repetindo. Além disso, exige consultar designações anteriores para contar a frequência de determinado participante, por exemplo, a fim de verificar se ele não está recebendo muitas designações enquanto outros participantes estão recebendo poucas.

Pensando em automatizar essas atividades, foi projetado e criado o Programa de Designações da Escola do Ministério Teocrático (PDEMT). Um sistema *desktop*, feito em Java™, que proporciona ao dirigente da EMT a facilidade de realizar as designações de forma automatizada, provendo a ele as informações necessárias para apoiar suas decisões. Além disso, para aumentar a facilidade em comunicar os participantes de suas designações, o PDEMT é integrado ao Google Calendar². Assim, cada participante terá suas designações marcadas em sua agenda pessoal do Google.

1.1 Objetivos

O objetivo desse trabalho é desenvolver um sistema computacional (PDEMT) capaz de apoiar a realização das atividades feitas manualmente pelo dirigente da Escola do Ministério Teocrático (JEOVá, 2008). Esse objetivo geral pode ser detalhado nos seguintes objetivos específicos :

1. Identificar e documentar os requisitos da ferramenta;
2. Realizar a modelagem comportamental e estrutural da ferramenta e documentá-la;
3. Definir a arquitetura da ferramenta e documentá-la;
4. Implementar a ferramenta.

No desenvolvimento do PDEMT, foram exercitados conhecimentos adquiridos em sala de aula ao longo do curso de Ciência da Computação, em especial das disciplinas de

¹ <<http://www.jw.org>>

² <<http://calendar.google.com>>

Engenharia de Software, Engenharia de Requisitos, Banco de Dados, Projeto de Sistema de Software e Programação Orientada a Objetos.

1.2 Metodologia

A metodologia utilizada para desenvolver este trabalho foi composta pelas seguintes atividades :

1. *Revisão Bibliográfica:* consultar boas práticas de Engenharia de Software e de Requisitos, uso de Banco de Dados Relacional, Programação Orientada a Objetos, Padrões de Projeto de Sistemas aplicados à linguagem de programação Java.
2. *Estudo de Tecnologias:* Nessa etapa, foi necessário o estudo de tecnologias utilizadas para o desenvolvimento do sistema, tais como: Linguagem de Programação Java; Ambiente de Desenvolvimento NetBeans IDE 8.0.2; Banco de Dados Postgree SQL; JDBC (biblioteca para realizar a persistência de dados); Java Swing (biblioteca para implementação da interface com o usuário); e Google Calendar API.
3. *Elaboração da Documentação do Sistema:* Nessa etapa, foi definida a documentação do sistema. A princípio foi elaborado o Documento de Levantamento de Requisitos, apresentando uma descrição geral do minimundo do sistema e definição dos requisitos funcionais e não funcionais e regras de negócio. Em seguida, foi elaborada o Documento de Especificação de Requisitos, apresentando a identificação dos subsistemas, casos de uso, modelo estrutural, modelo dinâmico e glossário do projeto. Por fim, foi elaborado o Documento de Projeto, contendo a arquitetura do software e projeto detalhado de cada um de seus componentes.
4. *Implementação e Testes:* Nessa etapa, o sistema foi implementado e testado, tendo sido realizados testes ao longo do desenvolvimento, bem como após a conclusão do sistema.
5. *Redação da Monografia:* Nessa etapa, foi realizada a escrita desta monografia. Vale ressaltar que a escrita da monografia ocorreu em paralelo com a finalização da implementação do sistema.

1.3 Organização do Texto

Neste capítulo, foi apresentada uma introdução, contendo uma breve descrição do domínio do problema, o contexto em que ele está inserido e a motivação que levou a esse trabalho. Além deste capítulo, a monografia possui mais 4 capítulos, são eles:

Capítulo 2 – Referencial Teórico: apresenta uma revisão bibliográfica sobre os principais temas relacionados ao projeto. Como fonte de pesquisa foram utilizados documentações de API's, livros e trabalhos acadêmicos que tratam sobre os assuntos relevantes a esse trabalho.

Capítulo 3 – Análise e Especificação de Requisitos: apresenta a descrição do mundo contemplado pelo sistema, seus requisitos, casos de uso e modelos de classe.

Capítulo 4 – Projeto e Implementação: apresenta a arquitetura definida para o sistema, detalha alguns dos componentes dessa arquitetura e apresenta algumas telas do sistema.

Capítulo 5 – Considerações Finais: Apresenta as considerações finais do trabalho, incluindo algumas dificuldades encontradas, contribuições e experiências adquiridas no desenvolvimento dessa ferramenta. Além disso, são identificados alguns possíveis trabalhos futuros que poderão surgir a partir deste.

2 Referencial Teórico

Este capítulo apresenta os principais aspectos teóricos que fundamentaram este trabalho e está organizado em 4 seções. A seção 2.1 aborda a Engenharia de Software, destacando os principais conceitos e processos utilizados. A seção 2.2 apresenta métodos utilizados de Engenharia de Requisitos. As seções 2.3.1 e 2.3.2 apresentam os padrões de Projeto de Software adotados para o PDEMT. Por fim, a seção 2.4 aborda conceitos sobre APIs utilizadas na codificação.

2.1 Ciclo de Vida

Dentre os principais objetivos da Engenharia de Software estão: oferecer software de boa qualidade, impedir atrasos na entrega e aumento dos custos de produção. Para alcançar esses objetivos, não basta apenas focar no produto gerado, mas também no seu processo de desenvolvimento, no ciclo de vida do software. Ao escolher um modelo dentre os vários existentes, é preciso levar em conta alguns fatores, por exemplo:

- Qual é o tipo de software a ser desenvolvido?
- Qual é o paradigma de desenvolvimento?
- Quão complexo é o sistema?
- O objetivo do software está bem definido? Os requisitos estão estáveis?
- A equipe de desenvolvimento está madura para realizar esse projeto?

Embora diferentes projetos requeiram processos com características específicas para atender às suas particularidades, é possível estabelecer um conjunto de atividades comuns à maioria deles: planejamento, especificação de requisitos, análise, projeto, implementação, testes e implantação (FALBO, 2014). Levando em conta essas atividades e os fatores listados acima, foi escolhido o modelo Cascata, que organiza as atividades do processo de desenvolvimento de forma sequencial, conforme a Figura 1.

No modelo escolhido, cada fase envolve a elaboração de um ou mais documentos, que devem ser aprovados antes de se iniciar a fase seguinte. Assim, uma fase só deve ser iniciada após a conclusão daquela que a precede. Uma vez que, na prática, essas fases se sobrepõem de alguma forma, geralmente, permite-se um retorno à fase anterior para a correção de erros encontrados. A entrega do sistema completo ocorre em um único marco, ao final da fase de Implantação.

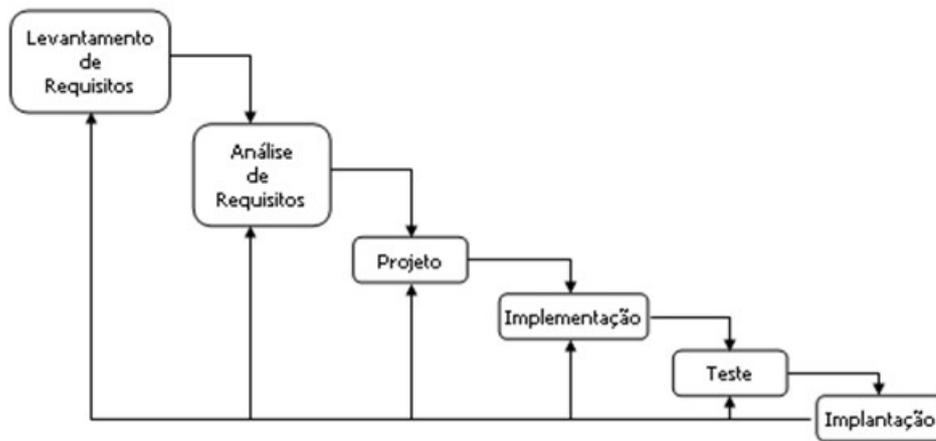


Figura 1 – O Modelo em Cascata

2.2 Engenharia de Requisitos

Um requisito de sistema de software pode ser definido como a especificação de um serviço que o sistema deve oferecer; como uma restrição sob a qual ele deve operar; como uma propriedade do sistema ou como uma restrição que deve ser satisfeita no seu processo de desenvolvimento (FALBO, 2012).

Para que o software cumpra seu papel em um negócio, é vital que ele funcione de acordo com os requisitos estabelecidos para a criação do software. Assim, é preciso identificar e entender plenamente os requisitos do negócio. O papel da Engenharia de Requisitos consiste em coletar os requisitos, analisá-los, documentá-los e gerenciá-los ao longo do ciclo de vida do software (AURUM; WOHLIN, 2005).

Entre as técnicas para realizar o levantamento desses requisitos, estão a leitura de documentos, entrevista e a observação. Em uma entrevista, o analista assume o papel de entrevistador, e mantém uma conversa direcionada, com um propósito específico, com o cliente, ou principal interessado no sistema. Existem algumas maneiras de se conduzir uma entrevista: pode ser fechada, seguindo um conjunto de perguntas bem escolhidas; ou aberta, sem um roteiro previamente escolhido, passando por vários assuntos.

A técnica de observação para o levantamento de requisitos é de ajuda pois muitas vezes as pessoas ficam imersas em seu trabalho e o fazem de modo intuitivo (FALBO, 2012), e nem sempre conseguem se aperceber dos detalhes de seus afazeres. Na técnica de leitura de documentos, a interpretação destes pode ajudar no levantamento de informações. Documentos como relatórios usados em tomadas de decisão, fichas e formulários, podem dar informações que seriam difíceis de se obter utilizando outras técnicas de levantamento, como entrevista e observação (FALBO, 2012).

Para criar os modelos gerados pela análise de requisitos foi adotada a UML - *Unified Modeling Language*, uma linguagem gráfica padrão para especificar, visualizar, documentar

e construir artefatos de sistemas de software (BOOCH; JACOBSON; RUMBAUGH, 2006), tais como :

- Diagrama de Classe : Um diagrama que exhibe um conjunto de classes e seus relacionamentos, contendo as entidades do sistema;
- Diagrama de Casos de Uso : Um diagrama que descreve as funcionalidades do sistema percebida pelos usuários; proveem uma visão das funcionalidades do sistema, sendo importantes para a organizá-las e modelá-las (FALBO, 2012);
- Diagrama de Sequência : Representa em detalhes a execução de um caso de uso, desde o ator que o solicita, passando pelos objetos que participam da interação e seus métodos, exibindo mensagens específicas, condicionais e *loops*, tudo em ordem cronológica. Isso proporciona ao leitor uma clara identificação do fluxo de controle ao longo do tempo.

A Figura 2 ilustra as atividades de modelagem propriamente ditas, bem como os diagramas e documentações produzidas para o PDEMT.

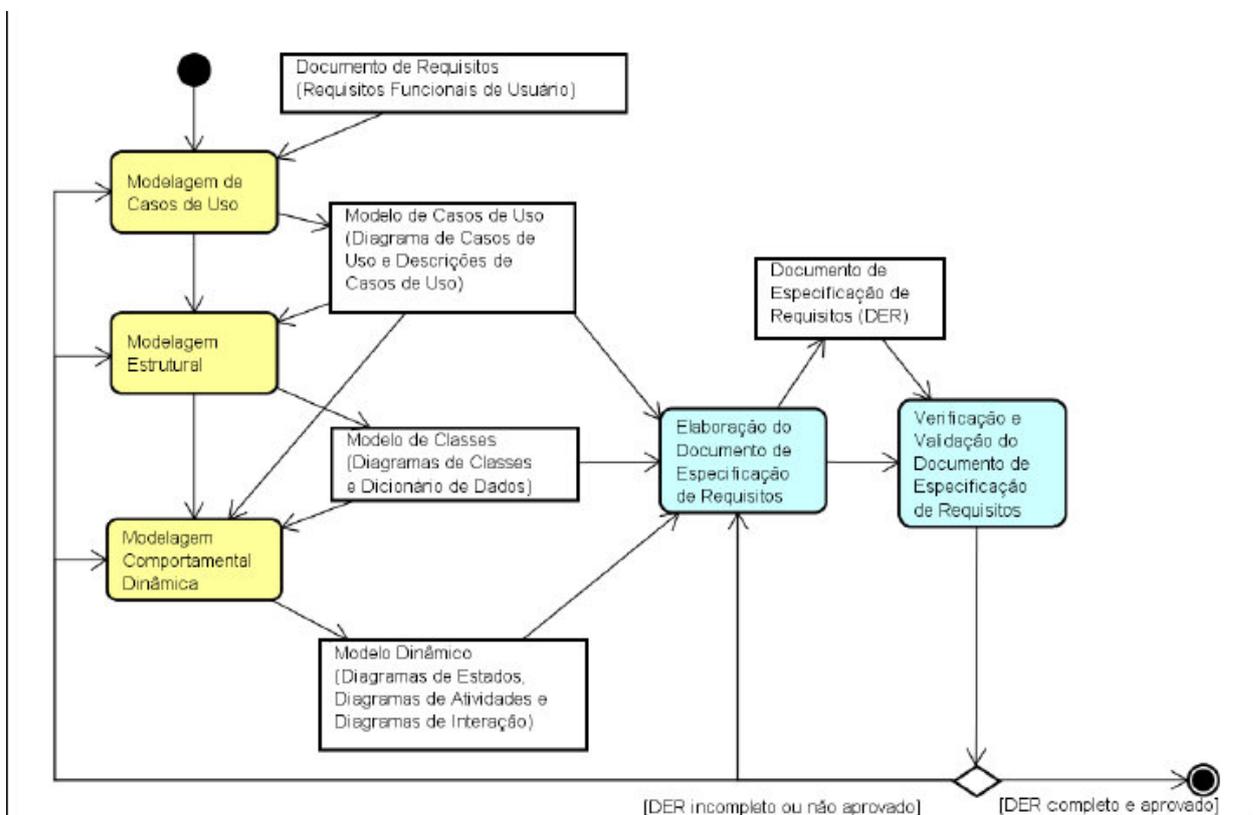


Figura 2 – Atividades de modelagem e artefatos produzidos

2.3 Padrões Utilizados na Implementação

O uso de padrões de projeto descrevem soluções para problemas recorrentes no desenvolvimento de software, e seu uso aumenta a qualidade do código, deixando-o mais flexível, elegante e reusável. A seguir os padrões usados no PDEMT.

2.3.1 Camada de Serviço

O padrão Camada de Serviço estabelece uma fronteira entre a aplicação e a camada de serviços, formada por um conjunto de operações disponíveis. Ela encapsula a lógica de negócios do aplicativo, controlando transações e coordenando as respostas na execução de suas operações (FOWLER, 2002). A Figura 3 ilustra isso.

Essa camada é caracterizada pela divisão da lógica de negócio em duas partes: a lógica do domínio, que tem a ver puramente com o domínio do problema e suas operações básicas; e a lógica de aplicação, que tem a ver com as responsabilidades de aplicação, ou seja, os casos de uso.

Com respeito a lógica de aplicação, ela pode ser implementada de várias maneiras: considerar que cada caso de uso dará origem a uma classe de aplicação; agrupar todos os casos de uso em apenas uma classe de aplicação; ou por fim, usar o meio termo entre essas duas abordagens, que seria usar várias classes de aplicação, agrupando os casos de uso por contexto.

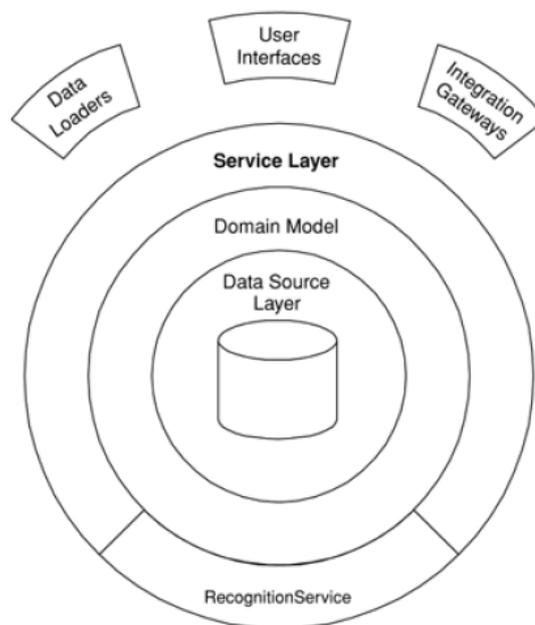


Figura 3 – Representação da Service Layer, ou Camada de Serviço em português

2.3.2 DAO

Data Access Object - DAO, é um padrão para persistência de dados que permite separar regras de negócio das regras de acesso a banco de dados; atua como um intermediário entre a aplicação e o banco de dados; faz com que o mapeamento entre objetos do domínio e tabelas do banco de dados seja totalmente isolada da lógica de negócio. Esse padrão define uma interface genérica de operações básicas de persistência, como inserir, alterar, recuperar, deletar e consultar e uma superclasse genérica que implementa essa interface usando um mecanismo de persistência (BAUER; KING, 2006).

2.4 APIs

API é o acrônimo de *Application Programming Interface*, e pode ser definido como um conjunto de rotinas e padrões estabelecidos por um software para a utilização das suas funcionalidades por outros sistemas. No contexto da Web, uma API é um conjunto definido de mensagens de requisição e resposta HTTP, geralmente expressado nos formatos XML ou JSON. Mesmo um sistema *desktop* pode fazer uso de uma API que usa requisições HTTP.

Nesse projeto, foi feito o uso de API's. Dentre elas destaca-se o Google Calendar, que faz requisições HTTP. Ao invés de implementar todas as manipulações de um calendário como criar, alterar, visualizar e remover eventos, utilizamos essa API, feita pelo Google, que já provê todas essas funcionalidades, inclusive os métodos para a autenticação em uma conta do Google. A API possui documentação e é de fácil uso.

Dentre as API's que não fazem uso da internet, temos o **iText-pdf**. O **iText-pdf** foi utilizado para a geração de relatórios em PDF. Essa API possui uma série de funções para criação e manipulação de arquivos em formato PDF. Para realizar a integração do banco de dados com a linguagem de programação Java, utilizamos a API **JDBC**, que provê os mecanismos necessários de integração com muitos bancos de dados diferentes. Por fim, o **jCalendar** foi utilizado para manipular datas e fazer com que o usuário do sistema tenha uma melhor experiência ao querer informar uma data como entrada de dados.

3 Análise e Especificação de Requisitos

A Engenharia de Requisitos é o processo pelo qual os requisitos de um produto de software são coletados, analisados, documentados, validados e gerenciados ao longo de todo o ciclo de vida do software (FALBO, 2012).

Este capítulo aborda alguns resultados da Engenharia de Requisitos para a construção do sistema PDEMT. Na seção 3.1, é apresentado o escopo do projeto; na seção 3.2, são apresentados diagramas de casos de uso e na seção 3.3, são apresentados os diagramas de classes. Os requisitos funcionais, requisitos não funcionais e regras de negócio podem ser encontrados no Apêndice.

3.1 Descrição do Escopo

Semanalmente, as Testemunhas de Jeová participam da Escola do Ministério Teocrático. Nessa escola são realizados 4 discursos: Destaques da Leitura Semanal da Bíblia (Destaques), Discurso Número 1, Discurso Número 2 e Discurso Número 3. O discurso do tipo Destaques é designado apenas para irmãos (homens) que possuem o cargo de Servo Ministerial ou Ancião. Quanto ao Discurso Número 1, qualquer irmão (novamente, apenas homens) pode realizá-lo; já o Discurso Número 2, apenas irmãs podem fazê-lo. Por fim, o Discurso Número 3 pode ser realizado tanto por irmãos quanto por irmãs. Todos os 4 discursos possuem fonte de matéria.

Em relação aos Discursos Número 1, 2 e 3, a pessoa que proferi-los será observada em um dos pontos de característica de oratória que se encontram no livro Beneficie-se da Escola do Ministério Teocrático, publicado pelas Testemunhas de Jeová. Além disso, as partes realizadas por uma irmã (Discurso Número 2 ou 3) terão uma ajudante (que deve ser uma irmã) e uma cena, a ser escolhida no livro Beneficie-se da Escola do Ministério Teocrático. Os discursos também possuem um tema: os Discursos Destaques e Número 1 tem como tema a própria fonte de matéria; já os Discursos Número 2 e 3 possuem temas específicos. Todas as informações referentes à programação do ano da Escola do Ministério Teocrático (data, tema, fonte dos discursos) se encontram no Nosso Ministério do Reino, também publicado pelas Testemunhas de Jeová.

A cada 2 meses acontece a Recapitulação da Escola do Ministério Teocrático. Nessa ocasião, apenas o discurso dos Destaques da Leitura Semanal da Bíblia é feito. O dirigente da Escola do Ministério Teocrático matricula e desmatricula os membros dessa escola, além de designar irmãos e irmãs para fazerem os discursos, definindo também a característica de oratória a ser observada, cena e ajudante.

O dirigente possui todas essas informações sobre discursos em uma folha. Com antecedência, os irmãos e irmãs que farão discursos recebem um aviso com todas as informações necessárias sobre o discurso para se prepararem. Também é fixada no Quadro de Anúncios da Congregação uma tabela com toda a programação da Escola do Ministério Teocrático.

3.2 Modelo de Casos de Uso

Os diagramas de casos de uso definem quais funcionalidades que um sistema deve oferecer. Tais diagramas possuem dois principais elementos: atores e casos de uso (FALBO, 2014). A Tabela 1 mostra os atores identificados no contexto do sistema.

Ator	Descrição
Dirigente da EMT	Executa as atividades de cadastro de alunos, cenas, oratórias e discursos, designações, consultas e relatórios.
Google Calendar	Cria as designações na agenda do Google.

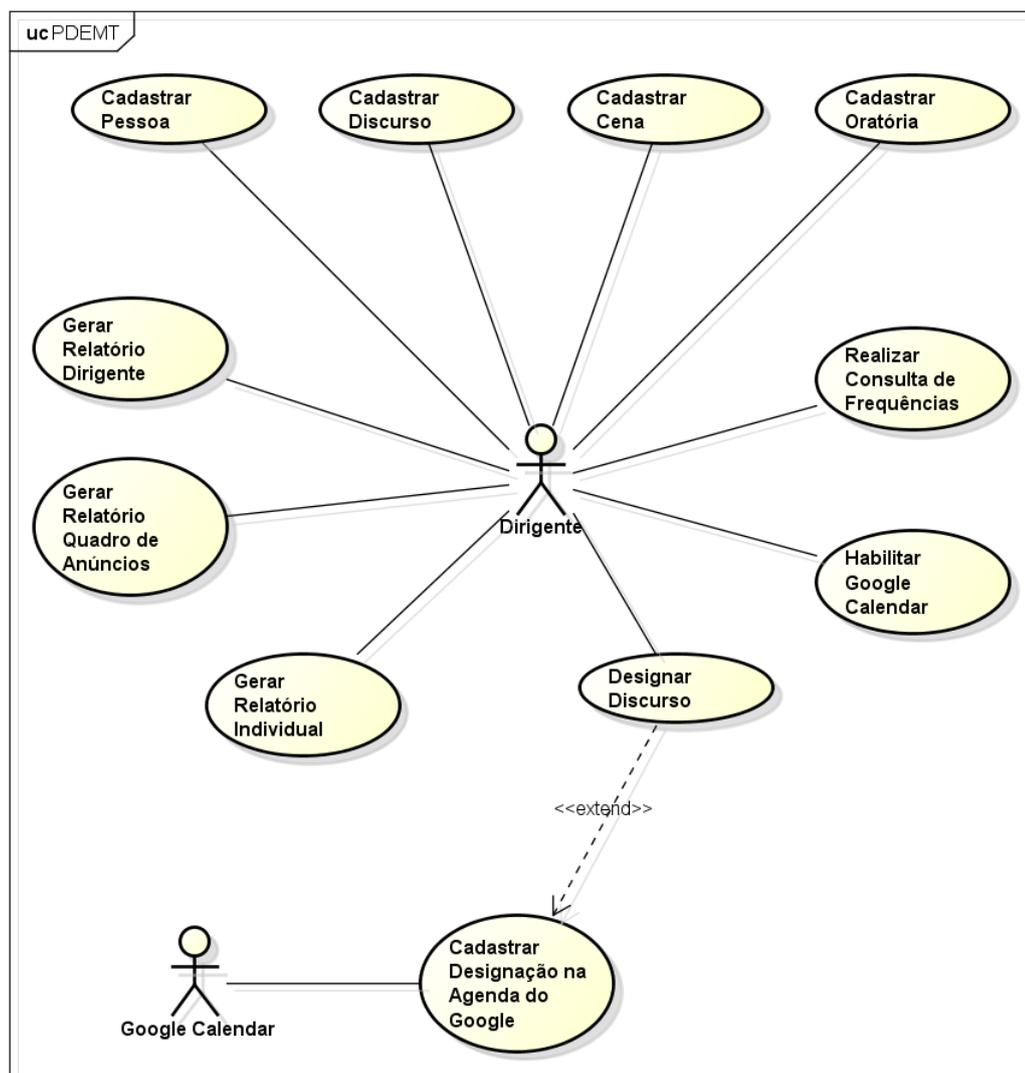
Tabela 1 – Atores do PDEMT

A Figura 4 apresenta os casos de uso do sistema, e em seguida temos breves descrições deles. A descrição completa de cada caso de uso encontra-se no Apêndice.

Os casos de uso **Cadastrar Pessoa**, **Cadastrar Discurso**, **Cadastrar Cena** e **Cadastrar Oratória** são do tipo cadastrais e incluem alteração, inclusão, consulta e exclusão. Podemos ver **Cadastrar Discurso** como um caso de uso especial, pois um discurso não pode ser cadastrado individualmente, mas cadastra-se a programação discursos de um ano inteiro, tendo como entrada o ano e a programação anual de discursos daquele ano em um arquivo texto.

As informações sobre todos esses casos de uso são necessários para que se possa implementar o caso de uso **Designar Discurso**, em que o dirigente da EMT designa um discurso, podendo ter uma cena, oratória e ajudante, a um aluno da EMT (JEOVÁ, 2008). Através do caso de uso **Habilitar Google Calendar**, o dirigente pode escolher incluir ou não a integração com o Google Calendar, em que o aluno da EMT terá marcado em sua agenda pessoal a designação do discurso. É exclusivamente nesse caso que entra em ação o ator Google Calendar, identificado na Tabela 1. Quando o Google Calendar é habilitado, o caso de uso **Criar Designação na Agenda do Google** é acionado, e cria-se então o evento, o discurso designado, na Agenda do Google.

O dirigente da EMT também pode gerar relatórios para determinado período de tempo. O caso de uso **Gerar Relatório Dirigente** entrega uma tabela contendo toda a programação de discursos e características de tais discursos como tema, fonte, data, quem irá proferi-lo, quem será a ajudante, qual será a cena e em que característica de oratória o



powered by Astah

Figura 4 – Diagrama de Casos de Uso

participante será observado. Para o caso de uso **Gerar Relatório Individual**, similar ao **Gerar Relatório Dirigente**, são fornecidas pelo sistema informações a respeito de cada parte, no entanto, ao invés de ter o formato de uma tabela, esse relatório é feito de forma individual, para que cada discurso com suas características possam ser entregues àqueles que farão o proferimento. Por fim, o caso de uso **Gerar Relatório Quadro de Anúncios** gera uma tabela contendo data, tema, orador e ajudante dos discursos referentes ao período de tempo escolhido, para que o dirigente possa conduzir a EMT da maneira como planejou e avisando a congregação e os alunos da EMT das designações que foram feitas.

Antes de fazer as designações, o dirigente da EMT pode **Realizar Consulta de Frequências**, verificando as frequências de uso das cenas, dos alunos, das oratórias e das ajudantes na EMT.

3.3 Diagrama de Classes

Abstraindo conceitos do mundo real para classes de objetos, atributos e relações, foi criado o diagrama de classes, que mostra os atributos e as associações entre as classes do sistema. A Figura 5 apresenta o diagrama de classes do sistema.

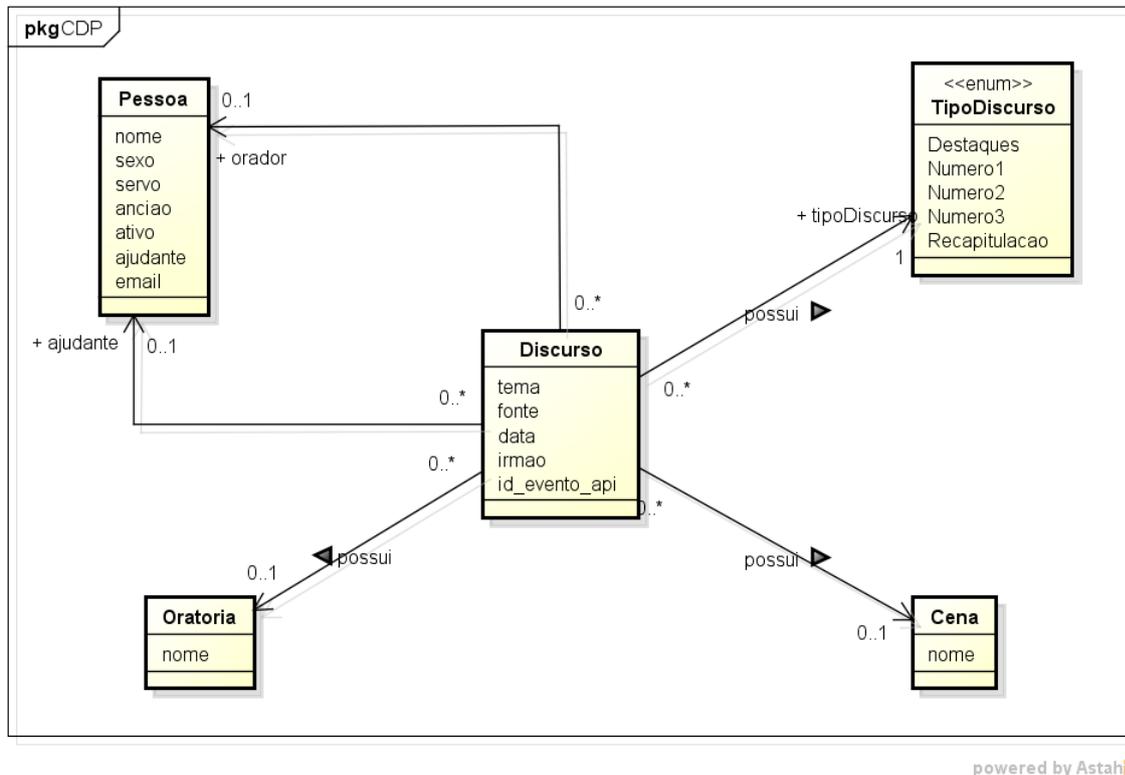


Figura 5 – Diagrama de Classes

Na EMT existem alguns **Tipos de Discursos** : Destaques, Número 1, Número 2, Número 3 e Recapitulação Escrita. De um **Discurso**, que é a principal entidade do sistema, deseja-se saber o tema, a fonte de matéria do discurso, a data em que o discurso será feito, o tipo de discurso e se o discurso será feito por um irmão (homem) ou irmã (mulher). De uma **Cena** e de uma **Oratória**, deseja-se saber o nome, os quais devem ser consultados no livro *Beneficie-se da Escola do Ministério Teocrático* (JEOVÁ, 2008). Orador e ajudante são pessoas, da classe **Pessoa**. De uma **Pessoa** deseja-se saber o nome, sexo, se possui o cargo de Servo Ministerial, se possui o cargo de Ancião, se está ativa para receber designações e se pode participar como ajudante.

Para que um **Discurso** possa ser designado, cada Discurso deve atender a alguns requisitos. Para designar um discurso do tipo **Destaques**, ele deve possuir apenas um Orador, do tipo **Pessoa**. Para designar um discurso do tipo **Número 1**, ele deve possuir um Orador, do tipo **Pessoa** e uma **Oratória** em que o Orador será observado. Para designar um discurso do tipo **Número 2**, a ser feito por uma irmã(mulher), ele deve possuir um Orador, do tipo **Pessoa**, uma **Oratória** em que o Orador será observado, uma

Cena em que se passará o Discurso e uma Ajudante, do tipo **Pessoa**. Para designar um discurso do tipo **Número 3**, se for irmã(mulher) que fará o proferimento, o discurso deve possuir um Orador, do tipo **Pessoa**, uma **Oratória** em que o Orador será observado, uma **Cena** em que se passará o Discurso e uma Ajudante, do tipo **Pessoa**. Se for proferido por um irmão(homem), o Discurso deve possuir um Orador, do tipo **Pessoa** e uma **Oratória**. Por fim, se for do tipo **Recapitulação**, o discurso não é designado para nenhuma pessoa específica, nada é feito.

4 Projeto e Implementação

Após a identificação, documentação e modelagem dos requisitos, ocorre a fase de projeto, em que o software é modelado de forma que os aspectos tecnológicos que serão utilizados para a implementação do sistema sejam levados em conta. Esses aspectos envolvem: linguagem de programação, bibliotecas e APIs utilizadas, características de interface com o usuário, arquitetura de software e forma de persistência de dados. Por fim, o produto de software é construído e testado.

Neste capítulo, são apresentados os principais aspectos de Projeto do PDEMT. Na seção 4.1, a arquitetura de software é descrita. Na seção 4.2, as APIs utilizadas no projeto são apresentadas. Na seção 4.3 é feito o detalhamento dos componentes da arquitetura. Por fim, na seção 4.4, são apresentadas algumas telas e características da ferramenta. O projeto completo encontra-se nos Apêndices, que também contém informações sobre as tecnologias utilizadas e sobre as táticas de projeto consideradas para atender os requisitos não funcionais identificados no Documento de Requisitos do PDEMT.

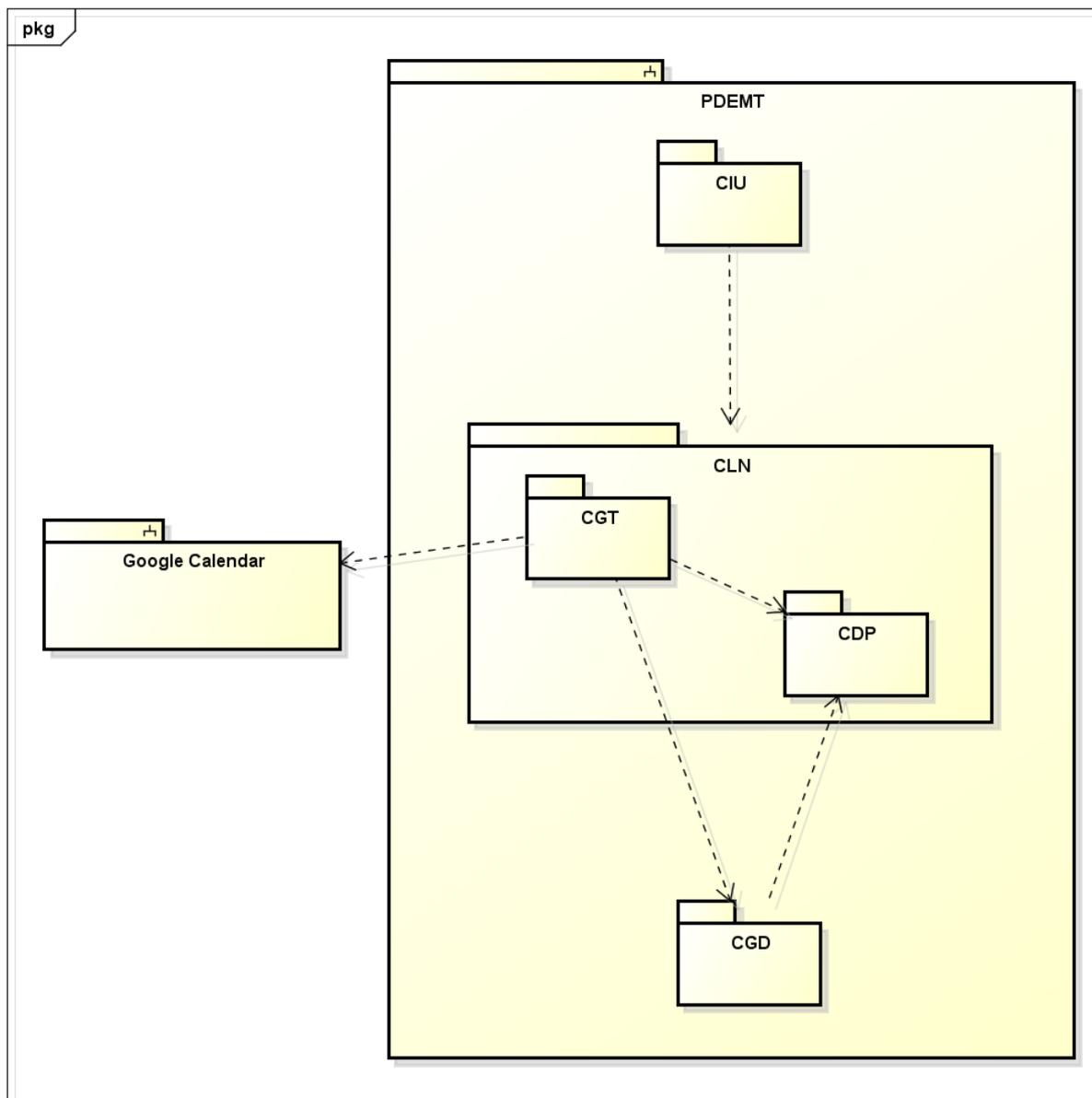
4.1 Arquitetura de Software

A arquitetura de software do PDEMT está dividida em camadas, conforme a Figura 6. Devido à complexidade do sistema, não foi necessário dividi-lo em subsistemas. A arquitetura é composta por 3 camadas, a saber:

- Camada de Interface com o Usuário (CIU), que trata de aspectos relacionados às interfaces gráficas com os usuários;
- Camada de Lógica de Negócio (CLN), onde são implementadas as funcionalidades do sistema; e
- Camada de Gerência de Dados (CGD), responsável pela persistência de objetos.

A camada CLN é dividida ainda em duas outras: Camada do Domínio do Problema (CDP) e Camada de Gerência de Tarefas (CGT). A Figura 6 retrata bem isso.

Destaca-se na CGT, especificamente falando da classe `Ap1Discurso`, a integração com o Google Calendar, para criação de eventos na Agenda do Google.



powered by Astah

Figura 6 – Arquitetura de Software do PDEMT

4.2 API's Utilizadas

Na implementação do PDEMT foram utilizadas duas APIs existentes: a do Google Calendar, provida pela Google, e a API JDBC, que faz parte da distribuição padrão de Java e o iText, responsável pela geração de relatórios.

4.2.1 Google Calendar

O objetivo da integração com essa API (GOOGLE, 2015b) é lembrar/avisar o participante da EMT de sua designação de discurso. Assim, tendo ele uma conta Google¹—

¹ <<https://accounts.google.com/>>

e, por conseguinte, um e-mail no Gmail²—, ele pode ser notificado, até mesmo por meio de um celular do tipo *smartphone* (ex.: celulares Android, iPhone, Windows Phone, etc.), pois este contém (ou pode ser instalado) o aplicativo Calendário integrado ao Google.

Para que o objetivo do projeto seja alcançado, decidiu-se criar uma agenda com o nome “Calendário do PDEMT”, em uma conta do Google. Assim, todas as designações são inseridas nessa agenda - são os chamados **Eventos**.

Dentre as propriedades de um Evento, existe uma chamada “Attendees”. Isso significa que um Evento possui uma lista de participantes. Assim, para que um orador/ajudante da EMT possa ser avisado de seu discurso pela agenda do Google, ele é marcado como um participante desse discurso, por meio do seu endereço de email do Gmail. Assim, o Evento criado em “Calendário do PDEMT” é replicado na agenda pessoal do participante.

O primeiro passo para utilização do Google Calendar é a autenticação. Muitos métodos foram encontrados na internet, sendo alguns depreciados e/ou desatualizados. A opção escolhida foi disponibilizada pela Google API Client Library for Java (GOOGLE, 2015a), como mostram as Listagens 4.1 e 4.2:

Listagem 4.1 – Método *authorize*, que faz a leitura do arquivo *client secrets*, usado na autenticação em uma conta do Google

```

1  public Credential authorize() throws Exception {
2      GoogleClientSecrets clientSecrets = GoogleClientSecrets.load(JSON_FACTORY
3          , new InputStreamReader(new FileInputStream(new File("client_secrets.
4              json"))));
5      if (clientSecrets.getDetails().getClientId().startsWith("Enter") ||
6          clientSecrets.getDetails().getClientSecret().startsWith("Enter ")) {
7          System.out.println("Enter Client ID and Secret from https://code.
8              google.com/apis/console/?api=calendar into client_secrets.json");
9          System.exit(1);
10     }
11     GoogleAuthorizationCodeFlow flow = new GoogleAuthorizationCodeFlow.
12         Builder(httpTransport, JSON_FACTORY, clientSecrets, Collections.
13             singleton(CalendarScopes.CALENDAR)).setDataStoreFactory(
14             dataStoreFactory).build();
15     return new AuthorizationCodeInstalledApp(flow, new LocalServerReceiver())
16         .authorize("user");
17 }

```

Listagem 4.2 – Construtor da classe que gerencia o Google Calendar

```

1  public GerenciadorGoogleCalendar() throws GeneralSecurityException,
2      IOException, Exception {
3      httpTransport = GoogleNetHttpTransport.newTrustedTransport();
4      dataStoreFactory = new FileDataStoreFactory(DATA_STORE_DIR);
5      credential = authorize();
6      client = new com.google.api.services.calendar.Calendar.Builder(
7          httpTransport, JSON_FACTORY, credential).setApplicationName(
8          APPLICATION_NAME).build();
9  }

```

² <<https://mail.google.com/>>

```

6     loadProperties();
7 }

```

Ao invés de usar login e senha para autenticação, esse método faz a leitura do arquivo *client secrets*, que pode ser obtido³ no formato *.json*, após a criação de um projeto no site Google Developer.

Para a criação de eventos, é preciso passar como parâmetro um identificador de calendário, que no caso do PDEMT estará guardado no arquivo de configuração, e o próprio evento, que é populado com as informações das designações.

4.2.2 JDBC

A API Java Database Connectivity (JDBC) permite o desenvolvimento de aplicações conectadas a banco de dados com grande performance. JDBC alcança seu objetivo por meio de um conjunto de interfaces em Java, sendo que este conjunto é implementado diferentemente por cada fabricante. Ao conjunto de classes que implementam o mecanismo de um banco de dados particular se dá o nome de driver JDBC. Para o PDEMT, foi utilizado o Driver do PostgreSQL, o *postgresql-9.3-1101*. As classes e interfaces mais importantes e utilizadas são : *DriverManager*, *Connection*, *PreparedStatement* e *ResultSet*. Abaixo segue uma breve descrição da utilização de cada uma delas.

Listagem 4.3 – Método que cria a conexão com o banco

```

1  public static Connection getConnection() throws IOException {
2      try {
3          Class.forName("org.postgresql.Driver");
4          return DriverManager.getConnection(getHost(), getLogin(), getPassword
5              ());
6      } catch (ClassNotFoundException e) {
7          e.printStackTrace();
8      } catch (SQLException e) {
9          throw new RuntimeException(e);
10     }
11     return null;

```

A partir da Listagem 4.3, observamos o uso da classe **DriverManager** na linha 3. É nessa linha que o driver que se deseja usar é registrado na classe, para que se possa usar os métodos estáticos para gerenciar o driver JDBC. Após o driver estar inicializado, pode-se abrir uma conexão através do método *getConnection*, que retorna um objeto **Connection**, que é o método apresentado na Listagem 4.3. Foi usada a sobrecarga do método que pede as credenciais de login, senha e host, que estão no arquivo de configuração do PDEMT.

³ <<https://console.developers.google.com/project/primeval-smile-94121/apiui/credential?authuser=0>>

Fazemos uso de **PreparedStatement** na Listagem 4.4. Dessa forma é criada uma instrução em SQL que é executada por meio da conexão criada pelo método `getConnection`, mostrado na Listagem 4.3.

Listagem 4.4 – Inserção de uma cena usando o JDBC

```

1  public void inserir(Cena entidade) throws SQLException {
2      String insert = "INSERT INTO cena(nome) VALUES(?)";
3      super.inserir(insert, entidade.getNome());
4  }
5
6  protected void inserir(String insertSql, Object... parametros) throws
7      SQLException {
8      PreparedStatement pstmt = getConnection().prepareStatement(insertSql);
9      for (int i = 0; i < parametros.length; i++) {
10         pstmt.setObject(i + 1, parametros[i]);
11     }
12     pstmt.execute();
13     pstmt.close();
14     closeConnection();
15 }

```

Por fim, **ResultSet** representa o conjunto de resultados de uma tabela do banco de dados. Ele inicia apontando para a primeira linha dos resultados e assim, usando os métodos `getInt`, `getString`, `getDouble`, `getFloat`, `getLong` e outros, é possível recuperar todos os dados retornados da consulta, conforme mostra a Listagem 4.5.

Listagem 4.5 – Busca de pessoas com parâmetros usando o JDBC

```

1  public ArrayList<Pessoa> buscarPorSexoAtivas(String sexo) throws SQLException
2  {
3      ArrayList<Pessoa> pessoas = new ArrayList<>();
4      String select = "SELECT * FROM pessoa WHERE sexo=? and ativo=true";
5      PreparedStatement stmt = getConnection().prepareStatement(select);
6      stmt.setObject(1, sexo);
7      ResultSet rs = stmt.executeQuery();
8      while (rs.next()) {
9          Pessoa p = new Pessoa();
10         p.setId((int) rs.getLong("id_pessoa"));
11         p.setNome(rs.getString("nome"));
12         p.setEmail(rs.getString("email"));
13         p.setSexo(rs.getString("sexo"));
14         p.setAjudante(rs.getBoolean("ajudante"));
15         p.setServo(rs.getBoolean("servo"));
16         p.setAnciao(rs.getBoolean("anciao"));
17         p.setAtivo(rs.getBoolean("ativo"));
18         pessoas.add(p);
19     }
20     rs.close();
21     stmt.close();
22     return pessoas;
23 }

```

4.2.3 iText-pdf

O uso dessa API provê a funcionalidade de gerar relatórios no formato PDF. Ela possui diversas características, como: gerar arquivos PDF, gerar formulários em PDF para preenchimento, dividir, fundir, ou sobrepor arquivos PDF, proteger seus arquivos PDF com senha, dentre outras⁴. Para criar e manipular os arquivos, a API possui métodos e classes com grande variedade de funcionalidades, tais como criar e editar tabelas e células, bem como todo seu *layout*; criar e editar parágrafos; alterar a fontes e cores e tamanho da página. Foram criados métodos para cada tipo de relatório a ser gerado pelo PDEMT, e ao gerar esses relatórios, são salvos em uma pasta nomeada com o período que abrange os relatórios.

Para a geração dos relatórios, salvo as particularidades de cada um, o procedimento básico foi: criar um documento, definindo seu tamanho (no caso, A4); definir o título do documento; criar uma tabela, criar o cabeçalho da tabela, criar células e preenchê-las com parágrafos; e por fim, adicionar essas células na tabela. As Figuras 23, 24 e 25 mostram exemplos de cada tipo de relatório gerado pelo PDEMT.

4.3 Detalhamento dos Componentes da Arquitetura

O PDEMT está organizado em três camadas: Camada de Lógica de Negócio, Camada de Interface com o Usuário e Camada de Gerência de Dados. A seguir o projeto dessas camadas é apresentado.

4.3.1 Camada de Lógica de Negócio (CLN)

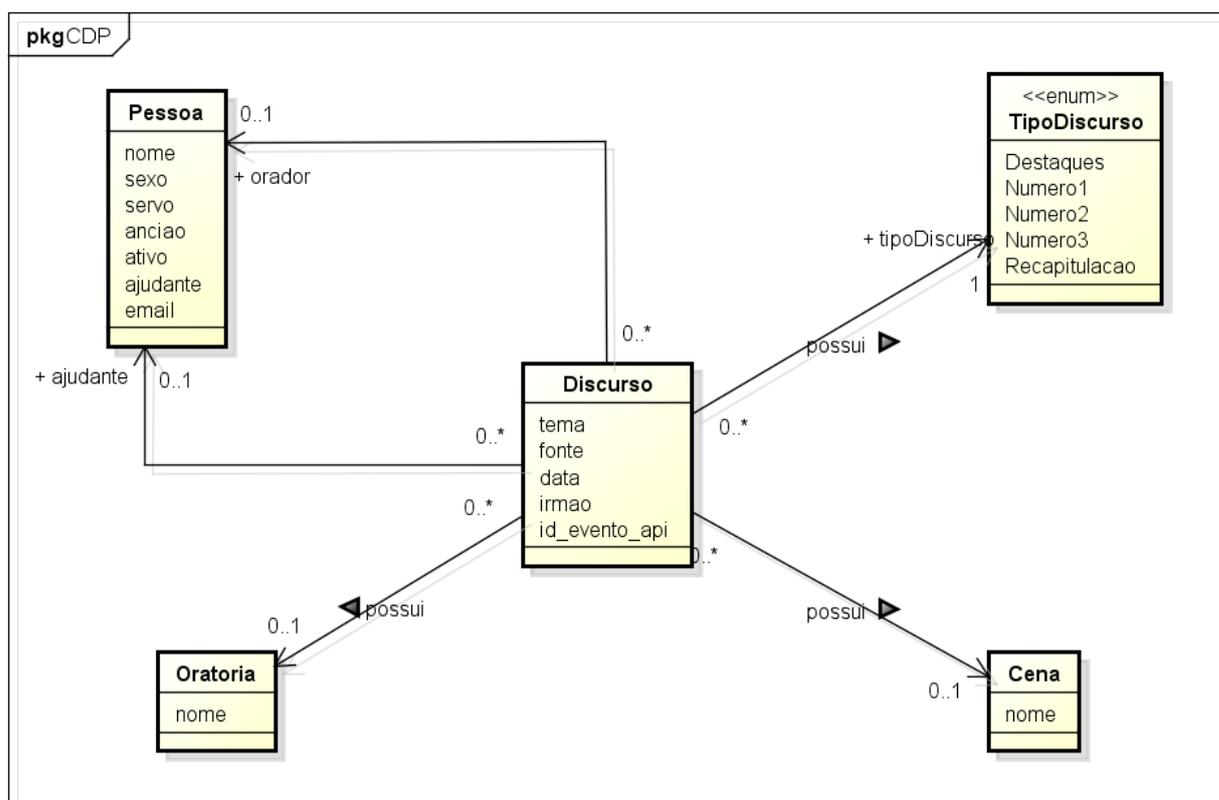
Para organizar a Camada de Lógica de Negócio, foi escolhido o padrão Camada de Serviço (FOWLER, 2002). Sendo assim, essa camada é dividida em dois componentes: Componente de Domínio do Problema (CDP) e Componente de Gerência de Tarefas (CGT), como apresentado anteriormente na Figura 6. Esse padrão utiliza um componente para tratar a lógica de aplicação (o CGT), o qual recebe as requisições da interface, e um componente para tratar os conceitos do domínio do problema, advindos do modelo conceitual estrutural elaborado na fase de análise (o CDP).

A Figura 7 apresenta o diagrama de classes do CDP. Partindo da Figura 5, as navegabilidades entre as classes foram identificadas, os tipos específicos de domínio foram representados como classes e a visibilidade dos atributos foi explicitada.

No projeto do CGT, optou-se por agrupar os casos de uso de mesmo contexto em uma mesma classe de aplicação, conforme as Figuras 8 e 9.

A Tabela 2 sumariza as relações existentes entre as classes do CGT e os casos

⁴ <<http://itextpdf.com/functionality>>



powered by Astah

Figura 7 – Componente de Domínio do Problema - CDP

de uso por elas tratados. É preciso dizer que o caso de uso **Designar Discurso** é formado pelos métodos **designarDestaques**, **designarNumero1**, **designarNumero2** e **designarNumero3**.

Classe	Casos de Uso
AplDiscurso	Cadastrar Discurso, Designar Discurso, Habilitar Google Calendar, Criar Designação na Agenda do Google
AplCena	Cadastrar Cena
AplOratória	Cadastrar Oratória
AplPessoa	Cadastrar Pessoa
AplEmitirRelatórios	Gerar Relatório Dirigente, Gerar Relatório Quadro de Anúncios, Gerar Relatório Individual
AplRealizarConsultas	Realizar Consulta de Frequência

Tabela 2 – Mapeamento entre Classes do CGT e Casos de Uso

4.3.2 Camada de Interface com o Usuário (CIU)

Devido a baixa complexidade e o fato de ter mecanismos suficientes para uma boa modularização, não foi necessária a criação de classes controladoras para realizar a

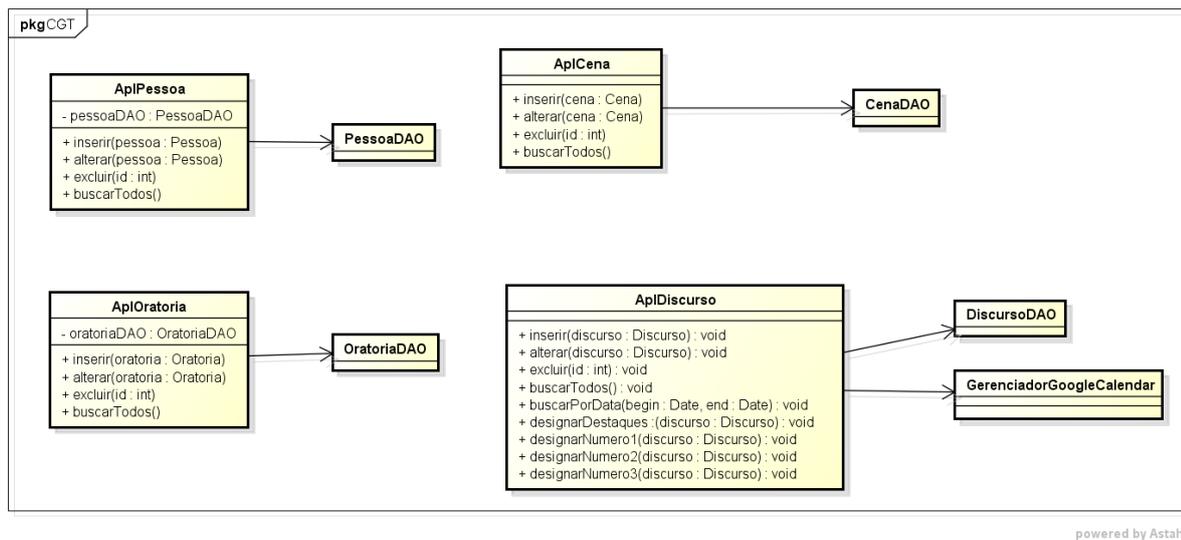


Figura 8 – Componente de Gerência de Tarefas dos casos de uso mais básicos e designação de discursos

interação entre o modelo e a visão do sistema. O padrão Camada de Serviço (FOWLER, 2002) foi o suficiente para uma boa organização do código. A Figura 10 mostra a relação entre as classes de visão e as classes da CGT, apresentadas na seção 4.3.1.

4.3.3 Camada de Gerência de Dados (CGD)

Para organizar e padronizar essa camada, foi adotado o uso de interfaces. As mais comuns e principais operações de um objeto foram declaradas em uma interface genérica chamada **IGenericDAO**, a qual todas as interfaces exclusivas dos objetos herdam dela, conforme a Figura 11.

Para cada classe de domínio a ser persistida, foram criadas uma classe *DAO e uma interface I*DAO correspondente, que implementa a interface genérica IGenericDAO. A primeira deve herdar de GenericDAO, uma classe abstrata que possui as funcionalidades básicas de acesso ao mecanismo de persistência, e deve implementar a interface DAO associada. A Figura 12 mostra o mapeamento do sistema.

A Figura 13 mostra o modelo Entidade-Relacionamento que foi feito para a implementação das tabelas no banco de dados.

4.4 O PDEMT

Nesta seção, são apresentadas algumas telas da ferramenta PDEMT. A parte cadastral da ferramenta permite o armazenamento de alunos, discursos, cenas e oratórias. A partir desses dados, é possível designar discursos e registrá-los em sua agenda pessoal do Google, gerar relatórios a partir das designações realizadas e fazer consultas que apoiam

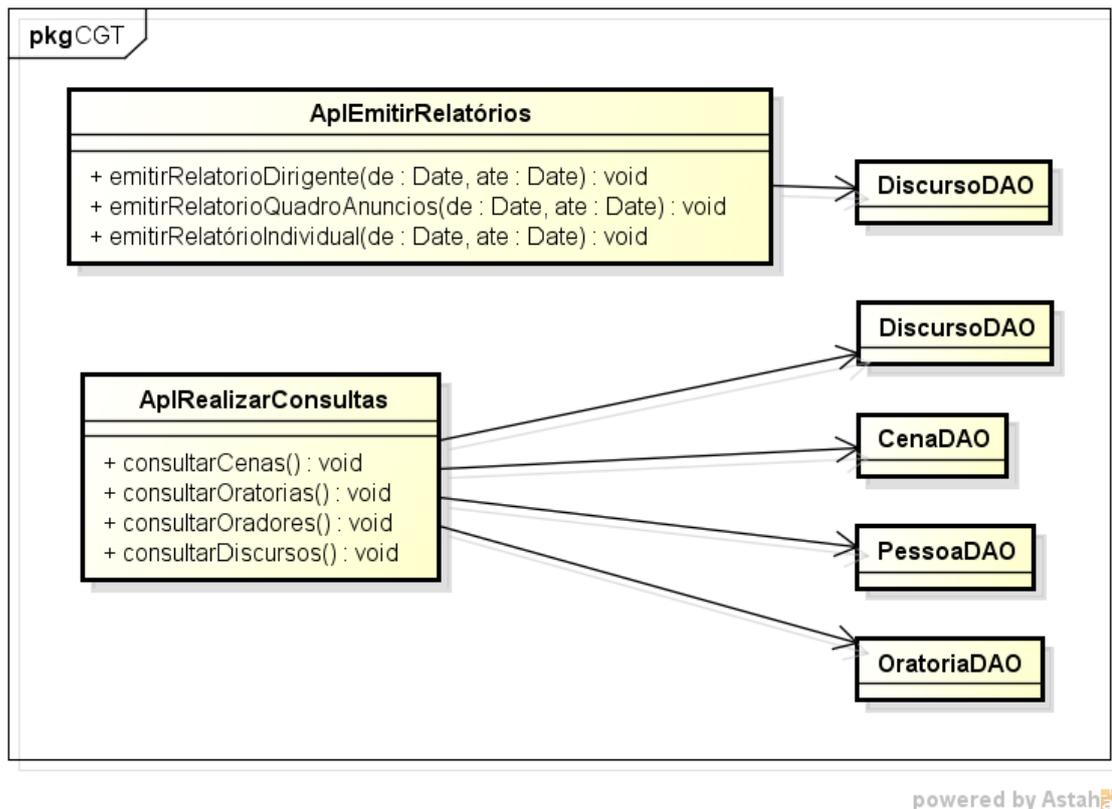


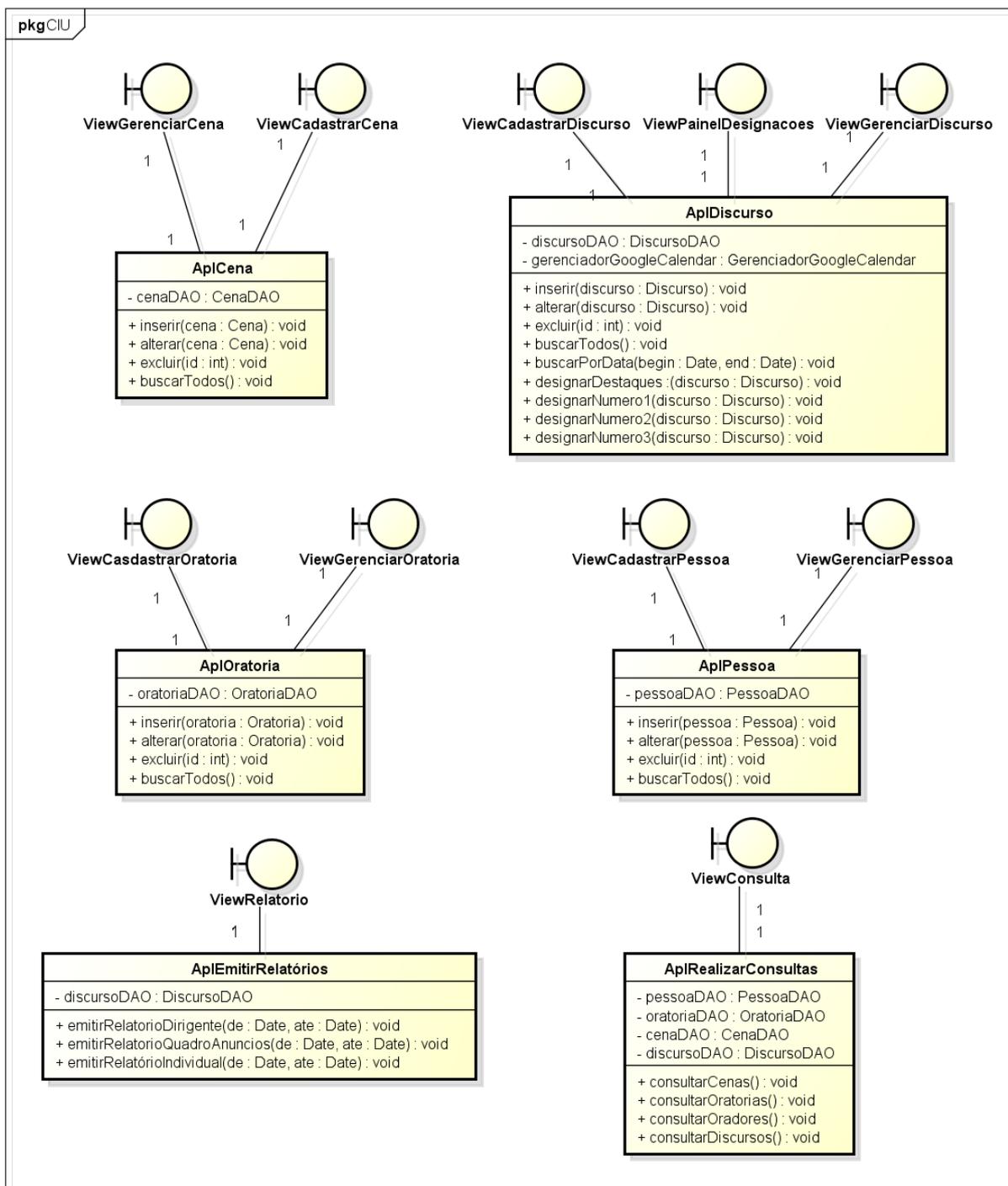
Figura 9 – Componente de Gerência de Tarefas dos casos de uso referentes a relatórios e consultas

as escolhas ao designar discursos. A Figura 14 mostra a tela principal do sistema.

A partir da tela principal, o usuário tem acesso a todas as funcionalidades do sistema. As funcionalidades estão distribuídas nos menus de acordo com a entidade, e não de acordo com o tipo de funcionalidade. Por exemplo, temos um menu chamado Alunos, que ao clicar nele são mostrados os itens de menu “Cadastrar Aluno” e “Gerenciar Alunos”. Esse é o padrão para todas as entidades de cadastro. A Figura 15 mostra a tela de cadastro de aluno, enquanto a Figura 16 mostra a tela de gerenciamento de alunos.

O PDEMT possui alguns padrões de tela. Por exemplo, as telas de cadastro são utilizadas exclusivamente para cadastrar entidades no sistema; só fazem a operação de criação. Já as telas de gerenciamento, fazem a função de consulta ou visualização, edição e exclusão.

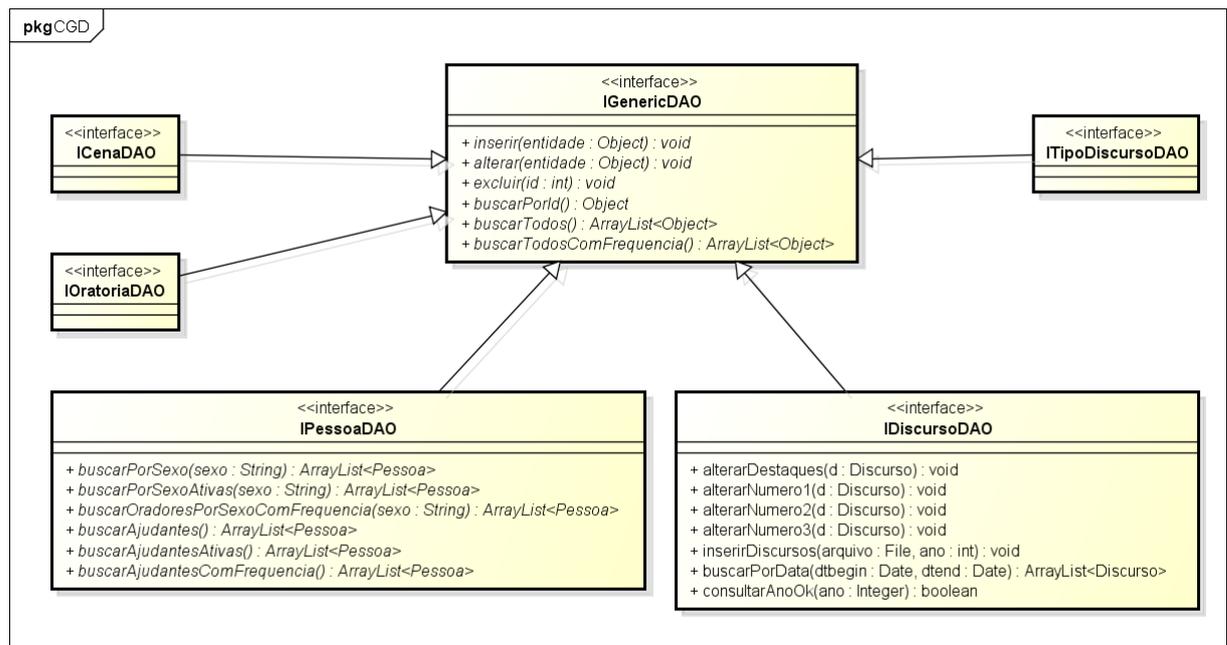
Após ter as entidades devidamente cadastradas, podemos usar a principal funcionalidade do PDEMT: a designação de discursos. Podemos selecionar o menu Discursos e em seguida a opção Designar, conforme a Figura 17. Após o clique, um *pop-up* é aberto, semelhante à Figura 18. Após a escolha do período que abrangerá as designações, é aberta uma tela contendo os discursos a serem designados, divididos primeiramente por semana, e depois por tipo de discurso, vide Figura 19.



powered by Astah

Figura 10 – Arquitetura da CIU

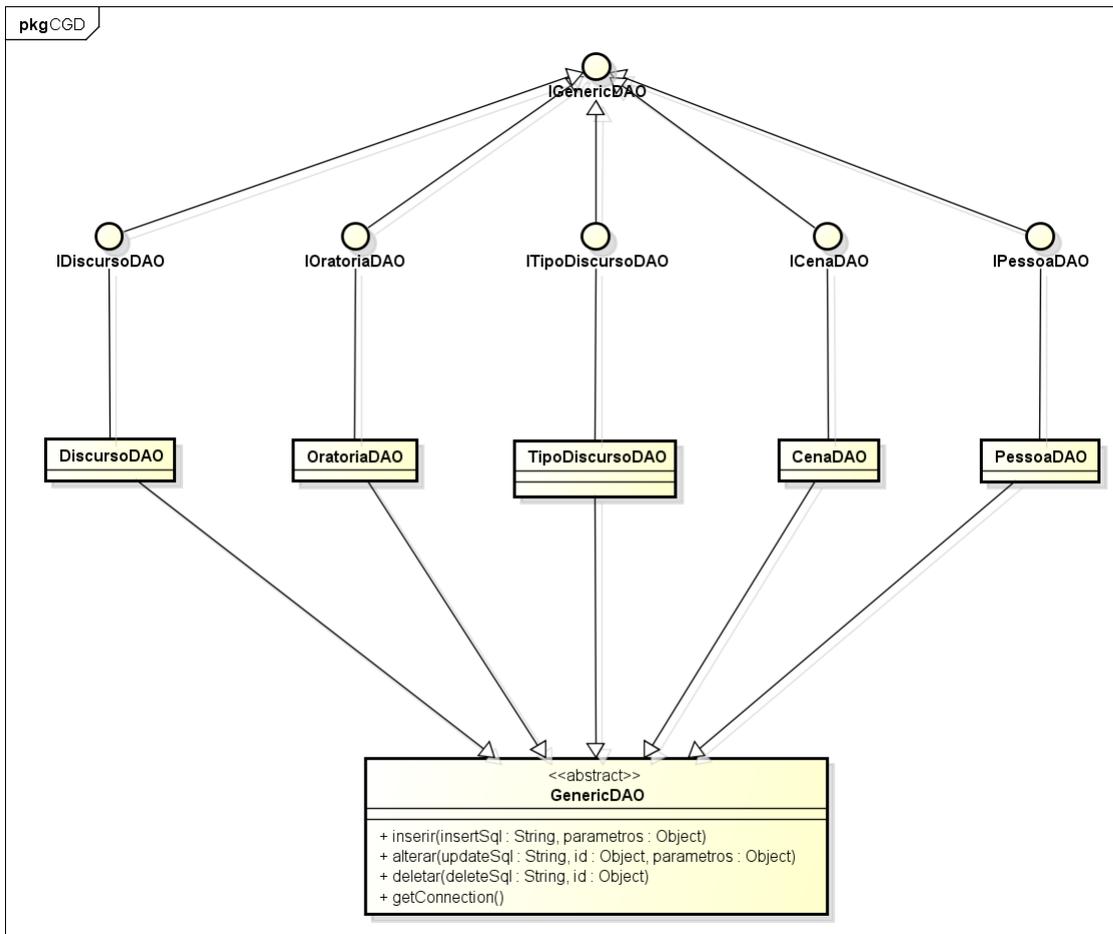
Após selecionar todas as opções para todos os discursos de cada semana, o usuário pode clicar no botão Salvar, que esta no painel de cada semana. A ação desse botão garante a persistência dos dados, salvando tudo sobre a designação. Caso a opção Habilitar Google Calendar estiver marcada, como na Figura 20, o PDEMT também criará eventos na agenda do Google de cada usuário, afim de que eles tenham ciência de suas designações. As Figuras 21 e 22 mostram uma designação recebida na agenda do Google.



powered by Astah

Figura 11 – Arquitetura da CGD

Finalmente, as Figuras 23, 24 e 25 mostram trechos dos relatórios gerados pelo PDEMT.



powered by Astah

Figura 12 – A Camada de Gerência de Dados

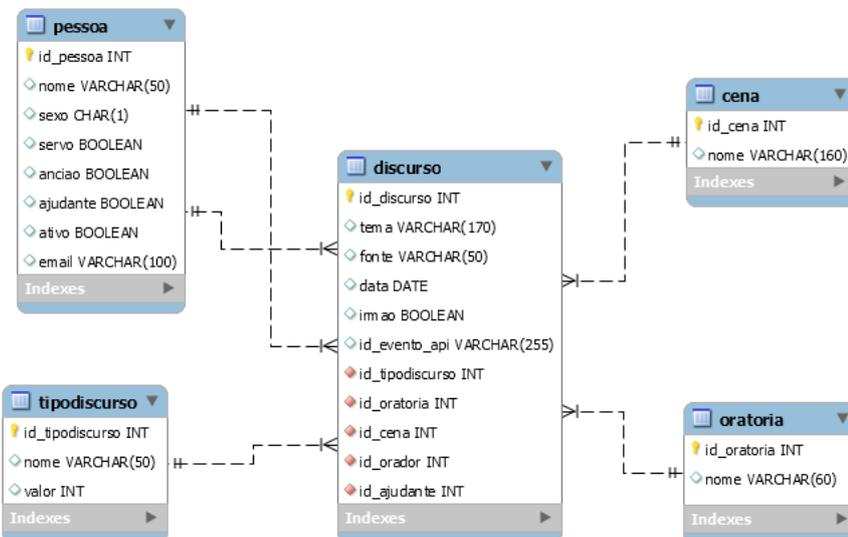


Figura 13 – Modelo Entidade-Relacionamento



Figura 14 – Tela principal do PDEMT

The screenshot shows a dialog box titled "Matricular Aluno". It contains the following fields and options:

- Nome: [Text input field]
- E-mail: [Text input field]
- Sexo: [Dropdown menu with "Masculino" selected]
- Servo:
- Ancião:
- Ajudante:
- Ativo:

At the bottom of the dialog are two buttons: "Cancelar" and "Salvar".

Figura 15 – Tela de Cadastro de Aluno

The screenshot shows a window titled "Gerenciar Alunos". On the left side, there is a list of student names: Adriana, Alexandre Babilone Fonseca (highlighted), Cilene, Emely, Irondilson, Júlio, Laisa Rossim, Nivaldo, Sara, Wagner, and Wallace. On the right side, there are fields and checkboxes for the selected student:

- Nome: Alexandre Babilone Fonseca
- E-mail: alexandrebabelone@gmail.com
- Sexo: [Dropdown menu with "Masculino" selected]
- Servo:
- Ancião:
- Ajudante:
- Ativo:

At the bottom of the window are three buttons: "Voltar", "Salvar Alterações", and "Excluir".

Figura 16 – Tela de Gerenciamento de Alunos

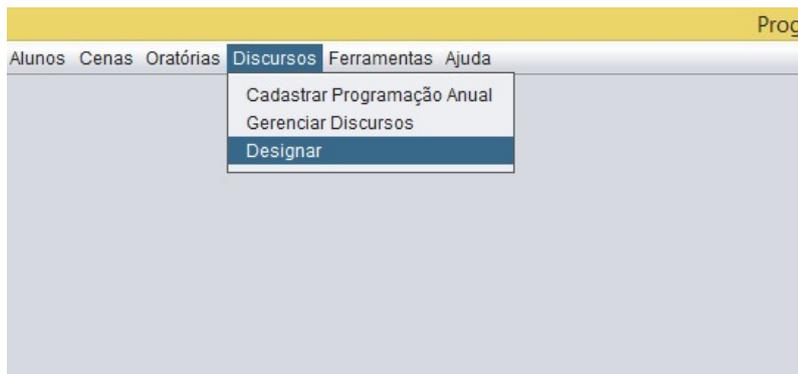


Figura 17 – Primeiro passo para Designar Discursos - selecionar a opção Designar no menu Discursos



Figura 18 – Segundo passo para Designar Discursos - escolher um período



Figura 19 – Terceiro passo para Designar Discursos - escolher oradores, cenas, oratórias e ajudantes para os discursos

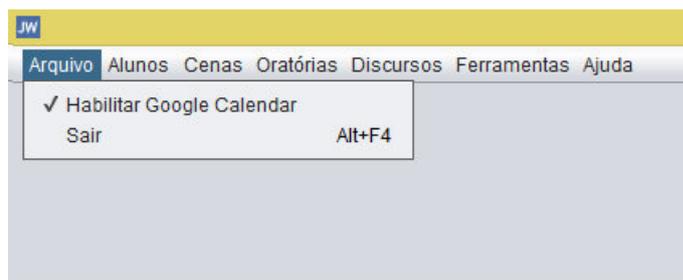


Figura 20 – Opção que permite habilitar ou não o uso da marcação de eventos na agenda do Google

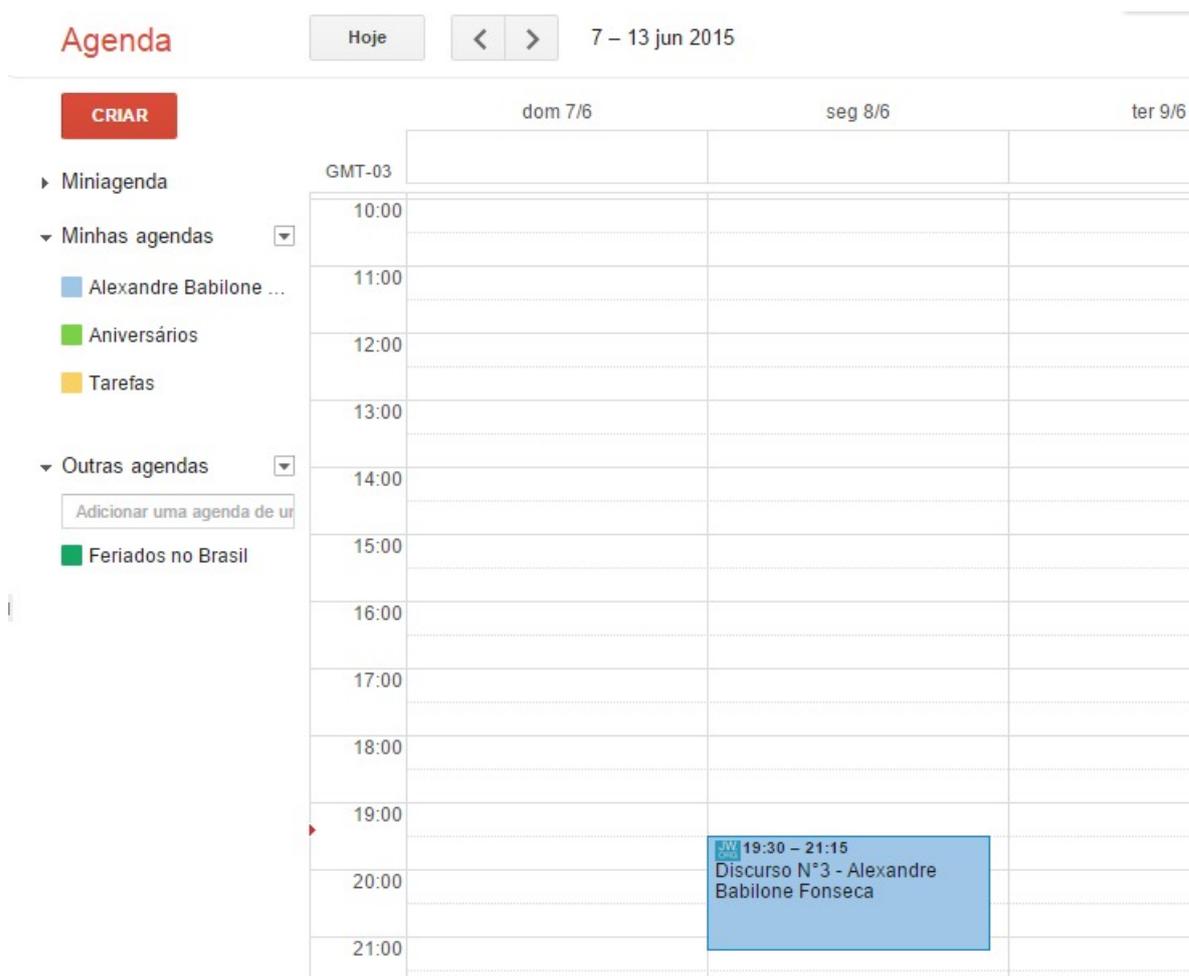


Figura 21 – Agenda pessoal de um aluno após a inserção de uma designação



Figura 22 – Detalhamento da designação na agenda do Google

Designações da Escola do Ministério Teocrático

Semana 2014-04-07	
Leitura da Bíblia	Alexandre Babilone Fonseca Fonte : Êxodo 7-10
Número 1	Walace Fonte : N.º 1: Êxodo 9:20-35 Estudo : 3 - Pronúncia correta
Número 2	Cilene Ajudante : Sara Tema: De que maneira Jesus vai voltar, e como 'todo olho o verá'? Fonte : rs p. 435 §4–p. 436 §4 Cena : 5 - Dirigindo estudo bíblico para alguém que não lê bem Estudo : 3 - Pronúncia correta
Número 3	Júlio Ajudante : Tema: Abisai — Seja leal e esteja disposto a perdoar seus irmãos Fonte : it-1 pp. 22-23 Cena : Estudo : 4 - Fluencia
Semana 2014-04-14	
Leitura da Bíblia	Nivaldo Fonte : Êxodo 11-14

Figura 23 – Trecho de um relatório feito para o dirigente da EMT. Esse relatório contém todas as características de todas as designações feitas durante o período escolhido para a geração do relatório. Para cada discurso, é mostrado quem será o orador, seu tema, fonte de matéria, ajudante, característica de oratória observada e a data.

Escola do Ministério Teocrático

Semana 2015-05-04	
Destaques da Bíblia	Wagner
Número 1	Alexandre Babilone Fonseca
Número 2	Laisa Rossim e Adriana
Número 3	Alexandre Babilone Fonseca
Semana 2015-05-11	
Destaques da Bíblia	Nivaldo
Número 1	Wagner
Número 2	Laisa Rossim e Sara
Número 3	Cilene e Laisa Rossim

Figura 24 – Trecho de um relatório feito para ser fixado no Quadro de Anúncios da Congregação. Esse relatório mostra a programação feita para as semanas do período selecionado pelo usuário, contendo o tipo de discurso e o nome de quem o proferirá.

Semana 2014-04-07
Cilene De que maneira Jesus vai voltar, e como ‘todo olho o verá’? Fonte : rs p. 435 §4–p. 436 §4 Estudo : 3 - Pronúncia correta Cena : 5 - Dirigindo estudo bíblico para alguém que não lê bem Ajudante : Sara
Semana 2014-04-07
Júlio .Abisai — Seja leal e esteja disposto a perdoar seus irmãos Fonte : it-1 pp. 22-23 Estudo : 4 - Fluencia Cena : Ajudante :

Figura 25 – Exemplo de 2 relatórios individuais, a serem entregues aos participantes que farão os discursos, contendo todas as informações que o aluno necessita para realizar o discurso.

5 Considerações Finais

Nesse capítulo, são apresentadas as conclusões obtidas durante a análise, projeto e desenvolvimento desse trabalho, bem como as perspectivas para possíveis trabalhos futuros.

5.1 Conclusões

Com a crescente busca por otimização do tempo, viu-se a oportunidade de se criar um sistema computacional para apoiar as atividades do dirigente da EMT. Mas além do ganho de tempo, viu-se que alguns processos poderiam ser aperfeiçoados. Assim, no PDEMT, aliou-se a otimização de atividades bem como a automatização delas. Relatórios que eram feitos a mão passariam a ser gerados automaticamente, prontos para impressão; atividades e informações buscadas com muito esforço, poderiam ser mostradas em segundos; por fim, a divulgação das atividades do dirigente poderia ser feita de maneira instantânea e sem nenhum esforço ou deslocamento. No Capítulo 1 foram identificados alguns objetivos para esse trabalho. A Tabela 3 apresenta a situação de cada um deles até a conclusão dessa monografia.

Objetivo	Situação	Comentários
Identificar e documentar os requisitos da ferramenta	Cumprido	Todos os requisitos - funcionais, não funcionais e regras de negócio - foram identificados e documentados no Documento de Requisitos do PDEMT.
Realizar a modelagem comportamental e estrutural da ferramenta e documentá-los	Cumprido	Foram elaborados diagramas de classes e de transição de estados, descritos no Capítulo 3 e detalhados no Documento de Especificação de Requisitos do PDEMT.
Definir a arquitetura da ferramenta e documentá-la	Cumprido	A arquitetura foi definida e foram elaborados os diagramas dos componentes que a compõem, descritos no Capítulo 4 e detalhados no Documento de Projeto do PDEMT.
Implementar a ferramenta	Cumprido Parcialmente	Por questões de tempo, um requisito não funcional não foi implementado.

Tabela 3 – Objetivos do projeto e sua situação até o término da monografia

Vale ressaltar que o sistema não ficou totalmente pronto. Por exemplo, o requisito não funcional RNF04 - Segurança de Acesso, não foi implementado devido a falta de

tempo para entrega da monografia. Além disso, o alto tempo de resposta quando algumas funções são chamadas pode incomodar os usuários finais, sendo possível aperfeiçoar esses tempos. Essas questões estão registradas e, em breve, serão corrigidas/aperfeiçoadas.

A experiência adquirida com a realização desse trabalho foi enorme. Passar por cada processo para a construção desse projeto, desde o levantamento de requisitos até a implementação e testes, contribuiu muito para o aprendizado do autor desse trabalho. Colocar em prática os conceitos aprendidos em sala de aula desde o início do curso até a conclusão, vendo como cada uma se integra com a outra, contribuiu para que se possa ter uma visão mais madura de todos os processos que envolvem a construção de um software. O uso de tecnologias conhecidas da graduação, como Java e Swing, juntamente com o uso de bibliotecas até então nunca vistas, como **iText-pdf** e **jCalendar** também contribuíram para o processo de aprendizado.

Por fim, a grande novidade foi a integração com a API Google Calendar, onde se pode fazer a ligação entre uma aplicação desktop com uma base de conhecimento que se encontra na nuvem, que possui características muito interessantes e práticas para o usuário.

Dentre as dificuldades encontradas para o desenvolvimento desse trabalho podemos destacar: estudo e entendimento da forma de autenticação do Google Calendar, escolha do padrão de implementação que melhor se adequa a sua necessidade e o curto período de tempo para o desenvolvimento do projeto e escrita da monografia. Por fim, a integração das diferentes disciplinas vistas durante o curso de Ciência da Computação foi um desafio. Com os temas de computação sendo normalmente abordados de forma distinta e isolada, este trabalho torna-se ainda mais relevante por integrar diversas áreas de modo teórico e, principalmente, prático. E com a realização deste projeto, foi possível ver e entender o funcionamento do todo o processo de construção de um software.

5.2 Trabalhos Futuros

No final do desenvolvimento de um software, tipicamente novas necessidades são identificadas. A manutenção e a evolução de software devem ser um trabalho constante, de forma que o ciclo de vida não finalize na homologação, mas permaneça ao longo de toda a vida do software.

Sendo assim, alguns trabalhos surgirão a partir deste. Como evolução do PDEMT, podemos especular a possibilidade de tornar o sistema multi-usuário, ou seja, transformar o PDEMT em um sistema Web. Com isso, dirigentes da EMT de várias congregações das Testemunhas de Jeová espalhados pelo mundo poderão ter acesso ao sistema.

Outro tipo de trabalho futuro a partir do PDEMT é a criação de outros sistemas

para apoio de outras atividades das Testemunhas de Jeová. Assim, poderia haver a integração do PDEMT com tais sistemas. Exemplos:

1. Seria interessante criar uma ferramenta para apoiar a designação de discursos públicos, integrada com o Google Maps, onde os oradores se deslocam entre as congregações para realizar discursos;
2. A criação de uma ferramenta para realizar as designações de microfone, indicador e leitor;
3. Por fim, a criação de uma ferramenta para gerenciar os territórios de pregação da congregação, integrado também com o Google Maps, para que se possa ter um melhor controle e cuidado com os territórios de pregação, que é o maior interesse das Testemunhas de Jeová.

Referências

AURUM, A.; WOHLIN, C. *Engineering and Managing Software Requirements*. [S.l.: s.n.], 2005. ISBN 978-3-540-28244-0. Citado na página 22.

BAUER, C.; KING, G. *Java Persistence with Hibernate*. [S.l.: s.n.], 2006. 904 p. ISBN 1932394885. Citado na página 25.

BOOCH, G.; JACOBSON, I.; RUMBAUGH, J. *Uml Guia do Usuario*. [S.l.: s.n.], 2006. 474 p. ISBN 8535217843. Citado na página 23.

FALBO, R. A. *Projeto de Sistemas de Software*. [s.n.], 2011. 127 p. Disponível em: <http://www.inf.ufes.br/~falbo/files/Notas_Aula_Projeto_Sistemas.pdf>. Nenhuma citação no texto.

FALBO, R. A. *Engenharia de Requisitos*. [s.n.], 2012. 179 p. Disponível em: <http://www.inf.ufes.br/~falbo/files/Notas_Aula_Engenharia_Requisitos.pdf>. Citado 3 vezes nas páginas 22, 23 e 27.

FALBO, R. A. *Engenharia de Software*. [s.n.], 2014. 144 p. Disponível em: <http://www.inf.ufes.br/~falbo/files/Notas_Aula_Engenharia_Software.pdf>. Citado 2 vezes nas páginas 21 e 28.

FOWLER, M. *Patterns of Enterprise Application Architecture*. [S.l.]: Addison Wesley, 2002. 560 p. ISBN 0321127420. Citado 3 vezes nas páginas 24, 38 e 40.

GOOGLE. *API Client Library for Java*. Google, 2015. Disponível em: <<https://developers.google.com/api-client-library/java/>>. Citado na página 35.

GOOGLE. *Google Calendar API*. [s.n.], 2015. Disponível em: <<https://developers.google.com/google-apps/calendar/>>. Citado na página 34.

JEOVá, T. de. *Beneficie-se da Escola do Ministério Teocrático*. Dezembro,2014. [s.n.], 2008. Disponível em: <<http://www.jw.org/download/?output=html&pub=be&fileformat=PDF&alllangs=0&langwritten=T&txtCMSLang=T&isBible=0>>. Citado 3 vezes nas páginas 17, 28 e 30.

JEOVá, T. de. *Nosso Ministério do Reino*. Dezembro,2014. [s.n.], 2014. Disponível em: <<http://www.jw.org/download/?issue=201412&output=html&pub=km&fileformat=PDF&alllangs=0&langwritten=T&txtCMSLang=T&isBible=0>>. Nenhuma citação no texto.

Apêndices

Documento de Requisitos

Projeto: Programa de Designações da Escola do Ministério Teocrático

Registro de Alterações:

Versão	Responsáveis	Data	Alterações
1.0	Alexandre Babilone Fonseca	11.09.2014	Criação do Documento
2.0	Alexandre Babilone Fonseca	23.10.2014	Edição do Documento
3.0	Alexandre Babilone Fonseca	07.11.2014	Edição do Documento

1. Introdução

Este documento apresenta os requisitos de usuário da ferramenta Programa de Designações da Escola do Ministério Teocrático e está organizado da seguinte forma: a seção 2 contém uma descrição do propósito do sistema; a seção 3 apresenta uma descrição do minimundo apresentando o problema; e a seção 4 apresenta a lista de requisitos de usuário levantados junto ao cliente.

2. Descrição do Propósito do Sistema

Este sistema tem como objetivo ajudar o dirigente da Escola do Ministério Teocrático a gerenciar os matriculados nessa escola, bem como tornar mais automatizada a designação de discursos aos matriculados. O sistema também oferecerá um serviço de geração de relatórios sobre as designações e participantes.

3. Descrição do Minimundo

Semanalmente, as Testemunhas de Jeová participam da Escola do Ministério Teocrático. Essa escola possui 4 discursos a serem realizados por seus membros: Destaques da Leitura Semanal da Bíblia (Destaques), Discurso Número 1, Discurso Número 2 e Discurso Número 3. O discurso do tipo Destaques da Leitura Semanal da Bíblia, é designado apenas para irmãos (homens) que possuem o cargo de Servo Ministerial ou Ancião. Quanto ao Discurso Número 1, qualquer irmão (novamente, apenas homens) pode realizá-lo; já o Discurso Número 2, apenas irmãs podem fazê-lo. Por fim, o Discurso Número 3 pode ser realizado tanto por irmãos quanto por irmãs.

Todos os 4 discursos possuem fonte de matéria. Em relação aos Discursos Número

1, 2 e 3, a pessoa que proferi-los será observada em um dos pontos de característica de oratória que se encontram no livro Beneficie-se da Escola do Ministério Teocrático, publicado pelas Testemunhas de Jeová. Além disso, as partes realizadas por uma irmã (Discurso Número 2 ou 3) terão uma ajudante (que deve ser uma irmã) e uma cena, a ser escolhida no livro Beneficie-se da Escola do Ministério Teocrático. Os discursos também possuem um tema: os Discursos Destaques e Número 1 tem como tema a própria fonte de matéria; já os Discursos Número 2 e 3 possuem temas específicos. Todas as informações referentes a programação do ano da Escola do Ministério Teocrático (data, tema, fonte dos discursos) se encontram no Nosso Ministério do Reino, também publicado pelas Testemunhas de Jeová.

A cada 2 meses acontece a Recapitulação da Escola do Ministério Teocrático. Nessa ocasião, apenas o discurso dos Destaques da Leitura Semanal da Bíblia é feito.

O dirigente da Escola do Ministério Teocrático matricula e desmatricula os membros dessa escola, além de designar irmãos e irmãs para fazerem os discursos, definindo também a característica de oratória a ser observada, cena e ajudante. O dirigente possui todas essas informações sobre discursos em uma folha.

Com antecedência, os irmãos e irmãs que farão discursos recebem um aviso com todas as informações necessárias sobre o discurso para se prepararem. Também, é fixado no Quadro de Anúncios da Congregação uma tabela com toda a programação da Escola do Ministério Teocrático.

4. Requisitos de Usuário

Tomando por base o contexto do sistema, foram identificados os seguintes requisitos de usuário:

Requisitos Funcionais

Identificador	Descrição	Prioridade	Depende de
RF01	O sistema deve permitir o cadastro de pessoas.	Alta	
RF02	O sistema deve permitir o cadastro de cenas do livro Beneficie-se da Escola do Ministério Teocrático.	Alta	
RF03	O sistema deve permitir o cadastro de oratórias presentes no livro Beneficie-se da Escola do Ministério Teocrático.	Alta	
RF04	O sistema deve permitir a inclusão/leitura/alteração/exclusão de uma pessoa, cena, ajudante e característica de oratória para um discurso em que essas características se fazem necessárias.	Alta	RF01,RF02,RF03,RF12
RF05	O sistema deve permitir a consulta de frequência de participação dos oradores cadastrados.	Média	RF01,RF04,RF12
RF06	O sistema deve permitir a consulta de frequência de uso das oratórias.	Média	RF03,RF04,RF12
RF07	O sistema deve permitir a consulta de frequência de uso das cenas.	Média	RF02,RF04,RF12
RF08	O sistema deve permitir a consulta de frequência de ajudantes.	Média	RF01,RF04,RF12
RF09	O sistema deve permitir a geração do Relatório do dirigente da Escola (referente a um período de tempo), no qual constará todas as características atribuídas por ele a cada um dos discursos.	Média	RF01,RF02,RF03,RF04,RF12
RF10	O sistema deve permitir a geração do Relatório do Quadro de Anúncios da congregação, contendo toda a programação de discursos e suas características, referente a um período de tempo.	Média	RF01,RF02,RF03,RF04,RF12
RF11	O sistema deve permitir a geração do Relatório Individual dos discursos.	Média	RF01,RF02,RF03,RF04,RF12
RF12	O sistema deve permitir o cadastro da programação anual da Escola do Ministério Teocrático que se encontra no Nosso Ministério do Reino.	Alta	

Regras de Negócio

Identificador	Descrição	Prioridade	Depende de
RN01	Para os discursos Destaques da Leitura Semanal da Bíblia e Discurso Número 1, apenas homens podem ser alocados para realizá-las.	Alta	
RN02	O Discurso Número 2 só pode ser alocado para uma mulher.	Alta	
RN03	O Discurso Número 3 pode ser alocado tanto para uma mulher quanto para um homem.	Alta	
RN04	Ajudantes só podem ser mulheres.	Alta	
RN05	Os discursos feitos por mulheres devem ter uma ajudante e uma cena.	Alta	
RN06	Os discursos feitos por homens não devem ter cena ou ajudante.	Alta	
RN07	Os matriculados que proferirem os Discursos Número 1, 2 e 3 devem ter uma característica de oratória associada a eles.	Alta	
RN08	Nas semanas da Recapitulação Escrita, apenas o discurso Destaques da Leitura Semanal da Bíblia é feito.	Alta	
RN09	Os Destaques da Leitura Semanal da Bíblia só podem ser feitos por irmãos que possuem o cargo de Servo Ministerial ou Ancião.	Alta	

Requisitos Não Funcionais

Identificador	Descrição	Categoria	Escopo	Prioridade	Depende de
RNF01	A ferramenta deve ser de aprendizado fácil, não sendo necessário nenhum treinamento especial para seu uso.	Facilidade de Aprendizado	Sistema	Média	
RNF02	A ferramenta deve ser de fácil operação, não sendo necessário uso contínuo para uma boa operação do sistema.	Facilidade de Operação	Sistema	Média	
RNF03	O sistema deve ser fácil de manter, de modo a acomodar novas funcionalidades.	Manutenibilidade	Sistema	Alta	
RNF04	O sistema deve garantir que os dados estão protegidos de acessos não autorizados.	Segurança de Acesso	Sistema	Alta	
RNF05	O sistema deve permitir a geração de relatórios em PDF.	Tecnologia	Funcionalidade	Alta	
RNF06	O sistema deve funcionar no sistema operacional Windows.	Tecnologia	Sistema	Alta	

Documento de Especificação de Requisitos

Projeto: Programa de Designações da Escola do Ministério Teocrático

Registro de Alterações:

Versão	Responsáveis	Data	Alterações
1.0	Alexandre Babilone Fonseca	29/10/2014	Emissão do Documento
1.1	Alexandre Babilone Fonseca	15/11/2014	Edição do Documento
1.2	Alexandre Babilone Fonseca	27/11/2014	Edição do Documento
2.0	Alexandre Babilone Fonseca	04/12/2014	Edição do Documento
3.0	Alexandre Babilone Fonseca	13/12/2014	Edição do Documento
3.1	Alexandre Babilone Fonseca	15/04/2015	Edição do Documento

1. Introdução

Este documento apresenta a especificação dos requisitos da ferramenta **Programa de Designações da Escola do Ministério Teocrático**. A atividade de análise de requisitos foi conduzida aplicando-se técnicas de modelagem de casos de uso, modelagem de classes e modelagem de comportamento dinâmico do sistema. Os modelos apresentados foram elaborados usando a UML. Este documento está organizado da seguinte forma: a seção 2 apresenta os subsistemas identificados, mostrando suas dependências na forma de um diagrama de pacotes; a seção 3 apresenta o modelo de casos de uso, incluindo descrições de atores, os diagramas de casos de uso e descrições de casos de uso; a seção 4 apresenta o modelo conceitual estrutural do sistema, na forma de diagramas de classes; a seção 5 apresenta o modelo comportamental dinâmico do sistema, na forma de diagramas de estado; finalmente, a seção 6 apresenta o glossário do projeto, contendo as definições das classes identificadas.

2. Identificação de Subsistemas

Devido à complexidade do Sistema, não houve necessidade de dividi-lo em subsistemas.

3. Modelo de Casos de Uso

O modelo de casos de uso visa capturar e descrever as funcionalidades que um sistema deve prover para os atores que interagem com o mesmo. Os atores identificados no contexto deste projeto estão descritos na tabela abaixo.

Ator	Descrição
Dirigente da Escola do Ministério Teocrático	Executa as atividades de cadastro de alunos, designação de discursos e consulta relatórios.
Google Calendar	Cria as designações na agenda do Google.

Tabela 1: Atores

A seguir, são apresentados os diagramas de casos de uso e descrições associadas.

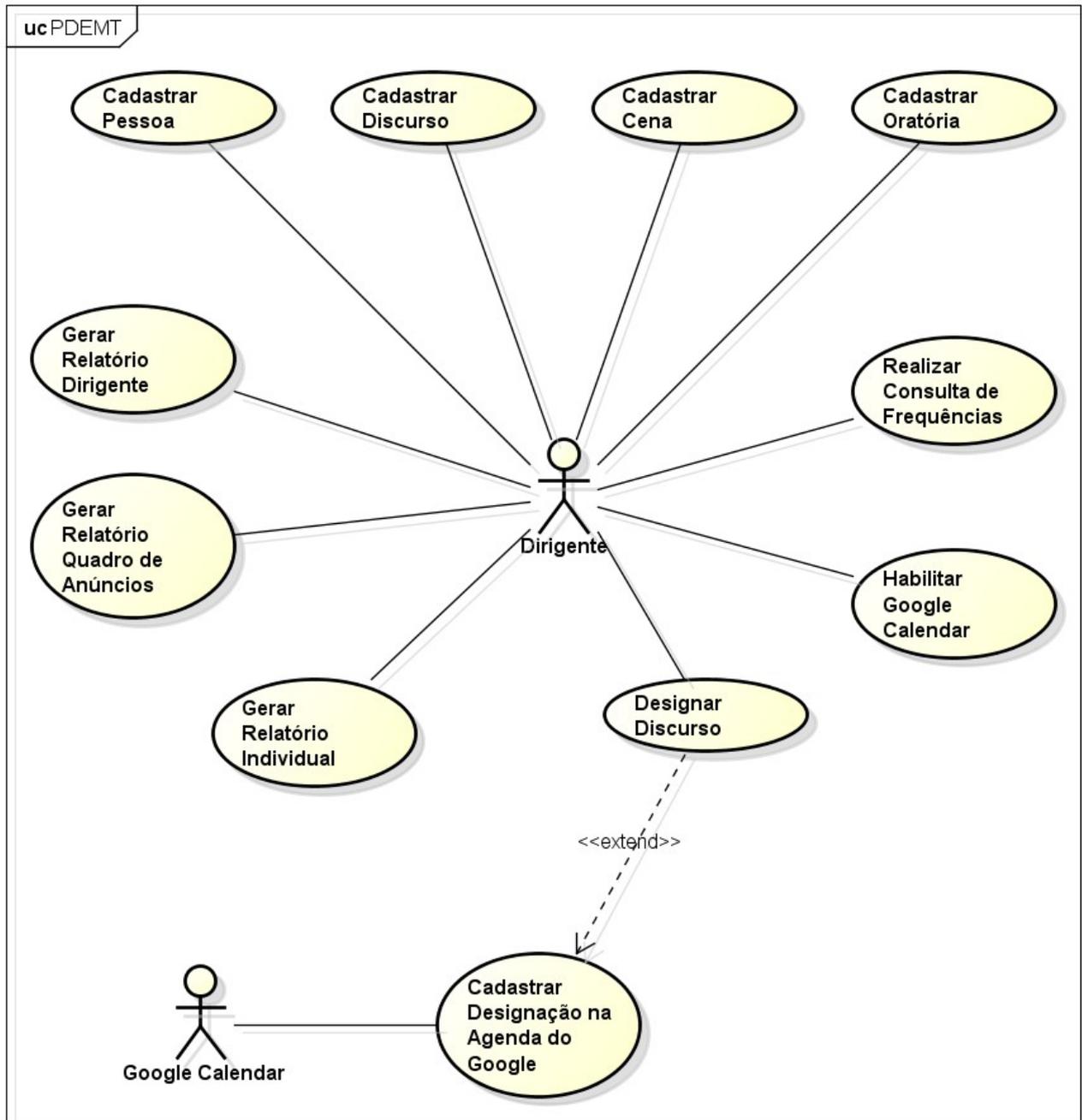


Figura 1: Diagrama de Casos de Uso

A seguir, são apresentadas as descrições de cada um dos casos de uso identificados. Os casos de uso cadastrais de baixa complexidade, envolvendo inclusão, alteração, consulta e exclusão, são descritos na tabela abaixo, segundo o padrão da organização.

Sistema	Programa de Designações da Escola do Ministério Teocrático				
Identificador	Caso de Uso	Ações Possíveis	Observações	Requisitos	Classes
UC01	Cadastrar Discurso	I, A, C, E	[I]: Informar: tema, fonte e data [E]: Discurso já designado pra não pode ser excluído.	RF12	Discurso
UC02	Cadastrar Pessoa	I, A, C, E	[I]: Informar: nome, sexo e se tem o privilégio de servo ou ancião. [E]: Pessoa já designada para algum discurso não pode ser excluída.	RF01	Pessoa
UC03	Cadastrar Cena	I, A, C, E	[I]: Informar: nome da cena [E]: Cena já escolhida para algum discurso não pode ser excluída.	RF02	Cena
UC04	Cadastrar Oratória	I, A, C, E	[I]: Informar: nome da característica de oratória. [E]: Característica de oratória já escolhida para algum discurso não pode ser excluída.	RF03	Oratória

Tabela 2: Casos de Uso Cadastrais

Os casos de uso de consulta mais abrangente que as consultas a um único objeto (já tratadas como parte dos casos de uso cadastrais), mas ainda de baixa complexidade, tais como consultas que combinam informações de vários objetos envolvendo filtros, estão descritos na tabela abaixo, segundo o padrão da organização.

Sistema	Programa de Designações da Escola do Ministério Teocrático			
Identificador	Caso de Uso	Observações	Requisitos	Classes
UC05	Realizar Consulta de Freqüências	O sistema mostra as frequências de uso de todas as Cenas, Oratórias, Ajudantes e Oradores ordenadas em ordem decrescente.	RF01, RF02, RF03, RF12	Pessoa, Discurso, Cena, Oratória
UC06	Gerar Relatório Individual	O relatório é gerado selecionando-se um período de tempo. É gerado um relatório no formato PDF para ser entregue individualmente aos participantes, com o tema do discurso, o tipo do discurso, a data da semana em que será proferido, a cena, a característica de oratória e o nome de quem proferirá o discurso.	RF01, RF02, RF03, RF12	Oratória, Cena, Discurso, Pessoa
UC07	Gerar Relatório Dirigente	O relatório é gerado selecionando-se um período de tempo. É gerado um relatório no formato PDF para o dirigente da Escola, com toda a programação da Escola, incluindo o tema do discurso, o tipo do discurso, a data da semana em que será proferido, a cena, a característica de oratória e o nome de quem proferirá o discurso, em ordem cronológica, abrangendo o período selecionado pelo usuário.	RF01, RF02, RF03, RF12	Oratória, Cena, Discurso, Pessoa
UC08	Gerar Relatório Quadro de Anúncios	O relatório é gerado selecionando-se um período de tempo no futuro. É gerado um relatório no formato PDF contendo o tema do discurso, o tipo do discurso, a data da semana em que será proferido e o nome de quem irá proferi-lo, abrangendo o período selecionado pelo usuário.	RF01, RF12	Discurso, Pessoa
UC9	Habilitar Google Calendar	Habilita a utilização ou não do Google Calendar para o agendamento de discursos na Agenda do Google.		

Tabela 3: Casos de Uso de Consulta

A seguir, são apresentados os casos de uso de maior complexidade que não puderam ser descritos segundo os formatos tabulares simplificados. Esses casos de uso são descritos segundo o padrão de descrição completa de casos de uso definido.

Descrição de Caso de Uso

Projeto: Programa de Designações da Escola do Ministério Teocrático

Identificador do Caso de Uso: UC10

Caso de Uso: Designar Discurso

Descrição Sucinta: Este caso de uso permite designar um discurso a ser proferido.

Fluxos de Eventos Normais

Nome do Fluxo de Eventos Normal	Precondição	Descrição
<i>Designar Discurso</i>	Haver pessoas, discursos, cenas e oratórias cadastradas.	<ol style="list-style-type: none">1 O dirigente escolhe o período desejado para as designações, em que o período mínimo é uma semana.2 O sistema registra a escolha e exibe a lista de semanas referentes ao período selecionado no passo anterior, com seus discursos.3 Para cada semana seleccione<ol style="list-style-type: none">3.1 para o Destaque<ol style="list-style-type: none">3.1.1 um orador3.2 para o Discurso 1<ol style="list-style-type: none">3.2.1 um orador3.2.2 uma oratória3.3 para o Discurso 2<ol style="list-style-type: none">3.3.1 uma oradora3.3.2 uma oratória3.3.3 uma cena3.3.4 uma ajudante3.4 para o Discurso 3<ol style="list-style-type: none">3.4.1 uma oradora3.4.2 uma oratória3.4.3 uma cena3.4.4 uma ajudante¹.4 Ao terminar a designação de cada semana, o dirigente deve salvar a designação para que o sistema registre as inclusões/alterações.5 O sistema registra as escolhas de cada semana.

Fluxos de Eventos Variantes

Nome do Fluxo de Eventos Normal Relacionado	Variante	Descrição
<i>Designar Discurso</i>	3.3.3 – Não informar cena	3.3.3b. A designação é feita sem uma cena.
<i>Designar Discurso</i>	3.4.3 – Não informar cena	3.4.3b. A designação é feita sem uma cena.

¹ Quando o Discurso 3 é feito por uma oradora, o dirigente deve ainda escolher uma ajudante e pode optar por escolher uma cena ou não. Quando a parte é feita por um irmão (homem), não é necessário nem ajudante nem cena.

Fluxos de Eventos de Exceção

Nome do Fluxo de Eventos Normal Relacionado	Condição de Exceção	Descrição
<i>Designar Discurso</i>	3.1.1 - Nenhum orador foi selecionado	3.1.1a. Uma mensagem é exibida informando que é necessário escolher um(a) orador(a) para proferir o discurso.
<i>Designar Discurso</i>	3.2.1 - Nenhum orador foi selecionado	3.2.1a. Uma mensagem é exibida informando que é necessário escolher um(a) orador(a) para proferir o discurso.
<i>Designar Discurso</i>	3.3.1 - Nenhum oradora foi selecionada	3.3.1a. Uma mensagem é exibida informando que é necessário escolher um(a) orador(a) para proferir o discurso.
<i>Designar Discurso</i>	3.4.1 - Nenhum orador(a) foi selecionado(a)	3.4.1a. Uma mensagem é exibida informando que é necessário escolher um(a) orador(a) para proferir o discurso.
<i>Designar Discurso</i>	3.2.2 - Nenhuma oratória foi selecionada	3.2.2a. Uma mensagem é exibida informando que é necessário escolher uma oratória para o discurso.
<i>Designar Discurso</i>	3.3.2 - Nenhuma oratória foi selecionada	3.3.2. Uma mensagem é exibida informando que é necessário escolher uma oratória para o discurso.
<i>Designar Discurso</i>	3.4.2 - Nenhuma oratória foi selecionada	3.4.2a. Uma mensagem é exibida informando que é necessário escolher uma oratória para o discurso.
<i>Designar Discurso</i>	3.3.4 - Nenhuma ajudante foi selecionada	3.3.4a. Uma mensagem é exibida informando que é necessário escolher uma ajudante para o discurso.
<i>Designar Discurso</i>	3.4.4 - Nenhuma ajudante foi selecionada	3.4.4a. Uma mensagem é exibida informando que é necessário escolher uma ajudante para o discurso.

Requisitos Relacionados: RF01, RF02, RF03, RF12

Classes Relacionadas: Discurso, Pessoa, Cena, Oratória

Projeto: Programa de Designações da Escola do Ministério Teocrático

Identificador do Caso de Uso: UC11

Caso de Uso: Cadastrar Designação na Agenda do Google

Descrição Sucinta: Este caso de uso permite a criação de um evento (a saber, o discurso designado) na Agenda do Google.

Fluxos de Eventos Normais

Nome do Fluxo de Eventos Normal	Precondição	Descrição
<i>Cadastrar Designação na Agenda do Google</i>	Haver pessoas, discursos, cenas e oratórias cadastradas; o discurso estar designado	Assim que o usuário confirma a designação de uma semana de discursos, o sistema faz requisições ao Google Calendar para que eventos possam ser criados tendo essas designações como base.

Requisitos Relacionados: RF01, RF02, RF03, RF12

Classes Relacionadas: Discurso, Pessoa, Cena, Oratória

4. Modelo Estrutural

O modelo conceitual estrutural visa capturar e descrever as informações (classes, associações e atributos) que o sistema deve representar para prover as funcionalidades descritas na seção anterior. A seguir, é apresentado o diagrama de classes do sistema. Na seção 6 – Glossário de Projeto – são apresentadas as descrições das classes presentes nos diagramas apresentados nesta seção.

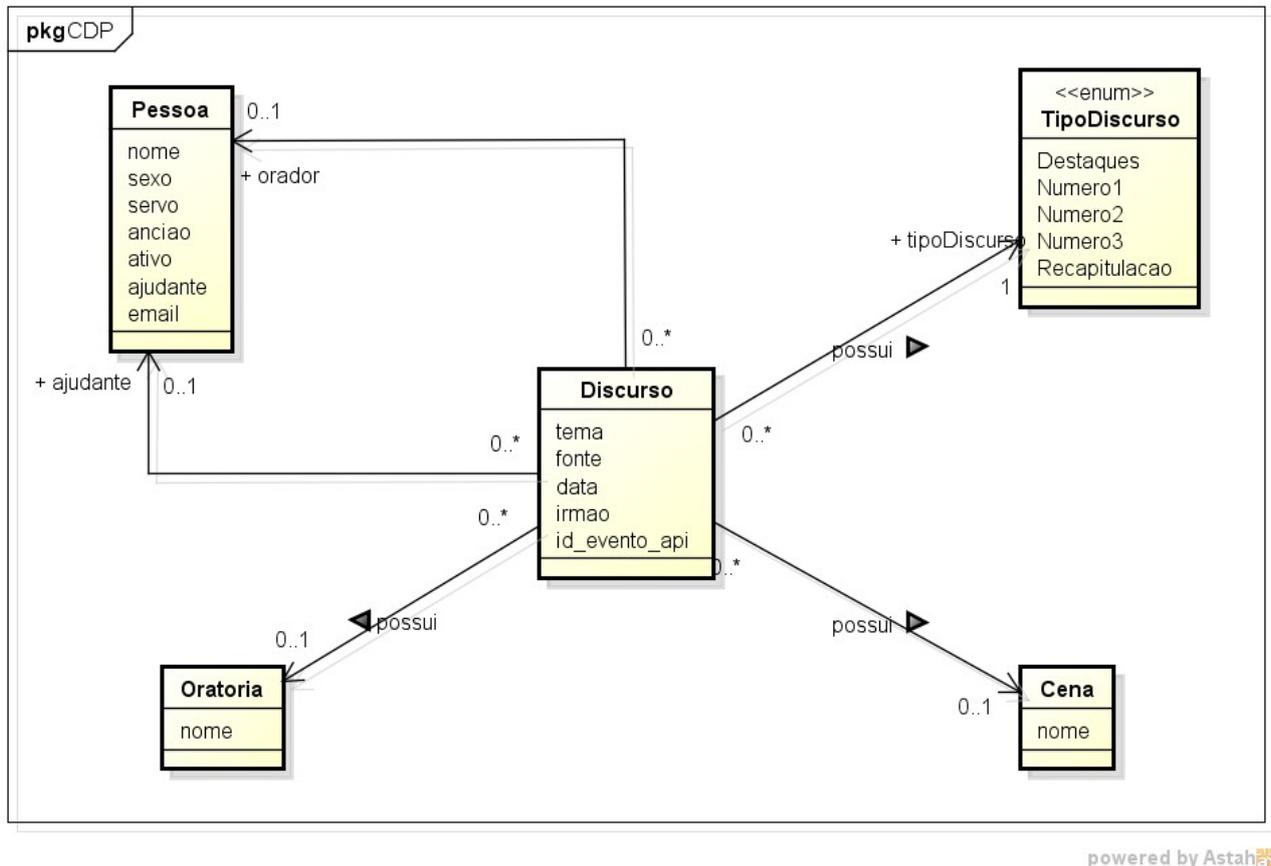


Figura 4 – Diagrama de Classes do Sistema.

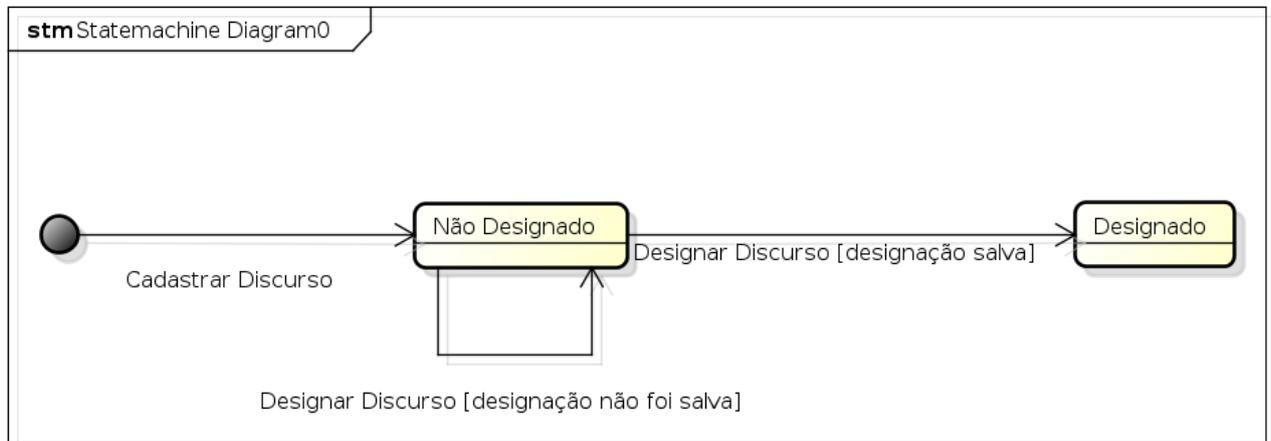
As seguintes restrições de integridade devem ser observadas:

- RI1: Somente pessoas do sexo feminino podem ter o papel de ajudante.
- RI2: Somente pessoas do sexo masculino podem proferir os discursos dos tipos: Destaques e Número 1.
- RI3: Somente pessoas do sexo feminino podem proferir os discursos do tipo Número 2.
- RI4: Somente servos e anciãos podem fazer o discurso do tipo Destaques.
- RI5: Discursos feitos por pessoas do sexo masculino não possuem ajudante e cena.
- RI6: Discursos do tipo Destaques não possui oratória.

5. Modelo Dinâmico

O modelo dinâmico visa capturar o comportamento dinâmico do sistema. A seguir, são apresentados os diagramas de estados elaborados no contexto deste projeto.

A Figura 5 apresenta o diagrama de estados da classe Discurso.



powered by Astah

Figura 5 – Diagrama de Estados da Classe Discurso.

6. Glossário de Projeto

Esta seção apresenta as definições dos principais conceitos envolvidos no projeto.

Pessoa			
Propriedade	Obrigatoriedade	Tipo	Descrição
nome	Obrigatório	Texto	Nome completo
Sexo	Obrigatório	Texto	Masculino ou Feminino
Servo	Obrigatório	Booleano	Indica se a pessoa possui cargo de Servo
Ancião	Obrigatório	Booleano	Indica se a pessoa possui cargo de Ancião
Ativo	Obrigatório	Booleano	Indica se a pessoa está matriculada na Escola ou não
Ajudante	Obrigatório	Booleano	Indica se a pessoa é ajudante ou não

Cena			
Propriedade	Obrigatoriedade	Tipo	Descrição
Cena	Obrigatório	Texto	Nome da cena

Oratória			
Propriedade	Obrigatoriedade	Tipo	Descrição
Oratória	Obrigatório	Texto	Nome da característica de oratória

Discurso			
Propriedade	Obrigatoriedade	Tipo	Descrição
Tema	Obrigatório	Texto	Nome do tema
Fonte	Não Obrigatório	Texto	Nome da fonte de matéria
Data	Obrigatório	Data	Data em que o discurso será/foi proferido
irmao	Obrigatório	Booleano	Indica se o orador é homem(irmão) ou mulher(irmã)
id_evento_api	Não Obrigatório	Texto	Faz referência a um identificador de evento do Google Calendar

TipoDiscurso			
Propriedade	Obrigatoriedade	Tipo	Descrição
Nome	Obrigatório	Texto	Nome do tipo de discurso

Documento de Projeto de Sistema

Projeto: Programa de Designações da Escola do Ministério Teocrático (PDEMT)

Registro de Alterações:

Versão	Responsáveis	Data	Alterações
1.0	Alexandre Babilone Fonseca	15/04/2015	Criação
2.0	Alexandre Babilone Fonseca	23/04/2015	Edição

1. Introdução

Este documento apresenta o documento de projeto (*design*) da ferramenta PDEMT. Este documento está organizado da seguinte forma: a seção 2 apresenta a plataforma de software utilizada na implementação da ferramenta; a seção 3 trata de táticas utilizadas para tratar requisitos não funcionais (atributos de qualidade); a seção 4 apresenta o projeto da arquitetura de software; por fim, a seção 5 apresenta o projeto dos componentes da arquitetura.

2. Plataforma de Desenvolvimento

Na Tabela 1: Plataforma de Desenvolvimento e Tecnologias Utilizadas são listadas as tecnologias utilizadas no desenvolvimento da ferramenta, bem como o propósito de sua utilização.

Tecnologia	Versão	Descrição	Propósito
Java	7	Linguagem de programação orientada a objetos e independente de plataforma.	Desenvolvimento de aplicativos em linguagem de programação orientada a objetos e independente de plataforma.
SWING		API de interface gráfica para uso com a linguagem de programação Java.	Construir interfaces gráficas para interação do usuário com o sistema.
JDBC		API escrita em Java que faz o envio de instruções SQL para qualquer banco de dados relacional.	Fazer acessos, incluindo leitura e escrita, ao banco de dados para ter informações persistidas.
PostgreSQL	8.4	Sistema Gerenciador de Banco de Dados Relacional gratuito.	Persistência dos dados manipulados pela ferramenta.
Itext-pdf	5.5.0	Biblioteca para criação de documentos em PDF.	Criação de relatórios em PDF.
jCalendar	1.4	Biblioteca de componentes gráficos para manipulação de datas.	Usar seus componentes para fazer a interação com o usuário.
Google Calendar API	3	API de integração com o calendário do Google	Registrar os eventos do sistema.
Google API Client Library for Java		API responsável pela autenticação em contas do Google	Autenticar o usuário do sistema em uma conta do Google

Tabela 1: Plataforma de Desenvolvimento e Tecnologias Utilizadas

Na Tabela 2: Softwares de Apoio ao Desenvolvimento do Projeto vemos os softwares que apoiaram o desenvolvimento de documentos e também do código fonte.

Tecnologia	Versão	Descrição	Propósito
NetBeans IDE	8.0.2	Ambiente de desenvolvimento (IDE) para a linguagem Java.	Facilitar a atividade de implementação de software.
Astah Community	6.9.0	Ferramenta para modelagem em UML.	Ferramenta de modelagem UML (<i>Unified Modeling Language</i>).
LibreOffice	4.3	<i>Suite</i> gratuita de aplicativos para escritório.	Apoiar a elaboração da documentação.

Tabela 2: Softwares de Apoio ao Desenvolvimento do Projeto

3. Atributos de Qualidade e Táticas

Na Tabela 3: Atributos de Qualidade e Táticas Utilizadas são listados os atributos de qualidade considerados neste projeto, com uma indicação se os mesmos são condutores da arquitetura ou não e as táticas a serem utilizadas para tratá-los.

Categoria	Requisitos Não Funcionais Considerados	Condutor da Arquitetura	Tática
Facilidade de Aprendizado, Facilidade de Operação	RNF01, RNF02	Sim	<ul style="list-style-type: none">• Prover ao usuário a capacidade de entrar com comandos que permitam operar o sistema de modo mais amigáveis. Para tal, as interfaces do sistema devem permitir, sempre que possível, a entrada por meio de seleção ao invés da digitação de campos.
Segurança de Acesso	RNF04	Sim	<ul style="list-style-type: none">• Identificar usuários usando <i>login</i> e autenticá-los por meio de senha.
Manutenibilidade	RNF03	Sim	<ul style="list-style-type: none">• Organizar a arquitetura da ferramenta segundo uma combinação de camadas e partições.• A camada de lógica de negócio deve ser organizada segundo o padrão Camada de Serviço.• A camada de gerência de dados deve ser organizada segundo o padrão DAO.• Separar a interface com o usuário do restante da aplicação, segundo o padrão MVC.
Tecnologia	RNF05	Sim	<ul style="list-style-type: none">• Utilização da biblioteca iText
Tecnologia	RNF06	Sim	<ul style="list-style-type: none">• Desenvolvimento do sistema na linguagem Java

Tabela 3: Atributos de Qualidade e Táticas Utilizadas

4. Arquitetura de Software

A arquitetura de software da ferramenta PDEMT esta dividida em camadas, conforme a Figura 1: Arquitetura do Sistema abaixo. Devido à complexidade do sistema, não foi necessário dividi-lo em subsistemas.

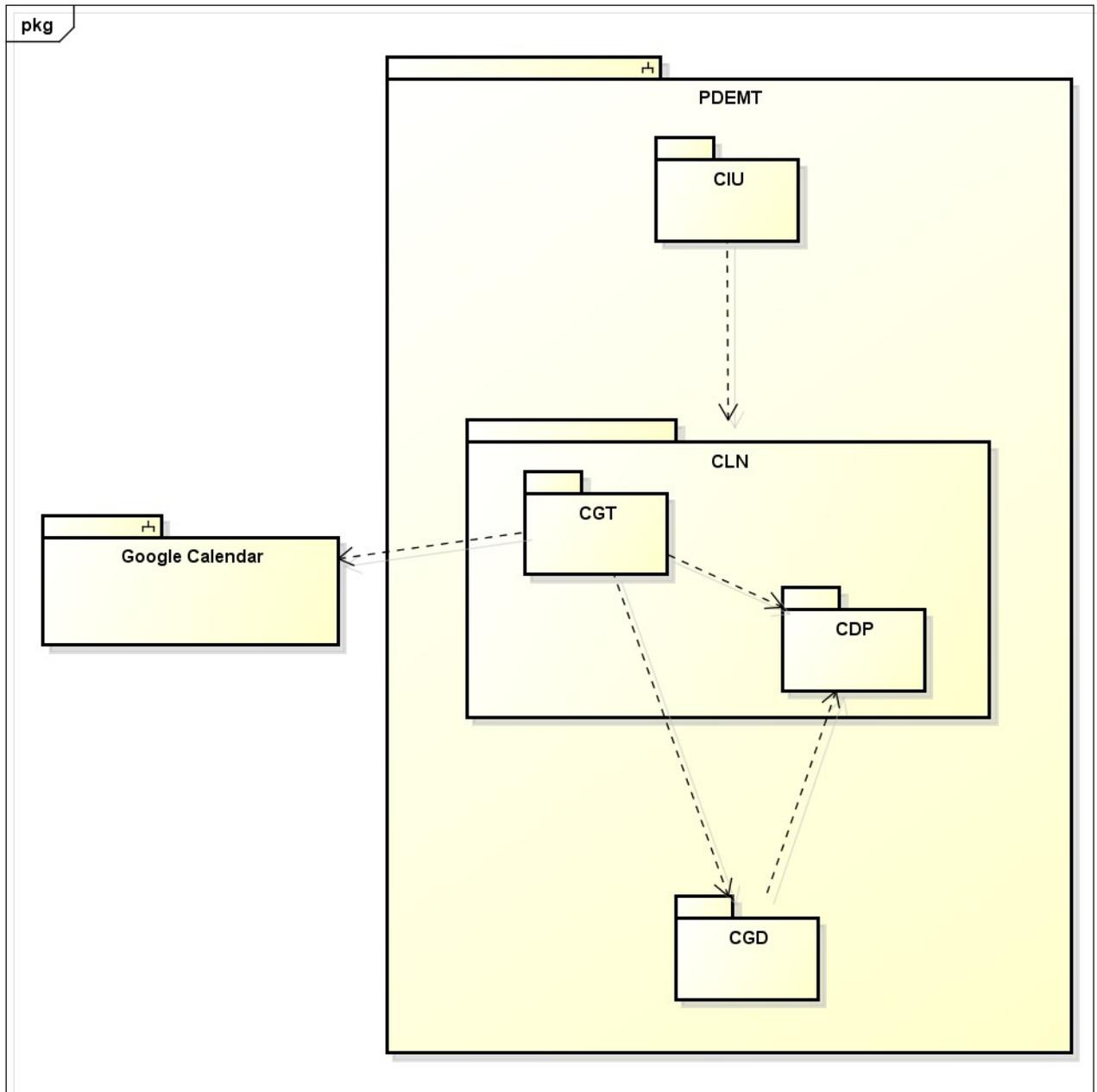


Figura 1: Arquitetura do Sistema

A arquitetura é composta por 3 camadas, a saber: camadas de Interface com o Usuário (CIU), que trata de aspectos relacionados às interfaces gráficas com os usuários; Lógica de Negócio (CLN), onde é implementada a lógica de negócio; e Gerência de Dados (CGD), responsável pela persistência de objetos. A camada CLN é dividida em duas outras: Camada do Domínio do Problema (CDP) e Camada de Gerência de Tarefas (CGT).

Em seguida, o projeto de cada uma dessas partições é apresentado.

5. Detalhamento dos Componentes da Arquitetura

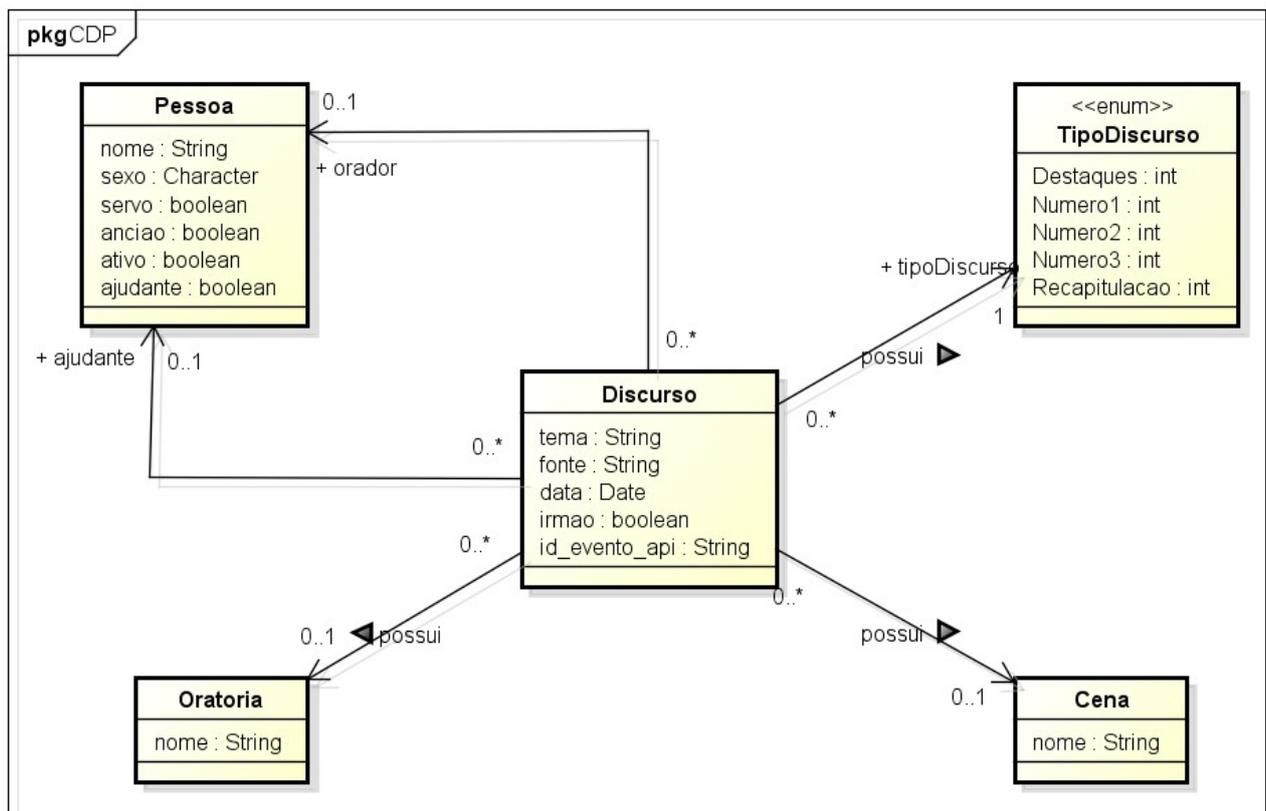
Conforme discutido anteriormente, PDEMT está organizado em três camadas: Camada de Lógica de Negócio, Camada de Interface com o Usuário e Camada de Gerência de Dados. A seguir o projeto dessas camadas é apresentado.

5.1 – Camada de Lógica de Negócio

Para organizar a camada de lógica de negócio deste subsistema, foi escolhido o padrão Camada de Serviço. Sendo assim, essa camada é dividida em dois componentes: Componente de Domínio do Problema (CDP) e Componente de Gerência de Tarefas (CGT), como apresentado anteriormente na Figura 1. Esse padrão utiliza um componente para tratar a lógica de aplicação (o CGT), o qual recebe as requisições da interface, e um componente para tratar os conceitos do domínio do problema, advindos do modelo conceitual estrutural elaborado na fase de análise (o CDP). A seguir, o projeto desses dois componentes é apresentado.

5.1.1 – Componente de Domínio do Problema (CDP)

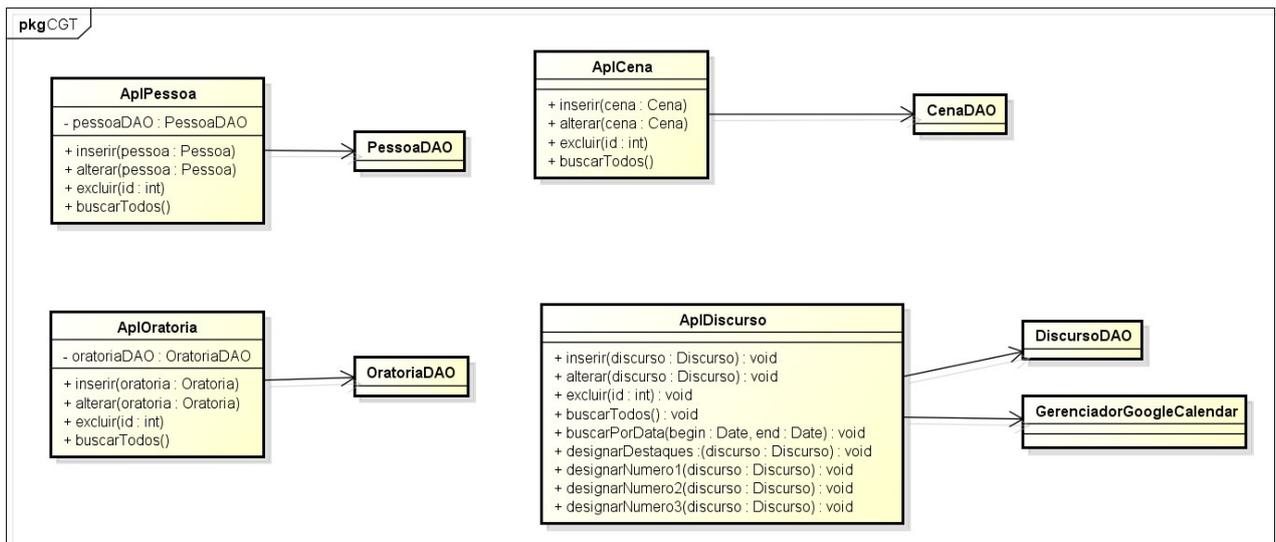
A Figura 2: Diagrama de Classe do CDP apresenta o diagrama de classes do CDP.



powered by Astah

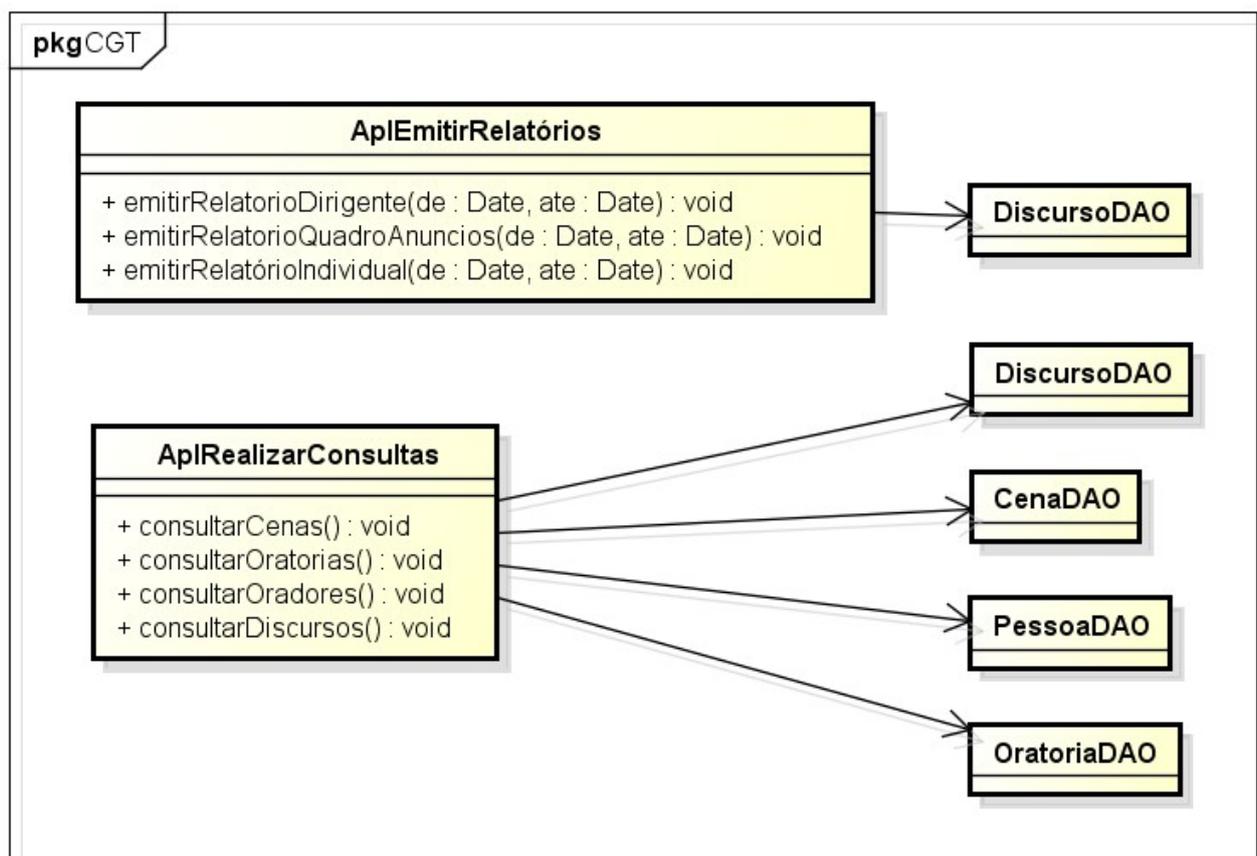
Figura 2: Diagrama de Classe do CDP

5.1.2 – Componente de Gerência de Tarefas (CGT)



powered by Astah

Figura 3: CGT dos casos de uso mais básicos e de designação de discursos



powered by Astah

Figura 4: CGT dos casos de uso referentes a relatórios e consultas

No projeto do CGT, optou-se por fazer as classes de aplicação seguindo fielmente o modelo de casos de uso. As Figura 3: CGT dos casos de uso mais básicos e de designação de discursos e

Figura 4: CGT dos casos de uso referentes a relatórios e consultas mostram as classes de aplicação do CGT.

A Tabela 4: Relação entre Classes do CGT e Casos de Uso. mostra as relações existentes entre as classes do CGT e os casos de uso por elas tratados. O caso de uso **Designar Discurso** é formado pelos métodos **designarDestques**, **designarNumero1**, **designarNumero2** e **designarNumero3**.

Classe	Casos de Uso
AplDiscurso	Cadastrar Discurso, Designar Discurso, Habilitar Google Calendar, Criar Designação na Agenda do Google
AplCena	Cadastrar Cena
AplOratoria	Cadastrar Oratoria
AplPessoa	Cadastrar Pessoa
AplEmitirRelatorios	Gerar Relatório Dirigente, Gerar Relatório Quadro de Anúncios, Gerar Relatório Individual
AplRealizarConsultas	Realizar Consulta de Frequência

Tabela 4: Relação entre Classes do CGT e Casos de Uso.

5.2 – Camada de Interface com o Usuário

Para organizar a camada de interface com o usuário, não se achou necessário adotar um padrão diferente do Service Layer já utilizado. Assim, as classes de aplicação são referenciadas diretamente pelas classes de visão. Seguindo o padrão escolhido, a classe de aplicação não tem ligação nenhuma com a classe de visão, sendo totalmente independente. Veja a Figura 5: Relação entre as classes de visão e aplicação.

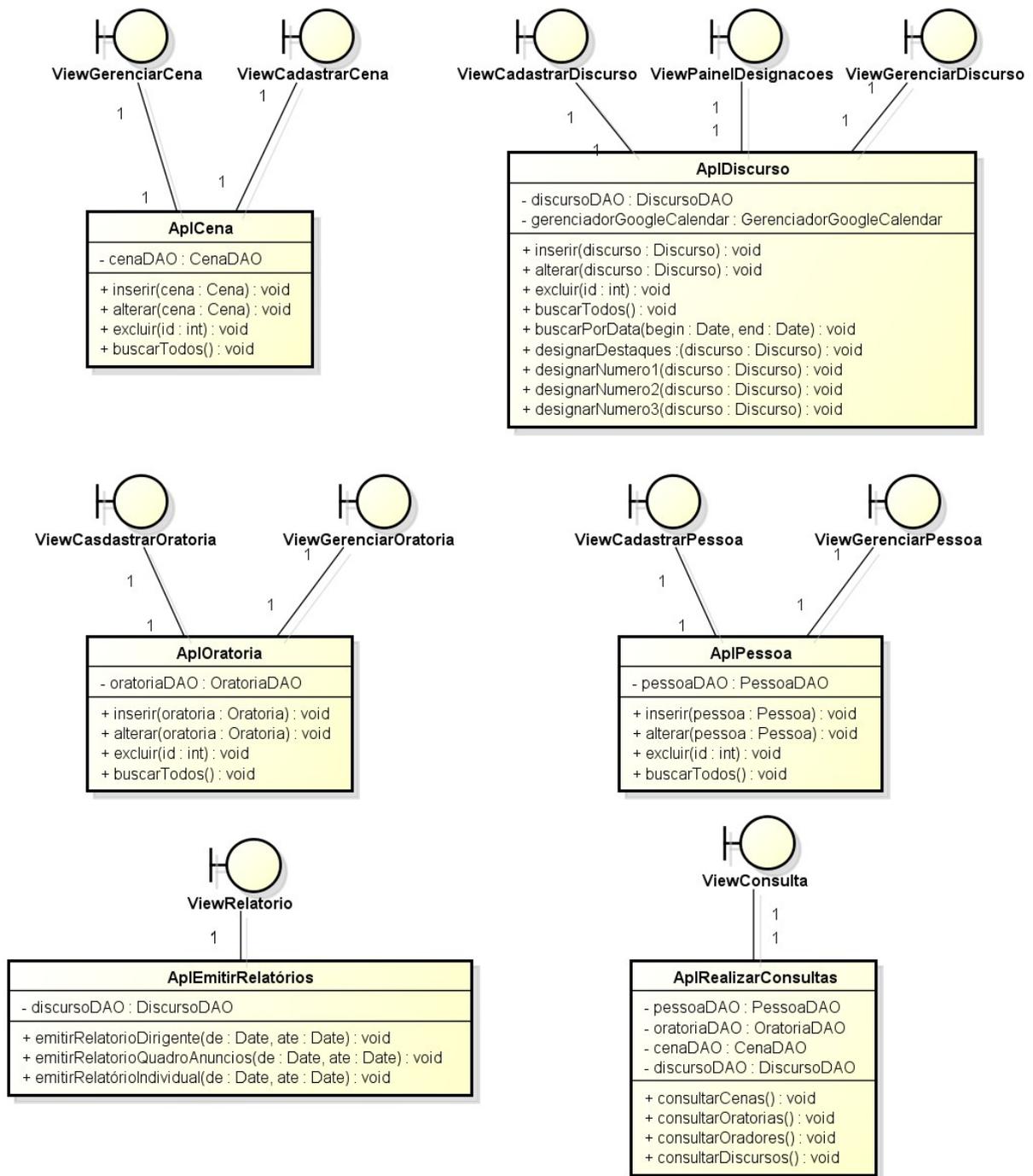


Figura 5: Relação entre as classes de visão e aplicação

Segue abaixo algumas das principais telas do sistema.



Figura 6: Tela principal do sistema

A screenshot of a dialog box titled 'Matricular Aluno'. The dialog box has a yellow title bar with the text 'Matricular Aluno' and standard window control buttons (minimize, maximize, close). The main area is light gray and contains the following fields and controls:

- 'Nome :' followed by a text input field.
- 'E-mail :' followed by a text input field.
- 'Sexo :' followed by a dropdown menu showing 'Masculino'.
- Four checkboxes on the right side:
 - Servo
 - Ancião
 - Ajudante
 - Ativo
- At the bottom, two buttons: 'Cancelar' and 'Salvar'.

Figura 7: Tela de matrícula de alunos

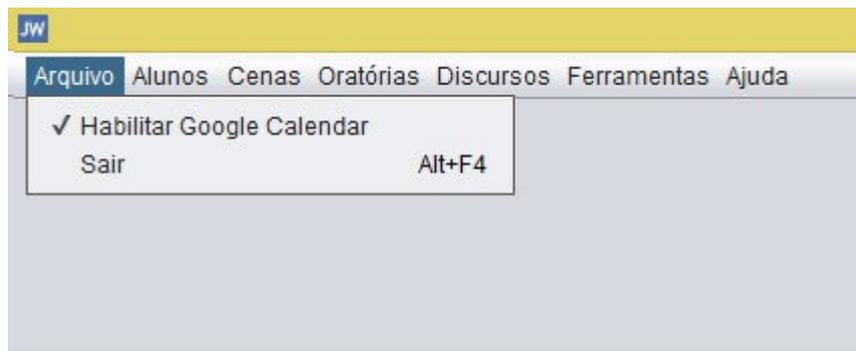


Figura 8: Item de menu para habilitar o uso do Google Calendar

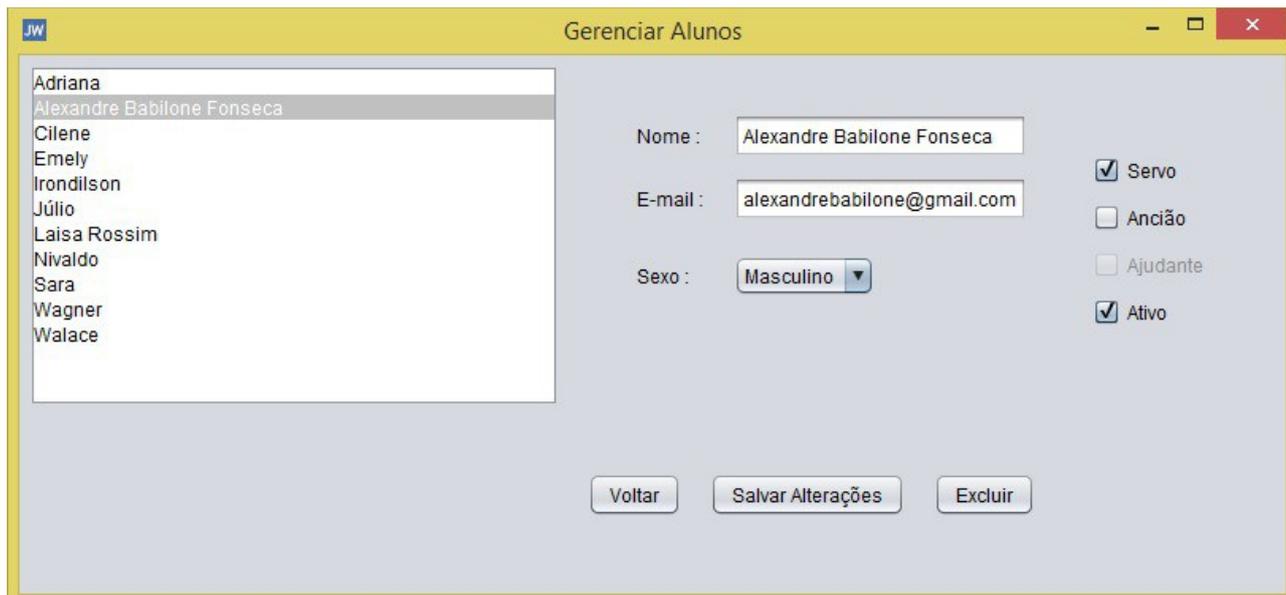


Figura 9: Tela de Gerenciamento de Alunos

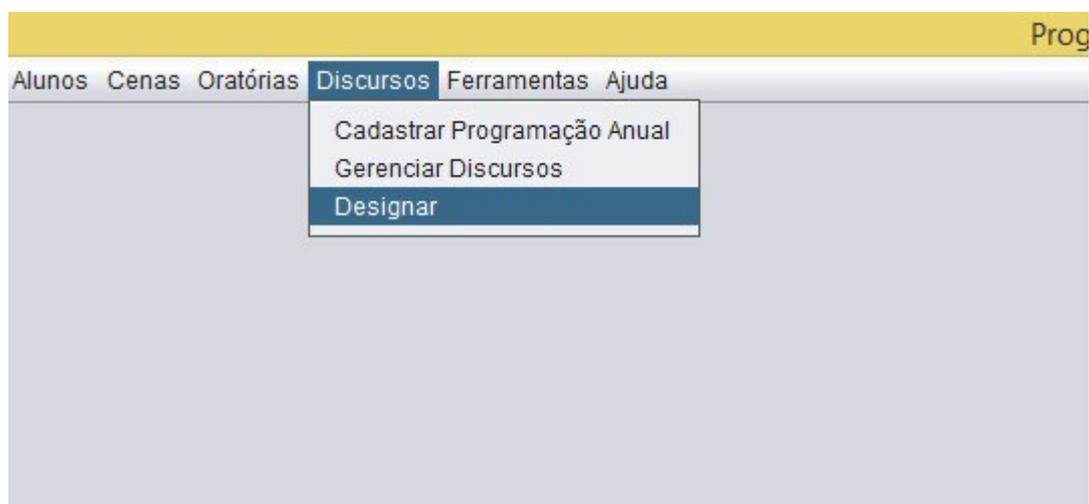


Figura 11: Selecionando o item de menu para designar discursos



Figura 12: Janela para escolha do período de designação



Figura 13: Janela que contém 2 semanas para serem designadas



Figura 14: Tela de geração de relatórios

Oradores Ajudantes Características de Oratória Cenas Servos e Anciãos

Todos

Discursos Realizados	Nome
5	Alexandre Babilone Fonseca
3	Cilene
2	Júlio
2	Wagner
1	Wallace

Somente Irmãos

Discursos Realizados	Nome
6	Nivaldo
5	Alexandre Babilone Fonseca
2	Júlio
2	Wagner
1	Irondilson
1	Wallace

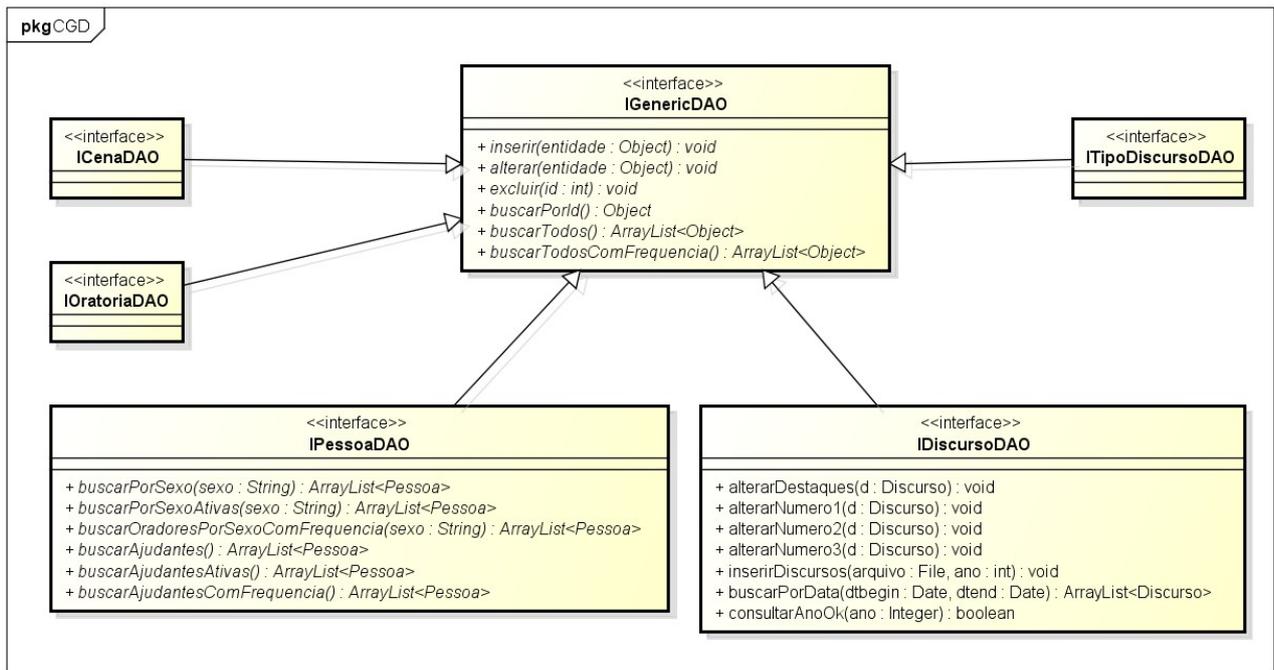
Somente Irmãs

Discursos Realizados	Nome
6	Laisa Rossim
3	Cilene
1	Sara

Figura 15: Tela de consulta de frequências

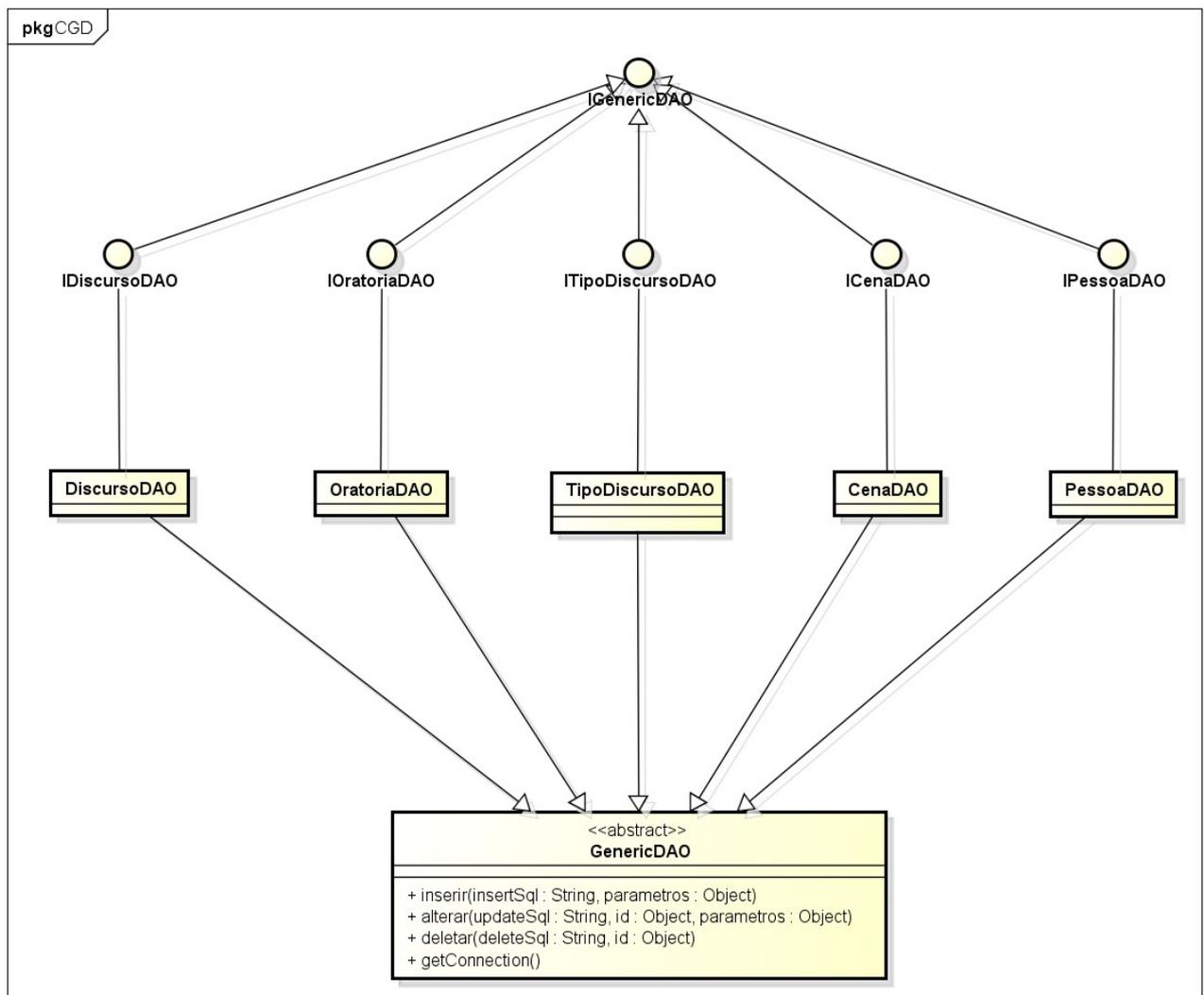
5.3 – Camada de Gerência de Dados

Para cada classe de domínio a ser persistida, devem ser criadas uma classe *DAO e uma interface I*DAO correspondente, que implementa a interface genérica IGenericDAO. A primeira deve herdar de GenericDAO, uma classe abstrata que possui as funcionalidades básicas de acesso ao mecanismo de persistência, e deve implementar a interface DAO associada. As Figura 16: Modelo de Interfaces DAO e Figura 17: Arquitetura DAO do sistema mostram a arquitetura claramente.



powered by Astah

Figura 16: Modelo de Interfaces DAO



powered by Astah

Figura 17: Arquitetura DAO do sistema

A implementação no banco de dados segue fielmente o modelo Entidade-Relacionamento abaixo :

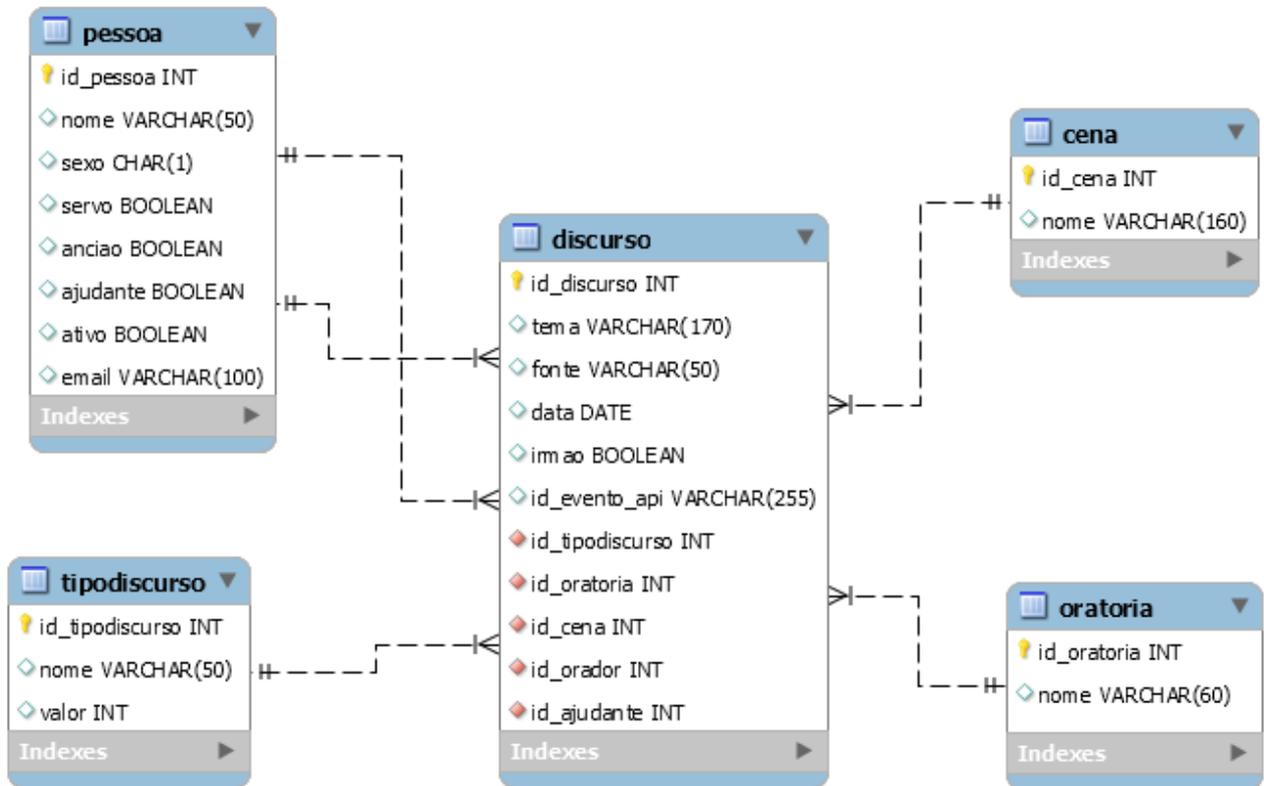


Figura 18: Modelo Entidade-Relacionamento