



UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO

CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA

ESCOLA DE INFORMÁTICA APLICADA

Software para Geração da Grade Horária do curso BSI da UNIRIO

Diego Rafael Di Berardino

Orientador

Alexandre Luis Correa

RIO DE JANEIRO, RJ – BRASIL

AGOSTO DE 2013

Software para Geração da Grade Horária do curso BSI da UNIRIO

Diego Rafael Di Berardino

Projeto de Graduação apresentado à Escola de
Informática Aplicada da Universidade Federal do
Estado do Rio de Janeiro (UNIRIO) para obtenção do
título de Bacharel em Sistemas de Informação.

Aprovada por:

Prof. Alexandre Luis Correa, D.Sc. (UNIRIO)

Prof. Gleison dos Santos Souza, D.Sc. (UNIRIO)

Prof. Mariano Pimentel, D.Sc. (UNIRIO)

RIO DE JANEIRO, RJ – BRASIL.

AGOSTO DE 2013

Agradecimentos

À minha família, pelo apoio e incentivo nesta jornada universitária.

Aos amigos da universidade pelos momentos de descontração e discussões sobre os mais diversos assuntos.

Aos professores do corpo docente do BSI pelos conhecimentos adquiridos e, especialmente ao meu orientador, o professor Alexandre Correa, pela atenção, dedicação, paciência e transmissão de conhecimentos.

RESUMO

O processo de geração da grade horária de um curso universitário é complexo e influencia a rotina diária de todos os envolvidos durante um período letivo. Uma grande diversidade de problemas associados a esse assunto já foram relatados e diferentes soluções já foram propostas, sendo estas tipicamente dependentes das situações específicas de uma instituição.

No curso de Bacharelado em Sistemas de Informação (BSI) da Universidade Federal do Estado do Rio de Janeiro (UNIRIO), este trabalho é realizado de forma manual, resultando em grande quantidade de esforço e comunicação. O presente trabalho visa propor um instrumento de apoio automatizado para o curso, com o objetivo de reduzir o labor atualmente gasto na geração da grade horária, respeitando as restrições e diretrizes definidas pela instituição.

Palavras-chave: grade horária, restrições.

ABSTRACT

The process of timetable definition for college education is complex and influences the daily routine of everyone involved during an entire semester. A wide range of problems associated with this issue have been reported and different solutions have been proposed, which are typically dependent on the specific properties and policies of each institution.

In the context of the course Bachelor of Information Systems (BIS) held at Federal University of the State of Rio de Janeiro (UNIRIO), this process is essentially manual requiring a lot of effort and communication. This work proposes a tool to support this process, aiming at reducing the labor spent on timetable definition, by automatically generating a timetable satisfying the constraints and guidelines set by the institution.

Keywords: timetable, constraints.

Sumário

1. INTRODUÇÃO	9
1.1. Contextualização	9
1.2. Motivação.....	10
1.3. Objetivo	10
1.4. Organização.....	10
2. REQUISITOS	11
2.1. Processo de Geração da Grade Horária	11
2.2. Modelo Conceitual	14
2.3. Restrições	15
2.4. Modelo de Casos de Uso	17
3. PROJETO DA SOLUÇÃO	22
3.1. Visão Geral da Aplicação	22
3.2. Tecnologias Utilizadas	24
3.2.1. Microsoft Excel	25
3.2.2. Java	25
3.2.3. Prolog	27
3.2.4. Bootstrap.....	27
3.2.5. Banco de Dados Relacional	28
3.3. Modelo Lógico de Dados	28
3.4. Módulo de Cadastros Básicos	30
3.5. Módulo de Carga dos Cadastros Básicos.....	34
3.6. Módulo Gerador da Grade Horária	37
4. CONCLUSÃO	40
4.1. Contribuições	40
4.2. Limitações	40
4.3. Trabalhos Futuros.....	41
4.4. Considerações Finais	41

Referências Bibliográficas	42
Anexo A – Documento de Visão.....	43
1. Objetivo.....	43
2. Visão Geral do Problema.....	43
3. Envolvidos no Processo de Geração da Grade Horária.....	44
4. Problemas Observados no Processo	45
5. Soluções	45
Anexo B – Especificação dos Casos de Uso (Concretos)	46

ÍNDICE DE FIGURAS

Figura 1 - Grade horária para o 1º período em 2011.1	13
Figura 2 - Grade horária para o 5º período em 2011.1	13
Figura 3 - Modelo Conceitual - Grade Horária	15
Figura 4 - Diagrama de Casos de Uso do Sistema	17
Figura 5 – Conceitos envolvidos no caso de uso Manter Informações das disciplinas do curso	18
Figura 6 – Conceitos envolvidos no caso de uso Manter informações de professores do curso.	19
Figura 7 – Conceitos envolvidos no caso de uso Definir ofertas de disciplinas do período letivo	19
Figura 8 – Conceitos envolvidos no caso de uso Definir grade horária disponível para o período letivo.....	20
Figura 9 – Conceitos envolvidos no caso de uso Definir preferências de horário dos professores para o período letivo.....	20
Figura 10 – Conceitos envolvidos no caso de uso Gerar grade horária para o período letivo	21
Figura 11 – Visão geral dos componentes da aplicação.....	23
Figura 12 – Modelo Lógico de Dados.....	29
Figura 13 – Planilha Excel: Cadastro de Disciplinas	31
Figura 14 – Planilha Excel: Cadastro de Professores.....	31
Figura 15 – Planilha Excel: Cadastro de Ofertas de Disciplinas.....	32
Figura 16 – Planilha Excel: Cadastro de Horários Disponíveis	33
Figura 17 – Planilha Excel: Cadastro de Preferências de Horários dos Professores.....	34
Figura 18 – Tela para Carga dos Cadastros Básicos	35
Figura 19 – Árvore de elementos da planilha Excel em XML.....	35
Figura 20 – Estrutura de classes de acesso a dados (padrão DAO).....	36
Figura 21 – Predicados correspondentes a Professor, Disciplina e Oferta.....	37
Figura 22 – Predicados referentes a slot e preferências de horários.....	37
Figura 23 – Código Java para integração com Prolog.....	38
Figura 24 – Exemplo de saída gerada pelo programa Prolog.....	39
Figura 25 – Exemplo de saída gerada pelo programa Prolog.....	39

1. INTRODUÇÃO

1.1. Contextualização

No âmbito da universidade, existem processos acadêmicos semestrais que são essenciais para o funcionamento de todos os cursos de graduação. Na Escola de Informática Aplicada (EIA) da UNIRIO, foram identificados dezoito processos de prestação de serviço [ENGIEL 2009], dentre os quais podem ser citados: inscrição em disciplinas, geração de relatórios da situação de alunos, geração da grade horária para o período letivo.

Este trabalho se insere no contexto do processo de geração da grade horária do curso de Bacharelado em Sistemas de Informação (BSI) oferecido pela EIA. O principal produto produzido pelo processo de geração da grade horária é um quadro de ofertas de disciplinas alocadas em horários e dias da semana que tenta satisfazer as preferências de professores e alunos, respeitando as restrições e diretrizes definidas pela instituição.

A alocação dos horários das disciplinas influencia fortemente a atividade de uma instituição de ensino durante um período letivo, sendo necessário que todos os envolvidos se adaptem à mesma. Uma grande diversidade de problemas associados a esse assunto já foram relatados e diferentes soluções foram propostas, sendo estas tipicamente dependentes das situações específicas de uma instituição. Assim, não há uma solução universal que possa ser adotada em todos os casos [HAMAWAKI 2005].

Wren (1996) define o problema da geração de grade horária escolar como um arranjo de elementos dentro de padrões de tempo ou espaço, no qual um conjunto de objetivos deve ser total ou quase totalmente atendido, satisfazendo um conjunto de restrições da forma mais completa possível. Fernandes (2002) classifica as restrições para o problema da grade horária escolar em restrições fortes e fracas: violações de restrições fortes (e.g., a alocação de um professor em duas salas diferentes em um mesmo horário) impedem a geração de uma solução válida, enquanto que violações de restrições fracas, apesar de não impedirem a geração de soluções válidas, podem afetar negativamente a qualidade da solução (e.g., a violação de regras de preferência dos professores por determinados horários pode gerar insatisfação de alguns professores).

1.2. Motivação

A grande quantidade de restrições de horários das disciplinas e a dificuldade de satisfazer todas as preferências de horários dos professores das diferentes linhas e departamentos fazem com que a obtenção da grade horária final para o semestre seja uma tarefa complexa. Atualmente esse trabalho é realizado de forma manual, resultando em uma demanda concentrada de esforço e comunicação. Além disso, a suscetibilidade a erros é inevitável, podendo gerar insatisfação dos interessados, i.e, professores e alunos.

O presente trabalho propõe um instrumento de apoio automatizado ao processo de geração da grade horária de um período letivo, de forma a reduzir o esforço atualmente despendido na sua realização.

1.3. Objetivo

O objetivo principal deste trabalho é implementar um sistema que, a partir das informações das disciplinas, professores alocados nas ofertas das disciplinas, preferências e restrições de horário, gere uma ou mais possibilidades de grade horária para um período letivo.

1.4. Organização

Este trabalho está dividido em quatro capítulos, incluindo esta introdução. O segundo capítulo descreve conceitualmente a solução proposta, incluindo os requisitos levantados, os casos de uso propostos e o modelo conceitual de dados.

O capítulo 3 aborda o projeto técnico da solução, as tecnologias utilizadas, as integrações entre elas e a arquitetura da solução.

O quarto capítulo é o de conclusão, citando as contribuições realizadas, as limitações do presente projeto e os espaços para trabalhos futuros.

Complementando os capítulos descritos acima, seguem-se os anexos A e B, contendo documentos relevantes para a solução proposta. O anexo A engloba o documento de visão do projeto, enquanto que o anexo B contém a descrição dos casos de uso citados no capítulo 2.

2. REQUISITOS

A solução proposta para o problema da definição da grade horária de um período letivo do BSI é um sistema que, a partir da definição das ofertas de disciplinas, professores alocados e suas preferências de horários, gera uma proposta de grade horária, considerando a sua distribuição pelos oito períodos que compõem o curso. Este capítulo descreve o contexto onde o sistema se insere e os requisitos da solução proposta.

Para se chegar a tal solução, foi necessário realizar um levantamento dos requisitos do sistema. Os requisitos foram identificados a partir de reuniões com a diretora da Escola de Informática Aplicada à época do levantamento, Professora Leila Andrade, e com o Professor Alexandre Correa, do quadro de professores do BSI. Além disso, diversas informações foram extraídas por meio da análise de grades horárias de períodos anteriores.

2.1. Processo de Geração da Grade Horária

Esta seção descreve o processo de definição da grade horária de um período letivo do BSI, os conceitos e regras envolvidos na sua elaboração.

O curso de BSI é oferecido semestralmente em turnos tarde/noite de segunda à sexta-feira. Atualmente, a grade curricular do curso é composta por oito períodos, sendo os quatro iniciais predominantemente vespertinos (14h às 18h) e os quatro finais predominantemente noturnos (18h às 22h). Cada período agrupa um conjunto de disciplinas que são classificadas em obrigatórias, optativas e eletivas. As disciplinas obrigatórias devem ser cursadas por todo aluno que deseje concluir o curso. Além das disciplinas obrigatórias, o discente deve cursar oito disciplinas optativas (disciplinas do curso que podem ser escolhidas de acordo com a preferência do aluno) e quatro disciplinas eletivas (disciplinas de outros cursos, também de livre escolha). Cada disciplina pode ter nenhuma, uma ou mais de uma oferta em um semestre. Em geral, as disciplinas obrigatórias são ofertadas semestralmente, o que não ocorre necessariamente com as disciplinas optativas.

O processo de definição da grade horária começa cerca de dois meses antes do início de um novo período letivo. Cada área de conhecimento da EIA (e.g., Redes, Banco de Dados, Engenharia de Software, Algoritmos/RCR) decide internamente as disciplinas que serão ofertadas e os respectivos professores alocados, obtém os horários de preferência de cada professor da área e, encaminha, através de um docente responsável, tais informações à coordenação do curso. Processo similar ocorre em outros departamentos que oferecem disciplinas do curso, e.g., Departamento de Matemática e Estatística (DME). A partir da definição das ofertas de disciplinas e das preferências de alocação de horário de cada professor, o coordenador do curso é encarregado de gerar a grade horária do período, tentando satisfazer as preferências de professores e alunos da melhor forma possível. Essa atividade é iterativa, pois pode haver a necessidade de negociações entre a coordenação e alguns professores para que algumas alocações sejam feitas em horários diferentes daqueles inicialmente sugeridos pelos professores, de forma a satisfazer restrições e orientações da coordenação.

As figuras a seguir apresentam fragmentos da grade horária de períodos letivos anteriores que serviram como base para o levantamento de requisitos do sistema. A Figura 1 apresenta a alocação das ofertas de disciplinas do 1º período do curso, onde é possível observar a predominância de aulas na faixa horária compreendida entre 14 e 18 horas. De forma geral, as disciplinas ocupam dois slots de duas horas (para disciplinas de 4 horas semanais) ou de três horas (para disciplinas de 6 horas semanais) de duração, distribuídos ou às segundas e quartas, ou às terças e quintas ou em dois slots consecutivos de duas horas às sextas-feiras. A grade ilustrada na Figura 2 corresponde às disciplinas do 5º período do curso, onde é possível observar a predominância de alocação na faixa horária das 18 às 22 horas.

Quadro Semanal de Aulas - Semestre 2011.1					
Hora	Segunda	Terça	Quarta	Quinta	Sexta
13-14	TIN0107-Téc.Progr. 1 Geiza Lab3		TIN0107-Téc.Progr. 1 Geiza		HTD0058-Teor. Práti.Disc. Lúcia Mostiako
14-15	TIN0107-Téc.Progr. 1 Geiza Lab3	TIN0108-Org.Comp. MORGANNA Lab3	TIN0107-Téc.Progr. 1 Geiza Lab3	TIN0108-Org.Comp. MORGANNA Lab3	HTD0058-Teor. Práti.Disc. Lúcia Mostiako
15-16	TIN0107-Téc.Progr. 1 Geiza Lab3	TIN0108-Org.Comp. MORGANNA Lab3	TIN0107-Téc.Progr. 1 Geiza	TIN0108-Org.Comp. MORGANNA Lab3	HTD0058-Teor. Práti.Disc. Lúcia Mostiako
16-17	TIN0112-Fund SI SEAN SIQUEIRA Lab3	TIN0106-Desenv.PáginaWEB PIMENTEL Lab3	TIN0112-Fund SI SEAN SIQUEIRA sala Lab3	TME0101-MatBásica Lab3	HTD0058-Teor. Práti.Disc. Lúcia Mostiako
17-18	TIN0112-Fund SI SEAN SIQUEIRA Lab3	TIN0106-Desenv.PáginaWEB PIMENTEL Lab3	TIN0112-Fund SI SEAN SIQUEIRA Sala lab3	TME0101-MatBásica Lab3	
18-19				TIN0106-Desenv.PáginaWEB PIMENTEL Lab3	
19-20				TIN0106-Desenv.PáginaWEB PIMENTEL Lab3	
20-21					
21-22					
Legendas					
	Disciplina Obrigatória Informática				
	Atividade Curricular de Extensão (estágio e cursos extras)				
	Disciplina Eletiva - disciplina de outro curso (mínimo de 16 créditos)				
	Disciplina Optativa - Disciplina de Informática (mínimo de 8 disciplinas)				

Figura 1 - Grade horária para o 1º período em 2011.1

Quadro Semanal de Aulas - Semestre 2011.1					
	Segunda	Terça	Quarta	Quinta	Sexta
16-17	TIN0163 - BANCO DE DADOS II FERNANDA BAIÃO		TIN0163 - BANCO DE DADOS II FERNANDA BAIÃO		
17-18	TIN0163 - BANCO DE DADOS II FERNANDA BAIÃO		TIN0163 - BANCO DE DADOS II FERNANDA BAIÃO		
18-19	TIN0126 -REDES II Beto	TIN0118 - Análise de Algoritmos Vania	TIN0126 -REDES II Beto	TIN0118 - Análise de Algoritmos Vania	
19-20	TIN0126 -REDES II Beto	TIN0118 - Análise de Algoritmos Vania	TIN0126 -REDES II Beto	TIN0118 - Análise de Algoritmos Vania	
20-21	TIN0171 -PCS ALEXANDRE CORREA	TIN0130-EMPREENDEDORISMO CELSO CAMPOS	TIN0171 -PCS ALEXANDRE CORREA	TIN0130-EMPREENDEDORISMO CELSO CAMPOS	
21-22	TIN0171 -PCS ALEXANDRE CORREA	TIN0130-EMPREENDEDORISMO CELSO CAMPOS	TIN0171 -PCS ALEXANDRE CORREA	TIN0130-EMPREENDEDORISMO CELSO CAMPOS	
Legendas					
	Disciplina Obrigatória Informática				
	Atividade Curricular de Extensão (estágio e cursos extras)				
	Disciplina Eletiva - disciplina de outro curso (mínimo de 16 créditos)				
	Disciplina Optativa - Disciplina de Informática (mínimo de 8 disciplinas)				

Figura 2 - Grade horária para o 5º período em 2011.1

2.2. Modelo Conceitual

As informações coletadas no levantamento de requisitos foram sintetizadas no modelo conceitual apresentado na Figura 3. Um curso é formado por uma sequência de períodos curriculares (no caso do BSI, períodos numerados de 1 a 8). Cada período curricular é formado por um conjunto de disciplinas recomendadas. Em cada período letivo (semestre/ano), são definidas as ofertas de disciplinas e os respectivos professores alocados. Cada professor alocado em uma oferta é lotado em um departamento, sendo que nem todas as disciplinas do curso são oferecidas por um único departamento (e.g., o curso de BSI contém disciplinas do Departamento de Matemática e Estatística).

Em cada período letivo são definidos todos os slots de horário disponíveis para alocação das ofertas de disciplinas do curso. Essa definição é representada no modelo por meio do conceito *Grade Disponível*, que define todos os slots de horário disponíveis para um determinado curso em um período letivo. Cada slot define um horário de início e de fim, o que permite representar slots de 2 horas ou de 3 horas (e.g. slot segunda-feira, 18 às 20; slot quarta-feira, 13 às 16).

Cada professor define suas preferências de horário para um período letivo em ordem de prioridade. Cada preferência, representada no modelo pelo conceito *PreferenciaHorario*, define um conjunto de slots que melhor atenderiam aos seus interesses. Por exemplo, o professor A poderia definir três preferências na seguinte ordem:

1 - 2as 16 às 18, 2as 18 às 20, 2as 20 às 22; 4as 16 às 18, 4as 18 às 20, 4as 20 às 22.

2 - 2as 18 às 20; 2as 20 às 22; 4as 18 às 20, 4as 20 às 22.

3 - 3as 16 às 18, 3as 18 às 20; 6as 18 às 20, 6as 20 às 22.

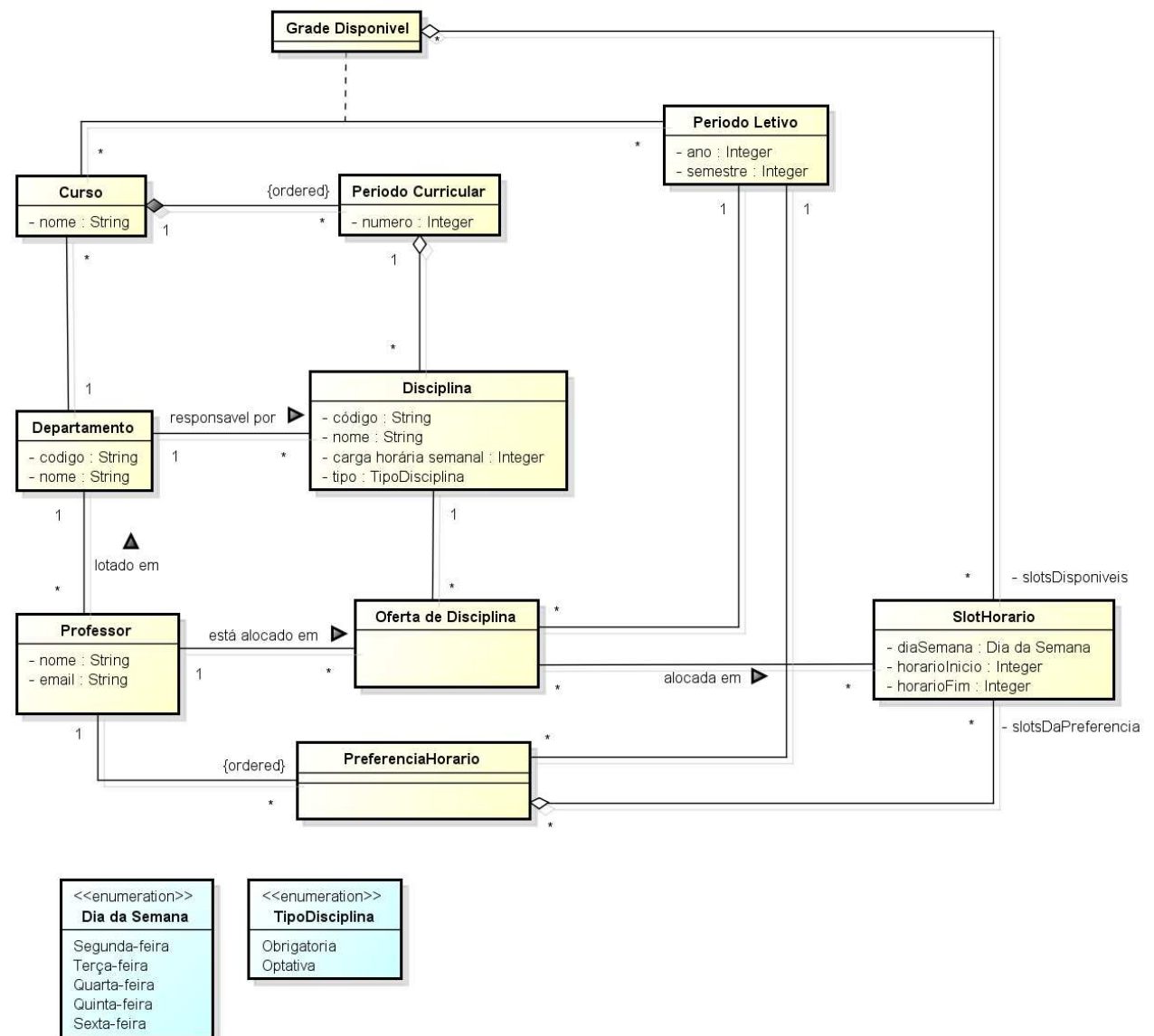


Figura 3 - Modelo Conceitual - Grade Horária

2.3. Restrições

O problema que o sistema tem a resolver consiste em definir um conjunto de instâncias da associação *Oferta de Disciplina alocada em Slot Horário* representada no modelo conceitual da Figura 3 a partir das definições das ofertas, dos slots de horário disponíveis para o período letivo e das preferências de horário de cada professor alocado em ofertas de disciplina para o período letivo, respeitando o conjunto de restrições existentes. O conjunto de restrições relevantes para a definição da grade

horária é apresentado a seguir. Cada restrição é definida em linguagem natural e também por meio de uma invariante expressa em OCL (Object Constraint Language). As restrições foram classificadas em restrições fortes, i.e., aquelas que não podem ser violadas, e restrições fracas, i.e., restrições que podem eventualmente ser violadas por uma solução.

Restrições Fortes

- 1) Não pode haver interseção de horário entre as ofertas de disciplinas de um mesmo período curricular.

context PeríodoLetivo

inv: ofertaDisciplina->forAll(oferta1, oferta2 |

oferta1.disciplina.periodoCurricular = oferta2.disciplina.periodoCurricular
implies

oferta1.slotHorario->intersection(oferta2.slotHorario)->size() = 0

- 2) Em um período letivo, não pode haver interseção de horário entre as ofertas de disciplinas ministradas por um mesmo professor.

context PeríodoLetivo

inv: ofertaDisciplina->forAll(oferta1, oferta2 |

oferta1.disciplina.professor = oferta2.disciplina.professor implies

oferta1.slotHorario->intersection(oferta2.slotHorario)->size() = 0

Restrições Fracas

- 3) Disciplinas do 1o ao 4o período devem ser alocadas preferencialmente no período vespertino.

context PeríodoLetivo

inv: ofertaDisciplina->forAll(oferta | oferta.disciplina.periodoCurricular.numero
<= 4 implies oferta.slotHorario->select(slot | slot.horarioFim > 18)->isEmpty())

- 4) Disciplinas do 5o ao 8o período devem ser alocadas preferencialmente no período noturno.

context PeríodoLetivo

inv: ofertaDisciplina->forAll(oferta | oferta.disciplina.periodoCurricular.numero >= 5 implies oferta.slotHorario->select(slot | slot.horarioFim <18)->isEmpty())

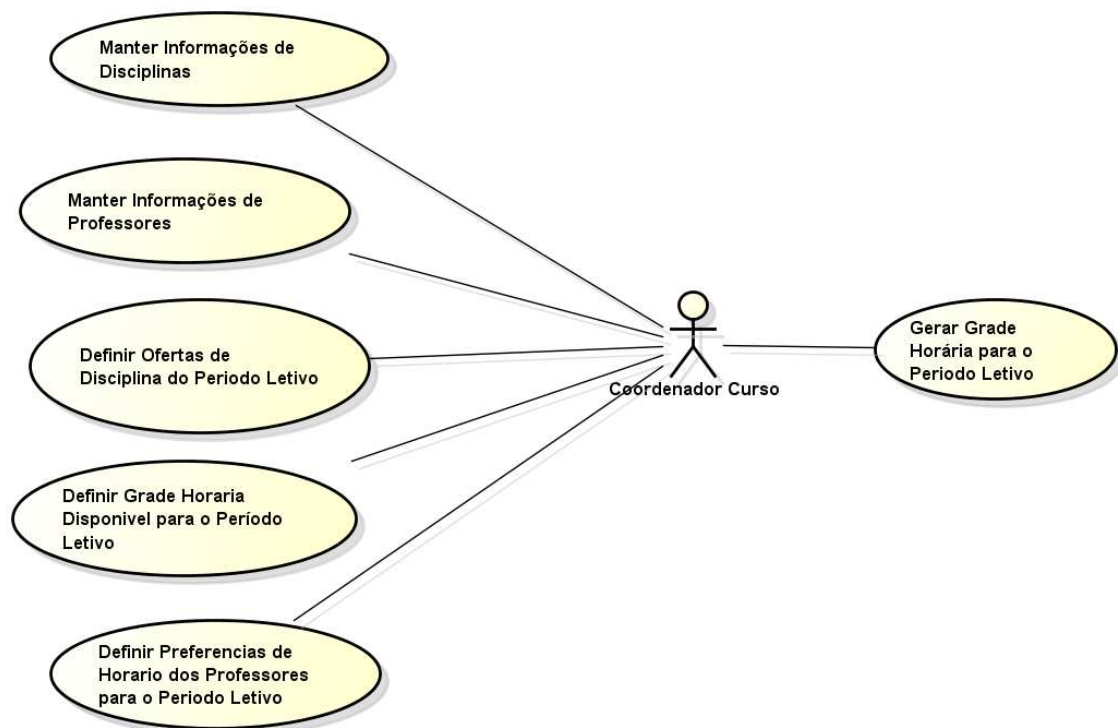
- 5) As ofertas de disciplina devem ser alocadas em horários da preferência do professor alocado na mesma.

context OfertaDisciplina

inv: professor.preferenciaHorario->exists(preferência | preferência.slotsDaPreferencia->includesAll(professor.ofertasDisciplina.slotHorario))

2.4. Modelo de Casos de Uso

Os requisitos funcionais para o sistema de grade horária foram organizados nos casos de uso presentes na Figura 4.



powered by astah

Figura 4 - Diagrama de Casos de Uso do Sistema

A solução proposta prevê apenas o coordenador do curso como usuário. Considerando que o volume de informações que precisam ser atualizadas é pequeno e que essas atualizações são realizadas apenas duas vezes por ano, optou-se por uma solução inicial mais simples. Essa decisão teve como propósito facilitar tanto a implementação como a implantação inicial da solução, já que o principal objetivo inicialmente desejado consiste em reduzir o esforço despendido pelo coordenador do curso para gerar a grade horária de um período letivo. Futuras evoluções desta solução poderão permitir que os próprios professores definam suas informações, ofertas de disciplinas e preferências de horário. A solução implementada, portanto, assume que essas informações vão continuar a ser definidas pelas áreas, e.g., Redes, Banco de Dados, Engenharia de Software, Algoritmos/RCR, e passadas para o coordenador do curso da mesma forma como atualmente.

Os objetivos de cada caso de uso identificado para o sistema são brevemente descritos a seguir.

- **Manter informações das disciplinas do curso**

Cadastrar as disciplinas que compõem o curso, registrando o código e nome da disciplina, o período curricular ao qual a mesma pertence, a carga horária semanal e o tipo (obrigatória / optativa). Este caso de uso possibilita a manutenção dos conceitos Período Curricular e Disciplina definidos no modelo conceitual (Figura 5).

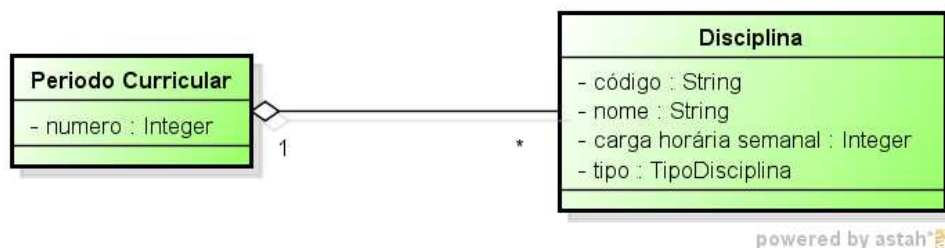


Figura 5 – Conceitos envolvidos no caso de uso Manter Informações das disciplinas do curso

- **Manter informações de professores do curso**

Cadastrar informações dos professores do curso e seus respectivos departamentos. Este caso de uso possibilita a manutenção dos conceitos Departamento e Professor definidos no modelo conceitual (Figura 6).



Figura 6 – Conceitos envolvidos no caso de uso Manter informações de professores do curso

- **Definir ofertas de disciplina do período letivo**

Cadastrar as ofertas de disciplina de um período letivo, informando a disciplina e o professor alocado à mesma. Este caso de uso possibilita a manutenção do conceito Oferta de Disciplina e das suas associações com os conceitos Professor, Disciplina e Período Letivo definidos no modelo (Figura 7).

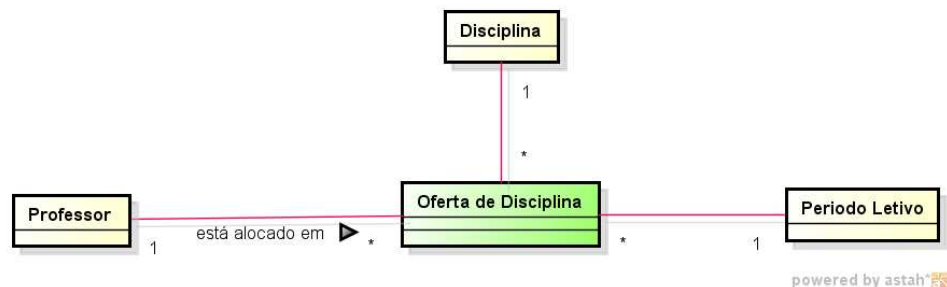


Figura 7 – Conceitos envolvidos no caso de uso Definir ofertas de disciplinas do período letivo

- **Definir grade horária disponível para o período letivo**

Cadastrar os slots disponíveis para alocação das ofertas de disciplinas do curso em um período letivo. Este caso de uso possibilita a manutenção dos conceitos Grade Disponível e SlotHorario definidos no modelo (Figura 8).

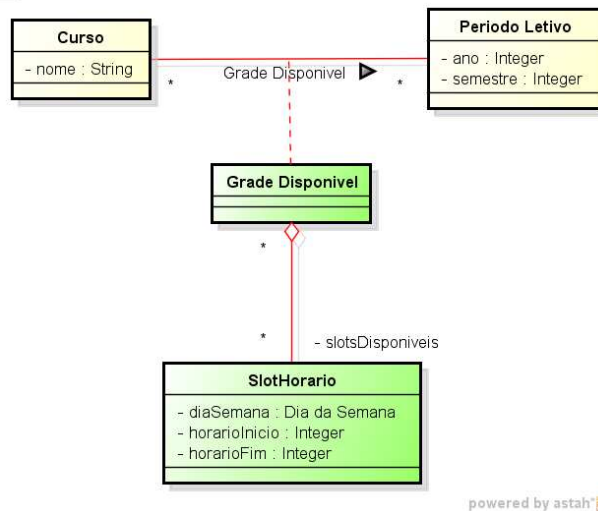


Figura 8 – Conceitos envolvidos no caso de uso Definir grade horária disponível para o período letivo

- Definir preferências de horário dos professores para o período letivo**
 Cadastrar as listas de preferências de horário dos professores que serão alocados nas ofertas de disciplinas em um período letivo. Este caso de uso possibilita a manutenção do conceito PreferenciaHorario e suas associações com Professor, Período Letivo e SlotHorario definidos no modelo conceitual (Figura 9). Apenas os slots disponíveis para o curso no respectivo período letivo poderão fazer parte das listas de preferências dos professores.

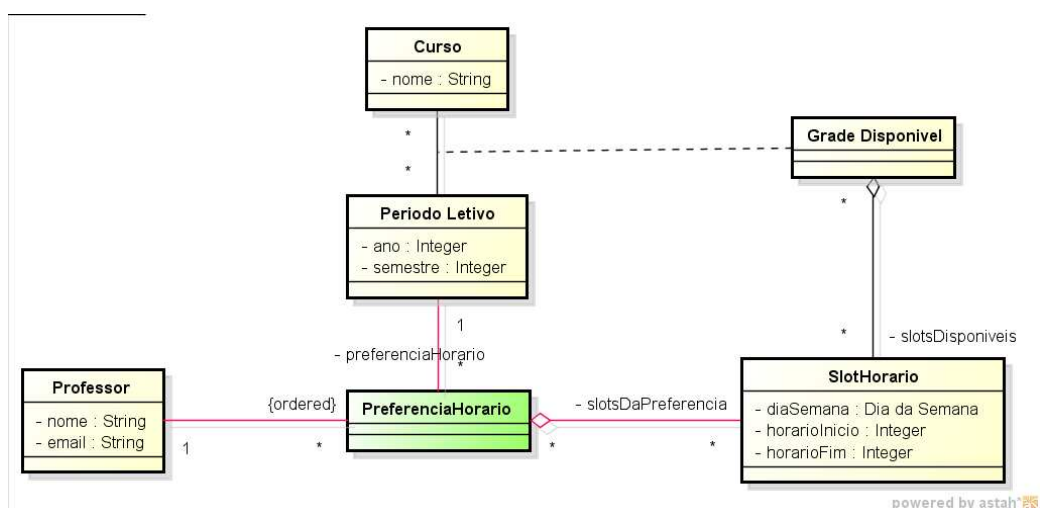


Figura 9 – Conceitos envolvidos no caso de uso Definir preferências de horário dos professores para o período letivo

- **Gerar grade horária para o período letivo**

Apresentar a grade horária gerada pelo sistema para o período letivo contendo as alocações das ofertas de disciplinas aos slots de horário disponíveis, respeitando todas as restrições fortes e tentando satisfazer ao máximo as restrições fracas, e.g., preferências de horário dos professores.

Este caso de uso é responsável por gerar instâncias da associação entre os conceitos Oferta de Disciplina e Slot Horário definidos no modelo conceitual Figura 10).

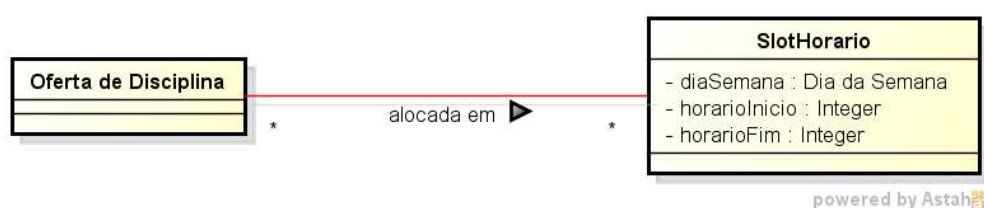


Figura 10 – Conceitos envolvidos no caso de uso Gerar grade horária para o período letivo

3. PROJETO DA SOLUÇÃO

O capítulo 2 descreveu os requisitos da solução proposta para o problema da geração de grade horária para o curso de Bacharelado de Sistemas de Informação da UNIRIO. As informações e as funcionalidades da solução proposta foram respectivamente representadas em um modelo conceitual e em um modelo de casos de uso. Esses modelos formam o alicerce para a construção da solução. Este capítulo apresenta o projeto técnico da solução, descrevendo os componentes do sistema, as tecnologias utilizadas na implementação de cada componente, as justificativas para a escolha das mesmas e a integração entre os componentes.

3.1. Visão Geral da Aplicação

Inicialmente, o usuário preenche uma planilha Excel com as informações necessárias para a geração da grade horária para o semestre, o que engloba: informações das disciplinas, professores, ofertas de disciplina, horários disponíveis e preferências de horário dos professores. Isso corresponde à implementação dos casos de uso *Manter Informações de Disciplinas*, *Manter Informações de Professores*, *Definir Ofertas de Disciplina do Período Letivo*, *Definir Grade Horária Disponível para o Período Letivo* e *Definir Preferências de Horário dos Professores para o Período Letivo*, descritos no capítulo 2.

Após isso, o usuário deve salvar a planilha em formato XML e acessar uma aplicação web para fazer a carga das informações presentes nessa planilha para o banco de dados. Realizada a carga dos dados, o usuário solicita a geração da grade horária. O módulo gerador da grade recupera as informações necessárias do banco de dados, gera um arquivo de fatos de Prolog e dispara a execução de um programa Prolog responsável pela geração de uma grade horária. Por fim, o módulo gerador da grade exibe a grade de horários resultante em uma página HTML. Esse processo representa a implementação do caso de uso *Gerar grade horária para o período letivo*.

A Figura 11 apresenta uma visão geral dos componentes da aplicação e o encadeamento entre eles através dos artefatos gerados.

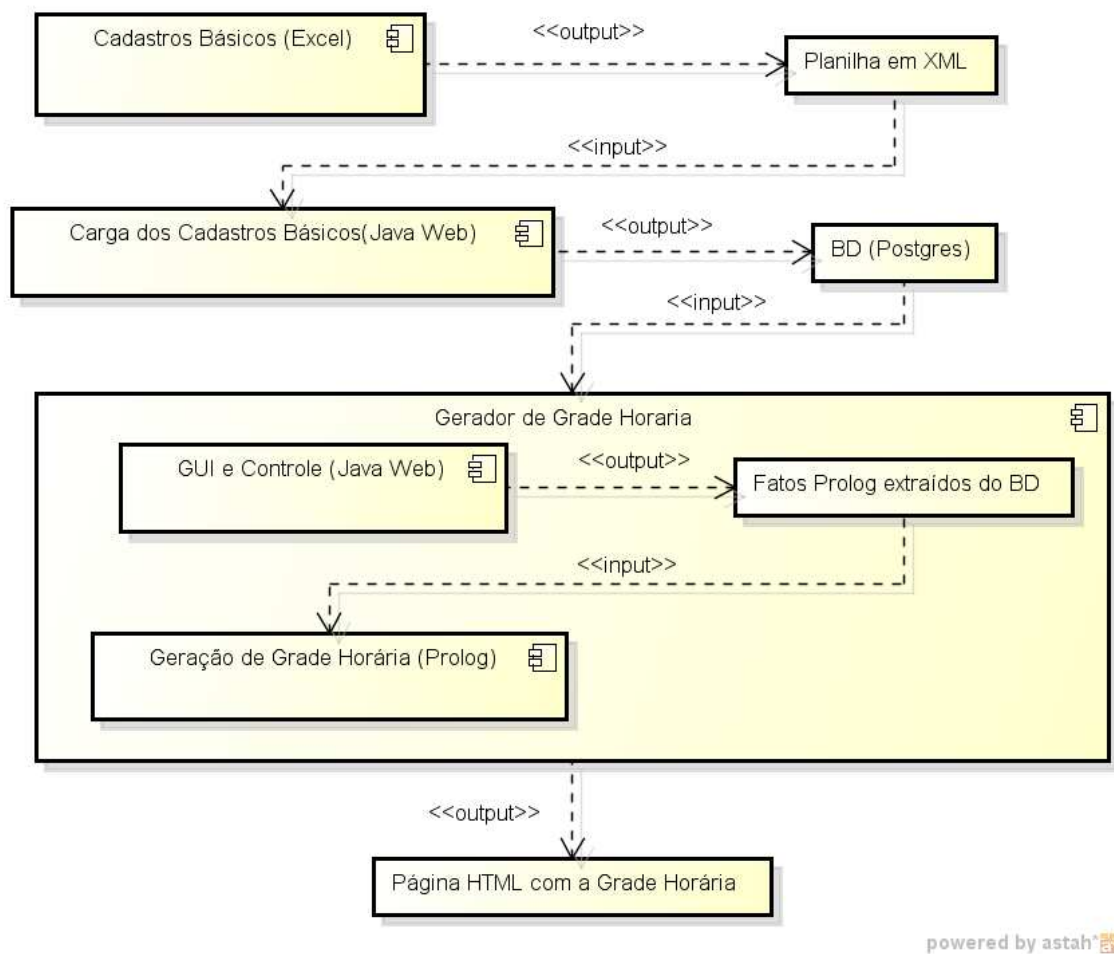


Figura 11 – Visão geral dos componentes da aplicação

O componente Cadastros Básicos representa a implementação dos casos de uso de cadastros básicos por meio de uma planilha Microsoft Excel. A saída produzida por este módulo é uma planilha Excel no formato XML contendo informações das disciplinas, professores, ofertas de disciplinas, slots e preferências de horários dos professores.

O componente Carga dos Cadastros Básicos realiza a carga do banco de dados utilizado pela aplicação a partir dos dados contidos na planilha em formato XML. A utilização de um banco de dados teve como objetivo deixar o sistema preparado para uma futura substituição do componente Cadastros Básicos por uma implementação onde os professores poderão registrar suas ofertas e preferências de horários por meio de uma aplicação web.

O componente Gerador da Grade Horária é formado por dois componentes internos: o componente GUI e Controle, implementado em Java, é responsável por ler do banco de dados Postgres os registros de disciplinas, professores, ofertas, slots e preferências de horários e transformar esses registros em um arquivo de fatos em Prolog. Esse arquivo de fatos é consumido pelo componente de geração de grade horária implementado como um programa Prolog. Esse componente Prolog retorna a grade horária gerada ao componente Java, que formata a grade horária gerada e a exibe para o usuário.

3.2. Tecnologias Utilizadas

O sistema foi implementado fazendo uso de diversas soluções e plataformas:

- Microsoft Excel: ferramenta utilizada pelo usuário no cadastramento das informações necessárias para o processamento da grade horária. Através do Excel, o usuário edita os dados das disciplinas, professores, ofertas de disciplina, slots de horário e preferências de horário dos professores.
- Java: foi utilizada a plataforma Java Enterprise Edition (Java EE) na implementação da aplicação web hospedada no container web Tomcat 7. Alguns frameworks para desenvolvimento de aplicações Java também foram utilizados: o *Struts 2* para controle da apresentação e navegação, o *Hibernate* para armazenamento e recuperação de objetos Java em um servidor de banco de dados relacional e o *JUnit* para testes unitários.
- Banco de Dados Relacional PostgreSQL: sistema gerenciador de banco de dados (SGBD) utilizado como fonte de dados para a aplicação Java responsável pela geração da grade horária.
- Prolog: usado na geração de soluções viáveis para o problema da grade horária a partir das informações de disciplinas, professores e restrições por meio da ferramenta SWI-Prolog.
- Bootstrap – framework para desenvolvimento de front-end para aplicações web.

As seções a seguir descrevem mais detalhes sobre as tecnologias utilizadas, justificando suas escolhas.

3.2.1. Microsoft Excel

O sistema utiliza o Excel como interface com o usuário para o cadastramento das informações a serem utilizadas no processo de geração da grade horária. O Excel possui uma interface intuitiva, conhecida pelos potenciais usuários da solução, possibilita agilidade na edição de informações e o seu conteúdo pode ser armazenado em formato XML, o que permite uma fácil integração com outras soluções.

A escolha do Excel para essa parte da solução deve-se ao fato de que, em sua versão inicial, consideramos a existência de apenas um usuário responsável por registrar essas informações: o coordenador do curso. Além disso, consideramos que o volume de dados para edição seria baixo a partir do segundo semestre de uso (basicamente mudanças pontuais em alocações de disciplinas e preferências por horários) e a frequência de utilização seria apenas eventual (uma vez por semestre).

No futuro, caso haja interesse na evolução para uma solução mais colaborativa e distribuída, onde os professores possam manter diretamente as informações sobre suas alocações e preferências, esta parte da solução poderia ser substituída por uma implementação em Java para internet. Entretanto, para esta versão inicial, optou-se pelo Excel com o objetivo de reduzir o esforço total de implementação envolvido na solução.

3.2.2. Java

Java é uma linguagem de programação orientada a objetos desenvolvida na década de 1990 pela Sun, especificamente projetada para minimizar dependências de plataforma de execução. O Java utiliza o conceito de máquina virtual, onde há uma camada extra entre a aplicação e o sistema operacional responsável por traduzir o que a aplicação deseja realizar para as respectivas chamadas do sistema operacional em que a máquina virtual esteja rodando no momento. Com isso, há independência de plataforma tanto em relação ao sistema operacional como em relação ao hardware e suas configurações.

O Java Enterprise Edition Platform (Java EE) é uma plataforma de programação para servidores na linguagem Java. Trata-se de uma série de especificações bem detalhadas, com APIs como o JDBC para acesso a banco de dados, Servlets e JSP (Java Server Pages) para desenvolvimento de aplicações Web com conteúdo dinâmico, entre outras. O sistema utiliza o Java EE para implementação da aplicação Web que roda hospedada em um container Tomcat 7 da Apache Software Foundation,.

O Eclipse 3.8 para Java EE foi a IDE (Integrated Development Environment) utilizada no desenvolvimento. Ao ambiente padrão do Eclipse, foi adicionado o Apache Maven, um plugin para o Eclipse que fornece recursos de automação de compilação e gerenciamento de bibliotecas e plugins Java de um ou mais repositórios na internet.

O Java foi escolhido pela grande quantidade de frameworks e componentes que minimizam a quantidade de código a ser escrita pelo desenvolvedor, e também pelo vasto material de apoio existente sobre a linguagem e o ambiente de desenvolvimento. A implementação do sistema utilizou os seguintes frameworks e componentes: Apache Struts 2, DOM XML parser, Hibernate e JUnit.

O *Apache Struts 2* é um framework de desenvolvimento da camada controladora em uma estrutura *Model View Controller* (MVC) de aplicações Web. O *Struts* separa os elementos *Model*, responsáveis pela lógica do programa e interação com banco de dados dos elementos *View*, que correspondem às páginas HTML apresentadas para o cliente, e também dos elementos *Controller*, que gerenciam a produção das respostas aos eventos e mediam a execução dos elementos *Model* e *View*. No sistema, foi utilizado como controlador um *servlet* conhecido como *Action Servlet*, que transmite as informações processadas no *Model* às *Java Server Pages (JSPs)* responsáveis por produzir as páginas HTML exibidas para o usuário.

Para a leitura da planilha Excel em formato XML, foi escolhida a API Java *DOM XML parser*. DOM é uma interface de programação, baseada no modelo de objetos, que possibilita a manipulação de documentos em XML. Essa interface faz a representação do conteúdo XML na forma de uma árvore de objetos, onde cada nó da árvore representa um componente do documento. É um modelo flexível e não limitado pela ordem em que as estruturas do arquivo são processadas, tornando possível a navegação pelo documento em qualquer ordem. Este último aspecto foi preponderante para a escolha desta tecnologia para o processamento de arquivos em formato XML.

Na parte de acesso ao banco de dados relacional, foi utilizado o *Hibernate* (versão 4.1.6). *Hibernate* é um framework que faz o mapeamento entre os objetos modelo da aplicação e as tabelas de um banco de dados relacional. Esse mapeamento pode ser feito por meio de definições expressas em arquivos XML ou em anotações Java. O *Hibernate* gera os comandos SQL, liberando o desenvolvedor do trabalho manual de escrevê-las, além de manter o programa portátil para outros bancos de dados

SQL. Esse framework foi utilizado na camada de acesso a dados para realização de operações CRUD (*Create, Update, Read, Delete*) das informações armazenadas no banco de dados relacional.

O framework *JUnit* foi utilizado para realização de testes unitários automatizados. Este framework, amplamente utilizado pela comunidade de desenvolvedores Java, permite a implementação e execução automática de diferentes testes de rotinas do sistema. Esses testes diminuem os riscos e impactos de eventuais mudanças, pois qualquer mudança realizada pode ser avaliada em relação à preservação do correto funcionamento de todas as rotinas do sistema de forma automática e rápida.

3.2.3. Prolog

Prolog é uma linguagem de programação que surgiu nos anos 70, sendo utilizada por muitos programadores para o desenvolvimento de aplicações que envolvem bases de dados relacionais, lógica matemática, resolução de problemas abstratos, diversas áreas de inteligência artificial, entre outros. Na programação em Prolog, o desenvolvedor pergunta quais relacionamentos formais e objetos ocorrem no problema, e quais relações são verdadeiras, ou seja, satisfazem a solução desejada. A abordagem do Prolog é voltada para descrever fatos e relacionamentos dentro de um problema, ao invés de prescrever uma sequência de passos executados pelo computador para resolver um problema, como ocorre nas linguagens de programação imperativas. Assim sendo, o Prolog é uma linguagem de programação que é usada para resolver problemas que envolvem objetos e relacionamentos entre objetos (Clocksin & Mellish, 2003).

O sistema utiliza o Prolog por meio da implementação de código aberto *SWI-Prolog*. Como o problema da grade horária pode ser imaginado como um problema de objetos (ofertas de disciplina, disciplina, professores, preferência de horário, horários) e relacionamentos entre tais objetos (o produto final, por exemplo, são associações entre disciplinas e slots de horário), a utilização do Prolog é proveitosa. Além disso, o Prolog possibilita a solicitação de nova solução para o problema proposto, para casos em que a resposta anterior não seja satisfatória.

3.2.4. Bootstrap

Framework que agiliza e facilita o desenvolvimento dos componentes de interface com o usuário de para aplicações Web. É formado por um conjunto de

templates HTML, CSS e código Javascript que foram utilizados na construção do layout da aplicação.

3.2.5. Banco de Dados Relacional

Todas as informações utilizadas pelo gerador de grade horária são lidas de um banco de dados relacional. O Sistema Gerenciador de Banco de Dados (SGBD) utilizado para implementação da solução é o PostgreSQL. Este software é atualmente um dos SGBDs de código aberto mais avançados e utilizados no mundo, contando com recursos como consultas complexas, chaves estrangeiras, visões, gatilhos (*triggers*), controle de transações, entre outros, e por tais motivos foi utilizado na solução.

3.3. Modelo Lógico de Dados

Esta seção descreve o modelo lógico do banco de dados da aplicação, apresentando as tabelas, os relacionamentos entre as tabelas do banco de dados, as chaves primárias, chaves estrangeiras e os atributos de cada tabela. A Figura 12 apresenta o modelo de dados da aplicação.

As informações das entidades Período Curricular e Disciplina do modelo conceitual foram unificadas em uma tabela Disciplina no modelo lógico. A informação de número do período curricular é utilizada na geração grade horária para impedir que ofertas de disciplinas do mesmo período sejam alocadas em horários concorrentes, ou para alocar preferencialmente as ofertas de disciplinas do 1º ao 4º período em horários vespertinos e do 5º ao 8º período em horários noturnos. Como a informação do número de período está sempre vinculada à disciplina, decidiu-se por realizar a fusão em uma só tabela no modelo lógico. Também foi criada uma coluna id como identificador de uma disciplina no modelo lógico.

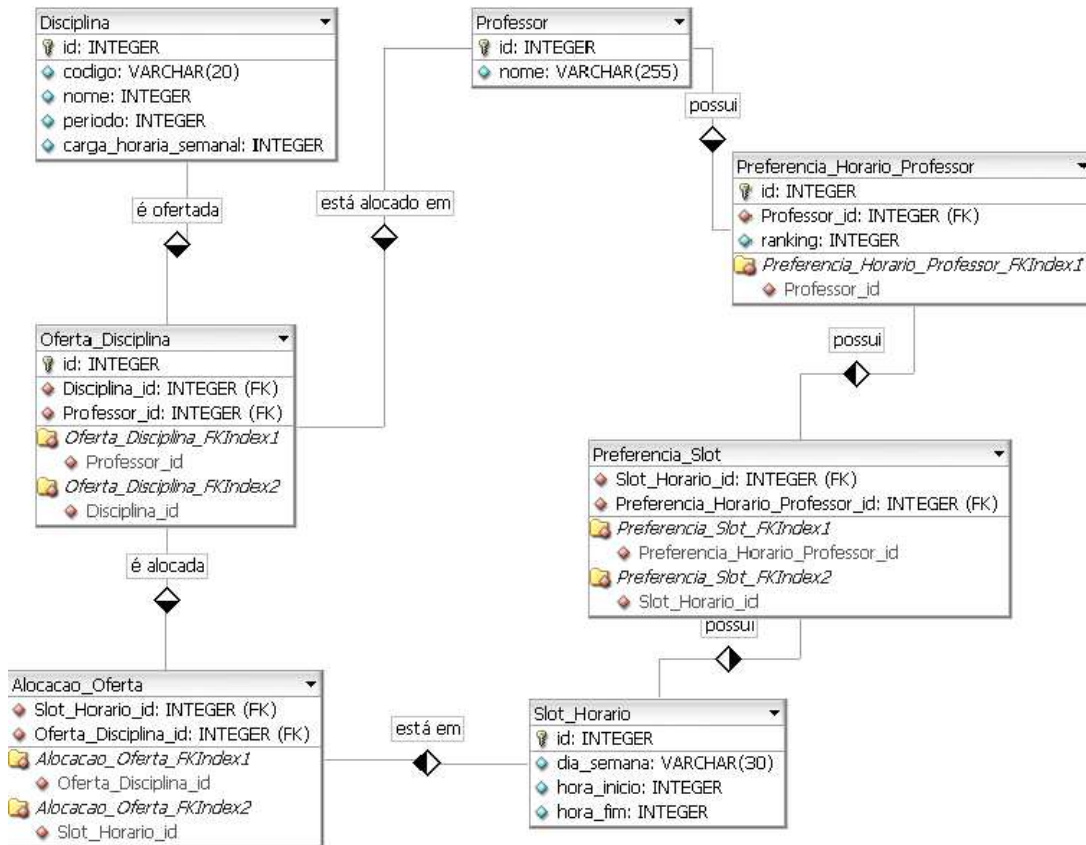


Figura 12 – Modelo Lógico de Dados

As informações da entidade Departamento e do seu relacionamento com a entidade Professor do modelo conceitual não foram representadas no modelo lógico, pois não foram utilizadas na geração da grade horária.

A entidade Professor do modelo conceitual foi mapeada para a tabela Professor no modelo lógico. Entretanto, apenas o nome do professor foi representado no modelo lógico. Além disso, foi criada uma coluna id como identificador de um professor no modelo lógico.

A entidade Oferta de Disciplina do modelo conceitual foi mapeada para a tabela Oferta_Disciplina no modelo lógico. Os relacionamentos da entidade Oferta de Disciplina com as entidades Disciplina e Professor deram origem às chaves estrangeiras disciplina_id e professor_id na tabela Oferta_Disciplina no modelo lógico. Já o relacionamento com a entidade Período Letivo não gera informação relevante para a geração da grade horária, pois nesta versão apenas os dados de um período letivo são armazenados no banco de dados. Também foi criada uma coluna id como identificador de uma oferta de disciplina no modelo lógico.

As informações da entidade SlotHorario deram origem a tabela Slot_Horario no modelo lógico. Contudo, as informações das entidades Curso e Período Letivo que, no Modelo Conceitual se relacionam com a entidade Grade Disponível, formada por slots de horário, não são representadas no modelo lógico, pois não foram utilizadas no processo de geração da grade. Além disso, foi criada uma coluna id como identificador de um slot de horário no modelo lógico.

A entidade PreferenciaHorario do modelo conceitual dá origem à tabela Preferência_Horario_Professor no modelo lógico. A chave estrangeira Professor_id vem do relacionamento com a entidade Professor e o atributo ranking é referente à ordem de preferência de horário do professor. O relacionamento muitos para muitos com a entidade SlotHorario gera uma tabela associativa Preferencia_Slot com referência a um slot de horário e uma preferência de um professor. As informações das entidades Curso e Período Letivo não são representadas no modelo lógico, pois não foram utilizadas no processo de geração da grade horária. Também foi criada uma coluna id como identificador de uma preferência de horário no modelo lógico.

O relacionamento da entidade Oferta de Disciplina com SlotHorario é de muitos para muitos e gera uma tabela associativa Alocação_Oferta no modelo lógico com chave estrangeira para as tabelas Oferta_Disciplina (Oferta_Disciplina_Id) e Slot_Horario (Slot_Horario_Id).

3.4. Módulo de Cadastros Básicos

O módulo de cadastros básicos para a geração da grade horária implementa os casos de uso *Manter Informações de Disciplinas*, *Manter Informações de Professores*, *Definir Ofertas de Disciplina do Período Letivo*, *Definir Grade Horária Disponível para o Período Letivo* e *Definir Preferências de Horário dos Professores para o Período Letivo*. Nesses casos de uso, o usuário preenche as planilhas Excel com os dados de disciplinas, professores, ofertas de disciplina, horários disponíveis e preferências de horário dos professores para o semestre. Cada caso de uso é implementado como uma tabela em uma aba separada da planilha. Cada planilha (arquivo XML) corresponde a um período letivo específico.

A Figura 13 apresenta a tabela para edição das disciplinas do curso onde o usuário define o código, o nome, o número do período curricular (entre 1 e 8) e a carga

horária da disciplina. A parte inferior da figura ilustra como esse trecho da planilha é salvo em XML.

CODIGO	NOME	PERIODO	CARGA
htd0058	TPD	1	4
tin0106	Des. Pag. Web	1	4

```

<Row>
  <Cell ss:StyleID="s62"><Data ss:Type="String">CODIGO</Data><NamedCell
    ss:Name="_FilterDatabase"/></Cell>
  <Cell ss:StyleID="s62"><Data ss:Type="String">NOME</Data><NamedCell
    ss:Name="_FilterDatabase"/></Cell>
  <Cell ss:StyleID="s62"><Data ss:Type="String">PERIODO</Data><NamedCell
    ss:Name="_FilterDatabase"/></Cell>
  <Cell ss:StyleID="s62"><Data ss:Type="String">CARGA</Data><NamedCell
    ss:Name="_FilterDatabase"/></Cell>
</Row>
<Row>
  <Cell><Data ss:Type="String">htd0058</Data><NamedCell ss:Name="_FilterDatabase"/></Cell>
  <Cell><Data ss:Type="String">TPD</Data><NamedCell ss:Name="NomesDisciplinas"/><NamedCell
    ss:Name="_FilterDatabase"/></Cell>
  <Cell><Data ss:Type="Number">1</Data><NamedCell ss:Name="_FilterDatabase"/></Cell>
  <Cell><Data ss:Type="Number">4</Data><NamedCell ss:Name="_FilterDatabase"/></Cell>
</Row>
<Row>
  <Cell><Data ss:Type="String">tin0106</Data><NamedCell ss:Name="_FilterDatabase"/></Cell>
  <Cell><Data ss:Type="String">Des. Pag. Web</Data><NamedCell
    ss:Name="NomesDisciplinas"/><NamedCell ss:Name="_FilterDatabase"/></Cell>
  <Cell><Data ss:Type="Number">1</Data><NamedCell ss:Name="_FilterDatabase"/></Cell>
  <Cell><Data ss:Type="Number">4</Data><NamedCell ss:Name="_FilterDatabase"/></Cell>
</Row>

```

Figura 13 – Planilha Excel: Cadastro de Disciplinas

A Figura 14 apresenta a tabela para edição dos nomes dos professores que serão alocados a ofertas de disciplinas do curso. A parte inferior da figura ilustra como esse trecho da planilha é salvo em XML.

NOME
Lucia Moutinho
Pimentel

```

<Row>
  <Cell ss:StyleID="s62"><Data ss:Type="String">NOME</Data><NamedCell
    ss:Name="_FilterDatabase"/></Cell>
</Row>
<Row>
  <Cell><Data ss:Type="String">Lucia Moutinho</Data><NamedCell
    ss:Name="_FilterDatabase"/><NamedCell ss:Name="NomesProfessores"/></Cell>
</Row>
<Row>
  <Cell><Data ss:Type="String">Pimentel</Data><NamedCell
    ss:Name="_FilterDatabase"/><NamedCell ss:Name="NomesProfessores"/></Cell>
</Row>

```

Figura 14 – Planilha Excel: Cadastro de Professores

A Figura 15 apresenta a tabela para edição das ofertas de disciplinas. O usuário seleciona, na coluna Disciplina, uma das disciplinas cadastradas na tabela de disciplinas e, na coluna Professor, um dos professores cadastrados na tabela de professores. A parte inferior da figura ilustra como esse trecho da planilha é salvo em XML.

	A	B	C
1	Disciplina	Professor	
2	TPD	Lucia Moutinho	
3	Des. Pag. Web	Pimentel	

```

<Row>
  <Cell ss:StyleID="s62"><Data ss:Type="String">Disciplina</Data><NamedCell
    ss:Name="_FilterDatabase"/></Cell>
  <Cell ss:StyleID="s62"><Data ss:Type="String">Professor</Data><NamedCell
    ss:Name="_FilterDatabase"/></Cell>
</Row>
<Row>
  <Cell><Data ss:Type="String">TPD</Data><NamedCell ss:Name="_FilterDatabase"/></Cell>
  <Cell><Data ss:Type="String">Lucia Moutinho</Data><NamedCell
    ss:Name="_FilterDatabase"/></Cell>
</Row>
<Row>
  <Cell><Data ss:Type="String">Des. Pag. Web</Data><NamedCell
    ss:Name="_FilterDatabase"/></Cell>
  <Cell><Data ss:Type="String">Pimentel</Data><NamedCell
    ss:Name="_FilterDatabase"/></Cell>
</Row>

```

Figura 15 – Planilha Excel: Cadastro de Ofertas de Disciplinas

A Figura 16 apresenta a tabela para edição dos slots de horário disponíveis para o período letivo. Cada slot é definido pelo dia da semana e por um intervalo de horário. A parte inferior da figura ilustra como esse trecho da planilha é salvo em XML.

	A	B
1	Slots	
2	Terça 16-18	
3	Quinta 18-20	
4	Sexta 14-16	
5	Sexta 16-18	

```

<Row>
  <Cell ss:StyleID="s62"><Data ss:Type="String">Slots</Data><NamedCell
  ss:Name="_FilterDatabase"/></Cell>
</Row>
<Row>
  <Cell><Data ss:Type="String">Terça 16-18</Data><NamedCell
  ss:Name="_FilterDatabase"/><NamedCell ss:Name="Slots"/></Cell>
</Row>
<Row>
  <Cell><Data ss:Type="String">Quinta 18-20</Data><NamedCell
  ss:Name="_FilterDatabase"/><NamedCell ss:Name="Slots"/></Cell>
</Row>
<Row>
  <Cell><Data ss:Type="String">Sexta 14-16</Data><NamedCell
  ss:Name="_FilterDatabase"/><NamedCell ss:Name="Slots"/></Cell>
</Row>
<Row>
  <Cell><Data ss:Type="String">Sexta 16-18</Data><NamedCell
  ss:Name="_FilterDatabase"/><NamedCell ss:Name="Slots"/></Cell>
</Row>

```

Figura 16 – Planilha Excel: Cadastro de Horários Disponíveis

A Figura 17 apresenta a tabela para edição das preferências de horário dos professores para o período letivo. Cada linha da planilha representa uma preferência de horário de um determinado professor. Cada preferência é definida por um conjunto de slots de horário correspondentes a colunas (Horário1, Horário2, etc) da mesma linha. No exemplo, a professora Lucia Moutinho tem preferência por alocação de suas ofertas às sextas-feiras de 14-16 e 16-18. Se ela tivesse outra opção de preferência, seria possível definir outra linha associada a essa professora com outros slots de horário (ex: segundas 16-18 e 18-20).

O professor e os horários são selecionados a partir dos elementos presentes na tabela de professores e de slots disponíveis para o período letivo. A parte inferior da figura ilustra como esse trecho da planilha é salvo em XML

	A	B	C
1	Professor	Horario1	Horario2
2	Lucia Moutinho	Sexta 14-16	Sexta 16-18
3	Pimentel	Terça 16-18	Quinta 18-20

```

<Row>
<Cell ss:StyleID="s62"><Data ss:Type="String">Professor</Data><NamedCell
  ss:Name="_FilterDatabase"/></Cell>
<Cell ss:StyleID="s62"><Data ss:Type="String">Horario1</Data><NamedCell
  ss:Name="_FilterDatabase"/></Cell>
<Cell ss:StyleID="s62"><Data ss:Type="String">Horario2</Data><NamedCell
  ss:Name="_FilterDatabase"/></Cell>
</Row>
<Row>
<Cell><Data ss:Type="String">Lucia Moutinho</Data><NamedCell
  ss:Name="_FilterDatabase"/></Cell>
<Cell><Data ss:Type="String">Sexta 14-16</Data><NamedCell
  ss:Name="_FilterDatabase"/></Cell>
<Cell><Data ss:Type="String">Sexta 16-18</Data><NamedCell
  ss:Name="_FilterDatabase"/></Cell>
</Row>
<Row>
<Cell><Data ss:Type="String">Pimentel</Data><NamedCell
  ss:Name="_FilterDatabase"/></Cell>
<Cell><Data ss:Type="String">Terça 16-18</Data><NamedCell
  ss:Name="_FilterDatabase"/></Cell>
<Cell><Data ss:Type="String">Quinta 18-20</Data><NamedCell
  ss:Name="_FilterDatabase"/></Cell>
</Row>

```

Figura 17 – Planilha Excel: Cadastro de Preferências de Horários dos Professores

3.5. Módulo de Carga dos Cadastros Básicos

O módulo de carga dos cadastros básicos realiza a transferência para o servidor do arquivo XML contendo as informações cadastradas no módulo de Cadastros Básicos, processa o arquivo transferido e gera registros correspondentes no banco de dados relacional.

Para realizar essa carga de dados, é necessário acessar a URL da aplicação, selecionar o menu ‘Cadastro de Dados’, escolher o arquivo XML e clicar no botão Enviar (Figura 18).

Arquivo para upload

Nenhum arquivo selecionado

© Diego Rafael Di Berardino - 2013

Figura 18 – Tela para Carga dos Cadastros Básicos

A primeira tarefa executada pelo módulo de carga dos cadastros básicos corresponde ao processamento do arquivo XML contendo a planilha Excel. Esse processamento é realizado por meio da API DOM, analisador XML disponível no Java. A interface DOM traduz um documento XML para uma representação na forma de uma árvore de objetos, onde cada nó representa um componente do documento. A Figura 19 apresenta a estrutura de nós gerados a partir da leitura de um arquivo Excel em XML. O nó raiz representa uma planilha (Worksheet) que contém uma lista de nós filhos, cada um destes representando uma linha (Row) da tabela. Cada nó do tipo Linha contém uma lista de nós filhos representando as células (Cell). Cada nó do tipo Célula, por sua vez, contém um nó filho do tipo Data correspondente ao valor presente na célula.

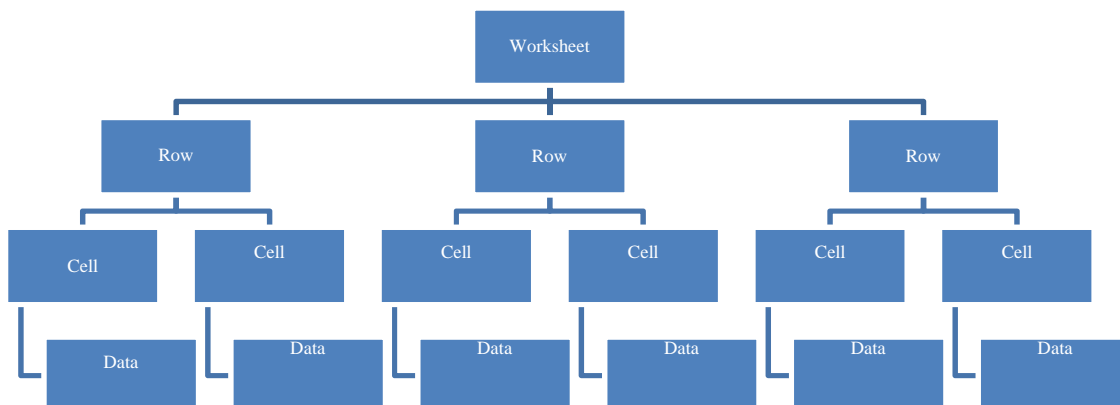


Figura 19 – Árvore de elementos da planilha Excel em XML

O processamento completo do documento XML gera uma árvore de objetos DOM, permitindo que a aplicação possa acessar qualquer item de informação através de operações de navegação pela árvore. Como a utilização do DOM XML requer a leitura de toda a estrutura XML para uma árvore em memória, o processamento de arquivos

XML grandes pode ser um problema. Contudo, esta não foi uma preocupação para o sistema de grade horária, uma vez que o documento XML utilizado na carga de dados do sistema é relativamente pequeno.

A segunda tarefa executada por este módulo consiste em navegar pela árvore DOM gerada a partir do documento XML e criar em memória instâncias de objetos correspondentes às informações presentes nos cadastros básicos, i.e., disciplinas, professores, ofertas de disciplinas, slots disponíveis e preferências de horários dos professores.

A última tarefa executada pelo módulo é gravar no banco de dados relacional as informações geradas pela tarefa anterior. Esta tarefa foi implementada utilizando o padrão de projeto DAO (*Data Access Object*), que permite a segregação dos aspectos de recuperação e armazenamento de informação em módulos especificamente criados para este fim. Neste módulo, para cada classe concreta (Disciplina, Professor, OfertaDisciplina, SlotHorario, PreferenciaHorario) foi implementada uma unidade DAO (DisciplinaDAO, ProfessorDAO, OfertaDisciplinaDAO, SlotHorarioDAO, PreferenciaHorarioDAO), como pode ser visto na Figura 20. Cada classe DAO fornece uma implementação específica para uma interface genérica DAO que define operações como: listar todas as instâncias da entidade referida, remover todas as instâncias da entidade, armazenar uma nova instância e recuperar uma instância por algum critério. A implementação dos serviços oferecidos pelas unidades DAO fez uso do *Hibernate*, framework Java que realiza o mapeamento objeto-relacional.

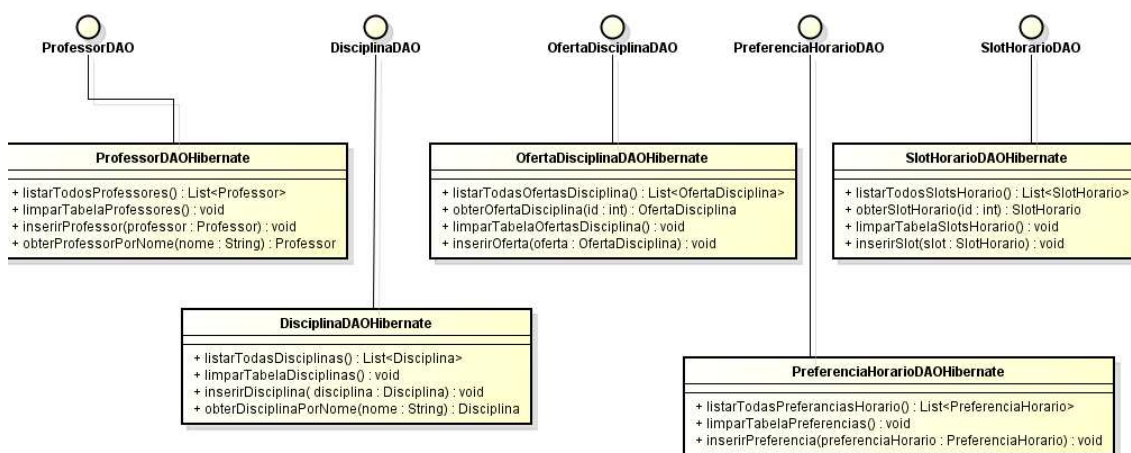


Figura 20 – Estrutura de classes de acesso a dados (padrão DAO)

3.6. Módulo Gerador da Grade Horária

Este módulo implementa o caso de uso *Gerar grade horária para o período letivo*. Para disparar este caso de uso, é necessário acessar a URL da aplicação e selecionar a opção *Gerar Grade Horária*. Este módulo foi implementado parte em Java (interface com usuário) e parte em Prolog (lógica de geração da grade horária).

Ao receber a solicitação de geração de grade horária, o componente Java deste módulo recupera as informações de professores, disciplinas, ofertas de disciplinas, slots de horário e preferências de horário de professor armazenadas no banco de dados relacional e gera um arquivo de fatos Prolog (arquivo base.pl).

A Figura 21 apresenta um trecho do arquivo de fatos gerados correspondentes aos registros de professores, disciplinas e ofertas de disciplinas presentes no banco de dados. A Figura 22 apresenta exemplos de fatos correspondentes à definição dos slots de horários disponíveis para a geração da grade e às preferências de horário.

```
%% professor(id,nome).
professor(124,'Lucia Moutinho').
professor(128,'Pimentel').
%% disciplina(id,codigo,nome,periodo,carga_horaria_semanal).
disciplina(136,'htd0058','TPD',1,4).
disciplina(137,'tin0106','Des. Pag. Web',1,4).
%% oferta(id, id_disciplina, id_professor).
oferta(146,136,124).
oferta(151,137,128).
```

Figura 21 – Predicados correspondentes a Professor, Disciplina e Oferta.

```
%% slotHorario(id,dia,hora_inicio,hora_fim).
slotHorario(12,'Terça',16,18).
slotHorario(19,'Quinta',18,20).
slotHorario(10,'Sexta',14,16).
slotHorario(15,'Sexta',16,18).
%% preferencia(id,id_professor,[id_slots],ranking)
preferencia(123,124,[10,15],1).
preferencia(127,128,[12,19],1).
```

Figura 22 – Predicados referentes a slot e preferências de horários

Integração Java-Prolog

O arquivo de fatos Prolog contém as informações que serão utilizadas no processo de geração de uma possível solução para o problema da grade horária. As restrições que uma possível grade horária para o período deve satisfazer foram implementadas por meio de regras em um programa Prolog (grade.pl). Desta forma, uma vez gerado o arquivo de fatos Prolog (base.pl), o componente Java ativa a execução do programa Prolog (grade.pl) por meio do software SWI-Prolog, captura a saída gerada, formata o resultado e o apresenta para o usuário.

A Figura 23 apresenta o trecho de código Java onde é feita a invocação do programa Prolog que gerará a grade horária. A linha em destaque contém a chamada do método *exec* da classe *Runtime* do Java, cujo parâmetro é o comando a ser executado que, neste caso, é o caminho de instalação do *SWI-Prolog*, concatenado a *-q* (modo “quiet” - silencioso), *-s* (indicando o arquivo de script Prolog que será executado), *-g* (*goal*) ‘executa’(nome do goal a ser satisfeito) *-t halt* (para finalizar a execução do *SWI-Prolog*).

```
public String obterSaidaArquivoProlog(){
    Runtime rt = Runtime.getRuntime();
    Process proc;
    String capturadorProlog = "";
    try {
        proc = rt.exec(caminhoInstalacaoProlog + " -q -s " + caminhoArquivo + " -g 'executa' -t halt");
        InputStream stdin = proc.getInputStream();
        InputStreamReader isr = new InputStreamReader(stdin);
        BufferedReader saidaArquivo = new BufferedReader(isr);
        String line = null;

        while ((line = saidaArquivo.readLine()) != null){
            capturadorProlog += line;
        }

    } catch (IOException e) {
        e.printStackTrace();
    }
    return capturadorProlog;
}
```

Figura 23 – Código Java para integração com Prolog

A Figura 24 apresenta um exemplo de resultado gerado pelo programa Prolog, que consiste de uma lista de pares (id_oferta_disciplina, [lista_ids_slots_horario]), contendo o identificador da oferta de disciplina e a lista de slots de horário alocados para a mesma.

```

2 ?- executa.
[ (1,[6,8,1,3]), (2,[11,13]), (3,[7,9]), (4,[17,19]), (5,[14]), (6,[15,10])][ (7,[20,16,18]), (8,[7,
9]), (9,[22,24]), (10,[17,19]), (11,[15,13])][ (15,[12,14]), (12,[6,11]), (13,[16,18]), (14,[7,9]),
(16,[17,19]), (17,[8,13])][ (18,[11,13]), (19,[16,18]), (20,[21,23]), (21,[12,14]), (22,[17,19]), (2
3,[22,24])][ (28,[12,14]), (27,[17,19]), (24,[11,13]), (25,[16,18]), (26,[21,23])][ (30,[21,23]), (2
9,[16,18]), (31,[17,19]), (32,[22,24]), (33,[20,25])][ (34,[16,18]), (35,[21,23]), (36,[17,19])][ (4
0,[22,24]), (38,[21,23]), (37,[16,18]), (39,[17,19])]]
true

```

Figura 24 – Exemplo de saída gerada pelo programa Prolog

Após obter a saída gerada pelo SWI-Prolog, o sistema necessita transformá-la em informações inteligíveis para o usuário. Para isso, o componente Java deste módulo gera uma página HTML a partir do resultado gerado pelo programa Prolog, recuperando por meio da camada de acesso a dados as informações completas das ofertas e slots de horário da grade gerada. A Figura 25 apresenta um exemplo de resultado gerado pelo programa.

Hora	Segunda	Terça	Quarta	Quinta	Sexta
13:00 às 14:00	TP1 Geiza		TP1 Geiza		
14:00 às 15:00	TP1 Geiza	Org. de Computadores Morganna	TP1 Geiza	Org. de Computadores Morganna	TPD Lucia Moutinho
15:00 às 16:00	TP1 Geiza	Org. de Computadores Morganna	TP1 Geiza	Org. de Computadores Morganna	TPD Lucia Moutinho
16:00 às 17:00	Fundamentos SI Sean	Des. Pag. Web Pimentel	Fundamentos SI Sean	Mat Bas Prof Mat Bas	TPD Lucia Moutinho
17:00 às 18:00	Fundamentos SI Sean	Des. Pag. Web Pimentel	Fundamentos SI Sean	Mat Bas Prof Mat Bas	TPD Lucia Moutinho
18:00 às 19:00				Des. Pag. Web Pimentel	
19:00 às 20:00				Des. Pag. Web Pimentel	
20:00 às 21:00					
21:00 às 22:00					

Figura 25 – Exemplo de saída gerada pelo programa Prolog

4. CONCLUSÃO

4.1. Contribuições

Este trabalho propõe uma solução para o problema da geração da grade horária com base em uma integração entre diversas tecnologias. A utilização da solução proposta ameniza o esforço, reduz o tempo necessário para confecção da grade e diminui a suscetibilidade a erros.

4.2. Limitações

O trabalho apresenta limitações, dentre as quais podemos destacar:

- O sistema não trata todas as restrições utilizadas atualmente para a geração da grade horária. Como exemplos de restrições não tratadas pela solução atual, podemos destacar: não pode haver presença de slots vagos entre slots ocupados em um dia da semana para um mesmo período, fenômeno conhecido como existência de buracos na grade; disciplina A que seja pré-requisito de uma disciplina B implica que A e B sejam ofertadas preferencialmente nos mesmos slots de horário.
- O sistema não permite o relaxamento seletivo de restrições fracas, caso o programa não consiga encontrar uma grade que satisfaça as restrições.
- O sistema não permite que o usuário realize mudanças manuais na grade gerada, caso o quadro de horários gerado pelo sistema não seja do seu agrado;
- O sistema não fornece uma funcionalidade de exportação da grade gerada para Excel.

4.3. Trabalhos Futuros

Algumas possibilidades para evolução do estágio alcançado por este trabalho são listadas nesta seção.

- A implementação de um sistema Web com controle de acesso e de funcionalidades, no qual os próprios professores possam cadastrar e alterar suas preferências de horário, substituindo as atuais planilhas para cadastramento dos dados necessários para a geração da grade horária;
- A implementação de uma funcionalidade que permita o usuário modificar uma grade horária gerada, movimentando ofertas de disciplina dinamicamente entre os horários até chegar a uma grade que atenda melhor às suas necessidades. Além disso, para cada mudança realizada pelo usuário, o sistema interagiria indicando quais as consequências geradas.
- Evolução das restrições consideradas e possibilidade de relaxamento das restrições fracas de forma manual ou automática, seguindo uma ordem de prioridade;
- A implementação de uma funcionalidade que gere um custo por violação de restrições fracas em uma grade de horários gerada.

4.4. Considerações Finais

Em relação à proposta de uma solução para o problema da grade horária, pode-se concluir que o resultado final é satisfatório, embora existam diversos pontos de evolução necessários para que sua utilização possa ser efetiva. Entretanto, o desenvolvimento deste trabalho possibilitou a sedimentação de diversos conhecimentos tanto na área de especificação como de projeto de sistemas. Além disso, a utilização de diferentes paradigmas de solução, tais como Excel, Java e Prolog, trouxe desafios não só dentro do próprio paradigma como também de integração entre eles, possibilitando uma intensa aquisição de conhecimento em arquitetura e projeto de soluções de software.

Referências Bibliográficas

APACHE STRUTS 2. < <http://struts.apache.org/development/2.x/>>. Acesso em 10 jun.2013

BOOTSTRAP < <http://getbootstrap.com/>>. Acesso em 19 ago.2013

CLOCKSIN,W.F.; MELLISH,C.S. – **Programming in Prolog**. Fifth Edition, Springer-Verlag, 2003, 298 p.

ENGIEL,P. – **Habilitando Processos de Prestação de Serviços à Participação e à Democracia – O caso da Escola de Informática Aplicada/UNIRIO**. 2009. 92 f. Monografia – Bacharelado em Sistemas de Informação, Universidade Federal do Estado do Rio de Janeiro .

FERNANDES,C.; CALDEIRA,J.P.; MELÍCIO,F.; ROSA,A. – **Infected Genes Evolutionary Algorithm for School Timetabling**. WSES International Conference, 2002 , p.1.

HAMAWAKI,C.D.L. – **Geração Automática de Grade Horária Usando Algoritmos Genéticos: O Caso da Faculdade de Engenharia Elétrica da UFU**. 2005. 104 f. Dissertação – Pós-Graduação em Engenharia Elétrica, Faculdade de Engenharia Elétrica, Universidade Federal de Uberlândia.

HIBERNATE. < <http://www.hibernate.org/>>. Acesso em 10 jun.2013

STERLING,L.; SHAPIRO,E. – **The Art of Prolog: Advanced Programming Techniques**. Second Edition, Cambridge: The MIT Press, 1994, 509 p.

UNIRIOTEC. < <http://www.uniriotec.br/>>. Acesso em 06 mar.2013

WREN,A. – **Scheduling, timetabling and rostering – a special relationship?** . 1st Internacional Conference, Lecture Notes in Computer Science. Berlin, 1996, notas.

Anexo A – Documento de Visão

1. Objetivo

O objetivo deste documento de visão é listar os envolvidos no processo de geração da grade horária do curso de Bacharelado de Sistemas de Informação (BSI) da UNIRIO, bem como descrever os problemas que ocorrem nesse processo e propor soluções para os mesmos.

2. Visão Geral do Problema

O curso de BSI é oferecido semestralmente em turnos tarde/noite de segunda à sexta-feira. Atualmente, a grade curricular do curso é composta por oito períodos, sendo os quatro iniciais predominantemente vespertinos (14h às 18h) e os quatro finais predominantemente noturnos (18h às 22h). Cada período agrupa um conjunto de disciplinas que são classificadas em obrigatórias, optativas e eletivas. As disciplinas obrigatórias devem ser cursadas por todo aluno que deseje concluir o curso. Além das disciplinas obrigatórias, o discente deve cursar oito disciplinas optativas (disciplinas do curso que podem ser escolhidas de acordo com a preferência do aluno) e quatro disciplinas eletivas (disciplinas de outros cursos, também de livre escolha). Cada disciplina pode ter nenhuma, uma ou mais de uma oferta em um semestre. Em geral, as disciplinas obrigatórias são ofertadas semestralmente, o que não ocorre necessariamente com as disciplinas optativas.

O processo de definição da grade horária começa cerca de dois meses antes do início de um novo período letivo. Cada área de conhecimento da EIA (e.g., Redes, Banco de Dados, Engenharia de Software, Algoritmos/RCR) decide internamente as disciplinas que serão ofertadas e os respectivos professores alocados, obtém os horários de preferência de cada professor da área e, encaminha, através de um docente responsável, tais informações à direção da EIA. Processo similar ocorre em outros departamentos que oferecem disciplinas do curso (e.g., Departamento de Matemática e Estatística (DME)). A partir da definição das ofertas de disciplinas e das preferências de alocação de horário de cada professor, a direção é encarregada de gerar a grade horária

do período, tentando satisfazer as preferências de professores e alunos da melhor forma possível. Essa atividade é iterativa, pois pode haver a necessidade de negociações entre a direção e alguns professores para que algumas alocações sejam feitas em horários diferentes daqueles inicialmente sugeridos pelos professores de forma a satisfazer restrições e orientações da direção.

A grande quantidade de restrições de horários das disciplinas e a dificuldade de satisfazer todas as preferências de horários dos professores das diferentes linhas e departamentos fazem com que a obtenção da grade horária final para o semestre seja uma tarefa complexa. Atualmente esse trabalho é realizado de forma manual, resultando em uma demanda concentrada de esforço e comunicação. Além disso, a suscetibilidade a erros é inevitável, podendo gerar insatisfação dos interessados, i.e, professores e alunos.

3. Envolvidos no Processo de Geração da Grade Horária

Envolvidos	Papel no Processo
Professores	Definem quem ministrará as disciplinas e estabelecem horários de preferência de cada um.
Coordenação do Curso	Recebe a definição de alocação dos professores nas disciplinas e suas preferências de horários, monta a grade horária para o semestre e a divulga para professores e alunos.
Alunos	São afetados durante todo o período letivo pela grade horária das disciplinas que cursam.

4. Problemas Observados no Processo

- Não há integração entre as diferentes linhas dentro do DIA durante o processo. Com isso, os professores não tem acesso à preferência de horários de professores de outras linhas, aumentando o conflito de horários entre eles.
- Não há integração entre os diferentes departamentos que fazem parte do curso durante o processo, também aumentando o conflito de horários.
- A comunicação entre professores e direção ocorre por email, acarretando em desorganização e possível perda de informações referentes às reivindicações e restrições sobre a grade horária.
- Não há registros de documentação sobre os critérios que são utilizados para priorizar uma disciplina sobre outra em caso de choques de horários.
- A montagem da grade horária é um processo extremamente complexo, com inúmeras restrições. Confeccionar essa grade é uma tarefa árdua, estressante e que toma muito tempo.
- Pode haver insatisfação de professores, nos casos em que suas disciplinas não sejam alocadas em horários de sua preferência.

5. Soluções

Um sistema que tenha as seguintes funcionalidades:

- Manter informações das disciplinas do curso;
- Manter informações de professores do curso;
- Cadastrar e manter as ofertas de disciplina do período letivo;
- Cadastrar e manter horários disponíveis para o período letivo;
- Definição de preferências de horário dos professores para o período letivo;
- Efetuar carga de dados;
- Geração da grade horária para o período letivo.

Anexo B – Especificação dos Casos de Uso (Concretos)

CASO DE USO 1 – MANTER INFORMAÇÕES DE DISCIPLINAS

Objetivo:	Manutenção das informações das disciplinas do curso BSI.
Ator(es):	Coordenador do Curso
Início:	O caso de uso é iniciado quando o ator abre a planilha Excel de carga de dados para geração da grade horária do curso e clica na aba 'Disciplina'.
Pré-condição (ões):	Nenhuma.
Fluxo Principal	
No	Ações
1	A planilha apresenta as colunas 'Código', 'Nome', 'Período' e 'Carga' referindo-se às informações sobre disciplinas necessárias para geração da grade horária.
2	O ator preenche, altera ou exclui uma linha completando as colunas 'Código', 'Nome', 'Período' e 'Carga', repetindo o processo para cada disciplina que deseja manter .
3	O ator salva a planilha no formato 'XML'.
Pós-Condição (ões):	Informações das disciplinas mantidas com sucesso.

CASO DE USO 2 – MANTER INFORMAÇÕES DE PROFESSORES

Objetivo:	Manutenção das informações dos professores do curso BSI.
Ator(es):	Coordenador do Curso
Início:	O caso de uso é iniciado quando o ator abre a planilha Excel de carga de dados para geração da grade horária do curso e clica na aba 'Professor'.
Pré-condição (ões):	Nenhuma.
Fluxo Principal	
No	Ações
1	A planilha apresenta a coluna 'Nome' referindo-se às informações sobre professores necessárias para geração da grade horária.
2	O ator preenche, altera ou exclui uma linha completando a coluna 'Nome', repetindo o processo para cada professor que deseje manter.
3	O ator salva a planilha no formato 'XML'.
Pós-Condição (ões):	Informações dos professores mantidas com sucesso.

CASO DE USO 3 – DEFINIR OFERTAS DE DISCIPLINA DO PERÍODO LETIVO

Objetivo:	Definição das informações das ofertas de disciplina para o próximo período letivo do curso BSI.
Ator(es):	Coordenador do Curso
Início:	O caso de uso é iniciado quando o ator abre a planilha Excel de carga de dados para geração da grade horária do curso e clica na aba 'Oferta'.
Pré-condição (ões):	Informações de Professores e Disciplinas cadastradas.
Fluxo Principal	
No	Ações
1	A planilha apresenta as colunas 'Disciplina' e 'Professor' referindo-se às informações sobre ofertas de disciplina necessárias para geração da grade horária. Na primeira coluna, a planilha disponibiliza uma combobox com os nomes de disciplina preenchidos no caso de uso 1 como possibilidades de escolha. Na segunda coluna, a planilha disponibiliza uma combobox com os nomes de professor preenchidos no caso de uso 2 como possibilidades de escolha.
2	O ator preenche, altera ou exclui uma linha completando as colunas 'Disciplina e 'Professor', repetindo o processo para cada oferta de disciplina que deseja manter .
3	O ator salva a planilha no formato 'XML'.
Pós-Condição (ões):	Ofertas de disciplina do período letivo definidas com sucesso.

CASO DE USO 4 – DEFINIR GRADE HORÁRIA DISPONÍVEL PARA O PERÍODO LETIVO

Objetivo:	Definição dos horários disponíveis para o próximo período letivo do curso BSI.
Ator(es):	Coordenação do Curso
Início:	O caso de uso é iniciado quando o ator abre a planilha Excel de carga de dados para geração da grade horária do curso e clica na aba 'Slots'.
Pré-condição (ões):	Nenhuma.
Fluxo Principal	
No	Ações
1	A planilha apresenta a coluna 'Slots' referindo-se às informações sobre horários disponíveis para o período letivo necessárias para geração da grade horária.
2	O ator preenche, altera ou exclui uma linha completando a coluna 'Slots', repetindo o processo para cada horário disponível que deseja manter. O formato de preenchimento deve obedecer ao formato 'dia' 'hora inicial'– 'hora final' (Ex: Segunda 13-14, significando segunda-feira de 13 às 14 horas).
3	O ator salva a planilha no formato 'XML'.
Pós-Condição (ões):	Grade horária disponível para o período letivo definida com sucesso.

CASO DE USO 5 – DEFINIR PREFERÊNCIAS DE HORÁRIO DOS PROFESSORES PARA O PERÍODO LETIVO

Objetivo:	Definição das preferências de horário dos professores para o próximo período letivo do curso BSI.
Ator(es):	Coordenador do Curso
Início:	O caso de uso é iniciado quando o ator abre a planilha Excel de carga de dados para geração da grade horária do curso e clica na aba 'Preferencia'.
Pré-condição (ões):	Informações sobre Professores e Slots de Horário Disponíveis cadastradas.
Fluxo Principal	
No	Ações
1	A planilha apresenta as colunas 'Professor', 'Horario1', 'Horario2',..., 'Horario10' referindo-se às informações sobre preferências de horário de professor necessárias para geração da grade horária. Na primeira coluna, a planilha disponibiliza uma combobox com os nomes de professor preenchidos no caso de uso 2 como possibilidades de escolha. Nas demais colunas, a planilha disponibiliza uma combobox com os horários (slots) preenchidos no caso de uso 4 como possibilidades de escolha.
2	O ator preenche, altera ou exclui uma linha as colunas 'Professor', 'Horario1', 'Horario2',..., 'Horario10', repetindo o processo para cada preferência de horário de professor que deseja manter.
3	O ator salva a planilha no formato 'XML'.
Pós-Condição (ões):	Preferências de horário dos professores para o período letivo definidas com sucesso.

CASO DE USO 6 – GERAR GRADE HORÁRIA PARA O PERÍODO LETIVO

Objetivo:	Geração da grade horária do curso BSI para o período letivo.	
Ator(es):	Coordenador do Curso	
Início:	O caso de uso é iniciado quando o ator acessa a aplicação de Geração Automática da Grade Horária.	
Pré-condição (ões):	Nenhuma	
Fluxo Principal		
No	Ações	
1	O sistema apresenta tela com o campo 'Arquivo de Carga de Dados' e botão 'Enviar'.	
2	O ator anexa no campo 'Arquivo de Carga de Dados' arquivo XML com os dados preenchido nos casos de uso 1,2,3,4 e 5.	
3	O ator clica no botão 'Enviar'.	
4	O sistema exibe mensagem de êxito 'Carga de dados efetuada'.	
5	O ator seleciona o link 'Gerar Grade Horária'	
6	O sistema apresenta grade horária gerada com abas para seleção dos períodos.	
Fluxo(s) Alternativo(s)		
Fluxo Alternativo:	1	Tentativa de efetuar carga de dados com o campo 'Arquivo de Carga de Dados' não preenchido
Passo:	2	do Fluxo Principal
No	Ações	

1	O ator clica no botão 'Enviar'.		
2	O sistema exibe mensagem de validação: 'O arquivo de carga de dados deve ser anexado.'		
Fluxo Alternativo:	2	Tentativa de efetuar carga de dados com o arquivo anexo em formato diferente ao XML.	
Passo:	2	do Fluxo	Principal
No	Ações		
1	O ator anexa no campo 'Arquivo de Carga de Dados' arquivo em formato diferente ao XML.		
2	O ator clica no botão 'Enviar'.		
3	O sistema exibe mensagem de validação: 'O arquivo de carga de dados deve estar no formato XML.'		
Fluxo(s) de Exceção (ões)			
Fluxo de Exceção:	1	Caso ocorra algum problema na carga de dados, o sistema apresenta uma mensagem de erro: "Não foi possível efetuar a carga de dados".	
Passo:	4	do Fluxo	Principal
Fluxo de Exceção:	2	Caso ocorra algum problema na geração da grade horária, o sistema apresenta uma mensagem de erro: "Não foi possível gerar a grade horária".	
Passo:	6	do Fluxo	Principal
Pós-Condição (ões):	Grade Horária do Curso gerada com sucesso.		