



**UNIVERSIDADE DE ÉVORA**

**ESCOLA DE CIÊNCIAS E TECNOLOGIA**

**DEPARTAMENTO DE INFORMÁTICA**

**Geração automática de horários para os  
cursos da UE**

**Nuno Alexandre Batista de Matos**

Orientação: Irene Pimenta Rodrigues

**Mestrado em Engenharia Informática**

Dissertação

Évora, 2013





**UNIVERSIDADE DE ÉVORA**

**ESCOLA DE CIÊNCIAS E TECNOLOGIA**

**DEPARTAMENTO DE INFORMÁTICA**

**Geração automática de horários para os  
cursos da UE**

**Nuno Alexandre Batista de Matos**

Orientação: Irene Pimenta Rodrigues

**Mestrado em Engenharia Informática**

Dissertação

Évora, 2013



# Sumário

A geração de horários é um problema complexo que consome tempo e recursos às instituições. Os mecanismos de geração automática permitem tornar este processo mais célere, mais eficiente e menos trabalhoso.

O Sistema denominado SGAHUE - Sistema de Geração Automática de Horário da Universidade de Évora, foi desenvolvido para a Universidade de Évora com o objectivo de articular melhor os seus recursos (salas, docentes, etc.). Nesta dissertação apresenta-se o sistema desenvolvido, as suas funcionalidades e as ferramentas e técnicas usadas na sua implementação.

O sistema é composto por três pilares fundamentais o modelo de dados, a interface e o gerador automático. A estratégia para gerar horários passou por definir um conjunto de restrições associadas aos horários, algumas são consideradas fortes ou invioláveis outras fracas, preferências.



*Automatic timetabling generator for  
the courses of UE*

## **Abstract**

Timetabling generation is a complex problem that consumes institution time and resources. Automatic timetabling allows an ease in the usage of such resources. The system, named SGAHUE - Sistema de Geração Automática de Horários da Universidade Évora - 'Automatic Timetabling generator for University of Évora', was developed for the University of Évora with the purpose of improving resource use such as rooms and lecturers. In this dissertation, we describe the system development and implementation, presenting some of its features. The automatic timetabling implementation was implemented with constraint logic programming, using a set of constraint hard constraint and soft ones. The system is supported on a Relational database, and a web interface for user interactivity.





# Agradecimentos

Quero agradecer a minha orientadora Irene Pimenta Rodrigues todo o apoio que me deu no desenvolvimento do projecto desta dissertação, gostava de expressar o meu apreço por todos os bons conselhos e orientações que me deu tanto para o trabalho como para a vida. Agradeço também toda a disponibilidade e paciência que teve comigo.

Queria a agradecer á minha família a oportunidade de poder ter aqui chegado, em especial á minha mãe Natividade pelo apoio e força que me deu ao longo da minha vida.



# Acrónimos

**SGAHUE** Sistema de Geração automática da Universidade de Évora

**MDS** Máximo de dias por Semana

**MHD** Máximo de Horas por Dia

**DISP** Disponibilidade

**NHMD** Número de Horas Máximas por Dia

**UE** Universidade de Évora

**SGBD** Sistema de Gestão de Bases de Dados

**SIIE** Sistema Integrado de Informação da Universidade de Évora

**API** Application Programming Interface - 'Interface de Programação de Aplicações'

**DAO** Data Access Object - 'Objecto de Acesso aos Dados'



# Conteúdo

<b>Sumário</b>	<b>i</b>
<b>Abstract</b>	<b>iii</b>
<b>Lista de Conteúdo</b>	<b>x</b>
<b>Lista de Figuras</b>	<b>xii</b>
<b>1 Introdução</b>	<b>1</b>
<b>2 Estado da Arte</b>	<b>5</b>
2.1 Técnicas para Geração de horários . . . . .	5
2.2 Sistemas de Geração de Horários . . . . .	8
2.3 Ferramentas Usadas . . . . .	11
2.3.1 YII Framework . . . . .	11
2.3.2 GNU Prolog/ISCO . . . . .	13
<b>3 Sistema de Gestão de Horários</b>	<b>15</b>
3.1 Conceitos . . . . .	15
3.2 Preferências . . . . .	17
3.2.1 Docente . . . . .	18
3.2.2 Curso . . . . .	18
3.2.3 Aula . . . . .	19
3.3 Descrição da funcionalidade . . . . .	19
3.3.1 Edição . . . . .	19
3.3.2 Consistência . . . . .	20
3.3.3 Geração . . . . .	20

3.3.4	Visualizar / Editar . . . . .	20
3.4	Arquitetura do Sistema . . . . .	21
3.5	Interface . . . . .	21
3.5.1	Edição . . . . .	22
3.5.2	Visualização . . . . .	25
<b>4</b>	<b>Implementação</b>	<b>27</b>
4.1	Representação dos dados . . . . .	28
4.1.1	Aulas . . . . .	29
4.1.2	Horário das aulas . . . . .	29
4.2	Funções e Triggers . . . . .	30
4.3	Geração . . . . .	32
<b>5</b>	<b>Conclusões</b>	<b>37</b>
	<b>Referências bibliográficas</b>	<b>39</b>
	<b>Modelo ER da Base de dados</b>	<b>43</b>
	<b>Partes do Modelo ER</b>	<b>45</b>
	<b>Horário</b>	<b>49</b>
	<b>Edição da DSD</b>	<b>51</b>
	<b>Preferências do Docente</b>	<b>53</b>
	<b>Preferências do Curso</b>	<b>57</b>
	<b>Salas</b>	<b>59</b>
	<b>Visualização</b>	<b>61</b>

# Lista de Figuras

3.1	Arquitetura do Sistema . . . . .	21
3.2	Página inicial da interface . . . . .	22
3.3	Porção da interface - Indicação das horas e dias máximos do docente . . . . .	23
3.4	Porção da interface - Indicação da disponibilidade diária do docente . . . . .	23
3.5	Porção da interface - Indicação das turmas de um curso/ano/semestre . . . . .	24
3.6	Porção da interface - Horas máximas preferência manhã, tarde, turno . . . . .	24
3.7	Porção da interface - Preferências salas, tipos de salas nas aulas . . . . .	25
3.8	Porção da interface - Edição das aulas no Horário . . . . .	25
3.9	Porção da interface - Opções de Visualização do horário . . . . .	26
4.1	Horário de um docente . . . . .	35
4.2	Horario de um docente . . . . .	36
1	Modelo ER da Base de dados . . . . .	44
2	Representação do curso e preferências . . . . .	46
3	Representação da aula . . . . .	46
4	Representação do horário das aulas . . . . .	47
5	Preferências das aulas por salas . . . . .	47
6	Preferências do curso/ ano /sem . . . . .	48
7	Preferências do curso/ ano /sem por salas . . . . .	48
8	Horário gerado e exportado pelo sistema . . . . .	50
9	Edição da DSD . . . . .	52

10	Preferências do Docente . . . . .	54
11	Diagrama das preferências do docente . . . . .	55
12	Preferências horária do curso . . . . .	58
13	Preferências de Tipos de Salas do Curso . . . . .	58
14	Reserva de uma sala . . . . .	60
15	Preferência de Salas na aula . . . . .	60
16	Visualização / edição do Horário . . . . .	62



# Capítulo 1

## Introdução

A geração de horários é um problema que todos os anos tem de ser resolvido por qualquer estabelecimento de ensino, afim de fazer uma boa gestão dos seus recursos.

Este problema, gerar um horário, de uma forma geral consiste em atribuir uma sala e um tempo a cada aula. Um horário é considerado válido quando certas regras são cumpridas nomeadamente um docente não pode ter duas aulas ao mesmo tempo, um curso /turma não pode ter duas aulas ao mesmo tempo e uma sala não pode ter duas aulas ao mesmo tempo.

Uma boa gestão de recursos é o objectivo de qualquer instituição, pois quanto melhor usar esses recursos mais eficaz e eficiente será. Neste trabalho procurou-se desenvolver ferramentas que permitam otimizar ou encontrar soluções para horários que mesmo que não sejam óptimas sejam eficazes.

Para encontrar uma solução boa de horários temos que ter em conta o conjunto de condições que devem ser satisfeitas, algumas são obrigatórias, fortes, outras podem ser encaradas como preferências fracas. Neste caso procurar e encontrar a solução do problema passa por definir as restrições fortes e fracas a que o sistema deve de dar resposta através de um conjunto de ferramentas e funcionalidades que permitem retirar alguma complexidade.

Neste trabalho procura-se resolver os seguintes desafios:

- 1) Desenvolver um modelo de dados que represente aulas.
- 2) Criar uma interface web que seja natural para o utilizador.
- 3) Definir restrições sobre os dados.

- 4) Definir as restrições para a gerar de horários.

Universidade de Évora enfrenta o desafio de fazer horários todos os anos. Aparentando ser um caso igual ao de tantas universidades, na verdade, apresenta um desafio, fisicamente a universidade encontra-se dispersa pela cidade, existindo alguns cursos que usam recursos (salas) que se encontram espalhados pela cidade, logo há a necessidade de garantir que as pessoas se conseguem deslocar até ao local a tempo de leccionar (docentes) ou assistir às aulas (alunos). Devido a este problema da deslocação também tem de se ter em conta que existem transportes a horas específicas. Actualmente, os horários são da competência de cada escola mas os recursos são comuns às escolas, assim na partilha de recursos há que ter em conta e resolver os possíveis pontos de conflito. O sistema desenvolvido centraliza todos os horários de todas as escolas, excepto os do curso de música, garantindo que toda a informação se encontra acessível e actualizada.

O trabalho desenvolvido teve em conta os seguintes aspectos:

- 1 Disposição espacial dos edifícios da Universidade de Évora
- 2 Um conjunto de utilizadores responsáveis pela geração dos horários.
- 3 Expressão de preferências.
- 4 Geração automática/manual.
- 5 Edição dos horários gerados.

Para o desenvolvimento do sistema usou-se uma base de dados relacional para guardar a informação dos horários e os dados para os gerar. A geração dos horários foi implementada como um problema de satisfação de restrições no gnuprolog. Para se aceder aos dados da bases de dados a partir do gnuprolog usou-se o ISCO. O sistema permite de receber e fazer horários parciais permitindo que os horários da UE possam ser feitos gradualmente. Para a interacção com o utilizador, temos uma interface que permite fazer a edição aos dados, visualizar os horários e editá-los se necessário.

## **Estrutura**

Ao logo deste trabalho abordam-se os vários aspectos do desenvolvimento realçando as áreas mais importantes de cada componente do sistema e detalhando alguns os aspectos fundamentais.

Esta dissertação encontra-se dividida em sete capítulos. No segundo capítulo faz-se uma descrição geral do estado da arte com a apresentação de algumas das técnicas usadas na geração de horários em aplicações que foram desenvolvidas nesse âmbito.

No terceiro capítulo descrevem-se algumas das ferramentas usadas no desenvolvimento, destacando algumas características que motivaram a sua escolha.

No quarto capítulo, faz-se uma descrição do sistema de gestão de horários, introduz-se os conceitos e entidades definidos para o modelo de dados, e apresentam-se as funcionalidades do sistema.

No quinto capítulo descreve-se a interface e as suas características, expondo algumas das facilidade que proporciona ao utilizador.

No sexto capítulo, apresenta-se a implementação, indicando o modelo de dados usado, a representação das aulas e dos horários, e as funções que garantem a manutenção deste na base de dados de informação.

No sétimo capítulo, descreve-se a geração e a definição das restrições.



# Capítulo 2

## Estado da Arte

Neste capítulo apresentam-se algumas das técnicas para gerar/gerir horários e alguns sistemas de gestão de horários. Dos sistemas que são apresentados alguns são comerciais outros opensource, sendo que os primeiros revelam muito pouca informação sobre a sua implementação pelo que só se descrevem as suas funcionalidades de alto nível. Alguns dos sistemas apresentados, além da geração de horários de aulas de Universidades, também fazem escalonamento de exames e gestão de eventos não curriculares.

### 2.1 Técnicas para Geração de horários

Nesta secção vamos descrever algumas das técnicas actuais para geração automática de horários e outros problemas de escalonamento.

Na geração automática usam-se algoritmos que possibilitem a geração de uma ou mais soluções obedecendo a um conjunto de restrições fortes e procurando maximizar um conjunto de restrições fracas. Vamos apresentar resumidamente alguns desses algoritmos que são utilizados em sistemas actuais de gestão de horários.

#### Algoritmos Heurísticos e Coloração de Grafos

Encontrar pontos de conflito num problema de escalonamento é comparável a um grafo, uma vez que podemos representar por ex as aulas como nós e os arcos como a partilha de determinados recursos, docentes, salas, cursos.

A construção do grafo é feito da seguinte forma:

- Cada aula é representada num vértice
- Cada arco representa um docente, curso-turma, ou sala. Esta ligação implica que não podem ter o mesmo horário.

O problema é resolvido colocando a pergunta: será que conseguimos colorir o grafo sem que se repitam as cores de um conjunto de  $P$  cores, em vértices que se encontrem ligados?[Car86]

Um dos problemas levantado por esta técnica é a quantidade mínima de cores que precisamos para colorir o grafo, para que nenhum vértice ligado a outro tenha a mesma cor.

Na procura das soluções aplicam-se heurísticas que permitem chegar a soluções que sejam satisfatórias, i.e. verifiquem as restrições fortes e satisfaçam algumas das fracas.

Esta técnica levanta alguns problemas a nível computacional pois à medida que o número de vértices cresce o problema cresce exponencialmente, Manvel 1981[Man81].

A coloração de grafos é muito utilizada em geração de horários de exames. A utilização desta técnica deve ser cuidada pois o problema escala muito rapidamente(exponencial Manvel 1981[Man81]), há um conjunto de heurísticas aplicadas à técnica de coloração de grafos [Car86] <sup>1</sup>.

- 1) "Maior grau primeiro- [WP67] Os vértices são ordenados por grau (o número de arcos de cada vértice). A coloração é feita escolhendo os do início da lista atribuindo a cor com mais baixo valor que não tenha conflitos, o objectivo é de que os vértices com mais ligações serão mais difíceis de colorir caso se espere que os seu vizinho sejam coloridos.
- 2) "Maior grau modificado primeiro "(Williams 1974[Wil74]) propôs que só o grau do vértice não era suficiente para determinar o quão difícil é atribuir um horário. Considerou que um nó seria crítico se a maior parte dos seus vizinhos o fosse também. Desenvolveu uma formula que determinava o valor crítico de cada nó, demonstrando que estes convergiam para o valor próprio da matriz de vértices adjacentes.

Existem mais heurísticas para aplicar a esta técnica, muitas são variações umas das outras, mas todas demonstram que a coloração de grafos apresenta desafios na computação, o problema escala exponencialmente quando se faz crescer o número de nós. Uma nota deixada por M. R. Williams - "*Aviso não passar muito tempo a tentar resolver algo que não tem solução.*"<sup>2</sup>[Wil74], num artigo em que o autor descreve procedimentos heurísticos para resolver o problema da coloração.

<sup>1</sup>As heurísticas apresentadas aplicaram-se ao escalonamento de exames.

<sup>2</sup>Tradução livre do texto do autor

Esta técnica evidencia problemas na geração de horários quando se acrescentam conjuntos de restrições adicionais i.e. preferências, sendo necessário uma maior manutenção de dados, logo há mais sobrecarga na altura da coloração[Car86].

### Programação por Restrições

A programação por restrições é uma técnica bastante usada no cálculo de horários. A satisfação de restrições consiste na representação de uma rede de restrições que contém um conjunto de  $N$  variáveis com um certo domínio mais um conjunto de restrições que se lhe aplicam. A solução da rede é o conjunto de valores atribuídos a todas as variáveis do problema que satisfazem as restrições impostas sobre as mesmas. A resolução do problema implica que seja encontrada uma solução ou todas.

Vários autores apresentam esta técnica recorrendo à programação em lógica por restrições a fim de resolver o problema da geração de horários. Existem na literatura algumas referências à geração de horários para o ensino e exames com o recurso CLP ("Programação Lógica por Restrições do inglês Constraint Logic Programming") como [DOOS97], [Dav98], [RM00], [AB94]

Algumas das abordagens propõem a introdução de heurísticas "inteligentes" para tornar o sistema ou mais eficiente, ou mais respeitador das preferências que o utilizador pretende. No trabalho desenvolvido por [AB94], a abordagem proposta passa por usar um CLP que permite introduzir heurísticas na resolução dos valores das variáveis.

Em [DOOS97], propõe-se uma ordenação das variáveis a instanciar uma vez que o espaço de estado é muito grande, a ordenação é feita de acordo com os seguintes critérios:

- Importância e requisitos críticos dos sujeitos
- Grau de dificuldade

Assim procedem a uma pesquisa no espaço de estados, fazendo verificações para a frente e utilizando os valores obtidos se não violam as restrições. Esta estratégia foi aplicada num caso real e provou não só resolver horários como maximizou a utilização das salas.

Em [RM00] o problema da geração é resolvido com uma anotação feita as variáveis do problema e a sua respectiva ordenação aplicando-lhe uma função que visa minimizar os conflitos entre as aulas dos alunos, ajudando assim a resolver partes do problema que estejam sobre restritas levando a uma diminuição do espaço de estados para pesquisa.

Em [Dav98] a abordagem efectuada para gerar horários para os exames "École des Mines de Nantes" foi efectuada com o recurso a um problema de satisfação de restrições mas não recorreram a um método de pesquisa exaustiva, mas sim a um algoritmo incompleto e um conjunto de técnicas afim de

- Limitar o risco de violação de restrições com recurso a heurísticas
- Reparações locais para reduzir o número de restrições violadas

A técnica de reparação é classificada entre as de pesquisa exaustiva e os que reparam a solução durante a geração, assumindo um compromisso de tempo de computação e qualidade dos resultados. A reparação consiste num conjunto de procedimentos que são invocados quando os anteriores falham.

## **Sistema Multi-Agente para Geração de Horários no Ensino Secundário**

Este sistema foi implementado para gerar horários do Ensino Secundário Português.

A abordagem do sistema baseia-se num sistema distribuído multi-agente que denominaram GHES - Gerador de Horários no Ensino Secundário.

Cada agente tem restrições próprias e objectivos globais na criação de horários que satisfaçam esses mesmos objectivos ou impossibilidades gerais tentando respeitar as suas preferências tanto quanto possível.

A coordenação deste sistema é descentralizada, i.e os agentes trocam mensagens entre si. Cada agente está responsável por avaliar os resultados da negociação com um custo associado num determinado intervalo de tempo. As funcionalidades destes agentes são as seguintes: pesquisa, envio/recepção de mensagens, armazenamento temporário e validação de mensagens.

Este trabalho foi desenvolvido no âmbito da tese de mestrado em Inteligência Artificial e computação da Universidade do Porto, em Dezembro de 2000 [dS00].

## **2.2 Sistemas de Geração de Horários**

Existe no mercado um conjunto de ferramentas que permitem gerar / gerir horários.

Nesta secção são apresentados alguns desses sistema, expondo as suas características.

### **Aplicações Livres**

No âmbito das aplicações livres e de código aberto, existe um conjunto de ferramentas disponíveis que podem assistir à geração de horários.



### **Drools Solver**

Este software[[tea](#)] é direccionado para a geração de horários de universidades e exames. Este sistema não está feito para ser usado pelo utilizador final mas mais para ser usado como uma framework de desenvolvimento. Esta framework usa técnicas para a geração como cálculo de pontuação baseado em regras. O gerador usa pesquisa local e pesquisa tabu.

A framework vem com uma interface que não permite alterar as restrições, só permite alterar os pesos das regras directamente nos ficheiros.

Está implementada totalmente em java e usa ficheiros xml para guardar os dados.

### **FET**

Este software[[Lal](#)] está preparado para gerar horários de escolas do Ensino Secundário, Universitário e possivelmente pode fazer escalonamento de exames.

- Geração automática, semi-automática ou manual
- Independente da plataforma executa em qualquer sistema
- Cada restrição tem um peso atribuído (0% a 100%)
- Permite indicar um vasto conjunto de restrições sobre docentes, salas, cursos.

Usa um algoritmo heurístico para gerar horários e possui um interface de desktop. As restrições que se podem definir sobre docentes, cursos e salas, podem ser de natureza temporal i.e. indisponibilidade do recurso durante um determinado período ou de natureza espacial, por exemplo o docente pretende dar aulas num determinada sala.

### **Aplicações comerciais**

Cada vez mais são procuradas formas eficientes e eficazes de acelerar as organizações e as instituições escolares não são excepção. Existe um conjunto de aplicações comerciais que oferecem solução ao problema da geração de horários. Contudo estas aplicações podem tornar-se parte do problema em vez se tornarem a solução. São aplicações de múltiplas valências na resolução de horários.

A pesquisa por estes produtos foi feita através da Internet e a informação recolhida está disponível nos seus sites, uma vez que a produção científica desde produtos não se encontra disponível.

Consideramos as seguintes ferramentas comerciais.

### Wise Timetable

Esta ferramenta [Ltd] é bastante completa, e propõem uma simples interacção na produção de horários, no seu site inclui um vasto conjunto de tutoriais de iniciação ao seu uso e apresenta as seguintes características:

- Geração automática e manual de horários em múltiplas semanas
- Propõe resultados rápidos com uma interface simples
- Exporta os horários para vários formatos (Calendário do Google , Outlook, pdf, etc.)
- Equaciona na geração distancia entre os vários edifícios.
- Permite gerar horários de substituição.

O algoritmo de geração é feito com recurso a heurísticas, a aplicação está implementada em C. Tem duas interfaces, uma de desktop e outra web. Na interface web permite manipular o horário.

### EventMap

Esta aplicação [Lim] é adequada para gerar horários de exames, Universidades e de escolas do Ensino Secundário, Está implementada em C++ e recorre a uma base de dados como armazenamento de dados. Apresenta aos utilizadores as seguintes interfaces, versão desktop, web, e linha de comandos.

Apresenta as seguintes características:

- Horários para aulas e salas
- Gestão de eventos não curriculares
- Modelação de restrições e relatórios
- Permite criar cenários e planeamento "e se..".

O algoritmo de geração é feito com recurso a heurísticas e a aplicação esta implementada em C. Tem duas interface uma de desktop outra web. Na interface web permite manipular o horário.

## 2.3 Ferramentas Usadas

O conjunto de software desenvolvido para o sistema de horários foi implementado com o recurso a algumas ferramentas de desenvolvimento rápido orientadas para a web e um sistema de gestão de bases de dados.

A escolha destas ferramentas foi feita com base nos objectivos que se queriam atingir i.e. um sistema independente da plataforma e concorrente, na medida em que múltiplos utilizadores podem interagir em simultâneo; tanto quanto possível permitir a edição dos dados de forma concorrente; garantir que não é possível aceder aos recursos dos horários em simultâneo, salas, docentes e aulas.

Teve-se em conta que o sistema devia ser maleável ao ponto de permitir que se façam alterações sem que este precise de ser reescrito significativamente. A extensibilidade do sistema também foi considerada, podendo-se acrescentar outros módulos de manipulação ou novas funcionalidades sem impacto no sistema.

Usou um conjunto diverso de linguagens garantindo a interoperabilidade das diversas camadas de software.

A camada de software e as linguagens de programação usadas são:

- PHP, Python, Prolog, PlSql
- Yii Framework
- Postgresql
- GnuProlog / ISCO

### 2.3.1 YII Framework

A escolha deste Framework, [Xue], foi feita com base na necessidade de manter um bom modelo de dados. A forma como o YII lida com o modelo de dados não obriga que as tabelas tenham um único atributo com chave primária como outras ex. django cakePHP, permitindo assim que se implemente um modelo de dados sem que se fique refém da framework mas podendo ainda explorar todo o seu potencial.

O YII fornece as seguintes vantagens:

- 1) Modelo de Programação MVC.
- 2) Integração simples com SGBD
- 3) Providencia funcionalidades de DAO/Activerecord, migrações e construtor de queries.

- 4) Autenticação e permissões baseadas em papéis.
- 5) Validação de formulários
- 6) Extensibilidade de bibliotecas disponíveis por partes do core do Yii.

A Programação MVC , consiste na separação entre o modelo de dados, as templates (Views), ie o que se apresenta ao utilizador, e controladores que estão encarregues atender os pedidos do utilizador, aplicar as acções e ou manipular os dados e devolver uma resposta. A interacção com dados persistente é feita pelo mecanismo DAO, permitindo o acesso à base de dados aliado ao acesso concedido. Disponibiliza funções através de ActiveRecord que são a tradução das tabelas da base de dados permitindo, de forma transparente, a validação de campos, inserções, actualizações e remoções de objectos. Lida com chaves estrangeiras compostas que também é uma características que se encontra em poucas framework do género.

A internacionalização é outras das características desta framework, permite que a interface seja traduzível resolvendo automaticamente, caso exista, a tradução para a linguagem que o utilizador usa no browser.

A framework disponibiliza, no seu core, funcionalidades que permitem limitar o nível de interacção que cada utilizador pode fazer. Possibilita que se identifique através de um módulo de autenticação e posteriormente comporta-se de acordo com o papel que esse utilizador tem.

## Módulos

Tirando partido da extensibilidade do Yii recorreu-se a alguns módulos exteriores ao seu core, como ao modulo gestão de utilizadores, que permitiu definir grupos de utilizadores facultando um modo persistente de armazenamento na base de dados. Este modulo teve de ser adaptado para integrar com o postgres bastando criar o código SQL das tabelas. A parte deste modulo o core do Yii providenciou todas as facilidades necessárias para criar a interface.

## Jquery weekcalendar

Calendário implementado em javascript tem o estilo do calendário do Google. Uma vista semanal e pode ser usado como agenda, neste caso foi usado para mostrar os horários. Como os horários são semanais e de ocorrência única, a utilização deste recurso não é completa, fixa-se a data e projectam-se os horários numa determinada semana.

## Templating

Para o desenvolvimento da templates foi usado o twitter Bootstrap [OT] que providencia um esquema de 12 colunas que permite dividir o ecrã. Esta framework de templating suporta um vasto conjunto de dispositivos adaptando-se a cada um. No entanto não é utilizável em certas edições.

### 2.3.2 GNU Prolog/ISCO

O GnuProlog [Dia] é um compilador de Prolog com resolução de restrições de domínios finitos, desenvolvido por Daniel Diaz. De todas as potencialidades da linguagem prolog e deste compilador em particular as mais relevantes para este trabalho são o resolvedor de restrições e o ISCO.

O resolvedor de restrições do GnuProlog apresenta as seguintes características:

- Variáveis de domínio finito integram de uma forma transparente o ambiente GnuProlog, não é necessário de explicitar que se trata de uma variável de domínio finito.
- Restrições de alto nível podem ser descritas com recurso a primitivas simples
- Um vasto conjunto de restrições a disposição: aritméticas, booleanas, simbólicas, etc.
- Um conjunto de heurísticas de enumeração.
- Disponibiliza ao utilizador de criar as suas próprias restrições.

O ISCO [RMA<sup>+</sup>13] é uma framework com integração transparente com o GNUProlog e permite aceder a bases de dados relacionais. Um programa ISCO acede aos dados que podem provir de varias fontes. A interação com as fontes dos dados é feita como se de um predicado de uma base de dados Prolog se tratasse. Para que o acesso possa ser feito aos dados, uma representação deste deve de ser declarada. A declaração é encarada com um predicado, mas os factos estão num sistema exterior.

Algumas da características do SQL foram passadas como chaves, índices, chaves estrangeiras, herança, etc.

Os predicados ISCO declarados permitem um conjunto de operações CRUD<sup>3</sup> sobre estes.

- Inserção de novo factos (Criar)
- Permite recolher de forma sequencial os factos (Leitura).

---

<sup>3</sup>Do inglês (Create)Cria, (Read)Lê, (Update)Actualiza, (Delete)Apaga

- Actualizar um facto(Actualizar).
- Remoção de factos específicos(Apagar).

# Capítulo 3

## Sistema de Gestão de Horários

Um sistema de gestão de horários compreende um conjunto de funcionalidades para importação de dados, edição de dados, verificação de consistência, visualização de dados e geração automática de horários.

Para se poderem gerar horários há um conjunto de dados que tem ser introduzido como a informação sobre os cursos, os docentes, as disciplinas e preferências.

O sistema é importa os dados fornecidos pelo sistema de Informação (SI), estes são traduzidos e materializados no conceito de aula i.e. aquilo que decorre todas as semanas a uma dada hora que inclui um ou mais docentes, uma ou mais disciplinas e um conjunto de alunos

O sistema também necessita de informação sobre os espaços onde as aulas decorrem i.e. edifícios e salas.

### 3.1 Conceitos

No trabalho desenvolvido existe alguns conceitos que necessitam de ser introduzidos a fim de se perceber melhor o problema. Os conceitos expostos são importantes na medida em que as decisões tomadas na sua representação e implementação implicaram certos compromissos em toda a arquitectura do sistema.

## **Aula**

A aula é a unidade fundamental da aplicação, esta agrega a informação dos cursos, disciplinas e docentes. A aula tem a duração em 1/2 horas, e pode ocorrer uma ou mais vezes numa semana.

Um horário tem a hora a que se inicia e termina cada ocorrência da aula e as salas onde decorrem.

## **Horário da Aula**

Um horário é definido pelo dia da semana, a hora de início e a duração que deve ser consecutiva, a duração tem granularidade 30 minutos, i.e. o resto da divisão inteira da duração por 30 minutos é 0. Uma aula pode ter uma ou mais ocorrências e cada ocorrência tem um horário. Os docentes e as salas também podem ter um ou mais horários.

## **Horário**

Um horário é um conjunto de horários de aulas, a sua ocorrência é semanal ao longo do semestre e são visualizados sobre as dimensões: curso, docente, e sala.

O horário apresenta a alocação efectuada para esse recurso num determinado semestre.

## **Plano de estudos**

O plano de estudos associa a cada curso/ano/semestre um conjunto de disciplinas. Este serve de guia para a validação das aulas ou seja um curso /ano semestre deve ter uma aula, as suas disciplinas.

## **Sala**

A sala é o espaço físico onde decorrem as aulas. As salas estão em edifícios que se encontraram espalhados pela cidade. Uma sala tem uma ocupação máxima e pode ter um função, aulas teóricas, práticas, laboratório de química geral, etc.

As salas podem ser ocupada por aulas ou ser alvo de uma reserva extra curricular.

Todas as salas tem um horário onde para cada meia hora se especifica se está livre ou ocupada.



### **Docente**

O docente lecciona aulas, cada aula pode ser leccionado por um ou mais docentes. Um docente não pode estar associado a aulas que tenham coincidência de horário.

### **Disciplina**

A disciplina é a unidade curricular que é leccionada numa aula, pode existir mais que uma aula com a mesma disciplina para o mesmo semestre de ano de um curso. Em regra a disciplina é dividida em varias vertentes i.e. teórica, prática, teórico-prática, etc. Uma aula tem de estar associada a uma ou mais disciplinas.

### **Curso**

Um curso tem um plano de estudos, uma lista de disciplinas que corresponde após conclusão a um grau de diploma. No plano de estudos as disciplinas são agrupadas por ano-semester.

O horário de um curso é definido por ano/semester e obtém-se coleccionando os horários de cada uma das aulas associadas ao curso ano semestre.

### **Turma**

A turma é um sub conjunto de alunos que frequentam um determinado curso/ano/semester. Este conceito é importante porque permite que no mesmo curso/ano/semester algumas aulas tenham horários coincidentes se as turmas forem diferentes.

## **3.2 Preferências**

Num sistema de gestão de horários tem de se introduzir um conjunto de dados adicionais que permitem guiar o problema, as preferências. Sem preferências o sistema de geração seria caótico do ponto de vista do resultado final pois bastaria cumprir as restrições fortes. Contudo estes horários não são aceitáveis. Com as preferências conseguimos controlar de uma forma efectiva os resultados. Algumas preferências são traduzidas em restrições rígidas ou seja a solução irá falhar se não forem cumpridas ou em restrições suaves, i.e mesmo que não sejam cumpridas o sistema retorna uma solução. Nos tópicos seguintes serão expostas as preferências que podem ser definidas e as implicações têm na geração dos horários.

### 3.2.1 Docente

Nos docentes podemos indicar as seguintes preferências fortes:

- número máximo de dias por semana MDS
- número máximo de horas por dia MHD
- Disponibilidade do docentes DISP

Com estas preferências podemos extrair as seguintes restrições: as aulas leccionadas do docente não podem ser em mais do que o número máximo dias por semana MDS, o número de horas de aulas em cada dia não pode exceder MHD e a disponibilidade do docente indica em que dias e horas este está disponível para leccionar.

Após a configuração das preferências de um docente é possível determinar se o conjunto de aulas que lhe estão associadas viola as preferências definidas.

### 3.2.2 Curso

Nos cursos existem duas preferências distintas. A primeira consiste em configurar para cada ano semestre:

- O número máximo de horas diárias (NHMD), preferência forte.
- Indicar se se prefere a maioria das aulas de manhã ou tarde ou ambos, preferência fraca.
- Turno, preferência forte.

O turno é o conjunto de horas diárias que se pretende que o curso tenha, por exemplo todos os dias das 8h as 20h, este turno é particularmente útil no caso de se pretender impor uma restrição forte sobre a hora de início e fim de cada dia.

A restrições impostas são: o curso no semestre de um ano não pode ter mais do que NHMD horas de aulas por dia, tendencialmente queremos colocar a maioria das aulas de manhã ou de tarde, e o horário do curso não pode estar fora do seu turno.

O segundo grupo de preferências que podemos definir no curso é as salas onde aulas do curso num dado ano e semestre podem decorrer, esta preferência é genérica na medida em que potencialmente servirá para centrar um determinado curso ano semestre num determinado edifício.

#### 3.2.3 Aula

Na aula podem-se indicar preferências ao nível das salas onde esta pode ser leccionada, as preferências podem ser indicadas de 2 formas, ou pelo tipo da sala ou indicando as salas individualmente. A aula por associação pode também receber as preferências de salas dos cursos.

### 3.3 Descrição da funcionalidade

No sistema de horários disponibiliza-se um conjunto de funcionalidades para a edição de aulas e preferências.

Existem três grupos de funcionalidades: edição, consistência, geração e visualização . Algumas destas funcionalidades surgem misturadas sendo difícil fazer uma separação clara das mesmas.

As funcionalidade apresentadas foram desenhadas para facilitar ao utilizador a gestão e a criação dos horários.

Nesta secção apresenta-se uma visão geral sobre cada uma das funcionalidades.

#### 3.3.1 Edição

Existem dois tipos de edição, a edição de dados relativos as aulas e a edição de preferências. A edição de aulas e de salas vem satisfazer a necessidade de poder corrigir /introduzir dados que foram inicialmente carregados do SIIUE.

Nas aulas existe um conjunto de operações que podemos fazer como:

- Juntar aulas do mesmo tipo
- Retirar cursos das aulas
- Separar aula pelo curso
- Duplicar uma aula

Nas aulas edita-se a duração, tipo, recorrência semanal e outros, sendo possível indicar o docente, a disciplina e o curso ano /semestre /turma. Na aula podemos indicar/editar o seu horário e a sala onde irá ocorrer.

A edição das preferências permite que se modifiquem os docentes e os cursos, esta informação é inicialmente composta por valores padrão. Podem-se indicar preferências relativas às salas nas aulas ou nos cursos.

### 3.3.2 Consistência

A verificação de consistência é necessária para podermos identificar problemas antes da geração dos horários ou após uma geração falhada afim de identificar os problemas. A consistência não é mais que um relatório com os possíveis problemas que podem levar ao insucesso da geração. A verificação de consistência produz os seguintes relatórios:

- Salas que estejam sobrecarregadas com aulas e que não possuam alternativas
- Docentes que tenham mais horas atribuídas do que as que foram definidas nas preferências.
- Aulas que dadas a sua frequência semanal e duração excedam a capacidade diária do curso.
- Verificação das turmas dos cursos e do cumprimento do plano de estudos.

### 3.3.3 Geração

A geração é a funcionalidade que permite gerar um horário, para que a geração seja efectuada tem de se indicar o curso, o ano e o semestre. A geração pode ser feita também para um conjunto de cursos.

A geração foi colocada a par da visualização para que se possa ver o resultado final sendo que um dos resultados possíveis é o insucesso da geração, podendo-se pedir uma nova geração.

A geração incide sempre sobre as aulas que ainda não tenham horário não alterando as que têm horário.

Para refazer um horário de um curso é preciso que o horário já feito seja removido.

Para a geração é disponibilizada uma funcionalidade que permite trancar o horário das aulas i.e. podemos bloquear os horários das aulas que não queremos que sejam limpos para uma geração futura, uma vez que a geração não refaz as aulas cujo o horário já tenha sido atribuído.

### 3.3.4 Visualizar / Editar

Um horário depois de ser criado pode ser visualizado e eventualmente alvo de correcções pontuais. A função de visualização permite visualizar horários sobre várias perspectivas nomeadamente por curso/ano/semestre, docente ou sala. Na visualização disponibiliza-se um conjunto de funções que permitem modificar um horário ou a sala de uma aula. Na visualização obtém-se o resultado final da geração que pode ser exportado para o formato digital(pdf).

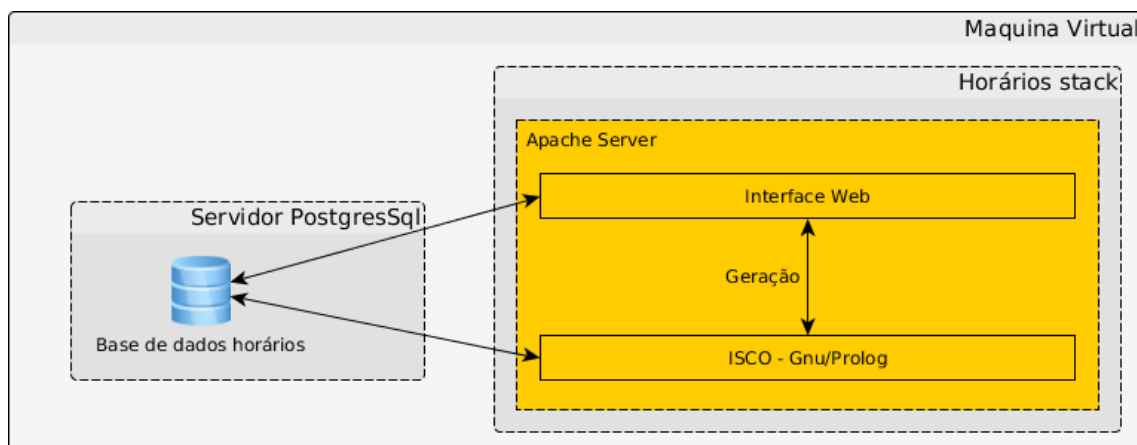


Figura 3.1: Arquitectura do Sistema

### 3.4 Arquitectura do Sistema

O sistema de gestão/geração de horários tem a arquitectura apresentada na fig.3.1, o diagrama mostra as várias camadas do sistema evidenciado as ligações entre elas. O sistema em cada uma das suas camadas é mais complexo que o diagrama apresentado.

As interações entre as várias camadas processam-se da seguinte forma: a interface web esta preparadas para receber os pedidos do utilizador via web, como por exemplo, consultar um horário, editar preferências, etc.; o gerador automático possui uma interface que permite que se lançado na interface, uma vez lançado tranca-se a funcionalidade aos outros utilizadores até que esta termine evitando a concorrência na geração. Ambas as camadas interagem com a base de dados para guardar os dados de forma permanente.

### 3.5 Interface

A interface é a camada visível da aplicação e pode ser acedida por qualquer browser moderno, permitindo que o seu acesso seja feito por um vasto conjunto de dispositivos em qualquer local com acesso a web. Pela interface podemos aceder a todas as funcionalidades necessárias para gerar e editar os horários da U.E. que inclui visualizar horários e indicar preferências das aulas cursos e docentes. A interface web disponibiliza-se num sistema que não necessita de ser instalado, bastando que o utilizador disponha no seu computador de um browser instalado. Como a plataforma fica exposta, tem um sistema de acesso que identifica o utilizador e concede determinados privilégios no que concerne à edição ou geração. O sistema de autenticação impede que o utilizador aceda a áreas reservadas do sistema. As permissões são herdadas através de grupos de utilizadores e permitem executar determinadas operações. O utilizador quando acede à aplicação tem o seguinte ecrã ver Figura 3.2. Nas secções seguintes apresentam-se os aspectos mais relevantes da



Figura 3.2: Página inicial da interface

interface e a forma como esta disponibiliza as funcionalidades e apresenta a informação ao utilizador.

### 3.5.1 Edição

A funções de edição permitem ao utilizador corrigir ou acrescentar a informação que será usada para gerir os horários. A informação editável é a que diz respeito às aulas, horários de aulas, preferências de docentes e cursos. Disponibiliza-se um modulo de edição de espaços, edifícios e salas que permite a geri-los. A gestão de espaços além da criação de salas e edifícios, associação de salas a edifícios e definição do tipo de sala e capacidade também incluiu a possibilidade de efectuar reservas de salas indicando impedindo o funcionamento de aulas no horário especificado. A edição oferece ao utilizador um sistema de avisos. Sempre que exista um problema nos dados introduzidos informa-se o utilizador do problema. Alguns erros impedem que se guarde os dados inseridos, outros informam o utilizador que potencialmente pode ocorrer um problema por exemplo na geração de horários.

#### Edição de aulas

A edição de aulas permite editar os docentes, disciplinas e cursos associados a uma aula. Também se podem editar outros atributos como o tipo de aula (Teórica, Prática,...), o número de alunos previstos, a duração e o número de vezes por semana que ocorre a aula. A edição de preferências de salas pode ser feita de duas formas, indicando o tipo de sala que se pretende para a aula, um edifício e um tipo de sala ou um conjunto específico de salas. Na edição de uma aula pode-se definir o horário indicando o dia da semana , a

Número Máximo de Dias por Semana

3

Número Máximo de Horas por Dia

6

Figura 3.3: Porção da interface - Indicação das horas e dias máximos do docente

	Segunda	Terça	(...)
08:00	08:00 to 00:00 Disponível	08:00 to 00:00 Disponível	08:00 to 00:00 Disponível
09:00			
10:00	(...)	(...)	(...)

Figura 3.4: Porção da interface - Indicação da disponibilidade diária do docente

hora e a sala, a indicação da sala é facultativa. O horário pode ser trancado, impedindo a sua remoção caso se tente limpar o horário de um curso que inclua a aula. Desta forma é possível indicar que aquele horário já não se altera, já se encontra fechado. No entanto esta funcionalidade é reversível podendo sempre libertar-se o horário (ver figura 9).

## Preferências

A interface oferece um conjunto de funcionalidades de alto nível que permite expressar as preferências para os cursos, os docentes e as salas das aulas.

## Docentes

Para os docentes pode definir-se o número máximo de dias por semana em que têm aulas e o número máximo de horas de aulas num dia fig. 3.3. Em complemento pode-se indicar a disponibilidade do docente nas semanas de aulas, marcando-a num mapa semanal fig. 3.4. As preferências do docente são interpretadas como restrições fortes através da interface (ver fig. 10).

## Cursos

Nos cursos deve-se definir a preferência por aulas de manhã/tarde para os horários dos ano-semester's do curso. Esta preferência é interpretada como uma restrição suave. A preferência pelo turno também é obrigatória e será interpretada como uma restrição forte

Ano	Semestre	Número de Turmas
1	impar	4
1	par	1
2	impar	2

Figura 3.5: Porção da interface - Indicação das turmas de um curso/ano/semestre

Horas Máximas Diárias	Manhã	Tarde	Turno
8	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Diurno
8	<input type="checkbox"/>	<input type="checkbox"/>	Diurno
8	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Diurno

Figura 3.6: Porção da interface - Horas máximas preferência manhã, tarde, turno

fig. 3.6. Junto as preferencias do curso tambem são indicadas as turmas para cada ano semestre fig. 3.5. Interface com as preferências do curso fig. 12.

A definição de um conjunto de salas para o curso ano semestre, não é obrigatória mas permite que se atribua um ou mais tipos de sala de um determinado edifício de forma a poder centrar o curso num conjunto de salas de alguns edifícios. (ver figura 13).

## Aulas

Nas aulas podem-se definir preferências sobre as salas, indicando a sala/edifício ou indicando o edifício / tipo de sala. As preferências podem ser indicadas como se vê na figura 3.7, sempre que não exista preferência de salas na aula são apresentadas as do curso da aula no respectivo ano semestre fig. 15.

## Espaços

Na edição de espaços pode-se inserir ou alterar a informação sobre salas ou edifícios da Universidade. A sala está contida num edifício, pode-se editar atributos como a capacidade ou tipo de sala. O tipo de sala pode estar relacionado com a sua função ou com o equipamento que lá se encontra. Podem-se introduzir edifícios novos visto que a Universidade de Évora nem sempre dispõe dos equipamentos necessários para leccionar certo tipo de aulas e recorre a espaços de terceiros. Nos edifícios é possível acrescentar, remover e alterar salas. Cada sala tem um horário semanal. As salas podem ser pré-reservadas para outras actividades impedindo que se coloquem aulas a essa hora na sala, para isso a interface tem um sistema de reserva de espaços Figura 14.



**Tipo de Sala**

Edifício      Tipo de Sala

**Sala(s)**

Edifício      Sala

Gravar

Figura 3.7: Porção da interface - Preferências salas, tipos de salas nas aulas

	Segunda	(...)
08:00		
09:00		<input checked="" type="checkbox"/> Física Geral I CLAV - 061-H Turma: B
10:00		
11:00	<input checked="" type="checkbox"/> Física Geral I CLAV - 061-E Turma: C	<input checked="" type="checkbox"/> Análise Matemática I CLAV - 133 Turma: A,B
12:00		
13:00	(...)	(...)

Figura 3.8: Porção da interface - Edição das aulas no Horário

### 3.5.2 Visualização

A interface permite visualizar os horários gerados Figura 16. Os horários podem ser vistos por curso /ano, docente ou sala, o semestre tem sempre que ser indicado fig. 3.9. O módulo de visualização também permite alguma edição permitindo que se façam correções no horário fig. 3.8. As correções são sempre verificadas de forma a que se garanta a consistência. Se alguma restrição é violada a alteração não tem efeito e o utilizador é avisado do problema. Nesta interface pode-se eliminar o horário de um curso-ano-semester completamente ou parcialmente.

The image shows a web interface for course scheduling. It is divided into three main sections: **Curso**, **Docente**, and **Sala**.

- Curso**: Contains a dropdown menu for the course name (currently showing "Engenharia Informática (B\_EI..."), a dropdown for the semester (currently "1"), a dropdown for the semester type (currently "impar"), a "Select Some" button, a "Ver" button, a green "Gerar" button, a red "Limpar" button, and a blue "Fixar/Libertar horario de todas as aulas" button.
- Docente**: Contains a dropdown menu for selecting a teacher ("Selecionar Docente"), a dropdown for the semester (currently "Semestre"), and a "Ver" button.
- Sala**: Contains a dropdown menu for selecting a building ("Selecionar Edifício"), a dropdown for the room name (currently "Sala"), a dropdown for the semester (currently "Semestre"), and a "Ver" button.

Figura 3.9: Porção da interface - Opções de Visualização do horário

# Capítulo 4

## Implementação

Neste capítulo apresenta-se a implementação do sistema de gestão de horários. Este sistema foi desenhado para se integrar com o SIIUE na entrada e na saída de dados.

**Dados** - Os dados sobre os cursos, planos de estudos, docentes, salas e distribuição de serviço docente são importados do SIIUE mas ainda requerem alguns ajustes para se conseguir gerar horários, é necessário acrescentar e editar alguma informação no sistema enquanto não for o SIIUE a recolhe-la.

**Modelo de dados** O modelo de dados visa reflectir a lógica dos dados, pelo que o seu desenho merece especial atenção. O cuidado no desenvolvimento do modelo de dados é fundamental uma vez que facilita a pesquisa e a manutenção da informação.

Na modelação dos dados teve-se em conta o objectivo de manter um conjunto de restrições fortes para garantir a consistência dos horários representados bem como a necessidade de usar um sistema de restrições para gerar os horários automaticamente.

A representação do tempo foi um decisão importante, que corresponde a um compromisso com consequências na representação de outras entidades e na eficiência do sistema de geração automática de horários.

Outros requisitos que tiveram que ser tidos em conta na modelação de dados foi a necessidade de manter horários incompletos e parciais, e a possibilidade de editar horários impondo horário a aulas individuais.

Os dados sobre os cursos, planos de estudo, disciplinas, docentes são importados directamente do SIIUE e modelados na base de dados do sistema porque permitem gerar as

restrições necessárias para a consistência dos horários.

No desenho da base de dados procurou-se tirar partido do Sistema de gestão de base de dados para gerir algumas das restrições mas nalguns casos teve-se que se desenvolver um conjunto de funções que garantem consistência, aliado a estrutura relacional do SGBD usado.

## 4.1 Representação dos dados

A estruturação e representação da informação efectuada teve em conta as possíveis formas para a representação de horários, as decisões tomadas foram influenciadas pelas várias camadas que interagem com os dados, geração automática e edição local, e tiveram como principal objectivo minimizar as manipulações pelas diferentes camadas do sistema.

Uma vez construído o modelo de dados a sua implementação foi feita tendo em conta as várias camadas de software que fazem parte da arquitectura do sistema Figura 3.1.

Do modelo de dados destacamos seguintes as entidades e relações, descritas abaixo, como as que estruturam os dados do sistema.

### Curso

Para os cursos foram guardadas as seguintes informações o código do curso, nome e grau. Esta informação vem do SIIUE.

Como cada curso necessita de um horário para cada ano semestre, criou-se uma relação para associar o ano-semester com o curso, além da informação sobre o número de turmas para o ano-semester do curso e as preferências para o horário tais como o turno pretendido, o número de horas máximas diárias, preferência pela manhã ou tarde (figura 2). Esta informação sobre o número de turmas do ano-semester de um curso e as preferências para o horário têm que ser editadas localmente.

### Docente

Os atributos da entidade docente incluem o identificador do docente, nome e o departamento, esta informação é importada do SIIUE.

No docente também se representam as preferências dos docentes que incluem: o número de dias por semana, o número de horas máximas por dia e o domínio de valores válidos para o horário das aulas do docente. Estas preferências serão usadas na geração e devem ser introduzidas depois da importação dos dados do SIIUE. A figura 11 representa o modelo E-R dos docentes e as suas preferências

### 4.1.1 Aulas

Para as aulas representa-se o número de horas da aula, o número de vezes que ocorrem na semana, os docentes que dão a aula, as disciplinas leccionadas na aula e os cursos a que é leccionada. As entidades aula, curso, docente e disciplina são modeladas com as seguintes tabelas <sup>1</sup>.

- Aulas previstas
- Relação aula curso
- Relação aula docente
- Relação aula disciplina
- Relação preferência pela sala

Com estas tabelas representa-se a entidade aula Figura 3.

### 4.1.2 Horário das aulas

A representação dos horários das aulas é feita criando um tempo para cada 30 minutos, um par identificador valor. Uma aula de 2 horas terá quatro tempos. Para lidar com o facto de uma aula poder ocorrer mais de uma vez por semana, uma aula pode ter vários horários.

Quando os dados são carregados do SIIUE ou uma nova aula é criada no sistema cria-se uma ocorrência de aula e os tempos da ocorrência.

As entradas nas tabelas são criadas de forma automática via trigger sec. 4.2 e sempre que se verifica uma alteração nos atributos da aula relativo a sua duração ou ocorrência semanal.

### Aulas - Salas

As aulas podem ter preferência por salas. Esta informação é representada indicando o conjunto de salas que pode ser usado para colocar a aula, ou por uma lista de edifício tipo de sala, ou por uma, ver a figura 5.

### Cursos - Preferências

Para as preferências dos cursos representa-se a seguinte informação em cada ano semestre de um curso: o número de turmas, preferência por aulas de manhã ou de tarde e um turno

---

<sup>1</sup>As tabelas apresentadas são exclusivamente a representação da aula.

(diurno, noturno, etc). O turno de um curso é construído com a mancha semanal onde se podem colocar aulas. Também se pode especificar para cada ano-semester de cada curso o conjunto de edifícios tipo de sala onde o as aulas podem decorrer. Desta forma podemos atribuir à partida um conjunto de tipo de sala num dado edifício.

Na figura 7 e na figura 6 apresentamos o modelo E-R para representar as preferências dos cursos.

## Docentes - preferências

A representação das preferências do docente foram feitas através da tabela 'docpref' que tem os atributos id docente, domínio, horas por dia e número dias por semana. Para cada docente regista-se o número de horas máximo que o docente pode leccionar num dia (horas por dia), o número máximo de dias da semana em que pode leccionar (Número de Dias), e ainda se permite registar a mancha da semana em que o docente pode leccionar (dom) Figura 11. A todos os docentes é atribuído uma preferência no início, o SGBD fica encarregue da criação de forma automática da mancha semanal através de um trigger sec.4.2. Estes dados podem ser editados definido as preferências específicas de cada docente.

## 4.2 Funções e Triggers

Para manter os dados consistentes criou-se um conjunto de funções do lado do SGBD que permite manter a consistência e facilitam a manipulação na edição e importação de dados.

Algumas das funções implementadas são usadas de forma automática como triggers associados a tabelas, outras permitem inserções e actualizações do dados de uma forma simples.

Os triggers estão encarregues de actualizar os dados e garantir a sua consistência, as funções facilitam a integração com outros sistemas dispensando o conhecimento detalhado do esquema da base de dados para alguns acessos, podem ser entendidas como a API de acesso aos dados.

### Triggers

Os triggers permitem que certos procedimentos sejam despoletados automaticamente quando são feitas determinadas operações sobre as tabelas.

Sempre que são feitas alterações nas aulas, nº de turmas, e ocupação de salas é necessário que outras tabelas sejam actualizadas para que as alterações produzam efeitos, os triggers implementados são os seguintes:

- Geração das ocorrências aulas

- Geração das slots das aulas
- Consistência entre o triplo salasslot, ocupação aula, ocupação genérica
- Desmarcação da sala quando se retira o horário da aula
- Inserção de turmas e remoção via preferências dos cursos
- Geração automática das preferências do docentes
- Actualização do número de turmas quando a remoção da mesma.

## Funções

A implementação de funções do lado da base de dados tem como objectivo facilitar a consulta e a inserção dos dados. Estas funções facultam uma API de acesso aos dados sem que seja preciso ter conhecimento prévio do modelo de dados. As funções que permitem a inserção de dados mantêm a consistência recusando a actualização ou a inserção caso sejam violadas algumas das restrições. A consulta dos dados não passa por mostrar vistas sobre tabelas mas sim por disponibilizar um conjunto de funcionalidades que permitem verificar a consistência e a recolha de alguma informação extra.

As funções implementadas são as seguintes:

- Verificar aulas curso turno - Verifica se as aulas do curso estão no turno atribuído ao curso.
- Verificar aula docente dom - Verifica se as aulas do docente satisfazem as suas preferências.
- Obter informação horas turmas curso - Calcula a carga horária por turmas para um ano semestre de um determinado curso.
- Determinar o semestre da aula - Retorna o semestre da aula
- Verificar restrições do curso ano semestre - Verifica se não há sobreposições de horário para o ano semestre de um dado curso.
- Dar horário a uma aula - Permite introduzir o horário de uma sala manualmente desde que não viole restrições.
- ocupar uma sala com aula / outra - Permite afectar uma sala a uma aula desde que não viole nenhuma restrição
- Salas livres - Lista as salas livres para um determinado horário
- Recolher a ocupação da sala - Permite ocupar uma sala num horário para uma actividade.
- Funções de população de dados - recolhem os dados do SIIUE

### 4.3 Geração

Na geração, a informação colocada na base de dados é usada para gerar os horários relativos a um determinado curso. A informação que está contida na base de dados é traduzida em restrições e em preferências. O cálculo dos horários é feito da seguinte forma:

- Recolha da informação
- Construção das restrições a resolver.
- Construção dos horários
- O resultado gerado é guardado na Base de dados.

A implementação para resolução do problema está feita em GNU Prolog/ISCO. O cálculo do horário é efectuado começando por recolher a informação das aulas que consiste na alocação que previamente foi efectuada de docentes cursos e disciplinas, depois recolhem-se as variáveis de cada ocorrência das aulas e finalmente as salas que se pretendem para a aula.

Para cada aula existem duas slots por cada hora, a slot é um conceito que se usa para representar as meias horas das aulas, que precisam de ser instanciadas, uma aula com uma hora e trinta minutos terá 3 slots de tempo a instanciar, tem de se garantir que as slots são instanciadas com números consecutivos para que a aula não seja fraccionada por meias horas não consecutivas.

Na base de dados existe um identificador único para cada slots, `idVarslot`, que usamos durante o processo de instanciação, as variáveis quando recolhidas podem ter o valor 0 ou um valor entre 1 a 192. Sempre que este valor é diferente de zero consideramos que já está instanciada. se o valor da variável é 0 precisa de ser calculado, esta abordagem permite que se possa lidar com horários parcialmente preenchidos, permitindo que estes possam ser feitos de forma iterativa.

Para cada variável com o valor 0, não instanciada, o seu domínio inicial é 1...192, contudo o domínio efectivo excluirá os valores do conjunto dos cursos ano semestre turma, e do docente que já foram instanciados.

As variáveis das slots têm que obedecer às seguintes restrições.

#### Restrições Fortes

- As slots das aulas tem de ter valores consecutivos.
- As slots das aulas de um curso/ano/semestre/turma não se podem sobrepor.
- As slots das aulas associadas a um determinado docente não se podem sobrepor



- As aulas de uma disciplina que tenha componente prática não podem ser leccionadas no mesmo dia.
- As aulas que têm mais do que uma ocorrência semanal devem ter as essas ocorrências em dias diferentes.
- As aulas consecutivas de um curso/ano/semestre/turma ou de um docente não podem estar em edifícios diferentes.
- As aulas de um curso/ano/semestre/turma num dia não podem exceder o número máximo de horas definido para o curso (6 horas é o valor inicial).
- Um docente não tem mais do que o número de horas por dia definido nas suas preferências (6 horas é o valor inicial). Esta restrição é obtida a partir da base de dados para cada docente em particular.
- Um docente não tem aulas em mais dias do que o definido nas suas preferências (3 dias é o valor inicial).

### Restrições Fracas

As restrições fracas são consideradas preferências, procura-se maximizar a sua satisfação, contudo há soluções que podem não as satisfazer.

Curso Preferência definida sobre o período em que as aulas ocorrem, de manhã ou de tarde.

Curso O número de horas de aulas em cada dia da semana deve ser semelhante.

Salas A alocação é feita com base nas preferências dos cursos ou tipos de sala ou no conjunto de salas preferido para a aula.

A atribuição de salas é feita afectando as slots das aulas com a sala da aula, obtém-se uma disjunção das salas possíveis para a aula.

### Resolução do Problema

A geração automática de um horário é obtida resolvendo o problema de satisfação de restrições sobre as variáveis das slots das aulas e as variáveis das salas de aulas.

Procura-se maximizar as restrições fracas recorrendo à pesquisa local do gnuprolog mas devido à complexidade do problema, exponencial no número de variáveis, tem de se utilizar um timeout.

O sistema foi usado para fazer os horários de três Escolas da Universidade de Évora para o primeiro semestre de 2013/2014. Foram 71 cursos que tinham algumas disciplinas

leccionadas pela Escola de Ciências Sociais que não fez todos os horários no sistema mas tinha os horários das disciplinas leccionadas a estes cursos. Os cursos em média tinha 2.5 turmas, o curso com mais turmas tinha quatro e o menor uma. Os 71 cursos tinham 921 aulas e 348 docentes. Universidade disponibilizou para os horários 242 salas distributivas por 15 edifícios.

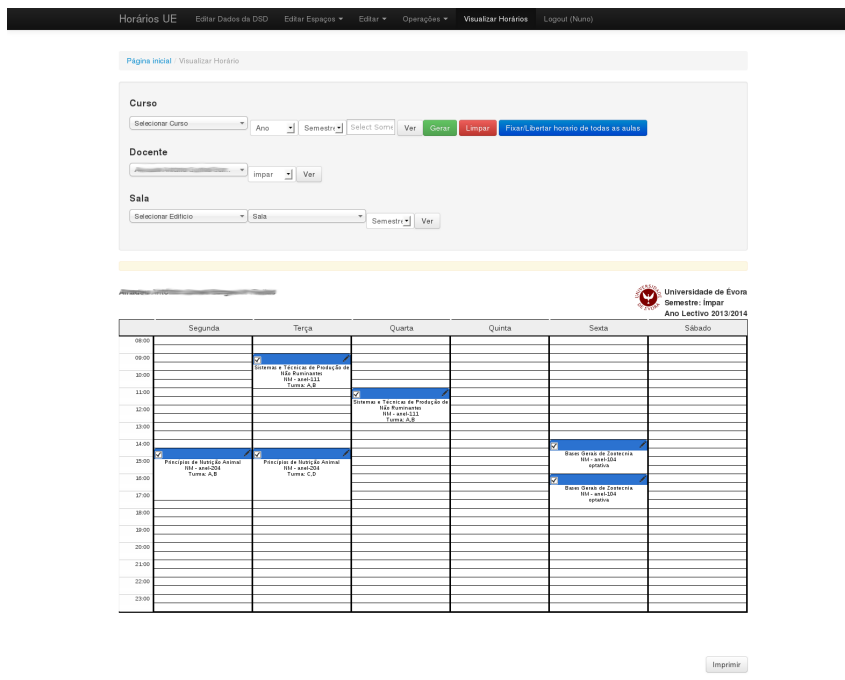


Figura 4.1: Horário de um docente



# Capítulo 5

## Conclusões

O desenvolvimento de soluções para horários é algo complexo, não só do ponto de vista computacional na busca de soluções mas também porque o produto final tem implicações sobre alunos e docentes, que de alguma forma colocam condicionantes nas suas vidas.

Até agora na Universidade de Évora, os horários eram feitos manualmente consumindo tempo e recursos. Embora, na aplicação seja possível fazer os horários à mão, o seu propósito é que estes seja feitos automaticamente com a intervenção humana dentro estritamente necessário.

A abordagem tentou minimizar a necessidade de ajustes aos dados e facilitar a definição das preferências na interface.

A utilização das ferramentas escolhidas contribuiu em muito para o sucesso da implementação. Uma vez que a informação está numa base de dados relacional, podemos tirar partido do sistema gestão de bases de dados para garantir consistência dos dados em muitos casos.

O uso do ISCO foi fundamental na medida em que permitiu o acesso aos dados da base de dados a partir do prolog, garantindo uma integração transparente com o gnuprolog. Para a geração o uso do gnuprolog é também importante não só pelo seu solver mas também porque o programa compilado pode ser invocado a partir da interface.

Na busca da melhor solução as ferramentas usadas para a implementação cumpriram com os objectivos, pois podemos conjugar um bom modelo de dados que denota a clara noção de uma aula e do seu horário.

A análise do problema e escolhas feitas levaram-nos a uma solução que consegue responder da melhor forma às necessidades da Universidade de Évora.



# Referências bibliográficas

- [AB94] Francisco Azevedo and Pedro Barahona. Timetabling in constraint logic programming. In *WORLD CONGRESS ON EXPERT SYSTEM*, 1994.
- [Car86] Michael W. Carter. A survey of practical applications of examination timetabling algorithms. *Operations Research*, 34(2):pp. 193–202, 1986.
- [Dav98] Philippe David. A constraint-based approach for examination timetabling using local repair techniques. In *Practice and Theory of Automated Timetabling II*, pages 169–186. Springer, 1998.
- [Dia] Daniel Diaz. <http://www.gprolog.org/>. Visualizado em 1/08/2013.
- [DOOS97] SB Deris, S Omatu, H Ohta, and PABD Samat. University timetabling by constraint-based reasoning: A case study. *Journal of the Operational Research Society*, pages 1178–1190, 1997.
- [dS00] Pedro Nuno Macedo Leite da Silva. Sistema multi-agente para geração de horários no ensino secundário. Master’s thesis, Universidade do Porto, Dezembro 2000.
- [Lal] Liviu Lalescu. <http://www.lalescu.ro/liviu/fet/>. Informação consultada a data 8/08/2013.
- [Lim] EVENTMAP Limited. <http://www.eventmap-uk.com/site/>. Informação consultada a data 7/08/2013.
- [Ltd] Wise Technologies Ltd. <http://www.wisetimetable.com/>. Informação consultada a data 10/08/2013.
- [Man81] Bennet Manvel. Coloring large graphs. *Congressus Numerantium*, 33:197–204, 1981.

- [OT] Mark Otto and Jacob Thornton. <http://getbootstrap.com/2.3.2/>. Visualizado em 1/08/2013.
- [RM00] Hana Rudová and Luděk Matyska. Constraint-based timetabling with student schedules. In *PATAT 2000 - Proceedings of the international conference on the Practice And Theory of Automated Timetabling*, pages 109–123, Constance (Germany), 2000.
- [RMA<sup>+</sup>13] I. Rodrigues, N. Matos, S. Abreu, R. Deneckere, and D. Diaz. Towards constraint-informed information systems. In *Research Challenges in Information Science (RCIS), 2013 IEEE Seventh International Conference on*, pages 1–10, 2013.
- [tea] Red Hat R&D team. <http://www.optaplanner.org/>. Informação consultada a data 10/08/2013.
- [Wil74] MR Williams. Heuristic procedures (if they work—leave them alone). *Software: Practice and Experience*, 4(3):237–240, 1974.
- [WP67] Dominic JA Welsh and Martin B Powell. An upper bound for the chromatic number of a graph and its application to timetabling problems. *The Computer Journal*, 10(1):85–86, 1967.
- [Xue] Qiang Xue. Yii framework. <http://www.yiiframework.com/>. Visualizado em 1/12/2013.



# Anexos



**Anexo**

**Modelo ER da Base de dados**

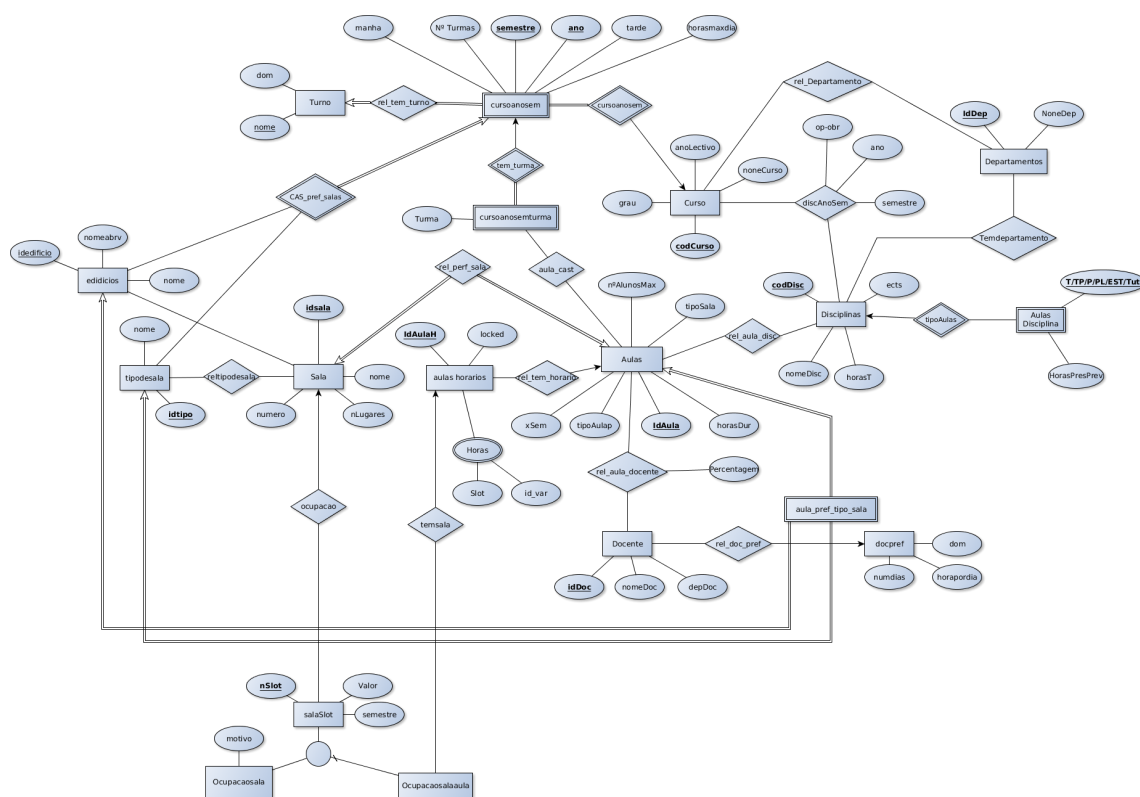


Figura 1: Modelo ER da Base de dados

**Anexo**

**Partes do Modelo ER**

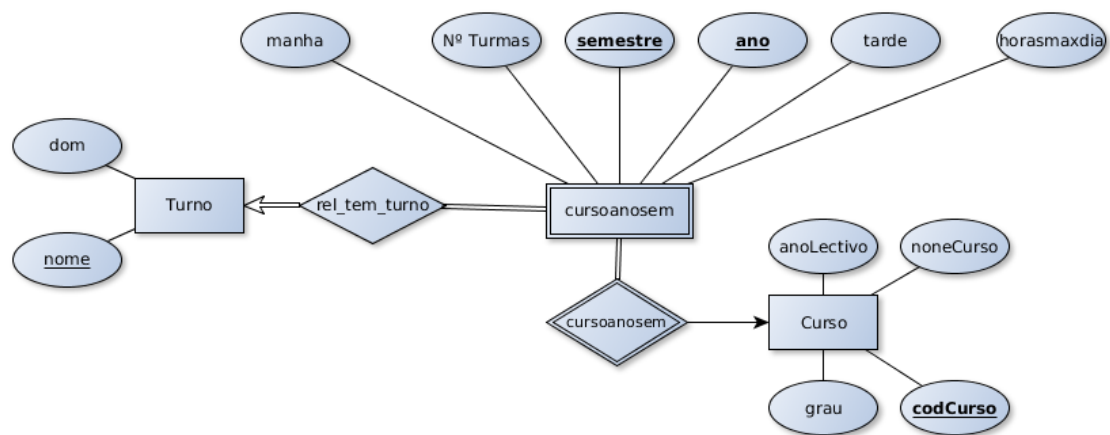


Figura 2: Representação do curso e preferências

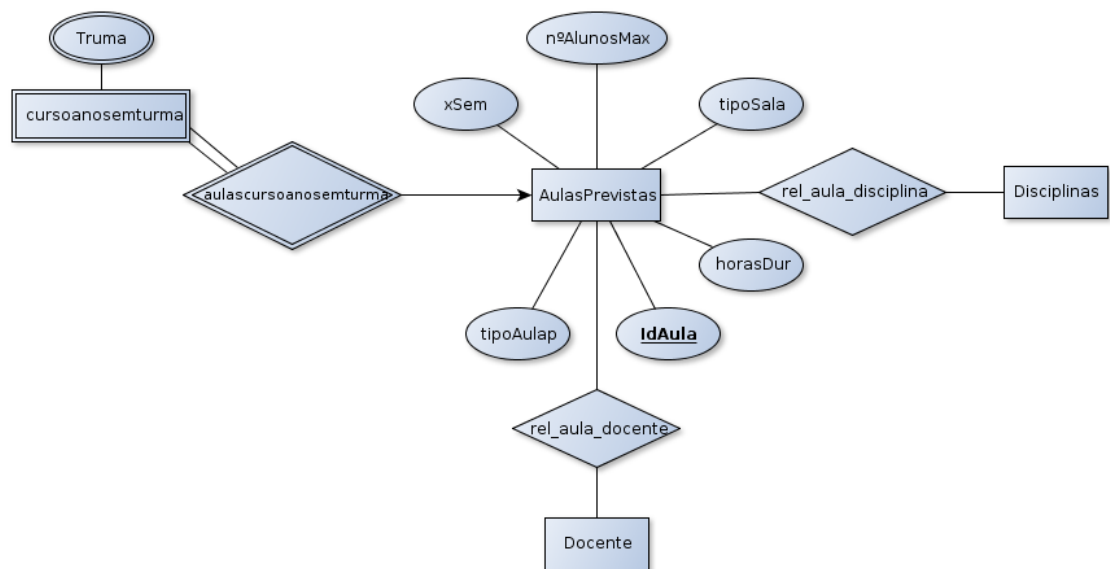


Figura 3: Representação da aula

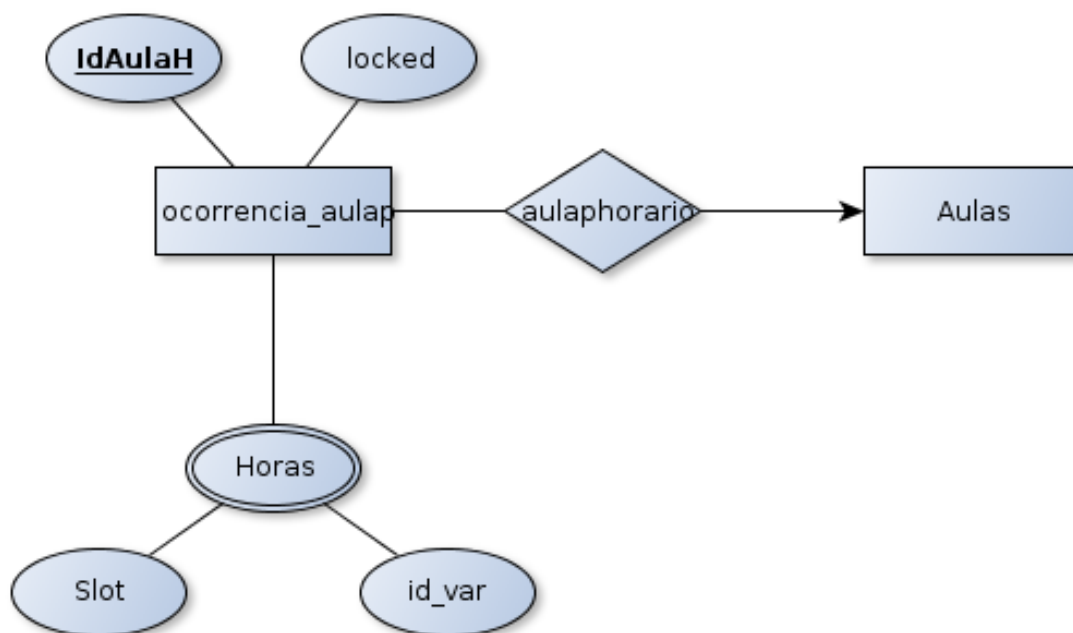


Figura 4: Representação do horário das aulas

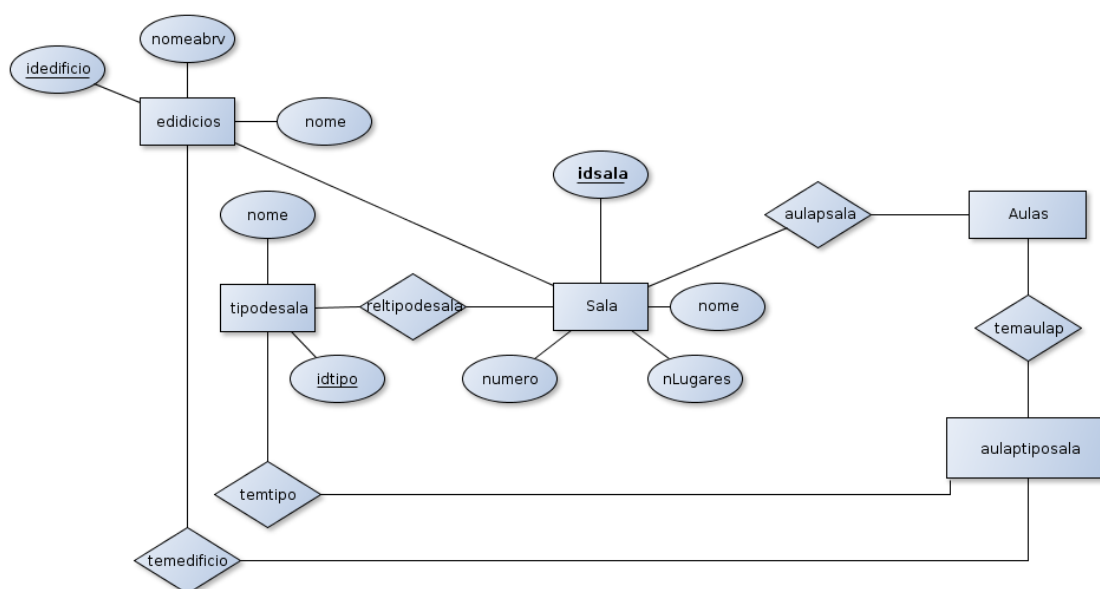


Figura 5: Preferências das aulas por salas

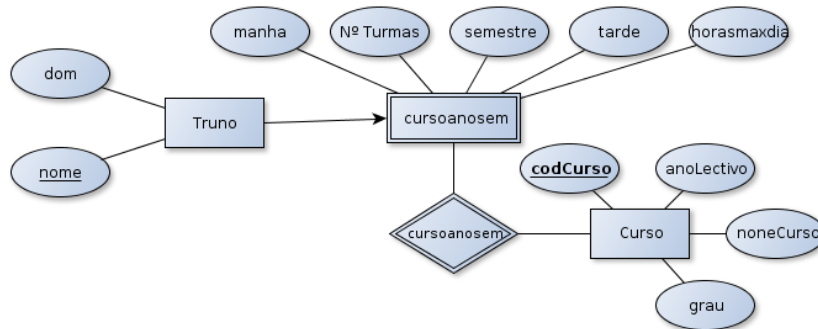


Figura 6: Preferências do curso/ ano /sem

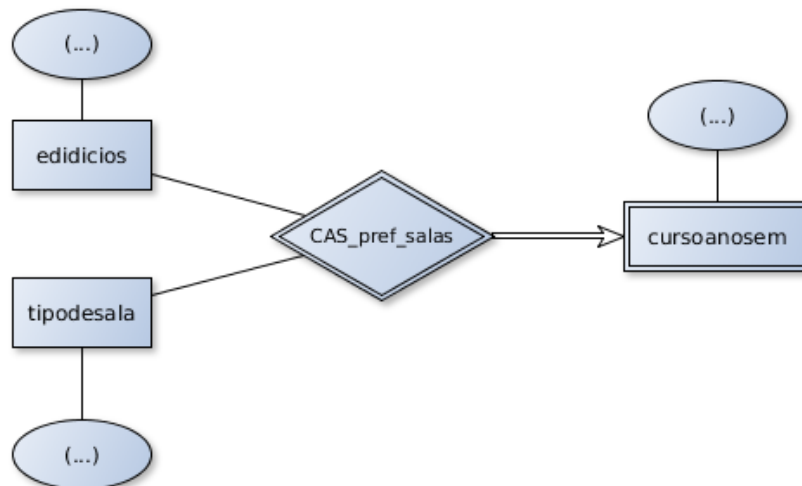



Figura 7: Preferências do curso/ ano /sem por salas



**Anexo**

**Horário**

**Engenharia Informática**  
**1º ano**  
**Semestre: Ímpar**  
**Turma: A,B,C,D**



**Universidade de Évora**  
**Ano Lectivo 2013/2014**

	Segunda	Terça	Quarta	Quinta	Sexta
08:00					
09:00		Física Geral I CLAV - 061-H Turma: B	Programação I CLAV - 139 Turma: C,D	Sistemas Digitais CLAV - 138 Turma: A	
10:00					Análise Matemática I CLAV - 130 Turma: A,B,C,D
11:00	Física Geral I CLAV - 061-E Turma: C	Análise Matemática I CLAV - 133 Turma: A,B	Sistemas Digitais CLAV - 139 Turma: C,D	Algebra Linear e Geometria Analítica I CLAV - 126 Turma: C,D	Inglês (I/INF) CLAV - 174 Turma: A,B,C,D
12:00					Física Geral I CLAV - 134 Turma: A,B,C,D
13:00					
14:00	Programação I CLAV - Anf.3 Turma: A,B,C,D	Sistemas Digitais CLAV - 129 Turma: A,B,C,D	Algebra Linear e Geometria Analítica I CLAV - 131 Turma: A,B	Análise Matemática I CLAV - Anf.3 Turma: A,B,C,D	
15:00					
16:00	Algebra Linear e Geometria Analítica I CLAV - Anf.2 Turma: A,B,C,D	Análise Matemática I CLAV - 131 Turma: C,D	Programação I CLAV - 137 Turma: B	Física Geral I CLAV - Anf.3 Turma: A,B,C,D	
17:00					
18:00	Física Geral I CLAV - 061-E Turma: D	Análise Matemática I CLAV - 129 Turma: X		Análise Matemática I CLAV - 129 Turma: X	
19:00					
20:00					
21:00					
22:00					

Gerado pelo SCAHEU

Figura 8: Horário gerado e exportado pelo sistema

**Anexo**

**Edição da DSD**

The screenshot shows a web application interface for editing the distribution of teaching services. At the top, there is a navigation bar with the following items: Horários UE, Editar Dados da DSD, Editar Espaços, Editar, Operações, Visualizar Horários, and Logout (Nuno). Below the navigation bar, there is a breadcrumb trail: [Página inicial](#) > [Editar a DSD](#). The main heading is "Editar a Distribuição de Serviço Docente". Below this heading is a form titled "Mostrar Aulas" with several dropdown menus and buttons. The form contains the following fields: Departamento, Disciplina, Disciplina, Curso, Ano, Semestre, Curso, Departamento, Docente, and Docente. At the bottom of the page, there is contact information: Contacto m91833@alunos.uevora.pt, Copyright © 2013, Todos os Direitos Reservados, and Powered by Yii Framework.

Horários UE Editar Dados da DSD Editar Espaços Editar Operações Visualizar Horários Logout (Nuno)

[Página inicial](#) [Editar a DSD](#)

### Editar a Distribuição de Serviço Docente

**Mostrar Aulas**

Departamento	Disciplina	Disciplina
Curso	Ano	Semestre
Curso		
Departamento	Docente	Docente

Contacto m91833@alunos.uevora.pt  
Copyright © 2013  
Todos os Direitos Reservados.  
Powered by Yii Framework.

Figura 9: Edição da DSD

Anexo

Preferências do Docente

Horários UE    Editar Dados da DSD    Editar Espaços    Editar    Operações    Visualizar Horários    Logout (Nuno)

[Página inicial](#) / [Lista de Docentes](#) / Preferências do Docente

## Preferência para o Docente

Número Máximo de Dias por Semana  
3

Número Máximo de Horas por Dia  
6

### Disponibilidade do Docente

	Segunda	Terça	Quarta	Quinta	Sexta	Sábado
08:00	08:00 to 00:00 Disponível	08:00 to 00:00 Disponível	08:00 to 00:00 Disponível	08:00 to 00:00 Disponível	08:00 to 00:00 Disponível	08:00 to 00:00 Disponível
09:00						
10:00						
11:00						
12:00						
13:00						
14:00						
15:00						
16:00						
17:00						
18:00						
19:00						
20:00						
21:00						
22:00						
23:00						

Figura 10: Preferências do Docente

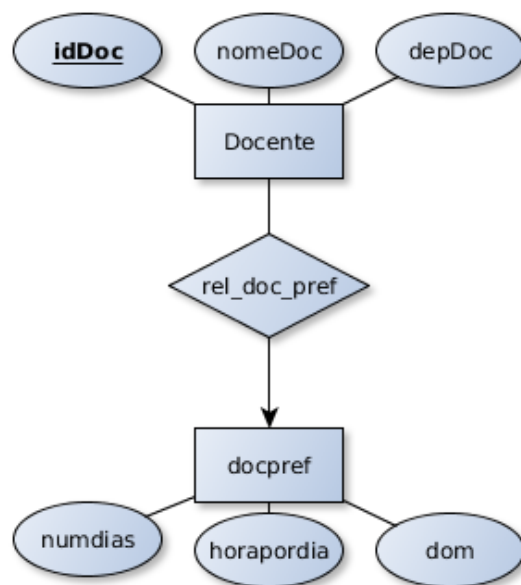


Figura 11: Diagrama das preferências do docente





**Anexo**

**Preferências do Curso**

## Preferências Engenharia Informática

Ano	Semestre	Número de Turmas	Horas Máximas Diárias	Manhã	Tarde	Turno
1	impar	<input type="text" value="4"/>	<input type="text" value="8"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Diurno
1	par	<input type="text" value="1"/>	<input type="text" value="8"/>	<input type="checkbox"/>	<input type="checkbox"/>	Diurno
2	impar	<input type="text" value="2"/>	<input type="text" value="8"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Diurno
2	par	<input type="text" value="1"/>	<input type="text" value="8"/>	<input type="checkbox"/>	<input type="checkbox"/>	Diurno
3	impar	<input type="text" value="2"/>	<input type="text" value="8"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Diurno
3	par	<input type="text" value="1"/>	<input type="text" value="8"/>	<input type="checkbox"/>	<input type="checkbox"/>	Diurno

[← Voltar](#)   [Gravar](#)

Contacto m9183@alunos.uevora.pt  
 Copyright © 2013  
 Todos os Direitos Reservados.  
 Powered by Yii Framework.

Figura 12: Preferências horária do curso

[Página inicial](#)   [Cursos](#)   Engenharia Informática

Preferência de Salas para o Curso *Engenharia Informática*

1º ano - semestre impar +

1º ano - semestre par +

Colégio Luis António Verney  Teóricas/Práticas (18)  ✖

2º ano - semestre impar +

2º ano - semestre par +

Colégio Luis António Verney  Teóricas/Práticas (18)  ✖

3º ano - semestre impar +

3º ano - semestre par +

Colégio Luis António Verney  Teóricas/Práticas (18)  ✖

[Gravar](#)

Contacto m9183@alunos.uevora.pt  
 Copyright © 2013  
 Todos os Direitos Reservados.  
 Powered by Yii Framework.

Figura 13: Preferências de Tipos de Salas do Curso

**Anexo**

**Salas**

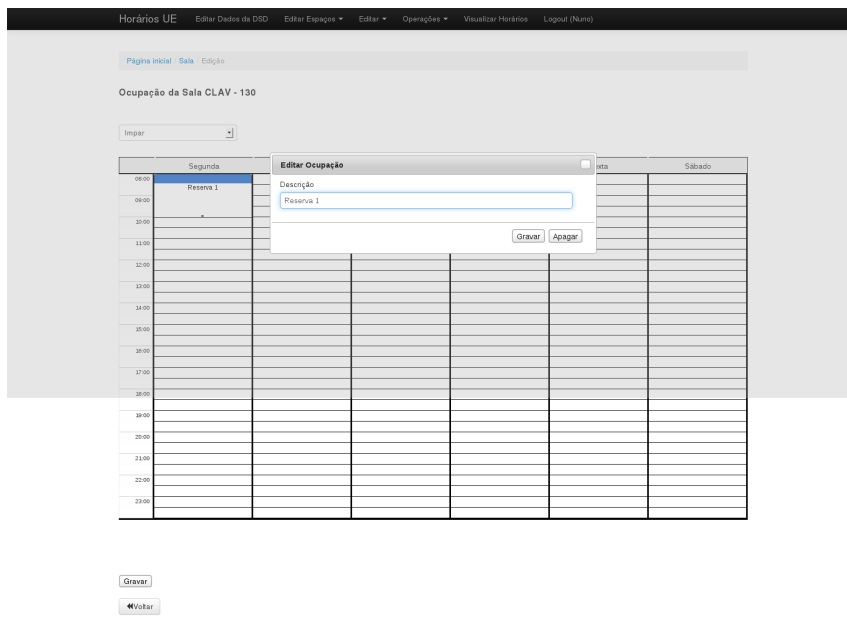


Figura 14: Reserva de uma sala

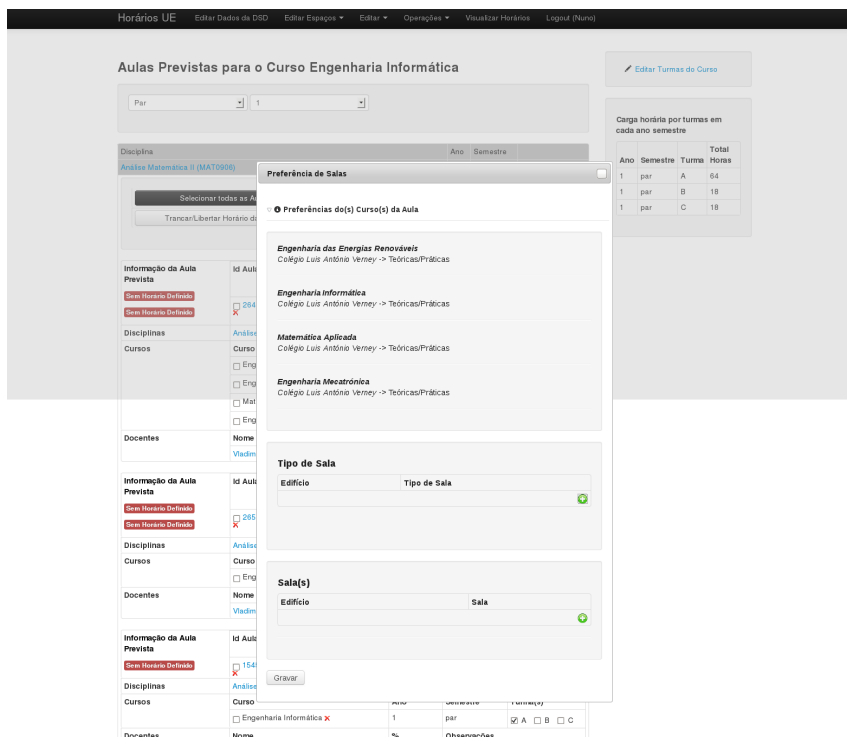


Figura 15: Preferência de Salas na aula

Anexo

Visualização

Horários UE    Editar Dados da DSD    Editar Espaços    Editar    Operações    Visualizar Horários    Logout (Nuno)

---


Página inicial | Visualizar Horário

**Curso**  
 Engenharia Informática (B\_EI...    1    impar    Select Some    Ver    Gerar    Limpar    Fixar/Liberar horario de todas as aulas

**Docente**  
 Selecionar Docente    Semestr...    Ver

**Sala**  
 Selecionar Edifício    Sala    Semestr...    Ver

---

**Engenharia Informática**  **Universidade de Évora**  
 1º ano    Ano Lectivo 2013/2014

**Semestre: Impar**

	Segunda	Terça	Quarta	Quinta	Sexta	Sábado
08:00						
09:00		✓ Física Geral I CLAV - 102.54H Turma: B	✓ Programação I CLAV - 139 Turma: C,D	✓ Sistemas Digitais CLAV - 138 Turma: A		✓ Análise Matemática I CLAV - 135 Turma: A,B,C,D
10:00						
11:00	✓ Física Geral I CLAV - 102.54 Turma: C	✓ Análise Matemática I CLAV - 135 Turma: A,B	✓ Álgebra Linear e Geometria Analítica I CLAV - 130 Turma: A	✓ Programação I CLAV - 139 Turma: E	✓ Inglês (ENF) CLAV - 174 Turma: A,B,C,D	✓ Física Geral I CLAV - 136 Turma: A,B,C,D
12:00						
13:00						
14:00	✓ Programação I CLAV - An3 Turma: A,B,C,D	✓ Sistemas Digitais CLAV - 138 Turma: A,B,C,D	✓ Álgebra Linear e Geometria Analítica I CLAV - 131	✓ Sistemas Digitais CLAV - 139 Turma: C,D	✓ Análise Matemática I CLAV - An3	
15:00						
16:00	✓ Álgebra Linear e Geometria Analítica I CLAV - An2 Turma: A,B,C,D	✓ Análise Matemática I CLAV - 131 Turma: C,D	✓ Física Geral I CLAV - 102.54 Turma: A	✓ Programação I CLAV - 137 Turma: B	✓ Física Geral I CLAV - An3 Turma: A,B,C,D	
17:00						
18:00	✓ Física Geral I CLAV - 102.54 Turma: D	✓ Análise Matemática I CLAV - 135			✓ Análise Matemática I CLAV - 139 Turma: X	
19:00						
20:00						
21:00						
22:00						
23:00						

[Imprimir](#)

---

Contacto m9183@alunos.uevora.pt  
 Copyright © 2013  
 Todos os Direitos Reservados.  
 Powered by [Yi Framework](#).

Figura 16: Visualização / edição do Horário