

Sistema de Gestão de Horários

Desenvolvimento de Sistemas de *Software* 24/25

Ana Cerqueira	A104188
Humberto Gomes	A104348
João Torres	A95748
José Lopes	A104541
José Matos	A100612

Departamento de Informática – Escola de Engenharia – Universidade do Minho
Licenciatura em Engenharia Informática

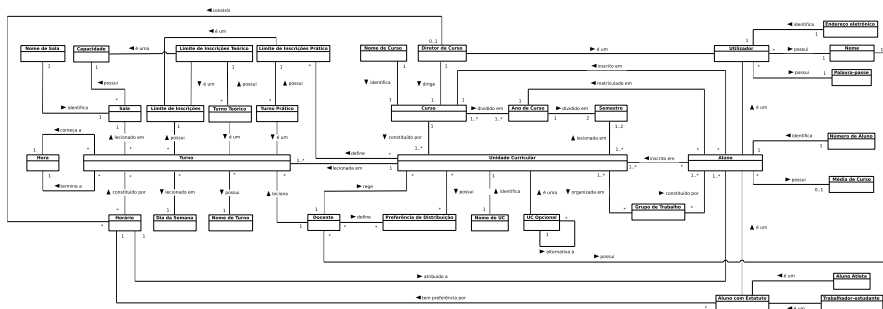
7 de janeiro de 2025

- 1 Análise de Requisitos
- 2 Modelação conceptual
- 3 Implementação
- 4 Apresentação do Produto Final
- 5 Conclusão

É pretendido o desenvolvimento de *software* onde:

- Diretores de curso sejam capazes de definir os horários dos alunos;
- Os alunos sejam capazes de consultar os seus horários.

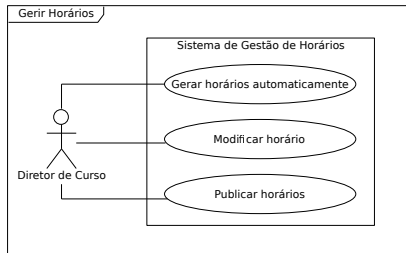
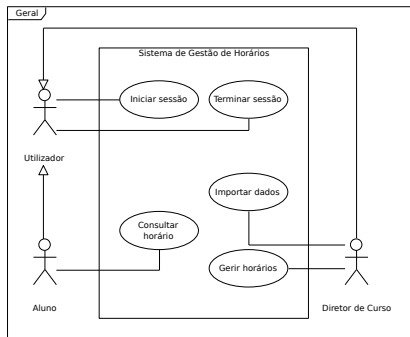
Análise de Requisitos – Modelo de Domínio



Modelo de Domínio construído.

- Construído com conhecimento próprio;
- Informação do enunciado para esclarecer detalhes;
- Definição de restrições sobre as relações.

Análise de Requisitos – Identificação dos Casos de Uso



Dois dos diagramas de casos de uso construídos.

- Identificação dos atores e das operações suportadas;
- Construção de diagramas.

Análise de Requisitos – Especificação dos Casos de Uso

Caso de uso	Iniciar sessão
Descrição	Ator autentica-se para poder realizar outras operações.
Cenários	Diretor de curso, Alunos
Pré-condição	<ul style="list-style-type: none">• Ator não tem sessão iniciada.
Pós-condição	<ul style="list-style-type: none">• Ator fica com sessão iniciada no sistema.
Fluxo normal	<ol style="list-style-type: none">1. Ator indica o seu endereço de email e palavra-passe.2. Sistema valida o endereço de email e a palavra-passe.3. Sistema inicia uma sessão para o ator.4. Sistema informa que a autenticação teve sucesso.
Fluxo de exceção 1	[combinação de email e palavra-passe incorreta] (passo 2) <ol style="list-style-type: none">2.1. Sistema informa que a autenticação não teve sucesso.

Exemplo de um caso de uso especificado.

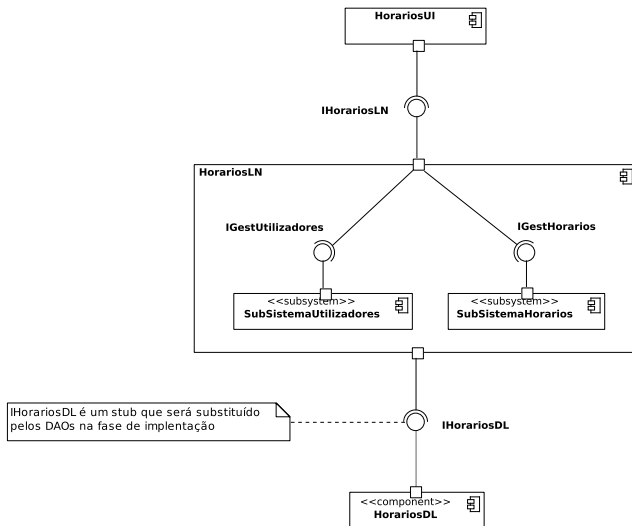
Modelação conceptual – Definição da API da Camada de Negócio

USE CASE:	Visualizar horário		
DESCRIÇÃO:	Ator consulta os turnos nos quais se encontra inscrito.		
CENÁRIOS:	Alunos	Identificar responsabilidades da LN	Definir API (Identificar operações)
PRÉ-CONDIÇÃO:	Ator tem sessão iniciada e o seu horário está disponível para consulta.		
PÓS-CONDIÇÃO:	O horário atribuído ao ator é-lhe apresentado.		
FLUXO NORMAL:			
	1. Ator solicita a consulta do seu horário.		
	2. Sistema consulta número de aluno do ator.	Obter número do aluno autenticado	obterNumeroAlunoAutenticado(): String
	3. Sistema apresenta o horário atribuído ao ator.	Obter horário	obterHorario(numeroAluno: String) : Map(String, Set(String))

Definição da API da camada de negócio para o caso de uso "Visualizar Horário".

- Identificação das responsabilidades da camada de negócio;
- Definição da sua API.

Modelação conceptual – Divisão das Operações por Subsistemas



Primeiro diagrama de componentes construído.

Modelação conceptual – Diagramas de Classe

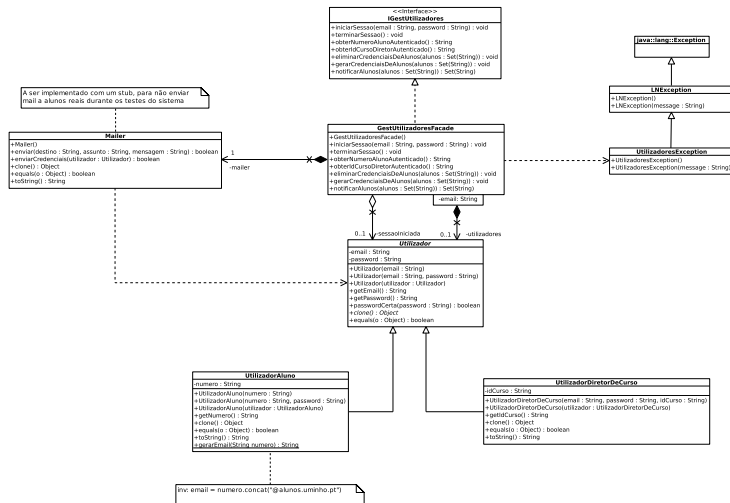


Diagrama de classes do subsistema dos utilizadores.

Modelação conceptual – Diagramas de Sequência

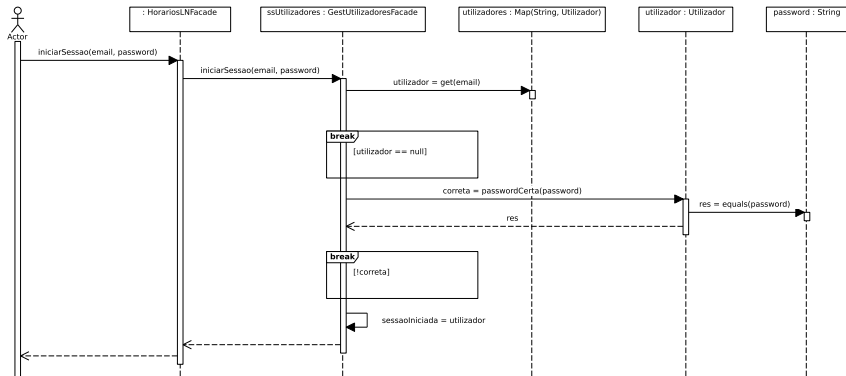


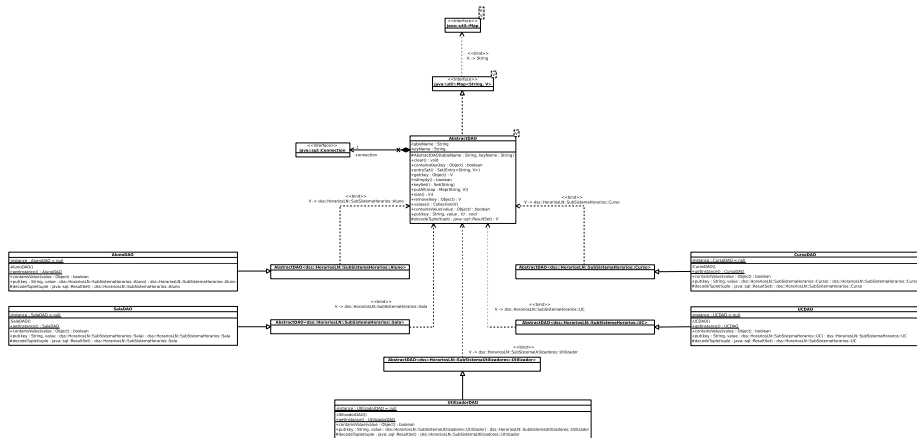
Diagrama de sequência da operação `iniciarSessao`.

- Modelação comportamental de todas as operações da camada de negócio.

Modelação conceptual – Geração Automática de Horários

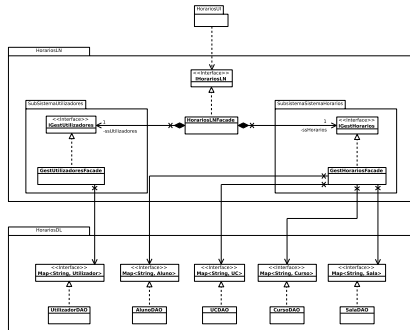
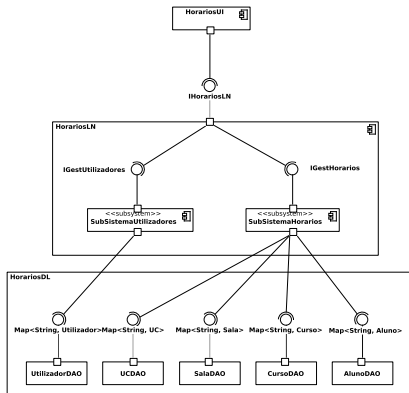
- Geração de um problema de programação inteira;
- Resolução um *solver* externo (*GNU Linear Programming Kit*);
- Protótipo descartável desenvolvido em Python.

Implementação – Adição de uma BD Relacional



Hierarquia do DAOs, utilizados para *Object-Relational Mapping*.

Implementação – Arquitetura final



Diagramas de componentes de *packages* finais.

Implementação – Alterações aos Diagramas de Classe

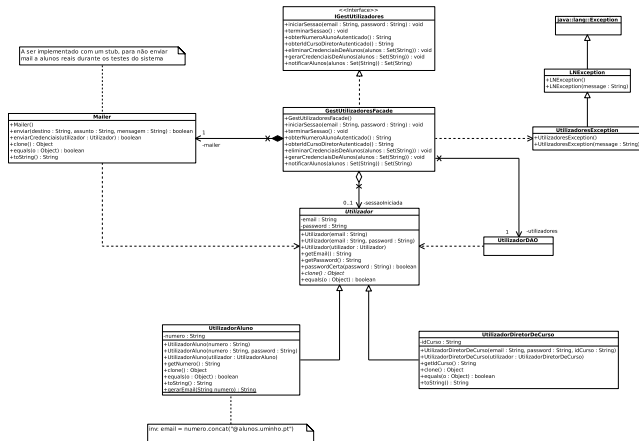


Diagrama de classes do subsistema dos utilizadores após adição de DAOs

- Substituição de associações qualificadas pelo uso dos DAOs;

Implementação – Alterações aos Diagramas de Sequência

- Adição de atualizações à base de dados (puts nos DAOs);
- Alteração a alguns métodos internos das classes;

Implementação – Camada de Apresentação

- Desenvolvimento da camada de apresentação na arquitetura MVC;
- Composição de um manual de instruções para os utilizadores.

```
Escolha uma opção ...
```

```
1 -> Reiniciar Semestre  
2 -> Importar Unidades Curriculares e Turnos  
3 -> Importar Alunos e Inscrições  
4 -> Adicionar Aluno  
5 -> Gerar Horários  
6 -> Modificar Horário  
7 -> Publicar Horários  
8 -> Terminar sessão
```

```
Opção > 
```

Menu principal para diretores de curso.

Implementação – Diagrama de Instalação

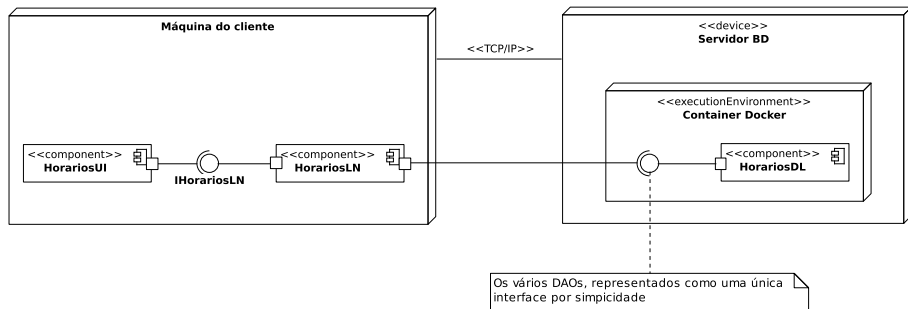


Diagrama de instalação construído.

- A ausência de tipos complexos permite uma fácil futura transferência da LN para outra máquina.
- A independência entre os subsistema permite uma fácil futura transição para uma arquitetura de micro-serviços.

Apresentação do Produto Final

Conclusão

Apesar de algumas dificuldades (divisão em subsistemas, implementação de preferências das UCs e adição de DAOs), sentimos que conseguimos desenvolver um sistema capaz de usar em ambientes reais e pelo próprio professor António Sousa.

Sistema de Gestão de Horários

Desenvolvimento de Sistemas de *Software* 24/25

Ana Cerqueira	A104188
Humberto Gomes	A104348
João Torres	A95748
José Lopes	A104541
José Matos	A100612

Departamento de Informática – Escola de Engenharia – Universidade do Minho
Licenciatura em Engenharia Informática

7 de janeiro de 2025