

ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO

INSTITUTO POLITÉCNICO DA GUARDA

# Gestão de Avaliações

R E L ATÓ R I O D O T R A B A L H O P R ÁT I C O

|  |
| --- |
| **Curso**  **Unidade Curricular**  **Ano Letivo**  **Docente**  **Data**  **Alunos** |

Engenharia Informática Engenharia de Software II   
2017/2018   
Maria Clara Silveira  
15/09/2018   
Fábio Martins nº1012502

Conteúdo

[Gestão de Avaliações 1](#_Toc534295009)

[Introdução 3](#_Toc534295010)

[Descrição do tema do projeto 4](#_Toc534295011)

[Diagrama de Contexto 5](#_Toc534295012)

[Padrões de Desenvolvimento 6](#_Toc534295013)

[Estado de Arte 7](#_Toc534295014)

[Tabela de atores, objetivos e respetivos casos de uso 8](#_Toc534295015)

[Diagrama de Casos de Uso 9](#_Toc534295016)

[Descrição de Casos de Uso 11](#_Toc534295017)

[Diagrama de Sequência dos Casos de Uso dentro da fronteira 12](#_Toc534295018)

[Diagrama de Classes 13](#_Toc534295019)

[Diagrama de Estados 14](#_Toc534295020)

[Diagrama de Pacotes 15](#_Toc534295021)

[Diagrama de Instalação 15](#_Toc534295022)

[Diagrama de Componentes 18](#_Toc534295023)

[Diagrama de Atividades 19](#_Toc534295024)

[Semântica de Classes 20](#_Toc534295025)

[Conclusão 21](#_Toc534295026)

[Anexos 22](#_Toc534295027)

## Introdução

Este projeto assenta em ajudar na elaboração dos horários para as frequências, exames, exames de recurso e época especial do curso de Engenharia Informática. Pretende assim ser uma ajuda tanto para os professores como para o administrador, a marcação deste tipo de avaliações para que não haja sobreposição de provas.

O projeto é concebido no ceio de duas unidades curriculares, Engenharia de Software II e Programação para a Internet, onde em concordância uma com a outra, estudamos o desenvolvimento do software em si, desde os métodos como se desenvolve, especificamente o método de desenvolvimento Ágil ate a elaboração de diversos casos de uso que apos serem aplicados os vários padrões de desenvolvimento de casos de uso resulta uma otimização das tarefas a serem concebidas num website desenvolvido com o modelo MVC em ASP.NET Core.

## Descrição do tema do projeto

### Neste projeto inicialmente irá existir autenticação, sendo atribuídas os seguintes estatutos: Administrador, Professor, Aluno.

### O aluno entra no site (a definir) e poderá ver os horários do curso de informática, e algumas informações gerais sobre o curso (ects´s, etc…). Caso o aluno efetue o login, poderá editar o seu perfil e terá acesso às disciplinas em que se encontra matriculado e informações sobre as datas das avaliações.

### O professor após efetuar o login, pode editar o seu perfil assim como pode também marcar avaliações às disciplinas no qual é docente.

### O administrador, pode gerir todas as cadeiras existentes no curso, gerir as provas de avaliação e controlar os possíveis erros das sobreposições das avaliações, seja por causa do ano das respetivas cadeiras, seja por causa do horário, resumindo, poderá gerir de forma automática todas as épocas (época de frequências, exames, exames de recurso e época especial).

## Diagrama de Contexto

O diagrama de contexto é composto por fluxos de dados( que simbolizam interfaces) entre o sistema e entidades externas, permitindo assi, identificar os limites dos processos, as áreas envolvidas com o processo, os relacionamentos com outros processos, elementos externos representando assim o sistema como um único processo.

O diagrama que se segue ilustra o diagrama de contexto que traduz a solução para o problema apresentado. Neste caso como entidades externas temos o Administrador, Professor e Aluno.

## Padrões de Desenvolvimento

Two TierReview

#### Hoje em dia, com a complexidade dos projetos, a necessidade de rever todos os casos de uso é enorme, pois trará benefícios económicos como a diminuição do tempo de execução.

#### As revisões ao longo de um projeto ajudam a validar, verificar e avaliar todo o processo de desenvolvimento.

#### Estas revisões não podem ser feitas por quem as executa, pois é muito difícil para os developers encontrarem erros no seu trabalho, pois estão muito familiarizados com todo o trabalho que desenvolveram. Portanto esta revisão é melhor ser feita por quem nunca olhou para aquilo, pois será mais fácil detetar erros e inconsistências.

#### Este padrão de caso de uso explica assim deverá existir revisões ao longo do projeto e que devem ser divididas por dois tipos de revisão:

#### -Várias revisões ao longo do projeto, com uma equipa pequena (3 a 4 elementos) e interna;

#### -Pelo menos uma revisão em que estão presentes toda a equipa envolvida no projeto, de modo a verificar todos os casos de uso, para se saber se estão todos bem descritos e fáceis de perceber, se são fáceis de implementar e se são precisos.

#### Depois desta revisão, se tudo estiver bem, então a equipa chegarão à conclusão que se encontra em QuittingTime(se os casos de uso responderem ás três questões fundamentais deste padrão).

#### 

## Estado de Arte

## Tabela de atores, objetivos e respetivos casos de uso

|  |  |  |
| --- | --- | --- |
| **Ator** | **Casos de Uso** | **Objetivos** |
| Administrador | Criar Professor | O administrador cria uma conta para cada professor |
| Modificar professor | O administrador pode alterar qualquer informação de um professor |
| Inativar Professor | o administrador pode inativar uma conta de um professor |
| Consultar Professor | O administrador pode consultar a conta de qualquer professor |
| Criar Aluno | O administrador cria uma conta para cada aluno |
| Modificar Aluno | O administrador pode modificar uma conta de aluno |
| Inativar Aluno | O administrador pode inativar uma conta de um aluno |
| Consultar Aluno | O administrador pode consultar uma conta de um aluno |
| Gerar Newsletter | O administrador gera uma newsletter e envia para todos os alunos/professores |
| Criar horário | O administrador pode adicionar disciplinas ao horário |
| Modificar horário | O administrador pode apagar disciplinas do horário |
| Professor | Consultar Conta | O professor pode consultar a sua conta |
| Consultar Horários | O professor pode consultar o horário |
| Reportar um Erro/Aviso | O professor pode reportar erros sobre o horário ao administrador |
| Aluno | Consultar Conta | O aluno pode consultar a sua conta |
| Consultar horário | O aluno pode consultar horário |

## Diagrama de Casos de Uso

Diagrama de casos candidatos e dentro da fronteira

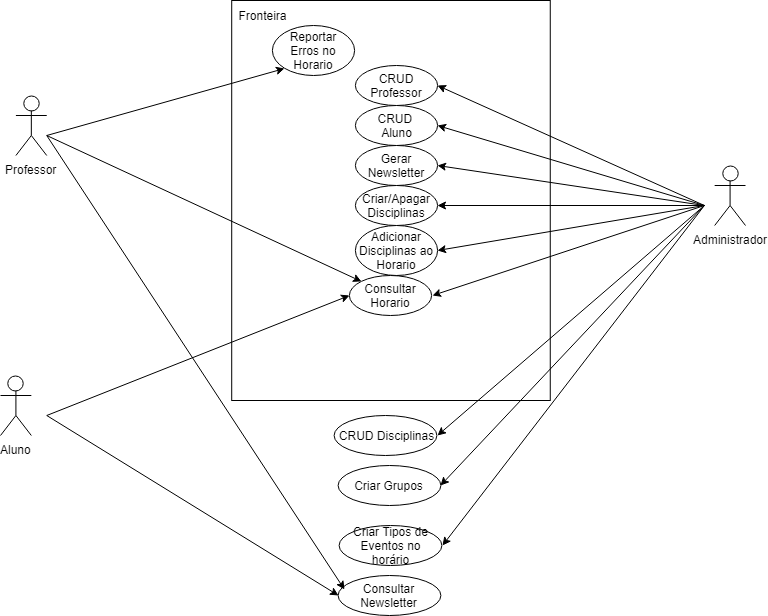
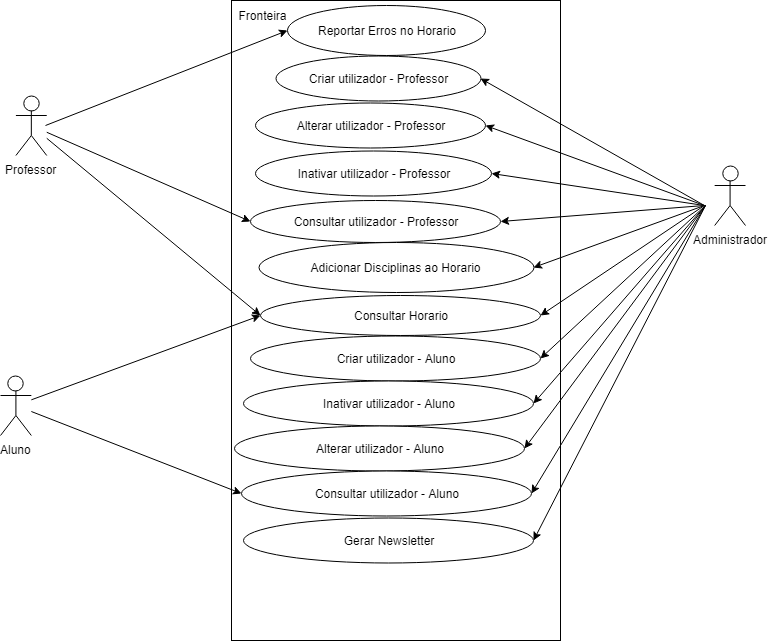


Diagrama de casos dentro da fronteira



## Descrição de Casos de Uso

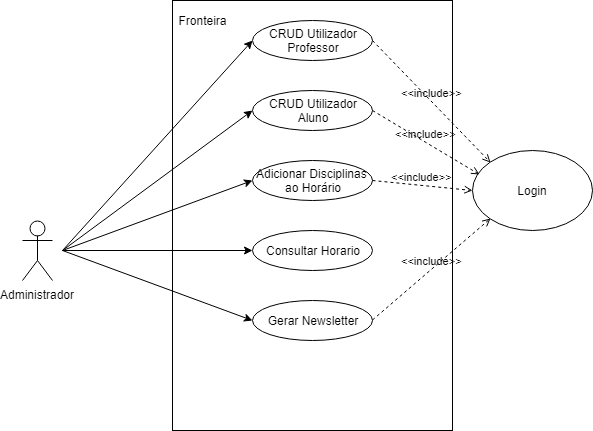
## Diagrama de Sequência dos Casos de Uso dentro da fronteira

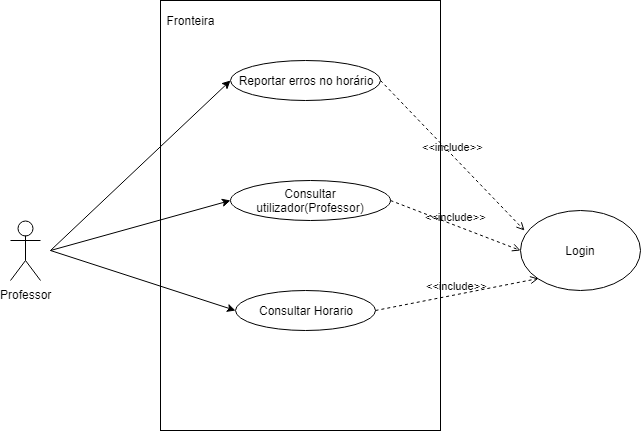
## Diagrama de Classes

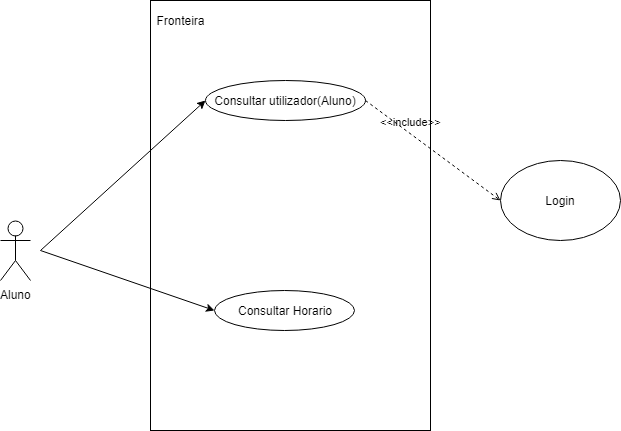
## Diagrama de Estados

## Diagrama de Pacotes

Diagrama de Pacotes-Atores







## Diagrama de Instalação

## Diagrama de Componentes

## Diagrama de Atividades

## Semântica de Classes

## Conclusão

## Anexos

Diário de Atividades

Prototipagem

Algoritmo do Digito de Controlo

#### Bilhete de Identidade e Cartão do Cidadão

#### MODO DE CÁLCULO:

#### Se o número tiver só 6 ou 7 dígitos acrescente 0 (zeros) à esquerda até perfazer 8 dígitos

#### MULTIPLIQUE O

#### 

#### 8.º dígito por 2

#### 7.º dígito por 3

#### 6.º dígito por 4

#### 5.º dígito por 5

#### 4.º dígito por 6

#### 3.º dígito por 7

#### 2.º digito por 8

#### 1.º dígito por 9

#### 

#### Adicione os resultados

#### Calcule o Módulo 11 do resultado, isto é, o resto da divisão do número por 11.

#### Se o resto for 0 ou 1, o dígito de controle será 0

#### Se for outro algarismo x, o dígito de controle será o resultado de 11 - x

#### public bool ValidateNumeroDocumentoCC(string numeroDocumento) { int sum = 0; bool secondDigit = false;

#### 

#### if(numeroDocumento.Length != 12) throw new ArgumentException("Tamanho inválido para número de documento.");

#### for (int i = numeroDocumento.Length-1; i >= 0; --i) { int valor = GetNumberFromChar(numeroDocumento[i]);

#### 

#### if (secondDigit) { valor \*= 2;

#### 

#### if (valor > 9) valor -= 9; }

#### 

#### sum += valor; secondDigit = !secondDigit;

#### }

#### 

#### return (sum % 10) == 0; }

#### 

#### public int GetNumberFromChar(char letter) { switch(letter) { case '0' : return 0; case '1' : return 1;

#### case '2' : return 2;

#### case '3' : return 3;

#### case '4' : return 4;

#### case '5' : return 5;

#### case '6' : return 6;

#### case '7' : return 7;

#### case '8' : return 8;

#### case '9' : return 9;

#### case 'A' : return 10;

#### case 'B' : return 11;

#### case 'C' : return 12;

#### case 'D' : return 13;

#### case 'E' : return 14;

#### case 'F' : return 15;

#### case 'G' : return 16;

#### case 'H' : return 17;

#### case 'I' : return 18;

#### case 'J' : return 19;

#### case 'K' : return 20;

#### case 'L' : return 21;

#### case 'M' : return 22;

#### case 'N' : return 23;

#### case 'O' : return 24;

#### case 'P' : return 25;

#### case 'Q' : return 26;

#### case 'R' : return 27;

#### case 'S' : return 28;

#### case 'T' : return 29;

#### case 'U' : return 30;

#### case 'V' : return 31;

#### case 'W' : return 32;

#### case 'X' : return 33;

#### case 'Y' : return 34;

#### case 'Z' : return 35;

#### }

#### 

#### throw new ArgumentException("Valor inválido no número de documento.");