

ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO

INSTITUTO POLITÉCNICO DA GUARDA

# Gestão de Horários das Provas

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   
2018/2019   
Maria Clara Silveira  
15/09/2018   
Fábio Martins nº1012502

Conteúdo

[Gestão de Horários das Provas 1](#_Toc536524133)

[Introdução 3](#_Toc536524134)

[Descrição do tema do projeto 4](#_Toc536524135)

[Diagrama de Contexto 5](#_Toc536524136)

[Padrões de Desenvolvimento 6](#_Toc536524137)

[Estado de Arte 7](#_Toc536524138)

[Tabela de atores, objetivos e respetivos casos de uso 8](#_Toc536524139)

[Diagrama de Casos de Uso 9](#_Toc536524140)

[Descrição de Casos de Uso 11](#_Toc536524141)

[Diagrama de Sequência dos Casos de Uso dentro da fronteira 22](#_Toc536524142)

[Diagrama de Classes 25](#_Toc536524143)

[Diagrama de Estados 26](#_Toc536524144)

[Diagrama de Pacotes 27](#_Toc536524145)

[Diagrama de Instalação 29](#_Toc536524146)

[Diagrama de Componentes 30](#_Toc536524147)

[30](#_Toc536524148)

[Diagrama de Atividades 32](#_Toc536524149)

[Semântica de Classes 33](#_Toc536524150)

[Classe Professor 33](#_Toc536524151)

[Classe Estudante 34](#_Toc536524152)

[Classe Horario 35](#_Toc536524153)

[Classe Disciplina 36](#_Toc536524154)

[Conclusão 37](#_Toc536524155)

[Anexos 38](#_Toc536524156)

## 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, Estudante.

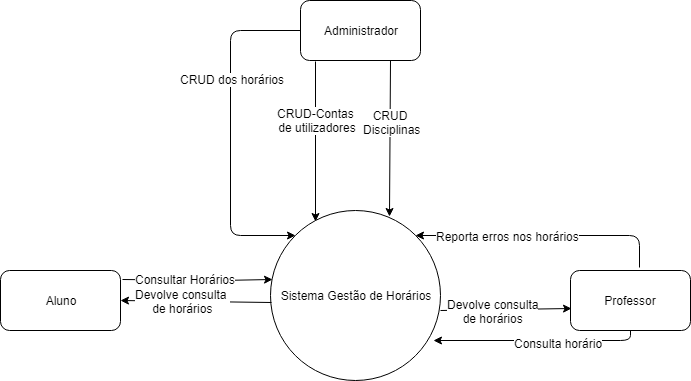
### O estudante entra no site e poderá ver os horários do curso de informática, após login com sucesso.

### O professor após efetuar o login poderá ver os horários do curso de informática.

### 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). O administrador tem a função de criar as contas de utilizadores(estudantes/professor) bem como as disciplinas existentes no curso de engenharia Informática.

## 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).

## 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 |
| Apagar Aluno | O administrador pode inativar uma conta de um aluno |
| Consultar Aluno | O administrador pode consultar uma conta de um aluno |
| Criar horário | O administrador pode adicionar horários de provas |
| Modificar horário | O administrador pode alterar horários de provas |
| Apagar horário | O administrador pode apagar horários de provas |
| Consultar horário | O administrador pode consultar horários de provas |
| Criar disciplina | O administrador pode adicionar disciplinas |
| Modificar disciplina | O administrador pode alterar disciplinas existentes |
| Apagar disciplina | O administrador pode apagar disciplinas |
| Consultar disciplina | O administrador pode consultar disciplinas |
| Professor | Consultar Horário | O professor pode consultar os horários |
| Estudante | Consultar Horário | O estudante pode consultar os horários |

## Diagrama de Casos de Uso

Diagrama de casos candidatos e dentro da fronteira

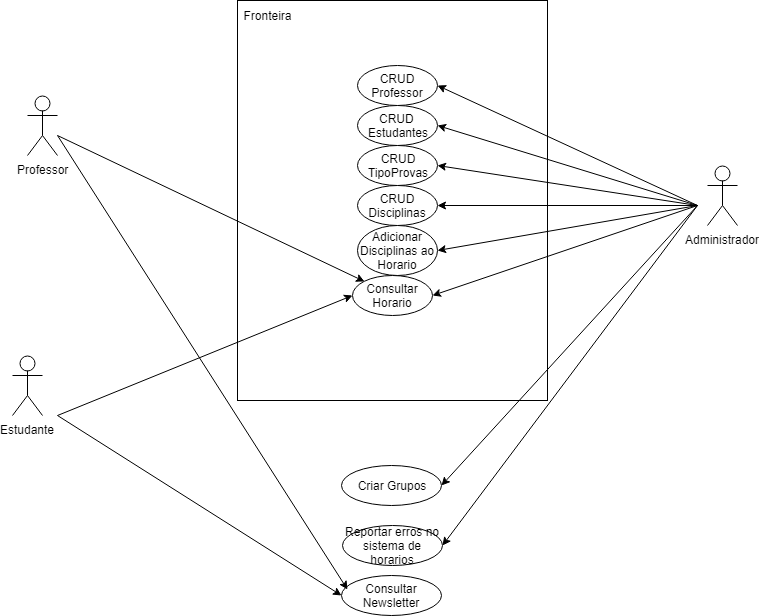
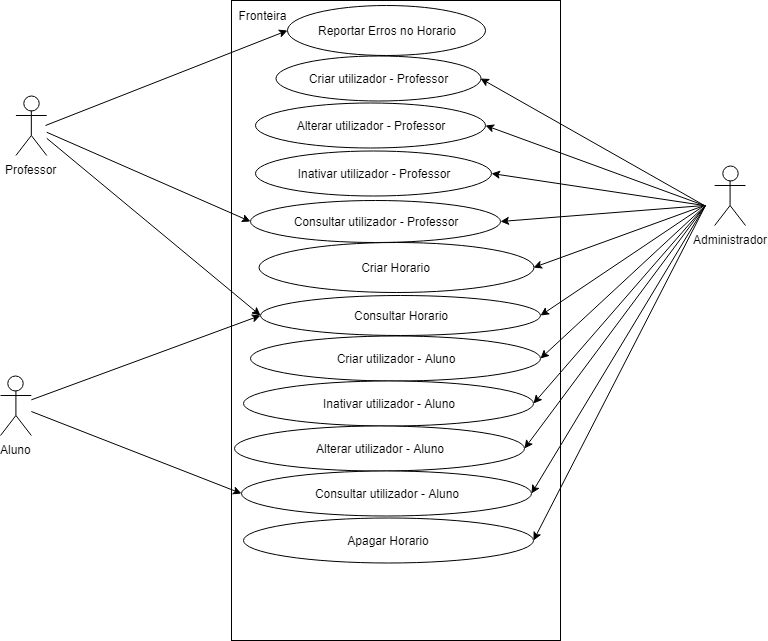


Diagrama de casos dentro da fronteira



## Descrição de Casos de Uso

|  |  |
| --- | --- |
| Nome do caso de uso | RF01, Autenticar Ator (Login) |
| Descrição do requisito | O objetivo deste caso de uso é controlar o acesso ao sistema de software |
| Tamanho | 21 |
| Pré-condição |  |
| Caminho Principal |  |
|  | 1-Acede ao software.  2-Pede a sua identificação (username e password).  3-Introduz username e password.  4-Apresenta o horário com os eventos. |
| Caminhos Alternativos | 3a-Se o username de utilizador estiver errado o sistema apresenta a mensagem de erro”Username ou password inválida”  3b- Se a password de utilizador estiver errado o sistema apresenta a mensagem de erro ”Username ou password inválida” |
| Suplementos ou Adornos |  |

|  |  |
| --- | --- |
| Nome do caso de uso | RF02, Adicionar prova no horário |
| Descrição do requisito | O objetivo deste caso de uso é a criação de eventos(provas). |
| Tamanho | 21 |
| Pré-condição | Ser admin e estar com login válido. |
| Caminho Principal |  |
|  | 1-O Administrador(ator) seleciona a opção “Marcação”  2-O sistema abre uma página com um calendário  3-O ator clica no dia em que pretende marcar a prova  4-O sistema devolve um formulário para criar evento com os campos: TipoProva, DataInicio,DiaInteiro,DataFim,Descricao,Ano  5-O ator preenche e clica em “Guardar”  6-O sistema cria esse evento e adiciona á tabela horário com um ID  7-É mostrado uma mensagem de sucesso |
| Caminhos Alternativos | 5a-Se DataInicio não for inserido o sistema devolve a seguinte mensagem:”Campo Data Inicio inválido!”  5b-Se DataFim não for inserido o sistema devolve a seguinte mensagem:”Campo Data Fim inválido!”  5c-Se Descricao não for inserido o sistema devolve a seguinte mensagem:”Campo descrição inválido!” |
| Suplementos ou Adornos |  |

|  |  |
| --- | --- |
| Nome do caso de uso | RF03, Criar User |
| Descrição do requisito | O objetivo deste caso de uso é criar novo User |
| Tamanho | 8 |
| Pré-condição | O Administrador tem o login válido. |
| Caminho Principal |  |
|  | 1-O diretor seleciona a opção “Criar Estudante”  2-O sistema abre o formulário com os dados a introduzir  3-O administrador insere os atributos nome, email, numero de estudante ano, estado do estudante  4-O administrador carrega na opção guardar  5-O sistema atribui automaticamente o ID ao novo estudante.  6-O sistema dá sucesso. |
| Caminhos Alternativos | 2a-O sistema tem opção de voltar à lista  3a-Se o atributo nome estiver nulo o sistema pede para inserir um nome.  3b- Se o atributo nome estiver incorreto o sistema pede para inserir um nome válido  3c- Se o atributo email estiver nulo o sistema pede para inserir um email.  3d- Se o atributo email estiver incorreto o sistema pede para inserir um email válido. |
| Suplementos ou Adornos | * Testar se o sistema deixa registar estudantes/professores sem os campos obrigatórios estarem preenchidos.   Teste1-Validar email  Utilizador insere no Email = asdasdadsa  O sistema avisa que email não é válido, necessita de um @.   * Teste2– Validação do Nome   Utilizador insere – Nome = Fabio12  O sistema notifica – O Nome apenas pode conter caracteres. |

|  |  |
| --- | --- |
| Nome do caso de uso | RF04, Visualizar Disciplinas |
| Descrição do requisito | O objetivo deste caso de uso visualizar todas as disciplinas |
| Tamanho | 7 |
| Pré-condição | O Administrador tem o login válido. |
| Caminho Principal |  |
|  | 1-O diretor seleciona a opção “Visualizar Disciplinas”  2-O sistema devolve a pagina com todas as disciplinas existentes  3-Essa mesma página(índex) contem opções para pesquisar e ordenação, assim como as opções de criar, editar e eliminar disciplina. |
| Caminhos Alternativos | 2a-O sistema tem opção de voltar à lista |
| Suplementos ou Adornos |  |

|  |  |
| --- | --- |
| Nome do caso de uso | RF05, Criar Disciplinas |
| Descrição do requisito | O objetivo deste caso de uso é criar uma disciplina |
| Tamanho | 17 |
| Pré-condição | O Administrador tem o login válido. |
| Caminho Principal |  |
|  | 1-O diretor seleciona a opção “Criar Disciplinas”  2-O sistema devolve a pagina para inserção de todos os campos para a criar de uma disciplina: Nome, ano e professor.  3-O administrador insere os dados e clica em “Criar”  4-O sistema redirecciona o user para a pagina índex das disciplinas |
| Caminhos Alternativos | 2a-Se não for inserido um Nome, o sistema mostra uma mensagem de erro:”Campo Nome inválido!”  2b- Se não for inserido um Ano, o sistema mostra uma mensagem de erro:”Campo Ano inválido!”  2c-Se não for inserido um Professor, o sistema mostra uma mensagem de erro:”Campo Professor inválido!” |
| Suplementos ou Adornos |  |

|  |  |
| --- | --- |
| Nome do caso de uso | RF06, Visualizar Estudantes |
| Descrição do requisito | O objetivo deste caso de uso visualizar todas os Estudantes |
| Tamanho | 7 |
| Pré-condição | O Administrador tem o login válido. |
| Caminho Principal |  |
|  | 1-O diretor seleciona a opção “Visualizar Estudantes”  2-O sistema devolve a pagina com todos os estudantes existentes  3-Essa mesma página(índex) contem opções para pesquisar e ordenação, assim como as opções de criar, editar e eliminar Estudante. |
| Caminhos Alternativos | 2a-O sistema tem opção de voltar à lista |
| Suplementos ou Adornos |  |

|  |  |
| --- | --- |
| Nome do caso de uso | RF07, Visualizar Professores |
| Descrição do requisito | O objetivo deste caso de uso visualizar todos os Professores |
| Tamanho | 7 |
| Pré-condição | O Administrador tem o login válido. |
| Caminho Principal |  |
|  | 1-O diretor seleciona a opção “Visualizar Professores”  2-O sistema devolve a pagina com todas as disciplinas existentes  3-Essa mesma página(índex) contem opções para pesquisar e ordenação, assim como as opções de criar, editar e eliminar Professores. |
| Caminhos Alternativos | 2a-O sistema tem opção de voltar à lista |
| Suplementos ou Adornos |  |

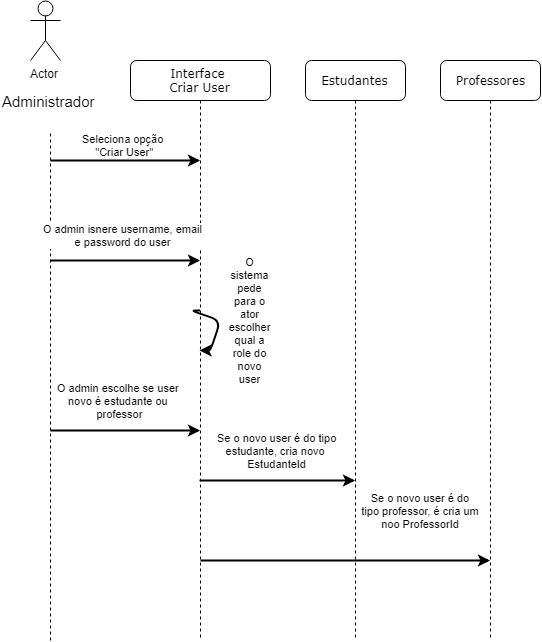
|  |  |
| --- | --- |
| Nome do caso de uso | RF08, Visualizar Tipos Prova |
| Descrição do requisito | O objetivo deste caso de uso visualizar todos os Tipos de Prova |
| Tamanho | 7 |
| Pré-condição | O Administrador tem o login válido. |
| Caminho Principal |  |
|  | 1-O diretor seleciona a opção “Visualizar Tipos Prova”  2-O sistema devolve a pagina com todos os TipoProva existentes  3-Essa mesma página(índex) contem opções para pesquisar e ordenação, assim como as opções de criar, editar e eliminar Tipos de Prova. |
| Caminhos Alternativos | 2a-O sistema tem opção de voltar à lista |
| Suplementos ou Adornos |  |

|  |  |
| --- | --- |
| Nome do caso de uso | RF09, Criar TiposProva |
| Descrição do requisito | O objetivo deste caso de uso Criar Tipos Prova |
| Tamanho | 17 |
| Pré-condição | O Administrador tem o login válido. |
| Caminho Principal |  |
|  | 1-O diretor seleciona a opção “Criar Tipos Prova”  2-O sistema devolve um formulário para a criar de tipos prova com os campos: Nome, Descrição.  3-O administrador insere dados nesses campos e clica em “Criar”  4-O sistema redireciona para a pagina Índex dos Tipo Prova. |
| Caminhos Alternativos | 2a- Se não for inserido um Nome, o sistema mostra uma mensagem de erro:”Campo Nome inválido!”  2b- Se não for inserido um Descricao, o sistema mostra uma mensagem de erro:”Campo Descricao inválido!” |
| Suplementos ou Adornos |  |

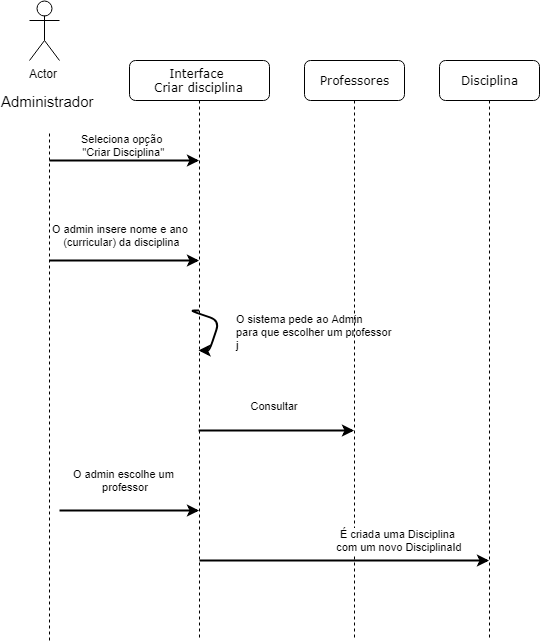
|  |  |
| --- | --- |
| Nome do caso de uso | RF10, Visualizar Horarios |
| Descrição do requisito | O objetivo deste caso de uso visualizar todos os Horarios das Provas |
| Tamanho | 7 |
| Pré-condição | O Administrador tem o login válido. |
| Caminho Principal |  |
|  | 1-O diretor seleciona a opção “Listar Provas”  2-O sistema devolve a pagina com todos os Horarios existentes  3-Essa mesma página(índex) contem opções para pesquisar e ordenação, assim como as opções de criar, editar e eliminar Horarios. |
| Caminhos Alternativos | 2a-O sistema tem opção de voltar à lista |
| Suplementos ou Adornos |  |

|  |  |
| --- | --- |
| Nome do caso de uso | RF11, Criar Horario |
| Descrição do requisito | O objetivo deste caso de uso visualizar todos os Tipos de Prova |
| Tamanho | 17 |
| Pré-condição | O Administrador tem o login válido. |
| Caminho Principal |  |
|  | 1-O diretor seleciona a opção “Criar Prova”  2-O sistema devolve um formulário para criar prova com os campos: Tipo Prova, DataInicio,DiaInteiro(bool),DataFim, Descricao, Ano.  3-O administrador insere os dados nos respetivos campos e clica em “criar”  4-O sistema redireciona para a pagina índex de horários. |
| Caminhos Alternativos | 2a- Se não for inserido um TipoProva, o sistema mostra uma mensagem de erro:”Campo TipoProva inválido!”  2b- Se não for inserido um DataInicio, o sistema mostra uma mensagem de erro:”Campo DataInicio inválido!”  2a- Se não for inserido um DataFim, o sistema mostra uma mensagem de erro:”Campo DataFim inválido!”  2a- Se não for inserido um Descricao, o sistema mostra uma mensagem de erro:”Campo Descricao inválido!”  2a- Se não for inserido um Ano, o sistema mostra uma mensagem de erro:”Campo Ano inválido!” |
| Suplementos ou Adornos |  |

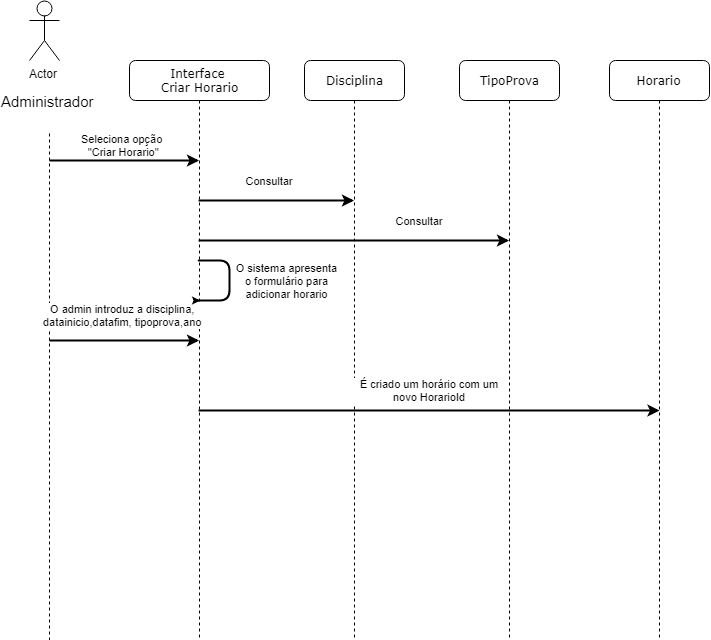
## Diagrama de Sequência dos Casos de Uso dentro da fronteira



###### Fig 1- Diagrama do caso de uso “Criar User”

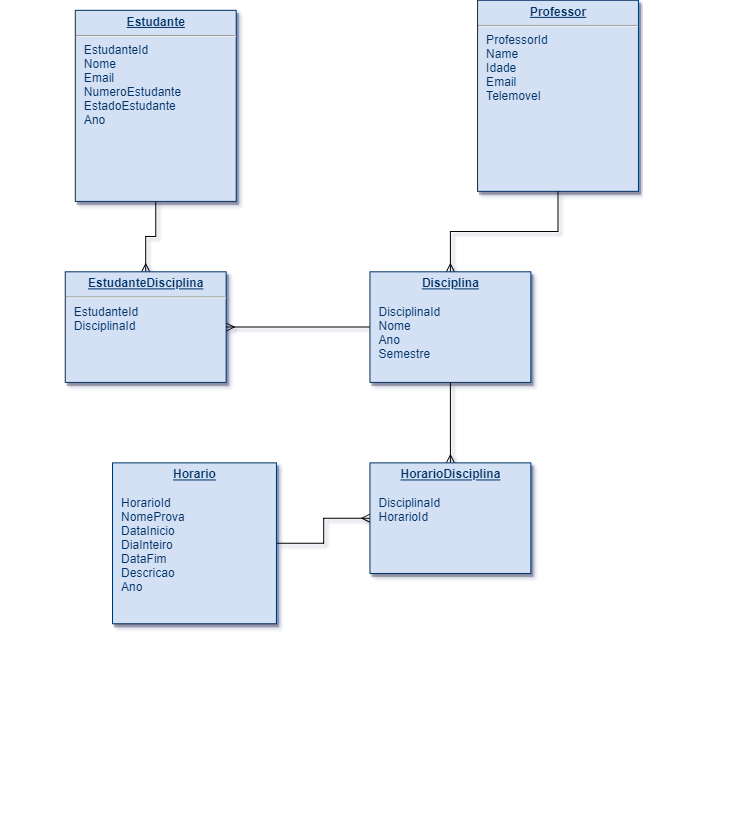


###### Fig 2- Diagrama do caso de uso “Criar Disciplina”



###### Fig 3- Diagrama do caso de uso “Criar Horario”

## Diagrama de Classes

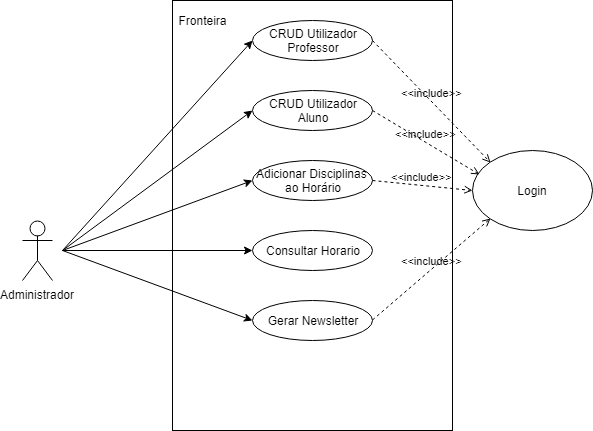


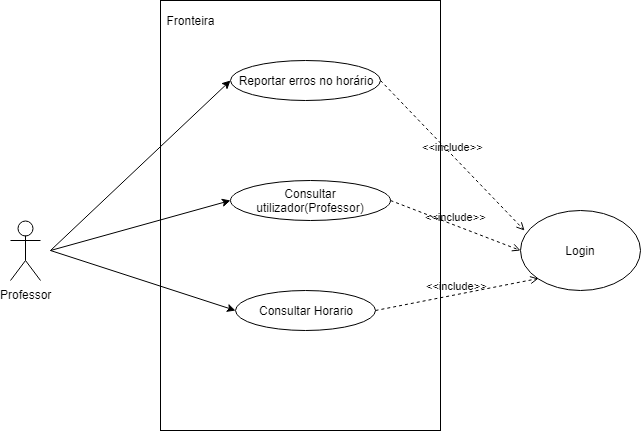
## Diagrama de Estados

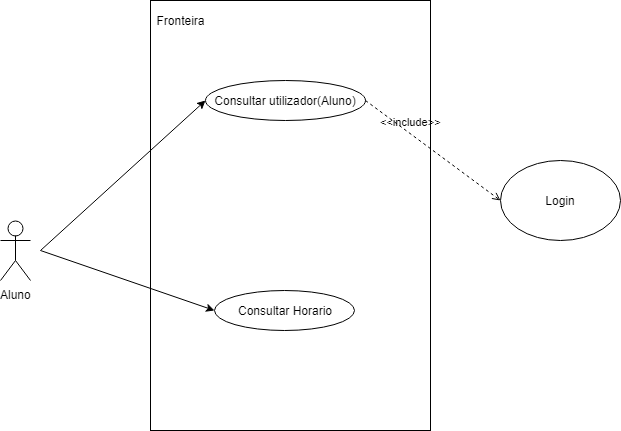
## 

## Diagrama de Pacotes

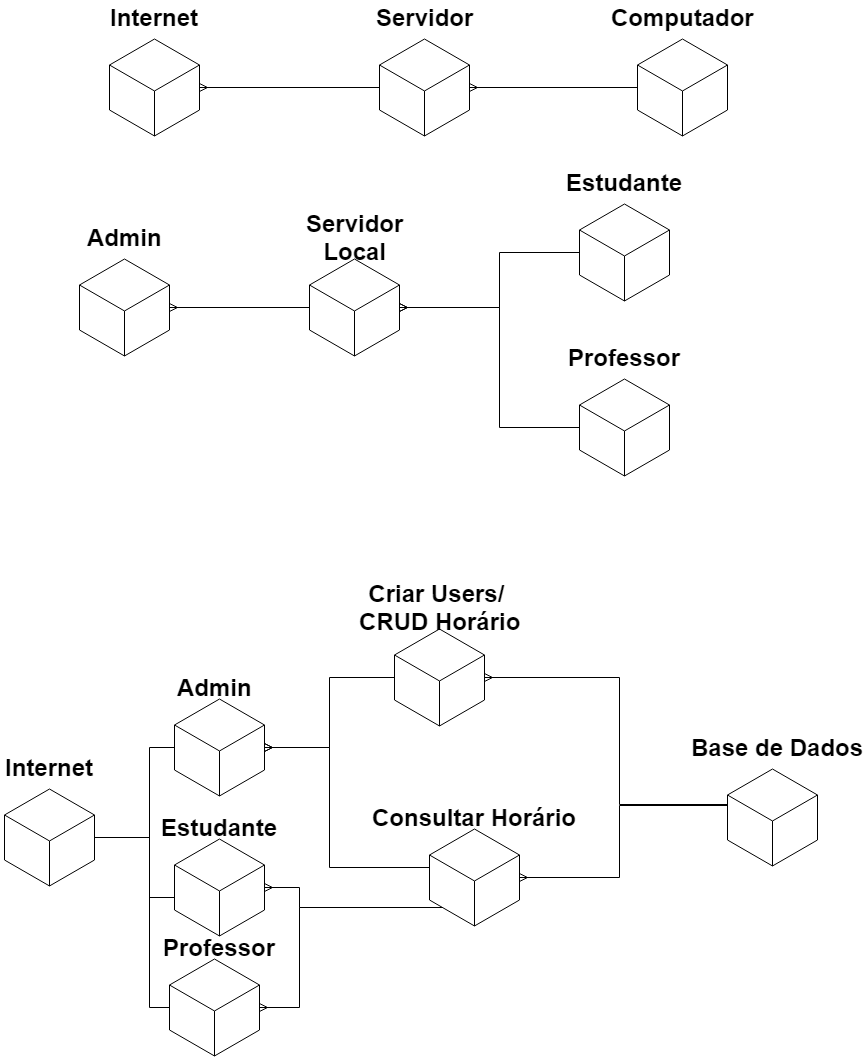
Diagrama de Pacotes-Casos de Uso







## Diagrama de Instalação



## Diagrama de Componentes

## 

**Descrição:**

Professor.exe: Responsável pelas operações relacionadas com o Professor(criar, alterar, apagar, consultar). Depende do componente Professor.dll pois necessita de informação acerca do Professor. Depende do componente ControloAcesso.dll para verificar se o utilizador possui permissões para executar as operações. Também depende da BaseDados.dll, pois necessita de guardar a sua informação numa base de dados.

Estudante.exe: Responsável pelas operações relacionadas com o Estudante(criar, alterar, apagar, consultar). Depende do componente Estudante.dll pois necessita de informação acerca do Estudante. Depende do componente ControloAcesso.dll para verificar se o utilizador possui permissões para executar as operações. Também depende da BaseDados.dll, pois necessita de guardar a sua informação numa base de dados.

Horarios.exe: Responsável pelas operações relacionadas com o Horário(criar, alterar, apagar, consultar). Depende do componente Horario.dll pois necessita de informação acerca do Horário. Depende do componente ControloAcesso.dll para verificar se o utilizador possui permissões para executar as operações. Também depende da BaseDados.dll, pois necessita de guardar a sua informação numa base de dados.

Disciplina.exe: Responsável pelas operações relacionadas com o Disciplina(criar, apagar, consultar). Depende do componente Disciplina.dll pois necessita de informação acerca das Disciplinas. Depende do componente ControloAcesso.dll para verificar se o utilizador possui permissões para executar as operações. Também depende da BaseDados.dll, pois necessita de guardar a sua informação numa base de dados.

## Diagrama de Atividades

## Semântica de Classes

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Classe Professor | | | | | | |
| Nome do Campo | **Tipo de Dados** | **Descrição** | **Valores válidos** | **Formato** | **Restrições** | **Valor por defeito** |
| ProfessorId | INT | Número sequencial que identifica o professor | >0 | Até 2147483648 caracteres | Só pode ser gerado pelo sistema e não pode ser alterados | - |
| Nome | String | Nome do professor | Carateres de A a Z | Entre 2 a 80 Carateres | Obrigatório/ Alterável | - |
| Idade | INT | Idade do professor | >0 | Até 2147483648 caracteres | - | - |
| Email | String | Email do professor | Todo o tipo de carateres com a presença obrigatória de um @ | - | Obrigatório | - |
| Telefone | INT | Telefone de contato do professor | Valores entre 0 e 9 | Até 9 caracteres | Obrigatório / Alterável | - |
| Operações | | | | | | |
| Nome | **Descrição** | | | | | |
| Consultar() | Permite operações como Consultar() da classe Horario. | | | | | |
| Criar() | Operação para criar um “aviso” no horário | | | | | |
| Editar() | Operação de editar dados do professor | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Classe Estudante | | | | | | |
| Nome do Campo | **Tipo de Dados** | **Descrição** | **Valores válidos** | **Formato** | **Restrições** | **Valor por defeito** |
| EstudanteId | INT | Número sequencial que identifica o professor | >0 | Até 2147483648 caracteres | Só pode ser gerado pelo sistema e não pode ser alterados | - |
| Nome | String | Nome do professor | Carateres de A a Z | Entre 2 a 80 Carateres | Obrigatório/ Alterável | - |
| NumeroEstudante | INT | Numero do estudante | >0 | Até 2147483648 caracteres | Obrigatório | - |
| EstadoEstudante | Booleano | Saber estado do estudante | True/false | - | Obrigatório | - |
| Ano | INT | Ano do aluno | Valores entre 1 e 3 | Um digito | Obrigatório / Alterável | - |
| Operações | | | | | | |
| Nome | **Descrição** | | | | | |
| Consultar() | Permite que operações como consultar() da classe horário e da classe Estudante estejam disponíveis | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Classe Horario | | | | | | |
| Nome do Campo | **Tipo de Dados** | **Descrição** | **Valores válidos** | **Formato** | **Restrições** | **Valor por defeito** |
| HorarioId | INT | Número sequencial que identifica o professor | >0 | Até 2147483648 caracteres | Só pode ser gerado pelo sistema e não pode ser alterados | - |
| NomeProva | String | Nome da prova | Valor | Entre 2 a 80 Carateres | Obrigatório/ Alterável | - |
| DataInicio | Date | Data Inicio da prova |  | DD-MM-AAAA HH:MM | Obrigatório | - |
| DiaInteiro | Bool | Se prova tem duração 24h | True/False |  |  |  |
| DataFim | Date | Data fim da prova |  | DD-MM-AAAA HH:MM | Obrigatório | - |
| DataFim | Date |  |  | - | Obrigatório / Alterável | - |
| Ano | Int | Ano escolar | 1-3 |  | Obrigatório / Alterável |  |
| Operações | | | | | | |
| Nome | **Descrição** | | | | | |
| Consultar() | Permite que operações como Criar(), Editar() da classe horário consultem as avaliações disponíveis | | | | | |
| Criar() | Operação que permite a criação de uma prova no horário | | | | | |
| Editar() | Operação de edição de uma prova no horário | | | | | |
| Desativar() | Permite remover uma prova do horário | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Classe Disciplina | | | | | | |
| Nome do Campo | **Tipo de Dados** | **Descrição** | **Valores válidos** | **Formato** | **Restrições** | **Valor por defeito** |
| DisciplinaId | INT | Número sequencial que identifica o professor | >0 | Até 2147483648 caracteres | Só pode ser gerado pelo sistema e não pode ser alterados | - |
| Nome | String | Nome da Disciplina | Carateres de A a Z | Entre 2 a 80 Carateres | Obrigatório/ Alterável | - |
| Ano | INT | Ano da Disciplina | >0 | Até 2147483648 caracteres | - | - |
| Operações | | | | | | |
| Nome | **Descrição** | | | | | |
| Consultar() | Permite que operações como Criar(), Editar() da classe Horário consultem as provas disponíveis | | | | | |
| Criar() | Operação que permite a criação de uma Disciplina | | | | | |
| Editar() | Operação de edição de Uma disciplina | | | | | |
| Desativar() | Permite que a Disciplina seja removida. | | | | | |

## Conclusão

#### Finalizando este relatório conseguimos ver a diferença fulcral no que toca a desenvolver software de uma forma mais premeditada e estudada, em comparação com projetos desenvolvidos em anos letivos passados, a qualidade do software assim como a capacidade de implementação de ideias é notoriamente diferente, e em jeito de conclusão aprendemos que todo este estudo seja de padrões de desenvolvimento, de casos de uso ou diagramas está para o desenvolvimento de um bom software, assim como um bom algoritmo está na base de um excelente código.

#### Desde as comparações com "sites tipo" onde vimos o que nos era importante, assim como o que não era até aos brainstorming’s baseados nestes e em outros exercícios propostos, com a introdução dos conceitos como padrões de desenvolvimento notamos a necessidade dos métodos de desenvolvimento, como o método Ágil, que no fim nos permitiu um desenvolvimento de software muito mais conciso e consistente, e apesar de não termos estado em contacto direto com os nossos Stakeholders, acreditamos que todo este conhecimento apreendido na realização deste trabalho e relatório nos permitiu construir algo que agradaria todas as partes interessadas no projeto.

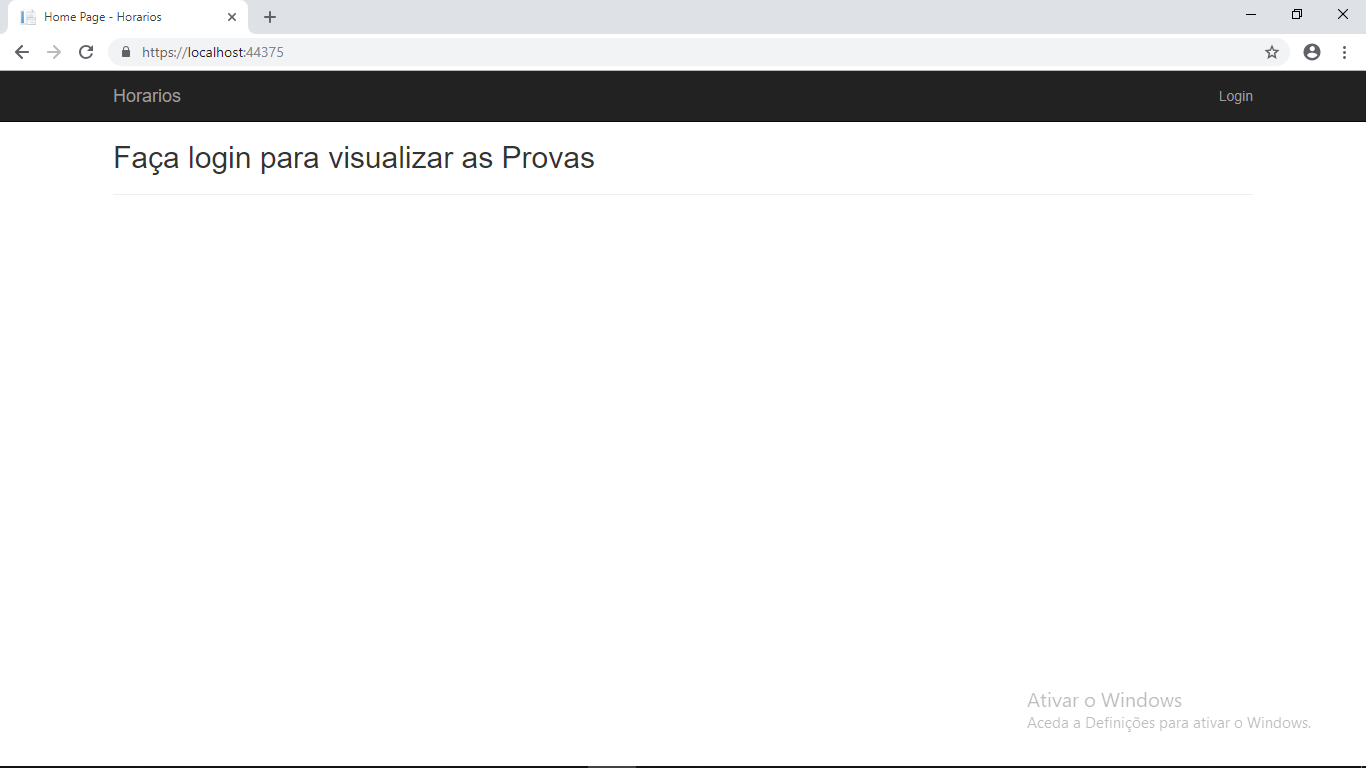
#### No âmbito da unidade curricular de Engenharia de Software II, concluímos que apesar da disciplina ser algo trabalhosa conseguimos notar diferenças facilmente visíveis na melhorada forma com que desenvolvemos o estudo do software e o próprio, desde a melhor organização de tempo com diários até a uma delegação de tarefas mais eficientes.

|  |  |  |
| --- | --- | --- |
| Data | Duração | Trabalho realizado |
| 5/10/2018 | 1h00min | Organização da ideia do projeto |
| 11/10/2018 | 4h00min | Criação inicial do relatório/projeto |
| 29/11/2018 | 6h30min | Implementação dos layouts(templates) no projeto |
| 29/11/2018 | 3h00min | Criação da descrição de Casos de uso |
| 24/12/2018 | 4h00min | Continuação da descrição de Casos de uso |
| 29/12/2018 | 3h25min | Criação das classes(models) |
| 01/01/2019 | 3h30min | Criação do controlodor backoffice |
| 03/01/2019 | 1h20min | Atualização do relatório do projeto(Diagrama de componentes/instalação) |
| 04/01/2019 | 1h20min | Atualização do controlador backoffice |
| 07/01/2019 | 3h00min | Scaffolding dos modelos existentes |
| 09/01/2019 | 6h00min | Modificação das views |
| 16/01/2019 | 0h20min | Atualização dos controladores |
| 19/01/2019 | 4h00min | Atualizações dos controllers/views/models |
| 20/01/2019 | 5h25min | Atualizações dos controllers/views/models |
| 20/01/2019 | 2h00min | Atualização do relatório |
| 22/01/2019 | 2h30min | Criação das roles |
| 23/01/2019 | 4h30min | Criação das roles |
| 27/01/2019 | 4h00min | Paginação /Pesquisa |
| 28/01/2019 | 4h45min | Paginação /Pesquisa |
| 28/01/2019 | 9h00min | Atualização do projeto em geral |

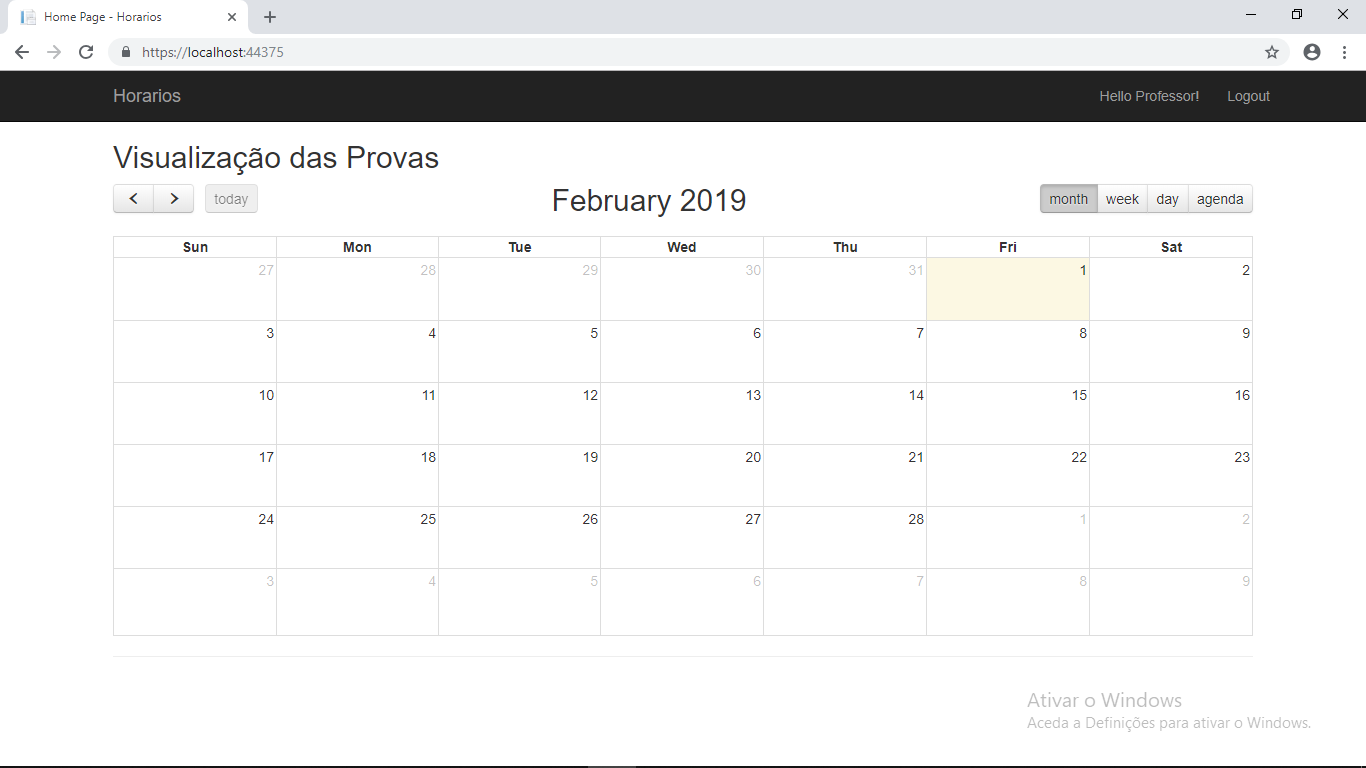
## Anexos

Trabalho realizado

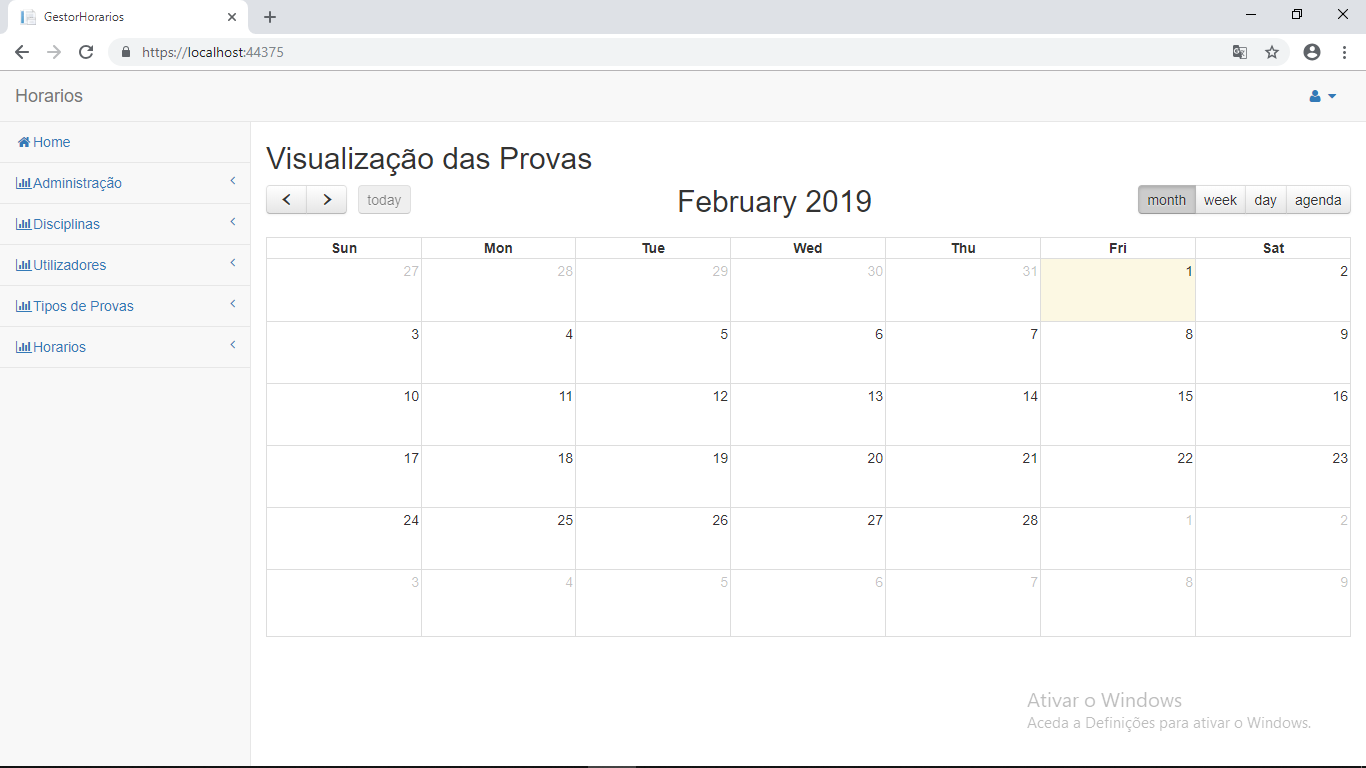
Prototipagem



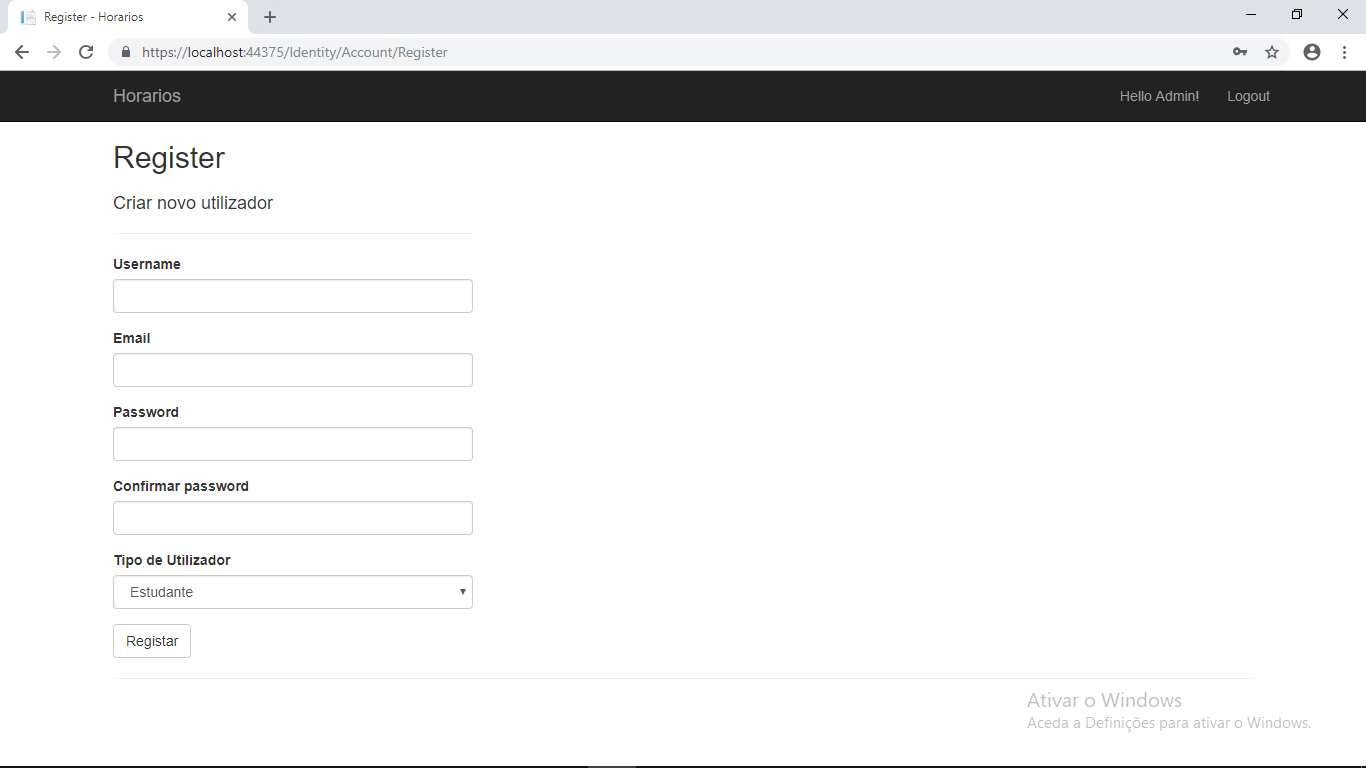
###### Fig. da página inicial da aplicação(sem login válido)



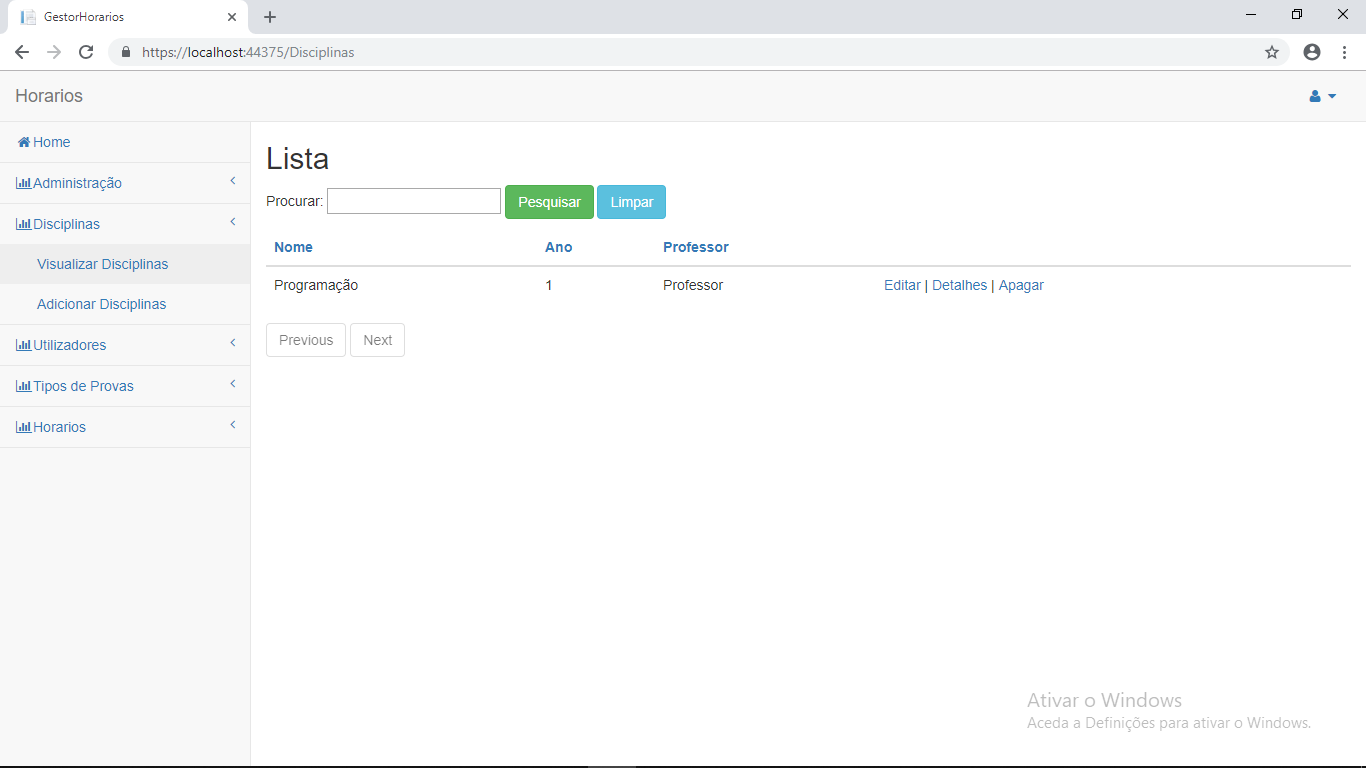
###### Fig. da página inicial da aplicação(com login válido)



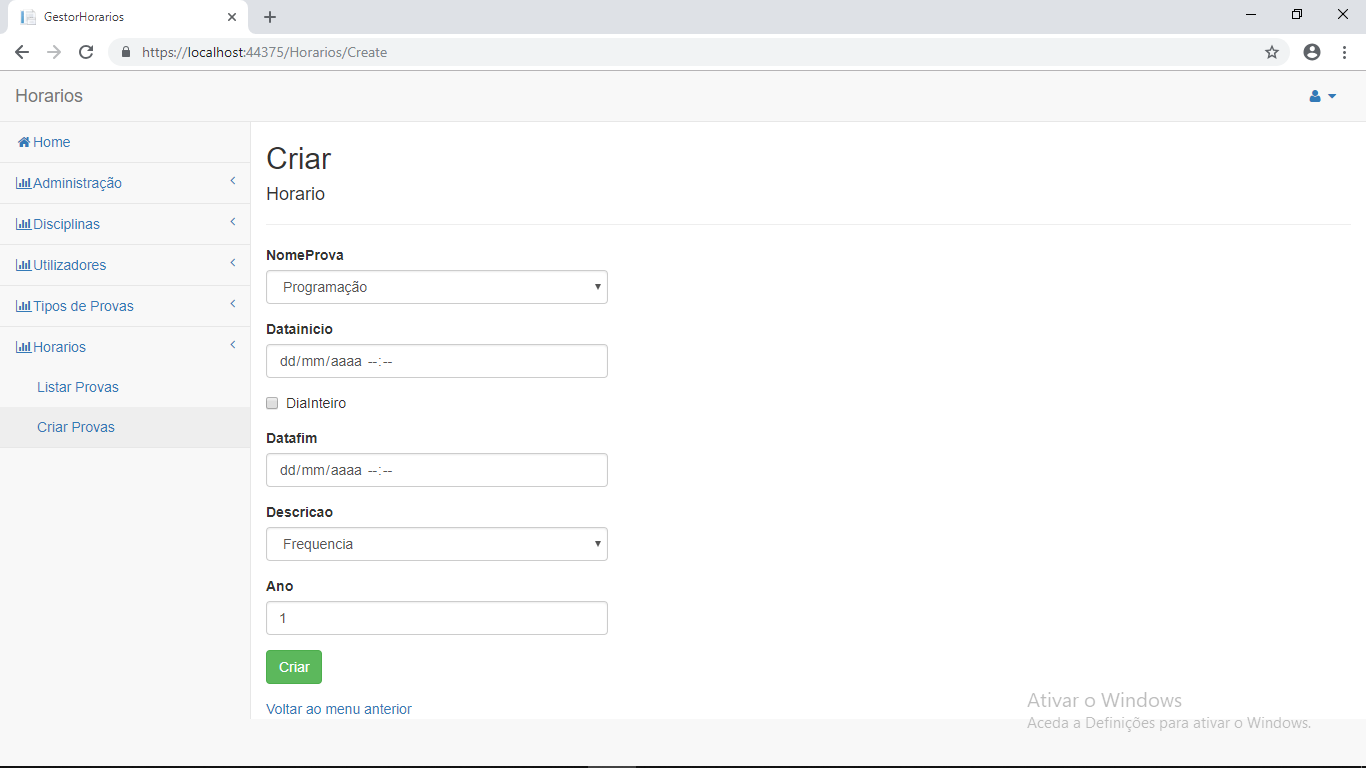
###### Fig. da página de administração do Administrador do sistema depois de estar logado.



###### Fig. da página para registar utilizadores no sistema



###### Fig. da página para listar disciplinas



###### Fig. da página para criar um horário

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.");