



UNIVERSIDADE DE TRÁS-OS-MONTES E ALTO DOURO

# Relatório 3: Identificação e Diagramação de Classes, e Diagramas de Estados do SIUTAD

---

Licenciatura em Engenharia Informática

Engenharia de Software

Prof. António Gouveia

Prof. José Martins

## **Autores**

Marcelo Pinto - 60102

Ricardo Cardoso - 28382

Vila Real, 2016

## **Resumo**

Um Diagrama de Classes consiste na representação da estrutura e relações entre as classes que compõem o sistema. Uma classe é uma estrutura abstrata que representa um tipo de objetos com características similares.

O Diagrama de Estados é uma representação do estado ou situação em que um objeto se pode encontrar no decorrer da execução de processos no sistema.

Neste relatório apresentamos a diagramação de Classes e de Estados do SIUTAD.

## Índice

1. Introdução .....	5
2. Enquadramento Teórico .....	5
3. Objetivos da Etapa do Trabalho Prático .....	6
4. Resultados Atingidos .....	6
4.1. Diagramação de Classes .....	6
4.1.1. Diagrama de Classes.....	7
4.1.2. Especificação de Classes.....	8
4.1.2.1. Utilizador .....	8
4.1.2.2. Turma .....	9
4.1.2.3. InscriçãoTurma .....	9
4.1.2.4. Departamento.....	10
4.1.2.5. RegistoAssiduidade.....	10
4.1.2.6. TipoRegisto.....	11
4.1.2.7. TipoDia .....	11
4.1.2.8. Gabinete.....	12
4.1.2.9. CategoriaProfissional .....	12
4.1.2.10. TipoDeUtilizador .....	13
4.1.2.11. MembrosGrupo .....	13
4.1.2.12. Grupo .....	14
4.1.2.13. InscriçãoAvaliação.....	14
4.1.2.14. Submissão .....	15
4.1.2.15. UCDocente.....	15
4.1.2.16. UnidadeCurricular .....	16
4.1.2.17. ElementoAvaliação .....	16
4.1.2.18. TipoAvaliação .....	17

4.1.2.19. Curso.....	17
4.1.2.20. TipoCurso.....	18
4.1.2.21. Pauta .....	18
4.1.2.22. InscriçãoUC.....	19
4.1.2.23. Notificação .....	19
4.1.2.24. EstadoPauta .....	20
4.1.2.25. ÉpocaAvaliação.....	20
4.1.2.26. TipologiaTurma .....	21
4.1.2.27. Aula .....	21
4.1.2.28. Assiduidade .....	22
4.1.2.29. Direção .....	22
4.1.2.30. InscriçãoUCPauta.....	23
4.2. Diagramas de Estados .....	23
4.2.1. Diagrama de Estados - Classe Utilizador .....	24
4.2.2. Diagrama de Estados - Classe Pauta.....	25
4.2.3. Diagrama de Estados - Classe ElementoAvaliação .....	26
4.2.4. Diagrama de Estados - Classe Submissão .....	27
4.2.5. Diagrama de Estados - Classe Grupo .....	28
4.2.6. Diagrama de Estados - Classe Aula .....	29
4.2.7. Diagrama de Estados - Classe Turma.....	30
4.2.8. Diagrama de Estados - Classe InscriçãoTurma.....	31
4.2.9. Diagrama de Estados - Classe Curso .....	32
5. Análise e Discussão dos Resultados .....	33
6. Bibliografia .....	33

## 1. Introdução

No âmbito da Unidade Curricular de Engenharia de Software foi-nos proposto a realização de um trabalho prático que consiste na modelação de um sistema de *software* através da aplicação da UML. O trabalho foi dividido em etapas, nas etapas 1 e 2 foram realizadas o levantamento de requisitos funcionais e não funcionais e a diagramação e especificação de Casos-de-Uso.

Nesta terceira etapa vamos fazer a diagramação de Classes e de Estados, com base nos relatórios 1 e 2 e na descrição informal do SIUTAD que nos foi fornecida.

## 2. Enquadramento Teórico

Um Diagrama de Classes consiste na representação da estrutura e relações entre as classes que compõem o sistema.

Uma classe é uma estrutura abstrata que representa um tipo de objetos com características similares. Uma classe define o comportamento de seus objetos através de métodos e os estados possíveis destes objetos através de atributos. Os atributos são propriedades de uma classe, e os métodos determinam o comportamento dos objetos de uma classe, podendo ou não, alterar os atributos da própria classe, que por definição devem ser acessíveis apenas aos métodos da própria classe, obedecendo a uma das regras principais na programação a objetos - encapsulamento.

O Diagrama de Estados é uma representação do estado ou situação em que um objeto se pode encontrar no decorrer da execução de processos no sistema. Os Diagramas de Estados têm como finalidade descrever o comportamento de um objeto, identificando os diferentes estados em que um objeto pode estar e o que provoca as transições de estados, todos os estados possíveis que os objetos de uma classe podem ter, assim como os eventos que causam a transição entre estados. São complementares à descrição de uma classe.

### 3. Objetivos da Etapa do Trabalho Prático

Os nossos objetivos para a presente etapa são:

- Identificar todas as classes representativas do SIUTAD, na nossa perspetiva;
- Identificar os relacionamentos das classes corretamente assim como a multiplicidade desses relacionamentos;
- Identificar os atributos e métodos de cada classe corretamente;
- Identificar as classes com comportamento dinâmico relevante para posterior diagramação de estados;
- Fazer uma correta Diagramação de Classes e Estados.

### 4. Resultados Atingidos

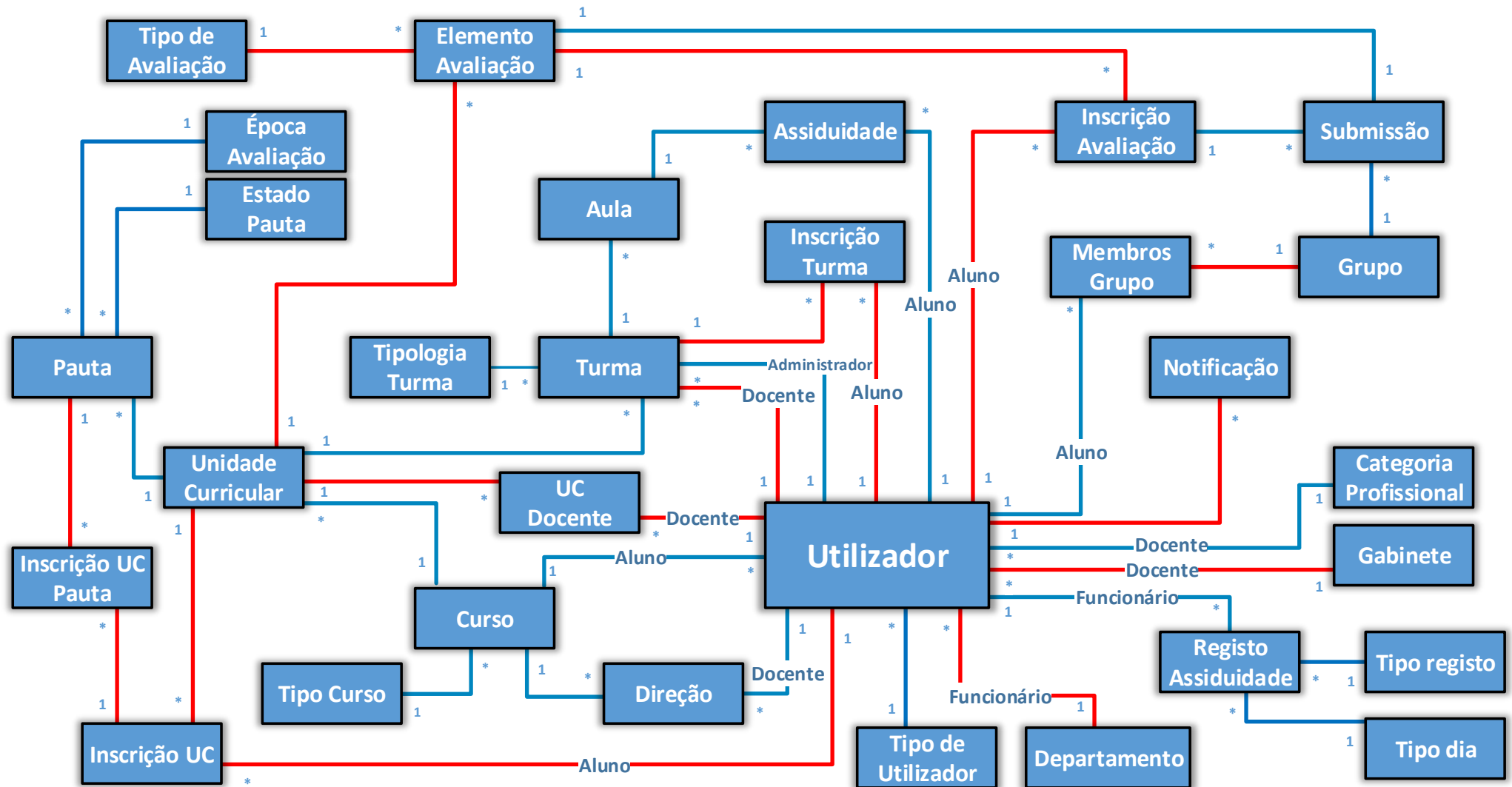
Nesta etapa do trabalho apresentamos os resultados atingidos, isto é:

- Diagramação de todas as classes necessárias no sistema SIUTAD;
- Diagramas de Estados.

#### 4.1. Diagramação de Classes

Na página seguinte apresentamos o Diagrama de Classes do SIUTAD. A mudança de cores apenas serve para uma melhor distinção das ligações e classes, oferecendo uma melhor visualização do diagrama.

### 4.1.1. Diagrama de Classes



### 4.1.2. Especificação de Classes

Todos os construtores, *gets* e *sets* de qualquer classe estão generalizados nos dois primeiros métodos para simplificação. Em algumas classes, nós achamos conveniente substituir esses métodos de acesso por métodos com nomes com mais significado.

#### 4.1.2.1. Utilizador

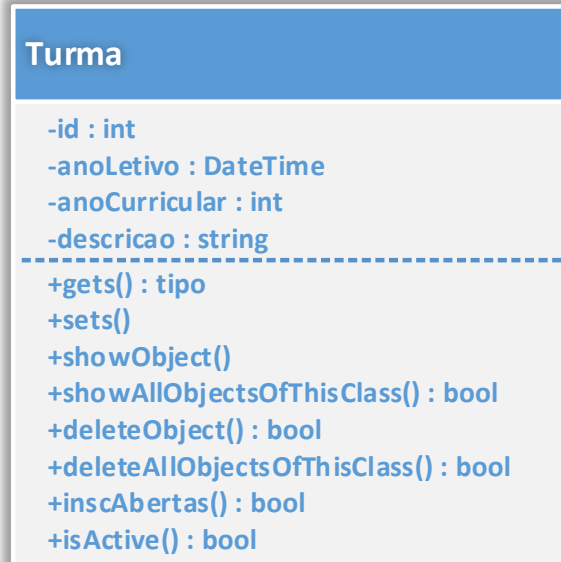
A classe utilizador é uma generalização do utilizador comum ou visitante, do docente, do aluno, do administrador e do funcionário.





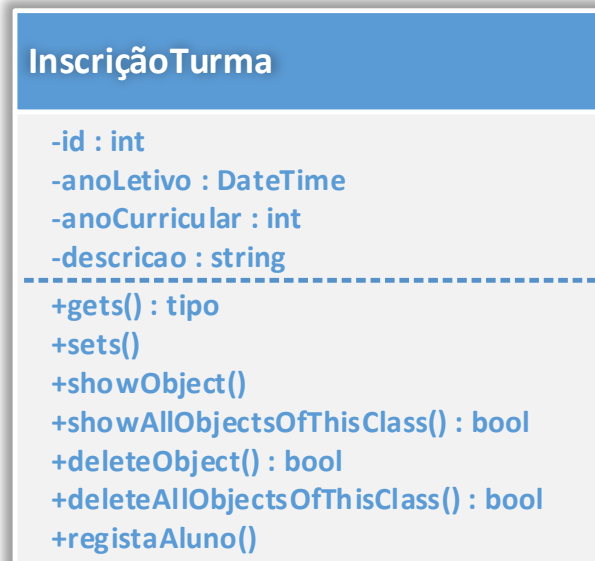
#### 4.1.2.2. Turma

A classe Turma é relativa a uma turma de alunos, com um docente, e pertencente a uma unidade curricular.



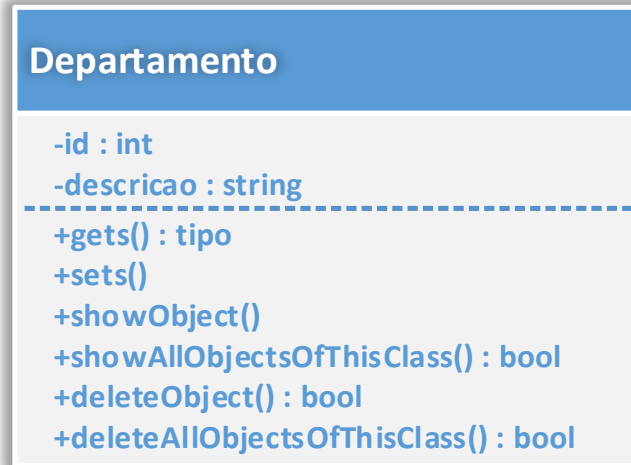
#### 4.1.2.3. InscriçãoTurma

A classe InscriçãoTurma é uma classe intermédia entre o utilizador aluno e a turma. É relativa a uma inscrição de um aluno numa turma. Cada aluno poderá ter várias inscrições e cada turma tem várias inscrições.



#### 4.1.2.4. Departamento

A classe Departamento é relativa ao departamento em que um funcionário trabalha. Um departamento pode ter vários funcionários.



#### 4.1.2.5. RegistoAssiduidade

A classe RegistoAssiduidade é relativa ao registo de assiduidade por parte dos funcionários.



#### 4.1.2.6. TipoRegisto

A classe TipoRegisto é relativa ao tipo de registo que o funcionário fez.

TipoRegisto
<div><div>-id : int</div><div>-tipo : string</div><div>-descricao : string</div></div> <div><div>+gets() : tipo</div><div>+sets()</div><div>+showObject()</div><div>+showAllObjectsOfThisClass() : bool</div><div>+deleteObject() : bool</div><div>+deleteAllObjectsOfThisClass() : bool</div></div>

#### 4.1.2.7. TipoDia

A classe TipoDia é relativa ao tipo de dia, que pode ser por exemplo: dia normal, feriado, ponte...

TipoDia
<div><div>-id : int</div><div>-data : DateTime</div><div>-tipo : string</div><div>-descricao : string</div></div> <div><div>+gets() : tipo</div><div>+sets()</div><div>+showObject()</div><div>+showAllObjectsOfThisClass() : bool</div><div>+deleteObject() : bool</div><div>+deleteAllObjectsOfThisClass() : bool</div></div>

#### 4.1.2.8. Gabinete

A classe Gabinete refere-se ao gabinete (simplicadamente sala) onde um docente pode trabalhar e atender alunos quando não está a lecionar.

Gabinete
<div><div>-id : int</div><div>-sala : string</div><div>-dimensao : float</div><div>-descricao : string</div></div> <div><div>+gets() : tipo</div><div>+sets()</div><div>+showObject()</div><div>+showAllObjectsOfThisClass() : bool</div><div>+deleteObject() : bool</div><div>+deleteAllObjectsOfThisClass() : bool</div></div>

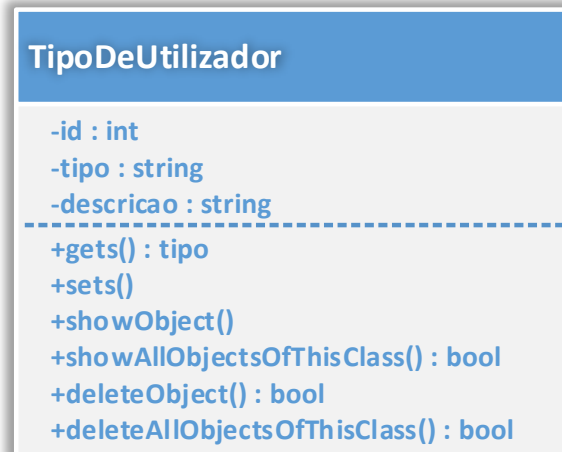
#### 4.1.2.9. CategoriaProfissional

A classe CategoriaProfissional refere-se à categoria profissional – académica – que o docente possui.

CategoriaProfissional
<div><div>-id : int</div><div>-categoria : string</div><div>-descricao : string</div></div> <div><div>+gets() : tipo</div><div>+sets()</div><div>+showObject()</div><div>+showAllObjectsOfThisClass() : bool</div><div>+deleteObject() : bool</div><div>+deleteAllObjectsOfThisClass() : bool</div></div>

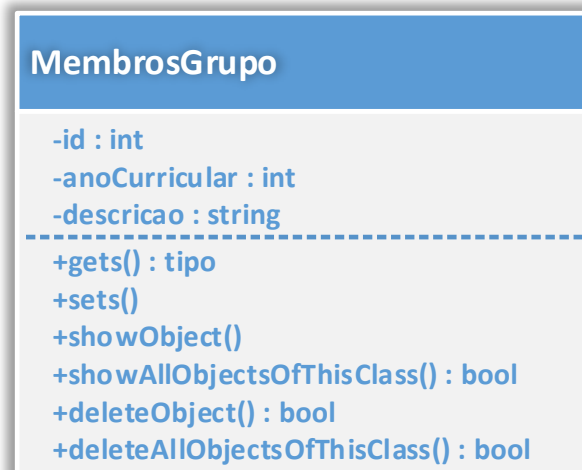
#### 4.1.2.10. TipoDeUtilizador

A classe TipoDeUtilizador refere-se ao tipo de utilizador da classe genérica Utilizador. Pode ser Docente, Aluno, Administrador, Funcionário, entre possíveis outros.



#### 4.1.2.11. MembrosGrupo

A classe MembrosGrupo é uma classe intermédia entre o aluno e o grupo.



#### 4.1.2.12. Grupo

A classe Grupo é uma classe com o objetivo da formação de um grupo de alunos para uma avaliação.

Grupo
<div><div>-id : int</div><div>-nome : int</div><div>-anoLetivo : DateTime</div><div>-descricao : String</div></div> <div><div>+gets() : tipo</div><div>+sets()</div><div>+showObject()</div><div>+showAllObjectsOfThisClass() : bool</div><div>+deleteObject() : bool</div><div>+deleteAllObjectsOfThisClass() : bool</div><div>+isActive() : bool</div><div>+convitesEnviados() : bool</div><div>+atualizaMembros()</div><div>+respostaConvite() : bool</div></div>

#### 4.1.2.13. InscriçãoAvaliação

A classe InscriçãoAvaliação é uma classe intermédia entre o aluno e o elemento de avaliação.

InscriçãoAvaliação
<div><div>-id : int</div><div>-anoLetivo : DateTime</div><div>-dataInscricao: DateTime</div><div>-nota : float</div><div>-descricao : string</div></div> <div><div>+gets() : tipo</div><div>+inscreverAluno()</div><div>+editarAluno()</div><div>+showObject()</div><div>+showAllObjectsOfThisClass() : bool</div><div>+deleteObject() : bool</div><div>+deleteAllObjectsOfThisClass() : bool</div></div>

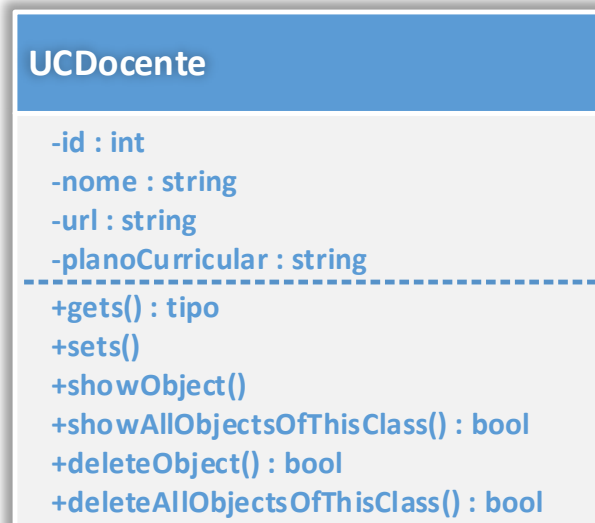
#### 4.1.2.14. Submissão

A classe submissão é uma classe com relação de um para um ao elemento de avaliação e o objetivo é que a submissão para um elemento de avaliação seja feita a partir desta classe.



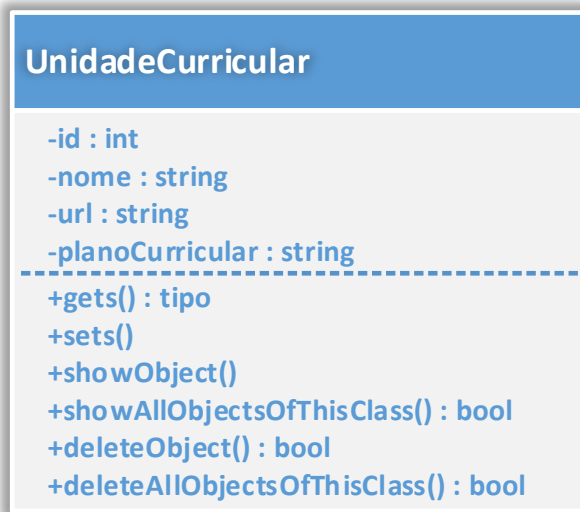
#### 4.1.2.15. UCDocente

A classe UCDocente é uma classe intermédia entre a classe UnidadeCurricular e a classe Docente.



#### 4.1.2.16. UnidadeCurricular

A classe UnidadeCurricular representa uma disciplina, que pertence a um curso.



#### 4.1.2.17. ElementoAvaliação

A classe ElementoAvaliação representa uma avaliação para uma determinada unidade curricular.





#### 4.1.2.18. TipoAvaliação

A classe TipoAvaliação representa o tipo de avaliação de um elemento de avaliação (trabalho de grupo, frequência, exame...).



#### 4.1.2.19. Curso

A classe Curso representa um curso. O curso é uma das unidades fundamentais de todo o sistema.



#### 4.1.2.20. TipoCurso

A classe TipoCurso representa o tipo de curso. Cada objeto desta classe deve estar relacionado com um ou mais cursos.



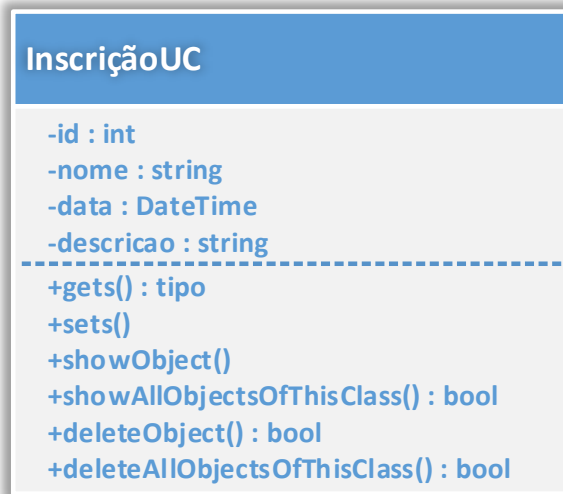
#### 4.1.2.21. Pauta

A classe Pauta é o conjunto de vários resultados de avaliações.



#### 4.1.2.22. InscriçãoUC

A classe InscriçãoUC é a classe intermédia entre o aluno e a unidade curricular.



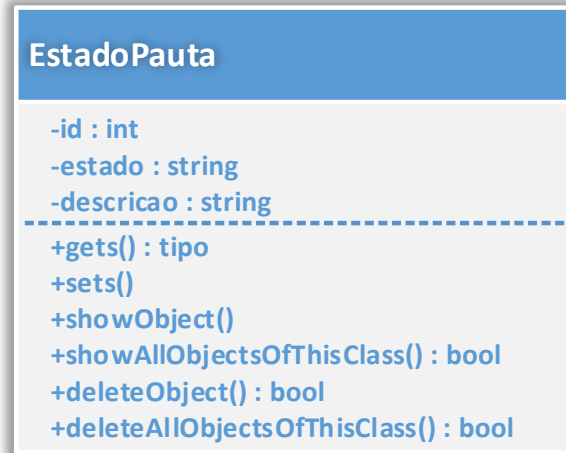
#### 4.1.2.23. Notificação

A classe Notificação é a classe que gere as notificações.



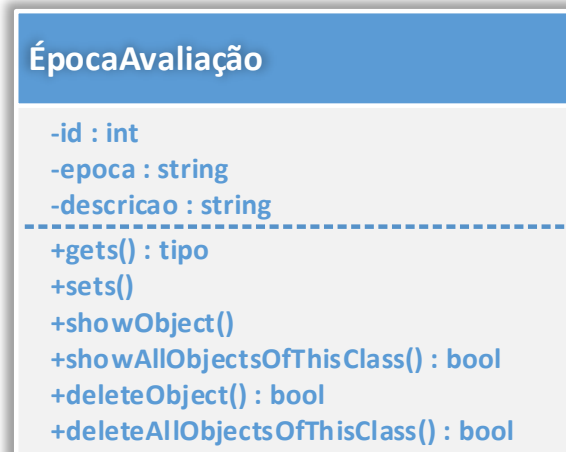
#### 4.1.2.24. EstadoPauta

A classe EstadoPauta é a classe que gere os estados da pauta (lacrada, não oficial, oficial).



#### 4.1.2.25. ÉpocaAvaliação

A classe ÉpocaAvaliação é a classe representativa da época de avaliação (época normal, época de recurso, época antecipada...).



#### 4.1.2.26. TipologiaTurma

A classe TipologiaTurma é a classe que representa o tipo de um objeto da classe Turma.

TipologiaTurma
<div><div>-id : int</div><div>-tipo : string</div><div>-descricao : string</div></div> <div><div>+gets() : tipo</div><div>+sets()</div><div>+showObject()</div><div>+showAllObjectsOfThisClass() : bool</div><div>+deleteObject() : bool</div><div>+deleteAllObjectsOfThisClass() : bool</div></div>

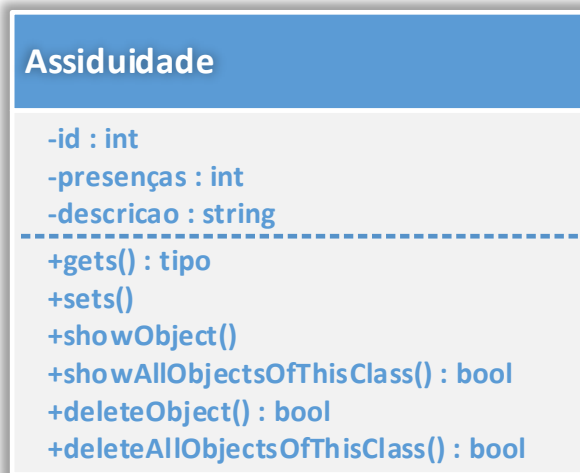
#### 4.1.2.27. Aula

A classe Aula é a classe que gere as aulas de uma Turma.

Aula
<div><div>-id : int</div><div>-sumario : string</div><div>-descricao : string</div></div> <div><div>+gets() : tipo</div><div>+sets()</div><div>+showObject()</div><div>+showAllObjectsOfThisClass() : bool</div><div>+deleteObject() : bool</div><div>+deleteAllObjectsOfThisClass() : bool</div><div>+isActive() : bool</div><div>+necessitaEdicao() : bool</div><div>+confirmaSumario() : bool</div><div>+editarDados()</div></div>

#### 4.1.2.28. Assiduidade

A classe assiduidade é a classe intermédia entre o aluno e a aula.



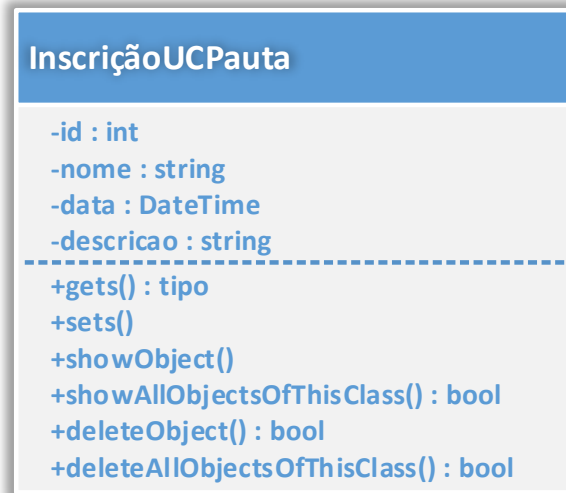
#### 4.1.2.29. Direção

A classe direção é a classe intermédia entre o Docente e o curso.



#### 4.1.2.30. InscriçãoUCPauta

A classe InscriçãoUCPauta é a classe intermédia entre a classe intermédia do aluno/UC e a classe Pauta.

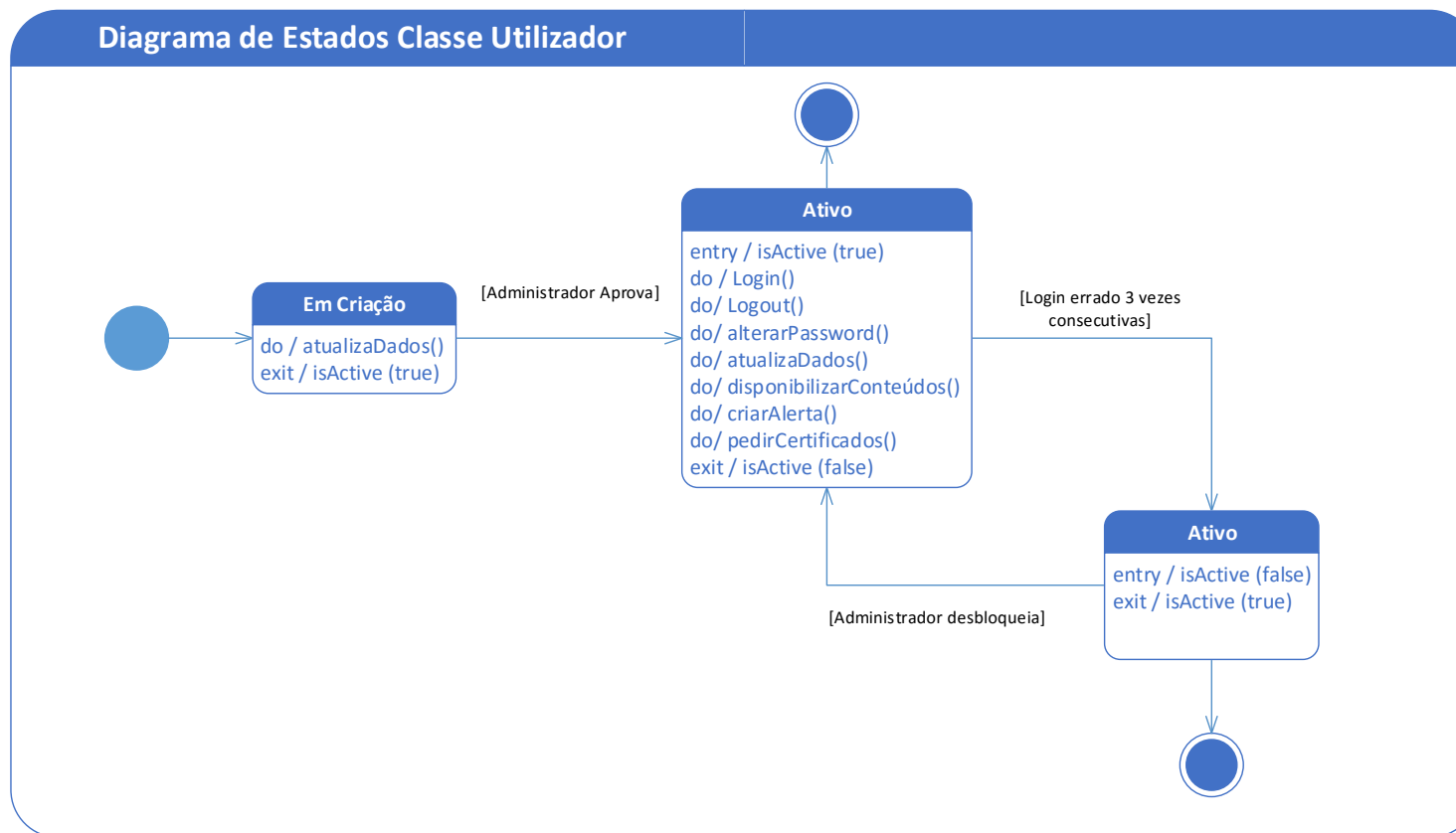


#### 4.2. Diagramas de Estados

Nas páginas seguintes apresentamos os Diagramas de Estados para as classes que consideramos relevantes com o intuito de melhor compreender o comportamento de um objeto dessa classe, isto é, de maneira a visualizar os estados por que um objeto dessa classe pode passar e os eventos que promovem e a troca de estados.

#### 4.2.1. Diagrama de Estados - Classe Utilizador

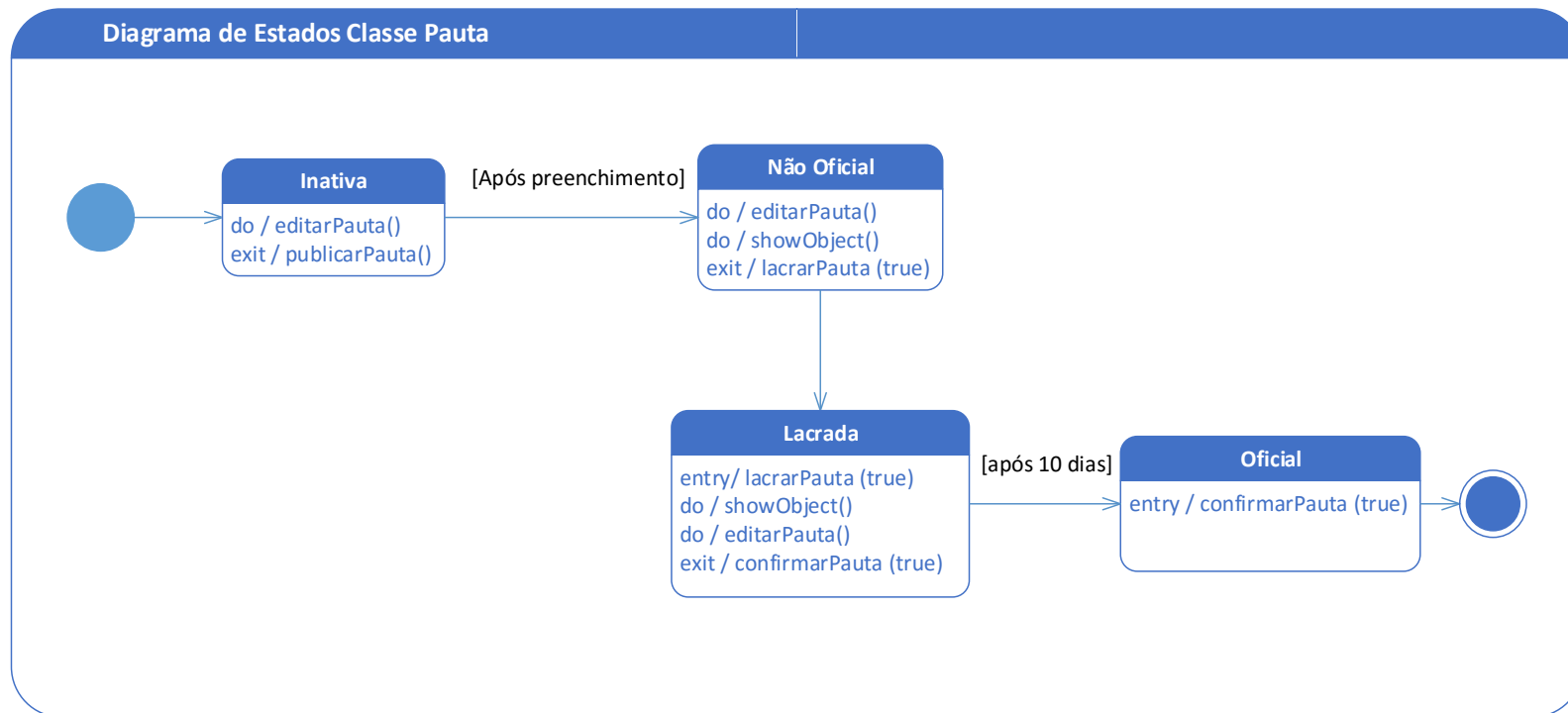
Neste diagrama observamos os estados por que cada objeto da classe Utilizador pode passar. Começando no estado em criação até que um administrador o aprove, passando de seguida para o estado Ativo onde pode realizar todas as operações designadas a um Utilizador (do seu tipo), caso erre as suas credenciais de login 3 vezes consecutivas o Utilizador passa de um estado Ativo para um estado Bloqueado até que um Administrador o desbloqueie.





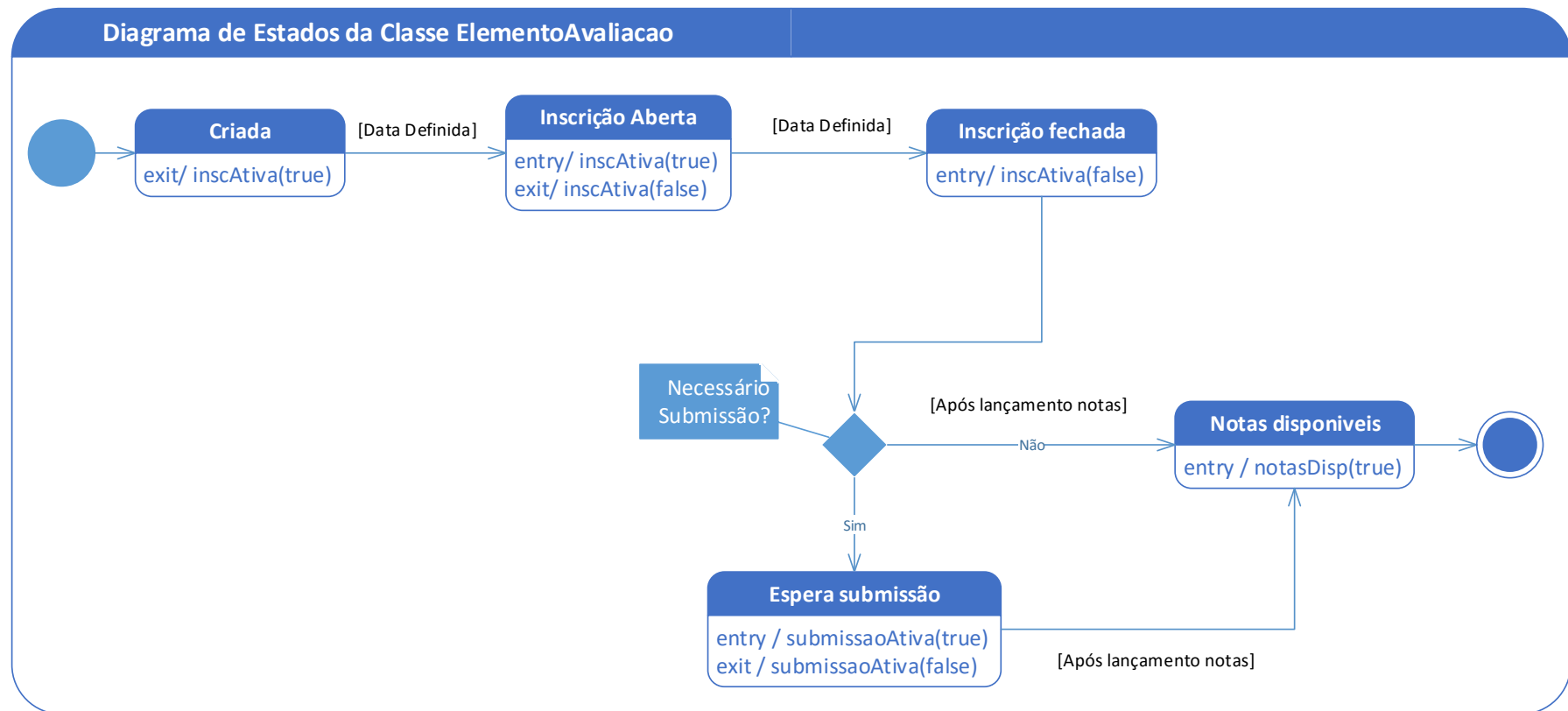
#### 4.2.2. Diagrama de Estados - Classe Pauta

Neste diagrama observamos os estados por que cada objeto da classe Pauta pode passar. Começando no estado Inativo até que o Docente responsável a preencha, passando para o estado Não-oficial, quando o Docente entender que a pauta já não irá sofrer mais alterações pode Lacrar a pauta passando esta para o estado Lacrada, após confirmação e decorridos 10 dias a Pauta passará para o estado Oficial.



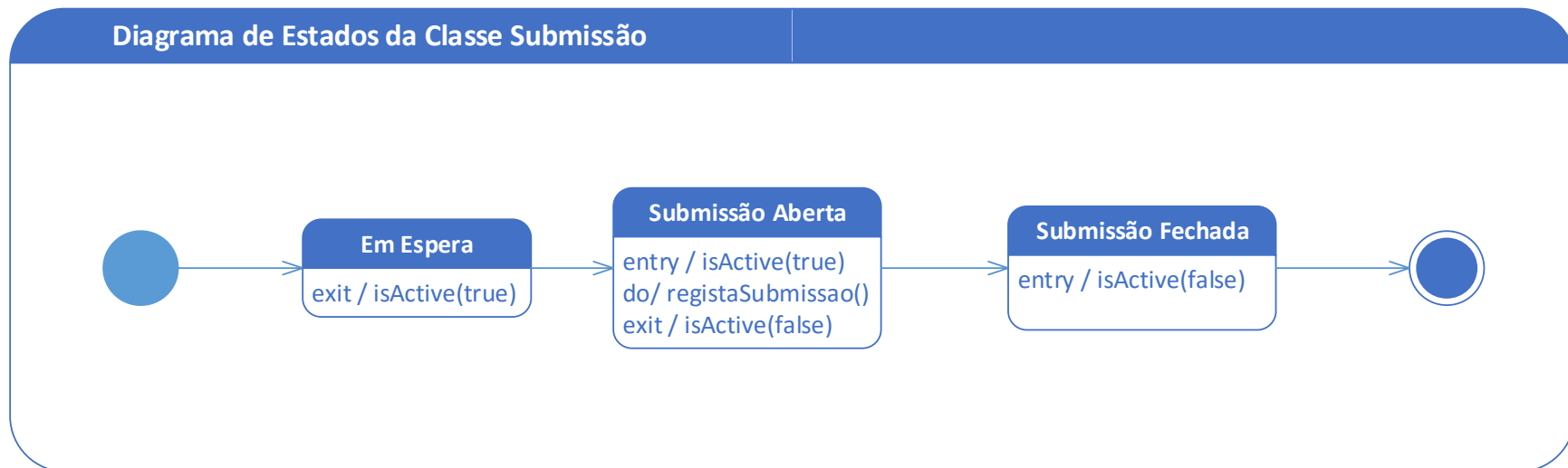
#### 4.2.3. Diagrama de Estados - Classe ElementoAvaliação

Quando um Elemento de Avaliação é criado este inicia no estado criado transitando para o estado Inscrição Aberta na data definida aquando a criação do mesmo. Decorrido o tempo de inscrição o objeto passa para o estado Inscrição fechada. Caso seja necessário efetuar uma submissão, passa para o estado Espera Submissão, senão passa para o estado Notas disponíveis, após lançamento das notas.



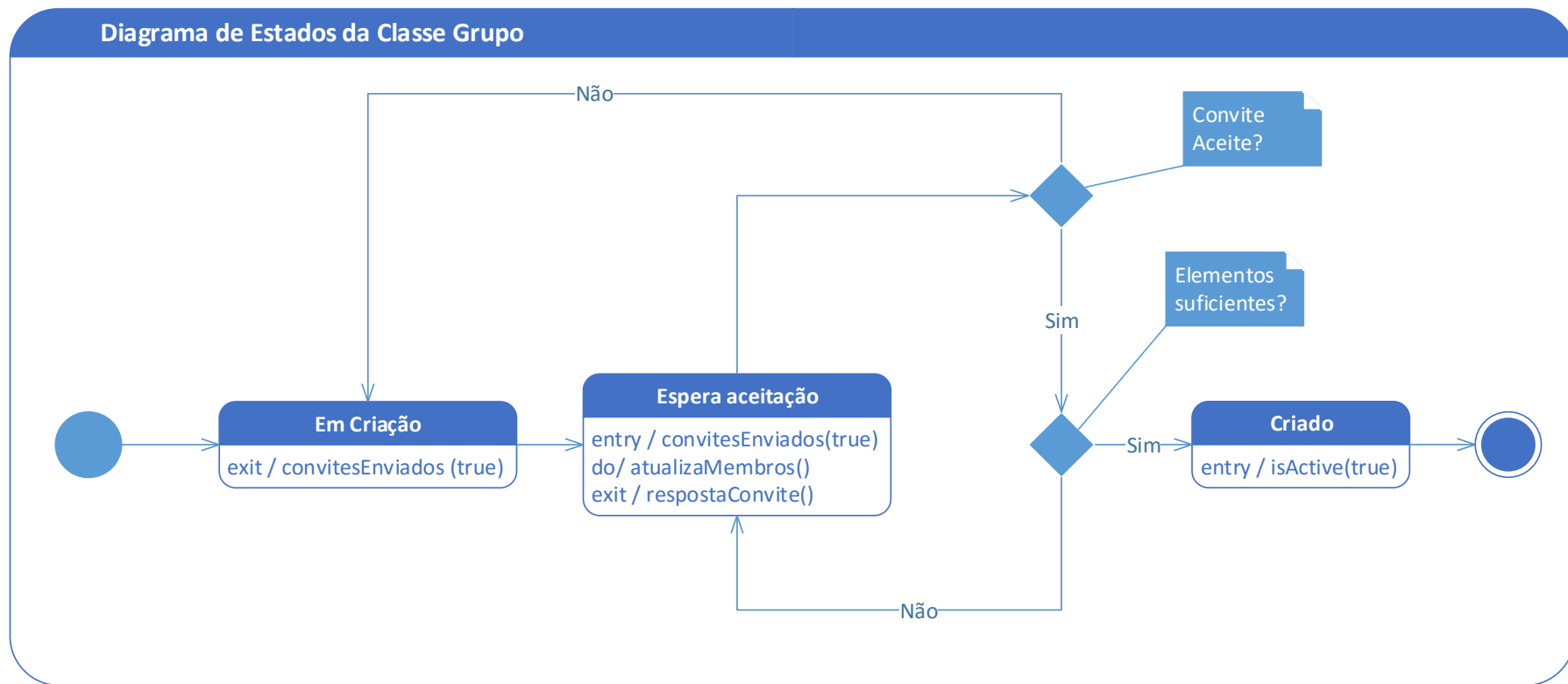
#### 4.2.4. Diagrama de Estados - Classe Submissão

Este diagrama representa os estados por que um objeto da classe Submissão passa sempre que é necessário realizar a submissão de um trabalho para um Elemento de Avaliação.



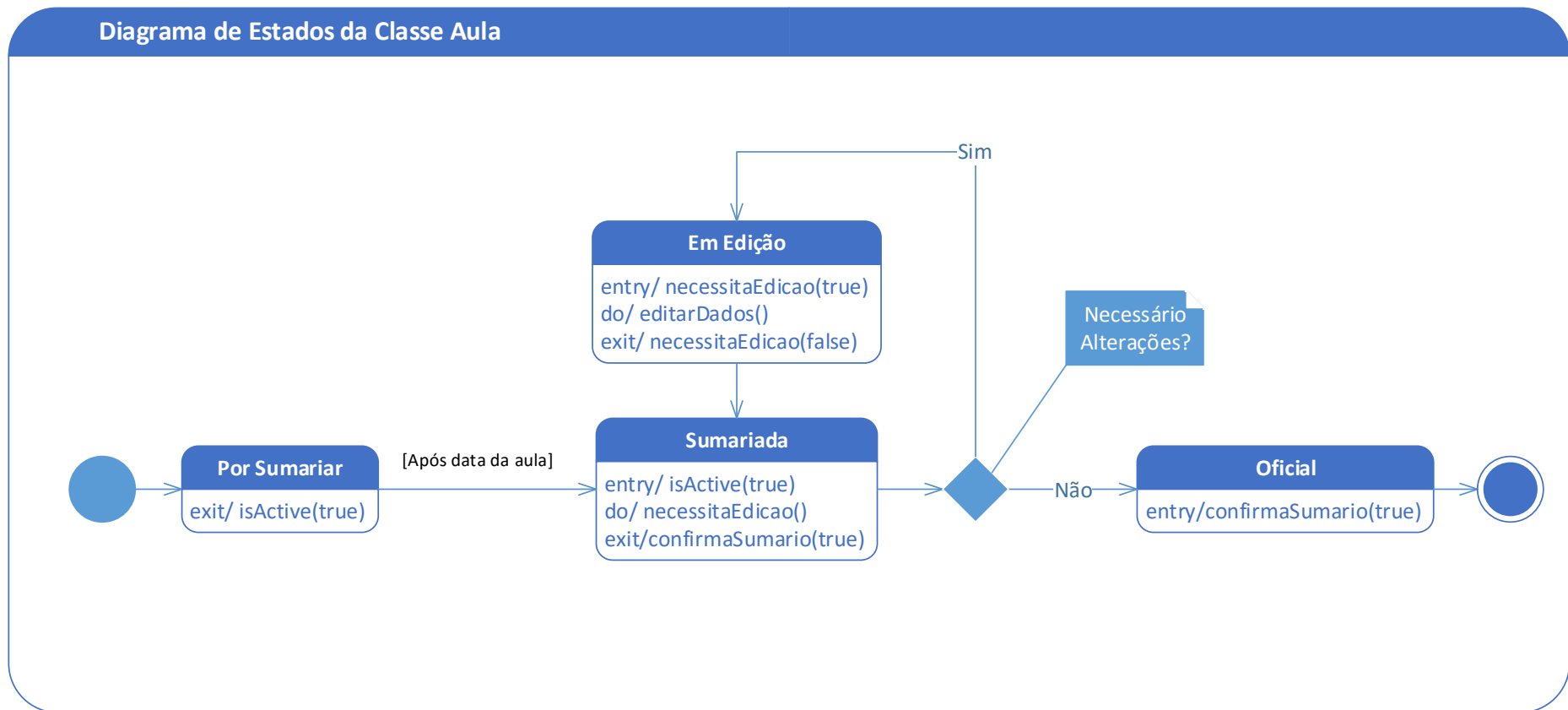
#### 4.2.5. Diagrama de Estados - Classe Grupo

Este diagrama representa os estados por que um objeto da classe Grupo irá passar até atingir o estado Criado. Inicia no estado Em criação onde serão enviados os convites para os possíveis constituintes do grupo, passando para o estado Espera aceitação. Quando recebe uma resposta ao convite é feita a verificação se o convite foi aceite passando para o estado em criação se a resposta for negativa e passando para o estado criado caso a resposta seja afirmativa e o grupo possua elementos suficientes.



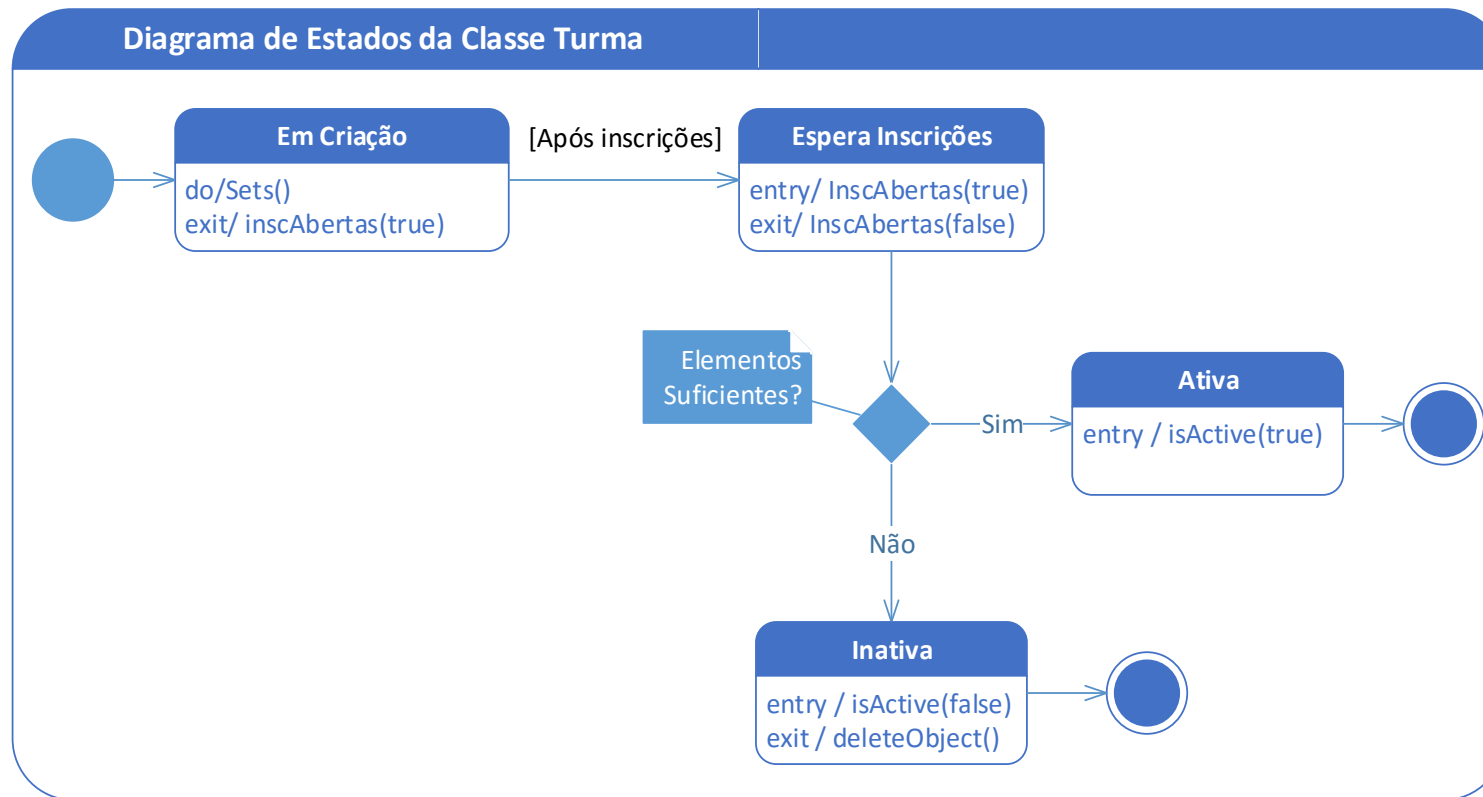
#### 4.2.6. Diagrama de Estados - Classe Aula

Este diagrama representa os estados em que um objeto da classe Aula se pode encontrar no sistema. Inicia no estado Por Sumariar passado para o estado Sumariada após a data da aula. Caso necessite de ser editado passa para o estado Em Edição, caso contrário, e após confirmação, passa para o estado Oficial.



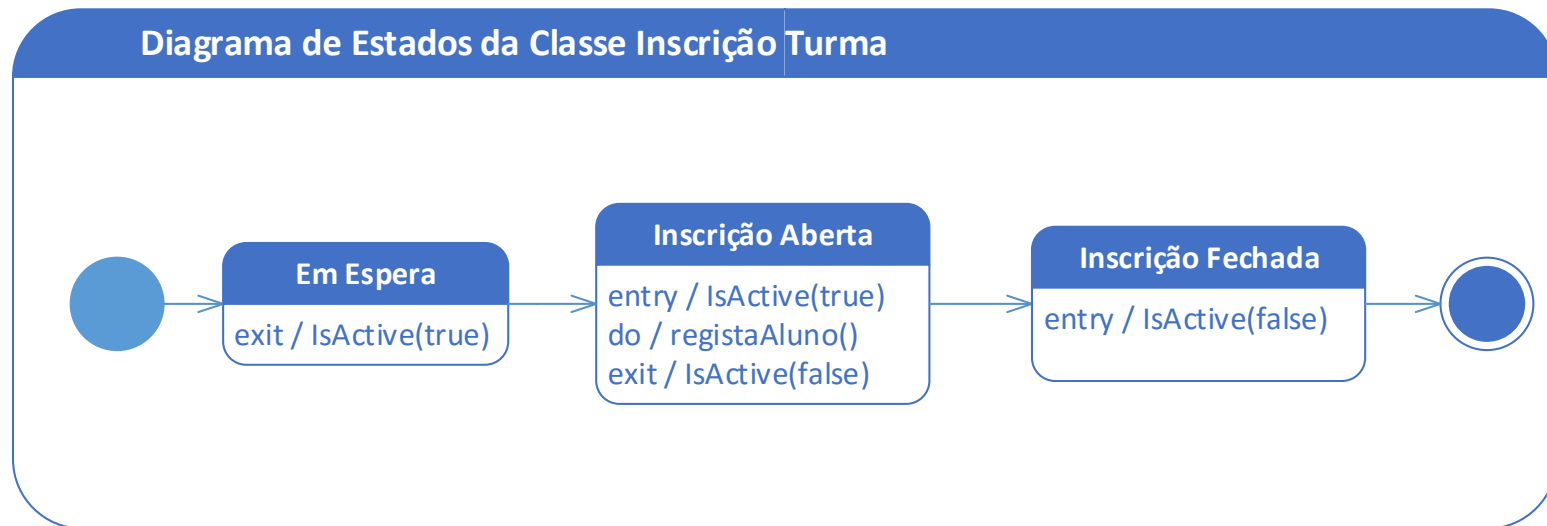
#### 4.2.7. Diagrama de Estados - Classe Turma

Este diagrama representa os estados por que um objeto da classe Turma. Inicia no estado Em criação até abrirem as inscrições passando para o estado Espera Inscrições. Após as Inscrições, se existirem elementos suficientes passa para o estado Ativo, caso contrário passa para o estado Inativo.



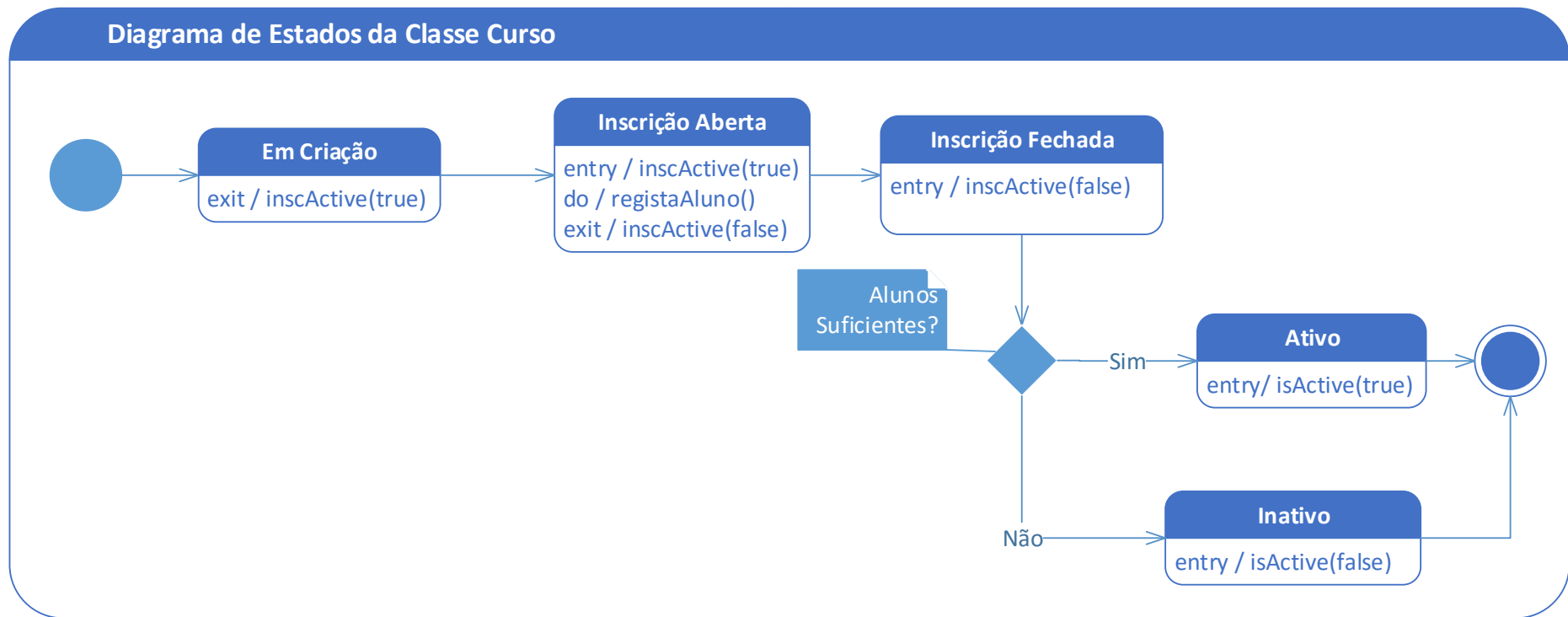
#### 4.2.8. Diagrama de Estados - Classe InscriçãoTurma

Este diagrama representa os estados por que um objeto da classe InscriçãoTurma passa sempre que é necessário realizar a inscrições de elementos numa Turma.



#### 4.2.9. Diagrama de Estados - Classe Curso

Este diagrama representa os estados por que um objeto da classe Curso. Inicia no estado Em criação até abrirem as inscrições. Após as Inscrições, se existirem elementos suficientes passa para o estado Ativo, caso contrário passa para o estado Inativo.





## 5. Análise e Discussão dos Resultados

Após a concretização desta etapa pensamos ter conseguido alcançar os objetivos a que nos propusemos para a 3ª etapa deste trabalho.

Consideramos ter conseguido realizar uma boa diagramação de classes, identificando todas as classes representativas do SIUTAD, na nossa visão do sistema, assim como os relacionamentos das classes e a sua multiplicidade. Os atributos e métodos de cada classe foram também discutidos e achamos que chegamos a uma boa identificação dos mesmos.

Para a diagramação de estados procuramos inicialmente identificar as classes com comportamento dinâmico relevante, e posteriormente fizemos a diagramação, de modo a melhor compreender o comportamento das classes escolhidas.

Assim, pensamos ter atingido os nossos objetivos, chegando a bons resultados.

## 6. Bibliografia

Gouveia, A. Adaptado de Varajão, João. Engenharia de Software – Diagrama de Classes. *Apresentação powerpoint aulas teóricas.*

Gouveia, A. Adaptado de Varajão, João. Engenharia de Software – Diagrama de Estados. *Apresentação powerpoint aulas teóricas.*

Laboratório de Engenharia de Software – PUC Rio. UML: Diagrama de Classes – Projeto de Sistemas de Software. *Apresentação powerpoint aula 1.*

Abdala, Daniel D. Engenharia de Software – Diagrama de Estados. *Apresentação powerpoint.*

Ramos, Pedro e Farinha, José. Fundamentos de Bases de Dados - UML: Diagramas de Classes. *Desenho de Bases de Dados Relacionais com UML*