



Universidade do Minho

Escola de Engenharia

Licenciatura em Engenharia Informática

Unidade Curricular de Laboratórios de Informática IV

Ano Lectivo de 2014/2015

Arit-Mat

**Ana Almeida, Bruno Pereira, João Mano,
Patrícia Rocha**

Data de Recepção	
Responsável	
Avaliação	
Observações	

Arit-Mat

**Ana Almeida, Bruno Pereira, João Mano,
Patrícia Rocha**

Maio, 2015

Resumo

O assistente pedagógico desenvolvido – Arit-Mat – permite auxiliar o estudo/aprendizagem de matemática, nomeadamente, nas operações de adição e subtração, dos alunos do terceiro ano do ensino básico. Para além de ensinar os alunos, esta aplicação permite que os mesmos sejam avaliados, através da realização de fichas de consolidação de conhecimentos e fichas de avaliação referentes aos vários assuntos. Assim, o aluno terá acesso ao seu *feedback* de aprendizagem, isto é, a aplicação com o rastreio das atividades realizadas pelo aluno, calcula estatísticas e informa o aluno sobre o seu desempenho, indicando tanto os assuntos que estão bem assimilados, como aqueles que precisam de ser mais trabalhados, que precisam de mais tempo para ficarem consolidados, garantindo assim o sucesso na aprendizagem.

O projeto será desenvolvido em três etapas distintas, nomeadamente:

- Fundamentação: caracterização da aplicação, levantamento e análise de requisitos de todos os parâmetros necessários ao desenvolvimento do sistema de *software*, por exemplo, viabilidade da “ideia”, concetualização de modelos “guia”, etc.;
- Modelação do sistema: modelar de forma completa todas as vertentes operacionais e funcionais de um sistema de *software* como arquitetura, interface, etc.;
- Desenvolvimento do produto: implementar e testar o sistema de *software* bem como reavaliar todas as especificações realizadas.

Todas estas etapas serão devidamente documentadas e abordadas na documentação.

Área de Aplicação: Matemática: Adição e Subtração no Ensino Básico (3º Ano).

Palavras-Chave: Matemática; Tutor; Contas; Soma; Adição; Aritmética; Aprender; Subtração; Ensino; Básico; Aprendizagem; Assistente; Pedagógico; 3º ano.

Índice

Resumo	i
Índice	ii
Índice de Figuras	iv
Índice de Tabelas	v
1. Introdução	1
1.1. Contextualização	1
1.2. Apresentação do Caso de Estudo	1
1.3. Motivação	2
1.4. Público-Alvo	2
1.5. Objetivos	2
1.6. Protótipo do Software	2
1.7. Protótipo de Logotipo	4
1.8. Diagrama de Gantt	5
2. Levantamento e Análise de Requisitos	6
2.1. Análise de Viabilidade	6
2.2. Levantamento de Requisitos	7
3. Modelação	9
3.1. Modelação Arquitetural	9
3.1.1. Modelo de Domínio	9
3.1.2. Modelo de Casos de Uso	12
3.1.3. Diagramas de sequência	17
3.1.4. Diagramas de Atividade	19
3.1.4.1. Resolução de Teste	19
3.1.4.2. Resolução de Exercício	20
3.1.4.3. Assistir a Lição	20
3.2. Modulação de Componentes	22
3.2.1. Diagrama de Classes	22
3.3. Modelação de Dados	23
3.3.1. Modelo Concetual	23
3.3.2. Modelo Lógico	29
3.3.2.1. Obtenção de Tabelas	30
3.3.2.2. Chaves Estrangeiras	32
3.3.2.3. Primeira Forma Normal	33
3.3.2.4. Segunda Forma Normal	34
3.3.2.5. Terceira Forma Normal	34
3.3.2.6. Restrições de Integridade	34
4. Conclusões e Trabalho Futuro	36

Anexos	40
I. Diagrama de Gantt	1
I.I.1. Fase 1	1
I.I.2. Fase 2	3
II. Modelo de Domínio	5
III. Diagrama de Classes	6
IV. Modelo Concetual	7
V. Modelo Lógico	8
VI. Dicionário de Dados – Entidades	9
VII. Dicionário de Dados – Relacionamentos	10
VIII. Dicionário de Dados – Atributos	11

Índice de Figuras

Figura 1 - Página Inicial	3
Figura 2 - Página inicial do aluno	4
Figura 3 – Logotipos	4
Figura 4 - Logotipo final	5
Figura 5 - Utilizadores do Sistema	9
Figura 6 - Ações do utilizador administrador	10
Figura 7 - Interveniência das funcionalidades de um aluno	10
Figura 8 - Registo de históricos dos alunos	11
Figura 9 - Elementos suporte para armazenar um modelo de ensino	12
Figura 10 - Diagrama <i>Use Case</i> Global	12
Figura 11 - Use Cases de Administração	13
Figura 12 - Especificação do <i>use case</i> “Adicionar Licao”	13
Figura 13 - Especificação do <i>use case</i> “Adicionar Exercicio”	14
Figura 14 - Casos de uso de um aluno	14
Figura 15 - Especificação de “Assistir Licao”	15
Figura 16 - Especificação de “Pedir Ajuda” (nova explicação)	15
Figura 17- Especificação do “Resolver Exercicio”	16
Figura 18 - Especificação de “Pedir Dica”	17
Figura 19 - Diagrama de Sequência “Assistir Licao”	17
Figura 20 - Diagrama de Sequência “Resolver Exercício”	18
Figura 21 - Diagrama de Atividade “Realizar Teste”	19
Figura 22 - Diagrama de Atividade “Realizar exercício”	20
Figura 23- Diagrama de Atividade “Assistir Lição”	22
Figura 24 - Entidade Aluno e seus atributos	23
Figura 25 - Entidade Teste, Exercicio e seu relacionamento	24
Figura 26 - Entidade Resposta e relacionamento com Exercicio	24
Figura 27 - Entidade Licao, atributos e seu relacionamento com Aluno	25
Figura 28 - Entidade Licao e Exercicio e seu relacionamento	25
Figura 29 - Relacionamentos ternários	26
Figura 30 - Atributo DificuldadeTeste de entidade Teste	26
Figura 31 - Entidade Aluno e seu atributo Configuracao	27
Figura 32 - Entidade Tipo	28
Figura 33 - Entidade Administrador	29

Índice de Tabelas

Tabela 1 - Entidades	9
Tabela 2 - Relacionamentos	10
Tabela 3 - Atributos (Aluno)	11
Tabela 4 – Atributos (Licao)	12
Tabela 5 - Atributos (Administrador)	12
Tabela 6 - Atributos (Teste)	13
Tabela 7 - Atributos (Exercicio)	13
Tabela 8 - Atributos (Resposta)	14
Tabela 9 - Atributos (Tipo)	14

1. Introdução

Nesta primeira fase do relatório, é apresentado o caso de estudo que originou o desenvolvimento deste sistema de tutorização, a sua contextualização, motivação e os objetivos, bem como uma breve explicação de todos os parâmetros abordados no restante relatório.

1.1. Contextualização

A matemática, apesar de poder parecer um enigma para grande parte da população, está presente em qualquer atividade quotidiana, desde a economia à própria natureza. Por este motivo, é considerada de extrema importância mas, constantemente mal compreendida pela população mundial.

É vital que desde cedo se motive a população mais jovem a encarar a matemática não como um obstáculo mas como uma ferramenta essencial para a sua evolução que, quando utilizada corretamente, permite que sejam ultrapassadas barreiras, favorecendo assim a resolução de grande parte dos problemas reais, com que somos deparados diariamente.

1.2. Apresentação do Caso de Estudo

Numa sala de aula em que as turmas são compostas por alunos do ensino básico, existe uma preocupação por parte do professor em transmitir conhecimento aos seus educandos. Dadas as várias unidades curriculares do aluno, matemática é a disciplina mais incompreendida e que, por isso, exige um trabalho mais minucioso realizado pelo professor, no sentido de minimizar a incompreensão é necessário ter em conta os materiais utilizados na leccionamento das aulas, isto é, existem para além das aulas expositivas, em que são expostos os conceitos, exercícios que podem ser colocados em fichas de consolidação de conhecimentos, bem como fichas de avaliação para testar esses mesmos conhecimentos. Estes instrumentos de diagnóstico – fichas de consolidação e avaliação – permitem ao professor identificar as verdadeiras dificuldades dos alunos, podendo este voltar a ensinar os assuntos que ficaram menos consolidados.

1.3. Motivação

A grande maioria da população possui graves dificuldades na resolução e compreensão de problemas matemáticos. A causa é normalmente um défice de aprendizagem desde os primeiros contatos com a matemática. Este foi o estímulo que levou a nossa equipa de trabalho a procurar desenvolver um assistente pedagógico nesta área.

Como as principais operações de aritmética correspondem, nomeadamente, à soma e subtração de algarismos, foi pensado que este assistente pedagógico deve focar-se na disponibilização de material sobre estes subtemas, por forma a incitar e proporcionar a compreensão dos mesmos, de forma dinâmica e interativa, porque é certo que todos os alunos se sentem muito mais interessados em aprender através de tecnologias de informação, pelas razões que foram expostas, principalmente a interatividade.

1.4. Público-Alvo

Dado o exposto, o público-alvo desta aplicação concentra-se nos alunos que se encontrem no terceiro ano do ensino básico, de qualquer idade, dado que o importante é que estes estejam sujeitos a um plano curricular nos temas ou subtemas de adição e subtração.

1.5. Objetivos

O objetivo principal, tendo em conta o caso de estudo apresentado, é desenvolver um assistente pedagógico capaz de ensinar as principais operações aritméticas – soma e subtração – a alunos entre os oito e nove anos de idade e documentar todo o processo envolvente.

Este assistente deverá oferecer todo o conhecimento necessário à consolidação dos temas soma e subtração, quebrando a dependência da existência de um professor humano. A independência referida permite aos alunos que aprendam ao seu ritmo próprio e em qualquer altura, de facto, a maximização do processo de aprendizagem é apenas lógica, é porém totalmente dependente da vontade, trabalho e esforço do aluno.

1.6. Protótipo do Software

Nesta fase inicial de desenvolvimento do produto, a interação do assistente pedagógico com o público-alvo, pretende-se que seja realizada da seguinte forma:

- O utilizador acede à aplicação e visualiza a página inicial – Figura 1 – composta por:
 - Um menu na parte superior da página;

- Abaixo deste menu, no canto superior direito da página, é dada a possibilidade de registo ou autenticação no sistema;
 - No centro da página é exibida uma pequena descrição da aplicação;
 - Por fim existe um *avatar* no canto inferior direito que a qualquer altura pode ser ativado ou desativado.
- Na página do utilizador aluno – Figura 2 – é exibido um menu que, mediante a opção encaminha o utilizador para a página escolhida.
 Caso seja escolhida a opção “lições” é apresentada uma página com todos os temas disponíveis para a aprendizagem, se a opção for “testes” é lhe dada a oportunidade de testar os seus conhecimentos, por fim aquando selecionada a opção “histórico”, o aluno visualiza o seu desempenho de todas as atividades realizadas até então.
 Para além destas opções, dado que a faixa etária do público-alvo é bastante reduzida, existem no canto inferior esquerdo ícones de personalização do fundo de ambiente de trabalho, por forma a tornar mais apelativa a utilização da aplicação.
 Para terminar centradas na página estão situadas duas caixas de texto relativas a informações sobre o utilizador em questão, desde o estado de conclusão das diversas atividades de aprendizagem.



Figura 1 - Página Inicial

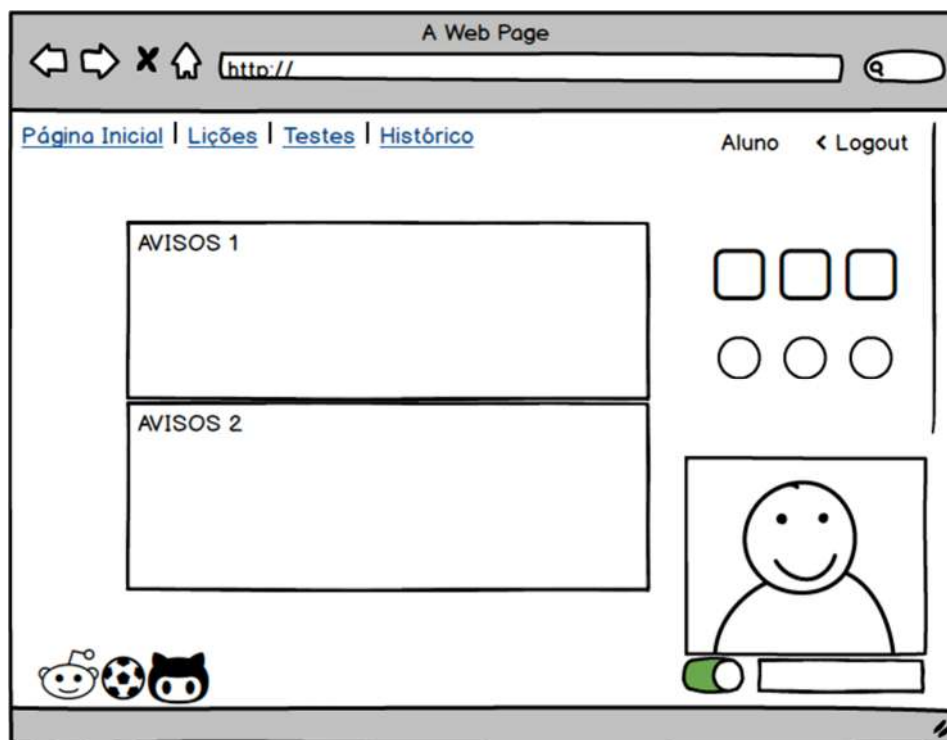


Figura 2 - Página inicial do aluno

1.7. Protótipo de Logotipo

Uma das características mais marcantes de um produto é o seu logotipo, funciona como a assinatura do produto, visualmente, reforça o significado do produto perante os seus utilizadores, por esta razão foram criados alguns protótipos de logotipos para a aplicação Arit-Mat – Figura 4.



Figura 3 – Logotipos

Depois de uma análise estética e um pouco intuitiva optou-se pelo seguinte logotipo para o produto:



Figura 4 - Logotipo final

1.8. Diagrama de Gantt

Apresentado o caso de estudo, é importante para o desenvolvimento da aplicação a calendarização de todas as atividades associadas ao processo de desenvolvimento do sistema. Assim, depois de identificadas todas as tarefas e, dado o tempo disponível para a conclusão da aplicação, no Anexo 1 apresenta-se o gráfico onde se visualiza o tempo, em dias, que cada tarefa exige, isto é o avanço das diferentes tarefas do projeto, permitindo assim o controlo do desenvolvimento do mesmo.

2. Levantamento e Análise de Requisitos

2.1. Análise de Viabilidade

Por forma a analisar e certificar o sucesso do tutor, é de extrema importância a realização de uma estimativa relativamente ao futuro uso da aplicação, que ditará o avanço ou não do processo de desenvolvimento do mesmo. Esta estimativa baseia-se no estudo do sistema, as suas políticas, funções, objetivos, etc.

Para tal, é necessário que se levem a cabo testes de viabilidade do produto. Neste caso particular foram realizados os seguintes testes:

1. **Viabilidade Operacional** – identificação da viabilidade do sistema, isto é, se a aplicação é de facto uma boa solução ao problema e qual a reação dos utilizadores alvo a esta aplicação.

Face aos atuais resultados de matemática na sociedade, achou-se claro que o método atual de ensino seria drasticamente melhorado aquando a implementação desta solução. Outro motivo para continuar a desenvolver este produto será eliminar a dependência de uma aprendizagem assistida e apenas disponível num dado período de tempo, tornando o acesso à educação de somas e subtrações muito mais fácil e independente cujo acesso está ao controlo do aluno.

Para além destes dois estímulos à continuidade do produto, apesar da extensibilidade e flexibilidade da aplicação estar dependente do acompanhamento dos programadores da aplicação, dado que não existe qualquer custo associado à obtenção deste produto e que é maximizada a eficiência do processo (quebra de dependências, etc.), existe uma motivação forte para o seguimento do projeto.

2. **Viabilidade Técnica** – neste teste é avaliado se existe acesso a todo o conhecimento requerido ao desenvolvimento do tutor.

Com o acesso a ferramentas de programação como Microsoft Visual Studio e ferramentas de documentação próprias como Visual Paradigm Professional torna-se possível a construção deste assistente pedagógico.

Além do acesso à tecnologia é preciso reconhecer se os responsáveis pela evolução e construção do programa são capazes e possuem conhecimentos para o sucesso da concetualização da aplicação. Visto que a equipa responsável tem experiência no processo de desenvolvimento de produtos de *software* e conhecimento em diversas linguagens de programação e os seus princípios, é obtido neste teste um resultado favorável ao andamento do projeto.

3. **Viabilidade de Cronograma** – nesta fase é avaliado se todo o processo é passível de conclusão dentro dos prazos estipulados.

Apesar de existir experiência com desenvolvimento de aplicações informáticas por parte da equipa, esta experiência é pouca e raramente obtida sobre projetos de tão larga escala o que torna mais desafiador e difícil conseguir definir que tipo de funcionalidades e características são, efetivamente, possíveis de realizar dentro dos prazos do projeto.

Tendo isto em conta, apesar da pouca experiência, foi estipulada uma estrutura completa e, acima de tudo, cujo término fosse possível dentro dos prazos limites.

4. **Viabilidade Económica** – o último teste de viabilidade, consiste em estudar a dependência monetária do projeto e, perante esta decidir se os custos de desenvolvimento são compensados pelos possíveis lucros e benefícios da aplicação desenvolvida.

Neste projeto não existem custos de desenvolvimento do sistema, custos operacionais nem qualquer custo esperado no desenvolvimento de um projeto, existe apenas gasto de tempo por parte da equipa de desenvolvimento e, futuramente, caso a aplicação se torne bem-sucedida com o treino dos utilizadores e instalação do tutor.

O facto de não existirem custos concretos associados ao projeto ajuda, mais uma vez, à decisão de prosseguir com o desenvolvimento da aplicação.

Depois de realizados e analisados estes testes, visto que na generalidade todos os resultados são motivadores, a decisão tomada é levar a cabo o projeto.

2.2. Levantamento de Requisitos

A compreensão e estipulação de todas as tarefas a realizar e cumprir no desenvolvimento da aplicação é uma parte essencial e crucial no início da projeção do programa. É através destas que a equipa se guiará durante a construção da aplicação, evita perdas de tempo e facilita a estruturação do processo.

Após diversos encontros com o cliente, onde foi definido verbalmente o que era pretendido, depois de analisadas e estudadas as instruções recebidas e, adicionando funcionalidades não referidas, mas fundamentais neste contexto, encontram-se os requisitos funcionais finais deste processo de desenvolvimento:

1. Registrar utilizadores mantendo um registo da sua informação;
2. Associar exercícios de acordo com o grau de dificuldade e tema e, conjugar os mesmos da melhor forma gerando testes de avaliação do conhecimento e progresso dos alunos;
3. Associar um grau de dificuldade a cada exercício para conseguir, por parte do sistema, fornecer exercícios direcionados ao grau de aprendizagem dos utilizadores/alunos, e, por parte dos utilizadores, analisar o desempenho devidamente associado à dificuldade da questão/exercício;
4. Os exercícios devem ser associados a respostas de escolha múltipla;

5. Armazenar e disponibilizar informação educativa sobre os temas adição e subtração que será disponibilizada para uso pessoal dos utilizadores, tendo como objetivo acompanhar e facilitar a compreensão da matéria;
6. Associar a cada uma destas informações (lições) exercícios de consolidação que os alunos poderão realizar;
7. Permitir a realização de exercícios em testes e lições por parte dos alunos;
8. Associar os testes a níveis, disponibilizados para os alunos por forma a avaliar o seu conhecimento de um determinado assunto;
9. Armazenar um histórico de dados associado a entidades particulares (utilizadores) com o intuito de melhor adaptar o produto a cada utilizador;
10. Capacitar a personalização do programa face a determinado utilizador, para que melhor se ajuste às suas necessidades e gostos;
11. Disponibilizar um modelo de aprendizagem individual.
12. Analisar o histórico de cada utilizador e, consoante o mesmo, sugerir diversas lições propícias à consolidação e melhoria das áreas de aprendizagem menos bem “pontuadas” em função do histórico analisado;
13. Agregar o histórico de todos os utilizadores, permitindo a realização de estatísticas sobre a matéria e determinar quais os temas mais problemáticos, quais as questões com maior número de falhas, etc.;
14. Registrar qual o estado de aprendizagem de um aluno por forma a identificar o seu progresso;
15. Gestão de diferentes elementos de dados (texto, imagem, som, vídeo, etc.) associados às funcionalidades do tutor, por forma a gerar uma experiência de aprendizagem mais adequada e acessível à perceção;
16. Permitir a gestão de lições (introdução e edição) por um utilizador administrador de modo a que o sistema consiga expandir o número de lições existentes.

Para além destes registos funcionais foram identificados ainda os seguintes requisitos não funcionais:

17. A dificuldade dos exercícios é estabelecida por um inteiro que pode tomar valores entre 1 e 10;
18. Uma lição tem no máximo 20 minutos de exposição de matéria;
19. Cada lição tem de apresentar uma e uma só explicação cujo formato é estabelecido em texto, imagem ou vídeo;
20. Um exercício é apresentado em formato texto ou imagem;
21. Tempo de teste máximo é calculado pela dificuldade dos seus exercícios utilizando uma fórmula bem definida;
22. Existem no mínimo 2 temas que os utilizadores podem escolher para personalizar o seu tutor;

23. Simplificar a interface do produto para facilitar o uso do mesmo e tornar mais agradável e simples o processo de aprendizagem;
24. Fornecer informação detalhada sobre o produto e quais as suas funcionalidades, de modo a esclarecer os utilizadores e encaminhar a uma devida utilização do produto;
25. Os exercícios disponibilizados devem ser associados a exatamente 4 respostas das quais apenas uma será a resposta correta.

3. Modelação

Terminada a identificação e registo de todos os requisitos da aplicação segue-se a modelação do sistema.

Os modelos apresentados nos próximos subcapítulos especificam em detalhe todo o sistema e o seu funcionamento, documentando tudo o que é pretendido desenvolver ao longo do desenvolvimento do produto.

3.1. Modelação Arquitetural

3.1.1. Modelo de Domínio

Analisando o caso de estudo e todos os componentes envolvidos foi construído o modelo de domínio respetivo – Anexo II – seguidamente, são enunciadas, dividindo o modelo domínio em partes, todas as noções que levaram à construção do modelo apresentado.

Começando pelos utilizadores do sistema, aluno e administrador, tanto um aluno como um administrador são utilizadores e, um utilizador possui um nome de utilizador e uma palavra passe obrigatoriamente, logo, no modelo de domínio tem-se:

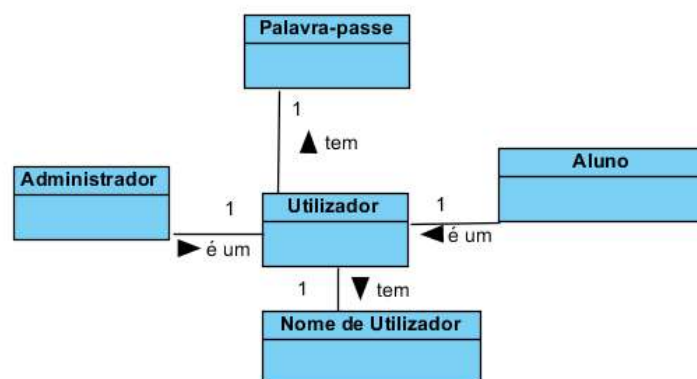


Figura 5 - Utilizadores do Sistema

O administrador tem uma ação bastante mais reduzida que os utilizadores aluno. O administrador apenas faz a gestão de lições e exercícios:

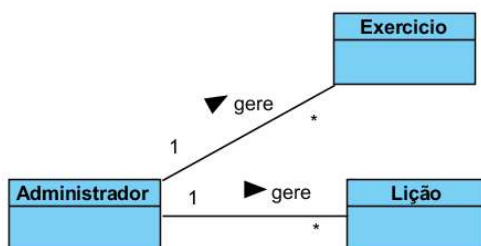


Figura 6 - Ações do utilizador administrador

No caso do aluno existem várias influências do mesmo no sistema.

Um aluno realizará lições e testes e, tanto os testes como as lições possuem exercícios, estes exercícios têm uma dificuldade que, por sua vez, gera no teste também uma dificuldade. Um exercício tem, ainda, uma resposta que poderá ser certa ou errada. Por último, aos exercícios e lições está associado um tipo, de forma a identificar que área da matéria está a ser apresentada naquele elemento.

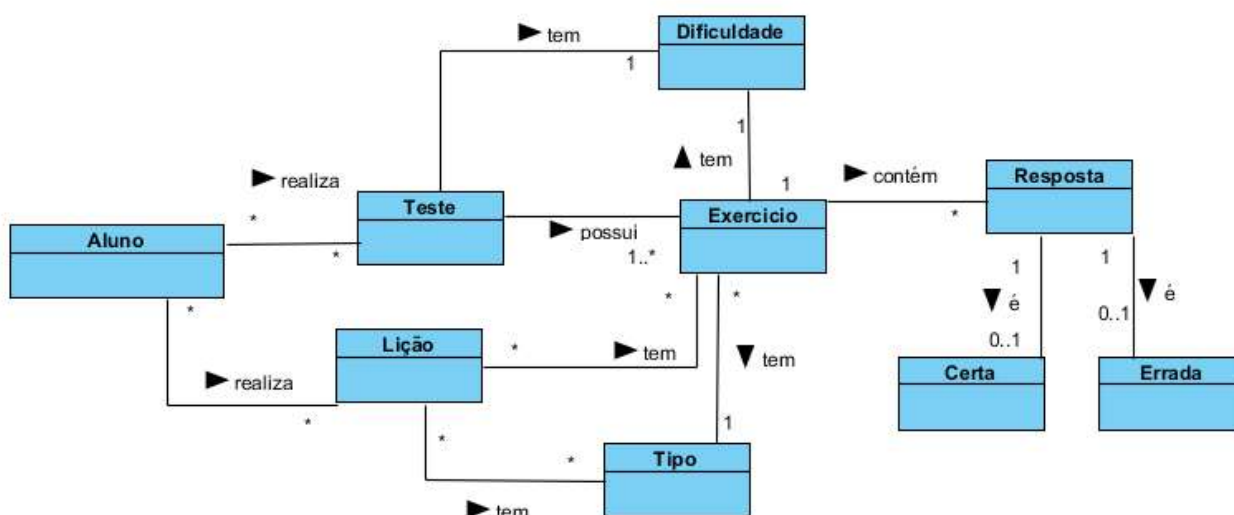


Figura 7 - Interveniência das funcionalidades de um aluno

Os restantes elementos foram criados pela necessidade de um modelo de aprendizagem, armazenamento de histórico sobre o aluno e existência de um “caminho” regular de apresentação de matéria.

Dado isto, existe no modelo, com o intuito de identificar o nível de aprendizagem de um aluno, um registo de histórico que contém vários históricos com uma data e um resultado. Este histórico é sobre uma lição, teste ou exercício. Possui também uma pontuação, gerada pela dificuldade dos exercícios realizados, que constitui um *ranking* de pontuações de alunos. Este *ranking* pode ser consultado pelos mesmos.

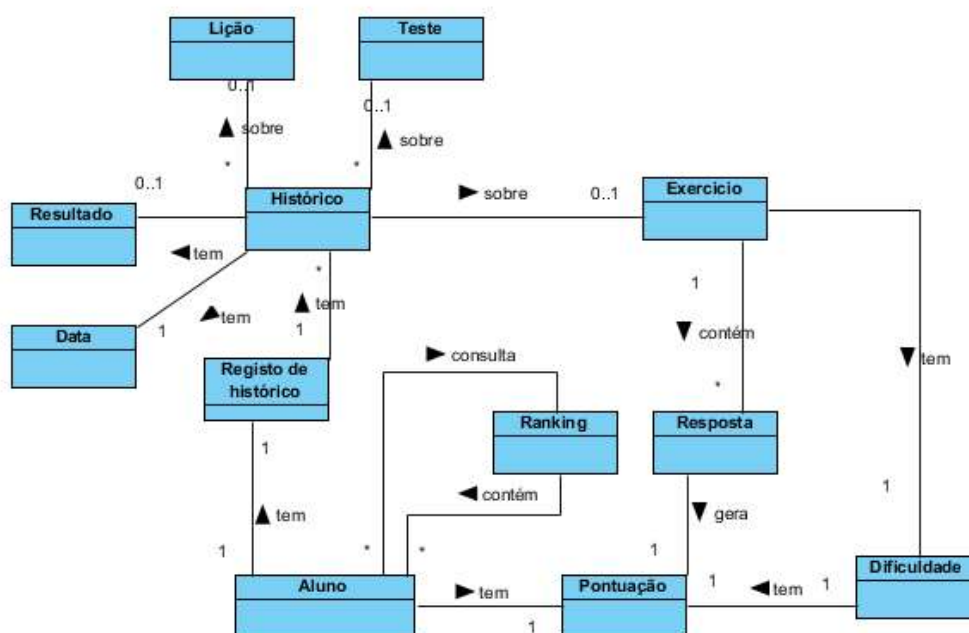


Figura 8 - Registo de históricos dos alunos

Estes elementos fornecem algum suporte à existência de um modelo de ensino. Porém, é necessário que exista, ainda, um registo de aprendizagem de cada aluno que conterá um agregado de estatísticas geradas pelos históricos acima mencionados, sendo permitida a visualização deste por parte dos alunos. Este registo de aprendizagem possui um tipo, do mesmo domínio dos tipos das lições e exercícios, por forma a registar a aprendizagem por tipo/tema de matéria ensinada.

Às lições é associado um número de ordem que estabelecerá a resposta a questões, que lição deverá ser apresentada de seguida e, que lição recomendar quando, na atual, o aluno apresentar um conhecimento demasiado baixo.

Estruturalmente os exercícios são um elemento de estudo que podem encontrar-se no formato texto, imagem ou vídeo. Os exercícios possuem várias dicas, de maneira a orientar o aluno para conceitos e regras que se pretende que o mesmo aprenda. Estas dicas são texto ou imagem.

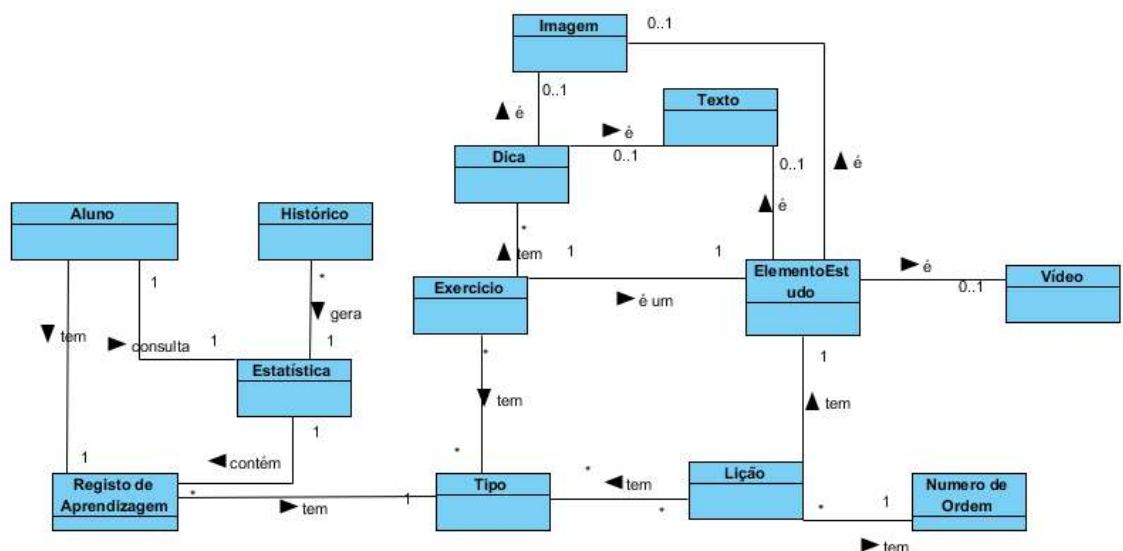


Figura 9 - Elementos suporte para armazenar um modelo de ensino

3.1.2. Modelo de Casos de Uso

Da análise às funcionalidades e requisitos que a aplicação deve possuir e satisfazer, foram gerados os diagramas de casos de uso. Estes descrevem em linguagem textual, os cenários esperados aquando uma interação do utilizador com o sistema.

O seguinte diagrama corresponde ao diagrama global que ilustra todas as interações existentes entre utilizador e sistema. Neste, apenas “Login”, “Logout” e “Registar” são casos de uso finais, ou seja, não representam subdiagramas de casos de uso.

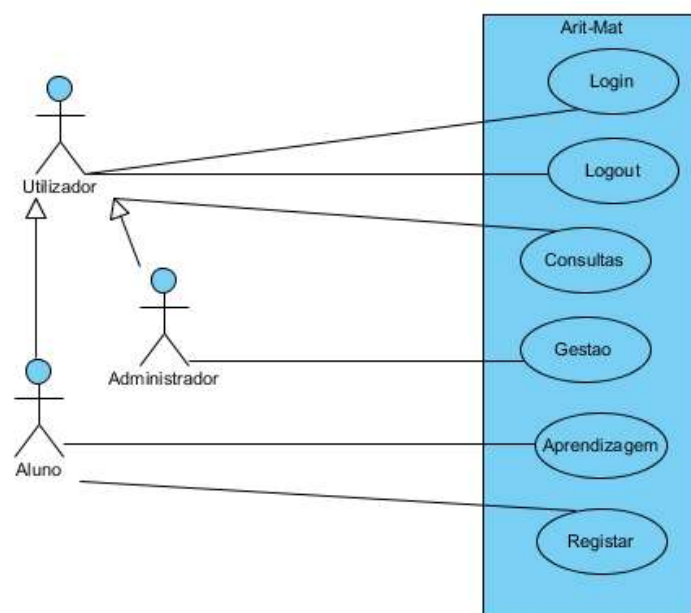


Figura 10 - Diagrama Use Case Global

Dada a simplicidade destes três *use cases* não será apresentada a sua especificação.

Do caso de uso “Gestao” é obtido o diagrama que corresponde à interação privilegiada de um administrador no sistema. Este conduz a mais dois casos de uso representativos de dois diagramas, que correspondem à gestão de lições e exercícios.

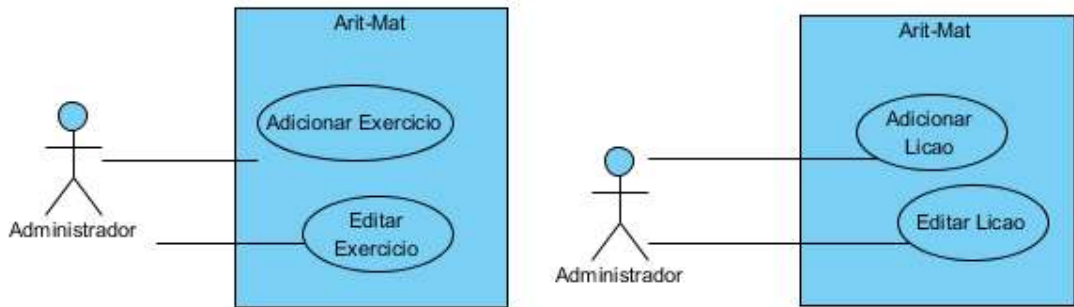


Figura 11 - Use Cases de Administração

Dada a simplicidade dos restantes apenas será apresentada a especificação do *use case* “Adicionar Licao” e “Adicionar Exercicio”.

Super Use Case	Adicionar Licao		
Author	Grupo 1		
Date	19/Abr/2015 17:09:32		
Brief Description	Permite adicionar uma licao ao sistema		
Preconditions	Administrador autenticado		
Post-conditions	Licao e adicionada ao sistema		
Flow of Events		Actor Input	System Response
	1	Informa que quer adicionar uma licao	
	2		Apresenta tipo de licoes
	3	Informa qual o tipo de licao	
	4		Pergunta se e nova licao ou apenas nova explicacao de uma licao existente
	5	Responde	
	6		Apresenta modos de apresentacao de licao
	7	Escolhe modo	
	8		Pede para inserir o elemento de estudo
	9	Inserir informacao	
	10		Processa informacao e intoduz licao

Figura 12 - Especificação do use case “Adicionar Licao”

Super Use Case	Adicionar Exercício	
Author	Grupo 1	
Date	19/Abr/2015 17:17:58	
Brief Description	Permite adicionar um exercício ao sistema	
Preconditions	Administrador autenticado	
Post-conditions	Exercício e inserido no sistema	
Flow of Events		Actor Input
		System Response
	1	Informa que quer adicionar exercício
	2	Apresenta tipos
	3	Escolhe tipo de exercício
	4	Apresenta lições desse tipo
	5	Escolhe a que lição quer associar o exercício
	6	Apresenta modos de apresentação de exercício
	7	Escolhe modo de apresentação
	8	Pede para inserir elemento
	9	Inserir elemento
	10	Processa e introduz exercício

Figura 13 - Especificação do use case “Adicionar Exercício”

O caso de uso “Aprendizagem”, do diagrama global inicialmente apresentado, corresponde às interações do utilizador aluno com o sistema.

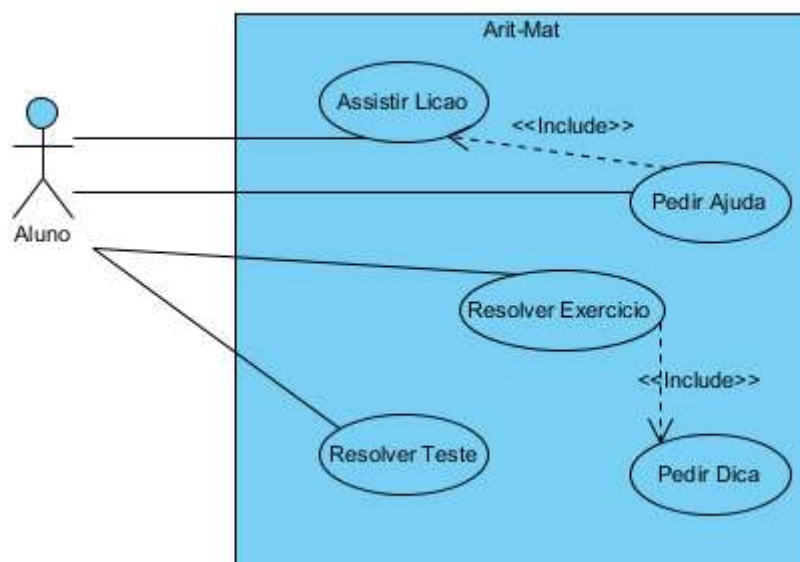


Figura 14 - Casos de uso de um aluno

Para este diagrama serão apresentadas todas as especificações pois consideram-se de grande importância. É de notar que todos os casos de uso implementados têm em conta a existência do

modelo de aprendizagem estipulado (ver diagrama de atividades – capítulo 3.1.4), daí que no caso de uso “Assistir Licao” seja incluído o caso de uso de “Pedir Ajuda”. Este diagrama corresponde à interação de um aluno que, depois de assistir a uma lição, não entendeu e quer uma explicação diferente. Assim, é apresentada uma nova lição para que este assista. O mesmo é feito para o caso de uso de realizar exercício, em que um aluno pode pedir ajuda.

Super Use Case	Assistir Licao	
Author	Grupo 1	
Date	10/Abr/2015 16:10:12	
Brief Description	Permite a um aluno assistir a uma dada lição	
Preconditions	Aluno autenticado	
Post-conditions	Aluno assiste a lição	
Flow of Events		Actor Input
	1	Aluno informa que quer assistir a dada lição
	2	
	3	Assiste a lição
		System Response
		Apresenta lição com número de explicação correspondente

Figura 15 - Especificação de “Assistir Licao”

Super Use Case	Pedir Ajuda	
Author	Grupo 1	
Date	10/Abr/2015 16:12:25	
Brief Description	Permite a um aluno pedir auxilio numa lição	
Preconditions	Aluno autenticado e assistiu a uma lição com uma dada explicação	
Post-conditions	Aluno recebe auxilio	
Flow of Events		Actor Input
	1	Aluno pede ajuda
	2	
	3	
		System Response
		Calcula numero da explicação da lição
		<<Include>> Assistir Licao
Excecao 1 (Passo 2) [Sem mais explicacoes]		Actor Input
	1	
	2	
	3	Aceita
	4	
		System Response
		Informa que nao possui mais explicacoes
		Recomenda assistir a lição anterior
		Atualiza numero de lição

Figura 16 - Especificação de “Pedir Ajuda” (nova explicação)

Use Case Descriptions:

Super Use Case	Resolver Exercício	
Author	Grupo 1	
Date	10/Abr/2015 16:16:50	
Brief Description	Resolução de um exercício por parte do aluno	
Preconditions	Aluno autenticado e lição correspondente assistida	
Post-conditions	Aluno realiza exercício e resposta registrada	
Flow of Events		Actor Input
	1	Informa que quer realizar exercício
	2	
	3	
	4	Realiza exercício
	5	
	6	
	7	
	8	
	9	
Excecao 1 (Passo 2) [Ja realizados todos os exercicios]		System Response
	1	Informa que os exercicios ja foram todos realizados
Alternativa 1 (Passo 4) [Pede dica]		Actor Input
	1	
	2	
Alternativa 2 (Passo 5) [Resposta errada e numero de respostas erradas menor que minimo]		System Response
	1	Informa que errou exercício
	2	Regressa a passo 8
Alternativa 3 (Passo 5) [Resposta errada e numero de respostas erradas entre minimo e maximo]		Actor Input
	1	
	2	
	3	
	4	
Alternativa 4 (Passo 5) [Resposta errada e numero de respostas erradas maior que maximo]		System Response
	1	Informa que errou exercício
	2	Recomenda voltar a assistir a lição com nova explicação
	3	Atualiza numero de explicação
	4	Regressa a passo 8
Excecao 2 (Passo 4) [Excedeu tempo de realizacao]		Actor Input
	1	
		System Response
	1	Informa que excedeu o limite de tempo

Figura 17- Especificação do “Resolver Exercício”

Super Use Case	Pedir dica	
Author	Grupo 1	
Date	10/Abr/2015 16:41:30	
Brief Description	Permite ao aluno pedir ajuda num exercicio	
Preconditions	Aluno autenticado e exercicio em execucao	
Post-conditions	Ajuda fornecida	
Flow of Events		Actor Input
		System Response
	1	Informa que precisa de dica
	2	Determina dicas ainda nao apresentadas
	3	Apresenta dica
Excecao 1 (Passo 2) [Dicas esgotadas]		Actor Input
	1	Informa que nao existem mais dicas

Figura 18 - Especificação de “Pedir Dica”

3.1.3. Diagramas de sequência

Após a construção dos diagramas de caso de uso foram construídos dois diagramas de sequência, considerados de maior importância.

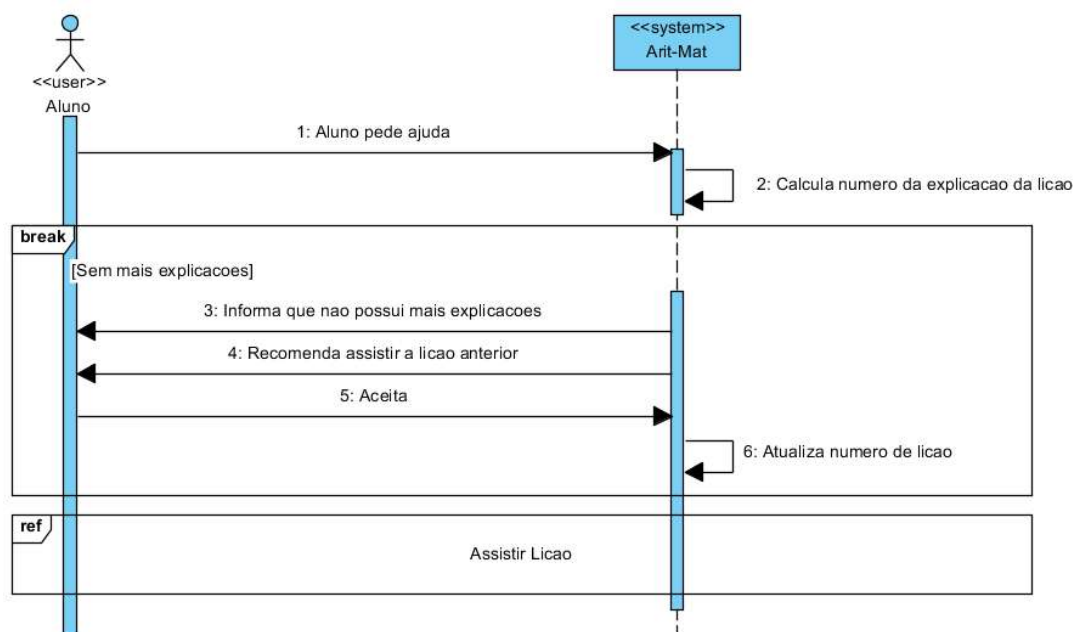


Figura 19 - Diagrama de Sequência “Assistir Licao”

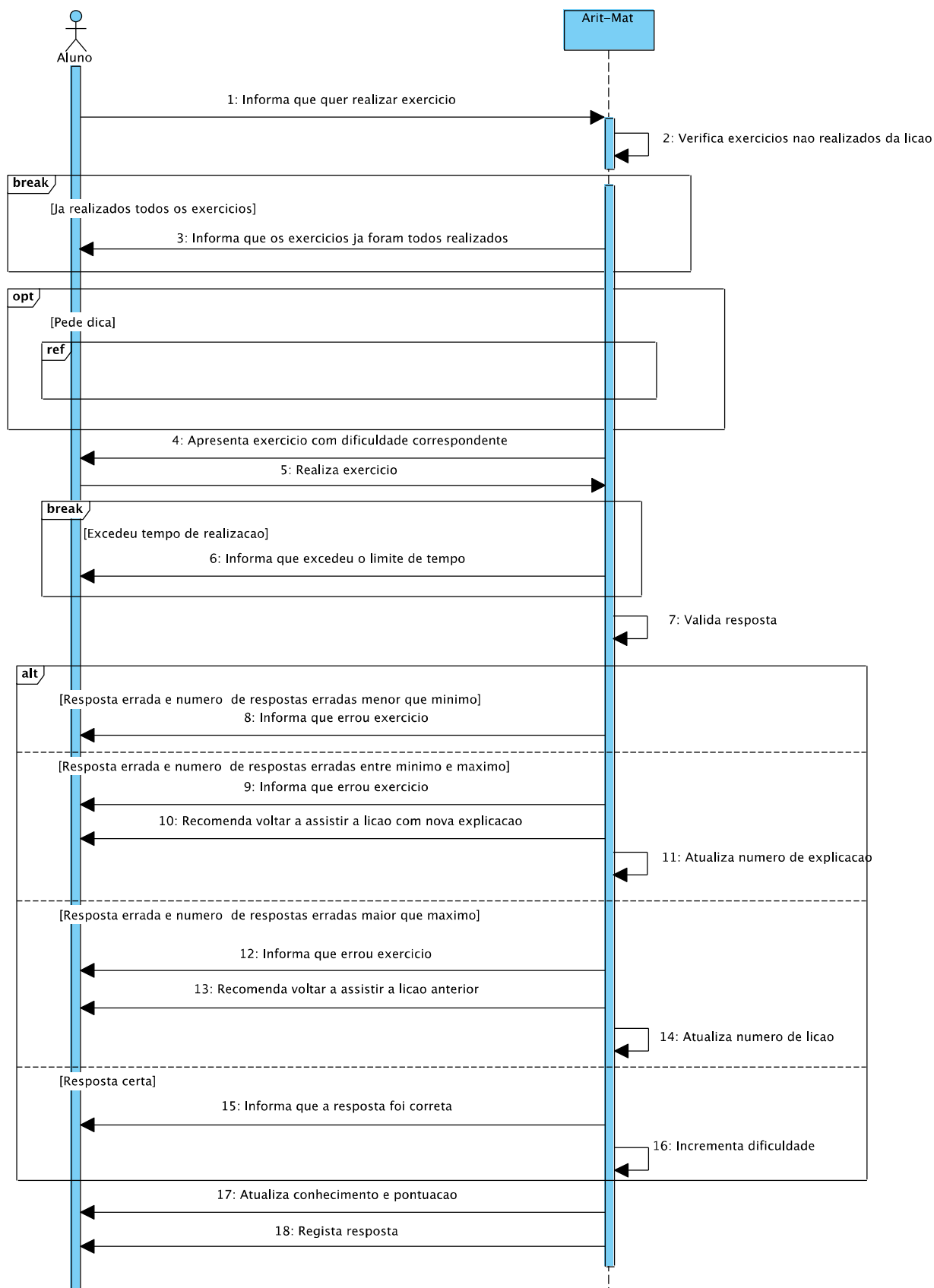


Figura 20 - Diagrama de Sequência “Resolver Exercício”

3.1.4. Diagramas de Atividade

Por forma a modular o processo de aprendizagem que se pretende seguir, foram construídos diagramas de atividade. Estes diagramas explicam todo o método envolvido no ensino de um tema/matéria do sistema.

Primeiramente, é apresentado o diagrama de atividade correspondente à realização do teste, seguido do diagrama de atividade da realização de um exercício e, finalmente, o mais complexo dos diagramas construídos, o diagrama que modela o método de ensino para a lição.

3.1.4.1. Resolução de Teste

Um processo de resolução de um teste inicia-se com a apresentação dos níveis de dificuldade para que o aluno possa escolher, depois da escolha é apresentado o enunciado do teste com a dificuldade pretendida. O aluno tem uma hora para realizar o teste e no final é calculado o resultado, atualizado o conhecimento em função do mesmo e apresentado o resultado ao aluno. Este processo origina o seguinte diagrama de atividade:

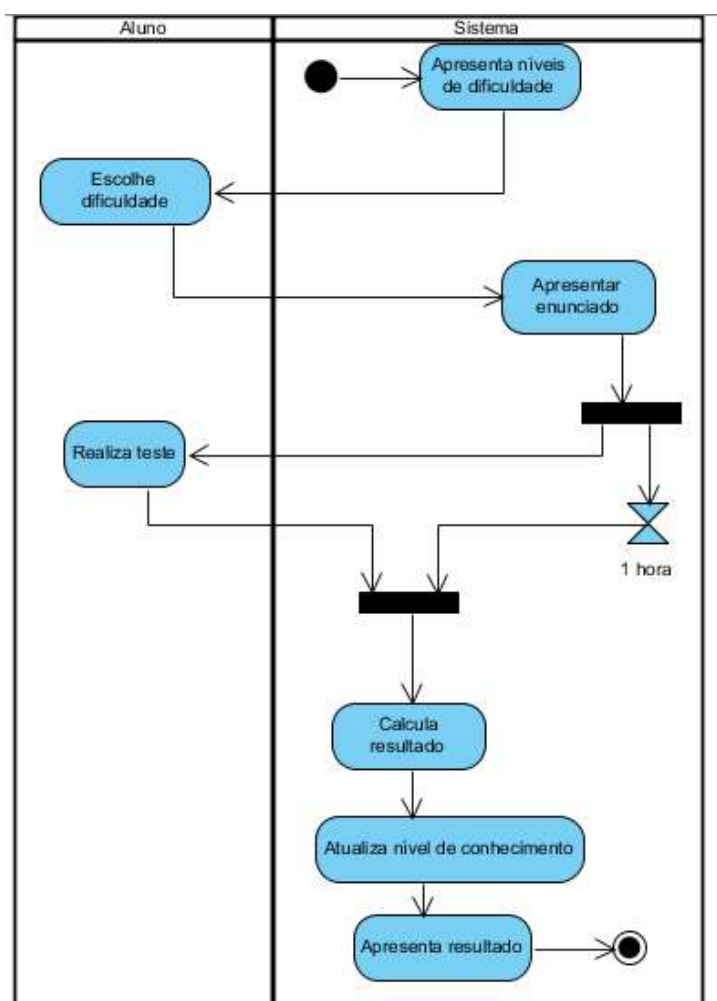


Figura 21 - Diagrama de Atividade "Realizar Teste"

3.1.4.2. Resolução de Exercício

Na realização de um exercício é apresentado o exercício. Caso tenha dúvidas, o aluno pode pedir uma dica (as dicas são limitadas e começam a repetir-se quando já foram todas expostas) quantas vezes desejar ou então resolve o exercício. Ao mesmo tempo, está a contabilizar-se o tempo decorrido e, caso este ultrapasse o permitido para aquele exercício o sistema obrigatoriamente termina e analisa a resposta introduzida (não tendo resposta considera-se resposta errada).

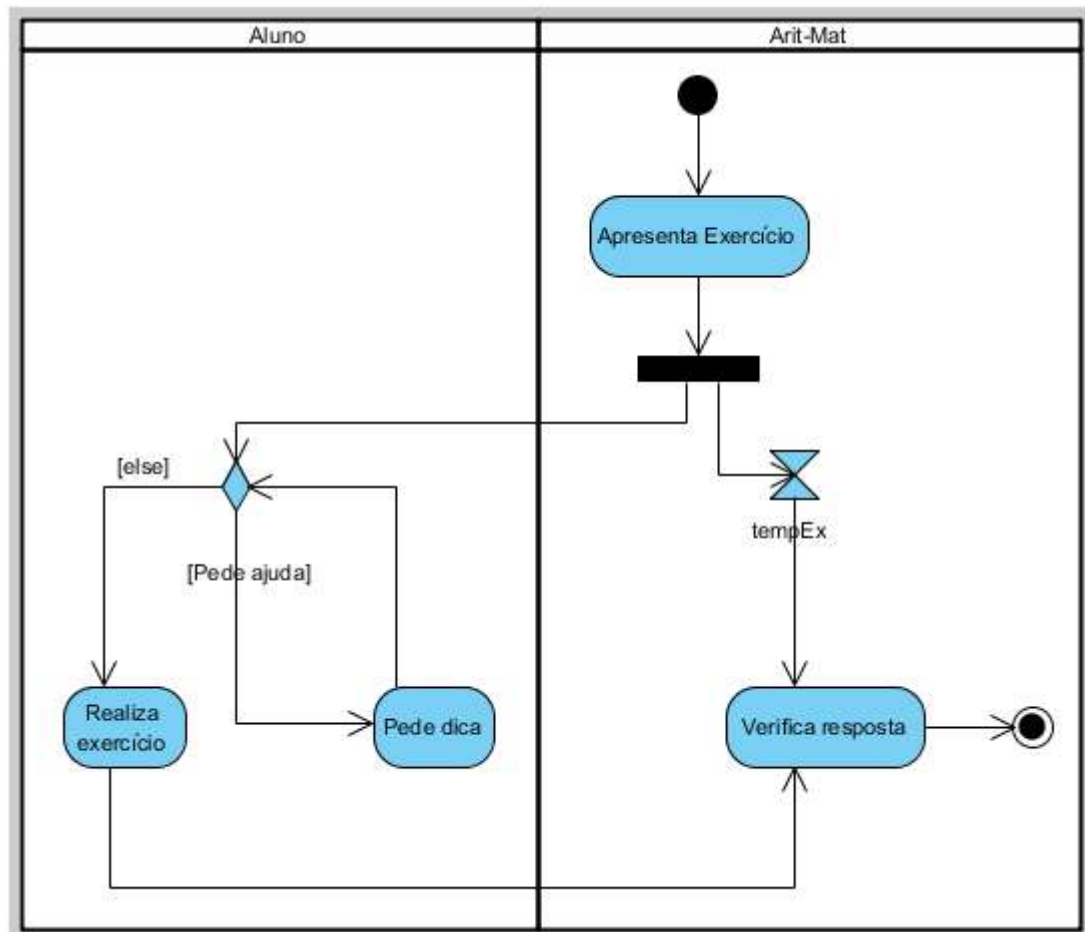


Figura 22 - Diagrama de Atividade "Realizar exercício"

3.1.4.3. Assistir a Lição

Na realização de uma lição é apresentada a lição correspondente ao NumLicao e NumExpl, note-se que estes são os identificadores de uma lição, sendo que o primeiro fornece um tipo de lição associado a um tema e, o segundo corresponde a uma dada explicação desse tema. Em conjunto formam uma lição particular.

O aluno assiste à lição durante o tempo que essa lição permite e, no final, são apresentadas as opções.

Caso este não tenha percebido e queira uma nova explicação sobre este tema, é atualizado o NumExpl e é apresentada a nova lição.

Caso queira repetir, a mesma lição é apresentada.

Caso queira avançar, é atualizado o NumLicao e termina processo, sendo que, posteriormente, será inicializado um novo processo de aprendizagem sobre a lição seguinte.

Caso o aluno tenha entendido a lição, é determinado qual a dificuldade do exercício a apresentar (pelo histórico da dificuldade dos exercícios realizados neste tema), e é realizado o exercício, seguindo a modelação do diagrama da realização do exercício.

Se o aluno acertar, é incrementado o conhecimento e pontuação e, se a dificuldade for máxima, é recomendado que este passe à lição seguinte. Caso o aluno aceite é incrementado o NumLicao e termina o processo, caso recuse são apresentadas as opções de sair ou continuar. Se é escolhido sair, o processo é terminado, senão volta-se a apresentar um exercício com a mesma dificuldade. Se a dificuldade não for máxima, é incrementada a dificuldade e apresentado um exercício dessa dificuldade.

Caso a resposta do aluno ao exercício esteja incorreta, é decrementado o conhecimento e pontuação e verificado o número de respostas erradas. Se este número for baixo, é apresentado um outro exercício da mesma dificuldade; se for médio é recomendado ao aluno que repita a lição com outra explicação, caso este aceite é atualizado o NumExpl e apresentada a nova lição, caso recuse é lhe dada a escolha de sair ou continuar, se quiser sair termina o processo, se quiser continuar é apresentado outro exercício da mesma dificuldade; por último, se o número de respostas erradas for alto, é recomendado que o aluno passe à lição anterior, este pode aceitar e é decrementado o NumLicao e terminado o processo, ou então, pode recusar e, mais uma vez, são lhe dadas as opções de continuar ou sair, sendo que, se escolher a primeira é apresentado um novo exercício e, se escolher a segunda o processo termina.

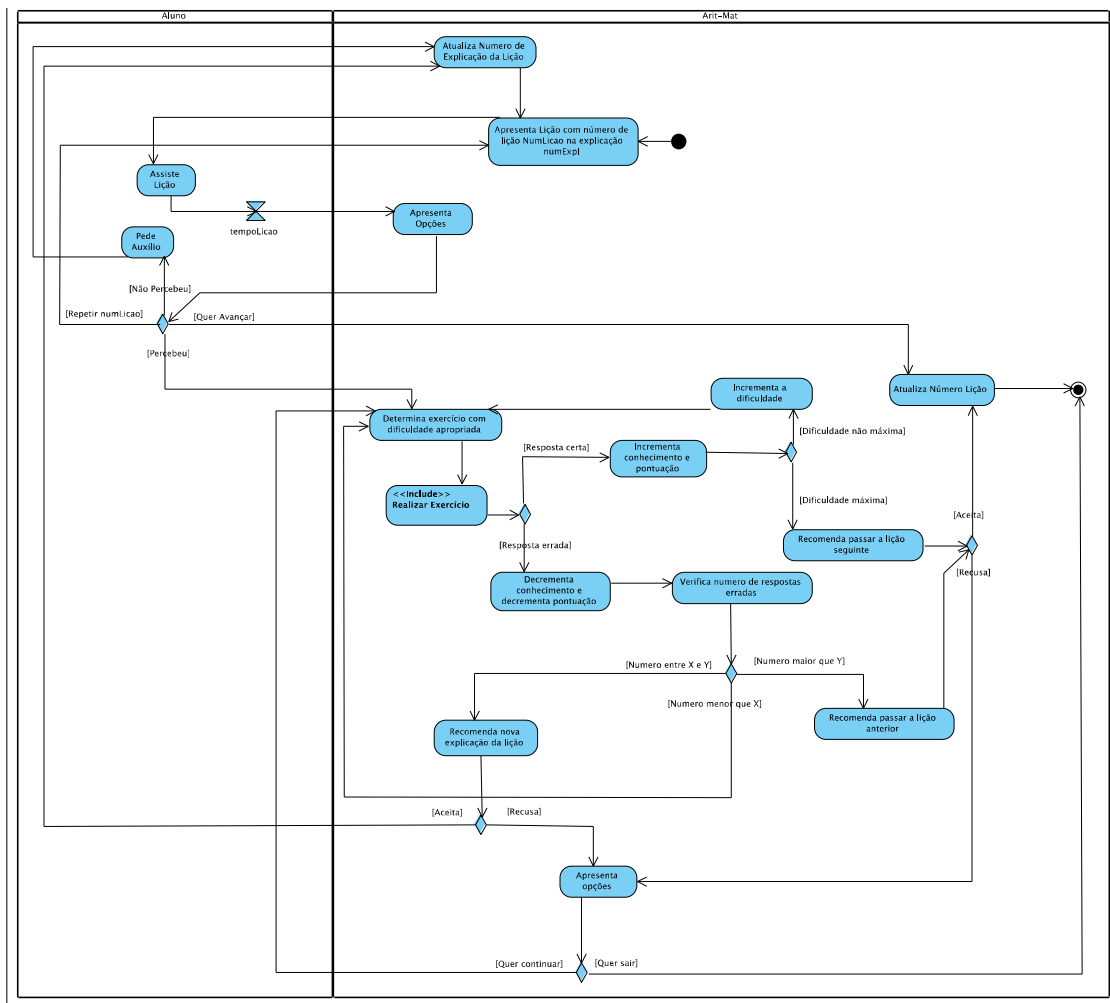


Figura 23- Diagrama de Atividade “Assistir Lição”

3.2. Modulação de Componentes

3.2.1. Diagrama de Classes

Para modular o vocabulário do sistema, decidindo sobre que abstrações estruturais fazem parte do sistema foi construído um diagrama de classes que apresenta como interagem as classes entre si e quais as suas responsabilidades na realização das operações solicitadas pelos atores. Na construção deste diagrama apenas foram considerados os elementos que se acharam de maior importância, ou seja, métodos como getters e setters, cuja simplicidade e necessidade é bastante óbvia, não foram abordados.

O diagrama pode ser consultado no Anexo III.

3.3. Modelação de Dados

3.3.1. Modelo Concetual

Tomando como base todos os requisitos expostos no capítulo 2.2 – Levantamento de Requisitos – foi construído um modelo concetual que procura, respeitando todas as restrições e requerimentos, iniciar a construção da base de dados que suportará a aplicação.

Cada requisito identificado constrói um elemento do modelo, assim sendo, de seguida apresenta-se, associado a cada requisito, os elementos que são construídos:

1. Registrar utilizadores mantendo um registo da sua informação;

Dado que o único tipo de utilização desta aplicação é para aprendizagem o seu único utilizador será um aluno, logo é justificada a existência de uma entidade Aluno e os seus atributos que permitem armazenar informação sobre o utilizador.

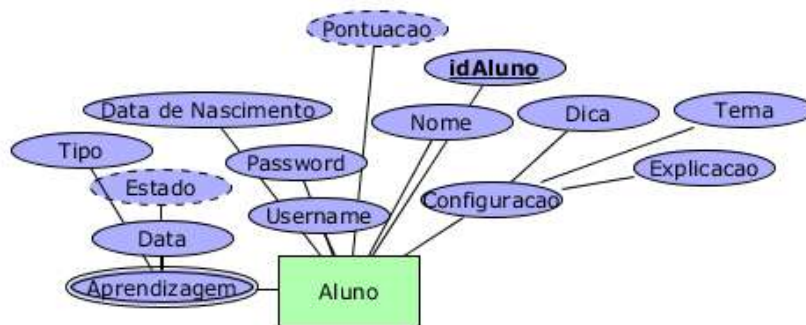


Figura 24 - Entidade Aluno e seus atributos

É ainda de notar a criação dos atributos compostos Aprendizagem e Configuracao que se justificam, respetivamente, pela necessidade de rastrear a evolução do aluno ao longo do processo de aprendizagem e pela necessidade de personalização do ambiente de trabalho para tornar o processo o mais aprazível possível.

2. Associar exercícios de acordo com o grau de dificuldade e tema e, conjugar os mesmos da melhor forma gerando testes de avaliação do conhecimento e progresso dos alunos; Com este requisito são identificadas as existências das entidades Exercicio e Teste bem como o seu relacionamento pois tem-se que os testes são conjugações de exercícios.

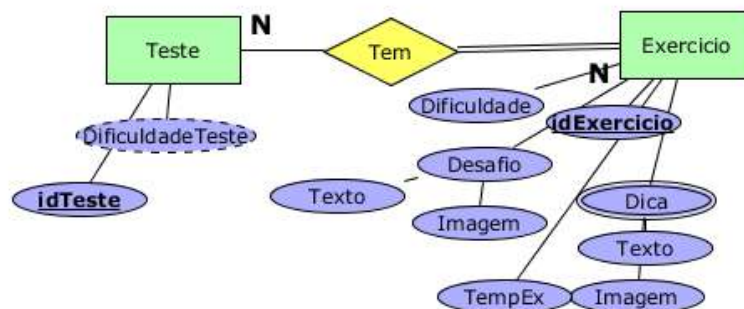


Figura 25 - Entidade Teste, Exercício e seu relacionamento

Um exercício possui ainda a pergunta que será representada pelo atributo Desafio e um tempo para a sua realização cujo valor será armazenado no atributo TempoEx.

3. Associar um grau de dificuldade a cada exercício para conseguir, por parte do sistema, fornecer exercícios direcionados ao grau de aprendizagem dos utilizadores/alunos, e, por parte dos utilizadores, analisar o desempenho devidamente associado à dificuldade da questão/exercício;

Um exercício possui então um atributo Dificuldade capaz de caracterizar a entidade Exercício que identifica a dificuldade do exercício em termos de matéria.

4. Os exercícios devem ser associados a respostas de escolha múltipla;

Por forma a suportar esta tarefa foi concetualizada a entidade Resposta que, por ser uma opção possui um texto dado pelo atributo Texto e, consoante o seu tipo (certa/errada), uma pontuação representada no atributo Pontuacao.

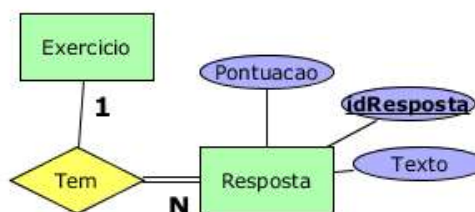


Figura 26 - Entidade Resposta e relacionamento com Exercício

5. Armazenar e disponibilizar informação educativa sobre os temas adição e subtração que será disponibilizada para uso pessoal dos utilizadores, tendo como objetivo acompanhar e facilitar a compreensão da matéria;

A partir deste pedido foi construída a entidade Licao e a ela associado um relacionamento com a entidade Aluno para representar a disponibilização do material e o processo de apresentação da lição a um aluno.

A entidade Licao possui um atributo Tempo que indica o tempo que demora a lição a ser apresentada e um atributo composto que será o conteúdo da explicação em formato

vídeo, imagem ou texto o que origina, para suportar todos os formatos três atributos (Video, Imagem e Texto).

A Licao é identificada pela combinação de um identificador de lição (NumLicao) e um identificador da explicação (NumExpl), o objetivo é conseguir suportar um modelo de aprendizagem interativo e portanto mais eficaz, mais à frente será evidenciado face ao modelo de aprendizagem quais as implicações no modelo concetual. Também o atributo derivado RespErradas e Data têm como base suporte à existência de um processo de aprendizagem estipulado e dinamizado.

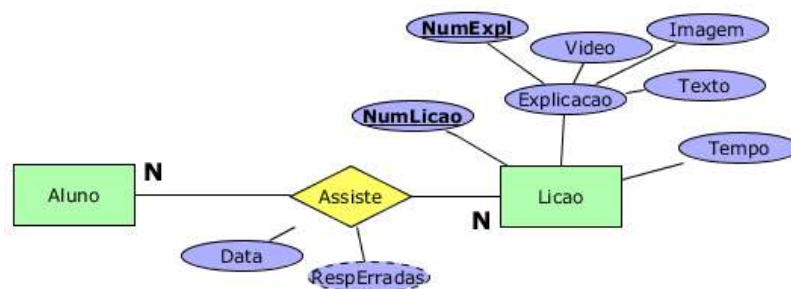


Figura 27 - Entidade Licao, atributos e seu relacionamento com Aluno

6. Associar a cada uma destas informações (lições) exercícios de consolidação que os alunos poderão realizar;

O relacionamento entre exercícios e lições é estabelecido pelo relacionamento existente no modelo concetual entre a entidade Exercício e Licao.

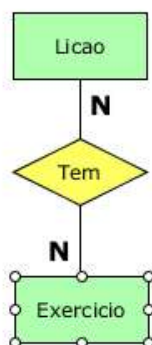


Figura 28 - Entidade Licao e Exercício e seu relacionamento

7. Permitir a realização de exercícios em testes e lições por parte dos alunos;
- Com o objetivo de satisfazer estas duas condições criaram-se os relacionamentos ternários entre as entidades Aluno e Teste e Aluno e Licao que, representam a realização de testes e lições por parte dos alunos.

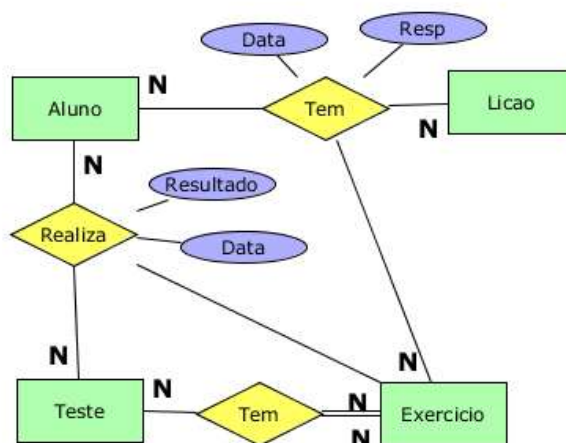


Figura 29 - Relacionamentos ternários

Apesar de já existir um relacionamento entre Aluno e Licao é de grande importância manter este relacionamento para conseguir identificar quais os exercícios realizados numa dada lição por um aluno, já o relacionamento binário indica que lições já foram visualizadas quebrando a dependência da existência de exercícios na lição, pois um aluno deverá poder assistir às lições sem realizar exercícios se o pretender. Com este relacionamento é possível, numa consulta, saber quantas respostas erradas (observando o atributo Resp do relacionamento ternário Aluno-Licao-Exercicio) e com isso preencher o atributo do relacionamento Aluno-Licao referido na análise ao requisito 4, este atributo só existe para um acesso mais imediato aquando o cálculo de estatísticas ou melhorias

Em relação ao relacionamento ternário de Aluno, Exercicio e Teste, este permite saber qual o resultado de um exercício num dado teste.

8. Associar os testes a níveis, disponibilizados para os alunos por forma a avaliar o seu conhecimento de um determinado assunto;

Este requisito é validado muito facilmente pela existência do atributo derivado DificuldadeTeste calculado à custa do atributo Dificuldade da entidade Exercicio, isto representa que a dificuldade de um teste é dependente da dificuldade dos exercícios que o compõem.

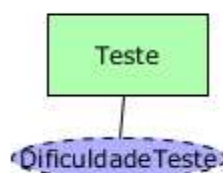


Figura 30 - Atributo DificuldadeTeste de entidade Teste

9. Armazenar um histórico de dados associado a entidades particulares (utilizadores) com o intuito de melhor adaptar o produto a cada utilizador;

O atributo composto Configuracao como já foi mencionado na comprovação do requisito 1, é utilizado para armazenar todas as informações de personalização do ambiente de trabalho.



Figura 31 - Entidade Aluno e seu atributo Configuracao

Assim sendo, o atributo Dica identifica se o aluno quer ou não receber dicas na eventual realização de exercícios, o atributo Explicacao identifica se o aluno permite que novas explicações sobre a mesma matéria lhe sejam apresentadas e finalmente o atributo Tema que indica qual o tema do ambiente escolhido pelo aluno.

10. Capacitar a personalização do programa face a determinado utilizador, para que melhor se ajuste às suas necessidades e gostos;

Como já foi mencionado no item anterior existe o atributo composto Configuracao para armazenamento de todas as informações sobre a configuração do ambiente de trabalho escolhido pelo aluno.

11. Disponibilizar um modelo de aprendizagem individual.

Este requisito implicou a construção de um processo ou método de ensino que teve implicações em todo o modelo concetual, quanto ao método de ensino este será detalhadamente explicado numa fase mais avançada do relatório onde é especificado o processo completo com o auxílio de um diagrama de atividades (ver capítulo 3.1.4). Nesta parte da análise vamos apenas mencionar quais as implicações referidas anteriormente.

Por forma a suportar um modelo de ensino encontrou-se a necessidade de separar, tanto as lições como os exercícios, em tipo de matéria e, tipificando o sistema já pensando numa possível expansão para outros conceitos e temas foi criada a entidade Tipo com um atributo identificador idTipo e um atributo Area que possui o texto correspondente à área da matéria. A necessidade surge quando o aluno quer aprender um determinado conceito ou tema os exercícios e lições apresentadas deverão incidir neste tipo.

Esta entidade Tipo é então relacionada com as entidades Exercício e Licao sendo que ambas possuem obrigatoriamente um tipo.

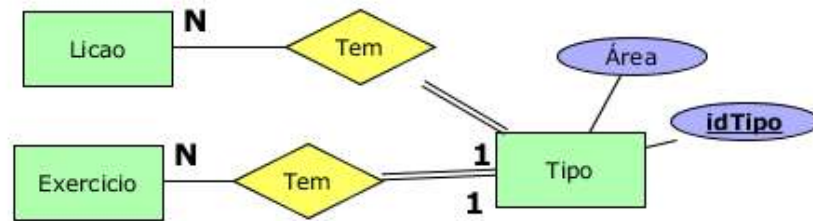


Figura 32 - Entidade Tipo

Além desta entidade foram estipulados os atributos Aprendizagem (Aluno), Dica (Exercício), Tempo (Licao), TempoEX (Exercício) entre outros para armazenar informações que representam quantitativamente o grau de aprendizagem (Aprendizagem), ajudas a exercícios para induzir a uma forma de pensar correta (Dica), informações sobre o tempo de exposição da matéria para dinamizar o processo tornando-o proativo e não estático.

É de notar que o identificador de uma lição (numLicao e numExpl) – ver análise ao requisito 5 – partilha do mesmo objetivo que os atributos mencionados no parágrafo anterior e serve para conseguir situar em que fase do processo o aluno está, ou seja, sabendo que as lições se estruturam numa ordem, com uma consulta o sistema é capaz de reconhecer em que lição o aluno vai e quais as explicações que este visualizou sobre a mesma. Assim, se necessário, o sistema consegue sugerir novas explicações caso o aluno falhe a um número elevado de exercícios ou ainda de sugerir recuar ou avançar lições conforme o seu desempenho nos exercícios das lições.

12. Analisar o histórico de cada utilizador e, consoante o mesmo, sugerir diversas lições propícias à consolidação e melhoria das áreas de aprendizagem menos bem “pontuadas” em função do histórico analisado;

Este requisito é inerente ao anterior e por isso é cumprido pela existência de um modelo de aprendizagem/ensino.

13. Agregar o histórico de todos os utilizadores, constituindo estatísticas sobre a matéria e perceber quais os temas mais problemáticos, quais as questões com maior número de falhas, etc.;

Para conseguir cumprir este requerimento apenas é necessário consultar as tabelas representativas da realização de exercícios e com a resposta construir uma estatística, pode ainda dividir-se a estatística por tema e data se pretendido.

14. Registrar qual o estado de aprendizagem de um aluno por forma a identificar o seu progresso;

Mais uma vez este requisito é complementar do requisito do modelo de aprendizagem/ensino – requisito 11 – e, por esta razão, o sistema está apto a corresponder ao pedido.

15. Gestão de diferentes elementos de dados (texto, imagem, som, vídeo, etc.) associados às funcionalidades do tutor, por forma a gerar uma experiência de aprendizagem mais adequada e acessível à perceção;

Em função deste requisito existem nas explicações da lição e na apresentação do conteúdo e dicas do exercício os atributos correspondentes a vídeo, texto e imagem.

16. Permitir a gestão de lições (introdução e edição) por um utilizador administrador de modo a que o sistema consiga expandir o número de lições existentes.

Este último requisito origina a entidade Administrador e respetivos atributos bem como o relacionamento de gestão entre esta e as entidades Licao e Exercicio dado que o administrador gere lições e exercícios.

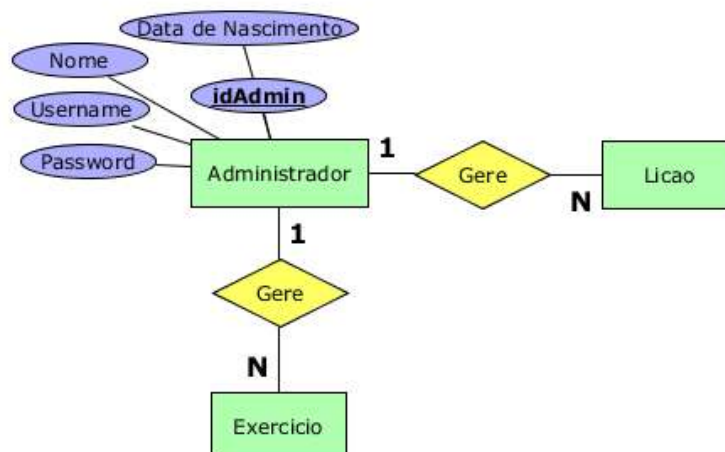


Figura 33 - Entidade Administrador

3.3.2. Modelo Lógico

Tendo como base o modelo concetual, apresentado no subcapítulo anterior, o modelo lógico surge do refinamento e mapeamento do modelo concetual, isto é, na normalização dos dados. A normalização garante que as tabelas derivadas do modelo de dados não evidenciam redundância de dados, permitindo um armazenamento consistente, um acesso eficiente aos dados e diminuindo a probabilidade de os dados se tornarem inconsistentes.

3.3.2.1. Obtenção de Tabelas

Todas as entidades apresentadas no modelo concetual originam uma tabela no modelo lógico, assim, são geradas sete tabelas respeitantes a Aluno, Lição, Exercício, Teste, Tipo, Resposta e Administrador.

Por cada relacionamento de n para m é, também, gerada uma tabela. Assim, dos relacionamentos, identificados como tal, surgem as tabelas Aluno_Licao, Teste_Exercicio, Licao_Exercicio.

Em relação aos relacionamentos ternários e, uma vez que, este tipo de relacionamento gera uma tabela, temos as tabelas Aluno_Exercicio_Licao, Aluno_Testes_Exercicio.

Dado que, os atributos multivalorados, dão origem a uma tabela, surgem as tabelas Aprendizagem e Dica.

Dado isto, existem, para esta aplicação, catorze tabelas, abaixo detalhadas.

Aluno (idAluno, Nome, Username, Password, DataNasc, Dica, Tema, Explicacao, Pontuacao)

Chave Primária idAluno

Chave Estrangeira não existe

Teste (idTeste, DificuldadeTeste)

Chave Primária idTeste

Chave Estrangeira não existe

Exercicio (idExercicio, Tipo, Administrador, Texto, Dificuldade, Imagem, TempoEx)

Chave Primária idExercicio

Chave Estrangeira Tipo **references** Tipo(idTipo)

Chave Estrangeira Administrador **references** Administrador(idAdministrador)

Resposta (idResposta, Exercicio, Pontuacao, Texto)

Chave Primária idResposta

Chave Estrangeira Exercicio **references** Exercicio(idExercicio)

Tipo (idTipo, Area)

Chave Primária idTipo

Chave Estrangeira não existe

Licao (idLicao, NumExpl, Tipo, Administrador, Texto, Video, Imagem, TempoLicao)

Chave Primária idLicao

Chave Primária Tipo, Administrador

Chave Estrangeira Tipo **references** Tipo(idTipo)

Chave Estrangeira Administrador **references** Administrador(idAdministrador)

Administrador (idAdministrador, DataNasc, Username, Password, Nome)

Chave Primária idAdministrador

Chave Estrangeira não existe

Aluno_Licao (Aluno, Licao, Explicacao, Data, RespErradas)

Chave Primária Aluno, Licao, Explicacao

Chave Estrangeira Aluno **references** Aluno(idAluno)

Chave Estrangeira Licao **references** Licao(idLicao)

Chave Estrangeira Explicacao **references** Licao(NumExpl)

Teste_Exercicio (Teste, Exercicio)

Chave Primária Teste, Exercicio

Chave Estrangeira Teste **references** Teste(idTeste)

Chave Estrangeira Exercicio **references** Exercicio(idExercicio)

Licao_Exercicio (Licao, Exercicio)

Chave Primária Licao, Exercicio

Chave Estrangeira Licao **references** Licao(idLicao)

Chave Estrangeira Exercicio **references** Exercicio(idExercicio)

Aluno_Exercicio_Licao (Aluno, Licao, Explicacao, Exercicio, Data, Resposta)

Chave Primária Aluno, Licao, Explicacao, Exercicio

Chave Estrangeira Aluno **references** Aluno(idAluno)

Chave Estrangeira Licao **references** Licao(idLicao)

Chave Estrangeira Explicacao **references** Licao(NumExpl)

Chave Estrangeira Exercicio **references** Exercicio(IdExercicio)

Aluno_Testes_Exercicio (Aluno, Teste, Exercicio, Data, Nota)

Chave Primária Aluno, Teste, Exercicio

Chave Estrangeira Aluno **references** Aluno(idAluno)

Chave Estrangeira Teste **references** Teste(idTeste)

Chave Estrangeira Exercicio **references** Exercicio(IdExercicio)

Aprendizagem (idAprendizagem, Aluno, Data, Estado, Tipo)

Chave Primária idAprendizagem, Aluno

Chave Estrangeira Aluno **references** Aluno(idAluno)

Dica (idDica, Exercicio, Texto, Imagem)

Chave Primária idDica, Exercicio

Chave Estrangeira Exercicio **references** Exercicio(idExercicio)

3.3.2.2. Chaves Estrangeiras

Em relação ao aparecimento das chaves estrangeiras, resultantes da tradução do modelo concetual para modelo lógico, conforme apresentado na subsecção acima, segue a explicação da presença das mesmas nas diferentes tabelas.

Quanto às chaves estrangeiras que surgem dos relacionamentos de 1:N, a entidade do lado 1 do relacionamento é designada como entidade pai e a entidade do lado N é designada como entidade filha. Dado isto, para representar este relacionamento, é colocada na tabela filho uma cópia da chave primária da entidade pai, representando uma chave estrangeira. Seguem, as chaves estrangeiras deste tipo de relacionamento, bem como a tabela a que pertencem.

Exercicio Nesta tabela, a chave estrangeira Administrador, aparece do relacionamento Administrador(1):Exercicio(N), logo, é colocada na tabela filho – Exercicio – uma cópia da chave primária da entidade pai – Administrador.

A chave estrangeira Tipo, aparece do relacionamento Tipo(1):Exercicio(N), logo, é colocada na tabela filho – Exercicio – uma cópia da chave primária da entidade pai – Tipo.

Resposta Nesta tabela, a chave estrangeira Exercicio, aparece do relacionamento Exercicio(1):Resposta(N), logo, é colocada na tabela filho – Resposta – uma cópia da chave primária da entidade pai – Exercicio.

Licao Nesta tabela, a chave estrangeira Tipo, aparece do relacionamento Tipo(1):Licao(N), logo, é colocada na tabela filho – Licao – uma cópia da chave primária da entidade pai – Tipo.

A chave estrangeira Administrador, aparece do relacionamento Administrador(1):Licao(N), logo, é colocada na tabela filho – Licao – uma cópia da chave primária da entidade pai – Administrador.

Quanto aos relacionamentos de N:M, para além de ser gerada uma tabela que representa o relacionamento, são incluídos todos os atributos que fazem parte do relacionamento, bem como as chaves primárias das entidades que compõem o relacionamento, representando as chaves estrangeiras. Estas chaves estrangeiras podem compor, em conjunto ou não, a chave primária da nova tabela. Seguem as chaves estrangeiras deste tipo de relacionamento, bem como a tabela a que pertencem.

Aluno_Licao Nesta tabela as chaves estrangeiras são, respetivamente, Aluno, Licao e Explicacao, que são chaves primárias nas entidades Aluno e Licao e, que por isso, constituem chaves estrangeiras na nova tabela.

Teste_Exercicio Nesta tabela as chaves estrangeiras são, respetivamente, Teste e Exercicio, que são chaves primárias nas entidades Teste e Exercicio e, que por isso, constituem chaves estrangeiras na nova tabela.

Licao_Exercicio Nesta tabela as chaves estrangeiras são, respetivamente, Licao e Exercicio, que são chaves primárias nas entidades Licao e Exercicio e, que por isso, constituem chaves estrangeiras na nova tabela.

Aluno_Exercicio_Licao Nesta tabela as chaves estrangeiras são, respetivamente, Aluno, Licao, Explicacao e Exercicio, que são chaves primárias nas entidades Aluno, Licao e Exercicio e, que por isso, constituem chaves estrangeiras na nova tabela.

Aluno_Testes_Exercicio Nesta tabela as chaves estrangeiras são, respetivamente, Aluno, Teste e Exercicio, que são chaves primárias nas entidades Aluno, Teste e Exercicio e, que por isso, constituem chaves estrangeiras na nova tabela.

Em relação aos atributos multivalorados, para cada um existente numa entidade, para além de ser gerada uma nova tabela a representar o atributo multivalorado é incluída nesta tabela a chave primária da entidade a que o valor multivalorado pertence, representando assim uma chave estrangeira. Seguem as chaves estrangeiras dos atributos multivalorados, bem como a tabela a que pertencem.

Aprendizagem Nesta tabela a chave estrangeira é Aluno, que é chave primária na entidade Aluno, isto é, entidade a que pertencia o atributo multivalorado e, que por isso, constitui chave estrangeira na nova tabela.

Dica Nesta tabela a chave estrangeira é Exercicio, que é chave primária na entidade Exercicio, isto é, entidade a que pertencia o atributo multivalorado e, que por isso, constitui chave estrangeira na nova tabela.

3.3.2.3. Primeira Forma Normal

Todas as tabelas apresentadas cumprem esta primeira forma, uma vez que, todas as chaves primárias possuem valores únicos, atómicos. Assim, em cada tabela a interseção de cada coluna e linha contém um e apenas um só valor.

3.3.2.4. Segunda Forma Normal

Para a verificação da segunda forma normal, foram analisadas apenas as tabelas cuja chave primária é composta. Assim, tendo em conta este requisito, é verificado que todos os seus atributos dependem funcionalmente de toda a chave e não apenas de um subconjunto da mesma.

3.3.2.5. Terceira Forma Normal

Nesta forma removem-se dependências transitivas mas, analisando as tabelas geradas, verifica-se que não existe nenhum atributo que seja transitivamente dependente de um outro atributo, assim sendo, o modelo encontra-se validado pela 3FN.

3.3.2.6. Restrições de Integridade

Para que uma base de dados se encontre num estado de integridade, esta deve seguir restrições de integridade de maneira a garantir informação consistente, definir validade dos dados. As restrições de integridade correspondem, nomeadamente, a integridade de domínio, entidade, referencial e organização do problema.

Integridade de Domínio regras que se aplicam aos atributos de uma dada tabela, definindo o domínio de cada atributo, isto é, que atributos não podem ter valores nulos, que existem atributos cujos valores devem estar dentro de um intervalo e, ainda, todos os atributos devem ter valores mediante o domínio definido. Por exemplo, os atributos definidos como não nulos, não devem aceitar valores nulos, um atributo que seja uma data de nascimento tem que ser inferior ou igual ao ano atual, e superior a um dado ano. Um atributo que tenha como domínio um tipo int, não deve aceitar valores nvarchar. Assim, sempre que um dado é inserido, alterado ou apagado o sistema de gestão de base de dados vai verificar se as regras definidas são respeitadas.

Uma vez que, no modelo lógico os atributos que não podem assumir valores nulos são identificados com um losango preenchido a cor azul, pode visualizar-se, no no Anexo V, em todas as tabelas, que atributos têm obrigatoriedade de preenchimento, de maneira a que os dados tenham consistência e a informação não seja incompleta.

Integridade de Entidade regras que se aplicam aos atributos identificados como chave primária, bem como às entidades. Isto é, não existem entidades repetidas, sendo cada

uma identificada unicamente por um nome. Além disto, ao ser declarado, numa tabela, um atributo como chave primária, o sistema de gestão de base de dados não permite que a tabela tenha dois tuplos com o mesmo valor nesse atributo, bem como valores nulos.

Integridade Referencial restrição de integridade que relaciona duas tabelas. Um atributo, que numa tabela, seja chave estrangeira tem, obrigatoriamente, que ser chave primária numa outra tabela. Isto é, cada valor do atributo chave estrangeira tem que existir, primeiro, no atributo da tabela onde este é chave primária. Para que o modelo exposto cumpra esta restrição, as tabelas que correspondem às entidades “filho” possuem uma chave estrangeira, mapeando esta, sempre, a chave primária das tabelas que correspondem às entidades “pai”.

Integridade de Organização do Problema para cumprir esta última regra de integridade foi levantado no modelo concetual os requisitos e restrições adequados ao contexto do problema e, a partir daí, gerado o modelo.

4. Conclusões e Trabalho Futuro

Durante o desenvolvimento da segunda fase deste trabalho, houve necessidade de redefinir alguns aspetos anteriormente pensados que, face ao curto espaço de tempo e aos objetivos principais, se acharam secundários e não essenciais. Tem-se por exemplo, o caso da existência de um perfil de professor no sistema, dado que a própria aplicação já se trata de um tutor considerou-se acessório este item.

É de notar que na análise de viabilidade, dada a baixa faixa etária do público-alvo e a total dependência do sucesso da aplicação nos mesmos, existe um risco na continuação do projeto, mas, dado que não existem quaisquer custos associados, foi decidido dar seguimento ao projeto. A documentação foi cingida aos elementos mais complexos e não triviais, isto é, apesar de estar representada e especificada a maior parte da arquitetura esperada para o sistema, alguns elementos comuns à generalidade das aplicações e bastante simples não foram abordados.

A maior dificuldade a apontar é relativa às decisões da arquitetura do sistema. Dado que estas são de extrema importância, foi necessário uma concentração e discussão extensa sobre quais as melhores soluções aplicáveis.

Sobre o trabalho futuro, a próxima fase que tratará da implementação e construção da aplicação pode concordar ou quebrar as decisões tomadas até este ponto. Por este motivo, espera-se uma inicialização imediata do processo, por forma a maximizar a produtividade tendo em conta o tempo disponível.

Bibliografia

[01] Ian Sommerville, "Software Engineering", 9ª edição, 2011

[02] Database Systems: A Practical Approach To Design, Implementation, and Management, 5a edição, 2009, Thomas Connolly, Carolyn Begg

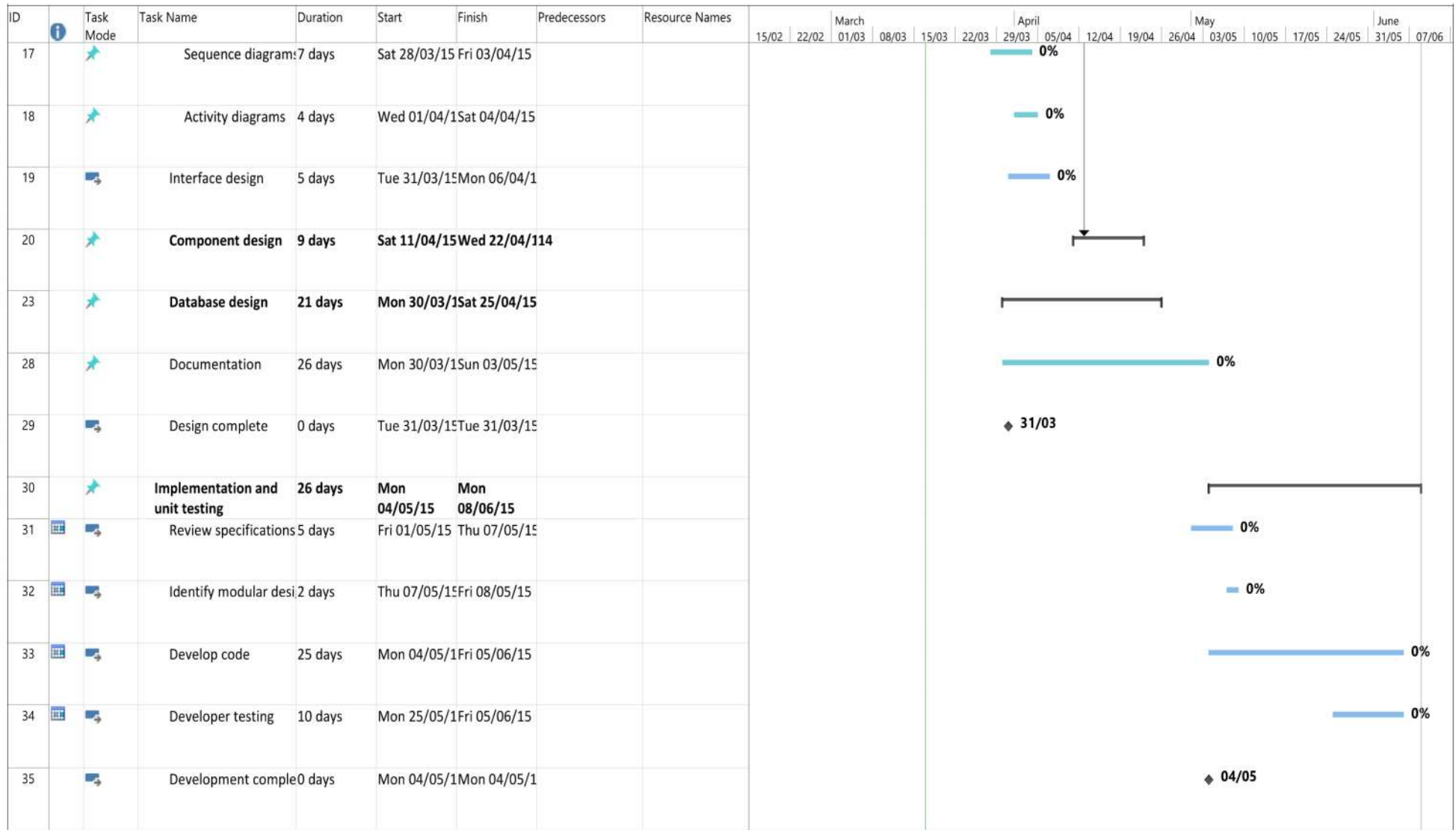
Referências WWW

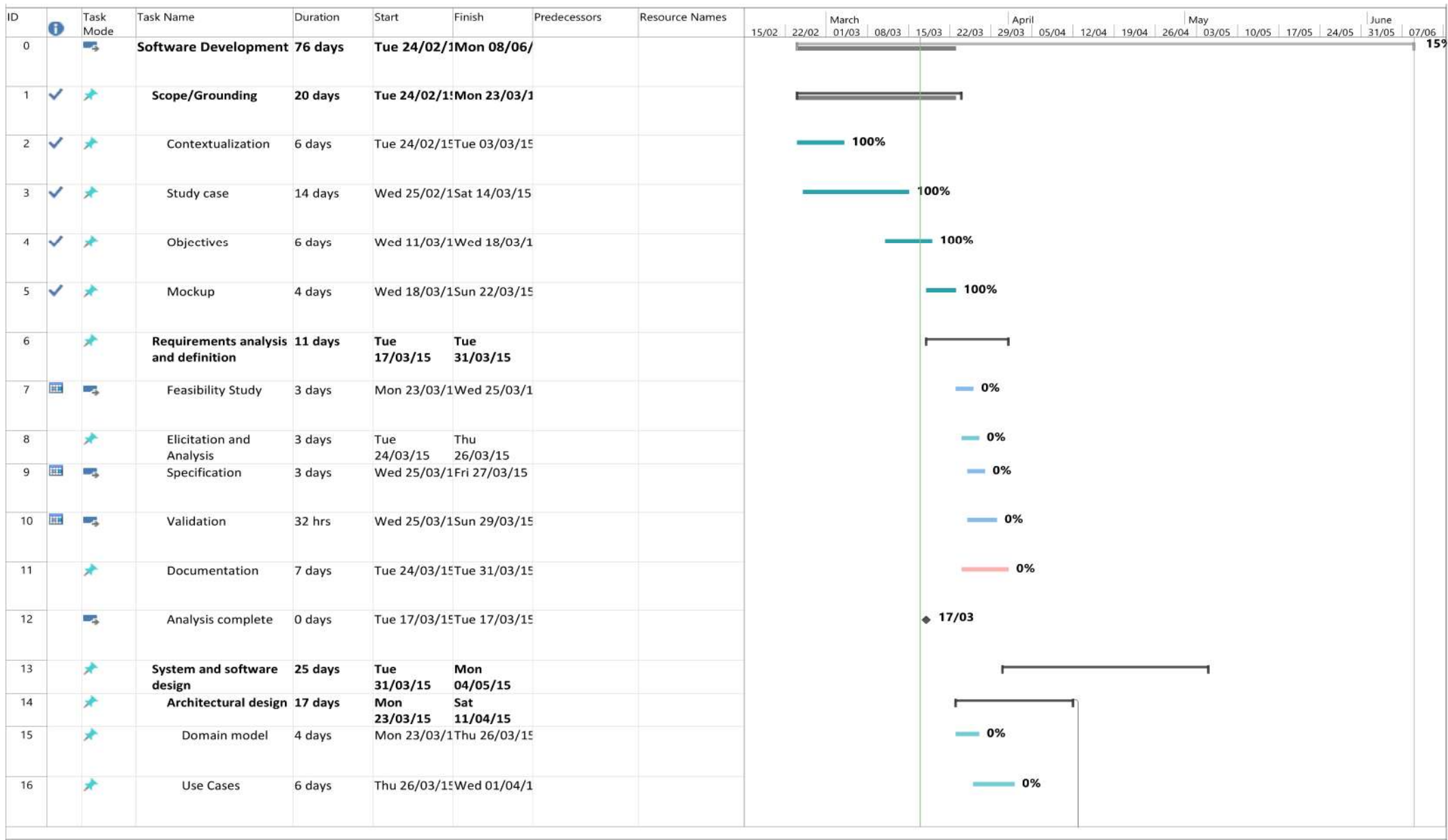
Lista de Siglas e Acrónimos

Anexos

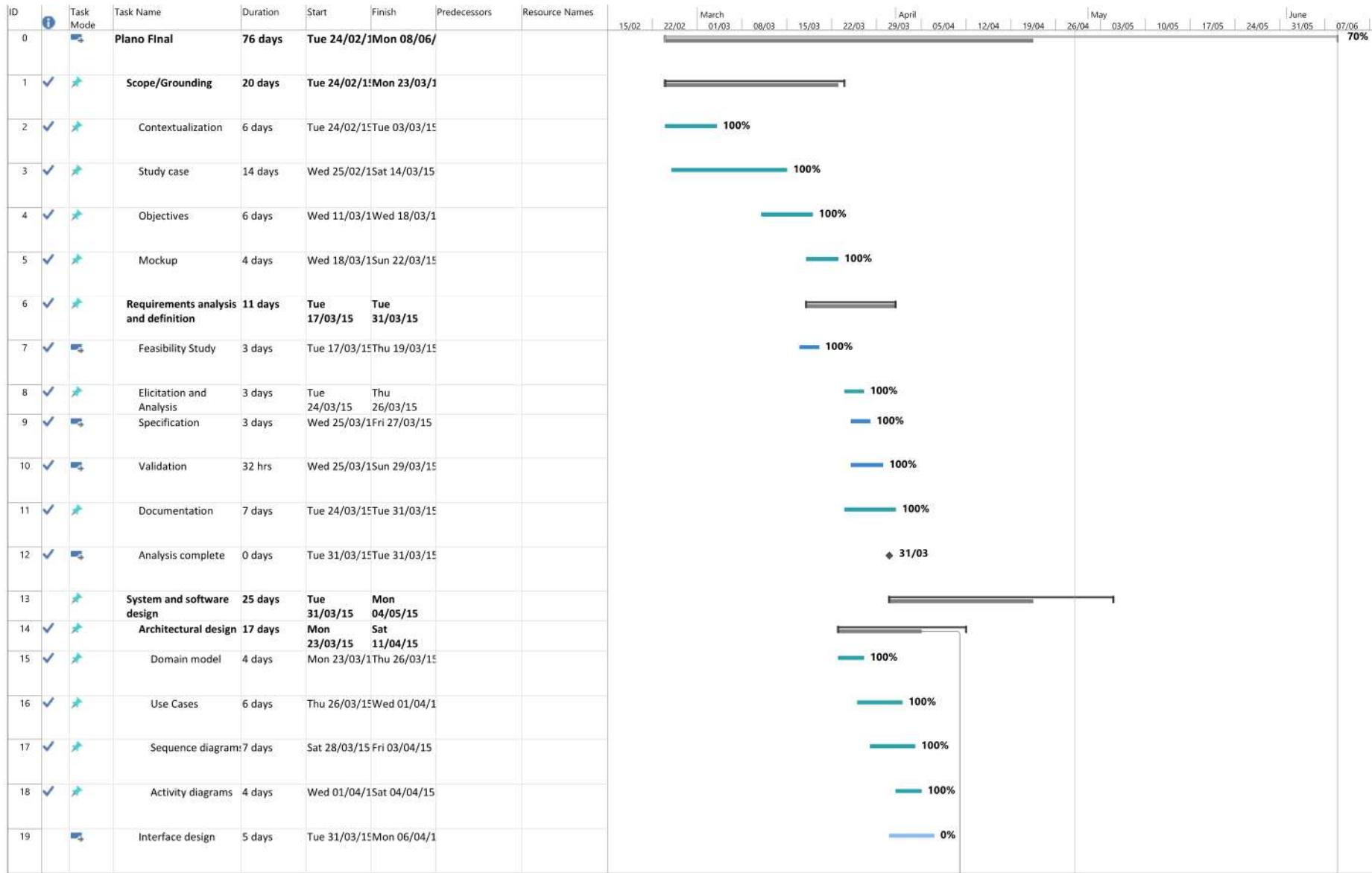
I. Diagrama de Gantt

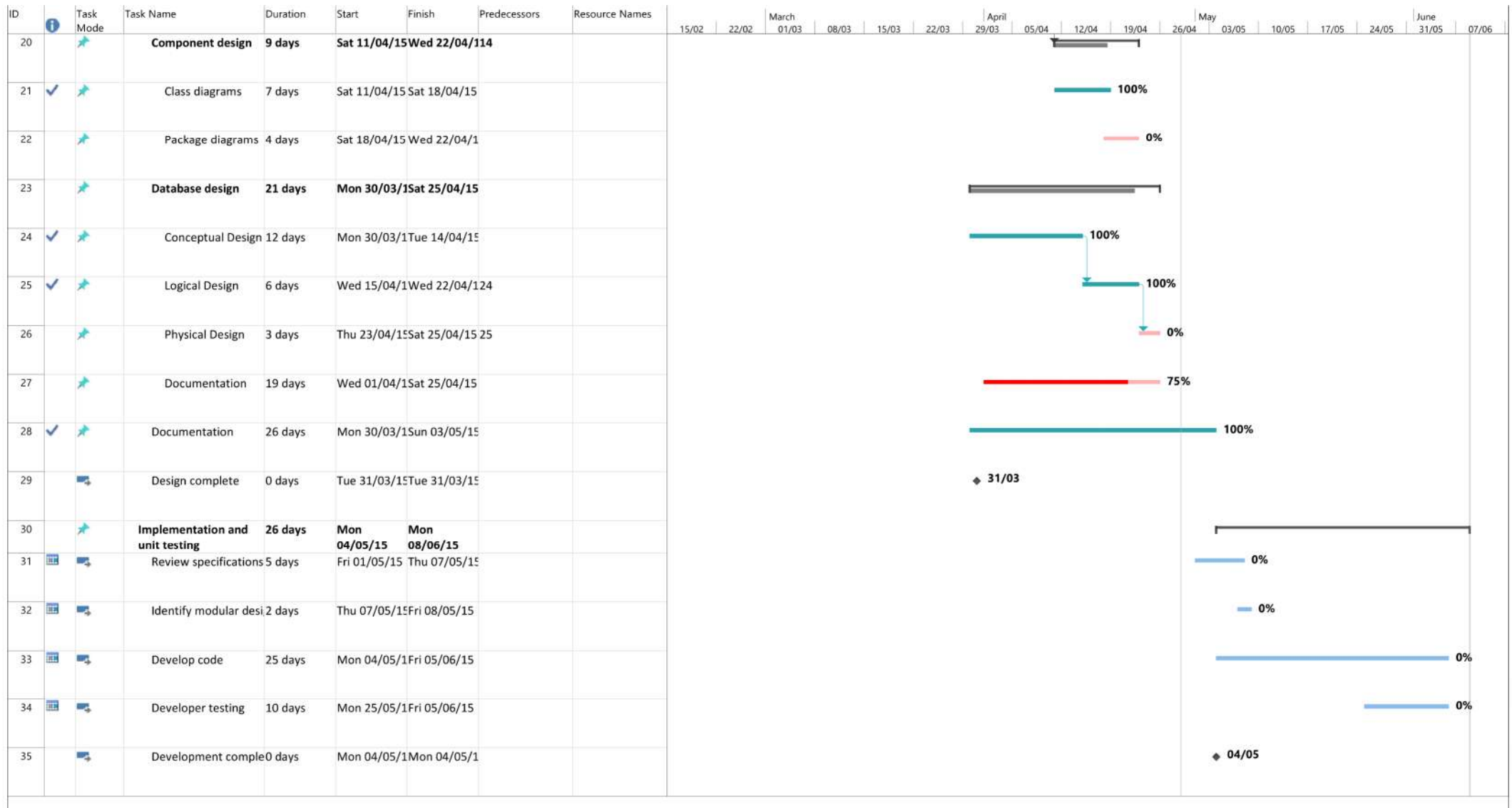
I.I.1. Fase 1



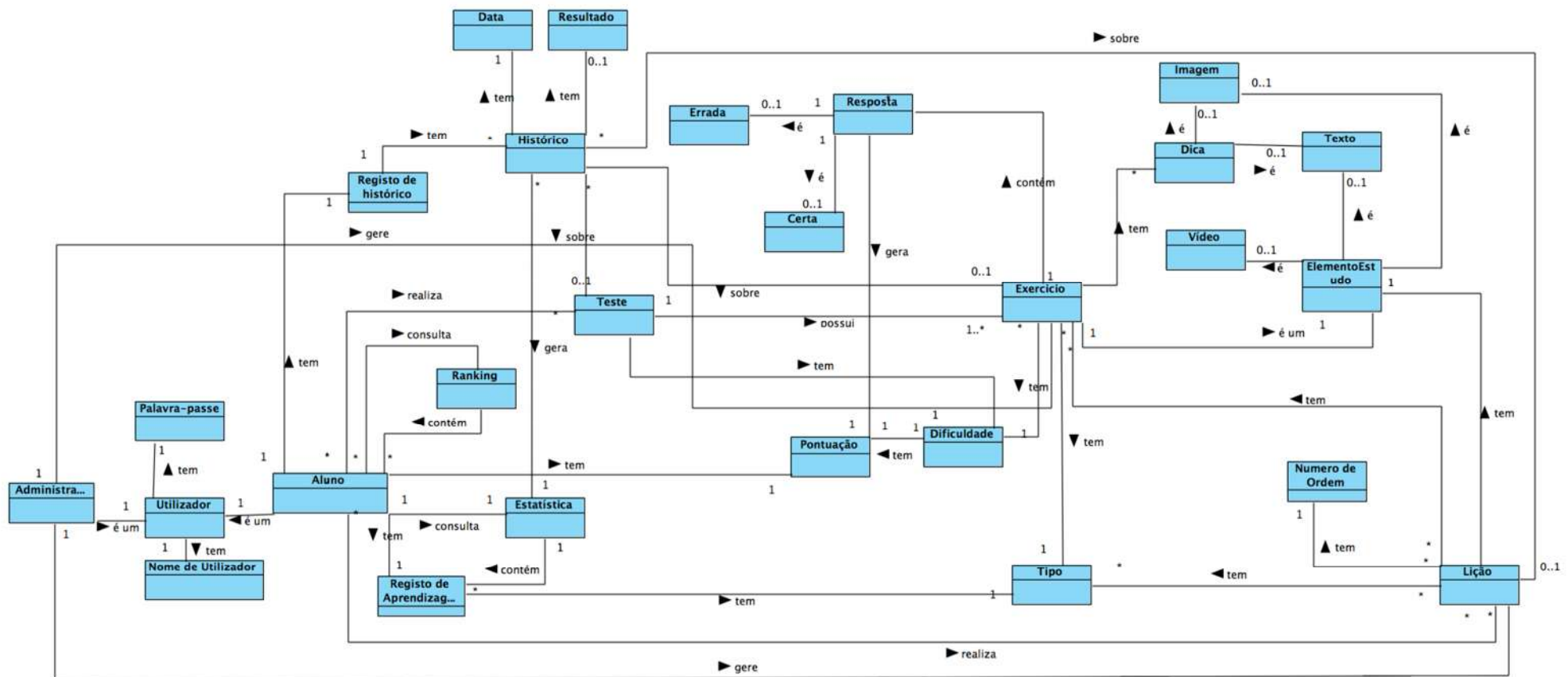


I.I.2. Fase 2

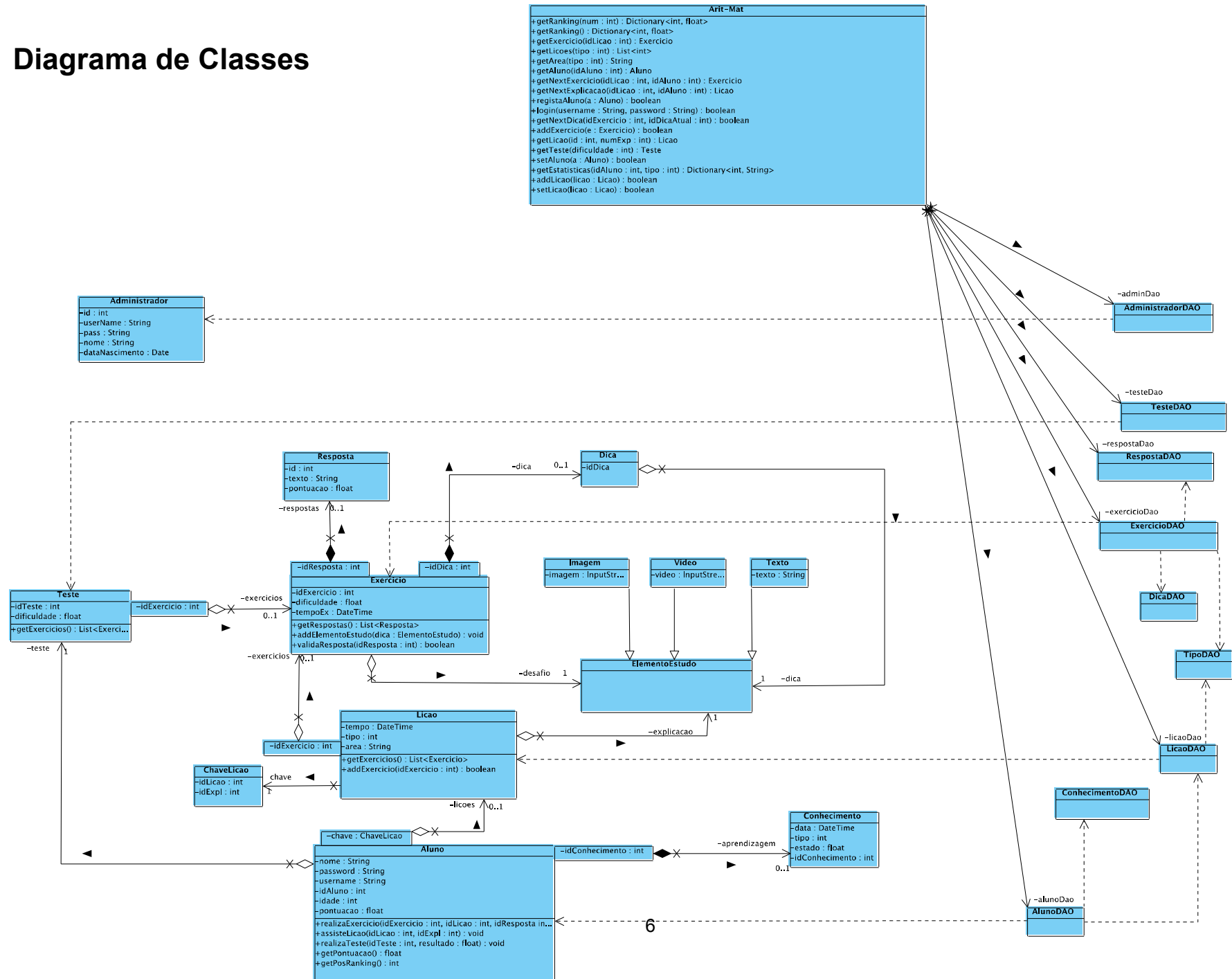




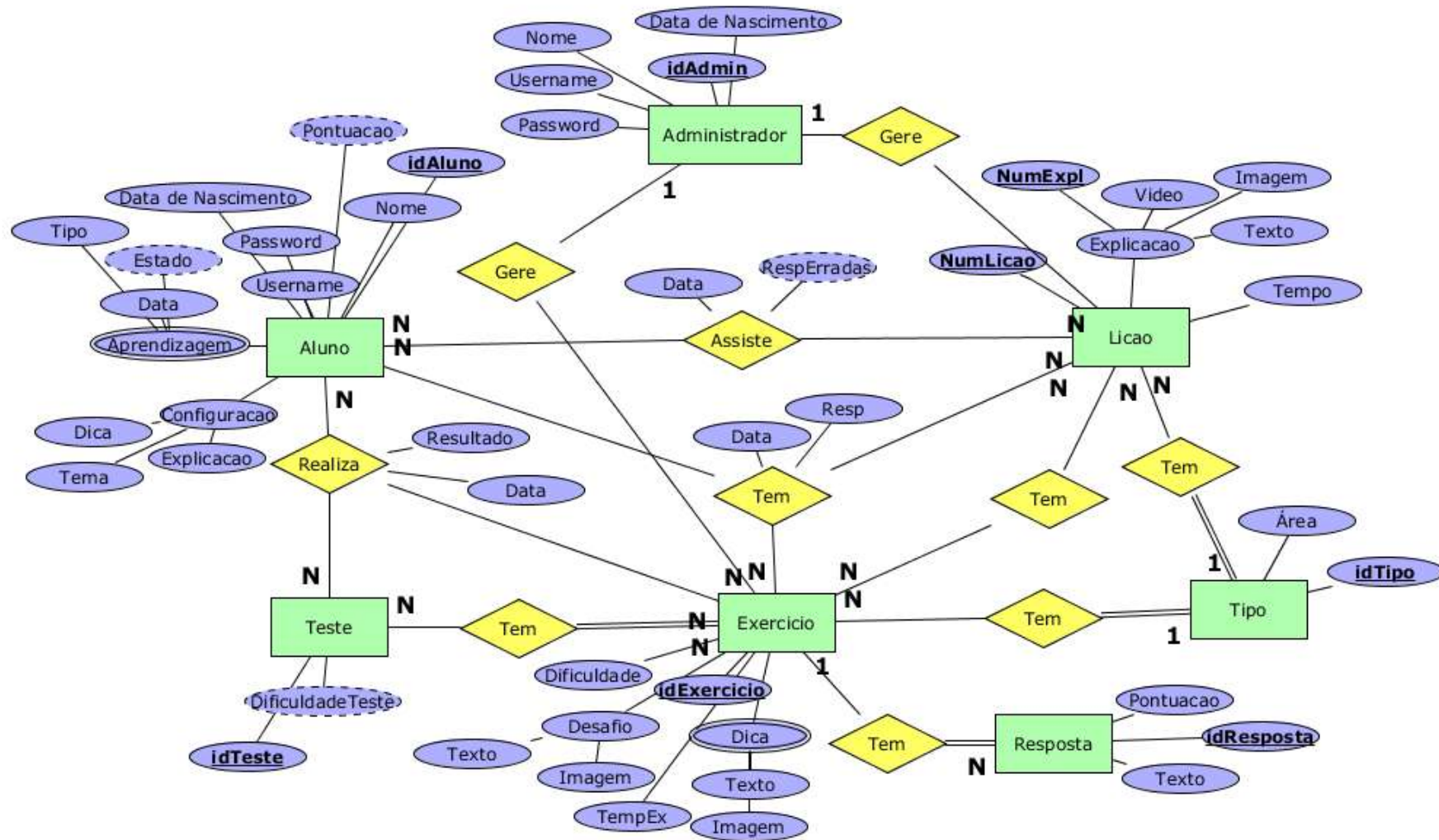
II. Modelo de Domínio



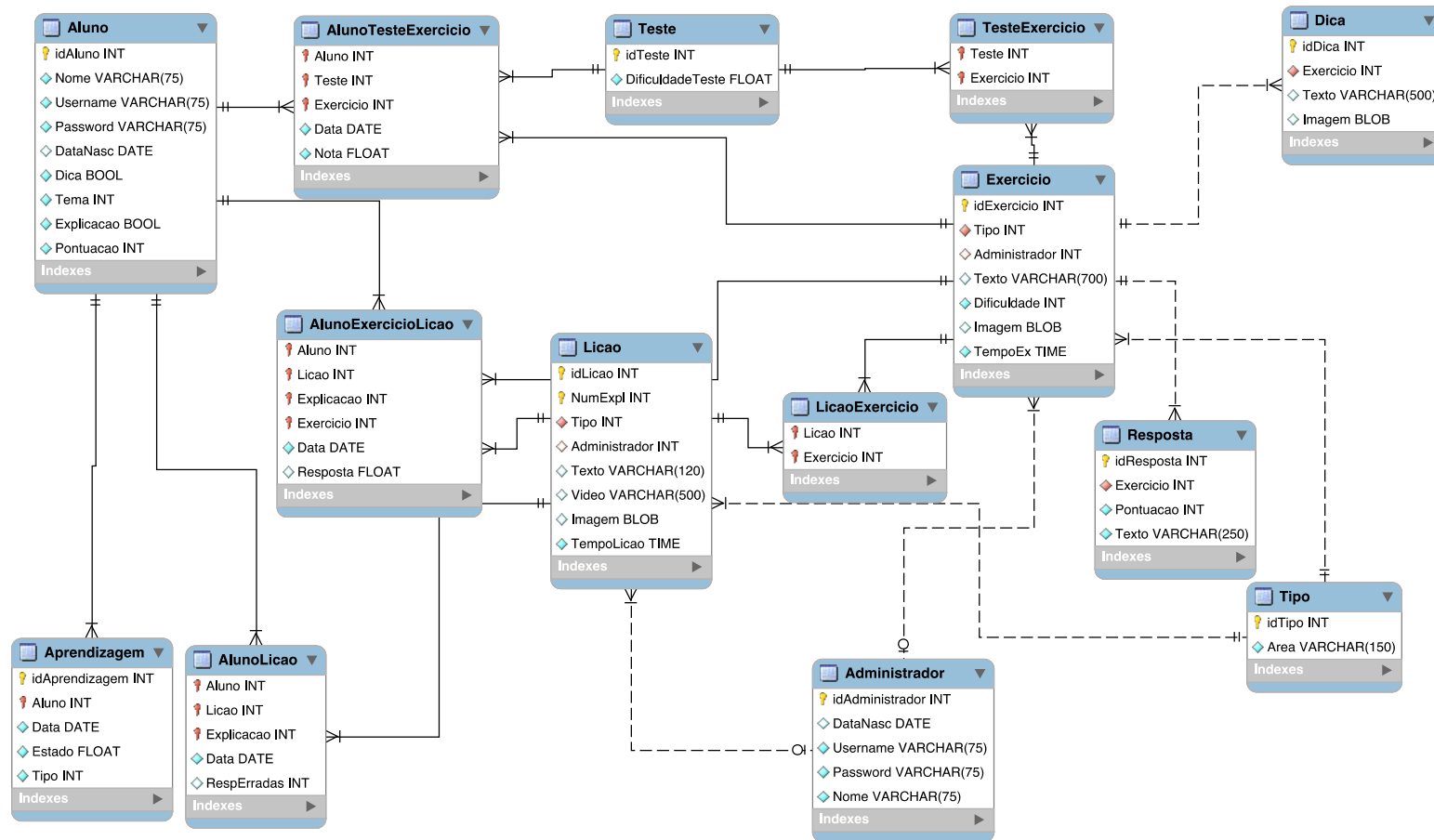
III. Diagrama de Classes



IV. Modelo Concetual



V. Modelo Lógico



VI. Dicionário de Dados – Entidades

Entidade	Descrição	Alias	Ocorrência
Aluno	Termo usado para se referir ao estudante que procura aprender os conceitos envolvidos nos temas de soma e subtração.		Aparece quando um estudante pretende ser ensinado sobre os temas de soma e subtração.
Licao	Termo referente a uma disponibilização de material de estudo com o intuito de ensinar determinados conceitos particulares existentes nos temas de soma e subtração.		Uma das entidades mais importantes dado que propiciam ao sucesso de uma boa aprendizagem, objetivo do produto. Surge quando um aluno, para se informar sobre os conceitos, assiste a uma exposição de matéria.
Teste	Conjunção de exercícios de forma ordenada e associado a uma pontuação que são disponibilizados aos alunos para avaliação dos seus conhecimentos.		Conceito utilizado para referência de um conjunto de exercícios apresentados e associados a uma pontuação concreta que, quando realizados por um aluno, permitem esclarecer se os conhecimentos foram ou não adquiridos.
Exercicio	Elemento singular representativo de um determinado desafio sobre a matéria de soma e subtração que é utilizado pelos alunos tanto para praticar como para serem avaliados.		Os exercícios surgem associados a testes e lições, no primeiro caso para avaliação e no segundo para prática. Associados a um determinado grau de dificuldade são de extrema importância para o processo de consolidação de conhecimentos.
Resposta	Conceito usado para representar uma opção de resposta de um exercício.		Associadas aos exercícios, simbolizam opções que correspondem a uma resposta possível à questão expressa no exercício.
Tipo	Termo que se refere ao tema da matéria, tipificando e relacionando lições e exercícios com os seus respetivos tipos de matéria abordada.		Utilizado para tipificar elementos como lições e exercícios, os tipos de matéria dada são relacionados aos mesmos para os conseguir agrupar por matéria exposta.
Administrador	Termo que representa um administrador do sistema.		Responsável pela gestão da informação do sistema, o administrador é capaz de adicionar e editar lições e exercícios

Tabela 1 - Entidades

VII. Dicionário de Dados – Relacionamentos

Entidade	Multiplicidade	Relacionamento	Multiplicidade	Entidade (Relacionada)
Aluno	N (opcional)	Assiste; Vê; Estuda.	N (opcional)	Licao
Teste	N (opcional)	Tem; Possui; Engloba.	N (obrigatório)	Exercicio
Licao	N (opcional)	Tem; Possui; Engloba.	N (opcional)	Exercicio
Licao	N (opcional)	Tem; Possui.	1 (obrigatório)	Tipo
Exercicio	N (opcional)	Tem; Possui.	N (obrigatório)	Tipo
Exercicio	1 (opcional)	Tem; Possui.	N (obrigatório)	Resposta
Aluno Teste Exercicio	N (opcional) N (opcional) N (opcional)	Aluno realiza exercício no teste	N (opcional) N (opcional) N (opcional)	Exercicio Aluno Teste
Aluno Licao Exercicio	N (opcional) N (opcional) N (opcional)	Aluno realiza exercício na lição	N (opcional) N (opcional) N (opcional)	Aluno Licao Exercicio
Administrador	1 (opcional)	Gere	N (opcional)	Exercicio
Administrador	1 (opcional)	Gere	N (opcional)	Licao

Tabela 2 - Relacionamentos

VIII. Dicionário de Dados – Atributos

Nome da Entidade	Atributo	Descrição	Tipo de Dados & Comprimento/ Domínio	Alias	Nulo	Tipo de Atributo	Valor por "default"
Aluno	<u>IdAluno</u>	Identifica unicamente um aluno	INT		Não	Simple	Auto incrementado
	Nome	Nome completo do aluno	VARCHAR (75)		Não	Simple	
	Data de Nascimento	Data de Nascimento de um aluno	DATE		Sim	Simple	
	Username	Nome de utilizador do aluno	VARCHAR (75)		Não	Simple	
	Password	Palavra passe do aluno	VARCHAR (75)		Não	Simple	
	Aprendizagem	Estado da aprendizagem de um aluno numa dada matéria numa dada data	--		--	Composto multivalorado (Data, Tipo, Estado)	
	Data	Data do registo de aprendizagem	DATE		Sim	Composto	
	Tipo	Código que identifica a matéria referida na aprendizagem	INT		Sim	Composto	
	Estado	Número identificador do estado/grau de aprendizagem	FLOAT		Sim	Composto, Derivado	
	Pontuacao	Número representativo da pontuação de um aluno	INT (positivo)		Não	Derivado	0
	Configuracao	Configuração do ambiente do tutor escolhido pelo aluno	--		--	Composto (Dica, Explicacao, Tema)	
	Dica	Representa se o aluno deseja ou não receber dicas	BOOL		Não	Composto	TRUE
	Explicacao	Indica se o aluno deseja ou não receber explicações extras	BOOL		Não	Composto	TRUE
	Tema	Identifica qual o tema do ambiente de trabalho do aluno	INT		Não	Composto	0

Tabela 3 - Atributos (Aluno)

Nome da Entidade	Atributo	Descrição	Tipo de Dados & Comprimento/ Domínio	Alias	Nulo	Tipo de Atributo	Valor por "default"
Licao	<u>NumLicao</u>	Identifica unicamente, em conjunto com <u>NumExpl</u> , uma lição	INT		Não	Simples	
	Explicacao	Conteúdo de uma lição	--		--	Composto(Imagem, Video, Texto)	
	Imagem	Elemento de estudo do formato imagem	BLOB		Sim	Composto	
	Video	Elemento de estudo do formato video	VARCHAR (500)		Sim	Composto	
	Texto	Elemento de estudo do formato texto	VARCHAR (700)		Não	Composto	
	<u>NumExpl</u>	Identifica unicamente, em conjunto com <u>NumLicao</u> , uma lição	INT		Não	Simples	
	Tempo	Duração da lição	TIME		Não	Simples	

Tabela 4 – Atributos (Licao)

Nome da Entidade	Atributo	Descrição	Tipo de Dados & Comprimento/ Domínio	Alias	Nulo	Tipo de Atributo	Valor por "default"
Administrador	<u>idAdmin</u>	Identifica unicamente um Administrador	INT		Não	Simples	Auto-Incrementado
	Data de Nascimento	Identifica a data de nascimento de um administrador	DATE		Sim	Simples	
	Nome	Nome do administrador	VARCHAR (75)		Não	Simples	
	Username	Elemento de estudo do formato vídeo	VARCHAR (75)		Não	Simples	
	Password	Elemento de estudo do formato texto	VARCHAR (75)		Não	Simples	

Tabela 5 - Atributos (Administrador)

Nome da Entidade	Atributo	Descrição	Tipo de Dados & Comprimento/ Domínio	Alias	Nulo	Tipo de Atributo	Valor por "default"
Teste	<u>idTeste</u>	Identifica unicamente um Teste	INT		Não	Simples	Auto-Incrementado
	Dificuldade Teste	Identifica a dificuldade de um teste	FLOAT		Não	Derivado	

Tabela 6 - Atributos (Teste)

Nome da Entidade	Atributo	Descrição	Tipo de Dados & Comprimento/ Domínio	Alias	Nulo	Tipo de Atributo	Valor por "default"
Exercicio	<u>idExercicio</u>	Identifica unicamente um exercício	INT		Não	Simples	Auto-Incrementado
	Dificuldade	Dificuldade de um exercício	INT		Não	Simples	
	Desafio	Desafio do exercício (texto ou imagem que apresenta em si o exercício)	--		--	Composto(Texto, Imagem)	
	Texto (Desafio)	Elemento de estudo do formato texto	VARCHAR (700)		Sim	Composto	
	Imagem (Desafio)	Elemento de estudo do formato imagem	BLOB		Sim	Composto	
	TempoEx	Tempo disponibilizado para a realização do exercício	TIME		Não	Simples	
	Dica	Ajuda em modo texto ou imagem que pode ser apresentada aquando a realização de um exercício	--		--	Multivalorado Composto(Texto, Imagem)	
	Texto (Dica)	Elemento de estudo do formato texto	VARCHAR (500)		Sim	Composto	
	Imagem (Dica)	Elemento de estudo do formato imagem	BLOB		Sim	Composto	

Tabela 7 - Atributos (Exercicio)

Nome da Entidade	Atributo	Descrição	Tipo de Dados & Comprimento/ Domínio	Alias	Nulo	Tipo de Atributo	Valor por "default"
Resposta	<u>idResposta</u>	Identifica unicamente uma resposta	INT		Não	Simples	Auto-Incrementado
	Texto	Elemento de exposição da resposta no formato texto	VARCHAR (250)		Não	Simples	
	Pontuacao	Cotação associada à resposta	INT		Não	Simples	

Tabela 8 - Atributos (Resposta)

Nome da Entidade	Atributo	Descrição	Tipo de Dados & Comprimento/ Domínio	Alias	Nulo	Tipo de Atributo	Valor por "default"
Tipo	<u>idTipo</u>	Identifica unicamente um Tipo de matéria	INT		Não	Simples	Auto-Incrementado
	Area	Identifica textualmente a área	VARCHAR (150)		Não	Simples	

Tabela 9 - Atributos (Tipo)