

Parte 1

Atenção:

1. As respostas da P4 devem ser colocadas numa folha em branco identificada como "**Parte 1**"
2. Deve colocar o seu número e nome no topo da folha anexa

P1 (4v)

Assinale com V as afirmações verdadeiras e com F as falsas. **Uma resposta errada desconta meia certa.**

- _____ Com Ecore também podemos modelar parcialmente características comportamentais de um elemento *EClass*.
- _____ A ferramenta Eclipse EMF permite a definição de sintaxes abstratas.
- _____ Num modelo de domínio os *Value Objects* não são imutáveis.
- _____ Um elemento Ecore *EEnum* permite definir uma lista de valores e pode ser utilizado para definir o tipo de elementos *EAttribute*.
- _____ Uma *Graphical Concrete Syntax* pode ser transformada numa *Textual Concrete Syntax*.
- _____ Numa abordagem MDSE uma ferramenta de modelação é obrigatoriamente uma ferramenta de desenho.
- _____ CIM, PIM e PSM referem-se a modelos com diferentes níveis de abstração em MDA, sendo um CIM o mais abstrato.
- _____ *Templates* são as estruturas Aceleo utilizadas para geração de modelos.
- _____ Em Aceleo um *template* tem de ser definido dentro de um módulo e um módulo pode conter vários *templates*.
- _____ Num modelo de domínio as associações permitem atingir alguns elementos a partir de outros e é desejável que essa navegação não seja restringida.
- _____ Num modelo de domínio os *Entities* não são imutáveis.
- _____ Uma regra ATL abstracta não é executada.
- _____ Em ATL podemos ter variáveis globais mas não operações globais.
- _____ Um elemento Ecore *EReference* tem obrigatoriamente um tipo associado.
- _____ Um modelo PIM não especifica comportamento.
- _____ Ecore é uma ferramenta que permite definir modelos.
- _____ Numa abordagem com linhas de produtos de software interessam as diferenças entre produtos.
- _____ Uma regra ATL para ser estendida tem de ser abstracta.
- _____ A operação *oclIsTypeOf(t : oclType)* retorna o valor booleano *true* caso *self* seja uma instância de *t*.
- _____ Uma transformação ATL pode utilizar expressões OCL.

P2 (1,5v)

Nas frases seguintes selecione a opção que permite obter uma afirmação correta. **Uma resposta errada desconta meia certa.**

1. A linguagem OCL permite
 - (a) validar atributos mas não definir atributos.
 - (b) definir, com base noutros elementos, atributos e operações.
 - (c) validações mas sem utilização de diferentes atributos de diferentes classes.
 - (d) nenhuma das anteriores.

2. Num modelo de domínio. . .

- (a) *services* representam comportamento e podem utilizar *entities* e *value objects*.
- (b) podemos ter *services* sem significado de negócio mas que são tecnicamente necessários.
- (c) *services* representam comportamento e não podem utilizar *entities*.
- (d) *services* representam comportamento e não podem utilizar *value objects*.

3. MDSE...

- (a) complementa a abordagem *Domain Driven Design*.
- (b) é complementada pela abordagem *Domain Driven Design*.
- (c) e *Domain Driven Design* são abordagens totalmente disitintas.
- (d) faz um uso menos intenso de modelos do que a abordagem *Domain Driven Design*.

4. No seguinte extrato definido com a ferramenta OCLinEcore e relativo a um metamodelo estudado nas aulas

```
class Topic extends MapElement { ... invariant EndAfterStart: self.end >= self.start; }
```

- (a) representa o que se designa como *invariant* e que pode ser definido na linguagem ecore sem recurso a OCL.
- (b) *self* poderá representar uma data.
- (c) *self* é obrigatoriamente uma instância de um elemento **MapElement** mas a validação pode não se aplicar a todas as suas instâncias.
- (d) *self* é obrigatoriamente uma instância de um elemento **MapElement** e a validação aplica-se a todas as suas instâncias.

P3 (1,5v)

Associe a cada conceito da primeira coluna o elemento da segunda coluna que melhor se relaciona. Uma resposta errada desconta meia certa.

_____ Extrato código ATL	(a) Suporta apenas relações binárias
_____ Modelo de classes UML	(b) Diagrama de estrutura
_____ Java	(c) <code>rule person2family { ... }</code>
_____ UML Component Diagram	(d) <code>let yearIncome:Real self.monthIncome*14 in ...</code>
_____ Software Product Lines	(e) Diagrama de comportamento
_____ Extrato código OCL	(f) Suporta associações n-árias
_____ UML Use Case Diagram	(g) General Purpose Language
_____ Metamodelação Ecore	(h) Strategic Reuse

P4 (2,5v) - Responda na folha "Parte 1"

Apresente uma breve **definição** e **descrição** para os seguintes conceitos.

1. **Software Product Line.**
2. **Agile Modeling.**

Parte 2

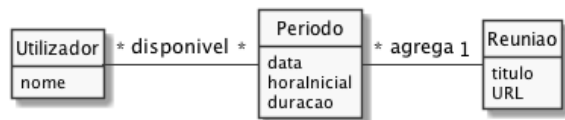
Atenção:

1. Todas as justificações e respostas desta parte devem ser colocadas numa folha em branco identificada como "Parte 2"
2. Nas perguntas com justificação esta vale 50%.
3. Deve colocar o seu número e nome no topo da folha anexa

P5 (2v) - Responda na folha "Parte 2"

Uma plataforma deverá permitir que uma pessoa introduza uma série de possibilidades para a realização de uma reunião e a aplicação indicará um URL que, divulgado posteriormente sem utilização da plataforma, permitirá que as pessoas indiquem

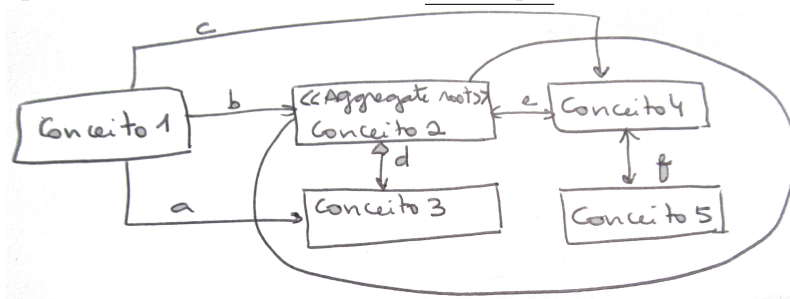
as suas disponibilidades. Não estão previstas nem são desejáveis outras funcionalidades. Considere o seguinte extrato de um diagrama de modelo de domínio sem indicação da direção das associações. **Preencha os espaços em falta de forma a completar corretamente o texto.** A justificação vale 50% da nota.



1. O conceito Utilizador deve ser modelado como _____ [*Entity* — *Value Object*].
2. A associação *disponivel* a ser navegável apenas a partir de _____ [*Utilizador* — *Período*] permitiria manter *Utilizador* independente de *Período* (não são relevantes outras considerações).
3. A associação *agrega* deve ser _____ [unidirecional — bidirecional].
4. No caso da *Entity* *Reuniao* o atributo que podemos usar para a sua identidade é o _____ [*titulo* — *URL*].

P6 (2v) - Responda na folha "Parte 2"

Considere o seguinte extrato de um diagrama de modelo de domínio, ainda incompleto. Sabe-se que *Conceito4* e *Conceito5* são *value objects*, e *Conceito3* é um *entity*. **Indique o que falta nos espaços em branco de forma a completar correctamente o texto. Justifique.**



1. A associação _____, que envolve um ou mais conceitos externos ao *aggregate*, é inválida.
2. A associação _____, dentro do *aggregate*, pode existir mas não deve ser bidirecional.
3. A associação _____, dentro do *aggregate*, não deve existir.
4. O conceito _____ deverá ter uma identidade local.

P7 (2v) - Responda na folha "Parte 2"

Considere o seguinte extrato produzido com a ferramenta OCLinEcore e semelhante a um metamodelo estudado nas aulas. O extrato está incompleto. O modelo representa Pessoas. Cada elemento *Person* representa uma pessoa. Cada pessoa tem obrigatoriamente um nome e tem uma ocupação, normalmente estudante (*default value*) e além disso pode ter um local de nascimento associado (*Country*).

```

1 package person : person = 'http://www.example.org/person'
2 {
3     abstract class Person
4     {
5         attribute age : ecore::EInt[?];
6         attribute name : String[ ... ] = '';
7         attribute occupation : String[?] = ...;
8         ... placeOfBirth : ...[?];
9     }
10    class Female extends Person;
11    class Male extends Person;
12    class Country
13    {
14        ... name : String[?] = '';
15    }
16 }
  
```

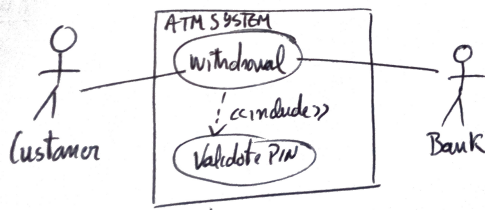
Indique o que falta nos espaços em branco de forma a completar corretamente o texto. Justifique.

1. Na linha 8, na primeira ocorrência de "...": _____, e na segunda ocorrência de "...": _____

2. Na linha 7 (nos "..."): _____.
3. Na linha 14 (nos "..."): _____.
4. Na linha 6 (nos "..."): ____.

P8 (2,5v) - Responda na folha "Parte 2"

Considere o seguinte diagrama de casos de uso cuja notação e semântica é similar à do UML.



1. Apresente um diagrama que ilustre um possível metamodelo que suporte instâncias como a ilustrada no diagrama anterior (use notação de diagramas de classes). Use os construtores possíveis no **ecore**. Não necessita de ser preciso na notação mas é necessário que explique o modelo que apresentar no contexto das possibilidade de metamodelação do **ecore**. Considere que existem as seguintes condicionantes: os casos de uso existem sempre no contexto de um **Subject** (no exemplo anterior é o "ATM System"); entre casos de uso podem existir apenas 2 tipos de relações: "extends" e "include"; os actores são sempre externos ao **Subject**; um modelo pode ter vários subjects.

P9 (2v) - Responda na folha "Parte 2"

Considere o seguinte código ATL relativo a uma transformação de um modelo Library com livros e escritores para um modelo Person (quase completo em P7).

```

1  -- @path Library=/Library/model/library.ecore
2  -- @path Person=/Person/model/person.ecore
3
4  module Author2Person;
5  create OUT : Person from IN : Library;
6
7  rule Author2Male {
8    from
9      a: Library!Author
10   to
11     p: Person!Male (
12       name <- a.name
13     )
14 }

```

Relativamente ao modelo Library considere o seguinte extrato produzido com a ferramenta OCLinEcore:

```

1  ...
2  class Author
3  {
4    attribute name : String[?] = '';
5    attribute gender : Gender[?];
6  }
7  enum Gender { serializable }
8  {
9    literal Female;
10   literal Male = 1;
11   literal Unknown = 2;
12 }

```

1. Altere a regra Author2Male de forma que apenas autores do sexo masculino sejam considerados nessa regra e que a ocupação da pessoa (**occupation**) passe a ser sempre *escritor*.
2. Defina uma regra Author2Female de forma a termos elementos Female partir de autores que **não sejam do sexo masculino**.

Answer Key for Exam A

Parte 1

Atenção:

1. As respostas da P4 devem ser colocadas numa folha em branco identificada como "Parte 1"
2. Deve colocar o seu número e nome no topo da folha anexa

P1 (4v)

Assinale com V as afirmações verdadeiras e com F as falsas. **Uma resposta errada desconta meia certa.**

- | | |
|-------|--|
| True | Com Ecore também podemos modelar parcialmente características comportamentais de um elemento <i>EClass</i> . |
| True | A ferramenta Eclipse EMF permite a definição de sintaxes abstratas. |
| False | Num modelo de domínio os <i>Value Objects</i> não são imutáveis. |
| True | Um elemento Ecore <i>EEnum</i> permite definir uma lista de valores e pode ser utilizado para definir o tipo de elementos <i>EAttribute</i> . |
| True | Uma <i>Graphical Concrete Syntax</i> pode ser transformada numa <i>Textual Concrete Syntax</i> . |
| False | Numa abordagem MDSE uma ferramenta de modelação é obrigatoriamente uma ferramenta de desenho. |
| True | CIM, PIM e PSM referem-se a modelos com diferentes níveis de abstração em MDA, sendo um CIM o mais abstrato. |
| False | <i>Templates</i> são as estruturas Aceleo utilizadas para geração de modelos. |
| True | Em Aceleo um <i>template</i> tem de ser definido dentro de um módulo e um módulo pode conter vários <i>templates</i> . |
| False | Num modelo de domínio as associações permitem atingir alguns elementos a partir de outros e é desejável que essa navegação não seja restringida. |
| True | Num modelo de domínio os <i>Entities</i> não são imutáveis. |
| True | Uma regra ATL abstracta não é executada. |
| False | Em ATL podemos ter variáveis globais mas não operações globais. |
| True | Um elemento Ecore <i>EReference</i> tem obrigatoriamente um tipo associado. |
| False | Um modelo PIM não especifica comportamento. |
| False | Ecore é uma ferramenta que permite definir modelos. |
| True | Numa abordagem com linhas de produtos de software interessam as diferenças entre produtos. |
| False | Uma regra ATL para ser estendida tem de ser abstracta. |
| True | A operação <i>oclIsTypeOf</i> (<i>t</i> : <i>oclType</i>) retorna o valor booleano <i>true</i> caso <i>self</i> seja uma instância de <i>t</i> . |
| True | Uma transformação ATL pode utilizar expressões OCL. |

P2 (1,5v)

Nas frases seguintes selecione a opção que permite obter uma afirmação correta. **Uma resposta errada desconta meia certa.**

1. A linguagem OCL permite
 - (a) validar atributos mas não definir atributos.
 - (b) definir, com base noutros elementos, atributos e operações.
 - (c) validações mas sem utilização de diferentes atributos de diferentes classes.
 - (d) nenhuma das anteriores.
2. Num modelo de domínio...
 - (a) *services* representam comportamento e podem utilizar *entities* e *value objects*.
 - (b) podemos ter *services* sem significado de negócio mas que são tecnicamente necessários.
 - (c) *services* representam comportamento e não podem utilizar *entities*.
 - (d) *services* representam comportamento e não podem utilizar *value objects*.

3. MDSE...

- (a) complementa a abordagem *Domain Driven Design*.
 - (b) é complementada pela abordagem *Domain Driven Design*.
 - (c) e *Domain Driven Design* são abordagens totalmente disitintas.
 - (d) faz um uso menos intenso de modelos do que a abordagem *Domain Driven Design*.
4. No seguinte extrato definido com a ferramenta OCLinEcore e relativo a um metamodelo estudado nas aulas
- ```
class Topic extends MapElement { ... invariant EndAfterStart: self.end >= self.start; }
```
- (a) representa o que se designa como *invariant* e que pode ser definido na linguagem ecore sem recurso a OCL.
  - (b) *self* poderá representar uma data.
  - (c) *self* é obrigatoriamente uma instância de um elemento **MapElement** mas a validação pode não se aplicar a todas as suas instâncias.
  - (d) *self* é obrigatoriamente uma instância de um elemento **MapElement** e a validação aplica-se a todas as suas instâncias.

### P3 (1,5v)

Associe a cada conceito da primeira coluna o elemento da segunda coluna que melhor se relaciona. Uma resposta errada desconta meia certa.

|                            |                                                                 |
|----------------------------|-----------------------------------------------------------------|
| (c) Extrato código ATL     | (a) Suporta apenas relações binárias                            |
| (f) Modelo de classes UML  | (b) Diagrama de estrutura                                       |
| (g) Java                   | (c) <code>rule person2family { ... }</code>                     |
| (b) UML Component Diagram  | (d) <code>let yearIncome:Real self.monthIncome*14 in ...</code> |
| (h) Software Product Lines | (e) Diagrama de comportamento                                   |
| (d) Extrato código OCL     | (f) Suporta associações n-árias                                 |
| (e) UML Use Case Diagram   | (g) General Purpose Language                                    |
| (a) Metamodelação Ecore    | (h) Strategic Reuse                                             |

### P4 (2,5v) - Responda na folha "Parte 1"

Apresente uma breve **definição** e **descrição** para os seguintes conceitos.

1. **Software Product Line.** Um conjunto de sistemas de software que têm em comum um determinado conjunto de funcionalidades , e que satisfazem as necessidades de um determinado segmento de mercado ou missão, e que são desenvolvidos tendo a mesma base (core). Assim, importam as semelhanças entre os vários sistemas mas também o que os distingue com uma gestão dessa variabilidade. A abordagem deve estar alinhada com a estratégia de longo prazo do negócio. Há uma redução de custos relacionados com o desenvolvimento, mas tipicamente apenas com um certo n° de sistemas individuais.
2. **Agile Modeling.** Agile Modeling é um conjunto de princípios e práticas que visam tornar ágil o processo de desenvolvimento de software. Assenta na ideia que a modelação não deve ser utilizada com o propósito de termos modelos mas é útil se permmitte obter software a partir de múltiplos modelos simples, cada um a descrever um único aspeto do sistema. Outros principios: feedback rápido,blablabla

## Parte 2

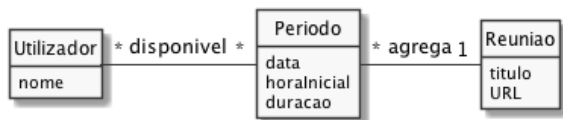
**Atenção:**

1. Todas as justificações e respostas desta parte devem ser colocadas numa folha em branco identificada como "Parte 2"
2. Nas perguntas com justificação esta vale 50%.
3. Deve colocar o seu número e nome no topo da folha anexa

### P5 (2v) - Responda na folha "Parte 2"

Uma plataforma deverá permitir que uma pessoa introduza uma série de possibilidades para a realização de uma reunião e a aplicação indicará um URL que, divulgado posteriormente sem utilização da plataforma, permitirá que as pessoas indiquem

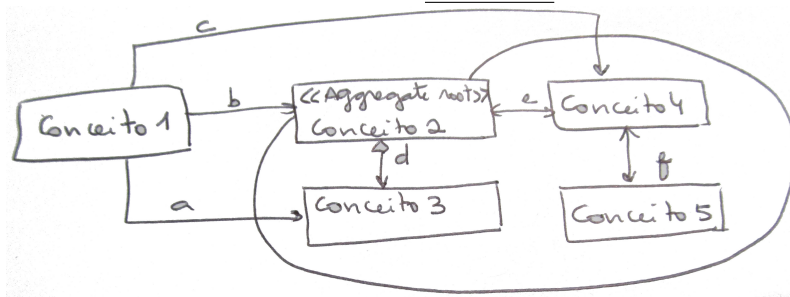
as suas disponibilidades. Não estão previstas nem são desejáveis outras funcionalidades. Considere o seguinte extrato de um diagrama de modelo de domínio sem indicação da direção das associações. **Preencha os espaços em falta de forma a completar corretamente o texto.** A justificação vale 50% da nota.



1. O conceito Utilizador deve ser modelado como Value Object [*Entity* — *Value Object*].
2. A associação disponivel a ser navegável apenas a partir de Período [ *Utilizador* — *Período*] permitiria manter *Utilizador* independente de *Período* (não são relevantes outras considerações).
3. A associação agrega deve ser unidirecional [unidirecional—bidirecional].
4. No caso da *Entity* *Reuniao* o atributo que podemos usar para a sua identidade é o URL [ *titulo* — *URL*].

## P6 (2v) - Responda na folha "Parte 2"

Considere o seguinte extrato de um diagrama de modelo de domínio, ainda incompleto. Sabe-se que *Conceito4* e *Conceito5* são *value objects*, e *Conceito3* é um *entity*. **Indique o que falta nos espaços em branco de forma a completar correctamente o texto. Justifique.**



1. A associação a ou c, que envolve um ou mais conceitos externos ao *aggregate*, é inválida.
2. A associação e, dentro do *aggregate*, pode existir mas não deve ser bidirecional.
3. A associação f, dentro do *aggregate*, não deve existir.
4. O conceito Conceito3 deverá ter uma identidade local.

## P7 (2v) - Responda na folha "Parte 2"

Considere o seguinte extrato produzido com a ferramenta OCLinEcore e semelhante a um metamodelo estudado nas aulas. O extrato está incompleto. O modelo representa Pessoas. Cada elemento *Person* representa uma pessoa. Cada pessoa tem obrigatoriamente um nome e tem uma ocupação, normalmente estudante (*default value*) e além disso pode ter um local de nascimento associado (*Country*).

```

1 package person : person = 'http://www.example.org/person'
2 {
3 abstract class Person
4 {
5 attribute age : ecore::EInt[?];
6 attribute name : String[...] = '';
7 attribute occupation : String[?] = ...;
8 ... placeOfBirth : ...[?];
9 }
10 class Female extends Person;
11 class Male extends Person;
12 class Country
13 {
14 ... name : String[?] = '';
15 }
16 }

```

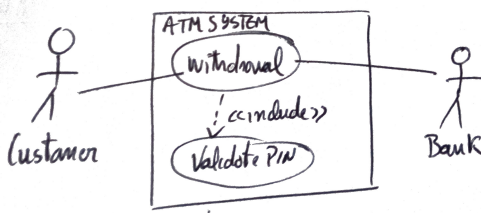
**Indique o que falta nos espaços em branco de forma a completar corretamente o texto. Justifique.**

1. Na linha 8, na primeira ocorrência de "...": property, e na segunda ocorrência de "...": Country

2. Na linha 7 (nos "..."): 'Student'.
3. Na linha 14 (nos "..."): attribute.
4. Na linha 6 (nos "..."): 1.

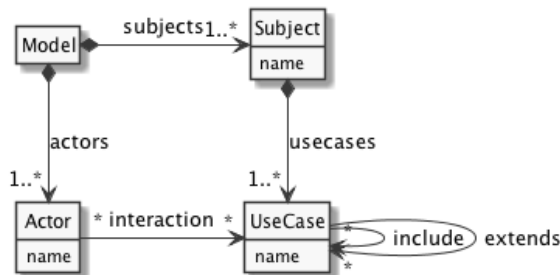
## P8 (2,5v) - Responda na folha "Parte 2"

Considere o seguinte diagrama de casos de uso cuja notação e semântica é similar à do UML.



1. Apresente um diagrama que ilustre um possível metamodelo que suporte instâncias como a ilustrada no diagrama anterior (use notação de diagramas de classes). Use os construtores possíveis no **ecore**. Não necessita de ser preciso na notação mas é necessário que explique o modelo que apresentar no contexto das possibilidade de metamodelação do **ecore**. Considere que existem as seguintes condicionantes: os casos de uso existem sempre no contexto de um **Subject** (no exemplo anterior é o "ATM System"); entre casos de uso podem existir apenas 2 tipos de relações: "extends" e "include"; os actores são sempre externos ao **Subject**; um modelo pode ter vários subjects.

**Resposta:** O metamodelo pode ser algo similar ao seguinte (pode haver variantes conforme o detalhe considerado e



as assunções).

## P9 (2v) - Responda na folha "Parte 2"

Considere o seguinte código ATL relativo a uma transformação de um modelo Library com livros e escritores para um modelo Person (quase completo em P7).

```

1 -- @path Library=/Library/model/library.ecore
2 -- @path Person=/Person/model/person.ecore
3
4 module Author2Person;
5 create OUT : Person from IN : Library;
6
7 rule Author2Male {
8 from
9 a: Library!Author
10 to
11 p: Person!Male (
12 name <- a.name
13)
14 }

```

Relativamente ao modelo Library considere o seguinte extrato produzido com a ferramenta OCLinEcore:

```

1 ...
2 class Author
3 {
4 attribute name : String[?] = '';
5 attribute gender : Gender[?];
6 }
7 enum Gender { serializable }
8 {
9 literal Female;
10 literal Male = 1;
11 literal Unknown = 2;
12 }

```



1. Altere a regra Author2Male de forma que apenas autores do sexo masculino sejam considerados nessa regra e que a ocupação da pessoa (occupation) passe a ser sempre *escritor*.

```
1 rule Author2Male {
2 from
3 a: Library!Author (a.gender = #Male)
4 to
5 p: Person!Male (
6 name <- a.name,
7 occupation <- 'escritor'
8)
9 }
```

2. Defina uma regra Author2Female de forma a termos elementos Female partir de autores que **não sejam do sexo masculino**.

```
1 rule Author2Female {
2 from
3 a: Library!Author (not (a.gender = #Male))
4 to
5 p: Person!Female (
6 name <- a.name
7)
8 }
```