

Enunciado EP

Parte 1: Modelagem e Base de dados

MAC 0350: Introdução ao desenvolvimento de sistemas de Software

João Eduardo Ferreira

jef@ime.usp.br

Décio Lauro Soares

deciolauro@gmail.com

Renato Cordeiro

renatocf@ime.usp.br

29 de abril de 2018

1 Instruções para entrega

Nesta primeira parte do EP vocês deverão trabalhar em grupo de até 3 pessoas.

Apenas um dos integrantes do grupo deverá submeter o arquivo compactado até o dia 10/05/2018 às 23:59 e este será denominado como NUSP1 a partir desse ponto.

Para a entrega, o arquivo deverá conter a seguinte estrutura:

```
EP1_NUSP1_NUSP2_NUSP3.zip
├── EP1_README.txt
├── EP1_NUSP1_REPORT.pdf
├── EP1_NUSP1_DDL.sql
├── EP1_NUSP1_DDL_CLEAN.sql
├── EP1_NUSP1_DML_INSERT.sql
├── EP1_NUSP1_DML_QUERY.sql
└── EP1_NUSP1_DML_CLEAN.sql
```

Onde NUSP1, NUSP2 e NUSP3 correspondem ao número USP de cada um dos integrantes, separados por um underscore. (Obs.: nos casos dos grupos individuais ou em duplas, basta omitir os NUSP's correspondentes)

À seção Resumo das atividades **3** apresenta uma versão resumida dos 10 itens que compõem a elaboração desse EP e as seções posteriores explicam cada um desses itens em detalhe.

O arquivo EP1_README.txt deve conter no cabeçalho o nome completo e NUSP de todos os integrantes e eventuais instruções de instalação/execução/uso que eventualmente forem necessárias. (Para essa primeira parte do EP, a menos que você utilize um SGBD **diferente** do PostgreSQL, provavelmente não haverá necessidade de incluir qualquer instrução)

No arquivo EP1_NUSP1_REPORT.pdf vocês deverão entregar um relatório formal (capa, sumário, ...) descrevendo cada um dos 10 itens desse EP, preferencialmente dividido em capítulos/seções para cada um dos itens.

Os arquivos *DDL.sql são explicados em detalhes no item VII enquanto que os arquivos *DML*.sql são explicados nos itens IX e X.

2 Critérios de avaliação

Esta primeira parte do EP será avaliada da seguinte maneira:

EP Parte 1 - relatório & scripts Nota máxima 10.0

Relatório PDF (6.5)	
Modelo Geral (I) + Descrições (II, III, IV) + Físico (VII, VIII, IX, X) - 2.0	
MER-X (V) - 2.0	
Modelo lógico (VI) - 2.0	
Estrutura do texto e apresentação das consultas 0.5 + bônus de até 0.5	
Scripts DDL (2.0)	
DDL.sql - 1.5	
DDL_CLEAN.sql - 0.5	
Scripts DML (1.5)	
DML_INSERT.sql 0.5	
DML_CLEAN.sql 0.5	
DML_QUERY.sql 0.5 + bônus de até 0.5	

3 Resumo das atividades

De forma resumida, as atividades previstas para o relatório à ser entregue para esta primeira parte do EP são:

- I. Complementar o modelo geral da figura 1 para incluir o módulo de controle de acesso;
- II. Realizar a descrição completa da previsão de entidades, relacionamentos, ... e suas justificativas;
- III. Realizar a descrição completa das restrições de domínio e apresentar exemplos de entrada (pelo menos dois) para cada relação;
- IV. Realizar a descrição do conjunto de funcionalidades esperadas;
- V. Escrever o Modelo Entidade-Relacionamento Estendido (MER-X) completo com base no modelo geral realizado no item I;
- VI. Realizar o mapeamento do MER-X do item V para o modelo lógico (incluindo informações sobre eventuais modificações necessárias);
- VII. Realizar o mapeamento do modelo lógico para o modelo físico;
- VIII. Escrever os scripts para criação/configuração da base de dados (preferencialmente usando postgresQL) e sua eventual remoção;
- IX. Criar o script para inserção de alguns dados (pelo menos 5 tuplas por relação) em SQL e sua posterior remoção;
- X. Apresentar 5 consultas (pelo menos 2 não triviais) em SQL e seus resultados sobre a base criada.

As seções a seguir apresentam a descrição detalhada sobre cada um dos itens.

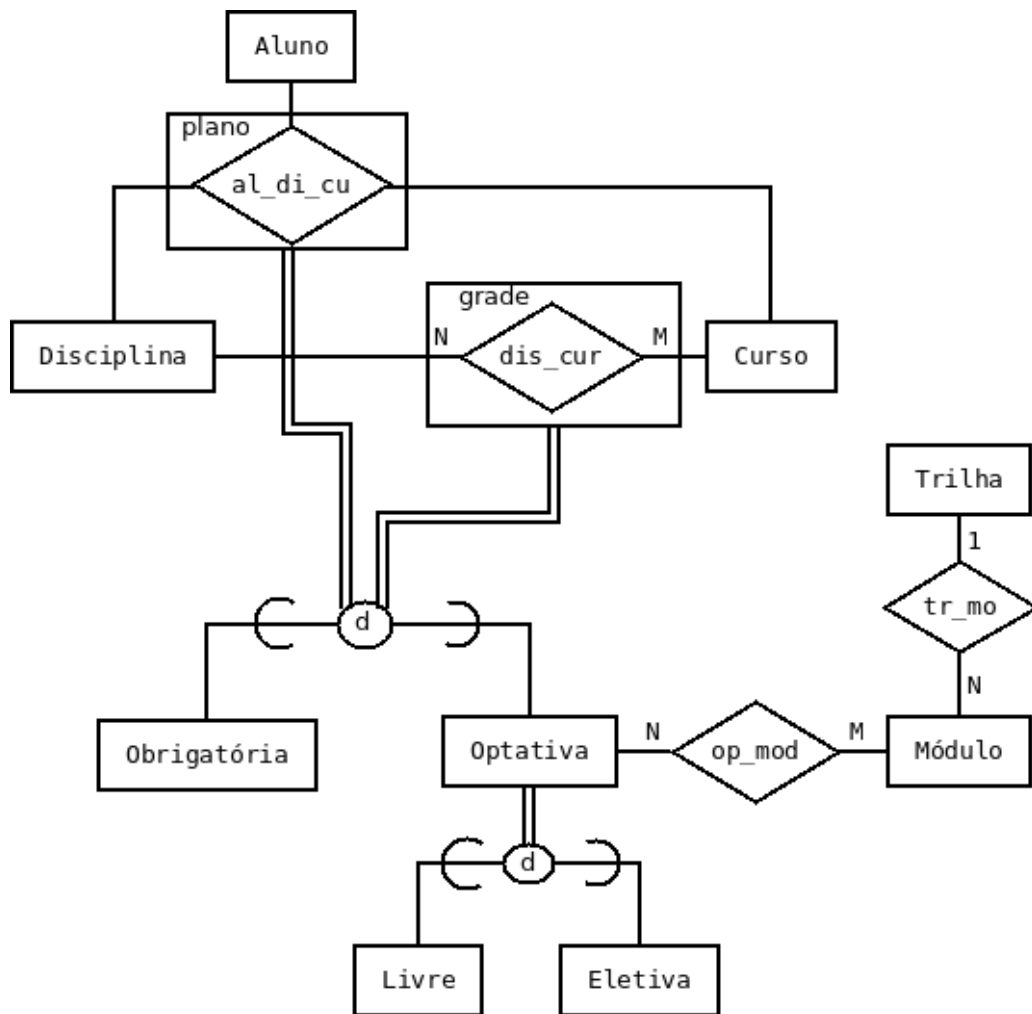


Fig. 1: Modelo geral desenvolvido em sala (sem controle de acesso)

4 Descrição detalhada: Item I.

Ao longo das primeiras aulas trabalhamos no modelo geral apresentado na figura 1.

Contudo, esse modelo não traz o módulo de controle de acesso (módulo de acesso).

Portanto, é preciso estender o modelo geral para incluir esse módulo, de forma análoga ao que foi desenvolvido na primeira prova (tutor/tutoria).

Sugerimos o uso da ferramenta Dia ([6]) para construção tanto do modelo geral como do MER-X à ser apresentado no item V.

Entretanto, uma vez que o modelo deste item é geral, você pode apresentá-lo usando à ferramenta/simbologia que desejar, desde que seu relatório apresente o modelo geral **com controle de acesso** e que eventuais simbologias que você inventar/usar estejam **devidamente explicadas**.

5 Descrição detalhada: Item II.

O modelo geral apresentado no item anterior é apenas um esboço em notação conveniente da base de dados que será gerada para o projeto em questão.

Para o bom desenvolvimento de sistemas, é preciso que haja uma descrição detalhada de cada uma das relações, seus atributos e justificativas.

Nesse sentido, o item II espera que você descreva de forma textual detalhada cada um dos componentes do modelo geral.

Você deve apresentar à descrição detalhada de **todas as relações presentes** no modelo (entidades regulares, entidades fracas, relacionamentos, agregações, especializações, ...).

É importante também **explicar à razão** da escolha/necessidade da criação de entidades mais elaboradas, como especializações, agregações, ..., bem como à razão/necessidade de relacionamentos N-ários ($N > 2$).

A subseção abaixo apresenta um possível exemplo da descrição da entidade regular Aluno do nosso modelo.

Caso você deseje complementar o trabalho, é nesta seção que você pode também incluir à parte correspondente em MER-X da relação que você está descrevendo, no caso do exemplo, seria a inclusão da figura correspondente à entidade regular Aluno em notação MER-X.

5.1 (Exemplo) Entidade Regular: Aluno

A entidade regular Aluno é uma entidade chave para nosso modelo e é nela que armazenaremos todos os dados das instâncias de cada um dos aluno da USP presentes na nossa base de dados.

Segundo nosso modelo, cada instância de Aluno será composta por seu número USP, que será utilizado como chave primária, seu CPF que será utilizado como chave secundária, seu nome que não admite valores do tipo NULL, seu endereço que será um atributo composto de logradouro, rua, ..., seu ... (continuar listando detalhadamente cada um dos atributos).

O listing abaixo¹ traz um exemplo da estrutura da entidade:

Listing 1: (Exemplo)A entidade regular Aluno

```
Aluno
-Número USP *Chave primária
-CPF **Chave secundária
-Nome
-Idade
-Sexo
-Curso
```

-Endereço

:Continuar listando

Enquanto que a figura 2 apresenta a representação em MER-X da entidade descrita.

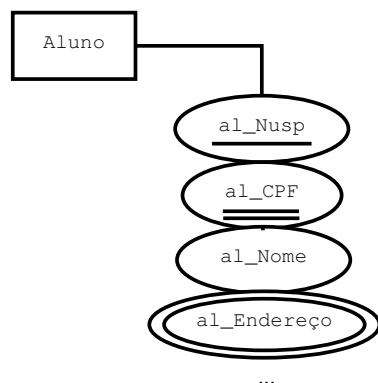


Fig. 2: Aluno em formato MER-X (Obs.: **Incompleto**)

6 Descrição detalhada: Item III.

No item anterior, você fez a descrição detalhada das relações e seus atributos. Nesse item você deve **explicitar** as restrições de domínio e suas justificativas.

A subseção abaixo traz um exemplo (incompleto) das restrições de domínio sobre a entidade Aluno apresentada anteriormente.

6.1 (Exemplo) Restrições Entidade Regular: Aluno

A entidade Aluno possui as seguintes restrições de domínio:

- O atributo al_Nusp é composto por uma sequência de 6 à 9 dígitos inteiros (Obs.: Não sei se existe NUSP com 6 dígitos e estou supondo que o sistema já prevê a extensão para Nusp com 9 dígitos, tudo isso foi descrito no item anterior :-)) e por ser chave não deve permitir a presença de entradas do tipo NULL;
- O atributo al_CPF é composto por uma sequência de 11 algarismos e deve ser constituída de uma sequência válida segundo às regras da Receita Federal (listar as regras de congruência, dígito verificador de estado, ...), além disso, por ser chave secundária esse atributo deve ser do tipo NOTNULL(NN);
- O atributo endereço é um atributo composto que será dividido em (CEP, NomeRua, NumRua, Compl, Bairro, ...), o CEP é composto por ... e deve ser do tipo NN, se possível, implementaremos um verificador de endereço via CEP, caso contrário, o nome da rua será uma string variável de tamanho X, o Compl pode ser do tipo NULL nos casos onde ele não exista, ...;
- O atributo sexo deve ser um char presente no conjunto (M, F, N) onde M representa Masculino, F feminino e N não declarado;
- : (Obs.: Continuar listando todas as restrições de domínio)

O listing 2 traz um exemplo de possíveis tuplas à serem inseridas em aluno:

Listing 2: Exemplos de entrada para as Entidades

```
Aluno: (al_Nusp, al_CPF, al_Nome, al_idade, al_sexo, ...)  
-7777777; 003.939.708-41; Juca Medonho; 22; M; ...  
-7777888; 076.713.684-58; Célia Nefasta; 18; N; ...
```

7 Descrição detalhada: Item IV.

Neste item você deve realizar uma descrição geral de quais funcionalidades estão previstas para o produto final. Neste momento você não deve se preocupar com questões de implementação, é importante apenas **listar todas as funcionalidades esperadas**.

Até porque, futuramente nas Partes 2 e 3 deste EP, algumas dessas funcionalidades deverão ser adaptadas/estendidas/descartadas,...

A subseção abaixo traz um exemplo de algumas (poucas) funcionalidades esperadas.

7.1 (Exemplo) Funcionalidades esperadas

Nosso projeto prevê as seguintes funcionalidades:

- Permitir ao aluno planejar um cronograma de disciplinas segundo uma trilha escolhida;
- Permitir ao administrador recuperar os dados dos alunos e suas notas nas disciplinas cursadas;
- Permitir ao administrador verificar o número de alunos que completaram determinada trilha;
- :

8 Descrição detalhada: Item V.

Nesse item você deve **apresentar o diagrama de Entidade-Relacionamento estendido** em notação compatível com MER-X ou usando a notação adaptada no contexto de sala de sala (vide [2] e/ou [5]) do modelo geral apresentado no item I.

Este modelo deverá ser completo e não serão admitidas outras notações/simbologias que não estejam presentes nas referências citadas.

9 Descrição detalhada: Item VI.

Nesse item você deverá fazer o **mapeamento do MER-X para o modelo lógico**.

Sugerimos que vocês utilizem os algoritmos de mapeamento apresentados em sala (vide [3] e/ou [5]) mas, contanto que seu mapeamento esteja correto, você pode utilizar qualquer algoritmo que desejar.

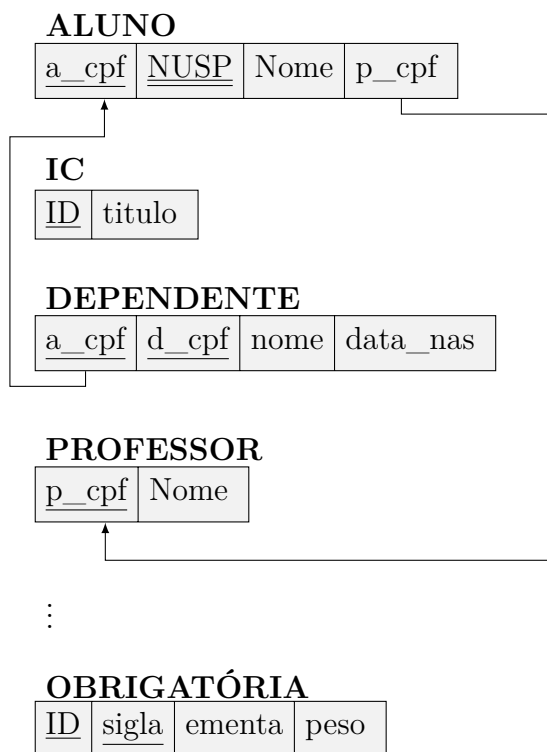
Tenham cuidado na apresentação, especialmente na parte referente às foreign keys.

Uma sugestão é usar o tikzpicture (vide [1]), o mapeamento apresentado a seguir foi gerado a partir do seguinte código:

```

1 \begin{tikzpicture}[relation/.style={rectangle split, rectangle split parts=#1, rectangle
    split part align=base, draw, anchor=center, align=center, text height=3mm, text
    centered}]\hspace*{-0.3cm}
2
3 % RELATIONS
4
5 \node (alunotitle) {\textbf{ALUNO}};
6
7 \node [relation=4, rectangle split horizontal, rectangle split part fill={lightgray!50},
    anchor=north west, below=0.6cm of alunotitle.west, anchor=west] (aluno)
8 {\underline{a\_cpf}}%
9 \nodepart{two} \underline{\underline{NUSP}}
10 \nodepart{three} Nome
11 \nodepart{four} p\_cpf
12 };
13
14 \node [below=1.3cm of aluno.west, anchor=west] (ictitle) {\textbf{IC}};
15
16 \node [relation=2, rectangle split horizontal, rectangle split part fill={lightgray!50},
    below=0.6cm of ictitle.west, anchor=west] (ic)
17 {\underline{ID}}%
18 \nodepart{two} titulo
19 };
20
21 \node [below=1.1cm of ic.west, anchor=west] (deptitle) {\textbf{DEPENDENTE}};
22
23 \node [relation=4, rectangle split horizontal, rectangle split part fill={lightgray!50},
    anchor=north west, below=0.6cm of deptitle.west, anchor=west] (dependente)
24 {\underline{a\_cpf}}%
25 \nodepart{two} \underline{d\_cpf}
26 \nodepart{three} nome
27 \nodepart{four} data\_nas
28 };
29
30 \node [below=1.4cm of dependente.west, anchor=west] (proftitle) {\textbf{PROFESSOR}};
31
32 \node [relation=2, rectangle split horizontal, rectangle split part fill={lightgray!50},
    anchor=north west, below=0.6cm of proftitle.west, anchor=west] (professor)
33 {\underline{p\_cpf}}%
34 \nodepart{two} Nome
35 };
36
37 \node [below=1.5cm of professor.west, anchor=west] (ell1) {$\vdots$};
38
39 \node [below=1.0cm of ell1.west, anchor=west] (obrigatoriatitle) {\textbf{OBRIGATÓRIA}};
40
41 \node [relation=4, rectangle split horizontal, rectangle split part fill={lightgray!50},
    anchor=north west, below=0.6cm of obligatoriatitle.west, anchor=west] (obrigatoria)
42 {\underline{ID}}%
43 \nodepart{two} \underline{sigla}
44 \nodepart{three} ementa
45 \nodepart{four} peso
46 };
47
48 % FOREIGN KEYS
49
50 \draw[-latex] (aluno.four south) -- ++(0,-0.2) -| ($(aluno.four south) + (2,-0.2)$) |-
    ($$(professor.one south) + (0.25,-0.50)$) -| ($(professor.one south) + (0.25,0)$);
51 \draw[-latex] (dependente.one south) -- ++(0,-0.2) -| ($(dependente.one south) + (-1,-0.2)$)
    |- ($(aluno.one south) + (0.25,-0.50)$) -| ($(aluno.one south) + (0.25,0)$);
52
53 \end{tikzpicture}

```



10 Descrição detalhada: Itens VII e VIII.

Para esses dois itens, você deverá fazer o **mapeamento do modelo lógico para o modelo físico** e escrever dois scripts SQL (*EP1_NUSP1_DDL.sql* e *EP1_NUSP1_DDL_CLEAN.sql*), sendo o primeiro para a criação e configuração da base de dados e o segundo para limpeza/remoção completa do banco.

Nesta primeira parte do projeto, não há necessidade de integrar os scripts SQL à qualquer tipo de interface em Python, ou seja, seu script deve apenas ser capaz de criar as relações no SGBD que você escolher (de preferência o PostgreSQL) e toda a análise será feita via interface direta com o banco (por exemplo, via *psql*).

Recomendamos fortemente que vocês escolham utilizar como SGBD o PostgreSQL, uma vez que já há um tutorial de instalação/configuração/uso do mesmo no Paca (vide [4]).

Caso você opte por utilizar outro SGBD, você **deverá incluir todos** os scripts de instalação e configuração, de preferência fazendo uso de ambiente virtual ou através da utilização de tecnologias como *Docker*.

Além disso, caso escolha outro SGBD, você também deverá providenciar **todos os comandos** para acesso ao SGBD e verificação de integridade da sua base de dados no SGBD que você escolher.

É importante ressaltar que projetos/relatórios receberão nota zero nesse quesito caso os scripts para outros SGBD's (que não sejam o PostgreSQL) não funcionem ou não sejam fornecidos os comandos para verificação da integridade do banco.

Por fim, seus scripts podem estar inclusos no corpo do relatório, mas **também devem** ser entregues em arquivos separados denominados *EP1_NUSP1_DDL.sql* e *EP1_NUSP1_DDL_CLEAN.sql*.

11 Descrição detalhada: Item IX.

Neste item, você deve providenciar dois scripts SQL denominados *EP1_NUSP1_DML_INSERT.sql* e *EP1_NUSP1_DML_CLEAN.sql*.

O primeiro *EP1_NUSP1_DML_INSERT.sql* deve conter à inserção de **pelo menos** 5 tuplas em cada uma das relações do seu banco, enquanto que o segundo deve remover todas essas tuplas **sem** alterar à estrutura do banco. (Ou seja, deve remover as tuplas mas deixar a relação inalterada)

12 Descrição detalhada: Item X.

Neste item, você deve providenciar um script SQL denominado *EP1_NUSP1_DML_QUERY.sql* contendo **pelo menos** 5 consultas das quais pelo menos 2 devem ser não triviais. (Ou seja, não vale fazer `5 SELECT * FROM ...`)

Visando evitar o retrabalho, é interessante que suas consultas estejam relacionadas com as funcionalidades apresentadas no item IV, pois de outra forma você terá mais trabalho nas próximas partes.

Por essa razão, **pode haver um bônus** na nota final caso suas consultas apresentem o grau de maturidade esperado.

Bibliography

- [1] Jacques Crémer. A very minimal introduction to tikz. <https://cremeronline.com/LaTeX/minimaltikz.pdf>. Accessed: 27/04/18.
- [2] Ramez Elmasri and Shamkant B. Navathe. *Database Systems, 7th Ed.* Pearson, 2015.
- [3] Décio Lauro Soares, João Eduardo Ferreira, and Renato Cordeiro. Slides da aula de revisão de mapeamento (2018). www.paca.ime.usp.br, 2018. Accessed: 27/04/18.
- [4] Décio Lauro Soares, Eduardo Dias Filho, and Bruno Padilha. Tutorial python e postgresql. www.paca.ime.usp.br. Accessed: 27/04/18.
- [5] Osvaldo Kotaro Takai, Isabel Cristina Italiano, and João Eduardo Ferreira. Introdução à banco de dados. www.ime.usp.br/~jef/apostila.pdf, 2005. Accessed: 27/04/18.
- [6] Dia volunteer development group. Dia diagram editor. <http://dia-installer.de/>. Accessed: 27/04/18.