MAC0350

Parte 2: Setup e design de interface

MAC 0350: Introdução ao desenvolvimento de sistemas de Software

João Eduardo Ferreira `jef@ime.usp.br`
Décio Lauro Soares `deciolauro@gmail.com`
Renato Cordeiro Ferreira `renatocf@ime.usp.br`

28 de maio de 2018

1 Instruções para entrega

Nesta parte do projeto vocês deverão trabalhar em **grupos de até 3 pessoas**.

Apenas um dos integrantes do grupo deverá submeter o arquivo compactado até o dia 03/06/2018 (domingo) às 23h55. Para a entrega, o arquivo deverá conter a seguinte estrutura:

```
EP2_NUSP1_NUSP2_NUSP3.zip
├── README.md
└── REPORT.pdf
```

A Seção 2 apresenta uma versão resumida dos itens que compõem a elaboração deste EP. As seções posteriores explicam cada um desses itens em detalhe.

O arquivo README deve conter, em seu início, o nome completo e NUSP de todos os integrantes. Fique à vontade para fazê-lo em *markdown* ou em *plain text*.

2 Resumo das atividades

De forma resumida, as atividades previstas para o relatório a ser entregue para esta segunda parte do projeto são:

I. Compreender as mudanças no modelo consolidado apresentado na Figura 1 através da leitura do sumário explicativo da Seção 3;

II. Apresentar um sumário conceitual dos perfis, serviços e consultas planejados;

III. Criar um *wireframe* com o planejamento da interface do sistema, apresentando as principais funcionalidades que serão implementadas para os usuários;

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

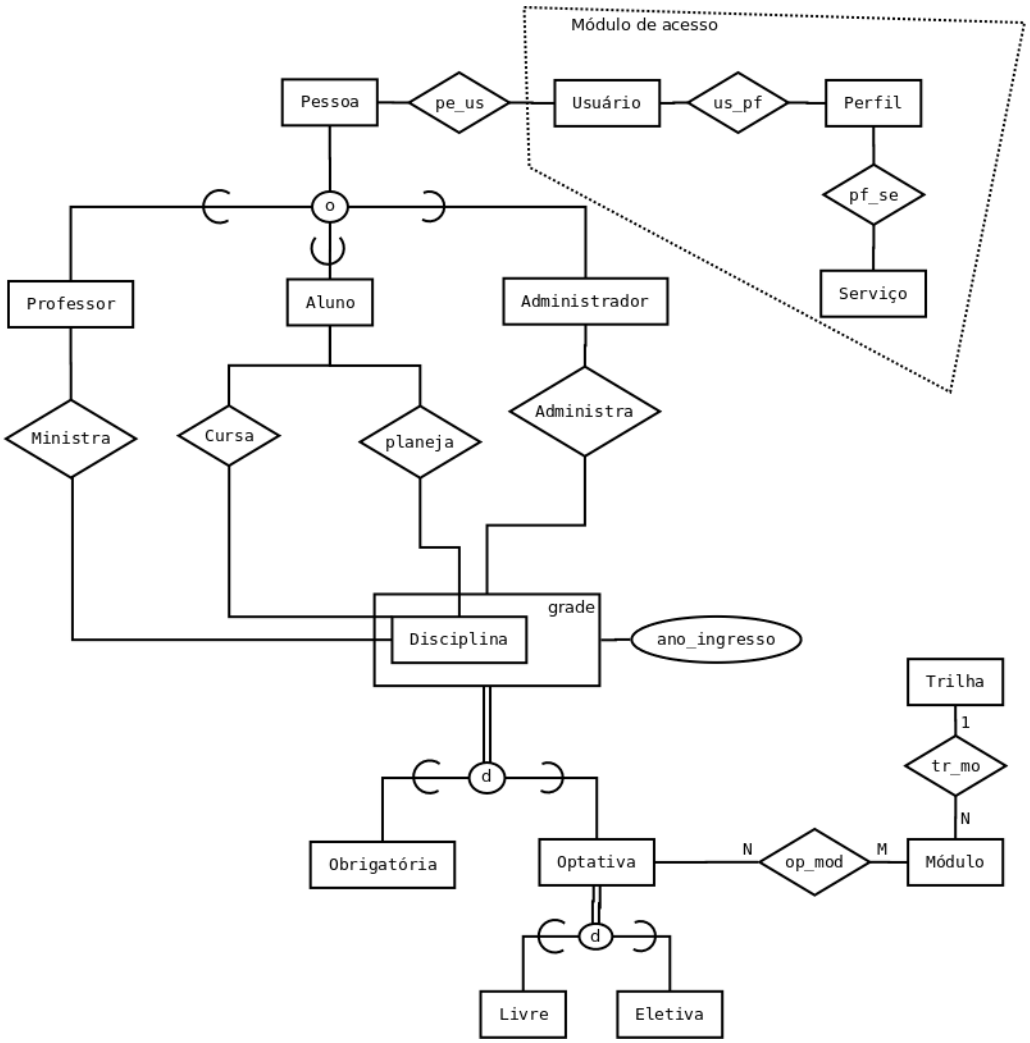

Figura 1: **Modelo de dados consolidado do Yggdrasil**, com módulo de controle de acesso e sem a entidade *Curso*.

3 Descrição detalhada do item I

Neste item do EP, trabalharemos com uma versão consolidada do modelo desenvolvido durante o EP1. A Figura 1 traz o modelo descrito de forma simplificada usando a simbologia do MER-X.

Os principais pontos de destaque são:

- Remoção da entidade `Curso` e consequente simplificação das relações;
- Inclusão (para os que não o haviam feito no EP1) do módulo de acesso;
- Inclusão da entidade `Pessoa` que tem **especialização parcial sobreponível** em três entidades: Aluno, Professor e Administrador;
- Inclusão do relacionamento `Cursa` e simplificação do relacionamento `Al\_Di\_Cu` no relacionamento `Planeja`;
- Adaptação da agregação `Grade` para comportar o modelo que agora representa um único curso.

Dado que vocês já fizeram a descrição completa do MER-X na primeira parte do projeto, faremos agora só uma descrição superficial das principais mudanças/adições que foram realizadas no modelo.

3.1 Módulo de acesso

Apesar de extensivamente trabalhado em sala, dado à importância do módulo de acesso ao nosso modelo, faremos uma descrição superficial de suas propriedades.

3.1.1 Usuário

Começando pela entidade `Usuário`, qualquer agente acessando o sistema, sendo ele representado por uma tupla em `Pessoa` ou não, somente deve ter acesso se passar por algum tipo de validação sobre a entidade `Usuário`.

Por exemplo, suponha que as tabelas abaixo mostrem uma instanciação válida da base em um dado momento:

Usuário

uid	uun	upassword
1	jef	'c386950aa5131b703f031267f77e1075'
2	decio	'1c0ef710ebb4f1f1f2fef6a007d2dc68'
3	renato	'ec02d2d95c27675d87dca50018d89192'
4	juca	'7b5a67574d8b1d77d2803b24946950f0'
5	guest	'084e0343a0486ff05530df6c705c8bb4'

Pessoa

pid	pn	ucpf
101	João	11111111111
777	Décio	22222222222
312	Renato	33333333333
467	Ana	44444444444

pe_us

pid	uid	expire
101	1	31/12/2023
777	2	31/12/2020
312	3	31/12/2020

Isso significa que as pessoas João, Décio e Renato podem ter acesso aos serviços disponíveis através dos seus nomes de usuário ("jef", "decio"e "renato"). Alguém cujo nome de usuário é "juca"também pode ter acesso aos serviços, apesar de não ter instância na relação *Pessoa*.

Contudo, Ana, apesar de estar instanciada na relação *Pessoa*, só terá acesso aos serviços caso se autentique como *guest*.

Assim, suponha que todos os usuários cadastrados, por meio do seu perfil não mencionado, tenham acesso ao serviço `criar_plano`. Pelo menos duas considerações que vocês já devem pensar são:

- O que acontece quando juca tentar criar um plano? Se ele não tem instância em *Pessoa*, também não terá instância em *Cursa*. Ele pode criar plano mesmo assim?
- Ana poderá planejar usando *guest*. Se for o caso, o que a impede de planejar por outra pessoa?

3.1.2 Perfil e Serviço

Além de `Usuário`, as relações `Perfil` e `Serviço` terminam de compor o módulo de acesso.

Em `Perfil`, mantemos o conjunto de descrições semânticas dos diferentes "tipos" de acesso aos serviços. Em `Serviços`, guardamos todo o conjunto de funcionalidades, controles, visualizações, etc., da nossa aplicação.

Por exemplo, considerando que as tabelas abaixo, juntamente com as tabelas anteriores mostrem uma instanciação válida da base em um dado momento:

Perfil

pid	pname
5	admin
7	prof-admin
15	prof
10	aluno
12	aluno-pae
30	guest

us_pf

pid	uid	expire
5	1	31/12/2023
7	1	31/12/2020
30	1	01/01/3000
12	2	08/01/2019
⋮		
30	5	01/01/3000

Serviço

sid	sname
1	inserir_admin
2	alterar_nota
3	planejar
4	cursar
⋮	
34	alterar_grade

pe_se

pid	sid	expires
5	1	31/12/2020
5	2	31/12/2005
7	2	31/12/2020
15	2	31/12/2020
⋮		
10	4	31/12/2018

Podemos ver pelas tabelas que `Perfil` controla o conjunto de `Serviço` acessível. Notamos que até 2005 alguém com o perfil de *admin* podia realizar a alteração de notas na relação `Cursa`. Essa regra expirou e a decisão de projeto foi criar um novo perfil *prof-admin* para cuidar desse novo caso de uso.

Vemos também que um usuário pode pertencer a uma infinidade de perfis e o mesmo serviço pode ser acessível por diferentes tipos de perfil.

Perguntas que devem nortear a implementação dessa parte são:

- Quais perfis criar?
- Quais serviços criar **além** do CRUD básico?
- Que perfis terão acesso à determinados serviços?

3.2 Entidades regulares Professor e Administrador

Essas duas entidades compõem à especialização/generalização de `Pessoa` e são praticamente autoexplicativas, inclusive seus relacionamentos `Ministra` e `Administra`.

3.3 Agregação grade

Diferentemente do modelo anterior, a agregação grade agora é composta apenas por um conjunto de disciplinas.

Ela certamente possuí um atributo ano de ingresso, já que é necessário controlar as diferentes grades ao longo dos anos.

É importante se atentar para o fato de que é `grade`, e não `disciplina`, que se especializa de forma disjunta em `Obrigatória` e `Optativa`.

A disciplina por si só não tem esse caráter, já que, por exemplo, na grade atual MAC0350 é obrigatória, mas pode ter sido optativa numa grade anterior (digamos 2003).

Assim, embora para essa entrega isso não seja relevante, será fundamental para a próxima.

3.4 Relacionamentos Cursa e Planeja

Como já foi mencionado anteriormente, o antigo relacionamento ternário `Al_Di_Cu` agora tornou-se o relacionamento binário `Planeja` entre `Aluno` e `Disciplina`.

Temos também à adição do relacionamento `Cursa`, que agora compõe o conjunto de disciplinas que efetivamente foi cursada pelo aluno (sua *execução*).

As principais considerações que devem ser feitas sobre esses relacionamentos são:

- `Planeja` vai guardar os planos, mas terá que ser atualizada ao final de cada período após à inclusão das notas em `Cursa`;
- `Disciplinas` possuem pré-requisitos. Logo, antes de permitir acesso à interface de planejamento, o sistema deve primeiro verificar as disciplinas já cursadas pelo aluno (para não permitir o planejamento de uma disciplina já feita) e deve também controlar a inclusão ("mostrar na tela") apenas disciplinas que respeitem as condições de pré-requisito. Por exemplo: suponha que MAC0002 tenha pré-requisito total em MAC0001. Sua interface não deve permitir (talvez nem mesmo mostrar) a opção de planejar MAC0002 sem antes "forçar" que o usuário planeje MAC0001.
- Outro ponto que muitos não consideraram no EP1 é que disciplinas geralmente possuem um semestre de oferecimento. Assim, supondo que MAC0001 tenha paridade impar (ou seja, só seja oferecida nos semestres impares 1o, 3o, 5o, etc.), seu sistema não deve permitir (talvez nem mesmo mostrar) MAC0001 na janela de planejamento do segundo semestre.

4 Descrição detalhada do item II

No segundo item para entrega neste EP, vocês devem apresentar a descrição do conjunto de buscas previstas sobre o modelo, a descrição dos serviços e dos perfis esperados.

Em tese vocês já deveriam ter está parte pronta do EP1 e agora só bastaria apresentar a síntese conceitual. Todas as descrições de consultas, perfis e serviços devem ser entregues juntamente com o material do *wireframe*.

4.1 Perfis

Nesta parte vocês devem fazer uma descrição sucinta de todos os perfis previstos na sua aplicação, por exemplo:

- Perfil `admin`: Responsável pela manutenção da base de dados, oferecido apenas para usuários com alto conhecimento técnico na área;
- Perfil `prof-admin`: Responsável pela gestão de curso / departamento, oferecido apenas aos professores coordenadores de departamento.
- ⋮

4.2 Serviços

Nesta parte vocês precisam apresentar os serviços conceitualmente, por exemplo:

- Serviço `insere_professor`: Insere um professor e estará disponível para os perfis (admin, prof-admin, ...);
- Serviço `consulta_prereq`: Retorna todos os pré-requisitos de uma dada disciplina e estará disponivel para os perfis (admin, prof-admin, prof, aluno, aluno-pae, guest, ...);
- ⋮

4.3 Consultas

Nesta parte, basta apresentar às buscas conceitualmente, por exemplo:

- Consultar média de notas por turma (serviço `consulta_médias`);
- Listar todos os professores que já ministraram alguma vez a disciplina X (serviço `consulta_professor_disciplina`);
- Listar as disciplinas que serão oferecidas no próximo semestre (serviço `consulta_oferecimentos`);
- ⋮

5 Descrição detalhada do item III

No terceiro item para entrega neste EP, vocês devem apresentar um *wireframe* da interface de usuário.

O **wireframe** é um guia visual da estrutura de uma página, cujo foco é representar a **disposição** do conteúdo nela. Ao contrário de um *mockup*, que é uma representação de alta fidelidade do layout da interface, vocês não precisarão se preocupar com o conteúdo real (como nomes e siglas corretos das disciplinas) ou detalhes de estilo (como cores) ao produzir o*wireframe*.

É possível imaginar que um professor gostaria de uma tela específica para cadastrar novas disciplina, ou que um professor-administrador queira uma tela para visualizar estatísticas do curso pelo qual ele é responsável. Por simplicidade, porém, vamos focar no projeto da tela principal do sistema, que será utilizada pelos alunos para visualizar os seus planos e registrar a realização desse plano.

Existem várias funcionalidades que devem ser incluídas nessa tela:

- **Visualização da estrutura do curso**: Uma pessoa deve ser capazes de visualizar a estrutura do curso BCC, representada pelas disciplinas divididas entre módulos e respectivas trilhas, para posteriormente criar planos e marcar as disciplinas realizadas. Isso poderá ser feito mesmo que a pessoa não esteja logada.

- **Seleção de disciplinas**: Uma pessoa deve ser capaz de selecionar as disciplinas para que façam parte de um plano ou sejam marcadas como realizadas. A seleção deve ser habilitada se e somente se as disciplinas requisito tiverem sido previamente habilitadas. Isso poderá ser feito mesmo que a pessoa não esteja logada.
- **Login**: Um usuário deve ser capaz de logar no sistema a partir de seu e-mail e senha.
- **Criação de novo plano**: Um aluno (logado) deve ser capaz de salvar a atual seleção de disciplinas como novo plano.
- **Listagem de planos salvos**: Um aluno (logado) deve ser capaz de ver sua lista de planos salvos.
- **Recuperação de plano**: Um aluno (logado) deve ser capaz de alternar a sua visualização (ao clicar no nome de um dos planos salvos) para ver a seleção de disciplinas atrelada àquele plano.
- **Recuperação de realização**: Um aluno (logado) deve ser capaz de alternar a sua visualização (ao clicar em um botão específico) para ver suas disciplinas realizadas.
- **Contagem de créditos**: Um aluno (logado) deve ser capaz de ver a contagem de créditos (divididas em suas categorias correspondentes) segundo as disciplinas marcadas na interface (que variarão se o plano ou realização estiverem sido mostradas).

Em tese, interfaces web modernas são construídas segundo a estratégia *mobile first*, buscando "crescer"a partir das telas pequenas e ir adequando-se para tamanhos maiores (responsividade). Para esse EP, porém, focaremos no projeto voltado para telas de laptop (que deverá ter a proporção de 1024x866 pixels).

Vocês deverão entregar duas figuras representando cada "estado"da página. A primeira figura deve conter a visualização para uma pessoa não logada (que não terá a lista de planos salvos) e outra figura para o aluno logado (com a opção de salvar os planos e recuperar a realização). É importante mencionar que estamos pensando em apenas **uma única página** contendo os elementos de interface citados acima.

Vocês podem se inspirar na versão original do Yggdrasil como ponto de partida. O site segue um formato clássico chamado "Santo Graal", que contém barra de navegação, barra lateral, rodapé e o conteúdo principal. Ele contém parte das funcionalidades citadas acima, em particular a representação da estrutura do curso, mas você pode imaginar opções diferentes. A bibliografia deste enunciado contém referências para vocês lerem mais sobre design, UX (experiência de usuário) e verem componentes que podem ser usados para criar uma interface.

Recomendamos utilizar o programa draw.io para criar o *wireframe*. Vocês devem salvar a imagem em formato PNG ou PDF e incluí-la no relatório junto com a especificação da Seção 4.

6 Critérios de avaliação

Esta segunda parte do EP (Primeira entrega) será avaliada da seguinte maneira:

```
EP Parte 2 - Nota máxima 10.0
├── Item I - 0 pontos
├── Item II - 2 pontos
└── Item III - 8 pontos
```

O item I, introdutório, serve apenas de revisão, portanto não será pontuado. O item II, derivado do conteúdo produzido no EP1, terá pontuação menor. O item III, principal adição desse EP, contém 8 funcionalidades que devem ser incluídos no projeto da interface, sendo que cada tópico valerá 1 ponto.

Referências Bibliográficas

[1] Awesome design list. https://github.com/gztchan/awesome-design. Accessed: 28/05/18.

[2] Awesome ux list. https://github.com/netoguimaraes/awesome-ux. Accessed: 28/05/18.

[3] Froala design blocks. https://www.froala.com/design-blocks. Accessed: 28/05/18.

[4] Front-end checklist. https://frontendchecklist.io/. Accessed: 28/05/18.

[5] Front-end design checklist. https://frontenddesignchecklist.io/. Accessed: 28/05/18.

[6] Holy grail (web design). https://en.wikipedia.org/wiki/Holy_grail_(web_design). Accessed: 28/05/18.

[7] Ramez Elmasri and Shamkant B. Navathe. *Database Systems, 7th Ed.* Pearson, 2015.

[8] 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.

[9] Matt Warcholinski. What is the difference between wireframe, mockup and prototype? https://brainhub.eu/blog/difference-between-wireframe-mockup-prototype/. Accessed: 28/05/18.