

TEXTO DE APOIO



AULA 2

Prática profissional em análise e desenvolvimento de sistemas

Professor Tomaz Mikio Sasaki



Universidade Presbiteriana
Mackenzie





Sumário



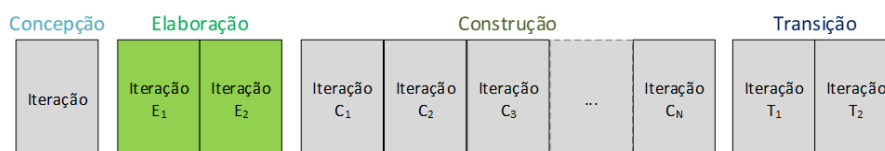
INTRODUÇÃO.....	3
PROTÓTIPOS DE TELA.....	4
MODELO DE DOMÍNIO	5
DIAGRAMAS DE SEQUÊNCIA.....	7
REFERÊNCIAS	10

ELABORAÇÃO DO PROJETO

INTRODUÇÃO

Em um Processo Unificado de desenvolvimento de software, a elaboração é a segunda fase (Figura 1).

Figura 1 – A elaboração é a segunda fase do Processo Unificado



Fonte: Elaborada pelo autor.

Essa fase geralmente é composta de duas ou três iterações, e seus principais resultados são: identificação dos principais requisitos, definição da arquitetura e estimativas mais precisas.

Nosso componente está organizado de forma a seguir aproximadamente um processo unificado, e o período da Aula 2 corresponde à fase de elaboração do projeto (Tabela 1) e será realizado em uma única iteração.

Tabela 1 – Nossa Aula 2 corresponde à fase de elaboração

Fase	Concepção	Elaboração	Construção	Transição		
Iteração	Única	Única	1	2	Única	
Aula	1	2	3	4	5	6

Fonte: Elaborada pelo autor.

Em nosso projeto, trabalharemos agora com os seguintes itens:

- Protótipos de tela
- Modelo de domínio
- Diagrama de classes de projeto
- Diagrama de sequência

PROTÓTIPOS DE TELA

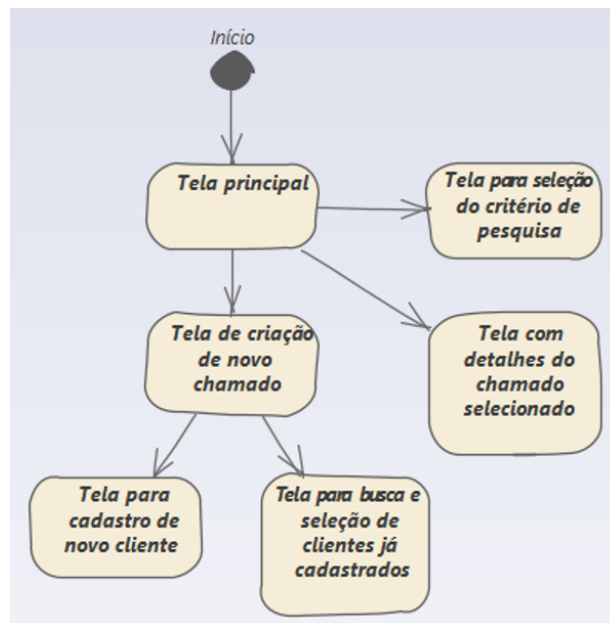
Os protótipos de tela ou *wireframes* **são os esboços preliminares das telas/páginas da aplicação. Eles mostram os aspectos essenciais do sistema de navegação entre as telas/páginas sem se preocupar ainda com o *design* do visual final.**

A equipe pode elaborar os protótipos de tela com maior ou menor nível de detalhes, mas eles precisam ter os seguintes itens essenciais:

- Navegação.
- Conteúdo de cada tela.
- Funcionalidades disponíveis em cada tela.

A navegação pode ser definida em um mapa de navegação (Figura 2), que é um diagrama que mostra os nomes das telas e os caminhos de navegação entre elas.

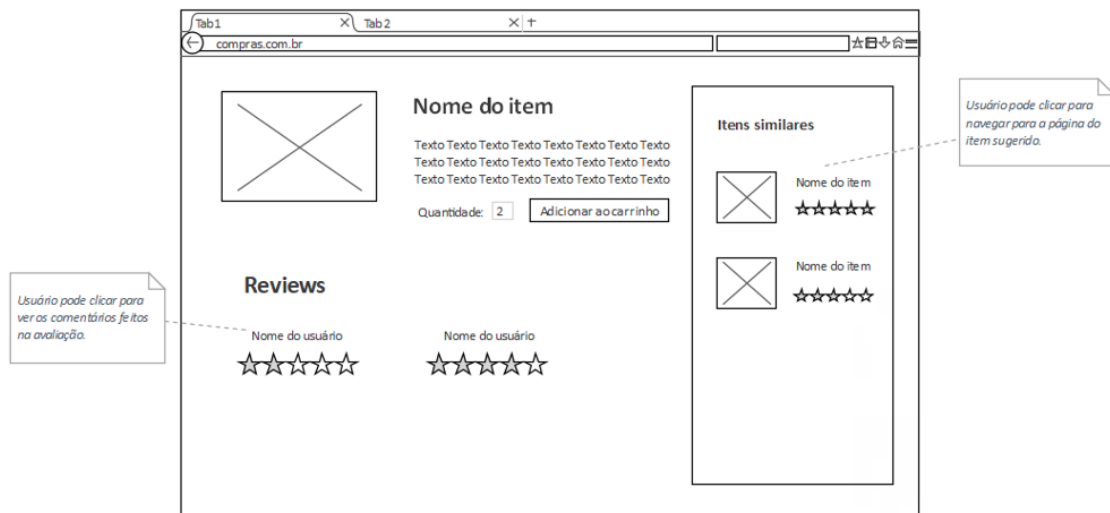
Figura 2 – Exemplo de mapa de navegação



Fonte: Elaborada pelo autor.

Para cada tela, deve-se elaborar um esboço (Figura 3) mostrando seu conteúdo e as funcionalidades disponíveis (campos de entrada de dados e controles para ações do usuário).

Figura 3 – Exemplo de esboço de uma tela



Fonte: Elaborada pelo autor.

MODELO DE DOMÍNIO

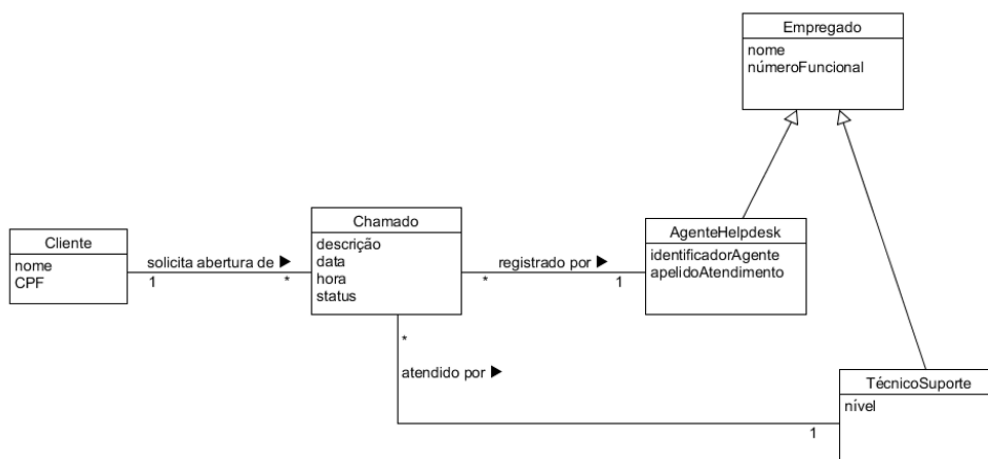
O modelo de domínio é representado como um diagrama com as classes conceituais do domínio do problema. Isso quer dizer que ele **não mostra classes de software** (por exemplo, aquelas responsáveis pelo acesso à base de dados ou pelo controle das interações com o usuário).

As principais características de um diagrama do modelo de domínio são:

- Mostram somente as classes conceituais.
- Em cada classe, mostram os atributos, mas **não mostram os métodos**.
- Mostram as associações, com nomes e multiplicidades. As agregações e composições, que são associações do tipo parte-todo, também são mostradas.
- Mostram os relacionamentos de herança.

A Figura 4 apresenta um exemplo de diagrama do modelo de domínio.

Figura 4 – Exemplo de diagrama do modelo de domínio



Fonte: Elaborada pelo autor.

O modelo de domínio é também chamado de “dicionário visual”, pois apresenta de forma gráfica os principais conceitos e os relacionamentos entre eles.

DIAGRAMAS DE CLASSES DE PROJETO

Uma aplicação típica precisa realizar diversos tipos de operações. Alguns exemplos destas operações são:

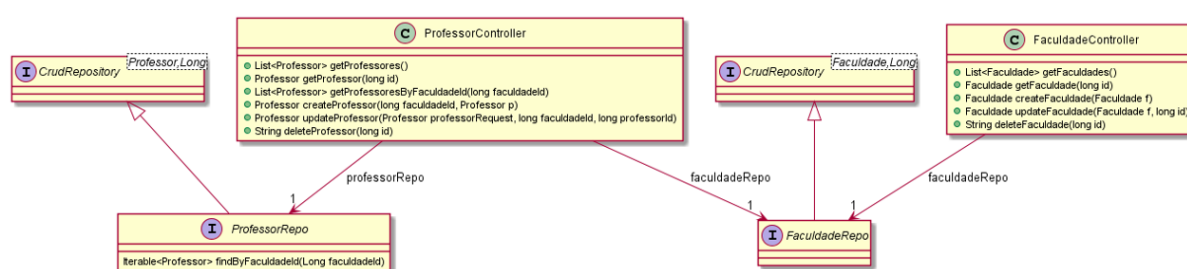
- Apresentar uma determinada tela na interface do usuário.
- Executar um determinado algoritmo em resposta a alguma ação realizada pelo usuário.
- Ler e armazenar informações em um banco de dados.
- Realizar transações com outros sistemas.

Precisamos delegar essas responsabilidades para classes específicas em nosso software e definir como serão as interações entre as instâncias dessas classes. Muitos chamam isso de elaborar o projeto (*design*), ou definir a arquitetura da solução.

Este projeto (*design*) é bastante influenciado pelo tipo de aplicação (por exemplo, se é uma aplicação web, uma aplicação móvel ou uma aplicação de *back-end*) e pela decisão de utilizar ou não algum *framework* específico. Com base nessas informações e decisões, podemos definir as classes de projeto.

A Figura 5 apresenta um exemplo de diagrama de classes de projeto.

Figura 5 – Exemplo de diagrama de classes de Projeto



Fonte: Elaborada pelo autor.

Esse exemplo mostra as classes de um projeto que utiliza o *framework* Spring Boot (CrudRepository é uma interface que está declarada nesse *framework*). A notação aplicada nesse diagrama é um pouco diferente da UML devido à configuração da ferramenta que foi utilizada para desenhar os diagramas, não porque os diagramas de projeto precisam ter notação diferente.

DIAGRAMAS DE SEQUÊNCIA

Nesse ponto do projeto, temos como mostrar que o desenho da solução é capaz de atender aos requisitos funcionais descritos pelos casos de uso.

Suponhamos que, na descrição do cenário principal de um dos casos de uso de um sistema acadêmico, tenhamos o seguinte trecho:

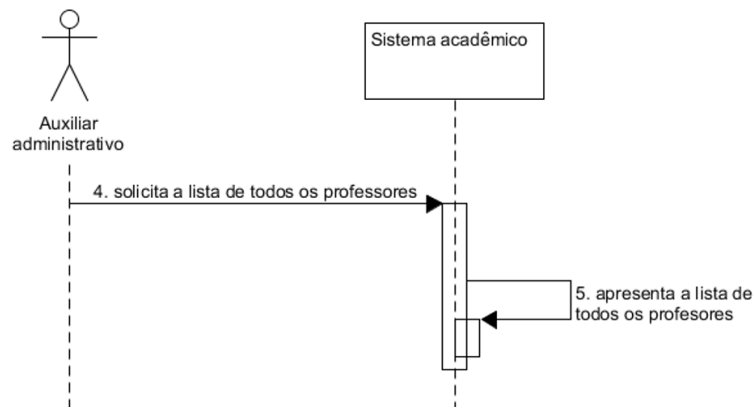
Tabela 2 – Exemplo de um trecho da descrição de um caso de uso

Passo	Descrição
...	...
4	Auxiliar administrativo solicita a lista de todos os professores.
5	Sistema acadêmico apresenta a lista de todos os professores.
...	

Fonte: Elaborada pelo autor.

Do ponto de vista sistêmico, poderíamos representar isso por um diagrama de sequência simplificado, que mostra apenas o ator e o sistema (Figura 6).

Figura 6 – Diagrama de sequência mostrando apenas o ator e o sistema



Fonte: Elaborada pelo autor.

Para mostrar que o desenho da nossa solução é capaz de atender aos requisitos funcionais desse trecho do caso de uso, podemos elaborar um diagrama de sequência que detalha as interações entre os componentes do sistema.

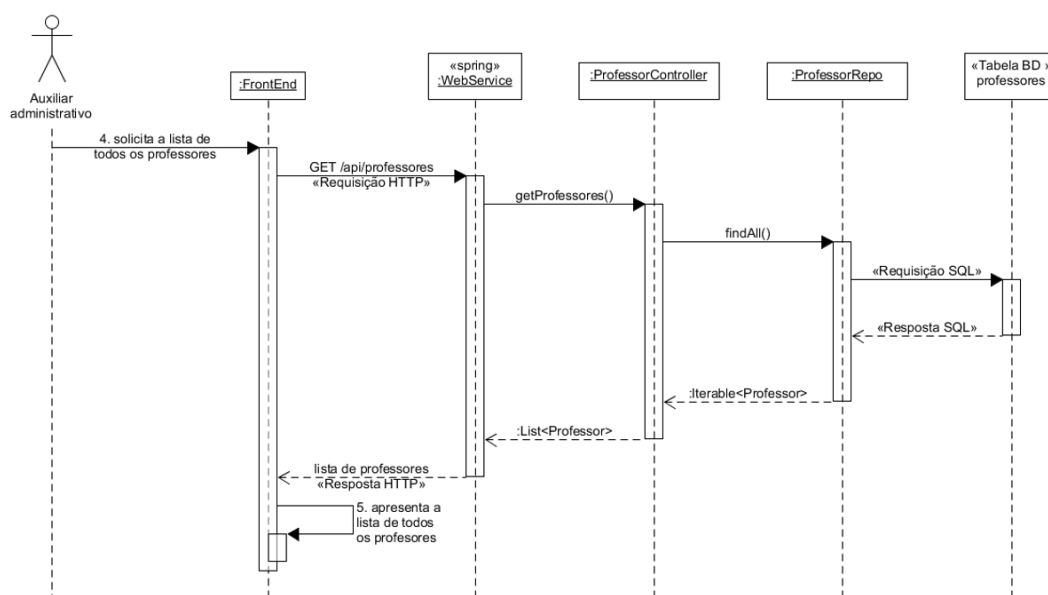
Imaginemos que esse Sistema Acadêmico do exemplo é uma aplicação web desenhada com os seguintes componentes:

- Front-end desenvolvido em HTML, CSS e Javascript.
- Web service REST.
- Tabelas em uma base de dados relacional.

Vamos também supor que o diagrama da Figura 5 represente as classes de projeto deste sistema.

A Figura 7 mostra um diagrama de sequência mais detalhado para descrever as interações que ocorrem na execução dos passos 4 e 5 do caso de uso.

Figura 7 – Diagrama de sequência mostrando as interações entre os componentes do sistema



Fonte: Elaborada pelo autor.

Fazendo isso para todas as descrições dos casos de uso, mostramos que o projeto (desenho, *design*, arquitetura) da solução é capaz de realizar os casos de uso.

REFERÊNCIAS

KALBACH, J. *Design de Navegação Web: otimizando a experiência do usuário*. Porto Alegre: Bookman, 2009.

LARMAN, C. *Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos e desenvolvimento interativo*. Porto Alegre: Bookman, 2011.