



**INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**
Rio de Janeiro

Disciplina

Modelagem de Domínio e
Conceitos de Orientação a Objetos

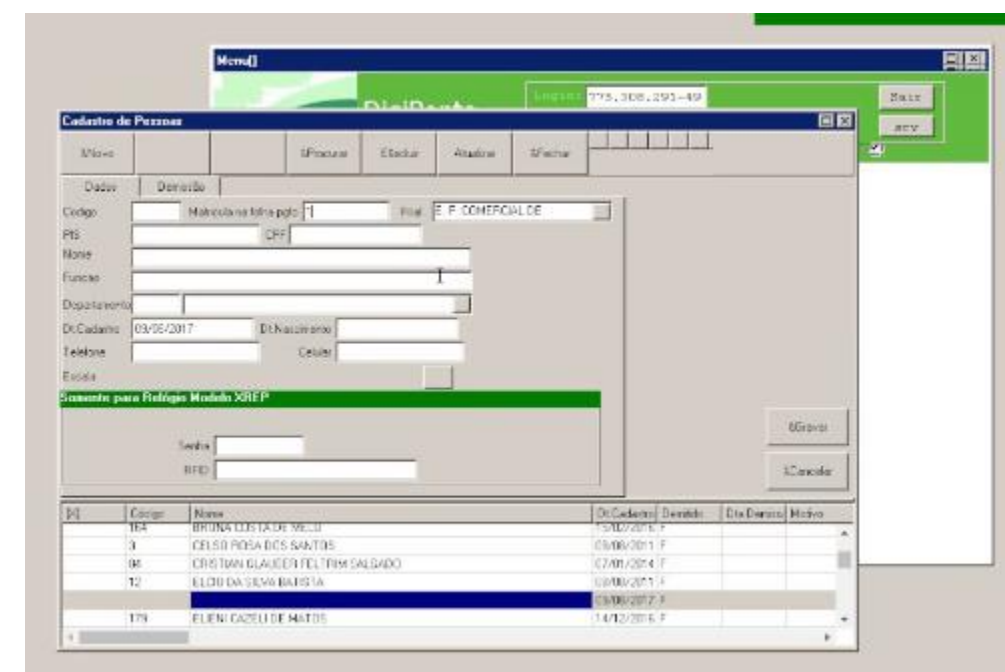
PROF. FERNANDO TAMBERLINI ALVES

Tipos de Aplicações

- ✓ Aplicações de Linhas de Comando (CLI)
- ✓ Aplicações Desktop
- ✓ Aplicações WEB
- ✓ Aplicações Mobile

```
root@main:~# ping -q namu.wiki
PING namu.wiki (153.149.99.38) 56(84) bytes of data:
^C
--- namu.wiki ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 0ms

root@main:~# curl https://namu.wiki && echo \n
Found. Redirecting to /w/NEBNB2X9NEBNACB4NECN9CB4NEFN82NA4:NEBNBCB9NEBNACB9B
root@main:~# pwd
/root
root@main:~# cd /var
root@main:/var# ls -la
total 120
drwxr-xr-x 15 root root 4096 9月 20 21:15 .
drwxr-xr-x 25 root root 4096 9月 19 13:15 ..
drwxr-xr-x 2 root root 4096 9月 27 22:02 cloning into 'git'
drwxr-xr-x 25 root root 4096 9月 11 11:11 remote: enumerating objects: 100% (2465/2465), done
drwxr-xr-x 2 root whoopsie 4096 9月 6 16:16 remote: compressing objects: 100% (1415/1415), done
drwxr-xr-x 88 root root 4096 9月 27 11:11 Receiving objects: 100% (50097/50097), 221.14 MiB | 1.86 MiB/s, done
drwxr-xr-x 2 root staff 4096 4月 10 7:07 Resolving deltas: 100% (30274/30274), done
drwxr-xr-x 1 root root 9 9月 27 21:07 updating files: 100% (4032/4032), done
drwxr-xr-x 16 root syslog 4096 9月 6 16:16
drwxr-xr-x 2 root mail 4096 2月 17 20:42 git
drwxr-xr-x 2 root whoopsie 4096 2月 17 21:07
drwxr-xr-x 2 root root 4096 2月 17 21:07 git status
drwxr-xr-x 1 root root 4 9月 27 21:07 Your branch is up to date with 'origin/main'.
drwxr-xr-x 2 root root 4096 9月 1 21:07 nothing to commit, working tree clean
drwxr-xr-x 431 root root 61440 9月 6 17:07
drwxr-xr-x 3 root root 4096 6月 17 20:17
root@main:/var# apt update
更新:1 http://dl.google.com/linux/chrome/deb
ビット:2 http://dl.google.com/linux/chrome/de
ビット:4 http://dl.winehq.org/wine-builds/deb
ビット:5 https://packagecloud.io/jonasgroeger
取得:6 https://download.01.org/gfx/ubuntu/16.
05 [archive.ubuntu.comへ接続しています] [see
```



App. Linhas de Comando

Descrição:

Aplicações de Linha de Comando são softwares que interagem com o usuário por meio de uma interface de texto, em que os comandos são inseridos diretamente no terminal ou prompt de comando.

Vantagens

- Performance: Geralmente, são mais leves e rápidas.
- Automação: Facilita a automação de tarefas por scripts.
- Recursos do sistema: Consome menos memória e processamento.
- Acessibilidade Remota: Fácil de usar em ambientes remotos via SSH.

Desvantagens:

- Curva de aprendizado: Requer que o usuário aprenda e memorize comandos específicos.
- Interface: Menos intuitivo e visualmente amigável para usuários finais.
- Funcionalidade limitada: Em geral, são menos interativas, o que pode limitar seu uso.

Aplicativos Desktop

Descrição:

São programas instalados diretamente no sistema operacional de um computador e executados localmente.

Vantagens

- Desempenho: Pode usar totalmente os recursos de hardware do dispositivo, o que é ideal para softwares pesados (edição de vídeo, design 3D, etc.).
- Funcionamento offline: A maioria dos aplicativos desktop pode ser usada sem conexão com a internet.
- Integração com o sistema operacional: Melhor acesso a APIs do sistema e integração com recursos do SO.

Desvantagens:

- Instalação e Atualização: Requer instalação manual e atualizações podem precisar ser baixadas e instaladas.
- Compatibilidade: Pode ser compatível apenas com sistemas operacionais específicos (Windows, macOS, Linux).
- Manutenção: É preciso instalar e manter em cada dispositivo.

Aplicativos WEB

Descrição:

São aplicações que funcionam através de navegadores e podem ser acessadas por meio de URLs, sem necessidade de instalação.

Vantagens

- Acessibilidade: Pode ser acessado em qualquer dispositivo com um navegador e conexão à internet.
- Atualização centralizada: Atualizações são feitas diretamente no servidor, o que significa que todos os usuários acessam a versão mais recente automaticamente.
- Compatibilidade cross-platform: Compatível com qualquer sistema operacional que tenha um navegador.

Desvantagens:

- Dependência de Internet: Em sua maioria, exigem uma conexão constante com a internet.
- Limitações de desempenho: Não tem o mesmo acesso ao hardware como os aplicativos desktop.
- Segurança: É suscetível a ameaças de segurança da web, como ataques de SQL injection ou XSS.

Aplicativos Mobile

Descrição:

Aplicações desenvolvidas para dispositivos móveis, como smartphones e tablets. Podem ser nativas (feitas para um SO específico) ou híbridas (multi-plataforma).

Vantagens

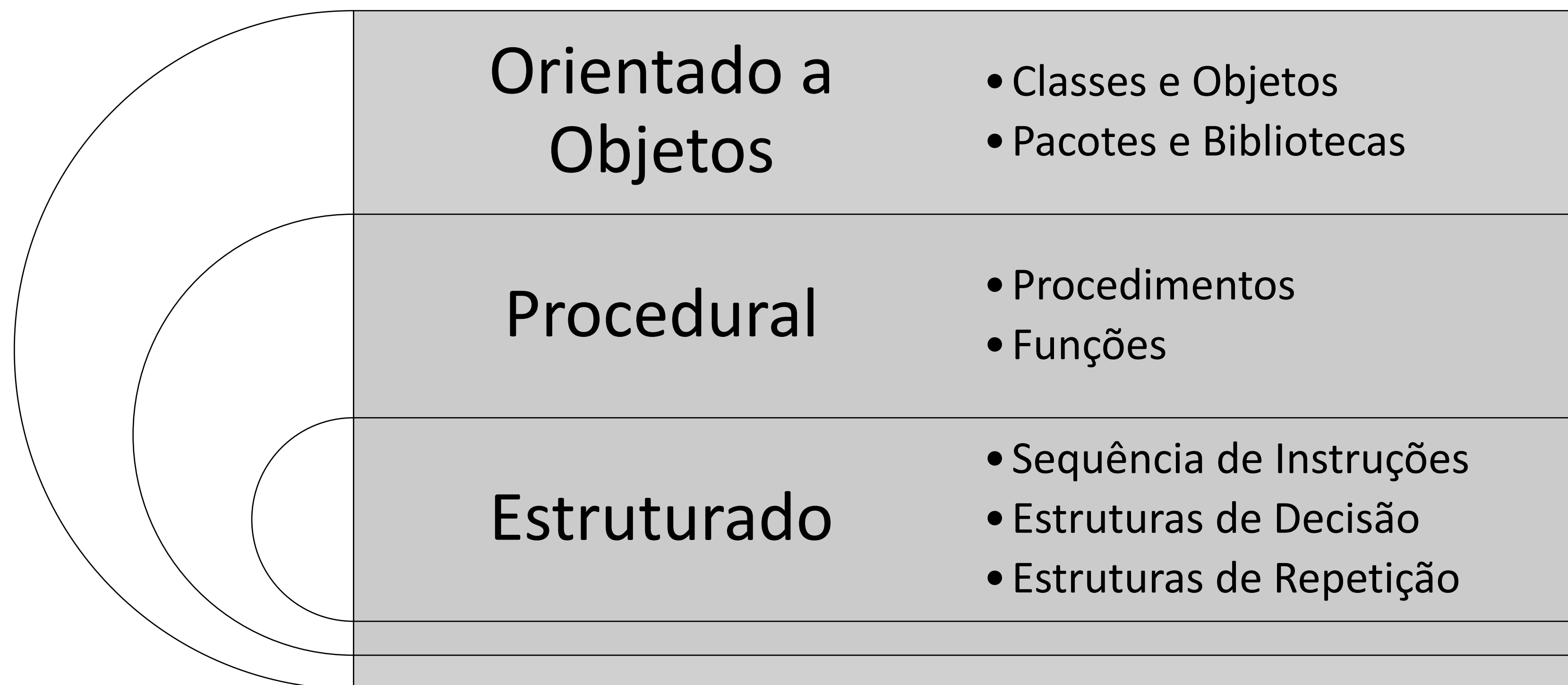
- **Acessibilidade e Mobilidade:** Permitem que os usuários utilizem serviços e recursos em qualquer lugar.
- **Notificações:** Oferece notificações que facilitam o engajamento e a comunicação com o usuário.
- **Acesso a hardware do dispositivo:** Pode utilizar recursos do dispositivo, como GPS, câmera, e sensores.

Desvantagens:

- **Compatibilidade:** Podem exigir desenvolvimento para diferentes sistemas operacionais (iOS, Android).
- **Limitações de Hardware:** Recursos como memória e processamento são mais limitados em dispositivos móveis.
- **Atualizações frequentes:** Necessita atualizações frequentes para acompanhar as mudanças nos sistemas móveis.

Paradigmas de Programação

- ✓ A orientação a objetos pode ser vista como um passo natural na evolução dos paradigmas



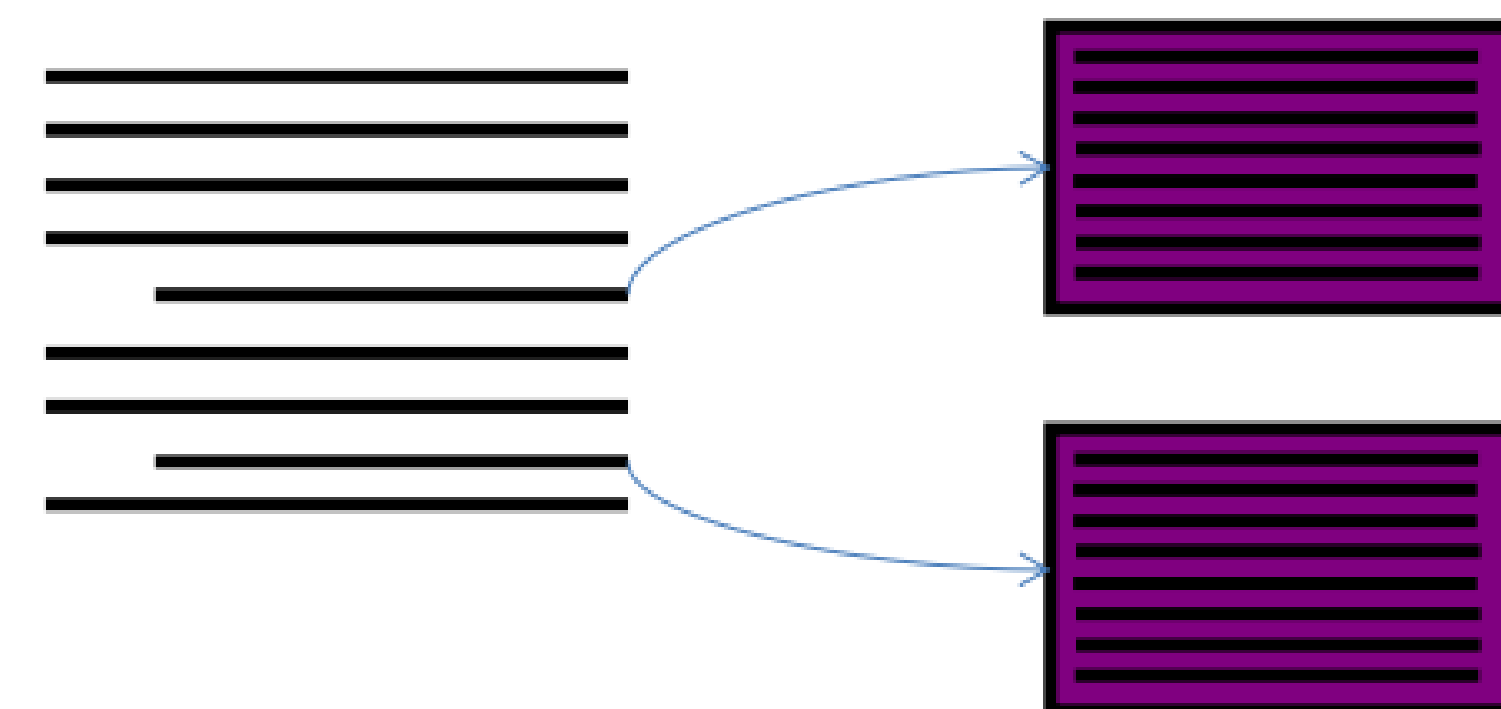
Paradigma Estruturado

- ✓ Só usa sequência, decisão e repetição
- ✓ Código mais fácil de ler, mas ainda difícil para sistemas grandes devido a repetição de código
- ✓ O que fazer se for necessário repetir uma sequência de linhas de código em diferentes locais?



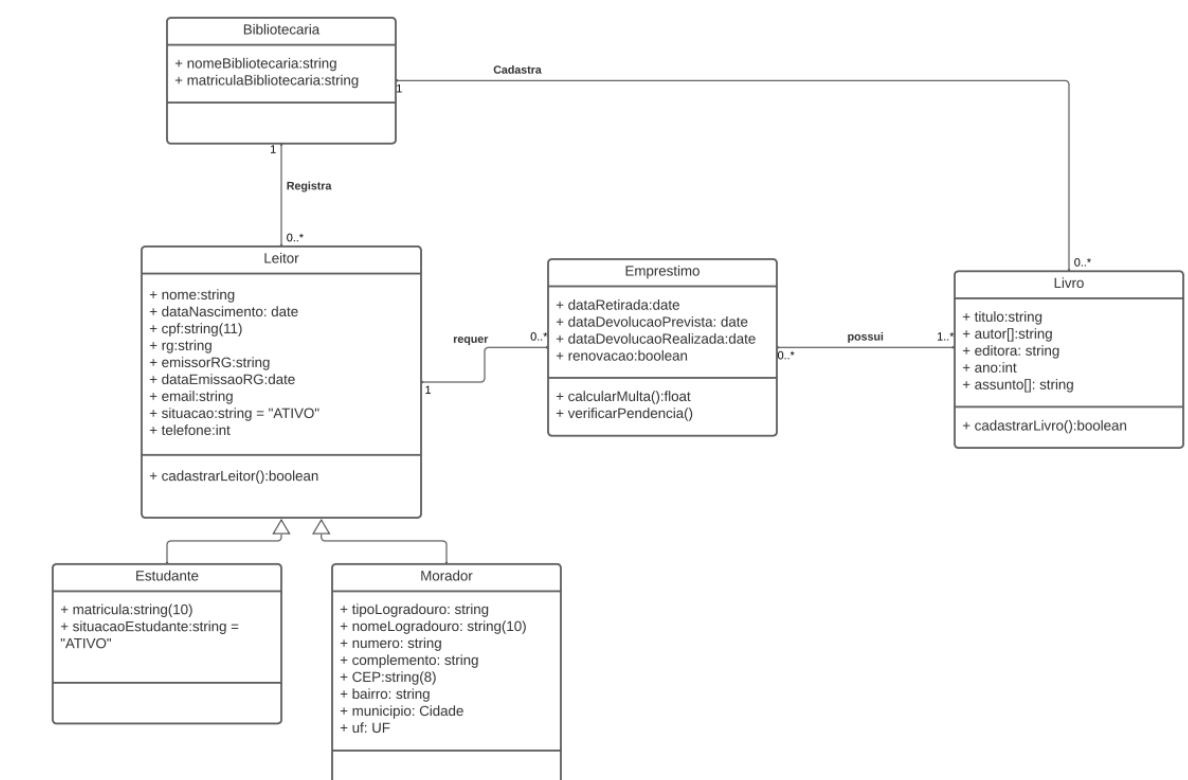
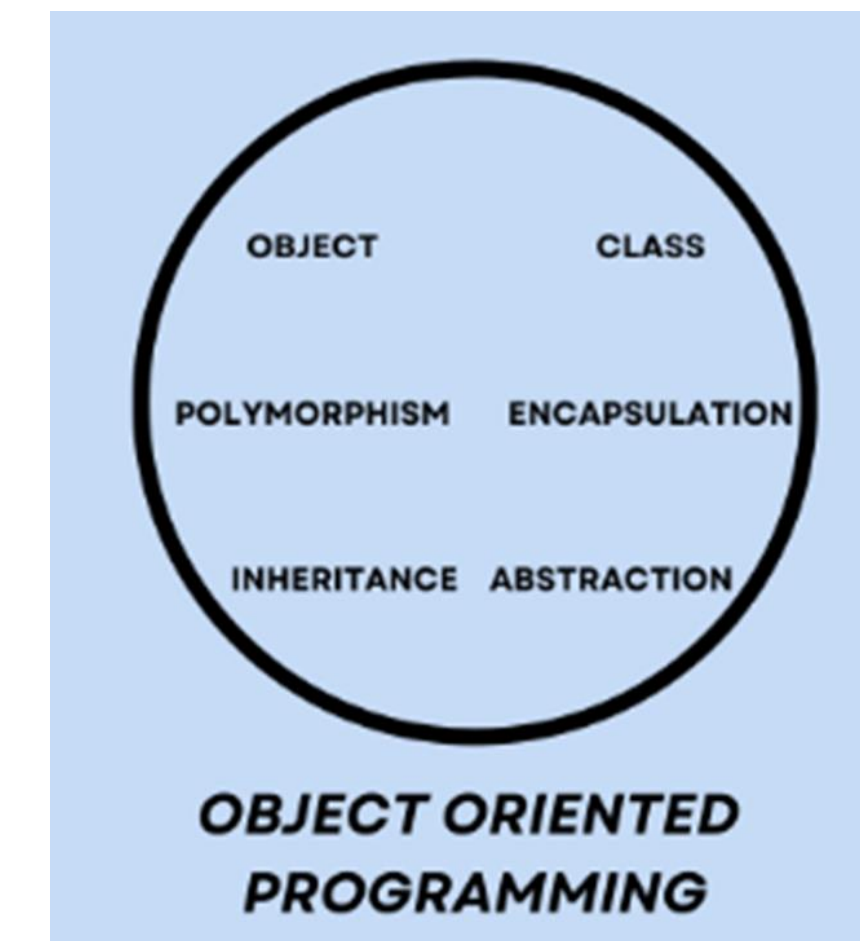
Paradigma Procedural

- ✓ Sinônimo: paradigma procedimental
- ✓ Uso de subprogramação
 - Agrupamento de código permitindo a criação de ações complexas
 - Atribuição de um nome para essas ações complexas
 - Chamada a essas ações complexas de qualquer ponto do programa
- ✓ Essas ações complexas são denominadas procedimentos, sub-rotinas ou funções



Paradigma Orientação a Objetos

- ✓ Classes e Objetos
 - Agrupamento de atributos e métodos
- ✓ Pacotes de Classes
 - Agrupamento de classes afins
 - Representam bibliotecas de apoio
- ✓ Facilitam o reuso e reaproveitamento de código
- ✓ Facilitam alta coesão e baixo acoplamento



Programação Orientada a Objetos

A Orientação a Objetos (OO) é um paradigma de programação que se baseia na ideia de organizar o código em torno de objetos, que são instâncias de classes. É uma abordagem poderosa e amplamente utilizada no desenvolvimento de software, pois ajuda a criar sistemas **mais organizados, modulares e reutilizáveis**.

A Orientação a Objetos promove uma abordagem mais natural e organizada para modelar o mundo real em código, tornando-o mais legível, manutenível e flexível. É amplamente utilizada em linguagens de programação como Java, C++, Python, C#, entre outras, e é aplicável a uma variedade de domínios de desenvolvimento de software, desde aplicações desktop até sistemas distribuídos e aplicativos web.

Programação Orientada a Objetos

Aqui estão os principais conceitos do paradigma de Orientação a Objetos:

Objeto: Um objeto é uma instância concreta de uma classe. Ele representa uma entidade do mundo real e possui atributos (propriedades) que descrevem seu estado e métodos que definem seu comportamento. Por exemplo, um objeto "Carro" pode ter atributos como "cor" e "velocidade" e métodos como "ligar" e "desligar".

Classe: Uma classe é um modelo ou uma definição abstrata que descreve as características comuns a um conjunto de objetos. Ela serve como um plano para a criação de objetos. A classe define os atributos e métodos que seus objetos terão em comum. Por exemplo, uma classe "Carro" define os atributos e métodos que todos os carros terão.

Paradigma Procedural

Organiza o programa em termos de algoritmos

Paradigma Orientado a Objetos

Organiza o programa em termos de objetos



- ✓ Podemos criar programa pensando em termos de objetos ao invés de algoritmos?
- ✓ O mundo é composto de objetos
 - Uma loja tem produtos, pedidos, estoque, etc.
 - Um restaurante tem mesas, garçons, comidas, bebidas, etc.
 - Uma universidade tem professores, alunos, disciplinas, etc.
 - Uma rodoviária tem ônibus, passageiros, bagagens, etc.
- ✓ E se criarmos programas basicamente criando objetos equivalentes ao mundo real, e fazendo com que esses objetos se comuniquem?

Exemplo: Agenda Eletrônica

Paradigma Procedimental

- Variáveis
 - Vetor de nomes
 - Vetor de endereços
 - Vetor de telefones
- Procedimentos
 - Listagem de todos os nomes
 - Listagem do endereço dado um nome
 - Listagem do telefone dado um nome
 - Adição de nome, endereço e telefone
 - Remoção de nome, endereço e telefone

Paradigma OO

- Objeto Agenda
 - Atributo : Vetor de Contatos
 - Métodos
 - Listagem de Contatos
 - Adição de um Contato
 - Remoção de um Contato
- Objeto Contato
 - Atributos: Nome, Endereço Telefone
 - Métodos
 - Exibição de nome, endereço e telefone
 - Edição de nome, endereço e telefone

P. Procedural x POO

Procedural

Foco no Processo

Reuso Limitado

Código Complexo e Longo

Dados Globais

Orientada a Objetos

Foco nos Objetos (Entidades e atores)

Facilita o reuso

Complexidade nas associações entre objetos

Dados Protegidos

P. Procedural x POO

**Exemplo: Aplicativo para
Restaurante**

**Exercitar a análise e projeto de
software sob ambos
paradigmas**

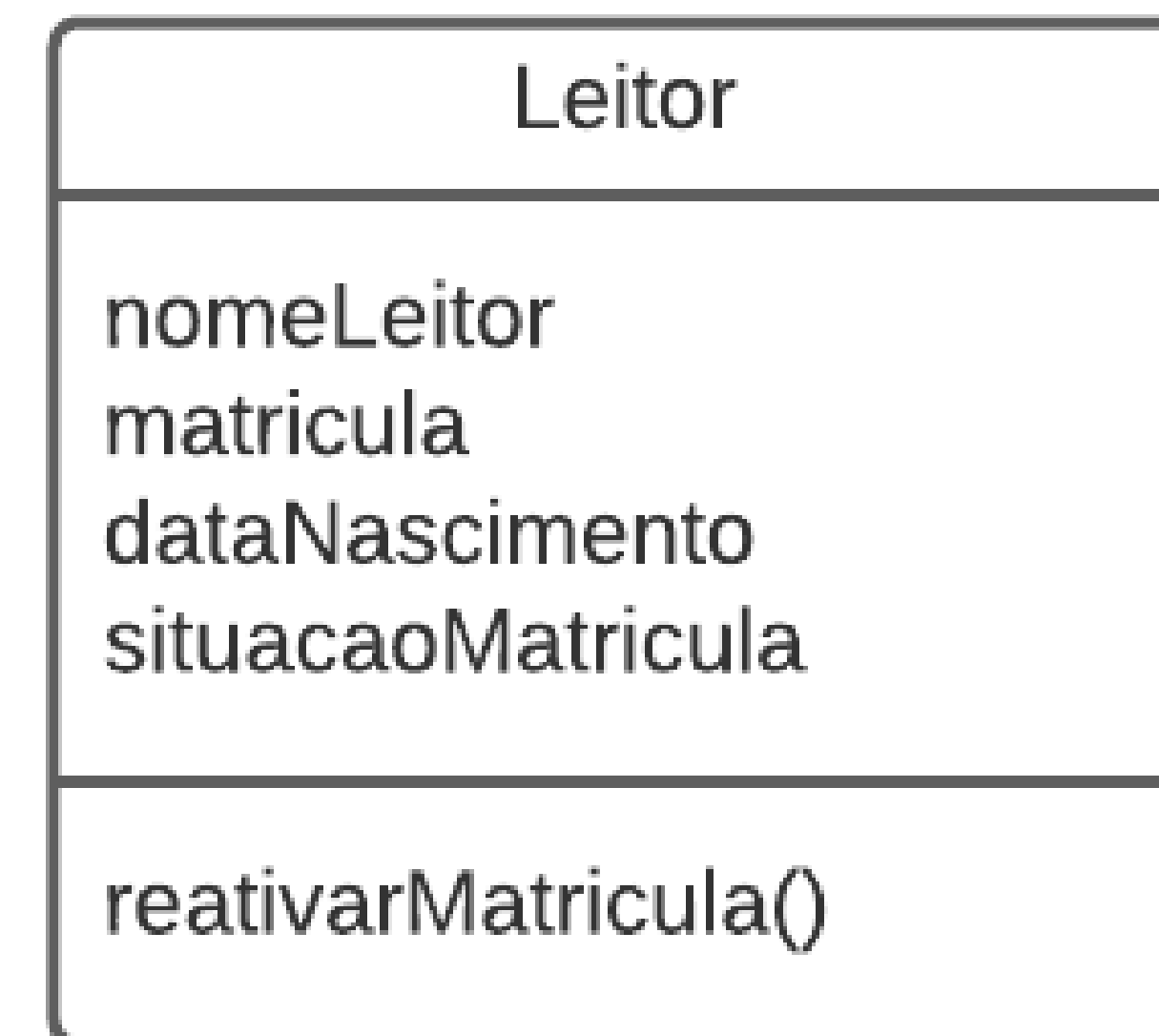
P. Procedural x PO



Modelo Conceitual x Diagrama de Classes

Classe

- É uma descrição de um conjunto de objetos que compartilham os mesmos atributos, operações, relacionamentos e semântica.
- É representadas por um retângulo que pode possuir até três divisões:
 - Nome da classe
 - Atributos da classe
 - Métodos da classe



- Existem várias notações para nomes de variáveis, métodos, classes, e outros elementos em programação. Cada uma dessas notações segue regras específicas sobre como as palavras devem ser formatadas e combinadas.
- Benefícios das Notações:
 - Legibilidade: Notações bem escolhidas facilitam a leitura e compreensão do código.
 - Convenção: Seguir uma convenção de nomenclatura ajuda na consistência do código e na colaboração em equipe.
 - Compatibilidade: Algumas linguagens ou ambientes têm convenções de nomenclatura específicas que devem ser seguidas para compatibilidade e boas práticas.
- Em resumo, a escolha da notação de nomenclatura adequada depende das convenções da linguagem de programação, ambiente de desenvolvimento e práticas recomendadas. O importante é manter consistência dentro do projeto ou equipe para garantir um código claro e fácil de entender.
- Acesso a página do professor para conhecer as principais notações.

Notações



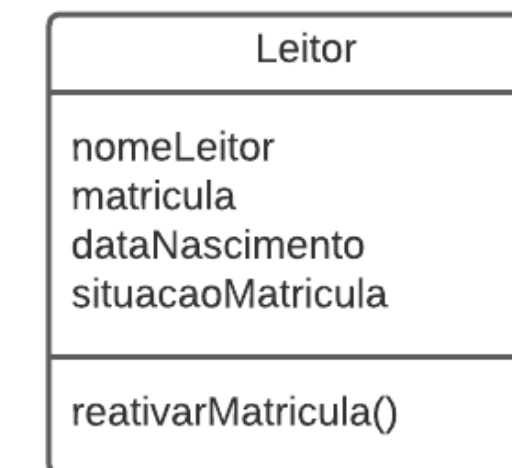
Qual é a diferença entre Classe e Objeto

Classe

- é definição do tipo;
- representa um conjunto de objetos de mesmo tipo;
- Classe = {obj1, obj2, obj3, ..., objN}

Objeto

- cada instância derivada da classe;
- é um elemento do conjunto representado pela classe



nomeLeitor: Maria
matricula:20230001
dataNascimento: 15/10/2017
Situacao:Ativo

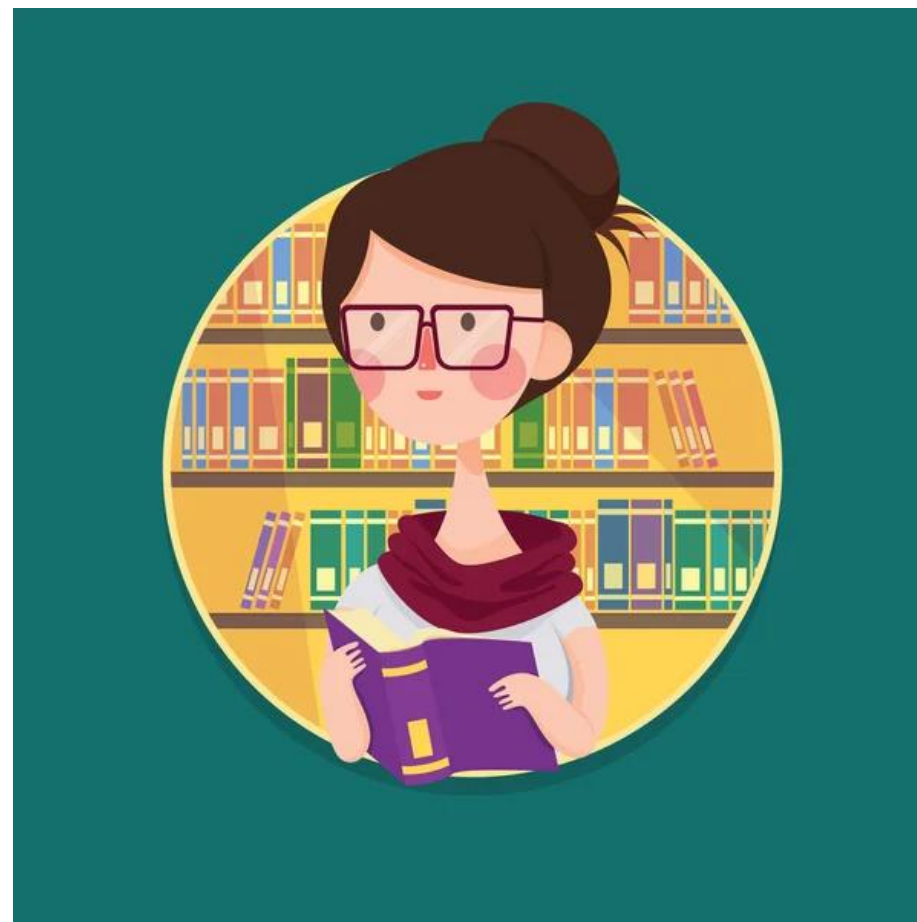


nomeLeitor: João
matricula:20230002
dataNascimento: 10/08/2017
Situacao:Ativo



nomeLeitor: Lia
matricula:20230003
dataNascimento: 20/12/2017
Situacao:Ativo

Exemplo Prático – SI Biblioteca



Antônia é a nova estagiária de Informática da Secretária de Educação de um pequeno município do interior de Minas Gerais. Infelizmente o município não tem orçamento para comprar uma solução de mercado para gestão da Biblioteca, sendo assim o prefeito quer que seja feito um sistema para automatizar as atividades rotineiras da única biblioteca da cidade.

O prefeito comprou vários livros, contudo a conduta rigorosa da bibliotecária Sra. Clotilde tem afastado os leitores da biblioteca. O prefeito quer que o sistema permita qualquer pessoa da cidade pesquise os livros disponíveis pela internet podendo ser pesquisado pelo Nome do Autor, Título ou Assunto. Ele também quer que o sistema agilize o empréstimo e a verificação de quem está com o empréstimo atrasado.

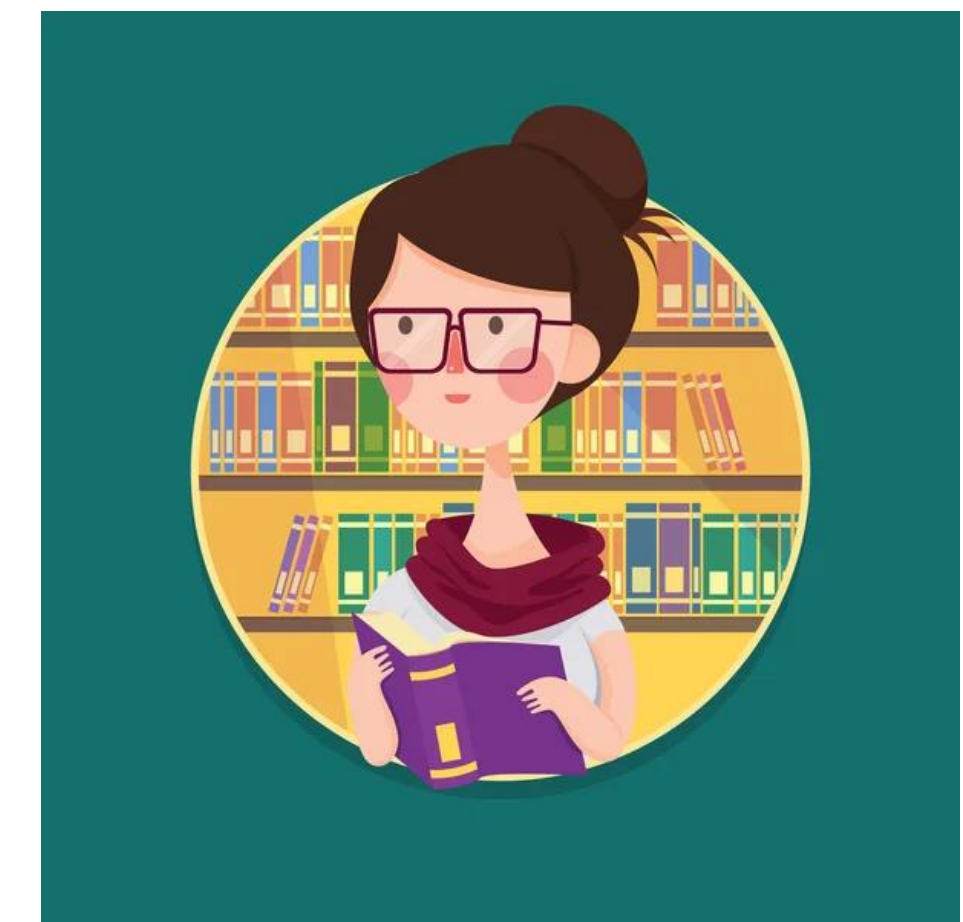
Diante desta demanda, o Secretário de Educação solicitou que a nova estagiária Antônia fosse na biblioteca da cidade para conversar com a bibliotecária Sra. Clotilde para levantar os requisitos do novo sistema.

Exemplo Prático – SI Biblioteca

Assim que a estagiária chegou na biblioteca, a Sra. Clotilde a interpelou com os seguintes dizeres:

“Estou cansada dessa gentinha que vem aqui coloca as mãos sujas nos meus livros fora as pessoas que os devolvem com rabiscos. Aqui nessa biblioteca só eu posso pegar nos livros antes do empréstimo. Senão eles pegam, folheiam, sujam e coloca na estante errada mesmo eu pedindo para eles colocarem no carrinho perto do balcão.

Você é a menina que veio a pedido do prefeito, certo? Bom, acho totalmente desnecessário um sistema já que eu dou conta. Mas vou explicar as regras daqui. Para pegar livro emprestado tem que ser aluno matriculado nas escolas da cidade ou morador da cidade. Podem ficar com até três livros emprestados e o tempo do empréstimo é de 15 (quinze) dias. Só pode renovar o empréstimo uma única vez. Antes de pedir empréstimo, o estudante ou morador deve estar cadastrado aqui. Para o cadastro é necessário a carteirinha de estudante ou comprovante de residência. Só estudante com matrícula ativa na escola pode pegar livro.”



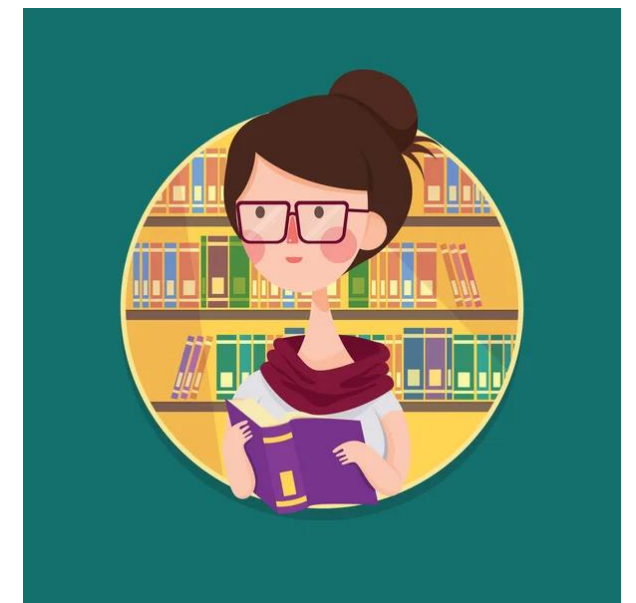
Exemplo Prático – SI Biblioteca

A estagiária Antônia anotou tudo que a Sra. Clotilde falou e no fim ela perguntou:

- Você poderia descrever em detalhes as principais atividades da biblioteca?

A Sra. Clotilde respondeu:

- Ora, você nunca esteve numa biblioteca antes? Inicialmente, o leitor pesquisa o livro em uma das 3 (três) gavetas de ficha (Autor, Título e Assunto). Ele anota os dados da ficha do livro e traz aqui no balcão. Eu verifico se o livro está emprestado. Caso o livro esteja aqui, eu peço a identificação do leitor e verifico se o cadastro dele está ok. Se tiver tudo certo, eu pego livro na estante e carimbo a data de devolução na ficha na contracapa. Falo bem alto que só pode renovar por uma vez e se devolver com atraso paga uma multa de R\$ 0,50 por dia. Pouca gente paga a multa, mas se tiver devendo não pega mais empréstimo. Eu verifico todos os dias os livros que estão atrasados e ligo para as pessoas cobrando a devolução. Se não tiver sido renovado ainda, eu dou a opção de renovar por uma única vez.



SI Biblioteca – Atividade Prática

A partir da entrevista com a bibliotecária e utilizando o poder da abstração, faça as seguintes atividades:



- 1) Identifique as entidades envolvidas (atores e objetos) no contexto (domínio) da SI Biblioteca;
- 2) Identifique os atributos (campos) das entidades do item 1;
- 3) Identifique as ações (transações) do SI Biblioteca.

Como identificar esses elementos (entidades)?

- Alguns são bem óbvios (Leitor e Livro), mas outros estão implícitos que requerem maior maturidade daquele que está fazendo o levantamento de requisitos.
- Após enumerar as entidades e ações óbvias, pode adotar seguinte estratégia:
 1. Listar as etapas do fluxo principal do sistema;
 2. Passo 1 – Isole os substantivos;
 3. Passo 2 – Analise os substantivos;
 4. Passo 3 – Isole e analise os verbos;
 5. Passo 4 – Para cada conceito verifique se ele é composto com outras partes do sistema.



SI Biblioteca – Atividade Prática

1. O Leitor chega ao balcão de atendimento da biblioteca e diz ao atendente que deseja emprestar um ou mais livros da biblioteca.
2. O Atendente seleciona a opção para adicionar um novo empréstimo.
3. O Atendente solicita ao leitor sua identidade, seja de estudante ou morador.
4. O Atendente informa ao sistema a identificação do leitor.
5. O Sistema exibe o nome do leitor e sua situação.
6. O Atendente solicita os livros a serem emprestados.
7. Para cada um deles, informa ao sistema o código de identificação do livro.
8. O Sistema informa a data de devolução de cada livro.

PASSO 1 – Isolar os substantivos

- Leitor
- Balcão
- Biblioteca
- Atendente
- Livros
- Opção
- Empréstimo
- Carteirinha
- Estudante
- Morador
- Sistema
- Identificação de leitor
- Nome do leitor
- Código de identificação do livro
- Data de devolução

Passo 2: Análise dos substantivos

Para cada substantivo, verifique se é relacionado a assuntos importantes no domínio do sistema.

Descarte aqueles que:

- fogem do escopo do sistema
- são similares a outros conceitos já identificados
- são propriedades de outros substantivos

Lembre-se:

Conceitos relevantes são aqueles que se referem a entidades que têm que ser lembradas pelo sistema: fazem algo, sabem algo, conhecem algo, ...

PASSO 2 – Analise os substantivos

- Leitor
 - ~~Balcão~~
 - Biblioteca
 - Atendente
 - Livros
 - ~~Opção~~
 - Empréstimo
 - ~~Carteirinha~~
- Estudante
 - Morador
 - ~~Sistema~~
 - ~~Identificação de leitor~~
 - ~~Nome do leitor~~
 - ~~Código de identificação do livro~~
 - ~~Data de devolução~~

Passo 3: Isole e analise os verbos

Isole os verbos que poderiam ser transformados em substantivos (possivelmente com a ajuda de outras palavras).

Concentre-se nos verbos que representam ações de interesse para o sistema, ou seja, aqueles relacionados a eventos e transações que possuem informações importantes e que devem ser lembradas pelo sistema.

SI Biblioteca – Atividade Prática

1. O Leitor chega ao balcão de atendimento da biblioteca e diz ao atendente que deseja emprestar um ou mais livros da biblioteca.
2. O Atendente seleciona a opção para adicionar um novo empréstimo.
3. O Atendente solicita ao leitor sua identidade, seja de estudante ou morador.
4. O Atendente informa ao sistema a identificação do leitor.
5. O Sistema exibe o nome do leitor e sua situação.
6. O Atendente solicita os livros a serem emprestados.
7. Para cada um deles, informa ao sistema o código de identificação do livro.
8. O Sistema informa a data de devolução de cada livro.

SI Biblioteca – Atividade Prática

PASSO 3 – Isole e analise os verbos

- ~~Emprestar~~
- ~~Adicionar~~
- ~~Informar~~

Passo 4: Para cada conceito verifique se ele é composto com outras partes do sistema

Para cada candidato a conceito, verifique se ele é composto de outras partes que sejam de interesse do sistema, mesmo que essas não apareçam explicitamente no texto.

Exemplo:

- Empréstimo normalmente refere-se a vários livros emprestados em uma mesma ocasião por um mesmo leitor.
- Linha do Empréstimo refere-se a cada livro emprestado.

(Obs. poderia ser também: Item do Empréstimo)

Outras possíveis entidades:

- Objetos físicos ou tangíveis:
 - Livro, Leitor
- Especificação de Projetos ou descrição de coisas:
 - EspecificacaoDeLivro, CategoriaDeLivro
- Lugares:
 - Biblioteca, SalaDeAula
- Transações:
 - Emprestimo, Reserva
- Linha de Itens de Transações:
 - LinhaDoEmprestimo

Outras possíveis entidades:

- Papéis desempenhados por pessoas:
 - Atendente, ChefeDeBiblioteca, Usuario
- Contêineres de outras coisas:
 - Estante, Armario, Sala
- Coisas em um contêiner:
 - CopiaDeLivro, Revista
- Catálogos:
 - CatalogoDeLivros, CatalogoDeRevistas
- etc, etc..

Como identificar atributos?

Substantivos podem ser candidatos a atributos de conceitos.



Cautela:

- não torne o modelo conceitual muito complexo desnecessariamente.
- limite-se a adicionar
 - atributos importantes para compreender o conceito
 - atributos que serão importantes para o futuro projeto do sistema

SI Biblioteca – Atividade Prática



1. Fazer cadastro no site do [LucidChart](#);
2. Criar um novo diagrama utilizando a biblioteca de formas “UML – Diagrama de Classes”;
3. Criar as classes do SI Biblioteca (cada entidade vai virar uma classe);
4. Criar os atributos identificados de cada classe;
5. Utilizar a convenção de nomenclatura para Javascript ([veja este artigo](#))

Dúvidas?



Dúvidas?



fernando.alves@ifrj.edu.br



ftamberlini.dev.br