# Projeto Integrador I – Adam Yoshida
# 03/09/2022 – Requisitos e Diagrama de Classes UML
# Atividades:

**Pesquisas sobre:**

- Definição de requisitos do usuário, do sistema, funcionais e não funcionais
- BDD (o que é, finalidade, quando usar, exemplo)
- UML (o que é, para que usar, quais são os diagramas)
- Modelo de Domínio

**Elaborar para o projeto em execução**

- Requisitos funcionais utilizando BDD
- Diagrama de classes: Modelo de Domínio

# Definições:

## O que são requisitos?

R: São descrições dos serviços fornecidos pelo sistema e das restrições sobre estes serviços.

## Requisitos do usuário:

R: Requisitos de usuário são declarações, em uma linguagem natural somada a diagramas, dos serviços que se espera que o sistema forneça para os usuários e das limitações sob as quais ele deve operar. Esses requisitos podem variar de declarações amplas das características necessárias do sistema até descrições precisas e detalhadas da sua funcionalidade.

## Requisitos do sistema:

R: Os requisitos de sistema são descrições mais detalhadas das funções, dos serviços e das restrições operacionais do sistema de software. O documento de requisitos de sistema (chamado às vezes de especificação funcional) deve definir exatamente o que deve ser implementado. Pode fazer parte do contrato entre o adquirente do sistema e os desenvolvedores de software.

## Requisitos funcionais:

R: Os requisitos funcionais de um sistema descrevem o que ele deve fazer e dependem do tipo de software que está sendo desenvolvido, dos usuários esperados para o software e da abordagem geral adotada pela organização ao escrever os requisitos. Quando são apresentados como requisitos de usuário, os requisitos funcionais devem ser escritos de modo compreensível para os usuários e gerentes do sistema. Os requisitos funcionais do sistema expandem os requisitos de usuário e são escritos para os desenvolvedores. Requisitos funcionais devem descrever em detalhes as funções do sistema, suas entradas, saídas e exceções.

## Requisitos não funcionais:

R: Os requisitos não funcionais, como o nome sugere, são aqueles que não possuem relação direta com os serviços específicos fornecidos pelo sistema aos seus usuários. Esses requisitos não funcionais normalmente especificam ou restringem as características do sistema como um todo. Eles podem estar relacionados a propriedades emergentes do sistema, como confiabilidade, tempo de resposta e uso da memória. Por outro lado, podem definir restrições à implementação do sistema, como a capacidade dos dispositivos de E/S ou as representações dos dados utilizados nas interfaces com outros sistemas.

# BDD

**Definição:** Behaviour Driven Development é técnica de desenvolvimento ágil que visa integrar regras de negócios com linguagem de programação, focando o comportamento do software.

O foco em BDD é a linguagem e as interações usadas no **processo de desenvolvimento de software**. Desenvolvedores que se beneficiam destas técnicas escrevem os testes em sua língua nativa em combinação com a linguagem ubíqua (*Ubiquitous Language*). Isso permite que eles foquem em por que o código deve ser criado, ao invés de detalhes técnicos, e ainda possibilita uma comunicação eficiente entre as equipes de desenvolvimento e **testes**.

**Linguagem Ubíqua:** é uma linguagem estruturada em torno do modelo de domínio e usada por todos os membros da equipe para conectar todas as suas atividades com o software. Numa equipe de desenvolvedores são: os jargões técnicos, terminologias das discussões do dia a dia ou uma linguagem incomum para pessoas de outros departamentos.

## Quando utilizar BDD?

O BDD funciona na prática como uma importante ferramenta de testagem e de início de atualização para softwares. O objetivo é melhorar as funcionalidades, escrevendo códigos que vão ao encontro das necessidades dos clientes.

## Exemplo de uso

```
Funcionalidade: Logar no Outlook 360
Eu como usuário
Quero fazer login na minha conta do Outlook 360
Para ter acesso ao meu correio eletrônico


Cenário: Logar com sucesso
    Dado que estou na página de login do Outlook 360
    Quando preencho o campo email com um email válido
    E clico em avançar
    E preencho o campo senha com uma senha válida
    E clico em entrar
    Então sou direcionado para minha caixa de entrada do Outlook 360

```

Figura 1. Exemplo de BDD mais detalhista

```
Funcionalidade: Logar no Outlook 360
Eu como usuário
Quero fazer login na minha conta do Outlook 360
Para ter acesso ao meu correio eletrônico


Cenário: Logar com sucesso
    Dado que estou na página de login do Outlook 360
    Quando me autentico com um usuário válido
    Então sou direcionado para minha caixa de entrada do Outlook 360

```

Figura 2. Exemplo de BDD mais generalista

# UML

**Definição:** Basicamente, UML (Unified Modeling Language) é uma linguagem de notação (um jeito de escrever, ilustrar, comunicar) para uso em projetos de sistemas. Esta linguagem é expressa através de diagramas. Cada diagrama é composto por elementos (formas gráficas usadas para os desenhos) que possuem relação entre si.

## Quando usar os diagramas UML?

Os diagramas UML devem ser usados em diversas situações, por exemplo:

- Quando é preciso oferecer uma visão padronizada do projeto e de suas especificações para todas as pessoas envolvidas no desenvolvimento;
- Para documentar e visualizar o funcionamento do software;
- Comunicar os protocolos de interfaces que serão consumidas por outros sistemas;
- Auxiliar a fase inicial de especificação dos requisitos do sistema;
- Quando é preciso documentar os desejos de algum cliente sobre as funcionalidades do software.

## Quais são os diagramas?

Eles são divididos em duas categorias: os **diagramas estruturais** e os **comportamentais**. Cada um deles é usado para especificar, documentar, modelar e visualizar aspectos específicos de uma aplicação.

## Diagramas UML estruturais

- Diagrama de classes;
- Diagrama de pacote;
- Diagrama de objeto;
- Diagrama de componentes;
- Diagrama de estrutura composta;
- Diagrama de implantação;

## Diagramas UML comportamentais

- Diagrama de atividades;
- Diagrama de sequência;
- Diagrama de casos de uso;
- Diagrama de estado;
- Diagrama de comunicação;
- Diagrama de visão geral da interação;
- Diagrama de tempo.

# Modelo de Domínio

Definição de abstrações de domínio, como políticas, procedimentos, objetos, relacionamentos e eventos. Ele serve como uma base de conhecimento sobre alguma área.

É uma representação de classes conceituais do mundo real, não de componentes de software

- Não descreve classes de software, nem objetos com responsabilidades
- É uma representação visual de classes conceituais ou objetos do mundo real em um domínio de interesse

Usando a notação UML, o modelo de domínio é ilustrado com um conjunto de diagramas de classes nos quais não são definidas operações. Ele deve mostrar:

- Objetos do domínio ou classes conceituais
- Associações entre classes conceituais
- Atributos de classes conceituais

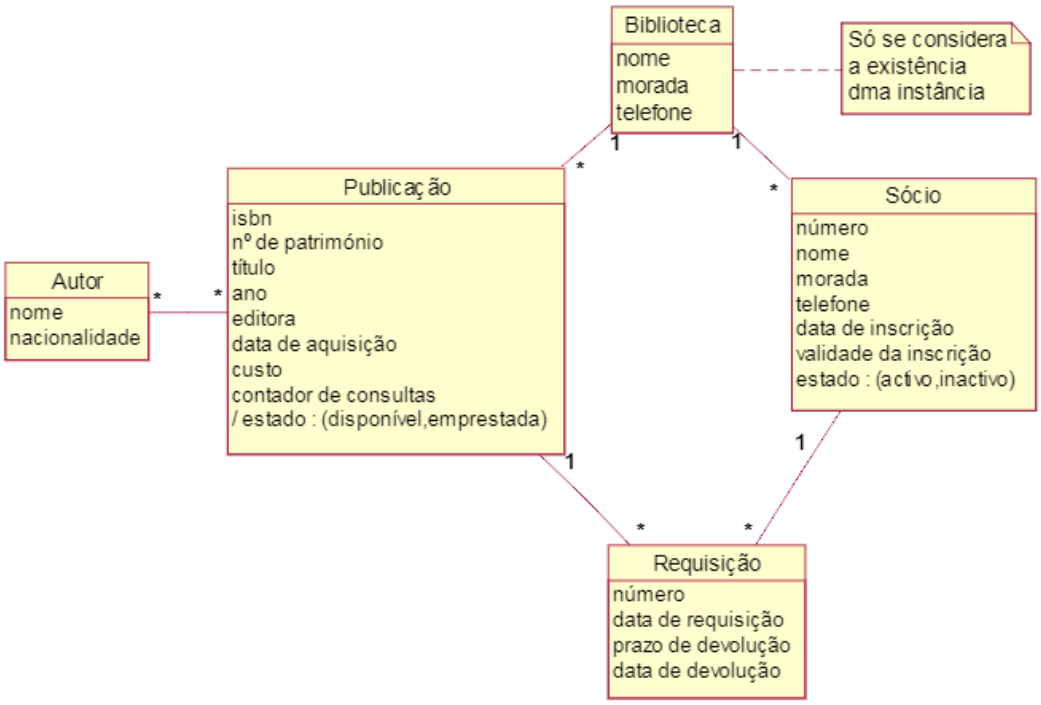


Figura 3. Exemplo de modelo de domínio