

Aula 00

Introdução a disciplina

Rogério Pereira Junior

rogeriopereirajunior@gmail.com

1 de abril de 2024

Agenda

- 1 Informações da disciplina
- 2 Conteúdos
- 3 Informática e Computação
- 4 Programação

Informações da disciplina

Informações Gerais

Programação Java



- **Professor:** Rogério Pereira Junior
- **Contato:**
rogerio.junior@prof.sc.senac.br

Indicadores

- 1 Planejar o desenvolvimento do software conforme análise de requisitos.
- 2 Criar algoritmos utilizando padrões de lógica de programação.
- 3 Construir aplicações, a partir da análise de requisitos, de acordo com as funcionalidades do sistema e linguagem de programação selecionada.
- 4 Integrar banco de dados, de acordo com a aplicação web e requisitos do projeto de software.

Elementos da Competência

1 Conhecimentos

- Lógica de Programação: conceito de algoritmo; algoritmos estruturados.
- Variáveis e constantes: expressões e operadores; estrutura condicional simples e composta; estrutura de repetição; vetores e matrizes.
- Desenvolvimento colaborativo: conceito e ferramentas para o desenvolvimento de sistemas.
- Linguagem de programação orientada a objetos: classes, atributos, métodos, propriedades, herança, polimorfismo, encapsulamento e ambientes de programação (IDE).

2 Habilidades

- Efetuar as quatro operações matemáticas básicas.
- Utilizar termos técnicos nas rotinas de trabalho.
- Testar algoritmos desenvolvidos.

■ Utilizar termos técnicos nas rotinas de trabalho

Método de avaliação

- Listas de exercícios individuais
- Teste
- Projeto prático em equipe

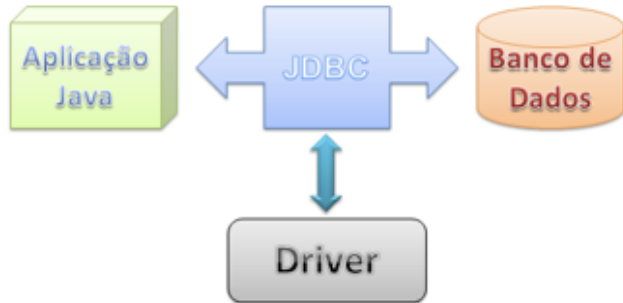
Conteúdos

Cronograma

- Introdução a computação com Java
- Tipos de dados
- Estruturas condicionais
- Estruturas de repetição
- Estrutura de dados simples
- Tópicos adicionais (datas, String, Leitura e escrita de arquivos)
- Funções e módulos
- Conceitos básicos de orientação a objetos
- Banco de dados e linguagem de consulta
- Banco de dados com JAVA
- Outros tipos de estrutura de dados
- Desktop

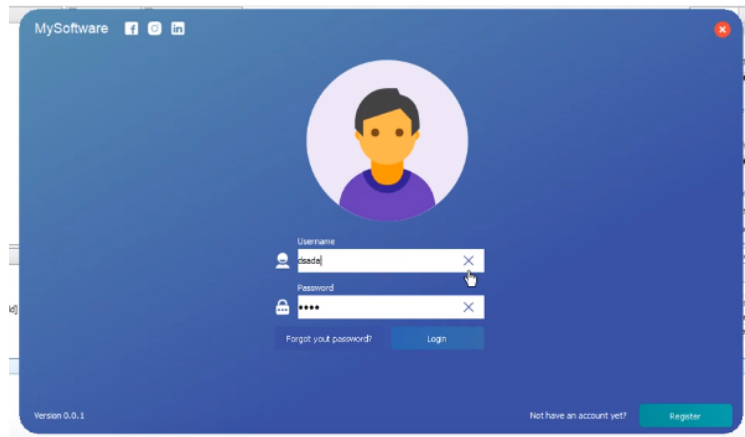
Objetivo inicial

- Programação básica com JAVA
- Banco de dados Integração Java e banco de dados



Objetivo final

- Criar uma aplicação desktop utilizando Java e banco de dados



Informática e Computação

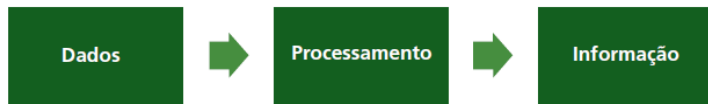
Informações Gerais

■ Atualizações a serem feitas

- 1 Processador
- 2 Memória
- 3 dispositivos de entrada
- 4 Sistem operacional
- 5 Front END e Back-END
- 6 processamento de dados
- 7 Exercicios de lógica: Livro de algoritmos

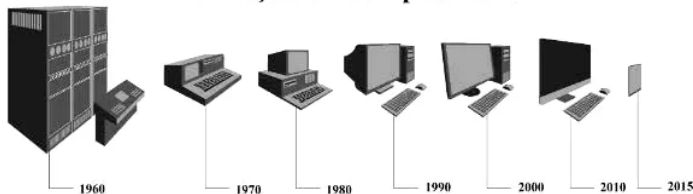
Informática

- O que é informática?



- Qual o objetivo de um computador?
- Quais suas vantagens?

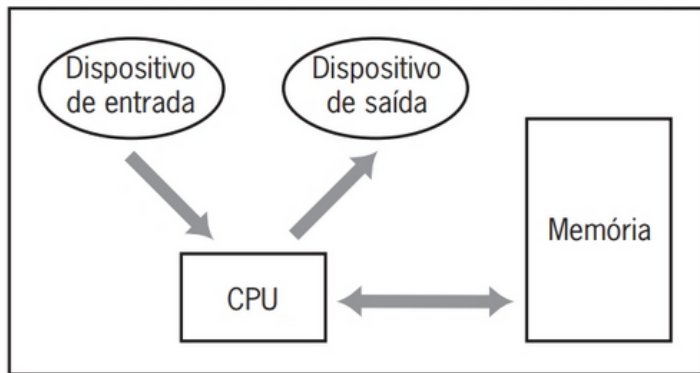
Evolução dos Computadores



- O computador é uma máquina inteligente?

Hardware

- Um sistema computacional moderno consiste em hardware e software.
- O **Hardware** consiste nos dispositivos físicos necessários para executar algoritmos.



Software

- Software é o conjunto desses algoritmos, representados como programas, em particular linguagens de programação.



- Software de sistema: Sistema operacional
- Software de aplicativos: programa projetado para uma tarefa específica, como editar um documento ou exibir uma página da Web.

Hardware e Software

Hardware



Software



Processador

- É a parte do computador responsável pelo processamento das instruções matemáticas/lógicas e programas carregados na memória.



- Uma espécie de cérebro da máquina realizando cálculos, fazendo tarefas, transformações e manipulações de dados.

Memória Principal

- A memória é a parte do computador que permitem o armazenamento de dados.
- Divida em primaria e secundária.



- Memória Principal: É um tipo de memória eletrônica que depende de uma fonte de energia
- Rápido acesso e que armazena os dados /informações

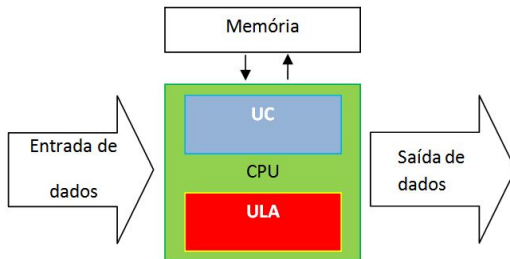
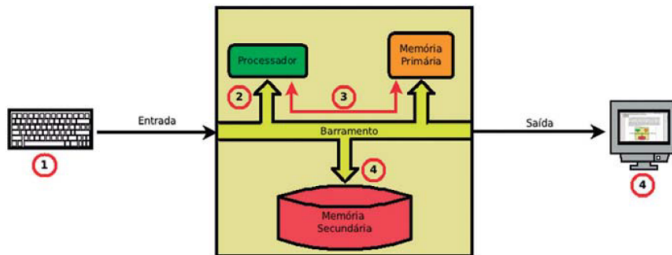
Memória Secundária

- NÃO dependem de uma fonte de energia para manter os dados armazenados
- Muito mais lentas - utilizam mecanismos mecânicos e elétricos (motores e eletroímãs)



- Resolve o problema da perda de dados causado pela Memória Principal
- São os componentes responsáveis por armazenar dados permanentemente.
- Mantêm os dados guardados mesmo após o desligamento do computador.

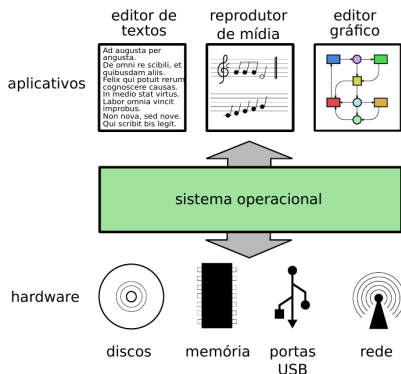
Hardware



QUEM GERENCIA ISSO TUDO?

Sistema Operacional

- É a camada de software que opera entre o hardware e os aplicativos.
- Conjunto de rotinas executados pelo processador, de forma semelhante aos programas dos usuários
- Principal função: Controlar o funcionamento do computador



Programação

Programação

- Programação é solucionar um problema e ensinar o computador a resolvê-lo para você
- Como se aprende programação?
 - Estudando técnicas e exercitando-as muito a partir da criação de pequenos programas
 - É quase como o processo de alfabetização de uma criança

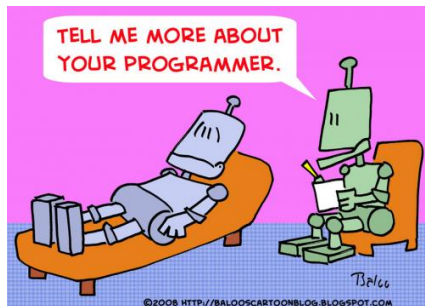


- Constrói-se sistemas que estão presentes em praticamente todas as áreas

Programação

O que é programar

Programar é escrever instruções para que um computador as execute



- As instruções podem solicitar ao computador que exiba alguma mensagem, colete dados ou lei ao conteúdo de um arquivo e salve no banco de dados

Programação

- Passado: muito difícil de programar - linguagens de baixo nível



- A criação de linguagens de auto nível (cuja leitura se parece mais com a do idioma inglês) facilitou a programação

Programação

Baixo Nível

```
global _start
section .text
_start:
    mov    rax, 1
    mov    rdi, 1
    mov    rsi, message
    mov    rdx, 13
    syscall
    ; exit(0)
    mov    eax, 60
    xor    rdi, rdi
    syscall
message:
    db     "Hello, World!", 10
```

Alto Nível

```
print("Hello, World!")
```

Programação

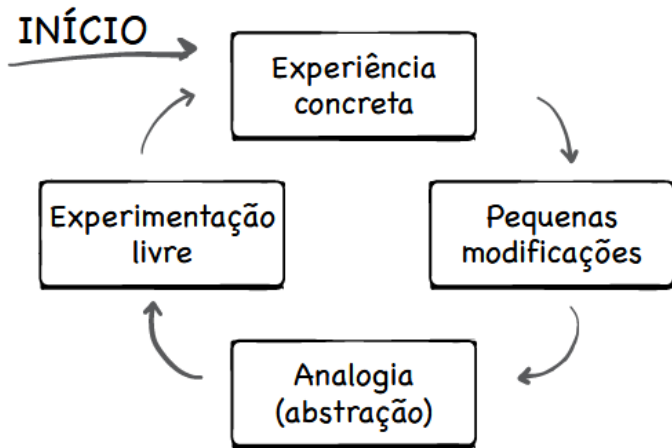
Resumindo

- **Código:** As instruções que os programadores escrevem para um computador executar
- **Programação:** Escrever instruções para um computador executar
- **Linguagem de programação de alto nível:** Linguagem de programação cuja a leitura é mais aproximada à do idioma inglês do que na leitura de uma linguagem de programação de baixo nível
- **Linguagem de programação de baixo nível:** Linguagem de programação cuja escrita tem aparência mais próxima do sistema binário (0s e 1s) do que de uma linguagem de programação de alto nível

Programação

- Programar computadores é uma tarefa que exige tempo e dedicação para ser corretamente aprendida.
- **É necessário deixar a mente se acostumar com a nova forma de pensar.**
- Você precisa responder a algumas perguntas antes de começar:
 - **Você quer aprender a programar?:**
 - **A vontade deve vir de você.**
 - **Programar é saber utilizar uma linguagem de programação para resolver problemas**
 - **Como está seu nível de paciência?**
 - **Erro comum: querer fazer coisas difíceis logo de início**
 - **Quanto tempo você pretende estudar?**
 - **Não tente aprender tudo ou entender tudo rapidamente.**
 - **Se não entender na 2ª tentativa, pare e volte a tentar amanhã.**
 - **Qual o seu objetivo ao programar?**

Programação



Front-End

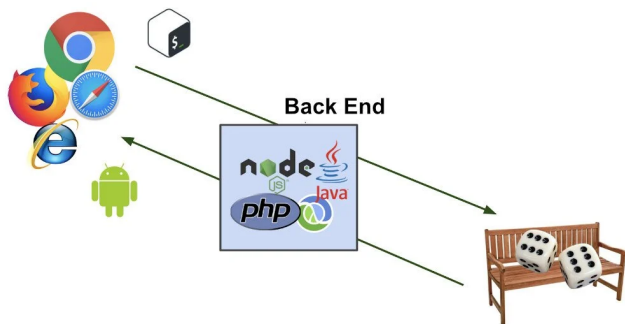
- O front-end é a parte do sistema com a qual o usuário interage diretamente. É a interface gráfica e as funcionalidades visíveis para o usuário.



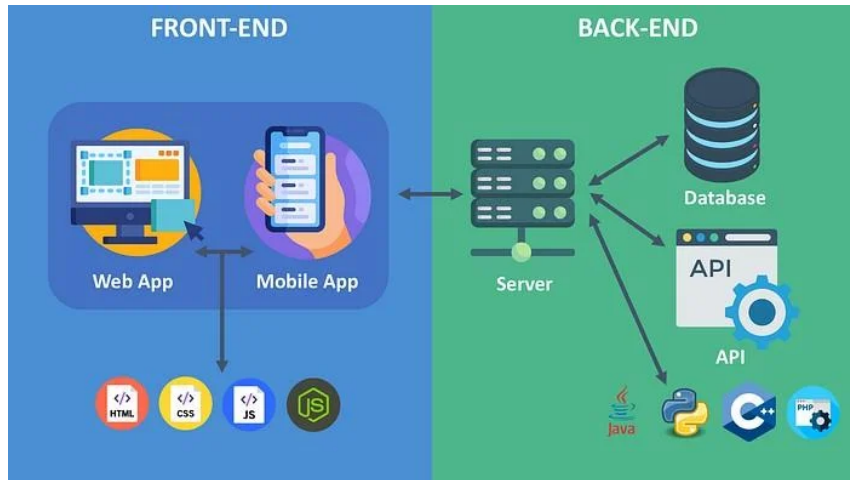
- Os desenvolvedores de front-end lidam com linguagens de marcação (como HTML), estilização (geralmente com CSS) e programação (geralmente com JavaScript).
- As tecnologias de front-end são responsáveis por tornar a experiência do usuário agradável e intuitiva, lidando com layouts, animações, interações e validações de formulários, entre outras

Back-End

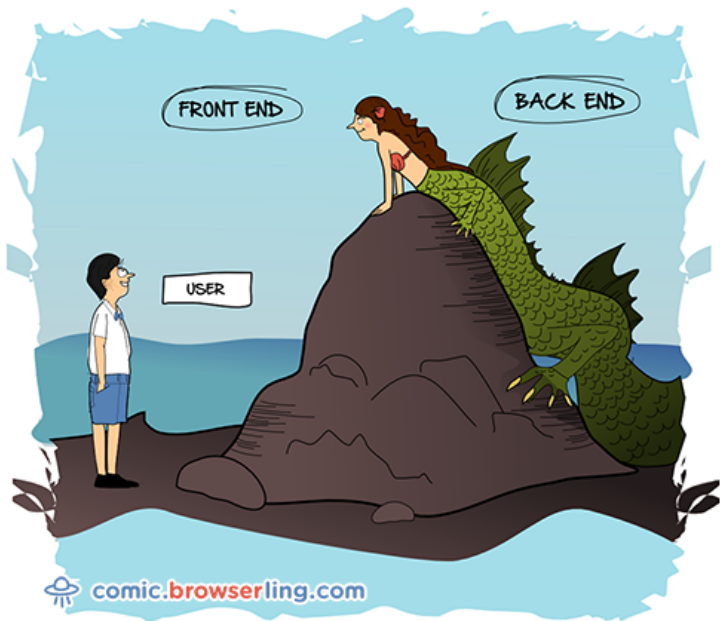
- É a parte "invisível" do sistema que lida com a lógica de negócios, processamento de dados, segurança, comunicação com o banco de dados e outras operações que não são visíveis ao usuário.
- Os desenvolvedores de back-end trabalham com linguagens de programação como Python, Java, entre outras, bem como com frameworks específicos para desenvolvimento web.



Front End e Back-End



Front End e Back-End



Programação

- Para fazer com que um problema seja resolvido corretamente por um sistema, deve-se dar atenção a diversos aspectos
 - **Compreender o que foi pedido**
 - Novamente, não sejam ansiosos e quererem resolver grandes aplicações no início
 - Perca algum tempo na leitura dos enunciados
 - **Realizar deduções na construção do programa**
 - Muitas das vezes é necessário realizar algumas deduções
 - 1 O carro está na garagem ou na frente de casa
 - 2 O carro não está na frente da casa
 - Considerando as afirmações 1 e 2 como verdadeiras, deduzimos que **O carro está na garagem**
 - **Enumerar as etapas a serem realizadas**
 - Dividir para conquistar
 - Analisar em qual passo eu vou precisar **decidir um caminho** ou ter que voltar para um passo anterior

Programação

- **Analisar outras possibilidades de solução**
 - Você tem 3 amigos e precisa somar a idade de 2 amigos que possuem as maiores idades. Como resolver este problema?
 - Forma 1
 - 1 Descobrir a maior idade
 - 2 Descobrir a segunda maior idade
 - 3 Somar
 - Forma 2
 - 1 Descobrir quem tem a menor idade
 - 2 Somar a idade dos outros dois
 - Novamente: Se deparar com alguma dificuldade para resolver, respire um pouco... Tome um água... tente pensar se poderia existir outra forma de solucioná-lo
- **Pensar em todos os detalhes**
 - Esquecer um ingrediente em uma receita de bolo pode nos trazer um resultado não esperado





Algoritmos

- O computador é uma maquina obediente para executar instruções







- De um lado do rio você tem um homem, uma raposa, uma galinha e um milho
- Seu objetivo é transportar um item por vez com uma canoa, porém:
 - Se deixar a raposa sozinha com a galinha, a raposa comerá a galinha
 - Se a galinha ficar sozinha com o milho, o milho será devorado
- Como resolver o problema?





Algoritmos

   	~ RIO ~	
---	---------	--





1. Leve a galinha para o outro lado.

 	~ RIO ~	 
---	---------	--





2. Volte sozinho.

  	~ RIO ~	
---	---------	--





3. Leve a raposa para o outro lado.

	~ RIO ~	  
---	---------	--





4. Volte com a galinha.

  	~ RIO ~	
---	---------	--





5. Leve o milho para o outro lado.

	~ RIO ~	  
---	---------	--

6. Volte sozinho.

 	~ RIO ~	 
---	---------	--

7. Leve a galinha para o outro lado.

	~ RIO ~	   
--	---------	--

Algoritmos

- Ao estruturar os passos (instruções) de um programa, estamos montando um **algoritmo!**
- Informalmente, um algoritmo é como uma receita.
- Problema: lavar roupa suja
- Algoritmo:
 - 1 Colocar a roupa em um recipiente
 - 2 Colocar um pouco de sabão e amaciante
 - 3 Encher de água
 - 4 Mexer tudo até dissolver todo o sabão
 - 5 Deixar de molho por vinte minutos
 - 6 Esfregar a roupa
 - 7 Enxaguar e Torcer

Algoritmos

Algoritmo

Sequência finita de instruções para se resolver um problema.

Programa

Programas de computador são algoritmos executados pelo computador (em linhas gerais).

- O computador é uma máquina que automatiza a execução de algoritmos.

Algoritmos

- Nós podemos ter algoritmos diferentes para resolver um problema
- Exemplo: Dar o troco
- Primeira solução:
 - 1 calcular a diferença entre o preço de compra e a quantidade de dinheiro que o cliente dá ao comerciante.
 - 2 O resultado desse cálculo é o valor total que o lojista deve devolver ao comprador.
 - 3 Seleccionamos as moedas e notas apropriadas
- Segunda solução
 - 1 O comerciante começa com o preço de compra e chega ao valor dado.
 - 2 Primeiro, as moedas são seleccionadas para aproximar o preço até o valor em reais
 - 3 Por fim, são seleccionados as notas para aproximar o preço até o valor dado

Vamos precisar de:

- Uma linguagem de programação: regras léxicas (ortografia) e sintáticas (gramática) para se escrever o programa
- Uma IDE: software para editar e testar o programa
- Um compilador: software para transformar o código fonte em código objeto
- Um gerador de código ou máquina virtual: software que permite que o programa seja executado

Linguagem de programação

Léxica

Diz respeito à correção das **palavras** "isoladas" (ortografia).

Exemplo (Português):

cachorro

caxorro



Linguagem de programação:

main

maim



Linguagem de programação

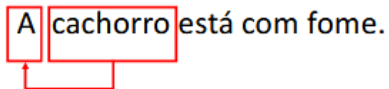
Sintática

Diz respeito à correção das **sentenças** (gramática).

Exemplo (Português):

O cachorro está com fome.

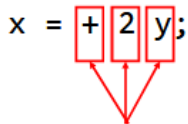
A cachorro está com fome.



Linguagem de programação:

$x = 2 + y;$

$x =$ +2y;

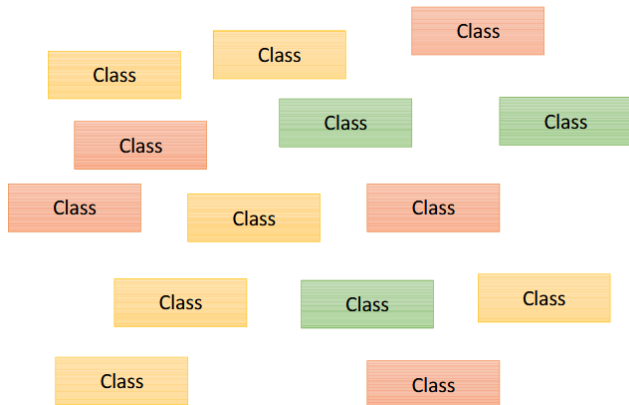


Porque JAVA

Simples	Java tem um conjunto de recursos conciso e coeso que a torna fácil de aprender e usar.
Segura	Java fornece um meio seguro de criar aplicativos de Internet.
Portável	Os programas Java podem ser executados em qualquer ambiente para o qual houver um sistema de tempo de execução Java.
Orientada a objetos	Java incorpora a moderna filosofia de programação orientada a objetos.
Robusta	Java incentiva a programação sem erros por ser fortemente tipada e executar verificações de tempo de execução.
Várias threads	Java fornece suporte integrado à programação com várias threads.
Neutra quanto à arquitetura	Java não tem vínculos com uma determinada máquina ou arquitetura de sistema operacional.
Interpretada	Java dá suporte a código para várias plataformas com o uso do bytecode.
Alto desempenho	O bytecode Java é altamente otimizado para obtenção de velocidade de execução.
Distribuída	Java foi projetada visando o ambiente distribuído da Internet.
Dinâmica	Os programas Java carregam grandes quantidades de informações de tipo que são usadas na verificação e resolução de acessos a objetos no tempo de execução.

Classe

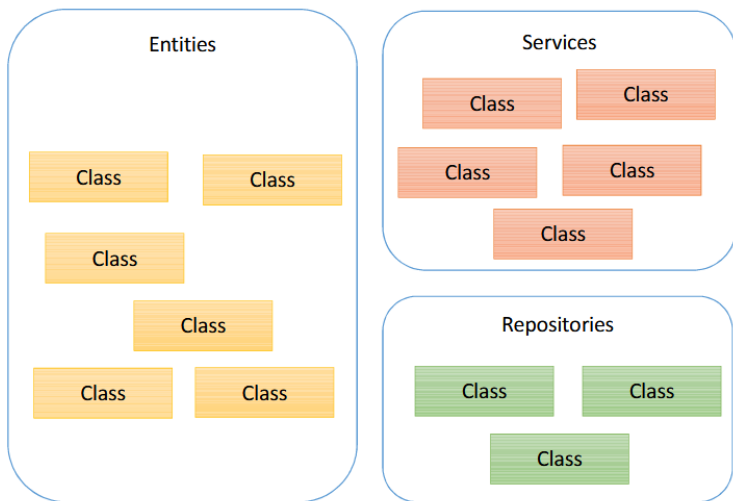
Programa Java: um conjunto de uma ou mais classes.



- De início vamos considerar uma classe como se fosse um arquivo cujo vamos escrever nossos programas
- Vamos melhorar esta definição posteriormente

Pacote

- Uma pasta para organização de nossas classes de acordo com o modelo de negócio projetado



Configurações e ações básicas

- Workspace (selecione a pasta aonde os projetos serão salvos)
- Mudar o layout: Window -> Perspective -> Open Perspective -> Other -> Java Zerar o layout: Window -> Perspective -> Reset Perspective
- Mostrar a aba Console: Window -> Show View -> Console
- Criar projeto: File -> New -> Java Project
- Criar classe:
 - Botão direito na pasta "src-> New -> Class
 - Package: depende do projeto (application, service, entity)
 - Nome da classe: Main (com M maiúsculo)
 - Marque a opção: public static void main(String[] args)

Configurações e ações básicas

Mudar o tamanho da fonte:

- CTRL +

- CTRL -

Indentação: CTRL + SHIFT + F

AutoCompletar: CTRL + SPACE

Importação automática de bibliotecas: CTRL + SHIFT + O