# Livro do Educador

## Desenvolvimento de Aplicativos Móveis

### Programação Orientada a Objetos

### APRESENTAÇÃO DO LIVRO DO EDUCADOR

Com os avanços tecnológicos, cada vez mais a utilização dos dispositivos móveis torna-se indispensável para as pessoas. Consequentemente, o mercado de trabalho necessita cada vez mais de profissionais desta área. Esperamos que, com este material, você possa desenvolver com seus alunos os conceitos e práticas essenciais para quem decide atuar na área de desenvolvimento de aplicativos móveis.

Assim, este livro foi desenvolvido para ajudá-lo a proporcionar uma base teórica e prática aos alunos, para formar profissionais capazes de exercer atividades relacionadas à desenvolvimento de aplicativos móveis.

Os conteúdos aqui presentes possibilitam que você consiga desenvolver em aula todos os conceitos básicos relacionados à lógica de programação, programação orientada a objetos, boas práticas e padrões de projeto. O material foi desenvolvido criteriosamente para que você consiga trabalhar os conteúdos de maneira simples e agradável, possibilitando ao aluno melhor interação com a aula.

Boa aula!

### OBJETIVO

Preparar o aluno para que tenha capacidade de se desenvolver e atuar como desenvolvedor de aplicativos móveis, com conhecimentos sólidos em Programação Orientada a Objetos - POO nas plataformas Android e iOS, senso crítico para projeto e design de software e total capacidade de autodesenvolvimento.

Introduzindo o aluno às técnicas básicas de programação, iniciando por conceitos sólidos em POO até conceitos mais aprofundados que abrangem todo um processo de desenvolvimento de software, como por exemplo, análise de requisitos, criação de esboços e interfaces gráficas e um pouco de gerência de projetos.

### ORIENTAÇÕES AO EDUCADOR

Este manual está dividido de acordo com as aulas que serão ministradas por você. Nele, você encontrará a melhor forma de trabalhar em aula cada conteúdo do curso. Para que você possa aproveitar bem o material, observe algumas instruções:

* observe os pontos importantes abordados em cada aula para conhecer dicas que te auxiliarão no desenvolvimento dos conteúdos;
* nos conteúdos complementares, você encontrará sugestões, como links de vídeos ou textos, que complementam o material e poderão te auxiliar no desenvolvimento das aulas;
* em cada aula são apresentadas estratégias para auxiliar o aluno no desenvolvimento das atividades;
* no TPD, você também encontrará dicas e estratégias para auxiliar os alunos.

### FICHA TÉCNICA

* **Público Alvo**

Jovens, acima de 16 anos, com Ensino Médio concluído ou em andamento, e que desejam atuar na área de Programação ou Desenvolvimento de App's.

* **Pré requisitos educador**

Formação Mínima: Técnico em Informática; preferencialmente com experiência em desenvolvimento de App, plataformas IOS e Android.

Formação Ideal: Graduação em Ciência da Computação, Engenharia da Computação, Sistemas de Informação ou áreas afins.

* **Pré-requisitos do aluno**

Ter conhecimento básico em Windows e raciocínio lógico.

* **Hardware mínimo dos computadores**

Microcomputador (desktop ou laptop) com processador de 2 gigahertz (GHz); memória RAM de 2 gigabyte (GB) para sistema operacional 32 bits, ou 4 GB para sistema operacional 64 bits; espaço em disco rígido de 20 GB; conectado à internet banda larga de alta velocidade.

* **Software das máquinas**

As máquinas dos alunos devem possuir a instalação do Sistema Operacional Windows 7 (ou superior), Ubuntu 16 ou OSX (10 ou superior), um navegador para internet, de preferência o Google Chrome. Na máquina do educador além dos softwares citados acima, é necessário ter instalado o Microsoft Office 2013 ou superior para o caso de sistemas operacionais Windows e OSX, um leitor de PDF, de preferência o Adobe Reader, também o programa Git e SourceTree (Windows e OSX).

* **Recomendações específicas**
* Salas de aula equipadas com um computador por aluno.
* Iluminação adequada.
* Aparelho de ar condicionado, com potência que atenda aos requisitos mínimos para a correta climatização do cliente.

O aparelho pode ser adaptado de acordo com a região do país.

* Quadro branco em ótimo estado de uso, com pincéis e apagador.
* Mesas e cadeiras conservadas.
* Um computador conectado à internet banda larga (100Mb ou similar) e outros equipamentos de apoio para vídeo, como televisor 42” para projeção ou projetor, e aparelho de som em perfeito estado de funcionamento.
* Tablet ou smartphone com sistema operacional Android 4.4 ou superior para a realização das atividades práticas.
* **Áreas de atuação do profissional**

Atuar em empresas públicas e privadas, instituições, indústrias, prestar serviços como profissional liberal, entre outros.

* **Faixa salarial da profissão**

A faixa salarial pode variar dependendo da experiência do profissional e da região na qual atua. O profissional de Programação Orientada a Objetos possuí a média salarial de R$ 1.200,00 a R$ 3.000,00.

# Unidade 1

## Aula 1 – Introdução

Olá educador, esta é a aula introdutório do nosso curso de Programação Orientada a Objetos (POO). Seguindo as instruções deste livro, você deve apresentar o curso ao aluno de forma motivacional. Os alunos virão para este curso com o intuito de aprender desenvolvimento mobile, e como pré-requisito para isto é necessário que eles aprendam programação básica e em seguida programação orientada a objetos. E é voltado nisso que este módulo (POO) foi construído, acrescentar uma bagagem ao aluno antes de entrar na programação mobile.

Devido a este fato a aula deve ser motivadora e tão prazerosa quanto aulas aulas dos módulos seguintes (Android e iOS)

### CARGA HORÁRIA

Conforme o plano de aula, esta aula terá duração de 1h30 e deverá ser conduzida de acordo com as orientações pedagógicas.

### OBJETIVO DA AULA

Ao final da aula, você deverá garantir que o aluno tenha subsídios para

* Compreender o mercado de trabalho em programação;
* Conhecer o histórico da Programação Orientada a Objetos (POO);
* Ter o primeiro contato ou revisar lógica de programação com os exercícios.

### ORIENTAÇÕES PEDAGÓGICAS

Para atender os objetivos de aprendizagem, você deverá conduzir o processo de ensino considerando a organização didática apresentada a seguir:

* 40 minutos de aula expositiva;
* 10 minutos para tirar as dúvidas dos alunos;
* 40 minutos para desenvolver as atividades propostas para a turma.

### TÓPICOS DE ESTUDO

Todos os tópicos a seguir, conforme livro do aluno, devem ser trabalhados de forma dinâmica, criativa, com embasamento teórico e prático voltado ao mercado de trabalho.

* Para quem é este livro;
* Paradigmas de programação;
* Histórico;
* Evolução das linguagens;
* Vantagens de POO;
* Desvantagens.

### PONTOS IMPORTANTES

Para garantir ao aluno um aprendizado significativo, resgate os principais conceitos de cada tópico abordado na aula. Neste momento, é importante que você utilize dicas e sugestões para reforçar os temas trabalhados, oferecendo também exemplos que possibilitem a relação entre teoria e prática.

### 1.1 Para quem é este livro

Educador, nesta aula é importante que você explique ao aluno que linguagem de programação é um método padronizado para comunicar instruções a um computador, um conjunto de regras sintáticas e semânticas usadas para definir um programa. Ressalte que o termo mencionado “paradigma de programação” será abordado na seção seguinte.

Explique ao aluno que a linguagem utilizada nos computadores é a linguagem binária, ela é composta de combinações de 0s (zeros) e 1s (uns), podendo ser chamada de palavra e, é armazenada no hardware do computador por meio de registradores, onde o zero significa ausência de energia e o um significa presença dela. Cada registrador armazena um bit (Zero ou Um) e uma palavra é o conjunto de 8 registradores, também chamado de byte. Estas combinações vão além: quanto maior o tamanho da palavra, maior o poder de processamento de cálculo do computador.

Os bytes podem ser convertidos para letras do nosso alfabeto. Repare que se permutarmos entre 0 e 1 os valores de um número de 8 dígitos, temos o número de letras do alfabeto e mais alguns caracteres especiais, ou seja, 82 = 64 possibilidades de combinações.

O computador entende desta maneira, mas seria quase impossível programarmos tarefas para ele executar usando esta linguagem de máquina, por isso foram criadas as linguagens de programação.

Linguagem de programação abstrai a linguagem de máquina para um nível compreensível e intuitivo suficiente para os humanos. Existem vários níveis de abstração para a linguagens de programação, desde o nível baixo, como a VHDL que descreve rotinas e sinais eletrônicos de hardware, linguagens de médio, nível como Assembly, passando por linguagens de alto nível como C, Java, C#, Python, Ruby e Javascript.

Resumindo, a linguagem de programação tem o objetivo de facilitar, ou abstrair, a programação de máquinas digitais.

### 1.2 - Paradigmas de programação

Educador, explique que existem muitas linguagens de programação, cada qual com sua vantagem e desvantagem. Essa grande diversidade se dá devido a necessidade que cada tipo de projeto de software requer, algumas linguagens se saem melhor para realização de um objetivo do que outras, como por exemplo as linguagens C e Java.

Java se sai melhor em softwares comerciais, para uso em PCs e também celulares, a linguagem C leva vantagem na hora de criar controladores embarcados, sistemas operacionais ou qualquer outro tipo de software que tenha que rodar em um computador de processamento limitado, como uma SmartHouse sendo controlada por um Arduino. Pesquise os termos que não são de seu conhecimento.

Esta diferença entre as duas linguagens é devida a diferença de abstração entre elas. A linguagem C é de baixo nível se comparado com Java. Nos tópicos seguintes, as diferenças entre linguagem procedural e orientada a objetos deixarão claro o que cada linguagem tem de vantagem sobre a outra.

Este tópico faz um comparativo entre os dois principais paradigmas de programação: Procedural ou estrutural e Orientado a Objetos, explique cada um deles e suas estruturas.

### 1.3 – Histórico

Educador, faça um breve resumo da história da programação orientada a objetos, mostrando alguns marcos durante este processo. A seguir, exemplos de imagens que podem ser utilizados para demonstração.

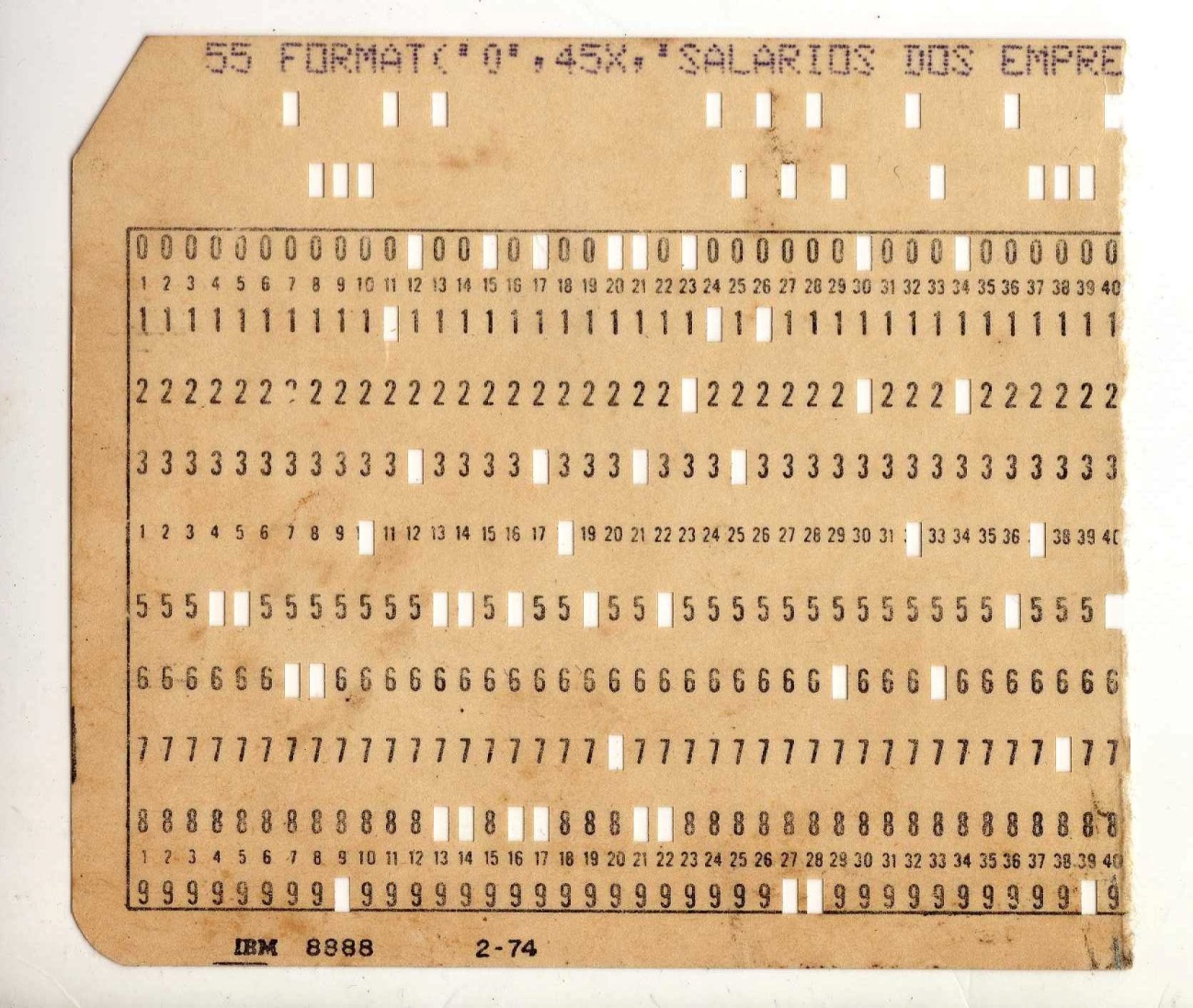


Figura 1. – Cartão perfurado usado para gerar holerite dos funcionários

<https://judsoncanto.files.wordpress.com/2012/12/cartc3a3o-perfurado-ibm-e1355092834371.jpg>



Figura 1. - Operadora de computador segurando uma fita magnética

<http://pcworld.com.br/idgimages/galerias/Evolucao-storage/03.jpg>

#### 1.3.1 Simula

Explique ao aluno que a linguagem de programação Simula foi o início da linguagem de programação, sendo primordial para o amadurecimento do conceito de linguagem orientada a objetos. Muitos esquecem dessa etapa histórica e abordam apenas a linguagem SmallTalk como sendo a primeira linguagem POO.

A Simula é baseada em ALGOL 60 e está em desuso atualmente, mas é bastante intuitiva e conseguimos ver que ela tem um nível de abstração um pouco mais elevada do que a linguagem procedural C, pois com ela podemos definir classes.

A Programação Simula foi desenvolvida para simular eventos discretos e antigamente ela foi utilizada para fins matemáticos e estatísticos. Aproveite para descrever os dois tipos de simulações de eventos discretos: modelos contínuos e modelos discretos.

#### 1.3.2 Alan Kay

Educador, comente sobre Alan Kay, o programador da Simula, um dos idealizadores e formador da linguagem orientada a objetos. Aproveite e explique sobre polimorfismo, fale que verbos são propriedades, ou seja, ação, processo ou estado, dos substantivos, o sujeito.

Além de ser considerado um dos pioneiros da computação, hoje Alan Kay ele se dedica a educação de crianças. Tem como propósito mais profundo da computação criar máquinas inteligentes o suficiente que irão entender e captar padrões no mundo.

Veja o vídeo da palestra de Alan Kay compartilhando uma ideia sobre ideias: https://www.ted.com/talks/alan\_kay\_shares\_a\_powerful\_idea\_about\_ideas?language=pt-br#t-16094

#### 1.3.3 SmallTalk

Educador, informe que a SmallTalk foi considerada oficialmente a primeira linguagem de programação orientada a objetos.

Mostre brevemente o histórico das mídias de entrada e saída do computador, se houver tempo disponível em aula.

http://pcworld.com.br/galerias/evolucao-dos-dispositivos-de-armazenagem/#imagem0

#### 

### 1.4 - Evolução das linguagens

Educador, explique a diferença entre as linguagens C, C++, C# e Objective C. Informe que C é uma linguagem estrutural e a diferença entre as demais linguagens é que ela não é orientada a objetos e que devido a sua fama e robustez a linguagem C derivou diversas outras linguagens, entre elas o C++, C# e Objective-C, todas orientadas a objetos.

A linguagem C++ é a extensão de C que adicionou recursos de orientação a objetos, ou seja, é a linguagem C orientada a objetos.

A Objective-C é uma linguagem adotada pela Apple para criação de seus aplicativos, sendo a principal e exclusiva. Todos os aplicativos OSX, iOS, watchOS e demais sistemas operacionais eram escritos em Objective-C, ela deriva diretamente do C++, então é possível utilizar recursos C++ para criação destes aplicativos.

No ano de 2014, a linguagem Objective-C passou a ser substituída pela nova e moderna Swift, que agora é a linguagem oficial da Apple. Diferente de Objective-C, Swift não é exclusiva da Apple, ou seja, é de código aberto (open source), ou seja, outros desenvolvedores, além da Apple, podem criar produtos utilizando essa linguagem.

A linguagem C# é outra derivada de C e C++. Criada pela Microsoft, está presente na maioria dos seus produtos recentes, entre eles o Windows Phone, ela também é open source e é utilizada para criação de jogos e animações 3D pela ferramenta Unity3D.

A seguir a ordem de lançamento das linguagens de programação.

* 1954 – Fortran;
* 1958 – Lisp;
* 1958 – Algol 58;
* 1964 – Simula;
* 1967 – BCPL;
* 1969 – B;
* 1969 – SmallTalk;
* 1971 – C;
* 1975 – Scheme;
* 1983 – C++;
* 1991 – Java.

Educador, é interessante tomar conhecimento dos diferenciais entre essas linguagens.

### 1.6 - Vantagens de POO

Neste item enumere todas as vantagens que a POO possui como abstração, reuso, manutenção e legibilidade.

### 1.7 - Desvantagem

**Educador,** após explicado as vantagens, ressalte as desvantagens da utilização da POO.

### 1.8 – Exercícios

As perguntas têm por objetivo fixar os principais conceitos abordados durante a aula. Para isso, viabilize o tempo necessário para o aluno responder às perguntas, acompanhando-os nas dúvidas. Para correção das perguntas, considere as respostas a seguir:

### Lógica de programação

Se você nunca programou, saiba que é preciso exercitar o cérebro, de forma que ele pense “logicamente”. Serão feitos alguns exercícios de lógica de programação utilizando pseudocódigo.

Pseudocódigo é uma modalidade de representação de algoritmos um pouco mais literal e semântica, quase próxima da linguagem natural do ser humano. É utilizado para prática de algoritmos e lógica de programação e é escrito como uma receita de bolo, ou como instruções para realizar algo. Veja um exemplo de algoritmo com pseudocódigo:

**Escreva um algoritmo que realize a soma de 1+2 e exiba o resultado.**

Algoritmo Soma;

VAR

digito1, digito2, soma : INTEIRO;

INÍCIO

digito1 <- 1;

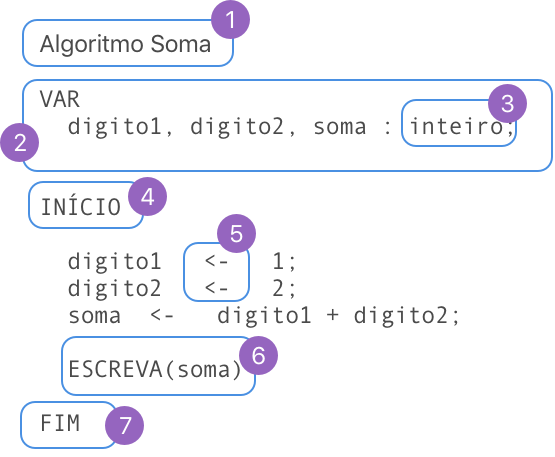
digito2 <- 2;

soma <- digito1 + digito2;

ESCREVA(soma);

FIM

Os ponto e vírgulas são características de algumas linguagens de programação como o Pascal, C++ e o Java, porém, em outras linguagens de programação, eles podem não aparecer no pseudocódigo, pois essas podem adotar outra semântica.



**Corpo do algoritmo**

Figura 1. 3 - Explicação do algoritmo soma

1. Título do algoritmo. Geralmente mostra uma descrição breve e objetiva do que o algoritmo se trata, ou seja, o que ele fará.
2. Área para declaração de variáveis a serem utilizadas no decorrer do algoritmo.
3. Tipo que as variáveis irão aceitar. Possíveis valores: inteiro, booleano, texto, ou decimal.
4. Indica o início da execução do código.

O corpo do algoritmo está entre o INÍCIO e o FIM. É onde fica a lógica de programação.

1. “<-” atribui um valor a uma variável.
2. Exibe um dado.
3. Indica o final da execução do algoritmo.

Agora faça você mesmo

1. Faça um algoritmo que receba dois números e ao final mostre a soma, a subtração, a multiplicação e a divisão dos números lidos.

Resposta

Algoritmo Soma;

VAR

digito1, digito2, soma : INTEIRO;

INÍCIO

digito1 <- 1;

digito2 <- 2;

soma <- digito1 + digito2;

ESCREVA(soma);

FIM

1. Escrever um algoritmo para determinar o consumo de um automóvel ao percorrer uma determinada distância. Será fornecida a distância total percorrida pelo automóvel e o total de combustível gasto.

**Resposta:**

**Algoritmo** Consumo

**VAR**

distância, combustível, média: **decimal**;

**INÍCIO**

**LEIA**(distância);

**LEIA**(combustível);

média 🡨distância /combustível;

**ESCREVA**(média);

**FIM**

1. Escrever um algoritmo que leia o nome de um vendedor, o seu salário fixo e o total de vendas efetuadas por ele no mês (em dinheiro). Sabendo que este vendedor ganha uma comissão de 15% do valor de seu salário para cada venda efetuada no mês, informar o seu nome, o salário fixo e salário com as comissões ao final do mês.

**Resposta:**

**Algoritmo** Salário

**VAR**

nome, salário, total, comissão\_por\_venda: **decimal**;

vendas: **inteiro**;

**INÍCIO**

**LEIA**(salário);

**LEIA**(nome);

**LEIA**(vendas);

comissão\_por\_venda <- salario \* 0,15;

total 🡨 salario + (comissão\_por\_venda \* vendas);

**ESCREVA**(“O funcionário ” + nome + “ de salário ” + salário + “ recebeu ” + total);

**FIM**

1. Escrever um algoritmo que leia o nome de um aluno e as notas das três provas que ele obteve no semestre. No final informar o nome do aluno e a sua média (aritmética).

**Resposta**: Educador, para responder essa questão é necessário explicar ao aluno que Iterador ou Iteração são formas de repetições para solução de problemas onde cada nova repetição recebe o valor da operação anterior até que a iteração alcance um limite estabelecido ou a solução seja alcançada. Ressalte que este assunto será abordado mais adiante na aula sobre Loops.

**Resposta:**

**Algoritmo** Aluno

**VAR**

nota, média: **decimal**;

i: **inteiro**; // Iterador

**INÍCIO**

média🡨0;

i 🡨3;

**ENQUANTO (**i >0)**FAÇA**

**LEIA**(nota);

média🡨média+ nota;

i 🡨 i - 1;

**FIM\_ENQUANTO**

média🡨média/3;

**ESCREVA**(média);

**FIM**

1. Ler dois valores para as variáveis A e B, e efetuar as trocas dos valores de forma que a variável A passe a possuir o valor da variável B e a variável B passe a possuir o valor da variável A. Apresentar os valores trocados.

**Resposta:**

**Algoritmo** Troca

**VAR**

A, B, aux: **inteiro**;

**INÍCIO**

**LEIA**(A);

**LEIA**(B);

aux 🡨 A;

A 🡨 B;

B 🡨 aux;

**ESCREVA**(“A: ” + A + “ B: ” + B);

**FIM**

**Resposta:** Educador, neste exemplo para que os dois valores se mantenham conservados é preciso que um deles fique guardado. A variável aux, neste caso, guarda o valor de A, para que A possa ser atribuído com o valor de B. Após A receber o valor de B, ambas terão os mesmos valores. Neste momento, se não tivéssemos uma variável auxiliar (aux) que preservasse o valor original de A, este seria perdido. Então aux seria uma variável backup de um dos valores (A ou B).

1. Ler uma temperatura em graus Celsius e apresentá-la convertida em graus Fahrenheit. A fórmula de conversão é: F=(9\*C+160) / 5, sendo F a temperatura em Fahrenheit e C a temperatura em Celsius.

**Resposta:**

**Algoritmo** Conversão

**VAR**

C, F **: DECIMAL**; //Já havia apontado essa alteração em outras oportunidades.

**INÍCIO**

**LEIA**(C);

F 🡨 (9\*C + 160)/5;

**ESCREVA**(“A temperatura em Fahrenheit é: “ + F);

**FIM**

1. Elaborar um algoritmo que efetue a apresentação do valor da conversão em real (R$) de um valor lido em dólar (US$). O algoritmo deverá solicitar o valor da cotação do dólar e também a quantidade de dólares que o usuário possui.

**Resposta:**

**Algoritmo** Câmbio

**VAR**

cotação, dólares, reais**: DECIMAL**; //Já havia apontado essa alteração em outras oportunidades.;

**INÍCIO**

**LEIA**(cotação);

**LEIA**(dólares);

reais🡨dólares\* cotação;

**ESCREVA**(“Valor em reais: R$“ + reais);

**FIM**

1. A Loja Mamão com Açúcar está vendendo seus produtos em 5 (cinco) prestações sem juros. Faça um algoritmo que receba um valor de uma compra efetuada pelo cliente e mostre o valor de cada um das prestações que ele pagará.

**Resposta**

**Algoritmo** Prestação

**VAR**

preço, prestação : DECIMAL;

**INÍCIO**

**LEIA**(preço);

prestação🡨preço/ 5;

**ESCREVA**(“5x de “ + prestação);

**FIM**

1. O custo ao consumidor de um carro novo é a soma do custo de fábrica com a percentagem do distribuidor e dos impostos (aplicados, primeiro os impostos sobre o custo de fábrica, e depois a percentagem do distribuidor sobre o resultado). Supondo que a percentagem do distribuidor seja de 28% e os impostos 45%. Escrever um algoritmo que leia o custo de fábrica de um carro e informe o custo ao consumidor do mesmo.

**Resposta:**

**Algoritmo** Custo

**VAR**

fábrica, consumidor**: DECIMAL**; //Já havia apontado essa alteração em outras oportunidades.;

**INÍCIO**

**LEIA**(fábrica);

consumidor 🡨fábrica\*1.45\*1,28;

**ESCREVA**(“Preço de fábrica: “ + fábrica+ “ e Preço de consumidor: “ + consumidor);

**FIM**

1. Faça um algoritmo que receba um número e diga se este número está no intervalo entre 100 e 200.

**Resposta**

**Algoritmo** Intervalo

**VAR**

num: inteiro;

**INÍCIO**

**LEIA**(num);

**SE**((num >= 100)**E**(num <= 200))**ENTÃO**

**ESCREVA**(“O número está dentro do intervalo”);

**SENÃO**

**ESCREVA**(“O número não está dentro do intervalo”);

**FIM\_SE**

**FIM**

1. Escrever um algoritmo que leia o nome e as três notas obtidas por um aluno durante o semestre. Calcular a sua média (aritmética), informar o nome e sua menção: aprovado (media >= 7.0), reprovado (media < 5.0) ou recuperação (média entre 5 a 6.9).

**Resposta:**

**Algoritmo** Aprovação

**VAR**

nome, nota, média: **decimal**;

i: **inteiro**; // Iterador

**INÍCIO**

**LEIA**(nome);

média🡨 0;

i 🡨 1;

**ENQUANTO** (i <= 3) **FAÇA**

**LEIA**(nota);

média🡨média+ nota;

**FIM\_ENQUANTO**

média🡨média/3;

**ESCREVA**(média);

**SE**(média< 5)**ENTÃO**

**ESCREVA**(nome + “reprovado”);

**SENÃO SE** ((media >= 5)**E**(média <7))**ENTÃO**

**ESCREVA**(nome + “recuperação”);

**SENÃO**

**ESCREVA**(nome + “aprovado”);

**FIM\_SE**

**FIM**

1. Ler 80 números e ao final informar quantos desses números estão no intervalo entre 10 (inclusive) e 150 (inclusive).

**Resposta:**

**Algoritmo** Lista

**VAR**

lista[80]:**inteiro**

i, quantidade: **inteiro**;

**INÍCIO**

i 🡨 1;

quantidade 🡨 0;

**ENQUANTO** (i <= 80)**ENTÃO**

**LEIA**(lista[i]);

**SE**((lista[i]>=10)**E**(lista[i]<= 150)) **ENTÃO**

quantidade 🡨quantidade + 1;

**FIM\_SE**

i 🡨 i + 1;

**FIM\_ENQUANTO**

**SE**(quantidade > 0)**ENTÃO**

**ESCREVA**(“Quantidade de números no intervalo: “ + quantidade);

**SENÃO**

**ESCREVA**(“Todos estão fora do intervalo”);

**FIM\_SE**

**FIM**

1. Faça um algoritmo que receba a idade de 75 pessoas e mostre uma mensagem informando "maior de idade" ou "menor de idade" para cada uma dessas pessoas. Considere a idade a partir de 18 anos como maior de idade.

**Resposta:**

**Algoritmo** Idade

**VAR**

idade[75]:**inteiro**

i: **inteiro**; // Iterador

**INÍCIO**

i<- 1;

**ENQUANTO** (i <= 75)**FAÇA**

**LEIA**(idade[i]);

**SE**(idade[i] >=18)**ENTÃO**

**ESCREVA**(“Maior de idade”);

**SENÃO**

**ESCREVA**(“Menor de idade”);

**FIM\_SENÃO**

**FIM\_ENQUANTO**

**FIM**

### 1.9 - TDP

**Educador, é neste momento que o aluno começará a desenvolver seu projeto.**

Ainda considerando a teoria relacionada à prática, é necessário que você seja um orientador do processo de desenvolvimento desta atividade, garantindo que o aluno tenha o conhecimento necessário para tornar-se um profissional qualificado para o mercado de trabalho.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

O **TDP,** ou **Trabalho de Desenvolvimento Prático**, é um projeto prático no qual você criará uma aplicação orientada a objetos completa, do início ao fim.

Este projeto não se trata de simples programação. Você conhecerá diversas etapas da concepção de um aplicativo. Desde o amadurecimento da ideia, passando pela fase de análise, modelagem, até a fase de desenvolvimento em si. Nas primeiras aulas, enquanto os conhecimentos em programação orientada a objetos não estiverem maduros, você participará da fase analítica do projeto, logo após isso, você passará para a fase de programação, arquitetura e assim por diante. Serão mostradas competências suficientes para que você possa criar um programa independente.

Definição inicial do trabalho

Você foi alocado no projeto Filmes

Você é um programador iniciante em uma grande empresa de software. Essa empresa gosta de colocar seus novos funcionários à prova para avaliá-los na capacidade de tomada de decisão. Você será avaliado.

Na hierarquia de uma empresa comum, sempre há um gerente de projetos liderando um grupo de programadores. O trabalho de um gerente de projetos consiste também em coletar informações do cliente sobre o projeto, delegar atividades ao time e criar métricas para alocação de recursos (humanos) nos projetos futuros.

Certo dia, você ouviu Nelson, seu gerente de projetos, conversando sobre um novo projeto com o diretor da empresa. Você não ouviu muito bem, mas conseguiu entender que se tratava de algo envolvendo filmes.

**Nelson está doente e precisa de um substituto**

Nelson agendou uma reunião para a semana seguinte para conversar com os desenvolvedores da empresa sobre o novo projeto e sobre a situação de saúde dele. Ele precisará de um substituto.

Que tal pesquisar um pouco mais sobre o assunto e ganhar destaque aos olhos de Nelson? Quem sabe você não o substitui no período de licença médica?

Tarefa

Pesquise na internet sites que falam sobre filmes. Tente procurar aplicativos (móveis ou desktop) relacionados a filmes.

Educador, o projeto do livro de POO consiste em criar uma aplicação desktop utilizando Java e os conceitos de POO. O projeto prático será um pequeno sistema que gerencia um catálogo de filmes. O usuário poderá consultar, adicionar, editar e deletar filmes, atores, produtores e produtoras, que estarão relacionados entre si. A unidade 4 apresenta imagens de como o sistema deverá se aparentar. As demais unidades possuem o passo a passo para a reprodução deste sistema. Consulte-as e reproduza o passo a passo com antecedência para que não haja problemas durante a aula. Tenha domínio total sobre esse projeto!

O aluno ainda não tem conhecimento suficiente para programar, sendo assim, ele deverá pesquisar referências e utilitários famosos que falem sobre filmes em sites, aplicativos, canais do Youtube, páginas do Facebook, blogs, fóruns e etc. O intuito principal é coletar padrões evidentes desses canais na hora de apresentar informações de um filme, como os catálogos de filmes que apresentam o título do filme, ano de lançamento, sinopse e elenco principal. Esta parte do projeto deve ser como uma dinâmica em grupo.

### COTEÚDOS COMPLEMENTARES

Educador, para agregar novos conhecimentos sobre os assuntos desta aula, conheça algumas sugestões de conteúdos:

* Leia o livro “**History of *Programming Languages (Acm Monograph Series*)”** do Richard L. Wexelblat. É uma leitura aprofundada sobre o conhecimento do histórico da programação.
* Uma leitura mais leve é o resumo na Wikipédia (https://pt.wikipedia.org/wiki/Linguagem\_de\_programa%C3%A7%C3%A3o - acessado em 03/10/2016) da obra ***Programming Language Structures*** (ORGANICK, E. I.;FORSYTHE, A. I.;PLUMMER, R. P. (1978)), que te mostrará os diferentes paradigmas de programação.

### REFERÊNCIAS

Pendente

**Referências indicadas:**

* StackOverflow:<http://stackoverflow.com/questions/tagged/java>
* GUJ: <http://www.guj.com.br/c/programacao/java>
* Devmedia: <http://www.devmedia.com.br/java/>
* Javafree: <http://javafree.uol.com.br/>
* Documentação oficial do Java: <https://docs.oracle.com/javase/7/docs/api/overview-summary.html>

**Referências indicadas:**

* Algumas imagens de cartões perfurados: <https://www.google.com.br/search?q=cart%C3%A3o+perfurado&espv=2&biw=1280&bih=614&source=lnms&tbm=isch&sa=X&ved=0ahUKEwib-8aeoe_NAhUIGx4KHdnCD-YQ_AUIBigB#imgrc=B8lBLGQA3QD-5M%3A>
* Algumas imagens de fitas magnéticas: <https://www.google.com.br/search?q=computador+fita+magn%C3%A9tica&source=lnms&tbm=isch&sa=X&ved=0ahUKEwi14MTgpO_NAhWK1h4KHcPMA70Q_AUICSgC&biw=1280&bih=614#imgrc=O8QwSh0DunKt_M%3A>
* Veja o artigo completo da Wikipédia: https://pt.wikipedia.org/wiki/Simula

### Conheça todas as referências utilizadas pelo autor para elaboração dessa aula no Livro do Aluno

Pendente