

INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
BAHIA

**CAMPUS: PORTO SEGURO**  
BR 367 – Km 57,5 – s/nº - Porto Seguro - BA  
CEP 45.810-000  
Tel: (073) 3288-6686  
Autorizado pela Portaria do MEC nº 1.981 de  
18 de dezembro de 2006  
CNPJ: 13.941.232/0001-96



## **INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA BAHIA**

### **CAMPUS PORTO SEGURO**

#### **CURSO SUPERIOR DE LICENCIATURA DA COMPUTAÇÃO**

##### **ESTUDANTES:**

RAYANA RIBEIRO BONFANTI

##### **PROFESSORA:**

POLLIANA ALMEIDA

#### **RELATÓRIO DAS PRÁTICAS DE ENSINO NA DISCIPLINA METODOLOGIAS E PRÁTICAS DE ENSINO I EM ALGORITMO**

**PORTO SEGURO – BA**

**2018**

## **Plano de Aula**

**Tema:** Manutenção Preventiva de Computadores

**Duração:** 4 horas

**Série:** 1 ano

### **Conteúdos**

- Conceito, função, importância e ligações da Placa-Mãe, Memória RAM e ROM, Processador-Cooler e Fonte.
- Desmontagem, limpeza e montagem do computador.

### **Objetivos**

- Entender do significado de cada componente;
- Compreender de sua importância, função e quais os tipos de ligações.
- Saber do conhecimento teórico e prática de como fazer uma manutenção preventiva de um computador.

### **Metodologias**

- Observação: aula será ministrada no laboratório de Montagem e Manutenção como forma de interação;
- Compreensão do conteúdo de cada componente utilizando o projetor com slide. Em caso de problemas ou necessidades a mais, será utilizado quadro negro, pincel e apagador.
- Acompanhamento da exposição dos componentes físicos como forma de interação durante o conteúdo;
- Desenvolvimento da prática da desmontagem, limpeza e montagem dos computadores que será avaliada em forma de observação prática.

### **Recursos**

- Projetor;
- Computador para o slide;
- Papel impresso com a atividade de revisão;
- 10 computadores para interação em aula;
- Slide com vídeos e fotos;

- Caso necessário: pincel, quadro negro e apagador.

### **Avaliações**

- Entendimento do significado de cada componente;
- Compreensão da importância, função e quais os tipos de ligações.
- Construção do conhecimento teórico e prática de como fazer uma manutenção preventiva de um computador.

# Curso de Manutenção Preventiva de Computadores

## Objetivos

- Ter conhecimento teórico e prático na limpeza de um computador.
- Proporcionar aos estudantes um contato/conhecimento com os componentes de computadores;

## Datas

- 05/11 à 08/11 - Inscrição e Divulgação;
- 09/11 - Resultado;
- 12/11 às 13h até 17h - Curso.

OBS: o horário pode terminar antes ou depois.

## Local

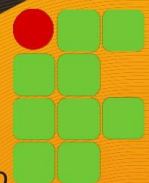
Laboratório 4 de Montagem e Manutenção no IFBA - Campus Porto Seguro.

## Informações

- Curso voltado para prática de ensino-aprendizagem e troca de conhecimentos.
- Serão selecionados 10 estudantes para a prática do curso.
- Ordem de convocados: menor idade, primeiro ano em diante e ordem de chegada.

**LINK da inscrição online:**

[https://docs.google.com/forms/d/1zk\\_quztRN0WNWfCsaCDtEIOGIDOaxEif-wGBrRZt3ZE/edit?usp=sharing](https://docs.google.com/forms/d/1zk_quztRN0WNWfCsaCDtEIOGIDOaxEif-wGBrRZt3ZE/edit?usp=sharing)



## Formulário de Inscrição

### FICHA DE INSCRIÇÃO PARA CURSO DE MANUTENÇÃO PREVENTIVA DE COMPUTADORES.

Local: IFBA - Porto Seguro, Laboratório de Montagem, nº 4.

Data do curso: 12/11 às 13h até 17h.

OBS: o horário pode terminar antes ou depois e serão selecionadas 10 estudantes.

05/11 até 08/11 - Inscrições Online.

09/11 - Resultado.

Dados Pessoais
Nome:
Tel (Celular):
Email:
Escola (Turma):
Sugestões (Comentários):
Inscrição Online
<a href="https://docs.google.com/forms/d/1zk_quztRN0WNWfCsaCDtEIOGIDOaxEif-wGBrRZt3ZE/edit?usp=sharing">https://docs.google.com/forms/d/1zk_quztRN0WNWfCsaCDtEIOGIDOaxEif-wGBrRZt3ZE/edit?usp=sharing</a>

## Reserva Laboratório

### Agendamento Laboratório de Montagem e Manutenção para Prática de Ensino

De: Rayana Ribeiro Bonfanti

Para: Silmara Maria Magnabosco

Eu, Rayana Ribeiro Bonfanti, estudante do curso Licenciatura da Computação, **matrícula:** 2017125013, venho por meio desta reservar o laboratório de montagem e manutenção, IFBA – *Campus* Porto Seguro para prática de ensino da disciplina Metodologia e Prática de Ensino I da professora Polliana Almeida Da Cruz Silva, será feito a aula no dia 12/11/2018 no horário de 13:00 às 17:00.

Porto Seguro, 31 de outubro de 2018.

Atenciosamente,

Rayana Ribeiro Bonfanti.

## Resultado das Inscrições

### RESULTADO – CURSO MANUTENÇÃO PREVENTIVA

Local: IFBA - Porto Seguro, Laboratório de Montagem, nº 4.

Data e Horário: 12/11 às 13h até 17h.

OBS: o horário pode terminar antes ou depois.

Inscrições online	
Inscrições presenciais	
Ana Beatriz Silva Santos	lindabia2003@gmail.com
Klarisse Neres de Jesus	klarisseneresdejesus15@gmail.com
Mariana Luisa Seara de Santana	marianaulisasearadesantana@gmail.com
Kailane Silva Santos	kaisilva0311@gmail.com
Maiza Rodrigues Pereira	maizarodrigues66@gmail.com
Ariane de Araujo Braz Esteves	arianeestevesbrazaraujo@gmail.com
Alaine Santos da Silva	alainelari2@gmail.com
Ester Santos da Silva	agendaestersilva25@gmail.com
Beatriz Barbosa de Amorim	beatrizbarbosadeamorim@gmail.com
Dulce Maria de Jesus Moreira	dulcemoreira527@gmail.com
Guilherme Lima Campos	gui.lcampos@hotmail.com
Wislan Uedley Souza Silva	needyandret13@gmail.com
Deivid Anthony Silva Santos	deividanthonyssantos@gmail.com
Alan Siste Coutinho	alansiste17@hotmail.com
Isaac de Oliveira Almeida	i.o321@hotmail.com
Robson da Limba Campos Junior	robson1144junior@gmail.com
Mateus Pereira Silva	teuspersi@gmail.com
Breno Monteiro Gonçalves Silva	brenobruthi@gmail.com
Gabriel Santana Portela Ernesto	portelagabriel1324@gmail.com
Convocados	
Mariana Luisa Seara de Santana	marianaulisasearadesantana@gmail.com
Mateus Pereira Silva	teuspersi@gmail.com
Gabriel Santana Portela Ernesto	portelagabriel1324@gmail.com
Breno Monteiro Gonçalves Silva	brenobruthi@gmail.com
Alaine Santos da Silva	alainelari2@gmail.com
Ariane de Araujo Braz Esteves	arianeestevesbrazaraujo@gmail.com
Klarisse Neres de Jesus	klarisseneresdejesus15@gmail.com
Kailane Silva Santos	kaisilva0311@gmail.com
Maiza Rodrigues Pereira	maizarodrigues66@gmail.com
Ana Beatriz Silva Santos	lindabia2003@gmail.com

# **Roteiro da aula - Desmontar, Limpar e Remontar um Computador**

Estudante(s): \_\_\_\_\_

## **Instruções**

### **[ETAPA 1]**

- Retire os parafusos, se necessário, para tirar a tampa do computador.
  - Utilize uma chave de fenda/estrela a depender do tipo de parafuso.

### **[ETAPA 2]**

- Será entregue um papel em branco, na qual você poderá dobrar e armazenar os parafusos retirados identificando sua origem.
  - Exemplo: Você tira 4 parafusos da tampa. Poderá colocar eles no papel dobrado identificando por “Tampa”.

### **[ETAPA 3]**

- Remova todos os fios cuidadosamente para não danificar nada.
- Remova cuidadosamente o cooler e realize a limpeza.
  - Limpe bem as hélices de forma a remover toda a poeira. Deixe o cooler em um local próximo até o fim da limpeza. Use o pincel, papel higiênico ou pano limpo. Caso necessário, retire a pasta térmica do cooler com papel higiênico e coloque uma nova.

### **[ETAPA 4]**

- Retire cuidadosamente a(s) memória(s) RAM. Pressione as laterais que a prendem na placa para removê-la.
  - Passe a borracha levemente nas vias de comunicação (parte dourada que conecta no slot). Não se deve tocar na parte dourada. Assim como o cooler, deixe a memória em um local próximo, envolvida em um pedaço de papel higiênico.

### **[ETAPA 5]**

- Retire cuidadosamente a placa de vídeo da placa mãe, caso exista



alguma. Caso seja necessário a retirada de parafusos use uma chave de fenda/estrela. Limpe as conexões, o mini-cooler e toda a placa.

- Passe a borracha levemente nas vias de comunicação (parte dourada que conecta no slot). Não se deve tocar na parte dourada. Passe um pano, pincel ou papel higiênico sobre as hélices do mini-cooler presente na placa de vídeo e também sobre toda placa cuidadosamente. Deixe a placa em um local próximo envolvida em um pedaço de papel higiênico.

#### **[ETAPA 6]**

- Retire o HD do gabinete. Use ferramentas para retirada de parafusos e assim que removê-los guarde em uma dobra de papel com a devida identificação.
  - Desconecte o cabo sata (cabo vermelho) do HD e da placa mãe. Limpe com um pedaço de papel higiênico todo o HD. Com um pouco de álcool isopropílico, limpe as entradas do cabo sata do HD. Deixe-o em um local próximo.

#### **[ETAPA 7]**

- Retire a fonte do gabinete. Use ferramentas para a retirada de parafusos, e realize os mesmos procedimentos que o do HD para guardar temporariamente os parafusos em questão.
  - Limpe com um pedaço de papel higiênico ou um pano seco toda a superfície da fonte. Leve-a para o jato de ar e aponte para a parte interna da fonte (em direção ao cooler). Deixe-a em um local próximo.
  - Caso queira abrir a fonte para limpar melhor, retire os parafusos que conter nela e limpe-a, após esse procedimento feche-a e coloque a fonte em um lugar próximo.

#### **[ETAPA 8]**

- Cuidadosamente retire o processador da placa mãe. Limpe todo o conteúdo de pasta térmica, tanto presente no componente como no slot.
  - Após limpar a pasta térmica, coloque uma nova. Sempre pegue no processador pelas laterais e JAMAIS pelos contatos. Deixe o

processador em um local próximo envolvido em um pedaço de papel higiênico.

#### **[ETAPA 9]**

- Agora que já removeu todos os componentes periféricos da placa mãe, retire a placa mãe usando de ferramentas, agora coloque em um lugar próximo e seus parafusos também.
- Agora leve todo o gabinete para o jato de ar, e limpe-o por completo.
  - Direcione o jato para todos os cantos do gabinete e procure retirar toda a poeira existente.

#### **[ETAPA 10]**

- Limpe a placa mãe (slot da memória RAM, da placa de vídeo, do cabo sata) com pincel ou papel limpo, se preciso com álcool isopropílico, coloque-a em um lugar próximos para secar, caso tenha usado álcool.
  - Procure não encharcar a escova com o álcool. Passe pelos slots levemente.

#### **[ETAPA 11]**

- Reconecte todos os componentes que retirou. Passe novamente a pasta térmica no processador. Feche a tampa do gabinete.
  - Ao colocar o processador tome cuidado, pois ele tem uma identificação a ser colocada que é uma seta no processador e no slot/soquete dele.

## **Plano de Aula**

**Tema:** Herança e Reescrita de Métodos

**Duração:** 1 hora

**Período:** 3º semestre

### **Conteúdos**

- Conceito de Herança: Superclasse e Subclasse; Reescrita de Métodos e Testes.

### **Objetivos**

- Entender do conceito de herança: superclasse, subclasse, significado do objeto em herança, palavras reservadas como: this, super, override e reescrita de métodos;
- Compreender da sua importância e saber como utilizar o conteúdo de herança em código.

### **Metodologias**

- Acompanhamento da exposição do conteúdo de cada conceito sobre herança e reescrita de métodos utilizando o projetor com slide;
- Compreensão dos códigos práticos para melhor entendimento durante o conteúdo;
- Observação: aula será ministrada no laboratório de Computação como forma de interação e prática;
- Desenvolvimento do conhecimento e habilidades da disciplina: técnica usada one-minute paper (papel de um minuto). No final de cada aula será pedido aos alunos entregarem uma folha escrevendo (sem identificação) os assuntos mais importantes que aprendeu e os que não aprendeu. Nesse caso não terá uma nota, mas será útil para aprendizagem de ensino.
- Desenvolvimento do conhecimento e habilidades da disciplina: a técnica usada application article (artigo de aplicação). O professor irá avisar no início da aula que durante a aula o aluno irá escrever em um papel identificando seu nome, aplicações do mundo real de acordo com tal conteúdo aprendido e um breve resumo, calculando ao total 0,5 ponto.

- Reações aos Métodos de Instruções: técnica usada exam evaluations (avaliações de exame). Usar uma lista de exercícios equivalente ao tempo em sala de aula para ser feita em sala de aula como forma de prática e avaliação e no final solicite aos alunos que avaliem como o teste avalia seu conhecimento ou habilidades, como também sugestões calculando assim 1,0 ponto.
- Lista retro informativa: usar uma lista de exercícios mais elaborada que a usada no exame evaluations com a quantidade mínimo de alunos contidos em sala para ser feita em casa com um período de no mínimo uma semana. Em outra de aula (caso necessário dois dias) será entregue a lista e ser feita o sorteio de questões em relação a alunos para fazerem o código no quadro (podendo conter 2 a 4 alunos ao mesmo tempo no quadro) e sua explicação breve, calculando cerca de 1,5 ponto.
- Prova Individual: referente todo processo usado anteriormente, fazer de questões equivalentes usadas em sala de aula em uma prova individual que dure o tempo devido da aula, calculando o restante da nota 10 da unidade, ou seja, 7 pontos.
- Em caso de problemas ou necessidades a mais, será utilizado quadro negro, pincel e apagador.

## **Recursos**

- Projetor;
- Computador para o slide;
- Lista de exercícios usada na técnica exam evaluations (avaliações de exame);
- Lista de exercícios na técnica de retro informação;
- Papel com a prova feita;
- Laboratório de Computação;
- Slide com conteúdo exposto;
- Caso necessário: pincel, quadro negro e apagador.

## **Avaliações**

- Entendimento do conceito de herança: superclasse, subclasse, significado do objeto em herança, palavras reservadas como: this, super, override e reescrita de métodos;
- Compreensão da sua importância e saber como utilizar o conteúdo de herança em código.

# **Instrumentos Avaliativos**

**Período:** 3º semestre

## **Instrumentos Avaliativos**

### Conhecimento e habilidade da disciplina:

- One-Minute Paper (papel de um minuto)
  - No final de cada aula, cerca de 5 minutos faltando será pedido aos alunos entregarem uma folha escrevendo (sem identificação) os assuntos mais importantes que aprendeu e os que não aprendeu. Para que na próxima aula caso seja necessário ter uma breve revisão dos conteúdos (em sua maioria) de dúvidas que seja possível esclarecer tal assunto. Nesse caso não terá uma nota, mas será útil para aprendizagem de ensino.
- Application Article (artigo de aplicação)
  - O professor irá avisar no início da aula que durante a aula o aluno irá escrever em um papel identificando seu nome, aplicações do mundo real de acordo com tal conteúdo aprendido e um breve resumo. Dessa forma, ao final da primeira unidade será avaliado todos os papéis de cada aula feita e avaliada com 0,5 ponto.

### Reações aos Métodos de Instruções:

- Exam Evaluations (avaliações de exames)
  - Usar uma lista de exercícios equivalente ao tempo em sala de aula como forma de prática e avaliação. Porém no final da lista de exercícios solicite aos alunos que avaliem como o teste avalia seu conhecimento ou habilidades, como também sugestões calculando assim 1,0 ponto.
- Lista retro informativa.
  - Usar uma lista de exercícios mais elaborada que a usada no exame evaluations com a quantidade mínimo de alunos contidos em sala para ser feita em casa com um período de no mínimo uma semana. Em um dia de aula (caso necessário dois dias) será entregue a lista e ser feita o sorteio de questões em relações a alunos (obs: as questões de identificação de erros são escolhidas duas para cada aluno, em caso de construção de código apenas uma) para fazerem o código no quadro (podendo conter 2 à 4 alunos ao mesmo tempo no quadro) e sua explicação breve, calculando cerca de 1,5 ponto.

- Prova Individual

- Referente todo processo usado anteriormente, fazer de questões equivalentes usadas em sala de aula em uma prova individual que dure o tempo devido da aula, calculando o restante da nota 10 da unidade, ou seja, 7 pontos.

# Lista Exam Evaluations

Período: 3º semestre

Estudante: \_\_\_\_\_

## Exercício1:

Crie a classe Veículo com os atributos: Peso em quilos, VelocidadeMaxima em Km/h e Preco em R\$. E com os métodos: EntraVeiculo para entrada dos dados e MostrarVeiculo para mostrar os dados.

Crie a subclasse CarroPasseio usando a superclasse Veículo como base. Inclua os atributos Cor (string) e Modelo (string). E os métodos: EntrarPasseio para entrada dos dados e MostrarPasseio para mostrar os dados.

Crie a subclasse Caminhao usando a superclasse Veículo como base. Inclua os atributos Toneladas (carga máxima), AlturaMaxima (int) e Comprimento (int). E os métodos: EntrarCaminhao para entrada dos dados e MostrarCaminhao para mostrar os dados.

## Exercício 2:

Crie uma classe chamada Empregado que inclui três partes de informações como variáveis de instância:

- nome (String), sobrenome (String), ctps (String) e cpf (String).

Crie uma classe chamada EmpregadoHorista que herda da classe Empregado, contendo mais esses atributos: horasTrabalhadas.

Crie uma classe chamada EmpregadoAssalariado que herda da classe Empregado, contendo mais esses atributos: salarioMensal.

As três classes devem ter um construtor padrão e completo, métodos get e set para cada variável de instância e empregado só é igual pelo seu cpf, além do método toString.

Escreva um aplicativo de teste chamado EmpregadoTeste que cria dois objetos EmpregadoHorista e EmpregadoAssalariado e exibe o salário de cada objeto. Então dê ao EmpregadoHorista um aumento de 10% e ao EmpregadoAssalariado um aumento de 20% e exiba novamente o salário de cada Empregado.

Introduza na classe Empregado uma variável de classe capaz de contabilizar o número de empregados que passaram pela empresa até a data, ou seja, um contador de empregados.

## Lista Exam Evaluations

Período: 3º semestre

### Exercícios Finais:

De acordo com a lista de exercícios dada em sala de aula, como foi proveitosa em quesitos de aprendizado?

( ) Ótimo ( ) Bom ( ) Regular ( ) Péssimo

O quanto aprendeu com a prática de exercícios?

( ) 100% ( ) 50% ( ) 25% ( ) 0% ( ) Outro (escreva a porcentagem)

Conseguiu entender o intuito da lista de exercícios?

( ) Sim ( ) Não

Conseguiu captar a ideia principal do proposto das questões?

( ) Sim ( ) Não

Soube aproveitar o conteúdo teórico e prático da aula nos exercícios?

( ) Sim ( ) Não

A lista de exercício foi:

( ) Fácil ( ) Médio ( ) Difícil

Estipule seu nível de habilidade ao conteúdo teórico e prático:

( ) 100% ( ) 50% ( ) 25% ( ) 0% ( ) Outro (escreva a porcentagem)

Estipule o quanto ainda precisa estudar mais:

( ) 100% ( ) 50% ( ) 25% ( ) 0% ( ) Outro (escreva a porcentagem)

Sugestões (Comentários):

---

---

---

---

---

---



# Lista Retro Informativa

Período: 3º semestre

Estudante: \_\_\_\_\_

## Exercício1:

Identifique, explique e corrija os erros na classe abaixo caso possível.

```
public class Data
{
    private int ano;
    private int mes;
    private int dia;
}

public class Hora
{
    private int hora;
    private int minuto;
    private int segundo;
}

public class DataHora extends Data, Hora
{
    public DataHora(int ano, int mes, int dia, int hora, int minuto, int segundo)
    {
        super(ano, mes, dia);
        super(hora, minuto, segundo);
    }

    public String toString()
    {
        return super.toString() + " " + super.toString();
    }
}
```

## Exercício 2:

Identifique, explique e corrija os erros na classe abaixo caso possível.

```
public class Funcionario
{
    private String nome;
    private int cpf;
    private double salario;
}

public class Gerente extends Funcionario
{
    private double comissao;

    public Gerente(String nome, int cpf, double salario, double comissao)
    {
        super(nome, cpf, salario, comissao);
    }
}
```

## Exercício 3:

Cria uma classe Pessoa, contendo os atributos: String cpf e String nome. Faça também

o construtor padrão e completo. Considere, como subclasse da classe Pessoa, a classe Empregado. Considere que para classe Empregado além dos atributos de Pessoa, contém codigoSetor (inteiro), salarioBase (double) e imposto(porcentagem, double). Implemente na classe Empregado o método calcularSalario e o método toString (OBS: Nesse caso é opcional fazer o toString em Pessoa).

# Prova Individual - Completa

Período: 3º semestre

Estudante: \_\_\_\_\_

INSTRUÇÃO: Considere que os geteres, seteres, toStrings, clone e equals que não forem solicitados estejam disponíveis para utilizações seguindo os padrões de nomes tipo de retornos lista de parâmetros e implementação vistos durante as aulas. A questão que não pedir Batman, faça Robin.

## Questão 1 (2,5 pontos):

Elabore uma classe Conta, com os seguintes membros:

- ✓ atributo Cliente cliente
  - Contendo atributos devidamente de cliente.
- ✓ atributo int num\_conta
- ✓ atributo float saldo
- ✓ método sacar (o saldo não pode ficar negativo). Fica seu critério o retorno.

Agora acrescente ao projeto duas classes herdadas de ContaBancaria:

ContaPoupança e ContaEspecial, com as seguintes características a mais:

## Questão 2 (1,0 pontos):

- ☐ Classe ContaPoupança:
  - ✓ atributo int dia de rendimento
  - ✓ Crie o construtor completo definido.

## Questão 3 (2,0 pontos):

- ☐ Classe ContaEspecial
  - ✓ atributo float limite;
  - ✓ redefinição do método sacar, permitindo saldo negativo até o valor do limite.  
Fica seu critério o retorno.

## Questão 4 (0,5 ponto):

O método equals implementado na classe Conta deve ser reescrito na classe ContaPoupança e ContaEspecial? A correção da questão deve ser feita a partir da justificativa apresentada para a sua resposta.

## Questão 5 (2,0 ponto):

Crie a classe Banco contendo um array de Conta usando o método Robin e implemente o método booleano **adicionaConta** (Conta, utilizando o método Batman) que recebe uma conta como parâmetro e retorna se ele cadastrou ou não no banco, seu número de

conta deve ser gerado automaticamente com 5 números. O número de cada conta não se deve repetir.

**Questão 6 (1,0 ponto):**

Implemente o método booleano **buscaConta** (Conta) que recebe uma conta como parâmetro e retorna se ele se encontra ou não no banco.

**Questão 7 (1,0 ponto):**

Após a implementação das classes acima, você deverá implementar uma classe Contas.Java, contendo o método main. Nesta classe, você deverá implementar:

- a) Cadastrar no mínimo duas contas no banco do tipo especial e poupança;
- b) Sacar um determinado valor da(s) sua(s) conta(s);
- c) Verificar se uma das contas está no banco ou não;
- d) Mostrar os dados da(s) conta(s).

**Extra - Questão 8 (0,25 ponto):**

Implemente o método booleano **removeConta** (Conta) que recebe uma conta como parâmetro e retorna se ele removeu ou não no banco, só será removido a conta que tiver saldo zerado.

**Extra - Questão 9 (0,25 ponto):**

Implemente o método booleano **buscaContaPorSalario** (Conta) que recebe um valor mínimo e máximo como parâmetro e retorna um array de contas relacionado a esse intervalo de valores, neste caso o salário.

# Prova Individual - Definitiva

Período: 3º semestre

Estudante: \_\_\_\_\_

INSTRUÇÃO: Considere que os geteres, seteres, toStrings, clone e equals que não forem solicitados estejam disponíveis para utilizações seguindo os padrões de nomes tipo de retornos lista de parâmetros e implementação vistos durante as aulas.

## Questão 1 (3,5 pontos):

Elabore uma classe Conta, com os seguintes membros:

- ✓ atributo Cliente cliente
  - Contendo atributos devidamente de cliente.
- ✓ atributo int num\_conta
  - Esse atributo define que duas contas são iguais.
- ✓ atributo float saldo
- ✓ método sacar (o saldo não pode ficar negativo). Fica seu critério o retorno.

Agora acrescente ao projeto duas classes herdadas de ContaBancaria:

ContaPoupança e ContaEspecial, com as seguintes características a mais:

## Questão 2 (2,0 pontos):

- ☐ Classe ContaPoupança:
  - ✓ atributo int dia de rendimento
  - ✓ Crie o construtor completo definido.

## Questão 3 (2,0 pontos):

- ☐ Classe ContaEspecial
  - ✓ atributo float limite;
  - ✓ redefinição do método sacar, permitindo saldo negativo até o valor do limite.  
Fica seu critério o retorno.

## Questão 4 (0,5 ponto):

O método equals implementado na classe Conta deve ser reescrito na classe ContaPoupança e ContaEspecial? A correção da questão deve ser feita a partir da justificativa apresentada para a sua resposta.

## Questão 7 (2,0 ponto):

Após a implementação das classes acima, você deverá implementar uma classe Contas.Java, contendo o método main. Nesta classe, você deverá implementar:

- e) Instanciar no mínimo duas contas do tipo especial e poupança, uma por

construtor padrão e outra por construtor completo;

- f) Sacar um determinado valor da(s) sua(s) conta(s);
- g) Verifique e mostra mensagem se as duas contas são iguais.
- h) Mostrar os dados da(s) conta(s).

## Proposta

**Série:** 6 ano do fundamental

<b>Tipos de dados</b>	Reconhecer que entradas e saídas de algoritmos são elementos de tipos de dados. Formalizar o conceito de tipos de dados como conjuntos.
<b>Introdução à Generalização</b>	Identificar que um algoritmo pode ser uma solução genérica para um conjunto de instâncias de um mesmo problema, e usar variáveis (no sentido de parâmetros) para descrever soluções genéricas.
<b>Linguagem visual de programação</b>	Compreender a definição de problema como uma relação entre entrada (insumos) e saída (resultado), identificando seus tipos (tipos de dados, por exemplo: número, string etc.). Utilizar uma linguagem visual para descrever soluções de problemas envolvendo instruções básicas de processos (composição, repetição e seleção). Relacionar programas descritos em linguagem visual com textos precisos em português.
<b>Técnicas de solução de problemas: decomposição</b>	Identificar problemas de diversas áreas do conhecimento e criar soluções usando a técnica de decomposição de problemas.
<b>Fundamentos de transmissão de dados</b>	Entender o processo de transmissão de dados: a informação é quebrada em pedaços, transmitida em pacotes através de múltiplos equipamentos, e reconstruída no destino.
<b>Proteção de dados</b>	Atribuir propriedade (direito sobre) aos dados de uma pessoa ou organização. Identificar problemas de segurança de dados do mundo real e sugerir formas de proteger dados (criar senhas fortes, não compartilhar senhas, fazer backup, usar antivírus etc.).

## Fundamental 6 ano

### Tipos de dados

Quando você pensa em entrada de dados nada mais é do que você explicar a entrada (colocar) a água dentro do copo (variável), pois você está entrando com dados na variável e dessa forma guardando-a. Quando diz saída de dados, ou seja, quero ver meus dados, então funciona de pegar aquela água que está no copo e me mostrar sua quantidade ou cor, entre outras características dela.

Um exemplo é escrever seu nome em um papel e colocar em um copo (variável), ou seja, está lendo/entrando dados no copo (variável). Quando um dia alguém quiser saber seu nome é só ir pegar o papel dentro do copo, pois assim terá sua saída de dados do copo (variável). Explicando conceito de armazenamento de dados em variáveis, explicando seus tipos: string (texto), int (inteiro) e double (decimal).



Por exemplo a água e o sal é como se fosse double e int, ou seja, um pode colocar no outro, pois são compatíveis. Porém o óleo que é string não pode colocar na água que é int ou double, pois não são compatíveis. E se dá a ideia de armazenamento, pois o copo é a variável em si e o que tem dentro (por exemplo, água) é o conteúdo, o valor armazenado.

### **Introdução à generalização**

Vamos pensar em uma ação genérica que quase tudo existe, uma ação de adicionar coisas, ou seja, adiciono uma pessoa no meu perfil do facebook, adiciono leite na minha receita de bolo, adiciono caderno na minha mochila. Entrando em algo mais abstrato, adiciono alunos na escola (sistema), adiciona funcionários a empresa (sistema), entre outros.

Para a ideia de generalização entra a questão do adiciona, pois, adicionar basicamente é algo igual para todos, vou colocar leite na receita do bolo, vou colocar o caderno na mochila. Assim o ato se torna basicamente igual. Isso dá a ideia de que coisas/ações e específico da área, soluções, estão praticamente prontas as lógicas para apenas implementar/usar elas. E essa é a ideia de generalização, além de ser algo mais genérico, pois não estou especificando as particularidades de cada ação, apenas o ato adicionar.

Sempre quando se faz uma ação adiciona, precisa se algo para adicionar e esse algo chamamos de variável (tipo de dado). Dessa forma para entender o conceito de parâmetros, nada mais é do que explicar em exemplos práticos como adicionar leite na receita do bolo, o leite é algo “externo” a receita e além disso ele é algo, então além de ser uma variável (tipo de dado) se torna um parâmetro, pois é algo externo a receita de bolo que precisa ser adicionada.



### **Linguagem visual de programação**

Usando de jogos, pois são muito atrativos e fazem com que entre no mundo da programação, mesmo que não percebam, um exemplo é o robot turtles: <https://www.youtube.com/watch?v=Kb--aP9lCz8>. Além de muitos outros jogos online também: <https://studio.code.org/projects/public>

Uma linguagem visual muito utilizada para aprendizado iniciante é a scratch, pois o estudante visualiza melhor e consegue ver resultado de forma interativa, o que intensifica mais ainda o conceito. Mas de início não é bom ser usado, apenas no final do fundamental.

Nesse jogo do flappy bird que é muito jogado, mostra em algoritmo scratch bem simples de como funciona visualmente os códigos e também seu resultado, o que torna interessante a interação de programação com resultados, pode ser acessado por esse site:

<https://studio.code.org/projects/flappy/eaJ0FQV-RBpujxcQfLNRCiK6iN2e3CQzjxs2SazXgAc/edit>



Colocar textos em português para que o aluno consiga decifrar e resolver tal problema, exercícios de matemática lógica são ótimos para isso. Bem como, sua resposta também pode ser transformada em linguagem visual programada. Alguns desses exercícios bons para serem utilizados é da Olimpíada Brasileira de Informática que pode ser encontrado nesse site: <https://olimpiada.ic.unicamp.br/pratique/i1/2017/f1/submarino/>

### **Técnicas de solução de problemas: decomposição**

Alguns testes de solução de problemas são muitos complexos, assim o mais usual para esse ano do fundamental é usar o teste de mesa. Verificando primeiramente antes de fazer a programação (mesmo que visual) de forma lógica, ou seja, pensar em ideias de resolver aquele problema.

E o teste de mesa ajuda nesse quesito, pois ele é algo simples apenas usando papel e lápis, assim destrinchando os pontos de dificuldade para então achar tal solução e quando encontrar passar para forma programada.

Exemplo:

— INÍCIO;

Recebe o **VALOR1**;

Recebe o **VALOR2**;

**RESULTADO** é igual a **VALOR1** vezes o **VALOR2**;

Apresentar na tela o **RESULTADO**;

— FIM;

O teste de mesa pode ser visto de várias formas, bem como sua inicialização de resolução de problemas, mas também de finalização do problema, assim verificando, fazendo a prova se realmente tal código funciona ou não.

### **Fundamentos de transmissão de dados**

Como são armazenadas na memória, pois nas memórias existe caixinhas de espaços para entrar os dados e nem sempre são suficientes, então precisa quebrar o dado para salvar na memória. Quando faz essa quebra, são guardados os endereços dos dois de forma que associe os dois como um só. Na mesma ideia de recuperação dos dados, precisa do endereço que associe todos os pedaços daquele dado que queira recuperar, então eles irão se juntar e voltar para saída ou qualquer outra coisa que queira fazer com ele.

Um exemplo é dos copos: pense em vários copos (cada uma com um código de 0 à 1, pois já sabemos o que significa começar do 0 invés do 1) de 200 ml que ao todo compõe 2 litros da memória total, porém a quantidade de água (dados) que quer colocar é 750 ml, assim será colocado em 4 copos diferentes (nos quais são 0, 1, 2 e 3), sendo 3 cheios e 1 sobrando 50ml. Quando for querer toda a água de volta é só associar os códigos dos copos, ou seja, só pegar os copos de código 0, 1, 2 e 3 e retornar, dar a saída da água para onde for preciso.

Um exemplo mais técnico é a entrada de um arquivo de filme para o hd do computador. Tenho um filme de tamanho 100, cada caixa na memória tem 10 de tamanho, então preciso colocar em 10 caixinhas, para retornar acontece o mesmo, associo todas as caixinhas devido ao que quero que é o filme de tamanho 100.

Armazenar:

código do filme\_1: 0x

código do filme\_2: 1x

tamanho do filme\_1: 100

tamanho do filme\_2: 35

tamanho de cada posição das caixas de memória: 10

0-1x	1-1x	2-1x	3-1x0x	4-0x	5-0x	6-0x	7-0x	8-0x	9-0x
10f2	10f2	10f2	5f2 5f1	10f1	10 f1	10 f1	10 f1	10 f1	10 f1
10-0x	11-0x	12-0x	13	14	15	16	17	18	19
10f1	10f1	10f1	5f1						

Recuperar:

Filme\_2:

0-1x = tamanho 10

1-1x = tamanho 10

2-1x = tamanho 10

3-1x = tamanho 5

OBS: lembrando que na posição 3 está identificando cada tipo de dado, assim não terá problemas ao recuperar e nem armazenar os dados.

### **Proteção de dados**

Mostrar dados, notícias e as políticas de backup, bem como as empresas basicamente elas têm poder 100% pelos dados independentes se podem ser prejudiciais ou não. Além disso, mostrar que a quantidade de dados está crescendo e sua manipulação não é bem feita, assim os dados podem ser mal organizados, manipulados em forma de recuperação como vimos no conteúdo anterior.

Então alguns deveriam ser tomados, bem como amostragem de notícias:

Primeiramente fazer backup, tem muitas discussões sobre backup online e offline, dessa forma fica a critério do usuário escolher qual a melhor forma naquele momento para ele, caso tenha a possibilidade de fazer offline (ou seja, fisicamente com um hd externo), seja possível fazer dessa forma, garantindo mais a segurança .

Não acessar sites em computadores públicos com senhas que sejam mais importantes, como bancos, entre outros. Evitando assim algum tipo de roubo. Também com w-ifi público, fique atento ao usar senhas salvas automaticamente, pois elas podem ser manipuladas pela rede.

Crie senhas fortes, normalmente damos a dica de sempre colocar letras minúsculas, maiúsculas, números e algum tipo de caractere (o mais utilizado é o @). Além disso, o mais importante usar um bom antivírus, a escolha do antivírus entra em discussão também, mas dados retirados de notícias com seus desempenhos podem facilitar a escolha do usuário para escolher o mesmo.

## Referências

Morimoto, Carlos E. **Hardware, o guia definitivo**: 1. ed. Sulina, 2010.

Souza, Janaina S. **Montagem e Manutenção de Computadores: Curso Técnico em Manutenção e Suporte em Informática**. Manaus: Centro de Educação Tecnológica do Amazonas, 2011.

YOUTUBE. **Informática Básica: ¿Qué hay por dentro de un computador?**. Disponível em: <<https://www.youtube.com/watch?v=BFNScZBPek0>>. Acesso em: 18 jul. 2018.

Apostila FJ-11 da Caelum:

– Seção 7.6, página 94 (discussão sobre herança).

Deitel, Paul. **Java: como programar**. 10 ed. – São Paulo: Pearson Education do Brasil, 2017.

Sierra, K; Bates, B. **Use a Cabeça! Java**. 2 ed. Edição: Alta Books, 2012.

Lowa State University. Classroom Assessment Techniques: Quick Strategies to Check Student Learning in Class. Disponível em:

<<http://www.celt.iastate.edu/teaching/assessment-and-evaluation/classroom-assessment-techniques-quick-strategies-to-check-student-learning-in-class>>. Acesso em 27 out. 2018.

Geeks For Geeks. Inheritance in Java. Disponível em:

<<https://www.geeksforgeeks.org/inheritance-in-java/>>. Acesso em 27 out. 2018.

Em anexo no e-mail:

Livro Ensinando Ciência da Computação sem o uso do computador

Pdf Ensino de Computação na Educação Básica da SBC

## Anexos

