

# Mil Tecnologia Comércio e Serviços de Informática Ltda

## Teste Avaliativo para Estagiário

### Perfil da Vaga:

Estagiário de Desenvolvimento (Java e Ruby)

1. Formação
  1. Nível Superior em andamento nas áreas de Ciência da Computação, Informática, Sistemas de Informação ou equivalente.
2. Requisitos
  1. Ter cursado as disciplinas de Lógica de Programação e Banco de Dados
  2. Sólido conhecimento em lógica de programação;
  3. Linguagem de programação Java
  4. Ter noção e/ou alguma experiência acadêmica com SQL
3. Conhecimentos Desejáveis
  1. Conhecimento em Metodologias Ágeis de Desenvolvimento
  2. Testes Automatizados
  3. Linguagem de programação Ruby (Framework Rails)
  4. Linguagem de programação Javascript (Framework AngularJS)
  5. Git

### Passos para realizar o teste:

1. Escolha a Linguagem de Programação: Java OU Ruby. (Escolha exclusivamente uma, se duas forem submetidas iremos analisar somente a de Ruby).
2. Escolha o Banco de Dados: PostgreSQL 8.4 ou posterior;
3. Use o GitHub ou Bitbucket como Repositório GIT;
4. Escolha IDE ou Editor de Texto de acordo com suas preferências;
5. Use um editor de texto para responder questões que não sejam de programação. Salve em PDF e realize o *commit* com seu nome.
6. Após finalizado envie e-mail para [luiseugenio@miltecnologia.com.br](mailto:luiseugenio@miltecnologia.com.br) até 23/05/2016 ao meio-dia contendo link para o repositório.
  1. Todos os *commits* devem ter data anterior ao envio do e-mail.

### Critérios a serem Avaliados:

- Raciocínio Lógico;
- Domínio da Linguagem de Programação escolhida;
- Domínio de SQL;
- Testes implementados (Unitário, Integração, Controller, ...);
- Coesão e Legibilidade do código;
- Padrões utilizados;
- Organização dos *Commits*;
- Domínio da Língua Portuguesa.

### ATENÇÃO

As provas serão corrigidas de acordo com a ordem de entrega.

O *feedback* com o resultado da Prova será dado via e-mail em um prazo de até 10 dias.

### Questão 1: Lógica

Solucione o seguinte desafio:

Quatro matérias são ensinadas no Colégio EEMF Kleber Cavalcante Gomes.

Use as dicas e descubra quando, onde e quem dá cada uma das matérias.

		Matéria				Horário				Sala			
		Matemática	Português	História	Geografia	08:00:00	09:00:00	10:00:00	11:00:00	1	2	3	4
Professor	Junior			X									
	Joaquim			X									
	Luis			X									
	Marcelo	X	X	V	X								
Sala	1												
	2												
	3												
	4												
Horário	08:00:00												
	09:00:00												
	10:00:00												
	11:00:00												

#### Dicas:

- a) ~~Marcelo dá aula de História.~~
- b) As aulas acontecem em horários e salas diferentes.
- c) Entre o professor que dá aula às 11:00 horas e o professor que dá aula na Sala 3, um dá aula de História e o outro dá aula de Português, não necessariamente nessa ordem.
- d) A aula do Junior é duas horas antes da aula que acontece na Sala 3.
- e) A aula do Marcelo é às 11:00 horas.
- f) Marcelo NÃO dá aula na Sala 2.
- g) A aula de Matemática acontece mais tarde do que a aula que acontece na Sala 4.
- h) A aula de Geografia acontece uma hora antes da aula do Joaquim.

#### Resposta:

Professor	Matéria	Horário	Sala
Junior			
Joaquim			
Luis			
Marcelo	História		

## Questão 2. Implementação (Full Stack)

### Problema

Uma prefeitura está passando por problemas financeiros e não tem condições de fazer o pagamento integral da sua folha de pagamento. O prefeito juntamente com um conselho decidiu que iria pagar com a verba disponível apenas aqueles Servidores que possuem os menores salários, pois, segundo sua interpretação, eles seriam os mais necessitados.

Os Servidores receberão os salários de acordo com algumas considerações:

- Não havendo possibilidade de atender todos os Servidores de uma mesma faixa salarial, a prioridade é para aquelas com maior idade. Caso ainda haja empate, a prioridade é para aquelas cujo nome venha primeiro na ordem alfabética.
- Não existem Servidores com mesmo nome.
- A idade sempre será um Integer e estará restrito entre 18 e 60.
- O salário sempre será um BigDecimal e estará no restrito entre R\$1,00 e R\$10.000,00.
- As entradas não estão ordenadas alfabeticamente.

### Entrada

A entrada consistirá de vários casos de teste. Cada caso de teste é constituído por:

1. Uma linha contendo um BigDecimal **K** e um Integer **J**.
  - **K** representa a verba disponível para o pagamento da folha. Valor: ( $0 \leq K \leq 1000000$ )
  - **J** representa o número de servidores. Valor: ( $0 \leq J \leq 100$ )
2. As próximas **J** linhas serão apresentadas no formato:  
<NOME> <IDADE> <SALARIO>

A entrada termina quando **K** e **J** forem informados como zero.

### Exemplo:

5.0 5
Zilneide 29 10.5
Daniel 30 12
Alexandre 30 12
Erica 68 12
Carolina 25 5

### Saída

A saída deverá seguir rigorosamente o exemplo abaixo, indicando para cada K o nome dos colaboradores que receberão salário. Assim, para cada caso de testes deve ser gerada a seguinte saída:

- A primeira linha deve conter a cadeia "Teste N", onde N é representa o número do caso de teste começando em 1.
- As linhas seguintes são os nomes dos colaboradores em ordem alfabética.
- A última linha deve ser em branco.

### Exemplo:

Teste 1
Carolina

## Questão 2. a) Implementação em Java

Se você optou por utilizar a Linguagem de Programação Java o seu primeiro *commit* deve possuir a Mensagem: “Dou mó valor desenvolver em Java :D”.

- Utilização facultativa da plataforma Web ou Desktop.
- Bônus: Utilizar no Backend: VRaptor 4 e/ou JPA2.

### 1) Atendendo aos requisitos:

- Implemente o CRUD de Colaborador.
- Implemente o Processo de Entrada dos Dados. As validações devem ser Server e Client Side.
- Implemente o Processo de Saída para ajudar a Prefeitura a descobrir quem são os colaboradores que deverão receber os salários.
- Gere o Relatório de Saída em Jasper. (Só será aceito se os arquivos: “.pdf”, “.jrxml” e “.jasper” estiverem no commit)

### 2) Analise o Trecho de Código Java:

```
package br.com.miltecnologia.java;

/**
 *
 * @author luis
 */
public class Aplicacao {

    static public abstract class Operacao {

        public abstract int executar(int p1, int p2);

    }

    static public class Classe1 extends Operacao {

        public Classe1(String s) {
            System.out.println(s);
        }

        private void imprimir() {
            System.out.println("Imprimir X");
        }

        public int executar(int p1, int p2) {
            return p1 - p2;
        }

    }

    static public class Classe2 extends Operacao {

        public int executar(int p1, int p2) {
            return p1 * p2;
        }

    }

    static public class Classe3 extends Classe1 {

        public Classe3(String s) {
            super(s);
        }

        public static void processar(Operacao op, int p1, int p2) {
            System.out.println(op.executar(p1, p2));
        }

    }

    public static void main(String[] args) {
        Classe3.processar(new Classe2(), 5, 3);
    }

}
```

Tendo como base o código acima e as características da programação orientada a objetos em Java, é **INCORRETO** afirmar que o(a):

- Java não permite herança múltipla.
- método **imprimir** não está disponível a objetos criados para a **Classe3**.
- linha `System.out.println(op.executar(p1, p2))` irá exibir o valor “15”.
- método `super(s)` na **Classe3**, ao ser executado, cria uma instância da classe super.
- código demonstra o uso de polimorfismo.

## Questão 2. b) Implementação em Ruby

Se você optou por utilizar a Linguagem de Programação Ruby o seu primeiro *commit* deve possuir a Mensagem: “Ruby é só o filé! \0/”.

- Bônus: Utilizar no *Backend*: Rails 4.x.
- Bônus: Utilizar no *Frontend*: Foundation e/ou AngularJS.

### 1) Atendendo aos requisitos:

- Implemente o CRUD de Colaborador.
- Implemente o Processo de Entrada dos Dados.
- Implemente o Processo de Saída para ajudar a Prefeitura a descobrir quem são os colaboradores que deverão receber os salários.
- Gere o Relatório de Saída em Jasper. (Só será aceito se os arquivos: “.pdf”, “.jrxml” e “.jasper” estiverem no *commit*)

### 2) Analise o Trecho de Código Ruby:

```
1  class NivelDeAcesso
2
3      def executar
4          acesso = NivelDeAcesso.new
5          acesso.metodo_publico
6          acesso.metodo_protegido
7          acesso.metodo_privado
8      end
9
10     def metodo_publico
11         puts "Método Público. :D"
12     end
13
14     protected
15
16     def metodo_protegido
17         puts "Método Protegido. :)"
18     end
19
20     private
21
22     def metodo_privado
23         puts "Método Privado. :("
24     end
25 end
26
27 NivelDeAcesso.new. executar
```

Tendo como base o código acima e as características da linguagem de programação **RUBY**, é **INCORRETO** afirmar que o(a):

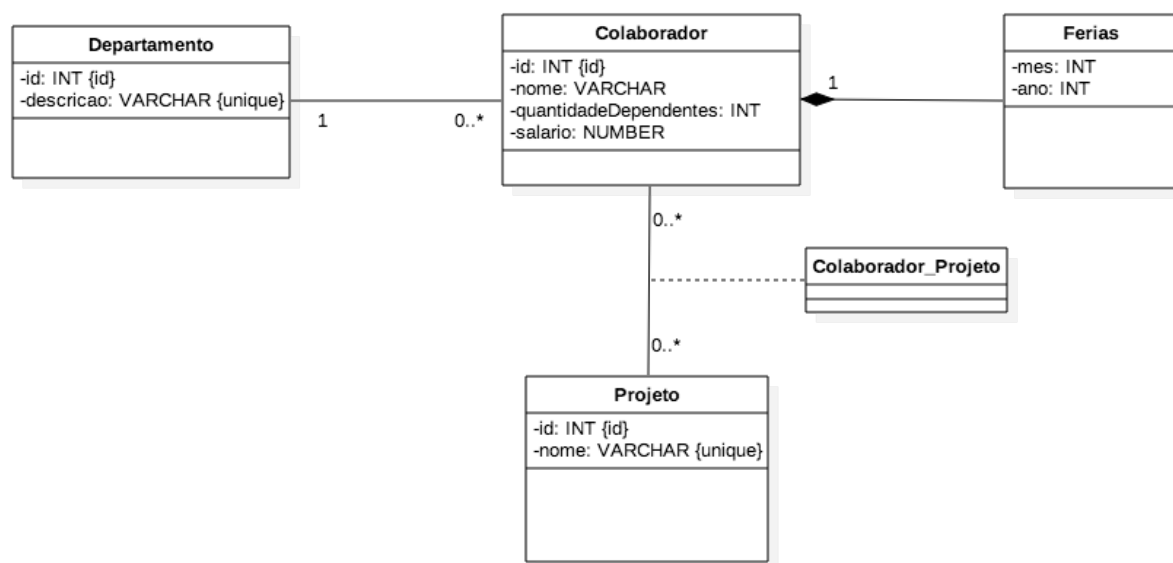
- em Ruby só existem 3 níveis de acesso a métodos.
- métodos Protegidos só são acessíveis para outros objetos da mesma classe.
- nem todos os Métodos Públicos podem ser acessados de dentro da mesma classe.
- execução da Linha 6 irá exibir o valor “#=> Método Protegido. :)”
- execução da Linha 7 vai resultar em Exception.

## **Questão 3. Testes**

**Utilizando ainda os dados do Problema da questão 2 e de acordo com a Linguagem de Programação escolhida:**

- a) Implemente os Testes, que julgar necessário, que demonstrem a validação dos campos obrigatórios em Colaborador
  - b) Implemente os Testes, que julgar necessário, do método que informa quem são os colaboradores que devem receber o salário em cada Caso de Teste.
- 
- Bônus: Implemente os Testes de Client Side para a a Validação do Processo de Entrada de Dados. (1ª questão. Item B)

## Questão 4. SQL



### Observações

- Os atributos que compõem a chave primária nas relações estão marcados com {id}.
- Um Colaborador pode tirar nenhum, um ou mais períodos de Férias, mas somente um período por ano. Cada período de Férias deve obrigatoriamente pertencer a um Colaborador.
- Um Projeto pode ter nenhum, um ou mais Colaborador, e um Colaborador pode participar de nenhum, um ou mais Projetos.
- Um Colaborador\_Projeto deve pertencer obrigatoriamente a um único Projeto e a um único Colaborador.

### 1) Crie script SQL que:

- a) Crie a tabela Colaborador (DDL).
- b) Insira na tabela Departamento os valores 15 e “Desenvolvimento e Pesquisa”, respectivamente, nos atributos “id” e “nome”.
- c) Liste os nomes dos colaboradores do departamento cujo identificador do departamento é 15 e que não tiraram férias no ano de 2014?
- d) Traga o total de colaboradores e a média de salários?
- e) Liste os nomes de projetos e seus respectivos nomes de colaboradores, ordenados por nome do projeto e nome do colaborador? Caso o projeto não possua colaboradores associados, somente o nome do projeto deve aparecer.

### 2) De acordo com o Diagrama:

- a) Quantos Colaboradores um Departamento pode possuir?
- b) Que tipo de Associação existe entre “Colaborador” e “Ferias”. Porque?

## **Questão 5. Redação**

**Quais os motivos que o levaram a escolher a Mil Tecnologia para fazer estágio?  
Elabore um texto descritivo com, no mínimo, 1500 e, máximo, de 2500 caracteres.**