



## **Ambientação Java 470**

## Sumário

Capítulo 1	
Primeiro capítulo.....	3
1.1. Tópico.....	3
REFERÊNCIAS	
BIBLIOGRÁFICAS.....	5
ANEXOS.....	7
Primeiro anexo.....	8
Tópico do Anexo .....	8

## Índice de tabelas

## Índice de Figuras

# **Instruções**

## **de Aulas, Exercícios e Testes**

### **1.1. Objetivos:**

- Entender a metodologia para o curso Java EAD do curso 470
- Sobre a execução do exercícios
- A importância dos Pré-Aulas e Conteúdo Obrigatório
- A correção automatizada dos exercícios

## 1.2. Introdução

Na ambientação metodologia EAD da 4Linux você conheceu como funciona o ambiente de EAD, o que é uma sala, o que o aluno deve fazer antes da aula, pos-aula e como estes serão avaliados pela 4Linux para obter o certificado do curso. Nesta apostila vamos fazer uma ambientação para o curso de Java que pertence a área de desenvolvimento e sofre algumas pequenas mudanças no sentido técnico e é diferente de qualquer outro curso EAD no mercado. Então leia todas as informações contidas aqui com cautela, anote as dúvidas e *post* no fórum do curso, assim o instrutor poderá tirar sua dúvida e quem sabe será a mesma dos seus outros colegas.

## 1.3. Entendendo a metodologia para o curso Java EAD

O que você vai precisar entender para poder ir avançando no curso e ir acompanhado o ritmo dos colegas é compreender de fato o processo do curso. Há uma ciclo de execução que deve ser cumprido pelo aluno para que este possa aprender da melhor forma possível o que foi contratado. E antes de querer colocar “mão na massa” à nível de código, ferramentas etc. O aluno precisa compreender o que fazer em cada etapa do curso, do contrário, há um risco deste ficar perdido. Por isso que este documento foi desenvolvido para orientar o aluno antes de tudo.

Você precisa entender que no curso temos isso:

- **Pré-Aula:** como o próprio nome já diz, você deve ler o conteúdo do pré-aula, antes de ir para a Aula. E também verificar se não há *Conteúdo Obrigatório* que pode envolver instalações de ferramentas, que serão usadas no momento *Síncrono* ou no exercício *Pos-aula*. Aqui o aluno vai encontrar passo-a-passo como instalar/configurar, ferramentas, além de um bom conteúdo teórico e prático suficiente para você ir para aula entendendo à nível básico o que está sendo discutido. Para as instalações que consideramos um pouco mais complexa, criamos um vídeo-aula para facilitar a vida nos nossos alunos.
- **Síncrono:** após ter lido o conteúdo prévio, no pré-aula, haverá o momento com o instrutor para aprofundamento do assunto. Não será uma repetição do que viu no pré-aula, e sim continuação do que não

colocamos lá.

- Pós-Aula: Aqui será os exercícios que vai receber, após o momento síncrono e que o aluno deve fazer. Até a data final do curso. Todos os exercícios do *pos-aula* tem uma nota que vai somar para ter a média final de aprovação de 60% (6 pontos) para obter o certificado. Esses exercícios eles vão avançando a medida que o curso toma esse rumo. E como estamos preparando o aluno para o mercado de trabalho, não espere exercício muito simples como acontece em ambientes academicos, forçamos ao aluno ter um problema e através de um base de conhecimento adquirida no curso busca o caminho da solução e com um pouco de proatividade do aluno este deve encontrar a solução. Isso é o que acontece em no cenário corporativo do desenvolvimento de software: Há problemas e nos temos que encontrar a solução.
- Complementar: Aqui vão entrar tópicos que podem ser artigos, exercícios ou referências sobre um determinado assunto, caso o aluno queira se aprofundar mais. É opcional, fazer o que está no complementar.

## 1.4. A importância do Pré-Aula e Conteúdo Obrigatório

O pré-aula é tão importante quanto estar presente no momento síncrono. Nele o aluno vai encontrar todas as informações que serão necessárias para que ele possa discutir, aprender durante o momento síncrono ou pós-aula. É comum alguns alunos ignorar o momento pré-aula e dar mais importância para o síncrono e pós-aula, porém na metodologia da 4LINUX cada momento tem um objetivo a ser cumprido no processo de aprendizado dos nossos alunos, então não pule nenhuma as etapas nem classifique qual delas são as mais importantes, e tenha em mente que TODAS são importantes.

O conteúdo obrigatório é como o próprio nome já diz, é OBRIGATÓRIO ser feito, do contrário o aluno não conseguirá avançar. Quer um exemplo? Você vai conseguir programa em Java se ter o Java JDK instalado? Não. Então são informações desse tipo que você vai encontrar em conteúdo obrigatório da aula. E a 4Linux não faz toda instalação das ferramentas de uma só vez, vamos instalando a

medida que a necessidade vai surgindo de forma natural no curso, pois é assim no mundo do desenvolvimento de software. Não montamos um servidor JEE se não sabemos e de fato ele será necessário no projeto XYZ.

### **1.4.1. Sobre os exercícios**

Os exercícios no curso Java da 4Linux é bastante diferente do que estamos acostumados a fazer na faculdade ou em outros cursos técnicos Java. Primeiro que nossos exercícios são focados em situações reais que um programador passa no dia-dia, são exemplos e especificações do “mundo real” e não exercícios simples ou incompletos. Você como programador Java deve lembrar que em um ciclo de desenvolvimento de software, será bem difícil um único programador desenvolver toda a solução. Este será responsável por desenvolver módulos ou partes específicas de um projeto, enquanto outros programadores estarão construindo outra parte e para a integração entre os profissionais envolvidos há várias ferramentas que ajudam nesse processo, e pensando nisso resolvemos mudar “radicalmente” a forma de como dar um bom exercício aos alunos e que eles sintam durante o curso como será a vivência deles desenvolvendo software em equipe.

Então, ao invés de receber o que deve ser feito (a especificação) dentro da apostila ou em um slide, vamos usar uma ferramenta que é adotada em vários projetos que tem a funcionalidade de manter todos atualizados sobre as tarefas que estão ocorrendo e as que estão já projetadas para vir. Usaremos o RedMine ([www.redmine.org](http://www.redmine.org)) e nela é onde o aluno após ler o conteúdo da apostila, vai acessar e verificar o ticket que foi aberto pelo instrutor, onde há uma especificação do que deve ser feito e o prazo de conclusão. Após o aluno ler a especificação do que deve ser feito e ele deve baixar o código-fonte do repositório SVN, implementar a solução no computador local dele, e depois enviar para o SVN atualizado.

Cada aluno terá seu próprio *branch* no SVN algo como: NomeAluno-NomeExercicio-branch veja um exemplo: *Camilo-ImpostoRendaAula1-Branch*

O aluno não poderá fazer checkout do branch do colega, a ferramenta mantém o histórico de todas as atividades no SVN, caso isso seja feito o aluno que fez o checkout não terá mais nota para aquele exercício. O envio do exercício para o instrutor é somente via ferramenta SVN, jamais por qualquer outro meio tais como: e-mail, fórum etc.

Os exercícios sempre tem um esqueleto que não pode ser alterado pelo aluno,

ele deve manter como está e dar continuidade, uma vez que usamos o recurso de Junit para que o próprio aluno possa testar se o exercício está correto ou não antes de enviar. Então ao receber o exercício focaremos aluno apenas a desenvolver a solução para atender a especificação do cliente. E onde que nos programadores passamos maior parte do tempo, buscando soluções para o problema de alguém.

Observação: Não se preocupe, na pre-aula da aula 1, há uma apostila explicando a instalação/configuração das ferramentas, também criamos algumas video-aula mostrando as coisas acontecendo. Mas, até chegar lá você precisa entender sobre este processo, do contrario poderá ter dificuldades em operar com as ferramentas dentro da metodologia do curso da 4Linux.

## 1.5. A correção automatizada dos exercícios

A 4Linux sempre está inovando em soluções e para o curso de Java, há uma inovação bem interessante para os nossos alunos, que são os : “Exercícios automatizados”. Como funciona?

O funcionamento é muito simples, você aluno será capaz de saber se já resolveu o próprio exercício sem precisar de uma análise previa do instrutor. Vamos resumir primeiro e depois por o passo-a-passo.

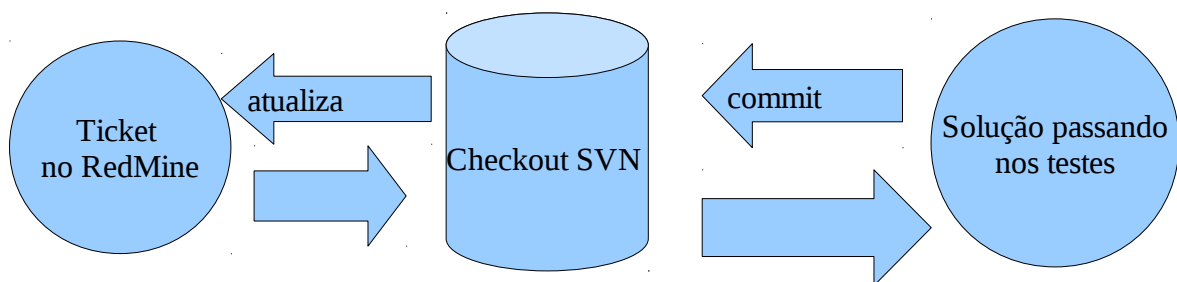
Resumindo: Após o momento síncrono com o instrutor, o aluno terá que fazer o *checkout* (download) do exercícios do repositório SVN. O aluno receberá apenas a estrutura do projeto e uma classe que terá no final a palavra *Test*, pois ela serve para testar a solução do aluno. Esta classe NÃO PODE ser manipulada pelo aluno, ela é apenas para que você tenha certeza que atendeu todos os requisitos do exercício e já pode enviar o instrutor. Pois, quando um exercício está atendendo todos os requisitos e a classe teste for executada, aparecerá uma “sinal” verdade na aba de testes no Eclipse.

Passo-a-passo:

1. Acessar a ferramenta de bugtrack e verificar a especificação do que deve ser feito no exercícios. O nro do ticket do exercício é passado na aula síncrona.

2. Fazer o checkout do repositório SVN, ou seja, fazer o “download” do esqueleto do exercício para a sua máquina.
3. Não alterar a classe de *Test* do contrário, não saberá se de fato você resolveu o exercício, uma vez que os testes já foram desenvolvidos para esse fim, e não podem ser alterado pelo Aluno.
4. Desenvolver a solução para o exercício
5. O aluno faz o *commit* para o seu próprio *branch*
6. Atualiza o ticket no RedMine, respondendo que já concluiu o exercício e informa o nro da revisão (*revision*). Uma vez feito isso não será considerado nenhuma outra *Revision* além da que consta no ticket do exercício.

Um exercício concluído só é considerado completo quando o aluno faz o ciclo do curso que é:



Observação: O instrutor não tem obrigação de corrigir o exercício assim que for enviado pelo aluno, mas este pode ser feito pelo mesmo para poder dar um retorno mais rápido.