

**Aula 00 (Prof.
Evandro)**

*TRT-Campinas 15ª Região (Analista
Judiciário - Tecnologia da Informação)
Arquitetura de Computadores - 2024
(Pós-Edital)*

Autor:

Evandro Dalla Vecchia Pereira

05 de Dezembro de 2024

Índice

1) Apresentação do Curso - Evandro	3
2) Apresentação Flashcards	6
3) Servidores de Aplicação JBoss Wildfly - Teoria	8
4) Servidores de Aplicação JBoss Wildfly - Questões Comentadas - Multibancas	16
5) Servidores de Aplicação JBoss Wildfly - Lista de Questões - Multibancas	22



APRESENTAÇÃO DO CURSO

Iniciamos nosso **Curso Regular de Sistemas Operacionais e Arquitetura de Computadores** em teoria e questões, voltado para provas **objetivas e discursivas** de concurso público. Tais assuntos são cobrados em diversos concursos em que há vagas específicas para a área de TI.

As aulas em PDF possuem por característica essencial a **didática**. Ao contrário do que encontramos em alguns livros, o curso todo se desenvolverá com uma leitura de fácil compreensão e assimilação.

Além disso, teremos videoaulas! Essas aulas destinam-se a complementar a preparação. Quando estiver cansado do estudo ativo (leitura e resolução de questões) ou até mesmo para a revisão, abordaremos alguns pontos da matéria por intermédio dos vídeos. Com outra didática, você disporá de um conteúdo complementar para a sua preparação. Ao contrário do PDF, evidentemente, **AS VIDEOAULAS NÃO ATENDEM A TODOS OS PONTOS QUE VAMOS ANALISAR NOS PDFS, NOSSOS MANUAIS ELETRÔNICOS**. Por vezes, haverá aulas com vários vídeos; outras que terão videoaulas apenas em parte do conteúdo. Nosso **foco** é sempre o **estudo ativo!**



APRESENTAÇÃO PESSOAL

Meu nome é Evandro Dalla Vecchia Pereira, sou autor do livro "Perícia Digital - Da investigação à análise forense", Mestre em Ciência da Computação (UFRGS), Bacharel em Ciência da Computação (PUCRS), Técnico em Redes de Computadores (Ecom/UFRGS) e em Processamento de Dados (Urcamp). Perito Criminal na área de Perícia Digital desde 2004 no Instituto-Geral de Perícias/RS. Professor de pós-graduação em diversas instituições, nas áreas de Perícia Digital, Perícia Criminal e Auditoria de Sistemas. Lecionei em cursos de graduação de 2006 a 2017, nas instituições PUCRS, Unisinos, entre outras e sou professor em cursos de formação e aperfeiçoamento de Peritos Criminais, Delegados, Inspetores, Escrivães e Policiais Militares.

No Estratégia Concursos leciono desde o começo de 2018, inicialmente na área de Computação Forense e, na sequência, também assumi as áreas de Arquitetura de Computadores e Sistemas Operacionais, tanto na elaboração de materiais escritos como na gravação das videoaulas.

Deixarei abaixo meus contatos para quaisquer dúvidas ou sugestões. Terei o prazer em orientá-los da melhor forma possível nessa caminhada que estamos iniciando.

Instagram: @profevandrodallavecchia

Facebook: <https://www.facebook.com/profevandrodallavecchia>



PARE TUDO! E PRESTE ATENÇÃO!!

Hoje eu faço parte de uma equipe **SENSACIONAL** de professores! Depois de muita luta conseguimos reunir **um time** de profissionais extremamente **QUALIFICADO** e sobretudo **COMPROMISSADO** em fazer o melhor pelos alunos. Para tal criamos um conjunto de ações para nos aproximarmos dos alunos, entendermos suas necessidades e evoluirmos nosso material para um patamar ainda mais diferenciado. São 3 as novidades que gostaria de convidá-lo a conhecer:

//estratégia tech



Nosso podcast alternativo ... livre, descontraído e com dicas rápidas que todo CANETA PRETA raiz deve ouvir. Já temos alguns episódios disponíveis e vários outros serão gravados nas próximas semanas ... acompanhe em:

<http://anchor.fm/estrategia-tech>



Telegram

a new era of messaging

Nosso grupo do Telegram é um local onde ouvimos os alunos e trocamos ideias com eles. Está crescendo a cada dia. A regra do grupo é: só vale falar sobre concursos. Lá divulgamos nossas aulas ao vivo e falamos sobre os concursos abertos, expectativas de novos concursos, revisões de véspera, e por aí vai...

http://t.me/estrategia_ti

Instagram



Criamos um perfil no Instagram ... e qual o objetivo? Fazer com que os alunos percam tempo nas redes sociais? Claro que não!! Estamos consolidando diversos posts dos professores! São dicas especiais, um patrimônio que deve ser explorado por todos os concurseiros de TI!

<http://instagram.com/estrategiaconcursosti>



ESTRATÉGIA FLASHCARDS

📖 Você tem dificuldade de estudar, memorizar e revisar os conteúdos que estuda em nossas aulas? Então nós temos a ferramenta perfeita para você!

Apresentamos o **Estratégia Cards**: app de flashcards que vai revolucionar sua forma de **estudar** e **revisar** conteúdos de provas de concurso público. Com nossa tecnologia inovadora e interface amigável, você dominará os tópicos mais complexos de maneira eficiente e divertida.

🌟 Recursos do Estratégia Cards:

Curadoria de Flashcards	Flashcards criados e revisados por professores especializados em cada área, com qualidade e voltados para concursos públicos.
Flashcards Personalizados	Crie seus próprios flashcards, cobrindo os principais tópicos e matérias dos concursos públicos.
Repetição Espaçada	Técnica de aprendizagem que envolve revisar informações em intervalos crescentes para melhorar a retenção de longo prazo e combater o esquecimento.
Estatísticas Personalizadas	Visualize graficamente o percentual de acertos, erros ou dúvidas dos decks estudados.
Modo Offline	Estude em qualquer lugar, mesmo sem conexão à internet, fazendo o download dos decks.
Estudo por Áudio	<i>Está dirigindo ou fazendo esteira e quer continuar estudando?</i> Basta utilizar a opção “Escutar”.
Decks Favoritos	Você pode escolher decks específicos como favoritos e visualizá-los em uma aba separada do app.
Opções de Estudo	Você poderá estudar todos os cards de um deck; ou apenas os que você errou; ou apenas os que você não estudou ainda; entre outras opções.

📱 E como eu consigo baixar?



É muito fácil! Basta pesquisar por “Estratégia Cards” na loja oficial do seu smartphone.

Se você tiver um Android, basta acessar a **Google Play**;



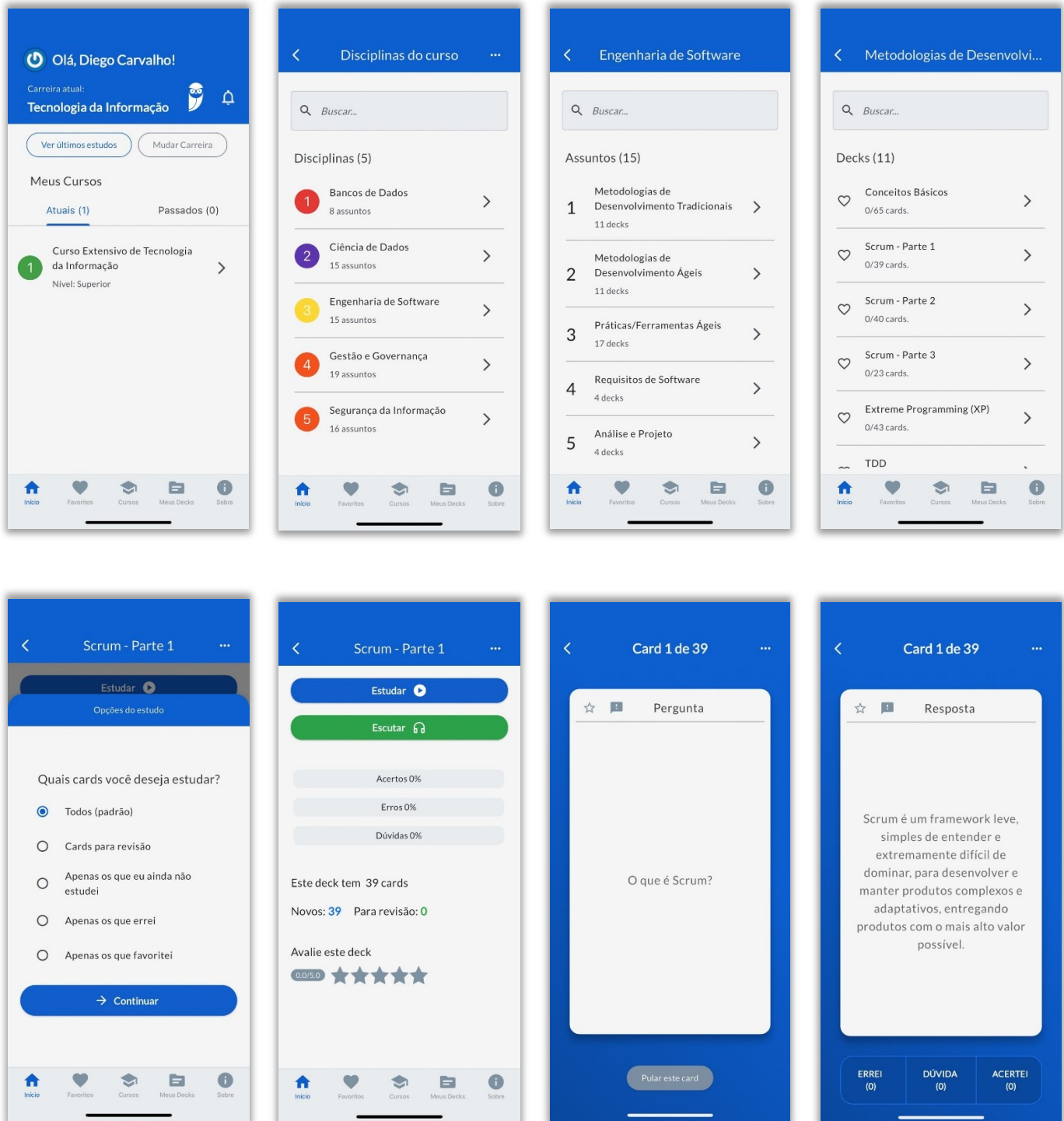
Se for tiver um iPhone, basta acessar a **App Store (iOS)**.



É para acessar?

Para acessar, basta ter uma conta no Estratégia Concursos. Em seguida, utilize suas credenciais de login e senha para acessar o aplicativo. Por fim, acessa a carreira de Tecnologia da Informação.

Como utilizar o app:



SERVIDOR DE APLICAÇÃO JBoss/WILDFLY

JBoss é um servidor de aplicação de código fonte aberto baseado na plataforma JEE, é implementado completamente na linguagem de programação Java, podendo ser utilizado em qualquer sistema operacional que suporte essa linguagem.

O JBoss Application Server (AS) é a versão *open source*, desenvolvida em comunidades JBoss/RedHat, traz inovações em um ritmo mais rápido com foco em novas funcionalidades, mas seu principal diferencial é que ele não possui suporte oficial.

O JBoss Enterprise Application Platform (EAP) é a versão paga, normalmente evolui a partir das inovações das versões estáveis do JBoss AS, é integrada com recursos como o JBoss Developer Studio e o com o JBoss Operations Network, após um processo de testes (desempenho, escalabilidade etc.), além de possuir suporte oficial.

O JBoss é uma plataforma de middleware¹ baseada em padrões abertos, e que mantém conformidade com a especificação Java EE. É servidor de aplicação que provê recursos de alta disponibilidade, *clustering*, mensageria, *cache* distribuído, entre outros. Além disso, ele também inclui APIs e *frameworks* de desenvolvimento para aplicações Java EE escaláveis.

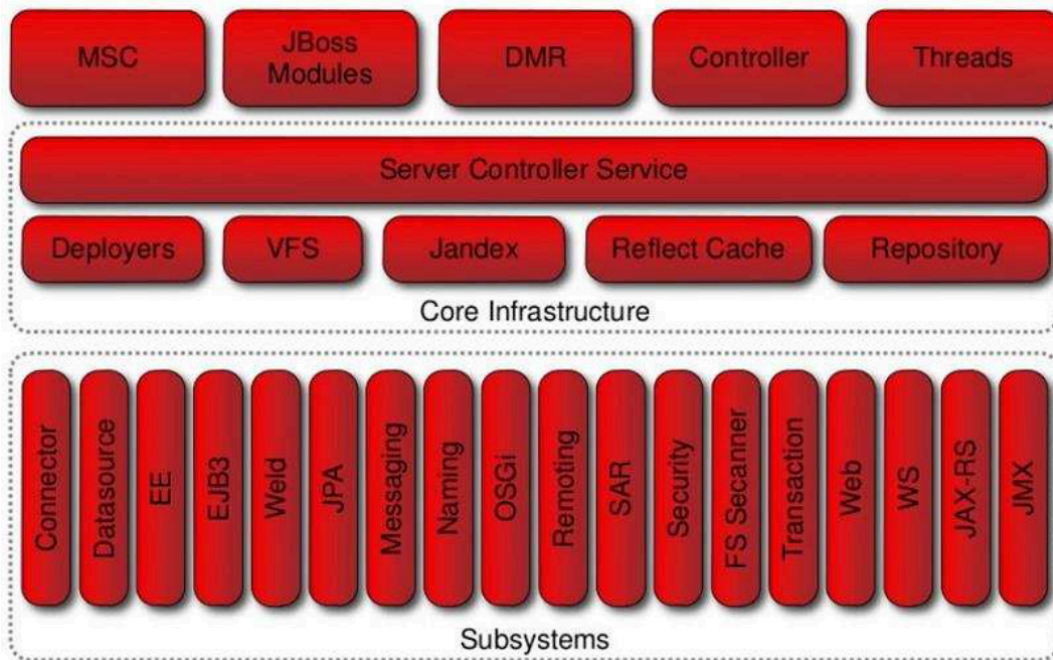
A partir da versão 8 o JBoss passou a se chamar Wildfly, além de ter várias melhorias e mudanças como a troca do *container* que era o JBossWeb para o Undertow e a utilização de módulos. O objetivo principal dos módulos é fornecer um mecanismo de isolamento e organização para os componentes do servidor e das aplicações implantadas. O **JBoss Modules** permite que os módulos sejam carregados e descarregados dinamicamente, o que torna a utilização de recursos otimizada.

Arquitetura e Modos

Na figura a seguir podemos ver a arquitetura do JBoss/Wildfly, cabendo um destaque aos subsistemas, onde encontramos algumas APIs JEE, tais como EJB, JPA, JMX, Mensageria, entre outros.

¹ Software que se encontra entre o sistema operacional e os aplicativos nele executados, funcionando de forma essencial como uma camada oculta de tradução. Permite a comunicação e o gerenciamento de dados para aplicativos distribuídos.





Com essa figura fica claro que o JBoss é um servidor de aplicação que implementa diversas características da especificação JEE. Um conceito essencial nas últimas versões é o de domínio. Em um servidor JBoss, um domínio fornece gerenciamento centralizado de várias instâncias de servidor e hosts físicos, enquanto um servidor standalone permite uma única instância do servidor. Configurações, implantações, socket, módulos, extensões e propriedades do sistema podem ser gerenciadas para grupos de servidores. Vamos ver mais detalhes desses dois modos a seguir.

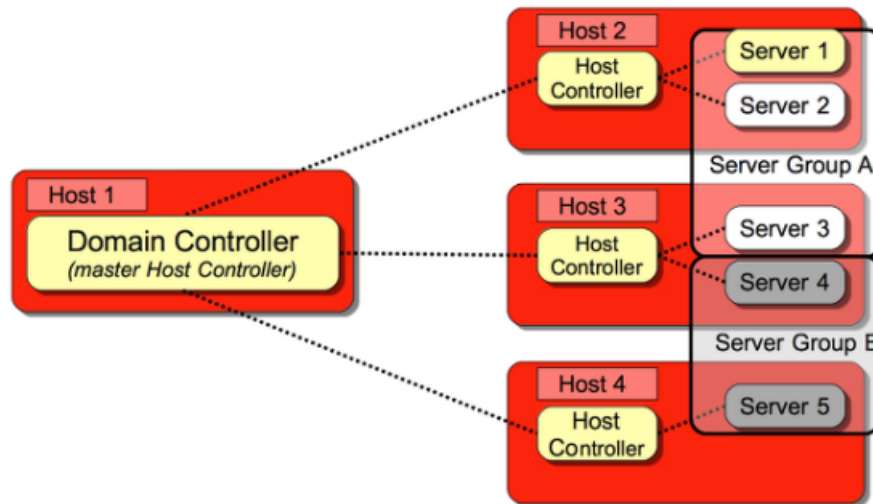
Modo Standalone: é o modo tradicional das versões anteriores. Basicamente implica em ter uma instalação diferente (ou um diretório standalone diferente) para cada instância de Wildfly. Ou seja, para cada Wildfly em execução no seu ambiente é necessário alterar seus próprios arquivos de configuração, suas próprias opções de execução para JVM etc. A inicialização nesse modo é realizada ao iniciar o script `JBOSS_HOME/standalone.sh` (no Linux) ou `JBOSS_HOME/standalone.bat` (no Windows).

Modo Domain: é o modo que foi introduzido no JBoss AS 7, permitindo gerenciar um conjunto de instâncias Wildfly, agrupando-os e assim permitindo compartilhar configurações comuns entre eles. Além de compartilhar configurações, é possível também através de um único console de gerenciamento iniciar ou parar instâncias (ou grupos inteiros), verificar seu status e estatísticas de cada subsystem etc. O gerenciamento das instâncias é coordenado pelo Domain Controller, tendo várias instâncias (JVMs) por Host e o controle total do ciclo de vida dos servidores via Host Controller. Para iniciar no modo domain é executado o script `JBOSS_HOME/domain.sh` no Linux ou `JBOSS_HOME/domain.bat` no Windows.

Quando o `domain.sh` (ou o `domain.bat`) é executado em um host, um processo conhecido como Host Controller é lançado, o qual é responsável pelo gerenciamento do servidor. Ele não lida diretamente com a carga de aplicações no servidor, sendo responsável por iniciar e terminar os processos que rodam em cada servidor, interagindo com o Domain Controller para ajudar no gerenciamento.



Por padrão, cada Host Controller lê sua configuração no arquivo domain/configuration/host.xml, que possui informações de configuração específicas do host. Abaixo podemos ver uma figura com quatro hosts, um Domain Controller e três Host Controllers, sendo que cada Host Controller pode ter um ou mais servidores.



A figura acima mostra uma configuração em modo domain, sendo que todas as configurações e o gerenciamento são realizados de forma centralizada, no Domain Controller. Todas as configurações realizadas para o Domain são replicadas nas instâncias JBoss. Adicionar um novo grupo de servidores, configurar logs, criar data sources, alterar portas, entre outras coisas, tudo isso é feito no [domain.xml](#) do arquivo de configuração do Domain Controller (master).

Características como *clustering*, *high availability* (HA), *fail-over* e outros recursos do JEE estão disponíveis nos dois modos. Domain Controller é quem controla o gerenciamento do *domain*. Nele estão as configurações que são compartilhadas entre as instâncias que estão nesse *domain*, e a política de gerenciamento de todos os servidores. O Domain Controller é basicamente um processo Host Controller que dependendo da arquitetura se torna o Domain Controller.

Foi reduzida a necessidade de editar arquivos de configuração XML manualmente. O gerenciamento da segurança é realizado de forma simplificada, inclusive para domínios de segurança. O diretório "modules" centraliza os módulos do servidor de aplicações, em vez do diretório "lib". Os diretórios "domain" e "standalone" possuem os arquivos de configurações para *deploys* (implantações) em modo domain e standalone.

O mecanismo de carga de classe é modular, então os módulos são carregados e descarregados sob demanda, o que propicia uma melhora na segurança, além de menores tempos para iniciar ou reiniciar o servidor. Duas bibliotecas iguais de versões diferentes podem conviver no mesmo servidor.

Instalação, Configuração e Implantação

Para a instalação, existe a opção de baixar o JBoss/Wildfly em binário, em arquivo zip ou com um instalador. A maneira mais rápida é fazer o download do binário, e descompactar em um diretório. Há alguns riscos que devem ser evitados na instalação, para evitar falhas de segurança.



Assim, após a instalação, há a necessidade de ajustar e personalizar a configuração do servidor, antes de colocar o servidor em produção.

O pré-requisito para instalar o JBoss é o Java Development Kit (JDK) ou Java Runtime Environment (JRE) instalado. O download do JBoss/Wildfly pode ser realizado no site da RedHat. Após o download, os arquivos de instalação devem ser colocados na pasta adequada, para iniciar a instalação.

Após a instalação, é necessário configurar o JBoss/Wildfly como serviço. Não é recomendado iniciá-lo, e deixá-lo configurado com o usuário root, pois isto pode comprometer a segurança de toda a plataforma, já que a plataforma Java oferece APIs para execução de códigos nativos do sistema operacional e mecanismos de gerenciamento remoto.

O ideal é que no Linux seja criado um usuário com privilégios adequados para iniciar o serviço do JBoss, e no Windows seja criado um usuário com poderes administrativos, mas com privilégios reduzidos. Como as questões de concurso geralmente abordam comandos (para a configuração ou para a administração), vamos dar uma olhada abaixo.

Após a instalação, é necessário criar um grupo:

```
# groupadd jboss
```

Para adicionar o usuário jboss no grupo jboss:

```
# useradd -s /bin/bash -d /home/jboss -m -g jboss jboss
```

Na sequência, a criação da estrutura de diretórios para armazenar o JBoss, atribuindo o dono e o grupo "jboss" a essa estrutura:

```
# mkdir /EAP_HOME/jboss
```

```
# chown jboss:jboss /EAP_HOME/jboss
```

```
# su jboss
```

Depois é necessário configurar a senha para que o usuário a utilize na conexão entre o Host Controller e o Domain Controller. Dando continuidade, agora vamos criar os perfis para o domain controller (master) e para os host controllers (slave), baseados no modo Domain e renomeá-los:

```
# cp -Rap /EAP_HOME/domain /EAP_HOME/master
```

```
# cp -Rap /EAP_HOME/domain /EAP_HOME/slave01
```

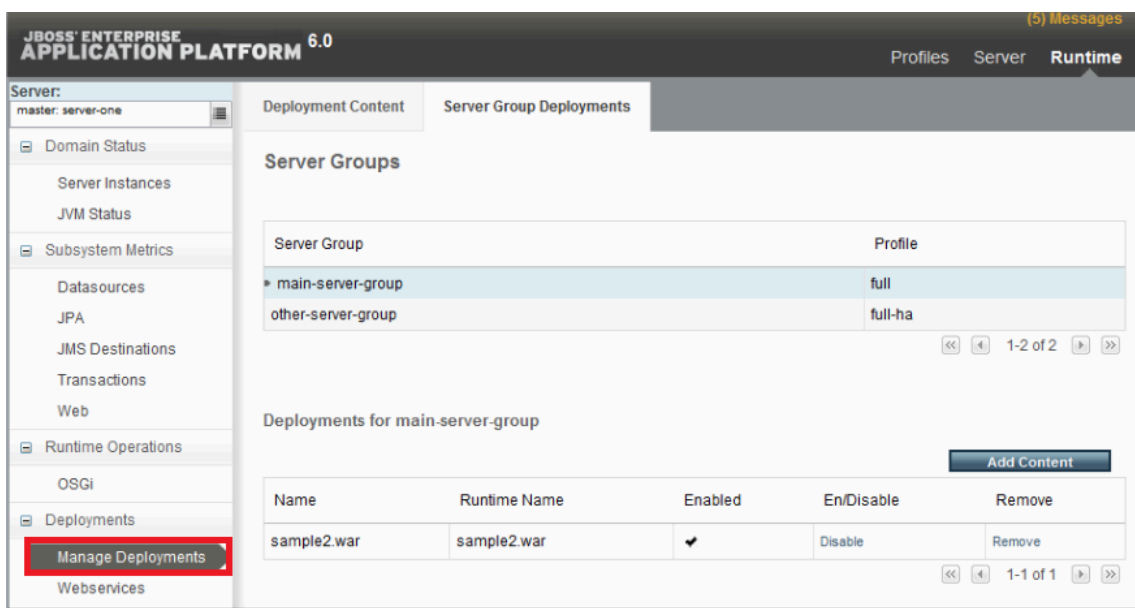
Após a definição do modo (standalone ou domain), é necessário definir um perfil para o JBoss/Wildfly, que é um conjunto de tecnologias ou subsistemas (subsystems) que serão utilizados na aplicação. Essa definição varia conforme os requisitos de cada sistema. O JBoss possui 4 perfis (profiles) por padrão: default, full, ha, full-ha.



Depois de concluída a configuração inicial, as aplicações devem ser “colocadas” no JBoss, processo conhecido como *deploy* (implantação).

O *deploy* varia conforme o tipo de perfil (*profile*) definido para o JBoss/Wildfly. Vamos ver como é feito o *deploy de uma aplicação no JBoss EAP*. Primeiro, devemos acessar o console Web na URL <http://x.x.x.x:9990/console/> (onde x.x.x.x é o endereço IP do servidor). Note que a porta utilizada pelo servidor é a 9990, mas roda a aplicação na 8080. Ou seja, o JBoss/Wildfly roda em uma porta e a aplicação em outra.

Em seguida devem ser inseridos o usuário e a senha, aqueles definidos na instalação. Para realizar um *deploy* e o gerenciamento das instâncias, pode ser utilizado o console:



O *deploy* também pode ser realizado através da linha de comando, o que propicia maior flexibilidade e facilidade na criação de *scripts* de administração. Abaixo um exemplo de *deploy* para todos os servidores:

```
# deploy /path/to/test-application.war --all -server-groups
```

Como já vimos, servidores de aplicação são a interface entre o componente e o sistema operacional específico que o suporta. Antes do componente ser executado no servidor, ele precisa ser montado em uma estrutura que o container possa entender e executar. No JBoss/Wildfly, a *estrutura de empacotamento* é definida na especificação JSR 088 Java EE Application Deployment Specification. Os formatos padrão são WAR, JAR e EAR.

Estrutura de Diretórios

Após a instalação, o JBoss/Wildfly cria uma estrutura de diretórios que deve ser conhecida, pois é necessária para a administração.

- bin/: contém os scripts de inicialização do JBoss EAP, no Linux e no Windows;
- appclient/: contém detalhes de configuração do container da aplicação cliente;



- `modules/`: contém módulos dinamicamente carregados pelo JBoss EAP quando requeridos pelos serviços;
- `standalone/`: arquivos de configuração, conteúdo do deploy, e outras áreas utilizáveis quando o JBoss/Wildfly executa como um servidor standalone;
- `domain/`: contém arquivos de configuração, conteúdo do deploy e outras áreas utilizáveis quando JBoss/Wildfly executa como domínio gerenciado (domain);
- `docs/`: possui diversos tipos de arquivos como exemplos de configuração, exemplos se como executar o Wildfly como serviço, licenças e outros arquivos que ajudam a aprender mais sobre o Wildfly;
- `welcome-content/`: diretório de uso interno do servidor que não deve ser modificado por usuários finais, possui páginas de boas vindas e páginas de erros;
- `jboss-modules.jar`: mecanismo de carga dos módulos.

Além da estrutura de diretórios apresentada, o JBoss/Wildfly cria automaticamente alguns caminhos (*paths*) padrão:

- `jboss.home`: diretório root do JBoss EAP;
- `user.home`: diretório de usuário comum;
- `user.dir`: diretório de trabalho do usuário atual;
- `java.home`: diretório de instalação do Java;
- `jboss.server.base.dir`: diretório root de uma instância de um servidor;
- `jboss.server.data.dir`: diretório que o servidor usa para persistência de dados no storage;
- `jboss.server.log.dir`: diretório que o servidor usa no armazenamento de logs;
- `jboss.server.tmp.dir`: diretório de arquivos temporários;
- `jboss.domain.servers.dir`: diretório no qual um host controller cria a área de trabalho de uma instância em um domínio gerenciado.

Outros conceitos

JGroups: Biblioteca de comunicação em grupo em Java utilizada por diversas aplicações distribuídas para comunicação entre diferentes nós em uma rede. Essa biblioteca oferece suporte à comunicação confiável e escalável em grupos de processos. Ao utilizar o JGroups com o JBoss, as aplicações podem aproveitar esses recursos para criar ambientes distribuídos robustos e escaláveis.

Arquillian: Plataforma de teste para aplicações JEE que simplifica a execução de testes de integração e funcionalidade em containers do JEE. Ferramenta projetada para ajudar os desenvolvedores a realizar testes integrados em ambientes mais próximos da produção, onde componentes como EJBs, CDI (Contexts and Dependency Injection), JPA (Java Persistence API), entre outros, são executados em um ambiente Java EE real.

Operação (Resumo)

De acordo com o que vimos na aula, um resumo da operação do JBoss é o seguinte:

- Configuração do JBoss em ambientes distintos (produção, desenvolvimento e homologação) com arquivos de configuração apropriados (`standalone.xml` ou `domain.xml`);



- Para a implantação de aplicações, pode-se utilizar a console de administração ou scripts CLI para fazer o deploy de aplicações em arquivos .war, .ear, .jar etc.;
- Gerenciamento de modos: Standalone (um único servidor) ou Domain Mode (gerenciamento de múltiplos servidores de forma centralizada);
- A configuração e o monitoramento dos logs (standalone ou domain) são fundamentais para a resolução de problemas e auditoria.

Monitoramento

Para o monitoramento de métricas de CPU, memória, e desempenho de aplicativos existem ferramentas, como Prometheus, Grafana, Nagios, ou até mesmo o JConsole (ferramenta padrão da JDK). Podemos destacar, também:

- MBeans e JMX: O JBoss utiliza Java Management Extensions (JMX) para expor métricas e operações administrativas através dos MBeans. Estes permitem observar a utilização de CPU, memória e status de aplicações;
- Alertas e Notificações: É importante configurar alertas para eventos específicos, como falhas no servidor, excesso de uso de CPU/memória etc. Com esses alertas é possível uma atuação proativa facilitada.

Administração (Resumo)

Conforme vimos nesta aula, vamos destacar alguns pontos importantes para a administração do JBoss:

- Console de Administração: Com a interface Web de administração é possível gerenciar deploys, configurar fontes de dados, ajustar a configuração de segurança, gerenciar clusters, entre outras atividades;
- CLI (*Command Line Interface*): É ideal para operações automatizadas, como scripts de deploy em diversos ambientes ou atualizações em grande escala;
- Gestão de Clusters: No modo Domain, é possível gerenciar diversas instâncias de servidor e balancear a carga entre elas;
- Backup e Recuperação: É importante determinar rotinas de backup dos arquivos de configuração e bases de dados associadas, pois são essenciais para a recuperação de desastres.

Ajustes de Desempenho

Com a intenção de manter o servidor JBoss/WildFly estável, seguro e com alto desempenho para aplicações Java empresariais, alguns processos e ferramentas são:



- Ajuste de JVM: Otimização da configuração da JVM (*Java Virtual Machine*) para alocação de memória, como o heap e o permgen/metaspase;
- Ajuste de Datasources (fontes de dados): Configuração de pools de conexão para limitar o uso de recursos e evitar vazamento de conexões, especialmente para conexões com banco de dados;
- Cache e Sessão: Utilização de cache para melhorar o desempenho em clusters, especialmente em aplicações que dependem de sessões de usuário;
- Perfis de Desempenho (Performance): O JBoss/Wildfly permite a configuração de diferentes perfis de desempenho, como "High Availability" para clusters ou "Full" para ambientes que exigem alta carga de processamento.

Referências Bibliográficas

O que é JBoss. Disponível em <<https://4linux.com.br/o-que-e-jboss/>>.

Red Hat JBoss Enterprise Application Platform. Disponível em <<https://www.redhat.com/pt-br/technologies/jboss-middleware/application-platform>>.

WildFly. A powerful, modular, & lightweight application server that helps you build amazing applications. Disponível em <<https://www.wildfly.org/>>.



QUESTÕES COMENTADAS - SERVIDOR DE APLICAÇÃO JBoss/WILDFLY - MULTIBANCAS

1. (IBFC/EBSERH - 2016) "JBoss é um servidor _____ de código fonte _____ baseado na plataforma _____ e implementado na linguagem de programação _____". Assinale a alternativa que completa correta e respectivamente as lacunas:

- A) de arquivos - aberto - JEE - JavaScript
- B) de aplicação - fechado - JSE - Java
- C) de arquivos - fechado - JEE - JavaScript
- D) de aplicação - aberto - JEE - Java
- E) de impressão - aberto - JSE - JavaScript

Comentários:

Conceitos básicos que vimos logo no primeiro parágrafo:

JBoss é um servidor de aplicação de código fonte aberto baseado na plataforma JEE, é implementado completamente na linguagem de programação Java, podendo ser utilizado em qualquer sistema operacional que suporte essa linguagem. O JBoss Application Server utiliza o arquivo standalone.bat (ou standalone.sh) para prover a sua inicialização. Portanto, a **alternativa D** está correta e é o gabarito da questão.

Gabarito: Letra D

2. (FUNCAB/CREA-AC - 2016) Um administrador de rede instalou o Jboss AS 7 no modo domain. Nesse caso, um dos processos principais, que coordena as instâncias e distribui o arquivo implantado para todas as instâncias do domínio, é denominado:

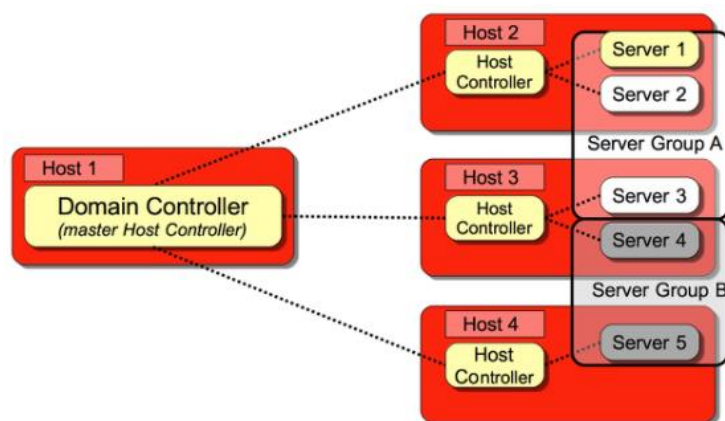
- A) application process.
- B) host controller.
- C) JVM process.
- D) system controller.
- E) standalone controller.



Comentários:

Quando o `domain.sh` (ou o `domain.bat`) é executado em um host, um processo conhecido como Host Controller é lançado, o qual é responsável pelo gerenciamento do servidor. Ele não lida diretamente com a carga de aplicações no servidor, sendo responsável por iniciar e terminar os processos que rodam em cada servidor, interagindo com o Domain Controller para ajudar no gerenciamento.

Por padrão, cada Host Controller lê sua configuração no arquivo `domain/configuration/host.xml`, que possui informações de configuração específicas do host. Abaixo podemos ver uma figura com quatro hosts, um Domain Controller e três Host Controllers, sendo que cada Host Controller pode ter um ou mais servidores.



Portanto, a **alternativa B** está correta e é o gabarito da questão.

Gabarito: Letra B

3. (FCC/TRT11 - 2017) Após instalar o servidor JBoss AS 5 para Windows, deve-se entrar na pasta `$JBOSS_HOME/bin` e digitar `run.bat` para iniciá-lo. Com o servidor iniciado, para acessar a área de login do JBoss AS Administration Console deve-se digitar,

- A) na linha de endereço do navegador, `http://localhost:8080/admin-console`.
- B) em linha de comando, `jboss -a console`.
- C) na linha de endereço do navegador, `http://localhost:8084/settings`.
- D) em linha de comando, `jboss -a mode=console`.
- E) na linha de endereço do navegador, `http://localhost:80/server-console`.

Comentários:



O deploy varia conforme o tipo de perfil (profile) definido para o JBoss/Wildfly. Vamos ver como é feito o deploy de uma aplicação no JBoss EAP. Primeiro, devemos acessar o console Web na URL <http://x.x.x.x:9990/console/> (onde x.x.x.x é o endereço IP do servidor). Note que a porta utilizada pelo servidor é a 9990, mas roda a aplicação na 8080. Ou seja, o JBoss/Wildfly roda em uma porta e a aplicação em outra.

Note que foi cobrada uma versão mais antiga (AS 5) e teve uma leve variada o nome, mas o que vale é focar na porta 8080, que é a chave da questão!

Portanto, a **alternativa A** está correta e é o gabarito da questão.

Gabarito: Letra A

4. (IBFC/EMBASA - 2017) Para que o servidor JBOSS Application Server 7 possa ser plenamente executado, existe a necessidade que esteja previamente instalado e configurado o:

- A) JMF
- B) JCE
- C) JXL
- D) JDK

Comentários:

JBoss é um servidor de aplicação de código fonte aberto baseado na plataforma JEE, é implementado completamente na linguagem de programação Java, podendo ser utilizado em qualquer sistema operacional que suporte essa linguagem.

Para suportar a linguagem Java, deve haver instalado o JDK (Java Development Kit), que é um ambiente utilizado para o desenvolvimento de softwares em Java. O JDK inclui o JRE (Java Runtime Environment), um interpretador/carregador, um compilador (javac), entre outros componentes.

Pode ter apenas o JRE instalado, que também funciona! Mas nas alternativas só encontramos o JDK!

Portanto, a **alternativa D** está correta e é o gabarito da questão.

Gabarito: Letra D

5. (FGV/Câmara de Salvador-BA - 2018) No âmbito do JBoss AS 7.x, os modos de operação disponíveis são denominados:



- A) Domain e Standalone;
- B) Elevated e Standard;
- C) Local e Remote;
- D) Monouser e Multiuser;
- E) Open e Authenticated.

Comentários:

Modo Standalone: é o modo tradicional das versões anteriores. Basicamente implica em ter uma instalação diferente (ou um diretório standalone diferente) para cada instância de Wildfly. Ou seja, para cada Wildfly em execução no seu ambiente é necessário alterar seus próprios arquivo de configuração, suas próprias opções de execução para JVM etc. A inicialização nesse modo é realizada ao iniciar o script `JBOSS_HOME/standalone.sh` (no Linux) ou `JBOSS_HOME/standalone.bat` (no Windows).

Modo Domain: é o modo que foi introduzido no JBoss AS 7, permitindo gerenciar um conjunto de instâncias Wildfly, agrupando-os e assim permitindo compartilhar configurações comuns entre eles. Além de compartilhar configurações, é possível também através de um único console de gerenciamento iniciar ou parar instâncias (ou grupos inteiros), verificar seu status e estatísticas de cada subsystem etc. O gerenciamento das instâncias é coordenado pelo Domain Controller, tendo várias instâncias (JVMs) por Host e o controle total do ciclo de vida dos servidores via Host Controller. Para iniciar no modo domain é executado o script `JBOSS_HOME/domain.sh` no Linux ou `JBOSS_HOME/domain.bat` no Windows.

Portanto, a **alternativa A** está correta e é o gabarito da questão.

Gabarito: Letra A

6. (FCC/MPE-PE - 2018) Em uma instalação padrão do JBoss Application Server (AS) 7, o diretório que contém a página de boas-vindas do AS é o

- A) standalone.
- B) welcome-content.
- C) appclient.
- D) server-welcome.
- E) bundles.



Comentários:

Após a instalação, o JBoss/Wildfly cria uma estrutura de diretórios que deve ser conhecida, pois é necessária para a administração.

- bin/: contém os scripts de inicialização do JBoss EAP, no Linux e no Windows;
- appclient/: contém detalhes de configuração do container da aplicação cliente;
- modules/: contém módulos dinamicamente carregados pelo JBoss EAP quando requeridos pelos serviços;
- standalone/: arquivos de configuração, conteúdo do deploy, e outras áreas utilizáveis quando o JBoss/Wildfly executa como um servidor standalone;
- domain/: contém arquivos de configuração, conteúdo do deploy e outras áreas utilizáveis quando JBoss/Wildfly executa como domínio gerenciado (domain);
- docs/: possui diversos tipos de arquivos como exemplos de configuração, exemplos se como executar o Wildfly como serviço, licenças e outros arquivos que ajudam a aprender mais sobre o Wildfly;
- welcome-content/: diretório de uso interno do servidor que não deve ser modificado por usuários finais, possui páginas de boas vindas e páginas de erros;
- jboss-modules.jar: mecanismo de carga dos módulos.

Portanto, a **alternativa B** está correta e é o gabarito da questão.

Gabarito: Letra B

7. (FCC/CLDF - 2018) Considere que o servidor de aplicações JBoss AS 7 está instalado e configurado em modo padrão em um computador com sistema operacional Windows 10. Para testar se o servidor JBoss está funcionando, utilizando um navegador, deve-se digitar o URL

- A) server://127.0.0.1:0800
- B) ftp://localhost:0800
- C) http://localhost:8080
- D) server://localhost:80
- E) http://127.0.0.1:80

Comentários:

O deploy varia conforme o tipo de perfil (profile) definido para o JBoss/Wildfly. Vamos ver como é feito o deploy de uma aplicação no JBoss EAP. Primeiro, devemos acessar o console Web na URL <http://x.x.x.x:9990/console/> (onde x.x.x.x é o endereço IP do servidor). Note que a porta



utilizada pelo servidor é a 9990, mas roda a aplicação na 8080. Ou seja, o JBoss/Wildfly roda em uma porta e a aplicação em outra. Portanto, a **alternativa C** está correta e é o gabarito da questão.

Gabarito: Letra C



LISTA DE QUESTÕES - SERVIDOR DE APLICAÇÃO JBoss/WILDFLY - MULTIBANCAS

1. (IBFC/EBSERH - 2016) "JBoss é um servidor _____ de código fonte _____ baseado na plataforma _____ e implementado na linguagem de programação _____". Assinale a alternativa que completa correta e respectivamente as lacunas:

- A) de arquivos - aberto - JEE - JavaScript
- B) de aplicação - fechado - JSE - Java
- C) de arquivos - fechado - JEE - JavaScript
- D) de aplicação - aberto - JEE - Java
- E) de impressão - aberto - JSE - JavaScript

2. (FUNCAB/CREA-AC - 2016) Um administrador de rede instalou o Jboss AS 7 no modo domain. Nesse caso, um dos processos principais, que coordena as instâncias e distribui o arquivo implantado para todas as instâncias do domínio, é denominado:

- A) application process.
- B) host controller.
- C) JVM process.
- D) system controller.
- E) standalone controller.

3. (FCC/TRT11 - 2017) Após instalar o servidor JBoss AS 5 para Windows, deve-se entrar na pasta \$JBASS_HOME/bin e digitar run.bat para iniciá-lo. Com o servidor iniciado, para acessar a área de login do JBoss AS Administration Console deve-se digitar,

- A) na linha de endereço do navegador, `http://localhost:8080/admin-console`.
- B) em linha de comando, `jboss -a console`.
- C) na linha de endereço do navegador, `http://localhost:8084/settings`.
- D) em linha de comando, `jboss -a mode=console`.



E) na linha de endereço do navegador, `http://localhost:80/server-console`.

4. (IBFC/EMBASA - 2017) Para que o servidor JBOSS Application Server 7 possa ser plenamente executado, existe a necessidade que esteja previamente instalado e configurado o:

- A) JMF
- B) JCE
- C) JXL
- D) JDK

5. (FGV/Câmara de Salvador-BA - 2018) No âmbito do JBoss AS 7.x, os modos de operação disponíveis são denominados:

- A) Domain e Standalone;
- B) Elevated e Standard;
- C) Local e Remote;
- D) Monouser e Multiuser;
- E) Open e Authenticated.

6. (FCC/MPE-PE - 2018) Em uma instalação padrão do JBoss Application Server (AS) 7, o diretório que contém a página de boas-vindas do AS é o

- A) standalone.
- B) welcome-content.
- C) appclient.
- D) server-welcome.
- E) bundles.

7. (FCC/CLDF - 2018) Considere que o servidor de aplicações JBoss AS 7 está instalado e configurado em modo padrão em um computador com sistema operacional Windows 10. Para testar se o servidor JBoss está funcionando, utilizando um navegador, deve-se digitar o URL

- A) `server://127.0.0.1:0800`



- B) ftp://localhost:0800
- C) http://localhost:8080
- D) server://localhost:80
- E) http://127.0.0.1:80

GABARITO



GABARITO

- 1- D
- 2- B
- 3- A

- 4- D
- 5- A
- 6- B

- 7- C



ESSA LEI TODO MUNDO CONHECE: PIRATARIA É CRIME.

Mas é sempre bom revisar o porquê e como você pode ser prejudicado com essa prática.



1 Professor investe seu tempo para elaborar os cursos e o site os coloca à venda.



2 Pirata divulga ilicitamente (grupos de rateio), utilizando-se do anonimato, nomes falsos ou laranjas (geralmente o pirata se anuncia como formador de "grupos solidários" de rateio que não visam lucro).



3 Pirata cria alunos fake praticando falsidade ideológica, comprando cursos do site em nome de pessoas aleatórias (usando nome, CPF, endereço e telefone de terceiros sem autorização).



4 Pirata compra, muitas vezes, clonando cartões de crédito (por vezes o sistema anti-fraude não consegue identificar o golpe a tempo).



5 Pirata fere os Termos de Uso, adultera as aulas e retira a identificação dos arquivos PDF (justamente porque a atividade é ilegal e ele não quer que seus fakes sejam identificados).



6 Pirata revende as aulas protegidas por direitos autorais, praticando concorrência desleal e em flagrante desrespeito à Lei de Direitos Autorais (Lei 9.610/98).



7 Concurseiro(a) desinformado participa de rateio, achando que nada disso está acontecendo e esperando se tornar servidor público para exigir o cumprimento das leis.



8 O professor que elaborou o curso não ganha nada, o site não recebe nada, e a pessoa que praticou todos os ilícitos anteriores (pirata) fica com o lucro.



Deixando de lado esse mar de sujeira, aproveitamos para agradecer a todos que adquirem os cursos honestamente e permitem que o site continue existindo.