

**Aula 00 - Prof. Evandro**  
*SEPLAN-PI (Analista Governamental -  
Tecnologia da Informação) Arquitetura de  
Computadores*

Autor:  
**Evandro Dalla Vecchia Pereira**

08 de Abril de 2024

# Índice

1) Servidores de Aplicação JBoss Wildfly - Teoria .....	4
2) Servidores de Aplicação JBoss Wildfly - Questões Comentadas - Multibancas .....	12
3) Servidores de Aplicação JBoss Wildfly - Lista de Questões - Multibancas .....	18
4) Servidores Web - Apache - Teoria .....	21
5) Servidores Web - Apache - Questões Comentadas - Cebraspe .....	34
6) Servidores Web - Apache - Questões Comentadas - FGV .....	43
7) Servidores Web - Apache - Questões Comentadas - FCC .....	48
8) Servidores Web - Apache - Questões Comentadas - VUNESP .....	51
9) Servidores Web - Apache - Questões Comentadas - Multibancas .....	53
10) Servidores Web - Apache - Lista de Questões - Cebraspe .....	59
11) Servidores Web - Apache - Lista de Questões - FGV .....	64
12) Servidores Web - Apache - Lista de Questões - FCC .....	67
13) Servidores Web - Apache - Lista de Questões - VUNESP .....	69
14) Servidores Web - Apache - Lista de Questões - Multibancas .....	71
15) Servidores Web - IIS - Teoria .....	75
16) Servidores Web - IIS - Questões Comentadas - Cebraspe .....	84
17) Servidores Web - IIS - Questões Comentadas - FGV .....	86
18) Servidores Web - IIS - Questões Comentadas - FCC .....	88
19) Servidores Web - IIS - Questões Comentadas - Vunesp .....	89
20) Servidores Web - IIS - Questões Comentadas - Multibancas .....	90
21) Servidores Web - IIS - Lista de Questões - Cebraspe .....	97
22) Servidores Web - IIS - Lista de Questões - FGV .....	99
23) Servidores Web - IIS - Lista de Questões - FCC .....	101
24) Servidores Web - IIS - Lista de Questões - Vunesp .....	102
25) Servidores Web - IIS - Lista de Questões - Multibancas .....	103
26) Servidores Web - Nginx - Teoria .....	108
27) Servidores Web - Nginx - Questões Comentadas - Cebraspe .....	123
28) Servidores Web - Nginx - Questões Comentadas - FGV .....	128



## Índice

29) Servidores Web - Nginx - Questões Comentadas - Vunesp .....	129
30) Servidores Web - Nginx - Questões Comentadas - Multibancas .....	130
31) Servidores Web - Nginx - Lista de Questões - Cebraspe .....	134
32) Servidores Web - Nginx - Lista de Questões - FGV .....	137
33) Servidores Web - Nginx - Lista de Questões - Vunesp .....	138
34) Servidores Web - Nginx - Lista de Questões - Multibancas .....	139



## 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 middleware<sup>1</sup> 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.

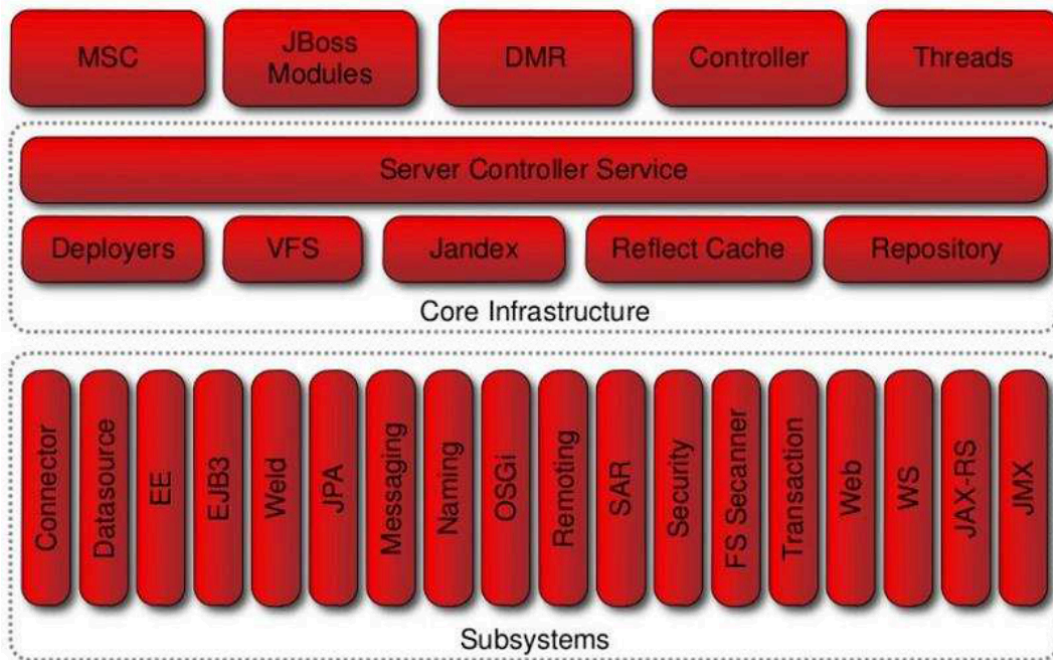
### Arquitetura e Modos

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.

---

<sup>1</sup> 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.

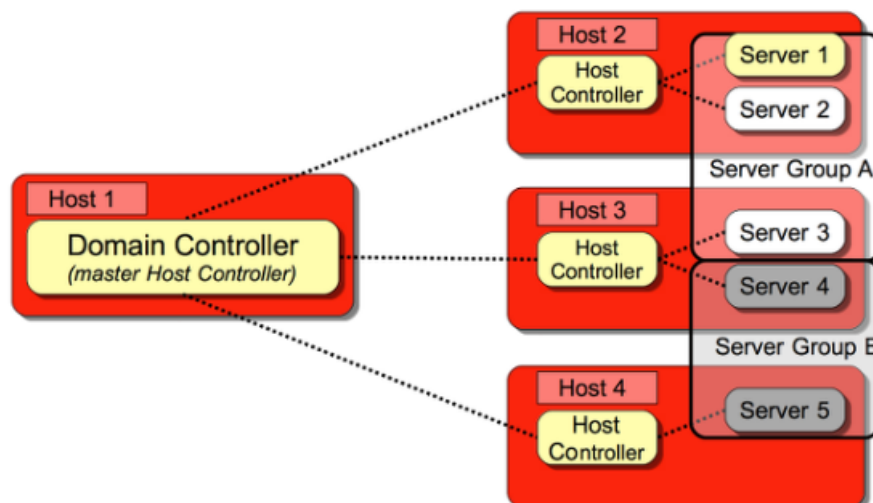
**Modo Standalone:** é o modo tradicional das versões anteriores. Basicamente implica em ter uma instalação diferente (ou um diretório standalone diferente) para cada instância de Wildfly. Ou seja, para cada Wildfly em execução no seu ambiente é necessário alterar seus próprios arquivos de configuração, 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
```

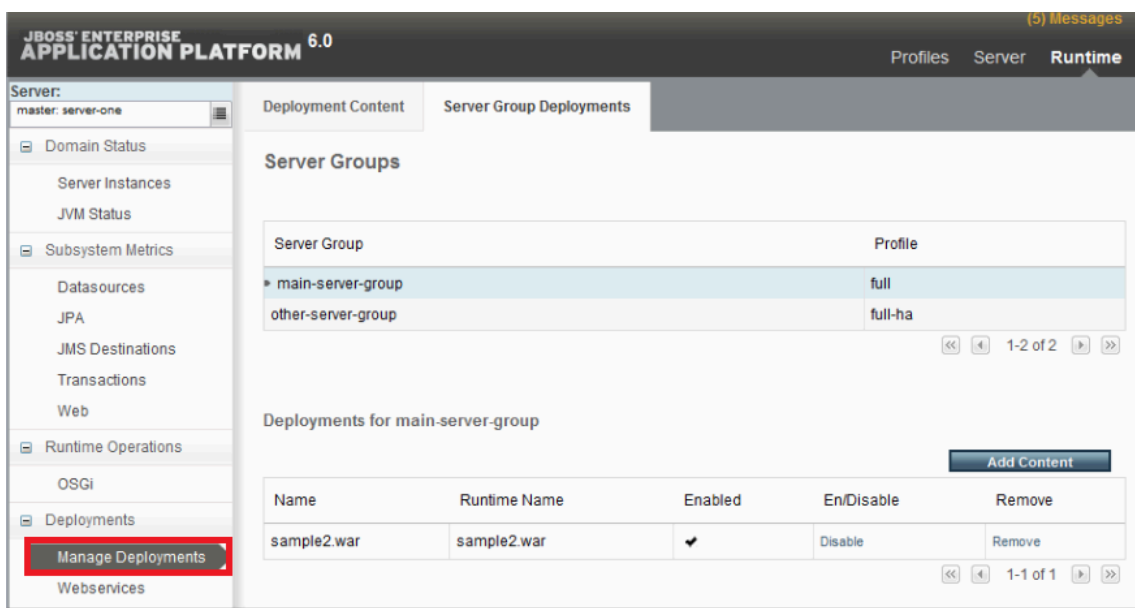
Após a definição do modo (standalone ou domain), é necessário definir um perfil para o JBoss/Wildfly, que é um conjunto de tecnologias ou subsistemas (subsystems) que serão utilizados na aplicação. Essa definição varia conforme os requisitos de cada sistema. O JBoss possui 4 perfis (profiles) por padrão: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

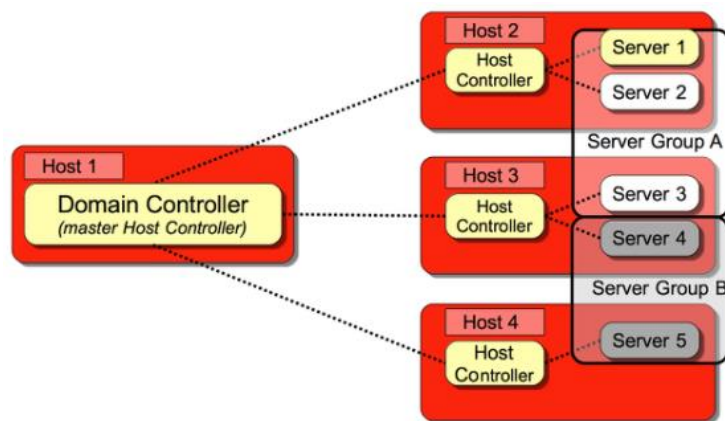
- A) application process.
- B) host controller.
- C) JVM process.
- D) system controller.
- E) standalone controller.



## 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





## SERVIDOR WEB APACHE

O Apache é um servidor Web gratuito e livre (possui código-fonte aberto) que funciona em ambiente multiplataforma (Windows, Novell, OS/2, Unix, Linux, FreeBSD etc.). No Linux o servidor utiliza o *daemon* httpd. Suas funcionalidades são mantidas através de uma estrutura de módulos, permitindo inclusive que o usuário escreva seus próprios módulos.

Existem dois tipos de páginas que podem ser adicionadas ao Apache: a página raiz e as subpáginas. A página raiz é especificada através da diretiva **DocumentRoot** e será mostrada quando o usuário entrar no domínio principal, como por exemplo <http://www.estrategiaconcursos.com.br>. Na configuração padrão do Apache, DocumentRoot aponta para o diretório /var/www (ou /var/www/html, dependendo da distribuição). Esse diretório será assumido como raiz caso os diretórios não sejam iniciados por uma "/". Vejamos:

- `home/estrategia` → Aponta para /var/www/home/estrategia
- `/home/estrategia` → Aponta para /home/estrategia

Em ambientes Unix-like os **arquivos de configuração**, por padrão, ficam localizados no diretório /etc/httpd/conf ou /etc/apache2. O servidor é configurado por um arquivo denominado **httpd.conf** (principal arquivo de configuração do Apache) e opcionalmente pode haver configurações para cada diretório utilizando arquivos com o nome **.htaccess**, onde é possível utilizar autenticação de usuário pelo próprio protocolo HTTP (combinação do arquivo .htaccess com o arquivo .htpasswd, que guardará os usuários e senhas criptografadas). No Linux, o arquivo de configuração principal geralmente é encontrado em /etc/httpd/conf/httpd.conf ou /etc/apache2/apache2.conf.

Os **arquivos .htaccess** ("arquivos de configuração distribuída") oferecem uma maneira de realizar mudanças nas configurações por diretório. Um arquivo, contendo uma ou mais diretrizes de configuração, é colocado em determinado diretório, e as diretrizes se aplicam apenas para aquele diretório e todos os seus subdiretórios. Uma observação importante é que é possível renomear o arquivo ".htaccess" para outro nome qualquer, sendo que para isso é necessário utilizar a diretriz "AccessFileName". Por exemplo, se quiser que o arquivo se chame ".configuracao", então pode-se adicionar a seguinte linha ao arquivo de configuração do servidor:

```
AccessFileName .configuracao
```

### Execução do Apache

**Apachectl** (*Apache HTTP Server Control Interface*): trata-se de um *front end* (interface para o usuário) para o servidor Apache. É utilizado para ajudar o administrador a controlar o funcionamento do *daemon* httpd. O *script* apachectl pode operar em dois modos:

- *Front end* simples que configura quaisquer variáveis de ambiente necessárias e então chama o httpd, passando argumentos de linha de comando;



- *Script* de inicialização, recebendo argumentos simples como *start*, *restart*, e *stop*, traduzindo-os em sinais apropriados ao *httpd*.

No caso do Linux, para inicializar o *httpd*, um comando possível é `apachectl start`, mas há outras formas também, dependendo da distribuição utilizada:

- Red Hat: `service httpd start`;
- Ubuntu: `/etc/init.d/apache2 start`.

O argumento "start" pode ser trocado por "restart", "stop" etc., para as demais ações.

Em geral, o *httpd* não deve ser chamado diretamente, mas deve ser chamado via *apachectl* nos sistemas Unix-like; como um serviço no Windows NT, 2000 e XP; e como uma aplicação no Windows 9x e ME. As opções (parâmetros) são:

- `-d serverroot`: configura o valor inicial para a diretiva *ServerRoot* como "serverroot". Isso pode ser substituído pela diretiva *ServerRoot* no arquivo de configuração. O padrão é `/etc/httpd`;
- `-f config`: usa as diretivas no arquivo de configuração na inicialização. Se "config" não começar com /, então será considerado um caminho relativo ao *ServerRoot*. O padrão é `conf/httpd.conf`;
- `-k start|restart|graceful|stop|graceful-stop`: sinaliza o *httpd* para iniciar, reiniciar ou parar. Quando utilizado no **modo de script de inicialização**, o *apachectl* aceita como **argumentos** (pode-se utilizar "-k" antes do argumento):
  - ✓ `start`: inicializa o *daemon* *httpd*;
  - ✓ `stop`: finaliza o *daemon* *httpd*;
  - ✓ `restart`: reinicializa o *daemon* *httpd*. Se não estiver rodando, o *httpd* é inicializado;
  - ✓ `fullstatus`: mostra um relatório completo do status do *mod\_status*. Para isso, o módulo *mod\_status* deve estar habilitado;
  - ✓ `status`: mostra um relatório resumido do status. Similar à opção `fullstatus`, mas a lista de requisições correntes que estão sendo servidas é omitida;
  - ✓ `graceful`: reinicializa o *daemon* de forma "gentil". Se não estiver rodando, o *httpd* é inicializado. A diferença para uma inicialização normal é que as conexões correntes não são abortadas;
  - ✓ `graceful-stop`: finaliza o *daemon* de forma "gentil". A diferença para uma finalização normal é que as conexões correntes não são abortadas;
- `-configtest`: executa um teste no arquivo de configuração. Após verificar a sintaxe, informa se está ok ou aponta os erros encontrados;
- `-C directive`: processa a diretiva de configuração "directive" antes de ler os arquivos de configuração;
- `-c directive`: processa a diretiva de configuração "directive" após ler os arquivos de configuração;



- -D parameter: define um parâmetro de configuração que pode ser usado com seções <IfDefine> nos arquivos de configuração para ignorar ou processar comandos condicionalmente na inicialização e reinicialização do servidor;
- -e level: define o nível de log para "level" durante a inicialização do servidor, o que é útil para aumentar temporariamente o detalhamento das mensagens de erro para encontrar problemas durante a inicialização;
- -E file: envia mensagens de erro durante a inicialização do servidor para o arquivo "file";
- -R directory: quando o servidor é compilado usando a regra SHARED\_CORE, isso especifica o diretório "directory" para os arquivos de objetos compartilhados;
- -h: mostra um breve resumo das opções de linha de comando disponíveis (trata-se de uma ajuda - help);
- -l: mostra uma lista de módulos compilados no servidor. Isso não listará os módulos carregados dinamicamente incluídos através da diretiva LoadModule;
- -L: mostra uma lista de diretivas junto com os argumentos esperados e os locais onde a diretiva é válida;
- -M: despeja uma lista de módulos estáticos e compartilhados que foram carregados;
- -S: mostra as configurações conforme análises realizadas no arquivo de configuração;
- -t: executa testes de sintaxe apenas para arquivos de configuração. O programa sai imediatamente após esses testes de análise de sintaxe com um código de retorno 0 (sintaxe OK) ou um código de retorno diferente de 0 (erro de sintaxe);
- -v: imprime a versão do httpd;
- -V: imprime a versão e os parâmetros de compilação do httpd;
- -X: executa o httpd no modo de depuração. Apenas um "worker" será iniciado e o servidor não será desconectado do console;
- -n name: "name" será o nome do serviço Apache a ser sinalizado;
- -w: mantém a janela do console aberta em caso de erro para que a mensagem de erro possa ser lida.

## Arquivos de configuração

Abaixo vemos os arquivos de configuração, que por padrão ficam em "/etc/apache" (mesmo já tendo falado do httpd.conf, vale a pena dar mais ênfase agora).



- **httpd.conf**: arquivo de configuração principal, possuindo diretivas que controlam a operação do *daemon* servidor. Um arquivo de configuração alternativo pode ser especificado através da opção "-f" da linha de comando;
- **srm.conf**: possui diretivas que controlam a especificação de documentos que o servidor oferece aos clientes. O nome desse arquivo pode ser substituído através da diretiva **ResourceConfig** no arquivo principal de configuração;
- **access.conf**: possui diretivas que controlam o acesso aos documentos. O nome desse arquivo pode ser substituído através da diretiva **AccessConfig** no arquivo principal de configuração.

## Módulos

Como já mencionado, as funcionalidades do Apache são mantidas através de uma estrutura de módulos, permitindo inclusive que o usuário escreva seus próprios módulos. Os **módulos** são implementados de duas maneiras: **estáticos** ou **dinâmicos**. Os estáticos são inseridos durante a compilação e são incorporados ao código binário do servidor. Os dinâmicos são ligados (link) ao código binário do servidor em tempo de execução. Existem diversos módulos, mas as bancas costumam cobrar poucos, os quais serão mostrados na sequência.

Primeiro vamos focar nos módulos com os principais recursos e os de multiprocessamento:

- **core**: "núcleo", principais recursos do servidor Apache que estão sempre disponíveis;
- **mpm\_common**: uma coleção de diretivas implementadas por mais de um módulo de multiprocessamento (MPM - *multi-processing module*);
- **worker**: módulo de multiprocessamento (MPM) implementando um servidor Web híbrido *multi-threaded* e multi-processo;
- **event**: Uma variante do MPM worker com o objetivo de consumir threads apenas para conexões com processamento ativo;
- **prefork**: implementa um servidor Web sem *threads*;
- **mpm\_netware**: módulo de multiprocessamento (MPM) implementando um servidor Web *multi-thread* otimizado para o sistema operacional Novell NetWare;
- **mpm\_os2**: MPM híbrido multi-processo e *multi-thread* para o sistema operacional OS/2;
- **mpm\_winnt**: MPM otimizado para o sistema operacional Windows NT.

Agora vamos ver alguns dos (outros) módulos (obs.: todos possuem o nome começado por "mod\_"):

- **mod\_ssl**: focado na segurança, adiciona a capacidade do servidor atender requisições utilizando o protocolo HTTPS. O HTTPS utiliza uma camada SSL para criptografar todos os



dados transferidos entre o cliente e o servidor, provendo maior grau de segurança, confidencialidade e confiabilidade dos dados. A camada SSL é compatível com certificados X.509, que são os certificados digitais fornecidos e assinados por grandes entidades certificadoras no mundo;

- **mod\_headers**: fornece diretivas para controlar e modificar cabeçalhos de solicitação e resposta HTTP. Os cabeçalhos podem ser mesclados, substituídos ou removidos;
- **mod\_rewrite**: fornece um mecanismo de reescrita baseado em regras para reescrever os URLs solicitados em tempo real. Por exemplo, pode reescrever um endereço "http" para "https";
- **mod\_sed**: filtra o conteúdo de entrada (solicitação) e saída (resposta) usando a sintaxe sed (sed é um editor de fluxo - usado para executar transformações em texto básico em um fluxo de entrada, seja em um arquivo ou entrada de um pipeline);
- **mod\_ldap**: tem a intenção de melhorar o desempenho de sites que dependem de conexões de back-end (parte de um sistema que lida com a lógica de negócios, processamento de dados e interações com o banco de dados) para servidores LDAP. Além das funções fornecidas pelas bibliotecas LDAP padrão, este módulo adiciona um pool de conexões LDAP e um cache de memória compartilhada LDAP;
- **mod\_mime**: associa as extensões do nome de arquivo solicitado ao comportamento do arquivo (manipuladores e filtros) e conteúdo (tipo MIME, idioma, conjunto de caracteres e codificação);
- **mod\_session**: fornece suporte para um servidor amplo por interface de sessão de usuário. As sessões podem ser usadas para controlar se um usuário está conectado ou para outras informações por usuário que devem ser mantidas disponíveis nas solicitações (utiliza cookies).

Para habilitar/desabilitar módulos, pode-se utilizar:

- **a2enmod**: comando para habilitar um módulo (abaixo um exemplo para o módulo rewrite)  

```
sudo a2enmod rewrite
```
- **a2dismod**: comando usado para desabilitar um módulo (abaixo um exemplo para o módulo rewrite)  

```
sudo a2dismod rewrite
```

## Seções e Diretivas

Diretrizes (ou diretivas) nos arquivos de configuração (httpd.conf e .htaccess) podem ser aplicadas ao servidor inteiro ou podem ser restritas na aplicação de determinados diretórios, arquivos, hosts ou URLs. Seções são grupos de diretrizes, com a seguinte sintaxe:



<Seção>

# Diretrizes... Obs.: cerquilha indica o início de comentário

</Seção>

Vamos ver o significado de algumas delas (não seria muito produtivo ver todas), além das seções, como por exemplo <Directory>, entre outras:

<Directory> é utilizada para definir um grupo de diretrizes que devem ser aplicadas apenas ao diretório definido, seus subdiretórios, e aos arquivos dentro deles. Exemplo:

```
<Directory "/usr/local/httpd/htdocs">
```

```
Options Indexes FollowSymLinks
```

```
</Directory>
```

<Files> limita o escopo das diretrizes pelos nomes de arquivos. Funciona igual ao <Directory>, mas agora o foco são arquivos. Exemplo:

```
<Files "?at.*">
```

```
# Aplica-se a cat.html, bat.html, hat.php etc., pois "?" significa um caractere qualquer
```

```
# e * significa qualquer coisa daquela posição em diante
```

```
# Colocar as diretrizes aqui...
```

```
</Files>
```

<Location> limita o escopo das diretivas incluídas por URL. É semelhante à diretiva <Directory> e inicia uma subseção que é encerrada com uma diretiva </Location>. As seções <Location> são processadas na ordem em que aparecem no arquivo de configuração, depois que as seções <Directory> e os arquivos .htaccess são lidos e depois das seções <Files>.

<VirtualHost> permite servir mais de um *site* no mesmo servidor (*sites* virtuais). Podem ser utilizadas diretivas específicas para o controle do *site* virtual, como nome do administrador, erros de acesso à página, controle de acesso e outros dados úteis para personalizar e gerenciar o *site*.

**Virtual Hosts baseados em nomes:** utilizam nomes para identificar os *sites* servidos e requerem somente um endereço IP. Assim é possível servir um número ilimitado de *sites* virtuais. O navegador do cliente deve suportar os cabeçalhos necessários para garantir o funcionamento desse recurso (os navegadores mais comuns possuem tal suporte). Exemplos:

```
<VirtualHost www.site1.com.br>
```

```
ServerName www.site1.com.br
```





```
ServerAdmin site1@site1.com.br  
DocumentRoot /var/www/www_site1_com_br  
TransferLog /var/log/apache/site1/access.log  
ErrorLog /var/log/apache/site1/error.log  
User www-data  
Group www-data
```

</VirtualHost>

<VirtualHost www.site2.com.br>

```
ServerName www.site2.com.br  
DocumentRoot /var/www/www_site2_com_br  
CustomLog /var/log/apache/site2/access.log combined  
ErrorLog /var/log/apache/site2/error.log
```

</VirtualHost>

*Virtual Hosts* baseados em endereços IP: permitem hospedar múltiplos sites em um único servidor, cada um associado a um endereço IP específico. Abaixo temos um exemplo de host virtual em qualquer endereço IP (\*), que escuta na porta 80.

<VirtualHost \*:80>

```
ServerName www.exemplo1.com  
DocumentRoot /caminho/para/o/site1
```

</VirtualHost>

Outro exemplo, com endereço IP específico seria:

<VirtualHost 192.168.1.150:80>

```
ServerName www.exemplo2.com  
DocumentRoot /caminho/para/o/site2
```

</VirtualHost>



**ServerRoot** define o diretório no qual o servidor reside. Normalmente, ele conterá os subdiretórios `conf/` e `logs/`. Caminhos relativos em outras diretivas de configuração (como `Include` ou `LoadModule`, por exemplo) são tomados como relativos a este diretório.

**AcceptFilter** permite otimizações específicas do sistema operacional para um socket pelo tipo de protocolo. A premissa básica é que o kernel não envie um socket para o processo do servidor até que os dados sejam recebidos ou uma solicitação HTTP inteira seja armazenada em buffer.

**Redirect** mapeia uma URL antiga para uma nova, solicitando ao cliente que busque novamente o recurso no novo local. Ex.:

```
Redirect "/servico" "http://abc.exemplo.com.br/servico"
```

**SetInputFilter** define o filtro (ou filtros) que processará as solicitações do cliente e a entrada POST quando forem recebidas pelo servidor. Isso é um acréscimo a quaisquer filtros definidos em outro lugar, incluindo a diretiva `AddInputFilter`. Se for especificado mais de um filtro, eles devem ser separados por ponto e vírgula na ordem em que devem processar o conteúdo.

A diretiva **Listen** instrui o `httpd` a escutar endereços IP específicos ou portas específicas. Por padrão, o servidor responde a requisições em todas as interfaces IP. Essa diretiva é obrigatória (a partir da versão 2.4), ou seja, se não estiver presente no arquivo de configuração, o servidor falhará ao iniciar. Diretivas múltiplas podem ser utilizadas para especificar diferentes endereços ou portas. Exemplo:

```
Listen 80
```

```
Listen 8000
```

Para especificar duas interfaces e suas portas, utiliza-se, por exemplo:

```
Listen 192.188.1.1:80
```

```
Listen 192.188.1.2:8000
```

Endereços IPv6 devem ser colocados entre colchetes:

```
Listen [2001:db8::a00:20ff:fea7:ccea]:80
```

Para o protocolo HTTPS (HTTP seguro), a porta padrão é a 443, ou seja, se nada for especificado, o servidor assume essa porta. Para configurar outra porta, utiliza-se:

```
Listen 192.188.1.2:9443 https
```

❗ Obs.: Se não especificar o protocolo e colocar uma porta qualquer (não padrão HTTP = 80, e não padrão HTTPS = 443), será assumido o protocolo HTTP.

❗ Importante: Várias diretivas `Listen` com o mesmo endereço IP e porta resultarão em um erro fatal, impossibilitando que o servidor Web Apache inicie corretamente (segundo uma banca de concursos).



A diretiva **IndexIgnore** adiciona à lista de arquivos a serem escondidos quando listar um diretório. Pode utilizar caracteres curingas ("?" e "\*"). Múltiplas diretivas **IndexIgnore** acrescentam na lista, ao invés de substituir. Por padrão, a lista contém "." (o diretório corrente). Exemplo:

```
<Directory "/var/www">  
  
    IndexIgnore *.bak .??* *~ *# HEADER* README* RCS CVS *,v *,t  
  
</Directory>
```

A diretiva **KeepAlive** habilita/desabilita conexões HTTP persistentes.

**MaxKeepAliveRequests** define o número máximo de requisições permitidas (0 = sem limite).

**KeepAliveTimeout** define o tempo (em segundos) a esperar para a próxima requisição do mesmo cliente (na mesma conexão). Exemplo:

```
KeepAlive On  
  
MaxKeepAliveRequests 50  
  
KeepAliveTimeout 10
```

**StartServers** define o número de processos filho (child) do servidor a serem criados na inicialização. Como o número de processos é controlado dinamicamente dependendo da carga, geralmente há poucas razões para ajustar esse parâmetro. O valor padrão difere de MPM para MPM. Para "worker" e "event", o valor padrão de **StartServers** é 3; **prefork** é padronizado para 5; e o padrão de **mpmt\_os2** é 2.

**ServerTokens** define se o campo de cabeçalho de resposta do servidor (que é enviado aos clientes) inclui uma descrição do tipo de sistema operacional genérico do servidor, bem como informações sobre módulos compilados. Abaixo exemplos de resposta de acordo com o parâmetro utilizado.

- **ServerTokens Full** (ou não especificado) - o servidor envia, por exemplo, "Server: Apache/2.4.2 (Unix) PHP/4.2.2 MyMod/1.2"
- **ServerTokens Prod[uctOnly]** - o servidor envia, por exemplo, "Server: Apache"
- **ServerTokens Major** - o servidor envia, por exemplo, "Server: Apache/2"
- **ServerTokens Minor** - o servidor envia, por exemplo, "Server: Apache/2.4"
- **ServerTokens Min[imal]** - o servidor envia, por exemplo, "Server: Apache/2.4.2"
- **ServerTokens OS** - o servidor envia, por exemplo, "Server: Apache/2.4.2 (Unix)"



**SSLEngine** alterna o uso do mecanismo de protocolo SSL/TLS. Essa diretiva deve ser usada dentro de uma seção <VirtualHost> para habilitar o SSL/TLS para esse host virtual. Por padrão, o mecanismo de protocolo SSL/TLS está desabilitado para o servidor principal e para todos os hosts virtuais configurados. Ex.:

```
<VirtualHost _default_:443>
```

```
    SSLEngine on
```

```
    #...
```

```
</VirtualHost>
```

**DirectoryIndex** define a lista de recursos a serem procurados, quando o cliente solicita um índice do diretório especificando uma barra ("/") no final do nome do diretório. A URL local é a (URL codificada em %) de um documento no servidor relativo ao diretório solicitado; geralmente é o nome de um arquivo no diretório. Vários URLs podem ser fornecidos, caso em que o servidor retornará o primeiro que encontrar. Se nenhum dos recursos existir e a opção "Indexes" estiver definida, o servidor gerará sua própria listagem do diretório. Ex.: DirectoryIndex index.html.

**ServerAlias** define os nomes alternativos para um host, para uso com hosts virtuais baseados em nomes. O ServerAlias pode incluir curingas, se apropriado. Ex.:

```
<VirtualHost *:80>
```

- ServerName servidor.exemplo.com.br  
ServerAlias servidor servidor2.exemplo.com.br servidor2  
ServerAlias \*.exemplo.com.br  
UseCanonicalName Off  
# ...

```
</VirtualHost>
```

**Alias** é usada para associar um caminho a um diretório físico no sistema de arquivos do servidor, o que permite criar atalhos ou aliases para simplificar o acesso a determinados recursos em um site. O uso comum do Alias é criar URLs mais amigáveis ou para apontar para diretórios fora do diretório de documentos principal. Sintaxe:

```
Alias /caminho-virtual /caminho-fisico-no-servidor
```

- /caminho-virtual: O caminho acessado na URL;
- /caminho-fisico-no-servidor: O caminho real no sistema de arquivos do servidor.

**UserDir** define o diretório real no diretório inicial de um usuário (home) para usar quando uma solicitação de um documento para um usuário é recebida. Exemplos:

UserDir	public_html	~bob/public_html/one/two.html
UserDir	/usr/web	/usr/web/bob/one/two.html
UserDir	/home/*/www	/home/bob/www/one/two.html



**ServerName** define o esquema de solicitação, nome do host e porta que o servidor usa para se identificar. **ServerName** é usado (possivelmente em conjunto com **ServerAlias**) para identificar exclusivamente um host virtual, ao usar hosts virtuais baseados em nomes.

**MaxRequestWorkers**: número máximo de conexões que serão processadas simultaneamente.

**HeartbeatMaxServers**: especifica o número máximo de servidores que podem enviar requisições "Heartbeat" para esse servidor. "Heartbeats" são usados para monitorar a "saúde dos nodos".

**ServerSignature** permite a configuração de uma linha de rodapé à direita em documentos gerados pelo servidor (mensagens de erro, listagens de diretório ftp etc.). Uma razão para habilitar essa diretiva (valor "On") é que em uma cadeia de proxies, o usuário geralmente não tem possibilidade de dizer qual dos servidores encadeados realmente produziu uma mensagem de erro. A configuração "Off" (padrão) suprime a linha do rodapé.

## Arquivos de Log

Para gerenciar um servidor Web de forma eficaz é necessário obter feedback sobre sua atividade e seu desempenho, assim como quaisquer problemas que possam ocorrer. O Servidor Web Apache fornece recursos de log muito abrangentes e flexíveis.

Os nomes dos arquivos de log podem ser definidos por diretivas, como por exemplo a **ErrorLog**, que define o nome do arquivo de log de erros. Porém, os nomes de arquivos são comumente "**error\_log**" em sistemas Unix-like e "**error.log**" em sistemas Windows e OS/2, quando se trata de erros. Em sistemas Unix-like também é possível fazer com que o servidor envie os erros para o syslog ou para um determinado programa.

Quando se trata de acessos em geral (os mais diversos tipos), o nome comumente utilizado é "**access\_log**" (Unix-like) ou "**access.log**" (Windows e OS/2). Esse log pode ser personalizado, de acordo com as necessidades.

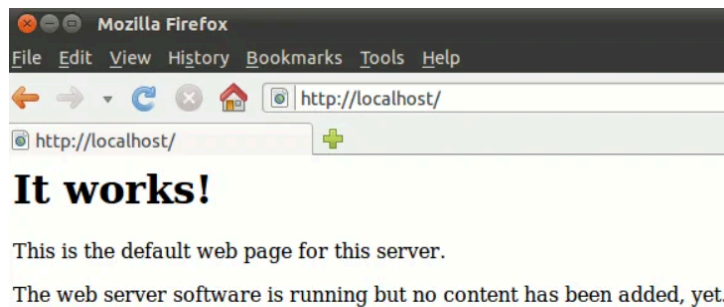
Por padrão, os arquivos de log ficam armazenados em:

- Linux (Debian/Ubuntu): /var/log/apache2/
- Linux (Red Hat/CentOS): /var/log/httpd/
- Windows: C:\Program Files\Apache Group\Apache2\logs\

## Testando o Servidor Apache

Após a instalação e configuração do servidor Web Apache, uma forma de testar é abrir um navegador no próprio servidor e colocar o endereço IP "127.0.0.1" (na verdade pode ser outras combinações de 127.x.x.x, mas o padrão é 127.0.0.1), ou "localhost". Então, na barra de endereços do navegador fica <http://127.0.0.1/>, ou <http://localhost/>. Se estiver funcionando, deve aparecer a página abaixo (ou semelhante):





Se o teste for realizado a partir de outro computador, deve-se colocar o endereço IP do servidor no lugar de "localhost" ou "127.0.0.1". Se o DNS estiver configurado, pode-se testar com o nome de domínio (ex.: <http://www.estrategiaconcursos.com.br>).

## Apache + PHP

PHP (*Hypertext Preprocessor*) é uma linguagem de programação amplamente utilizada no desenvolvimento de aplicativos Web dinâmicos. Ele roda do lado do servidor, processando a lógica antes que os resultados sejam enviados ao cliente (navegador). É bastante utilizado para trabalhar com banco de dados, lidar com formulários e gerar conteúdo dinâmico.

Algumas características do PHP são:

- Aprendizagem fácil: É fácil para iniciantes, sendo uma das linguagens de programação mais populares para o desenvolvimento Web;
- Interação com banco de dados: É altamente integrado com vários sistemas de banco de dados, como MySQL, PostgreSQL, entre outros;
- Flexibilidade: Pode ser incorporado diretamente no HTML, tornando a criação de páginas dinâmicas simples e intuitiva.

O Apache, em conjunto com o PHP trabalham da seguinte forma:

- Apache atua como servidor Web, gerenciando as requisições HTTP enviadas pelo navegador dos usuários;
- PHP é interpretado no servidor Apache quando o código PHP é solicitado. Ex.: quando uma página .php é acessada;
- Quando um usuário acessa um *site* com PHP (ex.: `index.php`), o Apache encaminha essa requisição para o interpretador PHP;
- O PHP processa o código e gera o conteúdo final, geralmente em HTML, possivelmente com dados dinâmicos de um banco de dados. O Apache então retorna esse conteúdo gerado ao navegador do usuário.





Para configurar o Apache para usar o PHP, geralmente é necessário instalar o PHP junto com o Apache, o que é feito comumente através de pacotes como o LAMP (Linux, Apache, MySQL, PHP) ou XAMPP (multiplataforma).

Os passos básicos para instalar no Linux são:

- Instalação do Apache: `sudo apt install apache2`
- Instalação do PHP: `sudo apt install php libapache2-mod-php`
- Para reiniciar o Apache: `sudo systemctl restart apache2`

Após essa configuração, pode-se criar arquivos `.php` e colocá-los no diretório de *sites* do Apache, normalmente `/var/www/html` (em sistemas Linux).

Algumas vantagens do uso Apache + PHP são:

- Flexibilidade e personalização: O Apache possui um sistema modular e o PHP é fácil de utilizar. Com isso, em conjunto eles oferecem uma grande flexibilidade para desenvolver e configurar servidores e aplicativos Web;
- Suporte de comunidade: Ambos possuem uma grande comunidade de suporte, com documentação abrangente, fóruns e recursos de aprendizado;
- Segurança: Configurações de segurança específicas podem ser aplicadas para proteger tanto o Apache quanto o PHP, garantindo a segurança dos *sites* e dados dos usuários.



## QUESTÕES COMENTADAS – CEBRASPE

1. (CEBRASPE/FUB/2015) O Apache, que provê páginas por meio do protocolo HTTP, possui código-fonte aberto, funciona em ambiente multiplataforma, tanto no Windows quanto no Linux.

Comentários:

O Apache é um servidor Web livre (possui código-fonte aberto) que funciona em ambiente multiplataforma (Windows, Novell, OS/2, Unix, Linux, FreeBSD etc.).

Gabarito: Certo

2. (CEBRASPE/STJ/2015) Com relação ao servidor Apache, julgue o próximo item.

As diretrizes <Directory> e <Files> são utilizadas em arquivos htaccess para permitir que usuários controlem o acesso a seus arquivos.

Comentários:

Essas duas diretrizes servem para delimitar quais diretrizes devem ser aplicadas a um diretório (<Directory>) e a arquivos (<Files>).

Gabarito: Errado

3. (CEBRASPE/STJ/2015) Com relação ao servidor Apache, julgue o próximo item.

Um administrador pode incluir uma configuração para determinado diretório por meio da diretriz <Directory>.

Comentários:

<Directory> é utilizada para definir um grupo de diretrizes que devem ser aplicadas apenas ao diretório definido, seus subdiretórios, e aos arquivos dentro deles. Exemplo:

```
<Directory "/usr/local/httpd/htdocs">  
    Options Indexes FollowSymLinks  
</Directory>
```

Gabarito: Certo

4. (CEBRASPE/TRT8/2016) A diretiva que limita a apresentação dos arquivos que têm a extensão .conf, em um servidor Apache Web Server, é

- A) Deny \*.conf
- B) Allow - conf
- C) Directory \*.conf
- D) Location - conf
- E) IndexIgnore \*.conf

Comentários:



A diretiva IndexIgnore adiciona à lista de arquivos a serem escondidos quando listar um diretório. Pode utilizar caracteres coringas ("?" e "\*"). Múltiplas diretivas IndexIgnore acrescentam na lista, ao invés de substituir. Por padrão, a lista contém "." (o diretório corrente). Exemplo:

```
<Directory "/var/www">  
    IndexIgnore *.conf  
</Directory>
```

Gabarito: E

5. (CEBRASPE/TRT8/2016) A diretiva que limita a apresentação dos arquivos que têm a extensão .conf, em um servidor Apache Web Server, é

- A) Deny \*.conf
- B) Allow - conf
- C) Directory \*.conf
- D) Location - conf
- E) IndexIgnore \*.conf

Comentários:

A diretiva IndexIgnore adiciona à lista de arquivos a serem escondidos quando listar um diretório. Pode utilizar caracteres coringas ("?" e "\*"). Múltiplas diretivas IndexIgnore acrescentam na lista, ao invés de substituir. Por padrão, a lista contém "." (o diretório corrente). Exemplo:

```
<Directory "/var/www">  
    IndexIgnore *.conf  
</Directory>
```

Gabarito: E

6. (CEBRASPE/ABIN/2018) A determinação da porta e do endereço que o servidor Apache irá escutar deve ser feita por meio da diretiva listen. Sem sua definição, o servidor Apache se mantém fora de operação.

Comentários:

A diretiva Listen instrui o httpd a escutar endereços IP específicos ou portas específicas. Por padrão, o servidor responde a requisições em todas as interfaces IP. Essa diretiva é obrigatória (a partir da versão 2.4), ou seja, se não estiver presente no arquivo de configuração, o servidor falhará ao iniciar.

Pessoal, a questão não deixa claro qual a versão do Apache! Segundo a documentação, a partir da versão 2.4 seria obrigatória a diretiva listen! Na minha opinião, caberia recurso, pois a banca definiu como errada a questão!

Gabarito: Errado



7. (CEBRASPE/EBSERH/2018) Para que arquivos para funcionamento de um sítio web armazenados no diretório /var/www/sitio01 fiquem acessíveis via HTTP usando o Apache, é necessário incluir, no arquivo de configuração do servidor, a seguinte linha.

DocumentRoot /var/www/sitio01

#### Comentários:

Existem dois tipos de páginas que podem ser adicionadas ao Apache: a página raiz e subpáginas. A página raiz é especificada através da diretiva DocumentRoot e será mostrada quando se entrar no domínio principal, como <http://www.estrategiaconcursos.com.br>. Na configuração padrão do Apache, DocumentRoot aponta para o diretório /var/www. Esse diretório será assumido como raiz caso os diretórios não sejam iniciados por uma /:

home/estrategia -> aponta para /var/www/home/estrategia

/home/estrategia -> Aponta para /home/estrategia

Gabarito: Certo

8. (CEBRASPE/TJ-AM/2019) Julgue o item a seguir, com relação a configurações do servidor Apache.

A diretiva DocumentRoot "/paginas" indica que o diretório paginas será o principal caminho para que o Apache procure por uma página inicial a ser exibida.

#### Comentários:

A página raiz é especificada através da diretiva DocumentRoot e será mostrada quando o usuário entrar no domínio principal, como por exemplo <http://www.estrategiaconcursos.com.br>. Na configuração padrão do Apache, DocumentRoot aponta para o diretório /var/www. Esse diretório será assumido como raiz caso os diretórios não sejam iniciados por uma "/". Vejamos:

- home/estrategia → Aponta para /var/www/home/estrategia
- /home/estrategia → Aponta para /home/estrategia

Gabarito: Certo

9. (CEBRASPE/TJ-AM/2019) Julgue o item a seguir, com relação a configurações do servidor Apache.

Se o daemon do Apache estiver em funcionamento, o comando `apachectl graceful`, ao ser executado, encerrará imediatamente a operação do servidor Apache.

#### Comentários:

Utilizando-se o `apachectl`, com o argumento "-k", temos:

- -k start|restart|graceful|stop|graceful-stop: sinaliza o httpd para iniciar, reiniciar ou parar. Quando utilizado no modo de script de inicialização, o `apachectl` aceita como argumentos:
  - ✓ start: inicializa o *daemon* httpd;
  - ✓ stop: finaliza o *daemon* httpd;
  - ✓ restart: reinicializa o *daemon* httpd. Se não estiver rodando, o httpd é inicializado;
  - ✓ fullstatus: mostra um relatório completo do status do `mod_status`. Para isso, o módulo `mod_status` deve estar habilitado;



- ✓ status: mostra um relatório resumido do status. Similar à opção fullstatus, mas a lista de requisições correntes que estão sendo servidas é omitida;
- ✓ graceful: reinicializa o *daemon* de forma "gentil". Se não estiver rodando, o httpd é inicializado. A diferença para uma inicialização normal é que as conexões correntes não são abortadas;
- ✓ **graceful-stop**: finaliza o *daemon* de forma "gentil". A diferença para uma finalização normal é que as conexões correntes não são abortadas.

Gabarito: Errado

10.(CEBRASPE/TJ-AM/2019) Julgue o item a seguir, com relação a configurações do servidor Apache.

Para que URLs possam ser reescritas, é correto utilizar o módulo mod\_rewrite.

Comentários:

mod\_rewrite: fornece um mecanismo de reescrita baseado em regras para reescrever os URLs solicitados em tempo real. Por exemplo, pode reescrever um endereço "http" para "https".

Gabarito: Certo

11.(CEBRASPE/Min. da Economia/2020) Julgue o item subsecutivo, a respeito de arquiteturas e servidores web.

No Apache httpd, a funcionalidade mais básica está incluída no núcleo do servidor; os recursos estendidos por meio de módulos devem ser compilados; módulos são estáticos; e o httpd deve ser recompilado para adicionar ou remover módulos.

Comentários:

Os módulos são implementados de duas maneiras: **estáticos ou dinâmicos**. Os estáticos são inseridos durante a compilação e são incorporados ao código binário do servidor. Os dinâmicos são ligados (link) ao código binário do servidor em tempo de execução.

Gabarito: Errado

12.(CEBRASPE/Min. da Economia/2020) Julgue o item subsecutivo, a respeito de arquiteturas e servidores web.

O servidor Apache http permite o gerenciamento descentralizado de configuração por meio de arquivos especiais chamados de .htaccess, não sendo permitido o uso de outro nome de arquivo.

Comentários:

Os arquivos .htaccess ("arquivos de configuração distribuída") oferecem uma maneira de realizar mudanças nas configurações por diretório. Um arquivo, contendo uma ou mais diretrizes de configuração, é colocado em determinado diretório, e as diretrizes se aplicam apenas para aquele diretório e todos os seus subdiretórios. Uma observação importante é que é possível renomear o arquivo ".htaccess" para outro nome qualquer, sendo que para isso é necessário utilizar a diretriz "AccessFileName". Por exemplo, se quiser que o arquivo se chame ".configuracao", então pode-se adicionar a seguinte linha ao arquivo de configuração do servidor:

```
AccessFileName .configuracao
```



Gabarito: Errado

13.(CEBRASPE/Min. da Economia/2020) Julgue o item subsecutivo, a respeito de arquiteturas e servidores web.

O servidor Apache http tenta manter um pool de threads de servidor sobressalentes ou ociosos, que estão prontos para atender às novas solicitações, de modo que os clientes não precisam esperar que novos threads ou processos sejam criados antes que suas solicitações possam ser atendidas, podendo essa configuração ser definida por meio da diretiva StartServers.

Comentários:

StartServers define o número de processos filho (child) do servidor a serem criados na inicialização. Como o número de processos é controlado dinamicamente dependendo da carga, geralmente há poucas razões para ajustar esse parâmetro. O valor padrão difere de MPM para MPM. Para "worker" e "event", o valor padrão de StartServers é 3; prefork é padronizado para 5; e o padrão de mpmt\_os2 é 2.

Gabarito: Certo

14.(CEBRASPE/APEX Brasil/2021) Servidores web como o Apache, que disponibilizam aplicações acessíveis pelo protocolo HTTP, também disponibilizam, por default, a porta 80 para escuta. Considerando esses servidores web, assinale a opção que indica o módulo necessário para que uma URL seja automaticamente redirecionada de HTTP para HTTPS, de modo que a conexão disponibilizada suporte criptografia.

- A) mod\_rewrite
- B) Mod\_ldap
- C) mod\_mime
- D) Mod\_session

Comentários:

mod\_rewrite: fornece um mecanismo de reescrita baseado em regras para reescrever os URLs solicitados em tempo real. Por exemplo, pode reescrever um endereço "http" para "https".

Gabarito: A

15.(CEBRASPE/SERPRO/2021) A respeito de servidores Apache, julgue o item a seguir.

Aplicando-se a configuração a seguir, é possível exibir imagens pelo servidor HTTP, desde que sejam disponibilizados os arquivos dentro do diretório /var/www/html.

```
DocumentRoot "/var/www/html"
```

Comentários:

Existem dois tipos de páginas que podem ser adicionadas ao Apache: a página raiz e as subpáginas. A página raiz é especificada através da diretiva `DocumentRoot` e será mostrada quando o usuário entrar no domínio principal, como por exemplo `http://www.estrategiaconcursos.com.br`. Na configuração padrão do Apache, `DocumentRoot`



aponta para o diretório /var/www. Esse diretório será assumido como raiz caso os diretórios não sejam iniciados por uma "/". Vejamos:

- home/estrategia → Aponta para /var/www/home/estrategia
- /home/estrategia → Aponta para /home/estrategia

Gabarito: Certo

16. (CEBRASPE/SERPRO/2021) A respeito de servidores Apache, julgue o item a seguir.

Para configurar uma porta número 8443 que responda a conexões HTTP seguras no endereço IP 192.168.10.10, deve ser informada a configuração a seguir.

Listen 192.168.10.10:8443 https

Comentários:

A diretiva Listen instrui o httpd a escutar endereços IP específicos ou portas específicas. Por padrão, o servidor responde a requisições em todas as interfaces IP. Essa diretiva é obrigatória (a partir da versão 2.4), ou seja, se não estiver presente no arquivo de configuração, o servidor falhará ao iniciar. Diretivas múltiplas podem ser utilizadas para especificar diferentes endereços ou portas. Exemplo:

Listen 80

Listen 8000

Para especificar duas interfaces e suas portas, utiliza-se, por exemplo:

Listen 192.188.1.1:80

Listen 192.188.1.2:8000

Endereços IPv6 devem ser colocados entre colchetes:

Listen [2001:db8::a00:20ff:fea7:ccea]:80

Para o protocolo HTTPS (HTTP seguro), a porta padrão é a 443, ou seja, se nada for especificado, o servidor assume essa porta. Para configurar outra porta, utiliza-se:

Listen 192.188.1.2:9443 https

Gabarito: Certo

17. (CEBRASPE/SERPRO/2021) A respeito de servidores Apache, julgue o item a seguir.

A configuração ServerTokens OS é capaz de controlar informações de cabeçalho que são retornadas pelo servidor ao cliente, bem como limitar a quantidade de informações por segurança, sem fornecer a versão do servidor, mas somente o nome dele.

Comentários:





ServerTokens define se o campo de cabeçalho de resposta do servidor (que é enviado aos clientes) inclui uma descrição do tipo de sistema operacional genérico do servidor, bem como informações sobre módulos compilados. Abaixo exemplos de resposta de acordo com o parâmetro utilizado.

- ServerTokens Full (ou não especificado) - o servidor envia, por exemplo, "Server: Apache/2.4.2 (Unix) PHP/4.2.2 MyMod/1.2"
- ServerTokens Prod[uctOnly] - o servidor envia, por exemplo, "Server: Apache"
- ServerTokens Major - o servidor envia, por exemplo, "Server: Apache/2"
- ServerTokens Minor - o servidor envia, por exemplo, "Server: Apache/2.4"
- ServerTokens Min[imal] - o servidor envia, por exemplo, "Server: Apache/2.4.2"
- **ServerTokens OS** - o servidor envia, por exemplo, "Server: Apache/2.4.2 (Unix)"

Gabarito: Errado

18.(CEBRASPE/CODEVASF/2021) No que se refere ao servidor de aplicação Apache, julgue o item a seguir.

Arquivos do tipo .htaccess são capazes de alterar configurações por diretório, podendo aplicar as configurações tanto no diretório onde se encontram quanto nos respectivos subdiretórios.

Comentários:

Os arquivos .htaccess ("arquivos de configuração distribuída") oferecem uma maneira de realizar mudanças nas configurações por diretório. Um arquivo, contendo uma ou mais diretrizes de configuração, é colocado em determinado diretório, e as diretrizes se aplicam apenas para aquele diretório e todos os seus subdiretórios. Uma observação importante é que é possível renomear o arquivo ".htaccess" para outro nome qualquer, sendo que para isso é necessário utilizar a diretriz "AccessFileName". Por exemplo, se quiser que o arquivo se chame ".configuracao", então pode-se adicionar a seguinte linha ao arquivo de configuração do servidor:

```
AccessFileName .configuracao
```

Gabarito: Certo

19.(CEBRASPE/Petrobras/2022) A respeito da configuração do servidor Apache, julgue o item subsequente.

A diretiva de configuração SSL Engine deve receber como parâmetro o caminho da chave privada do certificado digital que será utilizado para o servidor.

Comentários:

SSL Engine alterna o uso do mecanismo de protocolo SSL/TLS. Essa diretiva deve ser usada dentro de uma seção <VirtualHost> para habilitar o SSL/TLS para esse host virtual. Por padrão, o



mecanismo de protocolo SSL/TLS está desabilitado para o servidor principal e para todos os hosts virtuais configurados. Ex.:

```
<VirtualHost _default_:443>
```

```
    SSLEngine on
```

```
    #...
```

```
</VirtualHost>
```

Gabarito: Errado

20.(CEBRASPE/Petrobras/2022) A respeito da configuração do servidor Apache, julgue o item subsequente.

Se DirectoryIndex index.html for informado na configuração, o servidor, ao acessar diretórios, buscará por um arquivo de índice com nome index.html.

Comentários:

DirectoryIndex define a lista de recursos a serem procurados, quando o cliente solicita um índice do diretório especificando uma barra ("/") no final do nome do diretório. A URL local é a (URL codificada em %) de um documento no servidor relativo ao diretório solicitado; geralmente é o nome de um arquivo no diretório. Vários URLs podem ser fornecidos, caso em que o servidor retornará o primeiro que encontrar. Se nenhum dos recursos existir e a opção "Indexes" estiver definida, o servidor gerará sua própria listagem do diretório. Ex.: DirectoryIndex index.html.

Gabarito: Certo

21.(CEBRASPE/DPE-RO/2022) No Apache, a diretiva Listen é utilizada para um maior controle sobre a especificação de endereços e(ou) portas alternativas das requisições externas que o servidor web espera. Nela, o endereço a ser usado é o da interface de rede (assim como na diretiva BindAddress). A seguir, é apresentado um exemplo dessa diretiva.

```
Listen 192.168.1.1:80
```

```
Listen 192.168.7.1:81
```

```
Listen 50000
```

Nesse exemplo, o servidor httpd esperará por requisições do endereço 192.168.1.\*

A) na porta 80, de 192.168.7.1 na porta 81 e de qualquer endereço na porta 50000.

B) na porta 80, somente.

C) nas portas 80 e 50000, somente.

D) na porta 80 e de 192.168.7.1 na porta 81, somente.

E) nas portas 80 e 50000 e de 192.168.7.1 nas portas 81 e 50000.

Comentários:



O foco desta questão é a interpretação. No fim do enunciado diz que "o servidor httpd esperará por requisições do endereço 192.168.1.\*", então, podemos ver que escuta na porta 80 para 192.168.1.1 e porta 50000 para qualquer endereço IP (incluindo aqueles que começam com 192.168.1).

Para o endereço IP 192.168.7.1, escuta nas portas 81 e 50000 (esta última serve para qualquer endereço IP, na verdade).

Gabarito: E

22. (CEBRASPE/DPE-RO/2022) O Apache é um servidor web de alto desempenho, cujos arquivos de configuração apresentam as características a seguir.

I srm.conf: arquivo que possui as diretivas que controlam a operação do daemon servidor.

II httpd.conf: é o principal arquivo de configuração do Apache.

III access.conf: arquivo que contém as diretivas que controlam o acesso aos documentos.

Assinale a opção correta.

- A) Apenas o item I está certo.
- B) Apenas o item II está certo.
- C) Apenas os itens I e III estão certos.
- D) Todos os itens estão certos.
- E) Apenas os itens II e III estão certos.

Comentários:

Abaixo vemos os arquivos de configuração, que por padrão ficam em "/etc/apache".

- httpd.conf: arquivo de configuração principal, possuindo diretivas que controlam a operação do *daemon* servidor. Um arquivo de configuração alternativo pode ser especificado através da opção "-f" da linha de comando;
- srm.conf: possui diretivas que controlam a especificação de documentos que o servidor oferece aos clientes. O nome desse arquivo pode ser substituído através da diretiva ResourceConfig no arquivo principal de configuração;
- access.conf: possui diretivas que controlam o acesso aos documentos. O nome desse arquivo pode ser substituído através da diretiva AccessConfig no arquivo principal de configuração.

Gabarito: E



## QUESTÕES COMENTADAS – FGV

1. (FGV/MPE-BA/2017) O servidor web Apache é utilizado por diversas organizações em todo o mundo. Em relação ao servidor HTTP Apache versão 2.4, o comando que pode ser utilizado para verificar se o arquivo de configuração principal `httpd.conf` NÃO contém erro de sintaxe de configuração é:

- A) `httpd -t`
- B) `httpd -configtest`
- C) `httpd -l`
- D) `httpd -v`
- E) `httpd -f`

### Comentários:

`-t`: executa testes de sintaxe apenas para arquivos de configuração. O programa sai imediatamente após esses testes de análise de sintaxe com um código de retorno 0 (sintaxe OK) ou um código de retorno diferente de 0 (erro de sintaxe);

`-configtest`: executa um teste no arquivo de configuração. Após verificar a sintaxe, informa se está ok ou aponta os erros encontrados;

`-l`: mostra uma lista de módulos compilados no servidor. Isso não listará os módulos carregados dinamicamente incluídos através da diretiva `LoadModule`;

`-v`: imprime a versão do `httpd`;

`-f config`: usa as diretivas no arquivo de configuração na inicialização. Se "config" não começar com /, então será considerado um caminho relativo ao `ServerRoot`. O padrão é `conf/httpd.conf`.

Gabarito: A

2. (FGV/SEPOG-RO/2017) O servidor Apache HTTP, versão 2.4, tem a possibilidade de responder aos diversos endereços IPs, aos nomes de hosts e às portas em uma única máquina, em uma facilidade que é conhecida como host virtual, configurada no arquivo `httpd.conf`.

Com relação a essa facilidade, analise as afirmativas a seguir.

I. Várias diretivas `Listen` com o mesmo endereço IP e porta irão resultar em um erro fatal, impedindo que o servidor HTTP inicie corretamente.

II. O uso da diretiva `VirtualHost` no arquivo `httpd.conf` precisa ainda da respectiva configuração no servidor DHCP para seu correto funcionamento.

III. Várias diretivas `VirtualHost` podem ser utilizadas para configurar diversos hosts virtuais em uma mesma máquina, mas apenas se utilizarem portas diferentes.

Está correto o que se afirma em

- A) I, apenas.



- B) II, apenas.
- C) III, apenas.
- D) I e III, apenas.
- E) I, II e III.

#### Comentários:

- I. CORRETA - Não é possível escutar no mesmo endereço e porta, no mesmo servidor.
- II. ERRADA - O servidor DHCP envia os endereços IP a serem utilizados pelos clientes. A diretiva VirtualHost não necessita do DHCP. Na verdade, geralmente se utiliza endereço IP fixos em servidores!
- III. ERRADA - Podem utilizar a mesma porta, desde que sejam endereços IP diferentes.

Gabarito: A

3. (FGV/Câmara de Salvador-BA/2018) Considere os seguintes fragmentos de configuração para o Apache Web Server.

```
<... 172.30.40.40>  
    DocumentRoot "/www/exemplo1"  
    ServerName www.exemplo.com.br  
<...>  
  
<... 172.30.40.50>  
    DocumentRoot "/www/exemplo2"  
    ServerName www.exemplo.org.br  
</...>
```

O termo que substitui corretamente os trechos pontilhados é:

- A) Environment;
- B) HttpRequest;
- C) IPAddress;
- D) LocalServer;
- E) VirtualHost.

#### Comentários:

<VirtualHost> permite servir mais de um site no mesmo servidor (sites virtuais). Podem ser utilizadas diretivas específicas para o controle do site virtual, como nome do administrador, erros de acesso à página, controle de acesso e outros dados úteis para personalizar e gerenciar o site.

Virtual Hosts baseados em nomes: utiliza nomes para identificar os *sites* servidos e requerem somente um endereço IP. Assim é possível servir um número ilimitado de *sites* virtuais. O navegador do cliente deve suportar os cabeçalhos necessários para garantir o funcionamento desse recurso (os navegadores mais comuns possuem tal suporte). Exemplo:

```
<VirtualHost www.site1.com.br>
```

```
    ServerName www.site1.com.br
```



```
ServerAdmin site1@site1.com.br  
  
DocumentRoot /var/www/www_site1_com_br  
  
TransferLog /var/log/apache/site1/access.log  
  
ErrorLog /var/log/apache/site1/error.log  
  
User www-data  
  
Group www-data
```

</VirtualHost>

Gabarito: E

4. (FGV/Banestes/2018) A diretiva que pode ser incluída no arquivo de configuração do servidor Apache, versão 2.4, para que a URL

```
http://example.com/~pedro/file.html  
seja traduzida para o arquivo no endereço  
/var/www/pedro/docs/file.html
```

é:

- A) ServerAlias /var/www/~docs
- B) VirtualHost public\_html
- C) UserDir /var/www/\*/docs
- D) ServerName /var/www/~pedro/docs
- E) DocumentRoot docs

Comentários:

UserDir define o diretório real no diretório inicial de um usuário (home) para usar quando uma solicitação de um documento para um usuário é recebida. Exemplos:

UserDir	public_html	~bob/public_html/one/two.html
UserDir	/usr/web	/usr/web/bob/one/two.html
UserDir	/home/*/www	/home/bob/www/one/two.html

Gabarito: C

5. (FGV/Prefeitura de Niterói-RJ/2018) O arquivo httpd.conf é o arquivo de configuração principal do servidor Web Apache. Ele contém diretivas que controlam o funcionamento do servidor.

Assinale a opção que indica a diretiva que pode ser usada para especificar endereço e portas alternativas para o servidor web receber requisições externas.

- A) ServerRoot



- B) Listen
- C) AcceptFilter
- D) Redirect
- E) SetInputFilter

#### Comentários:

Para especificar duas interfaces e suas portas, utiliza-se, por exemplo:

```
Listen 192.188.1.1:80
```

```
Listen 192.188.1.2:8000
```

Gabarito: B

6. (FGV/Banestes/2021) O servidor HTTP Apache versão 2.4 é configurado por meio de diretivas presentes no arquivo de configuração denominado httpd.conf.

A diretiva que define o diretório base onde as páginas web devem ser armazenadas para disponibilização pelo servidor Apache é:

- A) DocumentRoot;
- B) ServerRoot;
- C) ServerPath;
- D) ServerName;
- E) ChRootDir.

#### Comentários:

Existem dois tipos de páginas que podem ser adicionadas ao Apache: a página raiz e as subpáginas. A página raiz é especificada através da diretiva DocumentRoot e será mostrada quando o usuário entrar no domínio principal, como por exemplo <http://www.estrategiaconcursos.com.br>. Na configuração padrão do Apache, DocumentRoot aponta para o diretório /var/www. Esse diretório será assumido como raiz caso os diretórios não sejam iniciados por uma "/". Vejamos:

- `home/estrategia` → Aponta para `/var/www/home/estrategia`
- `/home/estrategia` → Aponta para `/home/estrategia`

Gabarito: A

7. (FGV/TJ-TO/2022) A diretiva que pode ser utilizada no arquivo de configuração httpd.conf do servidor HTTP Apache versão 2.4 para definir o número limite de requisições simultâneas que serão atendidas pelo servidor é:

- A) MaxRequestWorkers;
- B) MaxOpenConnections;
- C) MaxKeepAliveRequests;





- D) StreamMaxMemSize;
- E) HeartbeatMaxServers.

Comentários:

O nome é bem intuitivo. MaxRequestWorkers: número máximo de conexões que serão processadas simultaneamente.

Gabarito: A



## QUESTÕES COMENTADAS – FCC

1. (FCC/Prefeitura de Teresina-PI/2016) Uma das formas de se iniciar o servidor Apache é por meio do comando

- A) apache inic
- B) apachectl start
- C) apachectl run
- D) apachectl go
- E) apache send

### Comentários:

Quando utilizado no modo de script de inicialização, o apachectl aceita como argumentos:

- start: inicializa o daemon httpd;
- stop: finaliza o daemon httpd;
- restart: reinicializa o daemon httpd. Se não estiver rodando, o httpd é inicializado;
- etc.

Gabarito: B

2. (FCC/DPE-RS/2017) Considere que o servidor Apache 2 foi instalado em um computador servidor com sistema operacional CentOS. Para configurar o acesso às páginas desse servidor Apache deve-se editar o arquivo:

- A) /etc/apache/init.conf
- B) /bin/apache/httpd.init
- C) /etc/conf/httpd.init
- D) /lib/apache/init.conf
- E) /etc/httpd/conf/httpd.conf

### Comentários:

Em ambientes Unix-like os arquivos de configuração, por padrão, ficam localizados no diretório /etc/httpd/conf ou /etc/apache2. O servidor é configurado por um arquivo denominado httpd.conf (principal arquivo de configuração do Apache) e opcionalmente pode haver configurações para cada diretório utilizando arquivos com o nome .htaccess, onde é possível utilizar autenticação de usuário pelo próprio protocolo HTTP (combinação do arquivo .htaccess com o arquivo .htpasswd, que guardará os usuários e senhas criptografadas).

Gabarito: E

3. (FCC/SEMEF Manaus-AM/2019) Um programador deseja reiniciar o Servidor HTTP Apache Versão 2.4 de forma que os visitantes ativos do site possam concluir os downloads em andamento antes de o servidor ser reiniciado. Para isso deve usar o comando



- A) apachectl -k graceful
- B) apachectl -j restart
- C) restartserver -j graceful
- D) start -s graceful-stop
- E) runserver -j restart

#### Comentários:

Quando utilizado no modo de script de inicialização, o apachectl aceita como argumentos (pode-se utilizar "-k" antes do argumento):

- start: inicializa o daemon httpd;
- stop: finaliza o daemon httpd;
- restart: reinicializa o daemon httpd. Se não estiver rodando, o httpd é inicializado;
- graceful: reinicializa o daemon de forma "gentil". Se não estiver rodando, o httpd é inicializado. A diferença para uma inicialização normal é que as conexões correntes não são abortadas;
- graceful-stop: finaliza o daemon de forma "gentil". A diferença para uma finalização normal é que as conexões correntes não são abortadas;
- etc.

Gabarito: A

4. (FCC/TRT5/2022) Considere o programa da configuração de host virtual do Apache HTTP server, que será executado em condições ideais.

< | >

```
DocumentRoot "/var/www/trt5.infra/"
```

```
ServerName trt5.infra
```

```
CustomLog /var/log/httpd/trt5_access.log combined
```

```
ErrorLog /var/log/httpd/trt5.infra_error.log
```

</VirtualHost>

Para que o servidor responda as requisições, somente na porta 80, a partir de qualquer endereço IP configurado, a lacuna | deve ser corretamente preenchida com

- A) VirtualHost any eq 80
- B) VirtualHost address\_Family=any; port=80
- C) VirtualHost \*:80
- D) VirtualHost any\_IP:80
- E) VirtualHost \*.\*

#### Comentários:



<VirtualHost> permite servir mais de um *site* no mesmo servidor (*sites* virtuais). Podem ser utilizadas diretivas específicas para o controle do *site* virtual, como nome do administrador, erros de acesso à página, controle de acesso e outros dados úteis para personalizar e gerenciar o *site*.

*Virtual Hosts* baseados em nomes: utiliza nomes para identificar os *sites* servidos e requerem somente um endereço IP. Assim é possível servir um número ilimitado de *sites* virtuais. O navegador do cliente deve suportar os cabeçalhos necessários para garantir o funcionamento desse recurso (os navegadores mais comuns possuem tal suporte). Exemplo:

```
<VirtualHost www.site1.com.br>
```

```
    ServerName www.site1.com.br
```

```
    ServerAdmin site1@site1.com.br
```

```
    DocumentRoot /var/www/www_site1_com_br
```

```
    TransferLog /var/log/apache/site1/access.log
```

```
    ErrorLog /var/log/apache/site1/error.log
```

```
    User www-data
```

```
    Group www-data
```

```
</VirtualHost>
```

Abaixo temos um exemplo de host virtual em qualquer endereço IP (\*), que escuta na porta 80.

```
<VirtualHost *:80>
```

```
    ServerName www.exemplo1.com
```

```
    DocumentRoot /caminho/para/o/site1
```

```
</VirtualHost>
```

Gabarito: C



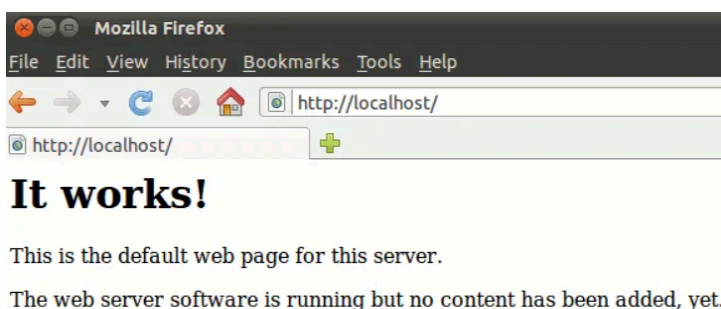
## QUESTÕES COMENTADAS – VUNESP

1. (VUNESP/Pref. de Registro-SP/2018) Considere o cenário em que você instalou um programa servidor de página web (Apache ou IIS) em um computador. Para testar se o servidor de página web está funcionando, utilizando um navegador web nesse computador, deve-se inserir na barra de endereços do navegador o endereço: http://

- A) 10.0.0.1.
- B) 127.0.0.1.
- C) 169.254.0.1.
- D) 172.16.0.1.
- E) 192.168.0.1.

### Comentários:

Após a instalação e configuração do servidor Web Apache, uma forma de testar é abrir um navegador no próprio servidor e colocar o endereço IP "127.0.0.1" (na verdade pode ser outras combinações de 127.x.x.x, mas o padrão é 127.0.0.1), ou "localhost". Então, na barra de endereços do navegador fica http://127.0.0.1/, ou http://localhost/. Se estiver funcionando, deve aparecer a página abaixo (ou semelhante):



Gabarito: B

2. (VUNESP/TJM-SP/2021) No arquivo de configuração httpd.conf de um servidor web Apache 2.4, foi encontrada a seguinte linha:

ServerSignature On

Essa configuração significa que

- A) será utilizada assinatura digital, que requer a instalação de um certificado, em documentos enviados do servidor web para os requisitantes.
- B) o protocolo HTTPS está habilitado, o que requer a instalação de um certificado digital, e o tráfego ocorrerá de forma segura entre os clientes e o servidor.
- C) uma listagem de conteúdo de arquivos será gerada e enviada automaticamente em respostas a requisições a diretórios que não possuam um arquivo de índice.



D) será exigido um usuário e senha para acessar determinados diretórios hospedados no servidor, definidos em arquivos indicados pela diretiva AuthUserFile na configuração de cada diretório protegido.

E) em documentos gerados pelo próprio servidor web, como páginas de erro, será adicionada uma linha de rodapé com informações do servidor.

#### Comentários:

ServerSignature permite a configuração de uma linha de rodapé à direita em documentos gerados pelo servidor (mensagens de erro, listagens de diretório ftp etc.). Uma razão para habilitar essa diretiva (valor "On") é que em uma cadeia de proxies, o usuário geralmente não tem possibilidade de dizer qual dos servidores encadeados realmente produziu uma mensagem de erro. A configuração "Off" (padrão) suprime a linha do rodapé.

Cuidado!!! É uma pegadinha, pois estamos acostumados com assinatura digital, e esse não é o caso!!!! É uma "assinatura" no rodapé, mesmo!

Gabarito: E

### 3. (VUNESP/UNICAMP/2022) No Apache HTTP Server, o módulo mod\_headers permite que

A) a URL da requisição seja modificada.

B) as requisições de entrada sejam balanceadas entre múltiplos servidores HTTP.

C) o servidor suporte SSL/TLS.

D) o método HEAD do protocolo HTTP seja suportado.

E) os cabeçalhos das requisições sejam customizados.

#### Comentários:

mod\_headers: fornece diretivas para controlar e modificar cabeçalhos de solicitação e resposta HTTP. Os cabeçalhos podem ser mesclados, substituídos ou removidos.

Gabarito: E



## QUESTÕES COMENTADAS – MULTIBANCAS

1. (FIOCRUZ/FIOCRUZ/2016) São exemplos de comandos para reiniciar o Apache nas versões linux RedHat e Ubuntu, respectivamente:

- A) service apache restart ou /etc/init.d/http restart.
- B) service httpd restart ou /etc/init.d/apache2 restart.
- C) /etc/init.d/apache2 reload ou /etc/init.d/apache2 restart.
- D) /etc/init.d/httpd reload ou /etc/init.d/apache2 restart.
- E) service apache reload ou /etc/init.d/http reload.

### Comentários:

Para inicializar o httpd, vimos que um comando possível é apachectl start, mas há outras formas também, dependendo da distribuição utilizada:

- Red Hat: service httpd start;
- Ubuntu: /etc/init.d/apache2 start.

O argumento "start" pode ser trocado por "restart", "stop" etc., para as demais ações.

Gabarito: B

2. (CCV-UFC/UFC/2016) Qual das diretivas abaixo deve ser configurada no arquivo de configuração do servidor Apache para informar se serão aceitas ou não conexões HTTP persistentes?

- A) Mutex.
- B) Timeout.
- C) KeepAlive.
- D) CacheEnable.
- E) HostnameLookups.

### Comentários:

A diretiva KeepAlive habilita/desabilita conexões HTTP persistentes. MaxKeepAliveRequests define o número máximo de requisições permitidas (0 = sem limite). KeepAliveTimeout define o tempo (em segundos) a esperar para a próxima requisição do mesmo cliente (na mesma conexão). Exemplo:

```
KeepAlive On
MaxKeepAliveRequests 50
KeepAliveTimeout 10
```

Gabarito: C

3. (CS-UFG/CELG-GT-GO/2017) Na configuração do Apache HTTP Server (httpd), o uso da diretiva <virtualHost> indica que o servidor Web irá





- A) executar em uma máquina virtual.
- B) executar em um servidor virtual em ambiente de nuvem
- C) rodar mais de um website em uma mesma máquina.
- D) ser replicado em várias máquinas, embora aparente ser um único host.

**Comentários:**

<VirtualHost> permite servir mais de um site no mesmo servidor (sites virtuais). Podem ser utilizadas diretivas específicas para o controle do site virtual, como nome do administrador, erros de acesso a página, controle de acesso e outros dados úteis para personalizar e gerenciar o site.

Gabarito: C

4. (SUGEP-UFRPE/UFRPE/2018) Numa instalação com o Servidor Apache existe a necessidade de alterar o arquivo de configuração. O arquivo de configuração padrão do Apache é o arquivo:

- A) httpd.conf
- B) apch.ini
- C) http.confg
- D) apache.ini
- E) apch.conf

**Comentários:**

No Linux é muito comum que o arquivo de configuração tenha a "extensão" .conf e muitas vezes o nome do arquivo é o mesmo nome do daemon, então fica httpd.conf. Ok, e no Windows? Segue o mesmo nome!

Gabarito: A

5. (Quadrix/CRM-PR/2018) Um servidor Apache pode hospedar muitos sites web diferentes, simultaneamente, com o uso do método chamado de Virtual hosting.

**Comentários:**

Exato! E através de diretivas, pode ser utilizada a <VirtualHost>.

Gabarito: Certo

6. (AOCP/UFPB/2019) Atualmente, o servidor web Apache é um dos mais utilizados na internet, pois a sua instalação e configuração são muito simples. Sobre o servidor web Apache, assinale a alternativa correta.

- A) Por padrão, o diretório raiz do servidor web é /var/www (no Debian) e /var/www/html (no Fedora). O diretório raiz pode ser alterado através da opção DocumentRoot, que fica dentro do arquivo principal de configuração do Apache.
- B) No servidor web Apache, somente pode ser hospedado um site por vez, não havendo forma de hospedar mais de um site por instalação.



- C) O servidor web Apache não é modular, ou seja, se for acessada uma página PHP, o próprio Apache faz a interpretação do código, sem requisitar nada ao módulo PHP.
- D) Para se verificar se o servidor web Apache foi instalado corretamente, deverá se acessar, por meio de um navegador, o site [apache.org](http://apache.org). Ele retornará o status do servidor web.
- E) O servidor web Apache é um sistema de código fonte fechado e pago.

#### Comentários:

- A) CORRETA - Existem dois tipos de páginas que podem ser adicionadas ao Apache: a página raiz e as subpáginas. A página raiz é especificada através da diretiva `DocumentRoot` e será mostrada quando o usuário entrar no domínio principal, como por exemplo <http://www.estrategiaconcursos.com.br>. Na configuração padrão do Apache, `DocumentRoot` aponta para o diretório `/var/www` (ou `/var/www/html`, dependendo da distribuição).
- B) ERRADA - Podem ser hospedados n *sites* e tem como aplicar *Virtual Hosts*.
- C) ERRADA - O servidor web Apache possui módulos estáticos (compilados junto com o servidor em si) e dinâmicos (carregados quando necessários, em tempo de execução).
- D) ERRADA - Para se verificar se o servidor web Apache foi instalado corretamente, pode-se acessar <http://localhost> (ou <http://127.0.0.1>) a partir de um navegador na máquina onde está o servidor Web.
- E) ERRADA - O Apache é um servidor Web gratuito e livre (possui código-fonte aberto) que funciona em ambiente multiplataforma (Windows, Novell, OS/2, Unix, Linux, FreeBSD etc.).

Gabarito: A

#### 7. (FEPESE/CELESC/2019) Qual o nome do principal arquivo de configuração do servidor Apache?

- A) `httpd.conf`
- B) `daemon.sh`
- C) `apache.conf`
- D) `apache.sh`
- E) `conf.sh`

#### Comentários:

Abaixo vemos os arquivos de configuração, que por padrão ficam em `"/etc/apache"`.

- `httpd.conf`: arquivo de configuração principal, possuindo diretivas que controlam a operação do *daemon* servidor. Um arquivo de configuração alternativo pode ser especificado através da opção `"-f"` da linha de comando;
- `srm.conf`: possui diretivas que controlam a especificação de documentos que o servidor oferece aos clientes. O nome desse arquivo pode ser substituído através da diretiva `ResourceConfig` no arquivo principal de configuração;



- access.conf: possui diretivas que controlam o acesso aos documentos. O nome desse arquivo pode ser substituído através da diretiva AccessConfig no arquivo principal de configuração.

Gabarito: A

8. (FEPESE/Prefeitura de Florianópolis-SC/2019) Analise as afirmativas abaixo sobre os logs dos servidores web Apache.

1. A apache possui ao menos dois logs denominados access\_log e error\_log.
2. Os logs do apache estão localizados por padrão em /var/log/apache2 em distribuições Linux Ubuntu.
3. Os logs de acesso do Apache não são passíveis de customização; isto é, não é possível acrescentar campos ou informações adicionais aos logs de acessos.

Assinale a alternativa que indica todas as afirmativas corretas.

- A) É correta apenas a afirmativa 1.  
B) São corretas apenas as afirmativas 1 e 2.  
C) São corretas apenas as afirmativas 1 e 3.  
D) São corretas apenas as afirmativas 2 e 3.  
E) São corretas as afirmativas 1, 2 e 3.

Comentários:

1. CORRETA - Os nomes dos arquivos de log podem ser definidos por diretivas, como por exemplo a ErrorLog, que define o nome do arquivo de log de erros. Porém, os nomes de arquivos são comumente "error\_log" em sistemas Unix-like e "error.log" em sistemas Windows e OS/2, quando se trata de erros. Em sistemas Unix-like também é possível fazer com que o servidor envie os erros para o syslog ou para um determinado programa.

Quando se trata de acessos em geral (os mais diversos tipos), o nome comumente utilizado é "access\_log" (Unix-like) ou "access.log" (Windows e OS/2).

2. CORRETA - Por padrão, os arquivos de log ficam armazenados em:

- Linux (Debian/Ubuntu): /var/log/apache2/
- Linux (Red Hat/CentOS): /var/log/httpd/
- Windows: C:\Program Files\Apache Group\Apache2\logs\

3. ERRADA - É possível personalizar os logs de acesso conforme a necessidade.

Gabarito: B

9. (UniFil/Prefeitura de M. C. Rondon-PR/2021) Criado em 1995 por Rob McCool, o servidor Apache ou Servidor HTTP Apache é o mais bem sucedido servidor web livre que existe. Trata-se de um servidor web muito popular, utilizado principalmente no Linux. Assim como qualquer servidor do tipo, o Apache é responsável por disponibilizar páginas e todos os recursos que podem ser acessados pelo internauta. Envio de e-mails, mensagens, compras online e diversas outras funções podem ser executadas graças a servidores como o Apache.



Assinale a alternativa que não faz parte das vantagens de um Servidor Apache.

- A) O Servidor Apache tem uma grande estabilidade no seu funcionamento, visto que o mesmo tem um baixo nível de bugs
- B) O Servidor Apache tem grande compatibilidade com sistemas e plataformas
- C) O Servidor Apache possui código aberto
- D) O Servidor Apache é de uso gratuito

Comentários:

A) DESVANTAGEM - O Servidor Apache, como qualquer outro software, sempre terá bugs, pois a cada atualização novas falhas podem surgir e outras são corrigidas. Acho complicado dizer que o Apache possua um alto nível de bugs, como a banca sugere, mas, perto das demais alternativas (abaixo), só sobrou esta como resposta.

B) VANTAGEM - O Servidor Apache tem grande compatibilidade com sistemas e plataformas (Windows, Unix-like, macOS, entre outros).

C) VANTAGEM - O Servidor Apache possui código aberto, sendo possível realizar auditorias no código-fonte.

D) VANTAGEM - O Servidor Apache é de uso gratuito, isso mesmo!

Gabarito: A

10.(Quadrix/CRT-04/2022) Quanto às tecnologias, às ferramentas, aos aplicativos e aos procedimentos associados à Internet, julgue o item.

O Apache é um exemplo de servidor web.

Comentários:

Apache é um dos servidores Web mais conhecidos, é gratuito e livre (possui código-fonte aberto) que funciona em ambiente multiplataforma (Windows, Novell, OS/2, Unix, Linux, FreeBSD etc.). No Linux o servidor utiliza o *daemon* httpd. Suas funcionalidades são mantidas através de uma estrutura de módulos, permitindo inclusive que o usuário escreva seus próprios módulos.

Gabarito: Certo

11.(CONSULPLAN/MPE-PA/2022) Considerando que Apache é um servidor web de código aberto utilizado para hospedagem de sites, analise as afirmativas a seguir.

I. Portas-padrão do Apache: HTTP na porta 80 e HTTPS na porta 443 (Certificado SSL).

II. A diretiva "alias" é responsável por mapear recursos no sistema de arquivos.

III. "Location" refere-se à localização do sistema de arquivos do servidor. "Directory" é usado para definir a URL.

IV. O arquivo de log de erro do Apache está em /var/log/httpd/access\_log.

V. Existem dois tipos de hospedagem virtual: baseado no nome e baseado no IP.

Está correto o que se afirma apenas em

A) III.



- B) III e IV.
- C) I, II e V.
- D) II, III e IV.

#### Comentários:

I. CORRETA - Essas são as portas padrão dos protocolos HTTP e HTTPS, ou seja, para qualquer servidor Web elas são as mesmas!

II. CORRETA - Alias é usada para associar um caminho a um diretório físico no sistema de arquivos do servidor, o que permite criar atalhos ou aliases para simplificar o acesso a determinados recursos em um site.

III. ERRADA - <Location> limita o escopo das diretivas incluídas por URL. É semelhante à diretiva <Directory> e inicia uma subseção que é encerrada com uma diretiva </Location>. As seções <Location> são processadas na ordem em que aparecem no arquivo de configuração, depois que as seções <Directory> e os arquivos .htaccess são lidos e depois das seções <Files>.

IV. ERRADA - O arquivo de log de erro do Apache está em /var/log/httpd/error\_log.

V. CORRETA - Virtual Hosts baseados em nomes utilizam nomes para identificar os sites servidos e requerem somente um endereço IP. Virtual Hosts baseados em endereços IP permitem hospedar múltiplos sites em um único servidor, cada um associado a um endereço IP específico.

Gabarito: C

12.(FEPESE/Pref. de Bal. Camboriú-SC/2023) Qual o nome e local correto do arquivo de configuração do Apache em distribuições Red Hat Enterprise Linux 9.xx?

- A) /etc/apache.conf
- B) /etc/httpd/admin.conf
- C) /etc/httpd/conf/main.conf
- D) /etc/httpd/conf/httpd.conf
- E) /etc/httpd/apache.conf

#### Comentários:

Em ambientes Unix-like os arquivos de configuração, por padrão, ficam localizados no diretório /etc/httpd/conf ou /etc/apache2. O servidor é configurado por um arquivo denominado httpd.conf (principal arquivo de configuração do Apache) e opcionalmente pode haver configurações para cada diretório utilizando arquivos com o nome .htaccess, onde é possível utilizar autenticação de usuário pelo próprio protocolo HTTP (combinação do arquivo .htaccess com o arquivo .htpasswd, que guardará os usuários e senhas criptografadas). No Linux, o arquivo de configuração principal geralmente é encontrado em **/etc/httpd/conf/httpd.conf** ou /etc/apache2/apache2.conf.

Gabarito: D



## LISTA DE QUESTÕES – CEBRASPE

1. (CEBRASPE/FUB/2015) O Apache, que provê páginas por meio do protocolo HTTP, possui código-fonte aberto, funciona em ambiente multiplataforma, tanto no Windows quanto no Linux.
2. (CEBRASPE/STJ/2015) Com relação ao servidor Apache, julgue o próximo item.  
As diretrizes <Directory> e <Files> são utilizadas em arquivos htaccess para permitir que usuários controlem o acesso a seus arquivos.
3. (CEBRASPE/STJ/2015) Com relação ao servidor Apache, julgue o próximo item.  
Um administrador pode incluir uma configuração para determinado diretório por meio da diretriz <Directory>.
4. (CEBRASPE/TRT8/2016) A diretiva que limita a apresentação dos arquivos que têm a extensão .conf, em um servidor Apache Web Server, é
  - A) Deny \*.conf
  - B) Allow - conf
  - C) Directory \*.conf
  - D) Location - conf
  - E) IndexIgnore \*.conf
5. (CEBRASPE/TRT8/2016) A diretiva que limita a apresentação dos arquivos que têm a extensão .conf, em um servidor Apache Web Server, é
  - A) Deny \*.conf
  - B) Allow - conf
  - C) Directory \*.conf
  - D) Location - conf
  - E) IndexIgnore \*.conf
6. (CEBRASPE/ABIN/2018) A determinação da porta e do endereço que o servidor Apache irá escutar deve ser feita por meio da diretiva listen. Sem sua definição, o servidor Apache se mantém fora de operação.



7. (CEBRASPE/EBSERH/2018) Para que arquivos para funcionamento de um sítio web armazenados no diretório /var/www/sitio01 fiquem acessíveis via HTTP usando o Apache, é necessário incluir, no arquivo de configuração do servidor, a seguinte linha.

DocumentRoot /var/www/sitio01

8. (CEBRASPE/TJ-AM/2019) Julgue o item a seguir, com relação a configurações do servidor Apache.

A diretiva DocumentRoot “/paginas” indica que o diretório paginas será o principal caminho para que o Apache procure por uma página inicial a ser exibida.

9. (CEBRASPE/TJ-AM/2019) Julgue o item a seguir, com relação a configurações do servidor Apache.

Se o daemon do Apache estiver em funcionamento, o comando apachectl graceful, ao ser executado, encerrará imediatamente a operação do servidor Apache.

10. (CEBRASPE/TJ-AM/2019) Julgue o item a seguir, com relação a configurações do servidor Apache.

Para que URLs possam ser reescritas, é correto utilizar o módulo mod\_rewrite.

11. (CEBRASPE/Min. da Economia/2020) Julgue o item subsecutivo, a respeito de arquiteturas e servidores web.

No Apache httpd, a funcionalidade mais básica está incluída no núcleo do servidor; os recursos estendidos por meio de módulos devem ser compilados; módulos são estáticos; e o httpd deve ser recompilado para adicionar ou remover módulos.

12. (CEBRASPE/Min. da Economia/2020) Julgue o item subsecutivo, a respeito de arquiteturas e servidores web.

O servidor Apache http permite o gerenciamento descentralizado de configuração por meio de arquivos especiais chamados de .htaccess, não sendo permitido o uso de outro nome de arquivo.

13. (CEBRASPE/Min. da Economia/2020) Julgue o item subsecutivo, a respeito de arquiteturas e servidores web.

O servidor Apache http tenta manter um pool de threads de servidor sobressalentes ou ociosos, que estão prontos para atender às novas solicitações, de modo que os clientes não precisam esperar que novos threads ou processos sejam criados antes que suas solicitações possam ser atendidas, podendo essa configuração ser definida por meio da diretiva StartServers.





14. (CEBRASPE/APEX Brasil/2021) Servidores web como o Apache, que disponibilizam aplicações acessíveis pelo protocolo HTTP, também disponibilizam, por default, a porta 80 para escuta. Considerando esses servidores web, assinale a opção que indica o módulo necessário para que uma URL seja automaticamente redirecionada de HTTP para HTTPS, de modo que a conexão disponibilizada suporte criptografia.

- A) mod\_rewrite
- B) Mod\_ldap
- C) mod\_mime
- D) Mod\_session

15. (CEBRASPE/SERPRO/2021) A respeito de servidores Apache, julgue o item a seguir.

Aplicando-se a configuração a seguir, é possível exibir imagens pelo servidor HTTP, desde que sejam disponibilizados os arquivos dentro do diretório /var/www/html.

```
DocumentRoot "/var/www/html"
```

16. (CEBRASPE/SERPRO/2021) A respeito de servidores Apache, julgue o item a seguir.

Para configurar uma porta número 8443 que responda a conexões HTTP seguras no endereço IP 192.168.10.10, deve ser informada a configuração a seguir.

```
Listen 192.168.10.10:8443 https
```

17. (CEBRASPE/SERPRO/2021) A respeito de servidores Apache, julgue o item a seguir.

A configuração ServerTokens OS é capaz de controlar informações de cabeçalho que são retornadas pelo servidor ao cliente, bem como limitar a quantidade de informações por segurança, sem fornecer a versão do servidor, mas somente o nome dele.

18. (CEBRASPE/CODEVASF/2021) No que se refere ao servidor de aplicação Apache, julgue o item a seguir.

Arquivos do tipo .htaccess são capazes de alterar configurações por diretório, podendo aplicar as configurações tanto no diretório onde se encontram quanto nos respectivos subdiretórios.

19. (CEBRASPE/Petrobras/2022) A respeito da configuração do servidor Apache, julgue o item subsequente.

A diretiva de configuração SSLEngine deve receber como parâmetro o caminho da chave privada do certificado digital que será utilizado para o servidor.

20. (CEBRASPE/Petrobras/2022) A respeito da configuração do servidor Apache, julgue o item subsequente.



Se DirectoryIndex index.html for informado na configuração, o servidor, ao acessar diretórios, buscará por um arquivo de índice com nome index.html.

21.(CEBRASPE/DPE-RO/2022) No Apache, a diretiva Listen é utilizada para um maior controle sobre a especificação de endereços e(ou) portas alternativas das requisições externas que o servidor web espera. Nela, o endereço a ser usado é o da interface de rede (assim como na diretiva BindAddress). A seguir, é apresentado um exemplo dessa diretiva.

```
Listen 192.168.1.1:80
```

```
Listen 192.168.7.1:81
```

```
Listen 50000
```

Nesse exemplo, o servidor httpd esperará por requisições do endereço 192.168.1.\*

- A) na porta 80, de 192.168.7.1 na porta 81 e de qualquer endereço na porta 50000.
- B) na porta 80, somente.
- C) nas portas 80 e 50000, somente.
- D) na porta 80 e de 192.168.7.1 na porta 81, somente.
- E) nas portas 80 e 50000 e de 192.168.7.1 nas portas 81 e 50000.

22.(CEBRASPE/DPE-RO/2022) O Apache é um servidor web de alto desempenho, cujos arquivos de configuração apresentam as características a seguir.

I srm.conf: arquivo que possui as diretivas que controlam a operação do daemon servidor.

II httpd.conf: é o principal arquivo de configuração do Apache.

III access.conf: arquivo que contém as diretivas que controlam o acesso aos documentos.

Assinale a opção correta.

- A) Apenas o item I está certo.
- B) Apenas o item II está certo.
- C) Apenas os itens I e III estão certos.
- D) Todos os itens estão certos.
- E) Apenas os itens II e III estão certos.



## GABARITO

01	02	03	04	05	06
Certo	Errado	Certo	E	E	Errado
07	08	09	10	11	12
Certo	Certo	Errado	Certo	Errado	Errado
13	14	15	16	17	18
Certo	A	Certo	Certo	Errado	Certo
19	20	21	22		
Errado	Certo	E	E		



## LISTA DE QUESTÕES – FGV

1. (FGV/MPE-BA/2017) O servidor web Apache é utilizado por diversas organizações em todo o mundo. Em relação ao servidor HTTP Apache versão 2.4, o comando que pode ser utilizado para verificar se o arquivo de configuração principal `httpd.conf` NÃO contém erro de sintaxe de configuração é:

- A) `httpd -t`
- B) `httpd -configtest`
- C) `httpd -l`
- D) `httpd -v`
- E) `httpd -f`

2. (FGV/SEPOG-RO/2017) O servidor Apache HTTP, versão 2.4, tem a possibilidade de responder aos diversos endereços IPs, aos nomes de hosts e às portas em uma única máquina, em uma facilidade que é conhecida como host virtual, configurada no arquivo `httpd.conf`.

Com relação a essa facilidade, analise as afirmativas a seguir.

I. Várias diretivas `Listen` com o mesmo endereço IP e porta irão resultar em um erro fatal, impedindo que o servidor HTTP inicie corretamente.

II. O uso da diretiva `VirtualHost` no arquivo `httpd.conf` precisa ainda da respectiva configuração no servidor DHCP para seu correto funcionamento.

III. Várias diretivas `VirtualHost` podem ser utilizadas para configurar diversos hosts virtuais em uma mesma máquina, mas apenas se utilizarem portas diferentes.

Está correto o que se afirma em

- A) I, apenas.
- B) II, apenas.
- C) III, apenas.
- D) I e III, apenas.
- E) I, II e III.

3. (FGV/Câmara de Salvador-BA/2018) Considere os seguintes fragmentos de configuração para o Apache Web Server.



```
<... 172.30.40.40>  
  DocumentRoot "/www/exemplo1"  
  ServerName www.exemplo.com.br  
<...>  
  
<... 172.30.40.50>  
  DocumentRoot "/www/exemplo2"  
  ServerName www.exemplo.org.br  
</...>
```

O termo que substitui corretamente os trechos pontilhados é:

- A) Environment;
- B) HttpRequest;
- C) IPAddress;
- D) LocalServer;
- E) VirtualHost.

4. (FGV/Banestes/2018) A diretiva que pode ser incluída no arquivo de configuração do servidor Apache, versão 2.4, para que a URL

```
http://example.com/~pedro/file.html  
seja traduzida para o arquivo no endereço  
/var/www/pedro/docs/file.html
```

é:

- A) ServerAlias /var/www/~docs
- B) VirtualHost public\_html
- C) UserDir /var/www/\*/docs
- D) ServerName /var/www/~pedro/docs
- E) DocumentRoot docs

5. (FGV/Prefeitura de Niterói-RJ/2018) O arquivo httpd.conf é o arquivo de configuração principal do servidor Web Apache. Ele contém diretivas que controlam o funcionamento do servidor.

Assinale a opção que indica a diretiva que pode ser usada para especificar endereço e portas alternativas para o servidor web receber requisições externas.

- A) ServerRoot
- B) Listen
- C) AcceptFilter
- D) Redirect
- E) SetInputFilter



6. (FGV/Banestes/2021) O servidor HTTP Apache versão 2.4 é configurado por meio de diretivas presentes no arquivo de configuração denominado httpd.conf.

A diretiva que define o diretório base onde as páginas web devem ser armazenadas para disponibilização pelo servidor Apache é:

- A) DocumentRoot;
- B) ServerRoot;
- C) ServerPath;
- D) ServerName;
- E) ChRootDir.

7. (FGV/TJ-TO/2022) A diretiva que pode ser utilizada no arquivo de configuração httpd.conf do servidor HTTP Apache versão 2.4 para definir o número limite de requisições simultâneas que serão atendidas pelo servidor é:

- A) MaxRequestWorkers;
- B) MaxOpenConnections;
- C) MaxKeepAliveRequests;
- D) StreamMaxMemSize;
- E) HeartbeatMaxServers.

## GABARITO

01	02	03	04	05	06
A	A	E	C	B	A
07					
A					



## LISTA DE QUESTÕES – FCC

1. (FCC/Prefeitura de Teresina-PI/2016) Uma das formas de se iniciar o servidor Apache é por meio do comando
  - A) apache inic
  - B) apachectl start
  - C) apachectl run
  - D) apachectl go
  - E) apache send
  
2. (FCC/DPE-RS/2017) Considere que o servidor Apache 2 foi instalado em um computador servidor com sistema operacional CentOS. Para configurar o acesso às páginas desse servidor Apache deve-se editar o arquivo:
  - A) /etc/apache/init.conf
  - B) /bin/apache/httpd.init
  - C) /etc/conf/httpd.init
  - D) /lib/apache/init.conf
  - E) /etc/httpd/conf/httpd.conf
  
3. (FCC/SEMEF Manaus-AM/2019) Um programador deseja reiniciar o Servidor HTTP Apache Versão 2.4 de forma que os visitantes ativos do site possam concluir os downloads em andamento antes de o servidor ser reiniciado. Para isso deve usar o comando
  - A) apachectl -k graceful
  - B) apachectl -j restart
  - C) restartserver -j graceful
  - D) start -s graceful-stop
  - E) runserver -j restart
  
4. (FCC/TRT5/2022) Considere o programa da configuração de host virtual do Apache HTTP server, que será executado em condições ideais.  
< | >  
DocumentRoot "/var/www/trt5.infra/"  
ServerName trt5.infra  
CustomLog /var/log/httpd/trt5\_access.log combined



```
ErrorLog /var/log/httpd/trt5.infra_error.log  
</VirtualHost>
```

Para que o servidor responda as requisições, somente na porta 80, a partir de qualquer endereço IP configurado, a lacuna I deve ser corretamente preenchida com

- A) VirtualHost any eq 80
- B) VirtualHost address\_Family=any; port=80
- C) VirtualHost \*:80
- D) VirtualHost any\_IP:80
- E) VirtualHost \*:\*

## GABARITO

01	02	03	04		
B	E	A	C		





## LISTA DE QUESTÕES – VUNESP

1. (VUNESP/Pref. de Registro-SP/2018) Considere o cenário em que você instalou um programa servidor de página web (Apache ou IIS) em um computador. Para testar se o servidor de página web está funcionando, utilizando um navegador web nesse computador, deve-se inserir na barra de endereços do navegador o endereço: `http://`

- A) 10.0.0.1.
- B) 127.0.0.1.
- C) 169.254.0.1.
- D) 172.16.0.1.
- E) 192.168.0.1.

2. (VUNESP/TJM-SP/2021) No arquivo de configuração `httpd.conf` de um servidor web Apache 2.4, foi encontrada a seguinte linha:

`ServerSignature On`

Essa configuração significa que

- A) será utilizada assinatura digital, que requer a instalação de um certificado, em documentos enviados do servidor web para os requisitantes.
- B) o protocolo HTTPS está habilitado, o que requer a instalação de um certificado digital, e o tráfego ocorrerá de forma segura entre os clientes e o servidor.
- C) uma listagem de conteúdo de arquivos será gerada e enviada automaticamente em respostas a requisições a diretórios que não possuam um arquivo de índice.
- D) será exigido um usuário e senha para acessar determinados diretórios hospedados no servidor, definidos em arquivos indicados pela diretiva `AuthUserFile` na configuração de cada diretório protegido.
- E) em documentos gerados pelo próprio servidor web, como páginas de erro, será adicionada uma linha de rodapé com informações do servidor.

3. (VUNESP/UNICAMP/2022) No Apache HTTP Server, o módulo `mod_headers` permite que

- A) a URL da requisição seja modificada.
- B) as requisições de entrada sejam balanceadas entre múltiplos servidores HTTP.
- C) o servidor suporte SSL/TLS.
- D) o método HEAD do protocolo HTTP seja suportado.
- E) os cabeçalhos das requisições sejam customizados.



## GABARITO

01	02	03			
B	E	E			



## LISTA DE QUESTÕES – MULTIBANCAS

1. (FIOCRUZ/FIOCRUZ/2016) São exemplos de comandos para reiniciar o Apache nas versões linux RedHat e Ubuntu, respectivamente:
  - A) service apache restart ou /etc/init.d/http restart.
  - B) service httpd restart ou /etc/init.d/apache2 restart.
  - C) /etc/init.d/apache2 reload ou /etc/init.d/apache2 restart.
  - D) /etc/init.d/httpd reload ou /etc/init.d/apache2 restart.
  - E) service apache reload ou /etc/init.d/http reload.
  
2. (CCV-UFC/UFC/2016) Qual das diretivas abaixo deve ser configurada no arquivo de configuração do servidor Apache para informar se serão aceitas ou não conexões HTTP persistentes?
  - A) Mutex.
  - B) Timeout.
  - C) KeepAlive.
  - D) CacheEnable.
  - E) HostnameLookups.
  
3. (CS-UFG/CELG-GT-GO/2017) Na configuração do Apache HTTP Server (httpd), o uso da diretiva <virtualHost> indica que o servidor Web irá
  - A) executar em uma máquina virtual.
  - B) executar em um servidor virtual em ambiente de nuvem
  - C) rodar mais de um website em uma mesma máquina.
  - D) ser replicado em várias máquinas, embora aparente ser um único host.
  
4. (SUGEP-UFRPE/UFRPE/2018) Numa instalação com o Servidor Apache existe a necessidade de alterar o arquivo de configuração. O arquivo de configuração padrão do Apache é o arquivo:
  - A) httpd.conf
  - B) apch.ini
  - C) http.confg
  - D) apache.ini



E) apch.conf

5. (Quadrix/CRM-PR/2018) Um servidor Apache pode hospedar muitos sites web diferentes, simultaneamente, com o uso do método chamado de Virtual hosting.

6. (AOCP/UFPB/2019) Atualmente, o servidor web Apache é um dos mais utilizados na internet, pois a sua instalação e configuração são muito simples. Sobre o servidor web Apache, assinale a alternativa correta.

A) Por padrão, o diretório raiz do servidor web é /var/www (no Debian) e /var/www/html (no Fedora). O diretório raiz pode ser alterado através da opção DocumentRoot, que fica dentro do arquivo principal de configuração do Apache.

B) No servidor web Apache, somente pode ser hospedado um site por vez, não havendo forma de hospedar mais de um site por instalação.

C) O servidor web Apache não é modular, ou seja, se for acessada uma página PHP, o próprio Apache faz a interpretação do código, sem requisitar nada ao módulo PHP.

D) Para se verificar se o servidor web Apache foi instalado corretamente, deverá se acessar, por meio de um navegador, o site apache.org. Ele retornará o status do servidor web.

E) O servidor web Apache é um sistema de código fonte fechado e pago.

7. (FEPESE/CELESC/2019) Qual o nome do principal arquivo de configuração do servidor Apache?

A) httpd.conf

B) daemon.sh

C) apache.conf

D) apache.sh

E) conf.sh

8. (FEPESE/Prefeitura de Florianópolis-SC/2019) Analise as afirmativas abaixo sobre os logs dos servidores web Apache.

1. A apache possui ao menos dois logs denominados access\_log e error\_log.

2. Os logs do apache estão localizados por padrão em /var/log/apache2 em distribuições Linux Ubuntu.

3. Os logs de acesso do Apache não são passíveis de customização; isto é, não é possível acrescentar campos ou informações adicionais aos logs de acessos.

Assinale a alternativa que indica todas as afirmativas corretas.

A) É correta apenas a afirmativa 1.



- B) São corretas apenas as afirmativas 1 e 2.
- C) São corretas apenas as afirmativas 1 e 3.
- D) São corretas apenas as afirmativas 2 e 3.
- E) São corretas as afirmativas 1, 2 e 3.

9. (UniFil/Prefeitura de M. C. Rondon-PR/2021) Criado em 1995 por Rob McCool, o servidor Apache ou Servidor HTTP Apache é o mais bem sucedido servidor web livre que existe. Trata-se de um servidor web muito popular, utilizado principalmente no Linux. Assim como qualquer servidor do tipo, o Apache é responsável por disponibilizar páginas e todos os recursos que podem ser acessados pelo internauta. Envio de e-mails, mensagens, compras online e diversas outras funções podem ser executadas graças a servidores como o Apache.

Assinale a alternativa que não faz parte das vantagens de um Servidor Apache.

- A) O Servidor Apache tem uma grande estabilidade no seu funcionamento, visto que o mesmo tem um baixo nível de bugs
- B) O Servidor Apache tem grande compatibilidade com sistemas e plataformas
- C) O Servidor Apache possui código aberto
- D) O Servidor Apache é de uso gratuito

10.(Quadrix/CRT-04/2022) Quanto às tecnologias, às ferramentas, aos aplicativos e aos procedimentos associados à Internet, julgue o item.

O Apache é um exemplo de servidor web.

11.(CONSULPLAN/MPE-PA/2022) Considerando que Apache é um servidor web de código aberto utilizado para hospedagem de sites, analise as afirmativas a seguir.

- I. Portas-padrão do Apache: HTTP na porta 80 e HTTPS na porta 443 (Certificado SSL).
- II. A diretiva "alias" é responsável por mapear recursos no sistema de arquivos.
- III. "Location" refere-se à localização do sistema de arquivos do servidor. "Directory" é usado para definir a URL.
- IV. O arquivo de log de erro do Apache está em /var/log/httpd/access\_log.
- V. Existem dois tipos de hospedagem virtual: baseado no nome e baseado no IP.

Está correto o que se afirma apenas em

- A) III.
- B) III e IV.
- C) I, II e V.
- D) II, III e IV.



12.(FEPESE/Pref. de Bal. Camboriú-SC/2023) Qual o nome e local correto do arquivo de configuração do Apache em distribuições Red Hat Enterprise Linux 9.xx?

- A) /etc/apache.conf
- B) /etc/httpd/admin.conf
- C) /etc/httpd/conf/main.conf
- D) /etc/httpd/conf/httpd.conf
- E) /etc/httpd/apache.conf

## GABARITO

01	02	03	04	05	06
B	C	C	A	C	A
07	08	09	10	11	12
A	B	A	C	C	D



## IIS (INTERNET INFORMATION SERVICES)

O IIS (*Internet Information Services*) é um servidor Web proprietário (e pago) criado pela Microsoft para seus sistemas operacionais para servidores. A função do IIS no Windows Server é oferecer uma plataforma para a **hospedagem de sites, serviços e aplicativos**, permitindo a integração das seguintes tecnologias: ASP.NET, FTP, PHP, WCF (Windows Communication Foundation - estrutura usada para a criação de aplicativos orientados a serviços) e o próprio IIS.

Uma de suas funcionalidades mais utilizadas é a geração de páginas HTML dinâmicas, através da tecnologia proprietária ASP (*Active Server Pages*), mas também pode utilizar outras tecnologias com adição de módulos de terceiros. Depois do lançamento da plataforma .NET (em 2002), o IIS ganhou também a função de gerenciar o ASP.NET. Este é formado basicamente por dois tipos de aplicações:

- Páginas Web: tradicionais, acessadas por usuários (extensão ASPX);
- Web Services: funções disponibilizadas pela rede, chamada por aplicativos ASMX.

O ASP.NET é compilado antes da execução, trazendo vantagens no desempenho em relação às opções interpretadas, como o ASP e o PHP.

Algumas características do IIS são:

- Maximiza a segurança da Web através de um consumo de servidor reduzido e do isolamento automático de aplicativo;
- Implanta e executa o ASP.NET, o ASP clássico e os aplicativos Web do PHP no mesmo servidor;
- Faz o isolamento de aplicativo concedendo aos processos de trabalho, por padrão, uma identidade exclusiva e uma configuração de área restrita (maior segurança);
- Adiciona e remove os componentes internos do IIS, e até mesmo os substitui por módulos personalizados;
- Agiliza o site através de um cache dinâmico interno e de uma compactação avançada;
- Usa o Gerenciador do IIS para configurar recursos do IIS e administrar *sites*;
- Usa o protocolo FTP (*File Transfer Protocol*) para permitir que proprietários de site carreguem e baixem arquivos;
- Usa o Windows PowerShell para automatizar o gerenciamento da maioria das tarefas de administração do servidor Web;
- Configura vários servidores Web em um *farm* de servidores que podem ser gerenciados usando o IIS;
- O diretório base do *site* padrão é "UNIDADE:\inetpub\wwwroot", geralmente "C:\inetpub\wwwroot";
- O diretório base para o FTP é "UNIDADE:\inetpub\ftproot", geralmente "C:\inetpub\ftproot" (note que só muda o "www" por "ftp");
- Uma ferramenta usada para provisionar sites e executar vários comandos para editar a configuração é a [AppCmd.exe](#).



## Pool de aplicativos

Um *pool* de aplicativos define um grupo de um ou mais processos de trabalho, configurado com definições comuns que atendam uma ou mais aplicações atribuídas a este *pool*. Cada *pool* de aplicativos utiliza 1 ou 2 modos de integração .NET (**modo integrado e modo clássico**) para executar aplicações ASP.NET. O modo definido para o *pool* de aplicativos define como será processada qualquer requisição que chegar a esse *pool*.

**Modo integrado:** permite que o IIS processe requisições no *pool* de aplicativos utilizando o "Integrated Pipeline", o que permite que os módulos do ASP.NET participem do processamento das requisições.

**Modo clássico:** utiliza o *pipeline* de processamento do IIS 6. Inicialmente as requisições são processadas através dos módulos do IIS7, as requisições do ASP.NET são transportadas para os "ISAPI Filter – aspnet\_isapi.dll", o *pipeline* de processamento do ASP.NET é separado do *pipeline* de processamento do IIS7. Ou seja, o fluxo de processamento é muito mais lento do que o modo integrado.

Algo importante de levar para sua prova é que se duas aplicações Web diferentes estiverem em um mesmo *pool*, falhas graves em uma aplicação podem afetar a outra. Daria para entrar em detalhes em alguns itens abordados acima, mas para fins de concurso não vale a pena. Vamos focar no que as bancas costumam pedir!

## Segurança

No Windows Server, durante a instalação do IIS, há a opção de segurança "**Filtragem de solicitações**", que serve para analisar as requisições feitas ao servidor Web e impedir alguns ataques de manipulação de URL. Por padrão, a filtragem de solicitações no IIS 7.0 permite um comprimento máximo de URL de 4096 caracteres e de cadeia de caracteres de consulta um máximo de 2048 caracteres.

A "Filtragem de solicitações" verifica todas as requisições recebidas no servidor e as filtra com base nas regras definidas pelo administrador. Muitos ataques maliciosos compartilham características em comum, como URLs muito longas ou solicitações de uma ação rara. Ao filtrar as solicitações, há uma tentativa de reduzir o impacto desses tipos de ataques.

## Instalação de componentes do IIS

O IIS e seus componentes são expostos pelo Windows como recursos opcionais. Desta forma, os usuários habilitam apenas os recursos necessários para que seus *sites* funcionem bem. Como efeito colateral, ao configurar o IIS por meio da API (*Application Programming Interface*), é possível que um recurso (ex.: Autenticação do Windows) ainda não esteja instalado no computador. Historicamente, a habilitação de recursos do IIS foi feita por meio de DISM.exe (Gerenciamento e Manutenção de Imagens de Implantação, ferramenta de linha de comando que pode ser usada para fazer a manutenção e preparar imagens do Windows), comandos do PowerShell ou da interface do usuário do recurso opcional do Windows.

A **API de Administração** do IIS expõe um **método simples para instalar/desinstalar esses recursos** para que os consumidores não necessitem alterar ambientes para habilitar os recursos dos quais





dependem. A API é um microsserviço executado na máquina de destino e é a única etapa necessária para disponibilizar o gerenciamento remoto à máquina.

## Verificando se um recurso está instalado

A primeira etapa do gerenciamento de um recurso é verificar se ele está instalado. Para qualquer recurso do IIS, pode-se determinar isso enviando uma solicitação para o seu ponto de extremidade de API. Se o ponto de extremidade retornar uma resposta "200 OK", o recurso está instalado. Se o recurso não estiver instalado, a API retornará uma resposta "404 Não Encontrada" com um objeto de erro JSON "não instalado" no corpo. Abaixo podemos ver um exemplo que usa o recurso de documento padrão do IIS. Vamos verificar se o documento padrão está habilitado através do envio de uma solicitação GET para o ponto de extremidade do documento padrão. Usamos o escopo na cadeia de caracteres de consulta com um valor vazio para especificar que estamos direcionando o escopo do servidor Web.

```
GET /api/webserver/default-documents?scope=
```

O recurso não está instalado:

```
404 Not Found
{
  "title": "Not found",
  "detail": "IIS feature not installed",
  "name": "Default Document",
  "status": "404"
}
```

O recurso está instalado:

```
200 OK
{
  "id": "{id}",
  "enabled": "true",
  "scope": "",
  "metadata": {
    "is_local": "true",
    "is_locked": "false",
    "override_mode": "allow",
    "override_mode_effective": "allow"
  },
  "website": null
}
```

## Instalando um recurso

A instalação do recurso é executada emitindo uma solicitação POST para o ponto de extremidade do recurso. Alguns recursos do IIS, como o Repositório Central de Certificados, exigem que as configurações iniciais sejam fornecidas durante a instalação. Por exemplo, suponha que o recurso de documento padrão estivesse retornando a resposta "404 Recurso Não



Instalado". Então, para enviar uma solicitação POST para o ponto de extremidade de documentos padrão instala o recurso e retorna as configurações de recursos, temos:

POST/api/webserver/default-documents

Resposta HTTP:

```
201 CREATED
Location: /api/webserver/default-documents/{id}
{
  "id": "{id}",
  "enabled": "true",
  "scope": "",
  "metadata": {
    "is_local": "true",
    "is_locked": "false",
    "override_mode": "allow",
    "override_mode_effective": "allow"
  },
  "website": null
}
```

## Desinstalando um recurso

Os recursos do IIS podem ser desinstalados emitindo uma solicitação DELETE para o recurso no nível do servidor Web. Para recursos do IIS que dão suporte à configuração nos níveis de *site* e aplicativo, é possível verificar se o objeto representa o escopo do servidor Web, garantindo que o campo de escopo esteja vazio.

Primeiro, deve-se obter o URI do recurso para desinstalar:

GET/api/webserver

```
{
  "id": "{id}",
  "_links": {
    ... // outros...
    "default_document": {
      "href":
"/api/webserver/default-documents/{def-doc-id}"
    }
  }
}
```

Depois, deve-se emitir uma solicitação DELETE para o ponto de extremidade de recursos:

DELETE/api/webserver/default-documents/{def-doc-id}



## Iniciando, parando e reiniciando o IIS

O comando `iisreset` pode ser utilizado para iniciar, parar ou reiniciar o IIS. Vejamos alguns parâmetros utilizados e como o IIS reage:

```
iisreset /stop
```

```
Attempting stop...  
Internet services successfully stopped
```

```
iisreset /start
```

```
Attempting start...  
Internet services successfully started
```

```
iisreset /restart
```

```
Attempting stop...  
Internet services successfully stopped  
Attempting start...  
Internet services successfully restarted
```

Para buscar ajuda (ver as opções):

```
iisreset /?
```

## Módulos

Em vez de manter a maioria das funcionalidades dentro do próprio servidor, o IIS inclui um motor de servidor Web no qual componentes (módulos) podem ser adicionados ou removidos. Os módulos são características individuais que o servidor utiliza para processar os pedidos. Por exemplo, módulos de autenticação são utilizados para autenticar as credenciais do cliente.

### Módulos nativos

A seguir veremos os módulos nativos que estão disponíveis em uma instalação completa do IIS 7 ou superior, lembrando que podem ser removidos ou substituídos por módulos personalizados, dependendo das necessidades.

#### Módulos HTTP

Módulos HTTP incluem módulos para responder a informações e dúvidas enviadas em cabeçalhos dos clientes, para retornar erros de HTTP, para redirecionar solicitações etc. Abaixo vemos a lista dos módulos HTTP com a descrição de cada um.

- **CustomErrorModule**: envia padrão e configura mensagens de erro HTTP quando um código de status de erro é definido em uma resposta;
- **HttpRedirectionModule**: suporta redirecionamento configurável para solicitações HTTP;



- **ProtocolSupportModule**: executa protocolo de ações relacionadas, tais como definir cabeçalhos de resposta e redirecionar cabeçalhos com base na configuração;
- **RequestFilteringModule**: adicionado ao IIS 7.5. Pedidos de filtros conforme configurado, para controlar o comportamento de protocolo e de conteúdo;
- **WebDAVModule**: adicionado ao IIS 7.5. Permite publicação mais segura de conteúdo através de HTTP sobre SSL.

## Módulos de segurança

Vários módulos no IIS executam tarefas relacionadas com a segurança no *pipeline* de processamento de solicitação. Além disso, existem módulos separados para cada um dos esquemas de autenticação, que permitem a seleção dos módulos para os tipos de autenticação necessárias, entre outras atividades. Abaixo vemos a lista dos módulos de segurança com a descrição de cada um.

- **CustomErrorModule**: envia padrão e configura mensagens de erro HTTP quando um código de status de erro é definido em uma resposta;
- **AnonymousAuthenticationModule**: executa a autenticação anônima;
- **BasicAuthenticationModule**: executa a autenticação básica;
- **CertificateMappingAuthenticationModule**: realiza a autenticação de certificado de mapeamento usando o Active Directory;
- **DigestAuthenticationModule**: realiza a autenticação Digest;
- **IISCertificateMappingAuthenticationModule**: realiza a autenticação de certificado de mapeamento usando a configuração do certificado IIS;
- **RequestFilteringModule**: executa tarefas URLScan, tais como: configurar verbos permitidos e extensões de nome de arquivo, o estabelecimento de limites, e digitalização para sequências de caracteres ruins;
- **UrlAuthorizationModule**: executa a autorização de URL;
- **WindowsAuthenticationModule**: executa autenticação NTLM integrada;
- **IpRestrictionModule**: restringe endereços IPv4 listados na lista de segurança IP na configuração.



## Módulos de conteúdo

Vários módulos no IIS executam tarefas relacionadas com o conteúdo do *pipeline* de processamento de solicitação, o que inclui módulos para processar solicitações de arquivos estáticos, para retornar uma página padrão quando um cliente não especificar um recurso em um pedido, entre outros. Abaixo vemos a lista dos módulos de conteúdo com a descrição de cada um.

- **CgiModule**: executa *Common Gateway Interface* (CGI) para construir processos de saída de resposta;
- **DefaultDocumentModule**: tentativas de retornar um documento padrão para as solicitações feitas ao diretório pai;
- **DirectoryListingModule**: lista o conteúdo de um diretório;
- **IsapiModule**: hospeda ISAPI DLLs de extensão;
- **IsapiFilterModule**: suporta filtro ISAPI DLLs;
- **ServerSideIncludeModule**: processos do lado do servidor;
- **StaticFileModule**: serve arquivos estáticos;
- **FastCgiModule**: suporta FastCGI, que oferece uma alternativa de alto desempenho para CGI.

## Módulos de compressão

Há dois módulos de compressão no IIS, a saber:

- **DynamicCompressionModule**: comprime respostas com Gzip;
- **StaticCompressionModule**: realiza pré-compressão de conteúdo estático.

## Módulos de cache

Diversos módulos executam tarefas relacionadas ao armazenamento em cache no *pipeline* de processamento de solicitação. Cache melhora o desempenho dos *sites* e aplicações Web através do armazenamento de informações processadas, como páginas da Web, na memória do servidor, e depois utilizam tal informação em visitas posteriores para o mesmo recurso. Abaixo vemos a lista dos módulos de cache com a descrição de cada um.

- **FileCacheModule**: fornece modo de usuário de cache para arquivos e identificadores de arquivo;



- **HTTPCacheModule**: fornece modo kernel e modo de usuário em cache HTTP.sys<sup>1</sup>;
- **TokenCacheModule**: módulo de cache de token utilizado para autenticação;
- **UriCacheModule**: relacionado ao armazenamento em cache de informações sobre recursos (arquivos, scripts, imagens etc.) associados a URIs (*Uniform Resource Identifiers*).

## Logging e módulos de diagnóstico

Diversos módulos executam tarefas relacionadas a diagnósticos no pipeline de processamento de solicitação. Os módulos de registro (log) suportam o carregamento de módulos personalizados e passam informações para o HTTP.sys. Abaixo vemos a lista dos módulos de diagnóstico com a descrição de cada um.

- **CustomLoggingModule**: carrega módulos personalizados;
- **FailedRequestsTracingModule**: suporta o recurso de rastreamento de falha de solicitação;
- **HttpLoggingModule**: passa informações e status de processamento para o HTTP.sys para registro;
- **RequestMonitorModule**: pedidos de faixas em execução em processos de trabalho e informações de relatórios com status de tempo de execução e Interface de Controle de Programação de Aplicativos (RSCA);
- **TracingModule**: relatórios de eventos para Rastreamento de Eventos para Windows (ETW).

## Módulos de apoio

Há dois módulos de apoio:

- **ManagedEngine**: fornece integração de módulos de código gerenciado no pipeline de processamento de solicitação;
- **ConfigurationValidationModule**: realiza a validação de problemas de configuração.

## Módulos gerenciados

Além de módulos nativos, o IIS permite o uso de módulos de código gerenciado para estender a funcionalidade. Os módulos gerenciados **dependem do módulo ManagedEngine**. A seguir veremos uma lista dos módulos gerenciados disponíveis com uma instalação completa do IIS 7 ou superior.

- **AnonymousIdentification**: gerencia identificadores anônimos, usados por recursos que suportam a identificação anônima, tais como o perfil do ASP.NET;

---

<sup>1</sup> O HTTP.sys atua como um driver de modo kernel e fornece funcionalidades essenciais para o processamento de solicitações HTTP. É utilizado pelo Internet Information Services (IIS) como a camada subjacente responsável pelo processamento de solicitações HTTP antes de serem encaminhadas para a aplicação ou serviço correspondente no espaço de usuário.



- **DefaultAuthentication**: garante que um objeto de autenticação está presente no contexto;
- **FileAuthorization**: verifica se um usuário tem permissão para acessar o arquivo solicitado;
- **FormsAuthentication**: suporta a autenticação de formulários;
- **OutputCache**: suporta cache de saída;
- **Profile**: gerencia perfis de usuário usando o perfil ASP.NET, que armazena e recupera as configurações do usuário em uma fonte de dados, como um banco de dados;
- **RoleManager**: gerencia uma instância RoleMain para o usuário atual;
- **SessionState**: defende a manutenção do estado de sessão, o que permite o armazenamento de dados específicos para um único cliente dentro de uma aplicação no servidor;
- **UrlAuthorization**: determina se para o usuário atual é permitido o acesso à URL solicitada, com base no nome de usuário ou a lista de funções das quais o usuário é membro;
- **UrlMappingsModule**: suporta o mapeamento de uma URL real para uma URL "mais amigável";
- **WindowsAuthentication**: define a identidade do usuário para uma aplicação ASP.NET quando a autenticação do Windows está ativada.

## Testando o Servidor IIS

Após a instalação e configuração do servidor Web IIS, uma forma de testar é abrir um navegador no próprio servidor e colocar o endereço IP "127.0.0.1" (na verdade pode ser outras combinações de 127.x.x.x, mas o padrão é 127.0.0.1), ou "localhost". Então, na barra de endereços do navegador fica <http://127.0.0.1/>, ou <http://localhost/>.

Se o teste for realizado a partir de outro computador, deve-se colocar o endereço IP do servidor no lugar de "localhost" ou "127.0.0.1". Se o DNS estiver configurado, pode-se testar com o nome de domínio (ex.: <http://www.estrategiaconcursos.com.br>).



## QUESTÕES COMENTADAS – CEBRASPE

1. (CEBRASPE/MEC/2011) A partir da instalação do IIS é disponibilizado um sítio-padrão cujas pastas estão instaladas no servidor, no caminho físico c:\inetpub\wwwroot.

Comentários:

O diretório base do site padrão é "UNIDADE:\inetpub\wwwroot", geralmente "C:\inetpub\wwwroot".

Gabarito: Certo

2. (CEBRASPE/MEC/2015) Julgue o próximo item, relativo ao IIS (Internet Information Server) 7.0.

O IIS 7.0 é composto por módulos que podem ser adicionados e removidos do servidor web e que permitem a customização de seu servidor, com adição e remoção de características específicas.

Comentários:

Em vez de manter a maioria das funcionalidades dentro do próprio servidor, o IIS inclui um motor de servidor Web no qual componentes (módulos) podem ser adicionados ou removidos. Os módulos são características individuais que o servidor utiliza para processar os pedidos. Por exemplo, módulos de autenticação são utilizados para autenticar as credenciais do cliente.

Gabarito: Certo

3. (CEBRASPE/PG-DF/2021) A respeito dos serviços de Internet e de sua instalação na plataforma Windows, julgue o item seguinte.

Caso um servidor Web IIS (Internet Information Services) seja instalado e configurado para servir conteúdo estático, este tipo de conteúdo será entregue ao usuário como uma página HTML exatamente como armazenada.

Comentários:

Conteúdo estático, como o nome já deixa claro, não há alteração nenhuma. Ou seja, uma página é mostrada exatamente como está gravada em disco (texto, imagens etc.).

Ao contrário das páginas estáticas existe a geração de páginas HTML dinâmicas, através da tecnologia proprietária ASP (*Active Server Pages*), mas também pode utilizar outras tecnologias com adição de módulos de terceiros.

Gabarito: Certo

4. (CEBRASPE/APEX Brasil/2022) Em relação à API de administração do IIS (Internet Information Services) da Microsoft, assinale a opção correta.

A) A configuração do IIS por meio da API de administração é feita utilizando-se o aplicativo cmd.exe.

B) A API de administração serve para auxiliar no processo de autenticação do usuário, fornecendo o serviço de Kerberos.





C) A API é um microserviço executado na máquina de destino e é a única etapa necessária para disponibilizar o gerenciamento remoto à máquina.

D) A API de administração do IIS baseia-se nos protocolos SOAP para fornecer uma interface que utiliza XML e WSDL para troca de mensagens e comandos.

#### Comentários:

A) ERRADA - Uma ferramenta usada para provisionar sites e executar vários comandos para editar a configuração é a AppCmd.exe.

B) ERRADA - O Kerberos é um protocolo de autenticação utilizado em ambientes Windows para autenticar usuários e serviços em uma rede. Ele não é uma funcionalidade diretamente associada à API de administração do IIS.

C) CORRETA - A API de Administração do IIS expõe um método simples para instalar/desinstalar esses recursos para que os consumidores não necessitem alterar ambientes para habilitar os recursos dos quais dependem. A API é um microserviço executado na máquina de destino e é a única etapa necessária para disponibilizar o gerenciamento remoto à máquina.

D) ERRADA - A API Microsoft.Web.Administration é uma API baseada em .NET que permite a manipulação programática das configurações do IIS.

Gabarito: C



## QUESTÕES COMENTADAS – FGV

1. (FGV/TJ-SC/2015) Em um servidor de aplicação Internet Information Services (IIS) são executadas três aplicações web diferentes (1), (2) e (3). As aplicações (1) e (2) foram colocadas em um mesmo application pool (A) e a aplicação (3) foi colocada em outro application pool (B). Está correto afirmar que:
- A) permissões de acesso requisitadas para (2) aplicam-se a (3);
  - B) falhas severas na execução de (1) podem afetar (2);
  - C) requisições feitas para (1) também podem ser tratadas por (2);
  - D) (1) e (3) podem usar um mesmo arquivo de log independentemente, sem precisar de um mutex;
  - E) (1) não pode ser interrompido sem que (2) também o seja.

### Comentários:

Um pool de aplicativos define um grupo de um ou mais processos de trabalho, configurado com definições comuns que atendam uma ou mais aplicações atribuídas a este pool. Cada pool de aplicativos utiliza 1 ou 2 modos de integração .NET (modo integrado e modo clássico) para executar aplicações ASP.NET. O modo definido para o pool de aplicativos define como será processada qualquer requisição que chegar a esse pool. Importante destacar que se duas aplicações Web diferentes estiverem em um mesmo pool, falhas graves em uma aplicação podem afetar a outra.

Gabarito: B

2. (FGV/TJ-RO/2021) No Windows Server, o Gerenciador dos Serviços de Informação da Internet (IIS) é reiniciado pela linha de comando:
- A) infostart
  - B) iisreset
  - C) iisrestart
  - D) restart p-iis
  - E) run iis

### Comentários:

O comando **iisreset** pode ser utilizado para iniciar, parar ou reiniciar o IIS. Vejamos alguns parâmetros utilizados e como o IIS reage:

```
iisreset /stop
```

```
Attempting stop...  
Internet services successfully stopped
```



```
iisreset /start
```

```
Attempting start...  
Internet services successfully started
```

```
iisreset /restart
```

```
Attempting stop...  
Internet services successfully stopped  
Attempting start...  
Internet services successfully restarted
```

Gabarito: B

3. (FGV/TJ-TO/2022) O módulo nativo que pode ser adicionado à seção de configuração <globalModules> do Internet Information Services (IIS), versão 7, para habilitar um conjunto de regras de segurança com a finalidade de rejeitar solicitações suspeitas é:

- A) DirectoryListingModule;
- B) FailedRequestsTracingModule;
- C) ProtocolSupportModule;
- D) RequestFilteringModule;
- E) RequestMonitorModule.

#### Comentários:

RequestFilteringModule: adicionado ao IIS 7.5. Pedidos de filtros conforme configurado, para controlar o comportamento de protocolo e de conteúdo.

Gabarito: D



## QUESTÕES COMENTADAS – FCC

1. (FCC/TRT19/2011) O serviço que faz do Windows 2003 um Servidor Web é o IIS (Internet Information Services). Quando o IIS é instalado da maneira padrão, é disponibilizado um site com uma única página chamada iisstart.htm. Essa página pode ser encontrada no caminho físico

- A) c:\inetpub\wwwroot
- B) c:\root\
- C) c:\root\www
- D) c:\http\www
- E) c:\net\web

### Comentários:

As bancas gostam dessa...repetindo:

O diretório base do site padrão é "UNIDADE:\inetpub\wwwroot", geralmente "C:\inetpub\wwwroot".

Gabarito: A

2. (FCC/TRE-SP/2017) Hipoteticamente, o Técnico, responsável pela administração do servidor com Windows Server 2012 do TRE-SP, realizou a instalação do serviço IIS com a configuração padrão de fornecimento. Considerando-se que ele não atribuiu o endereço IP e desligou o servidor da rede de computadores para evitar acesso externo, para que esse profissional realize o teste local do servidor IIS, ele deve utilizar um navegador e acessar, na Barra de endereços do navegador, a URL:

- A) http://127.0.0.1
- B) ftp://10.0.0.1
- C) https://192.168.0.1
- D) http://192.168.0.1
- E) https://255.255.255.0

### Comentários:

Na verdade, essa não é uma questão exclusiva para o IIS, pode ser para o Apache ou outro servidor Web! Para testar o servidor Web que está instalado na própria máquina é só fazer referência ao protocolo HTTP e o endereço do localhost (127.0.0.1), ou seja: "http://127.0.0.1". Se você colocar agora em seu navegador não deve aparecer uma mensagem como "Não é possível acessar esse site", a não ser que você tenha instalado um servidor Web (IIS, Apache ou outro)!

Gabarito: A



## QUESTÕES COMENTADAS – VUNESP

1. (VUNESP/Pref. de P. Prud.-SP/2016) Por padrão, o servidor web IIS 7.5 armazena o conteúdo do site web no diretório:

- A) %SystemDrive%\var\wwwroot
- B) %SystemDrive%\inetpub\www
- C) %SystemDrive%\inetpub\DocumentRoot
- D) %SystemDrive%\opt\www
- E) %SystemDrive%\inetpub\wwwroot

### Comentários:

O diretório base do site padrão é "UNIDADE:\inetpub\wwwroot", geralmente "C:\inetpub\wwwroot".

O diretório base para o FTP é "UNIDADE:\inetpub\ftproot", geralmente "C:\inetpub\ftproot" (note que só muda o "www" por "ftp").

Gabarito: E

2. (VUNESP/Pref. de Registro-SP/2018) Considere o cenário em que você instalou um programa servidor de página web (Apache ou IIS) em um computador. Para testar se o servidor de página web está funcionando, utilizando um navegador web nesse computador, deve-se inserir na barra de endereços do navegador o endereço: http://

- A) 10.0.0.1.
- B) 127.0.0.1.
- C) 169.254.0.1.
- D) 172.16.0.1.
- E) 192.168.0.1.

### Comentários:

Após a instalação e configuração do servidor Web IIS, uma forma de testar é abrir um navegador no próprio servidor e colocar o endereço IP "127.0.0.1" (na verdade pode ser outras combinações de 127.x.x.x, mas o padrão é 127.0.0.1), ou "localhost". Então, na barra de endereços do navegador fica <http://127.0.0.1/>, ou <http://localhost/>.

Se o teste for realizado a partir de outro computador, deve-se colocar o endereço IP do servidor no lugar de "localhost" ou "127.0.0.1". Se o DNS estiver configurado, pode-se testar com o nome de domínio (ex.: <http://www.estrategiaconcursos.com.br>).

Gabarito: B



## QUESTÕES COMENTADAS – MULTIBANCAS

1. (Quadrix/SERPRO/2014) IIS e Apache são servidores cuja finalidade é:
- A) prover serviços de impressão para o Windows e o Linux, respectivamente.
  - B) prover serviços WEB.
  - C) prover serviços de WEB e e-mail, respectivamente.
  - D) prover serviços de WEB para Linux e Windows, respectivamente.
  - E) viabilizar programas para execução de páginas HTML também conhecidos como browsers.

### Comentários:

O Apache surgiu para sistema Unix-like, mas é multiplataforma (existe para Windows, por exemplo). O IIS é proprietário da Microsoft. Ambos são servidores Web.

Gabarito: B

2. (UFRJ/UFRJ/2015) Considere as seguintes afirmativas acerca do servidor Web IIS 7:
- I – Nesta versão do IIS não é possível mais instalar e configurar o serviço FTP.
  - II - No IIS 7, os pools de aplicativos podem ser executados dos seguintes modos: integrado ou clássico.
  - III – No IIS 7, a porta padrão para servir páginas em HTTPS é a 4343.

### Pode-se afirmar que:

- A) apenas II e III estão corretas.
- B) apenas I e II estão corretas.
- C) apenas II está correta.
- D) apenas I está correta.
- E) I, II e III estão corretas.

### Comentários:

(I) O FTP nunca foi desabilitado para o uso com o IIS. (II) Um pool de aplicativos define um grupo de um ou mais processos de trabalho, configurado com definições comuns que atendem uma ou mais aplicações atribuídas a este pool. Cada pool de aplicativos utiliza 1 ou 2 modos de integração .NET (modo integrado e modo clássico) para executar aplicações ASP.NET. O modo definido para o pool de aplicativos define como será processado qualquer requisição que chegar a esse pool. (III) A porta padrão para o HTTPS é 443 e a do HTTP é a 80.

Gabarito: C

3. (Colégio Pedro II/Colégio Pedro II/2016) Assinale a alternativa que NÃO apresenta uma característica do servidor de aplicação IIS.
- A) Gera páginas HTML dinâmicas.



- B) Também é um servidor de aplicativo.
- C) Executa códigos PHP, Perl, Javascript e ASP.
- D) Usa o protocolo FTP para permitir que proprietários de sites carreguem e baixem arquivos.

#### Comentários:

De tudo o que é descrito nas alternativas, sabemos que o IIS não executa códigos Perl! Lembrando: O IIS implanta e executa o ASP.NET, o ASP clássico e os aplicativos Web do PHP no mesmo servidor.

Gabarito: C

4. (IBFC/EBSERH/2016) Ao ser instalado o IIS (Internet Information Services), no disco rígido C:, por padrão os diretórios que serão criados para hospedagem de páginas e para FTP serão respectivamente:

- A) C:/iisserver/wwwroot e C:/iisserver/ftpboot
- B) C:/inetpub/wwwroot e C:/inetpub/ftpboot
- C) C:/iisserver/wwwdir e C:/iisserver/ftpdir
- D) C:/inetpub/wwwdir e C:/inetpub/ftpdir
- E) C:/inetpub/rootwww e C:/inetpub/rootftp

#### Comentários:

Para hospedagem, já vimos em algumas questões que é "C:\inetpub\wwwroot" (note que a barra está invertida nas alternativas, mas tudo bem). Para o FTP é só substituir o "www" por "ftp": "C:\inetpub\ftpboot".

Gabarito: B

5. (INAZ do Pará/DPE-PR/2017) No Windows Server 2012, durante a instalação do Servidor WEB (IIS), que opção de segurança deve ser utilizada para analisar as requisições feitas ao servidor Web e impedir os ataques manipulando URLs?

- A) Autenticação Digest, resolve o problema do firewall e de redes internas e externas.
- B) Autenticação de Mapeamento de Certificado de Cliente.
- C) Autenticação do Windows.
- D) Restrições de IP e Domínio.
- E) Filtragem de solicitações.

#### Comentários:

Segurança: No Windows Server, durante a instalação do IIS, há opção de segurança "Filtragem de solicitações", que serve para analisar as requisições feitas ao servidor Web e impedir alguns ataques de manipulação de URL. Por padrão, a filtragem de solicitações no IIS 7.0 permite um comprimento máximo de URL de 4096 caracteres e de cadeia de caracteres de consulta um máximo de 2048 caracteres.





A “Filtragem de solicitações” verifica todas as requisições recebidas no servidor e as filtra com base nas regras definidas pelo administrador. Muitos ataques maliciosos compartilham características em comum, como URLs muito longas ou solicitações de uma ação rara. Ao filtrar as solicitações, há uma tentativa de reduzir o impacto desses tipos de ataques.

Gabarito: E

6. (Quadrix/CRQ 4ª Região-SP/2018) Considerando os conceitos de servidores Web IIS e Apache, julgue o item.

O servidor web IIS é de uso exclusivo em sistemas operacionais Windows e o Apache roda apenas em sistemas que se baseiam em Unix/Linux.

Comentários:

A primeira parte está correta, pois o IIS só roda no Windows! Mas o Apache roda no Windows, Linux, MacOS, entre outros.

Gabarito: Errado

7. (Quadrix/CRQ 4ª Região-SP/2018) Considerando os conceitos de servidores Web IIS e Apache, julgue o item.

O Apache baseia-se em software livre e pode ser usado gratuitamente em servidores web. O IIS é um software proprietário e seu uso é pago.

Comentários:

O IIS (Internet Information Services) é um servidor Web proprietário (e pago) criado pela Microsoft para seus sistemas operacionais para servidores. A função do IIS no Windows Server é oferecer uma plataforma para a hospedagem de sites, serviços e aplicativos, permitindo a integração das seguintes tecnologias: ASP.NET, FTP, PHP, WCF (Windows Communication Foundation - estrutura usada para a criação de aplicativos orientados a serviços) e o próprio IIS.

Ao contrário, o Apache baseia-se em software livre e pode ser usado gratuitamente.

Gabarito: Certo

8. (Quadrix/CRQ 4ª Região-SP/2018) Considerando os conceitos de servidores Web IIS e Apache, julgue o item.

O IIS é configurado exclusivamente pela sua interface web de gerência e as configurações são guardadas internamente, não sendo possível acessá-las com editor de texto.

Comentários:

Esse “exclusivamente” acende um alerta na cabeça do candidato!

O IIS e seus componentes são expostos pelo Windows como recursos opcionais. Desta forma, os usuários habilitam apenas os recursos necessários para que seus *sites* funcionem bem. Como efeito colateral, ao configurar o IIS por meio da API (*Application Programming Interface*), é possível que um recurso (ex.: Autenticação do Windows) ainda não esteja instalado no computador.

Outra coisa: arquivos de configuração podem ser acessados via editor de texto! Mais um absurdo na questão.



Gabarito: Errado

9. (Quadrix/CRQ 4ª Região-SP/2018) Considerando os conceitos de servidores Web IIS e Apache, julgue o item.

O documento-padrão apresentado por um servidor web, tanto IIS quanto Apache, é o index.html, presente na pasta inicial, caso não seja indicado outro.

Comentários:

Por padrão, o index.html é o nome padrão que os servidores Web IIS e Apache utilizam, mas é possível alterar.

Gabarito: Certo

10.(Quadrix/CRQ 4ª Região-SP/2018) Considerando os conceitos de servidores Web IIS e Apache, julgue o item.

Apesar do crescimento do uso dos servidores IIS, ele nunca conseguiu superar a quantidade de sites servidos por seu concorrente, Apache, que ainda é o mais usado na Internet.

Comentários:

Questão de 2018, mas ainda está correta! Mas se fosse comparar Apache com Nginx, aí a resposta poderia ser diferente! O fato é que o IIS ficou para trás em comparação ao Apache (e também ao Nginx).

Gabarito: Certo

11.(FUNRIO/Câm. de S. J. de M.-RJ/2018) Um técnico de informática instalou o IIS em um computador usando os parâmetros "default". Nesse contexto, os arquivos html e de imagem relacionados a um site serão armazenados em uma pasta referenciada pelo seguinte caminho:

- A) c:\inetpub\wwwroot
- B) c:\inetpub\webroot
- C) c:\inetpub\sites
- D) c:\inetpub\index
- E) c:\inetpub\default

Comentários:

O diretório base do site padrão é "UNIDADE:\inetpub\wwwroot", geralmente "C:\inetpub\wwwroot".

O diretório base para o FTP é "UNIDADE:\inetpub\ftproot", geralmente "C:\inetpub\ftproot" (note que só muda o "www" por "ftp").

Gabarito: A

12.(IBFC/Câm. de Feira de Sant.-BA/2018) Quando comparamos atualmente os servidores IIS (Internet Information Services) e o APACHE pode-se afirmar que:

- A) mundialmente os dois rodam predominantemente em sistemas operacionais Windows
- B) o servidor IIS ao longo do tempo dominou o mercado, deixando o APACHE em 3º lugar



- C) os dois tem o propósito de servir conteúdo baseado em requisições do tipo HTTP  
D) tanto o IIS como o APACHE são servidores desenvolvidos dentro da filosofia Open Source

**Comentários:**

- A) ERRADA - o IIS roda apenas no Windows, mas o Apache não!  
B) ERRADA - o IIS é menos utilizado que o Apache, e menos que o Nginx também!  
C) CORRETA - os dois são servidores Web, então possuem o propósito de servir conteúdo baseado em requisições do tipo HTTP.  
D) ERRADA - o IIS é desenvolvido pela Microsoft.

Gabarito: C

13. (IBFC/IDAM/2019) Quanto às diferenças entre os servidores Apache e IIS, analise as afirmativas abaixo, dê valores Verdadeiro (V) ou Falso (F).

- ( ) o Apache pode rodar em várias plataformas como o Windows, Unix e Linux.  
( ) normalmente o servidor IIS utiliza a sua linguagem proprietária o ASP.  
( ) somente o Apache é capaz de responder as requisições HTTP de máquinas clientes.

Assinale a alternativa que apresenta a sequência correta de cima para baixo.

- A) V, F, F  
B) V, V, F  
C) F, V, V  
D) F, F, V

**Comentários:**

- (V) O Apache é multiplataforma, se você entrar do site <http://apache.org/>, poderá ver opções de download para Windows e Linux, por exemplo.  
(V) Não somente o ASP, que é proprietária a Microsoft, como também outras linguagens.  
(F) Claro que não! O IIS também faz isso! Além de outros servidores Web, pois esse é o conceito fundamental de um servidor Web.

Gabarito: B

14. (IBFC/EBSERH/2020) Em relação aos servidores APACHE e IIS, analise as afirmativas abaixo e dê valores Verdadeiro (V) ou Falso (F).

- ( ) ambos os servidores podem executar páginas dinâmicas.  
( ) tanto o APACHE como o IIS rodam nativamente no Linux.  
( ) os dois servidores suportam conexões seguras do tipo HTTPS.

Assinale a alternativa que apresenta a sequência correta de cima para baixo.

- A) V, F, F  
B) V, F, V



- C) F, V, V
- D) V, V, F
- E) F, F, V

**Comentários:**

(V) os dois podem executar páginas dinâmicas (ex.: ASP no IIS, PHP no Apache e no IIS).

(F) o IIS não roda nativamente no Linux.

(V) ambos servidores suportam conexões seguras do tipo HTTPS, assim como os outros servidores Web também!

Gabarito: B

15. (INSTITUTO MAIS/SETEC/2021) Em um ambiente Windows Server 2016, no qual se tem a necessidade de disponibilizar, por meio de um Servidor Web, páginas Active Server Pages (.ASP), nativamente, é necessário executar o serviço

- A) Apache.
- B) IIS.
- C) DNS.
- D) ZABIX.

**Comentários:**

IIS: uma de suas funcionalidades mais utilizadas é a geração de páginas HTML dinâmicas, através da tecnologia proprietária ASP (*Active Server Pages*), mas também pode utilizar outras tecnologias com adição de módulos de terceiros. Depois do lançamento da plataforma .NET (em 2002), o IIS ganhou também a função de gerenciar o ASP.NET. Este é formado basicamente por dois tipos de aplicações:

- Páginas Web: tradicionais, acessadas por usuários (extensão ASPX);
- Web Services: funções disponibilizadas pela rede, chamada por aplicativos ASMX.

Gabarito: B

16. (CEFET-MG/CEFET-MG/2022) O IIS (Internet Information System), na sua versão 7 ou superior, possui um utilitário que facilita a administração por meio de comandos intuitivos, sem a necessidade de ferramentas com interface gráfica. Esse utilitário pode criar, por exemplo, um backup dos arquivos de configuração do IIS.

O nome desse utilitário é:

- A) Lsass.exe
- B) W3wp.exe
- C) Rundll32.exe
- D) Appcmd.exe
- E) Taskmgr.exe



### Comentários:

Dentre as características do IIS listadas na aula, vimos essa:

- Uma ferramenta usada para provisionar sites e executar vários comandos para editar a configuração é a AppCmd.exe.

Gabarito: D



## LISTA DE QUESTÕES – CEBRASPE

1. (CEBRASPE/MEC/2011) A partir da instalação do IIS é disponibilizado um sítio-padrão cujas pastas estão instaladas no servidor, no caminho físico c:\inetpub\wwwroot.

2. (CEBRASPE/MEC/2015) Julgue o próximo item, relativo ao IIS (Internet Information Server) 7.0.

O IIS 7.0 é composto por módulos que podem ser adicionados e removidos do servidor web e que permitem a customização de seu servidor, com adição e remoção de características específicas.

3. (CEBRASPE/PG-DF/2021) A respeito dos serviços de Internet e de sua instalação na plataforma Windows, julgue o item seguinte.

Caso um servidor Web IIS (Internet Information Services) seja instalado e configurado para servir conteúdo estático, este tipo de conteúdo será entregue ao usuário como uma página HTML exatamente como armazenada.

4. (CEBRASPE/APEX Brasil/2022) Em relação à API de administração do IIS (Internet Information Services) da Microsoft, assinale a opção correta.

A) A configuração do IIS por meio da API de administração é feita utilizando-se o aplicativo cmd.exe.

B) A API de administração serve para auxiliar no processo de autenticação do usuário, fornecendo o serviço de Kerberos.

C) A API é um microsserviço executado na máquina de destino e é a única etapa necessária para disponibilizar o gerenciamento remoto à máquina.

D) A API de administração do IIS baseia-se nos protocolos SOAP para fornecer uma interface que utiliza XML e WSDL para troca de mensagens e comandos.



## GABARITO

01	02	03	04		
Certo	Certo	Certo	Certo		



## LISTA DE QUESTÕES – FGV

1. (FGV/TJ-SC/2015) Em um servidor de aplicação Internet Information Services (IIS) são executadas três aplicações web diferentes (1), (2) e (3). As aplicações (1) e (2) foram colocadas em um mesmo application pool (A) e a aplicação (3) foi colocada em outro application pool (B). Está correto afirmar que:
  - A) permissões de acesso requisitadas para (2) aplicam-se a (3);
  - B) falhas severas na execução de (1) podem afetar (2);
  - C) requisições feitas para (1) também podem ser tratadas por (2);
  - D) (1) e (3) podem usar um mesmo arquivo de log independentemente, sem precisar de um mutex;
  - E) (1) não pode ser interrompido sem que (2) também o seja.
2. (FGV/TJ-RO/2021) No Windows Server, o Gerenciador dos Serviços de Informação da Internet (IIS) é reiniciado pela linha de comando:
  - A) infostart
  - B) iisreset
  - C) iisrestart
  - D) restart p-iis
  - E) run iis
3. (FGV/TJ-TO/2022) O módulo nativo que pode ser adicionado à seção de configuração <globalModules> do Internet Information Services (IIS), versão 7, para habilitar um conjunto de regras de segurança com a finalidade de rejeitar solicitações suspeitas é:
  - A) DirectoryListingModule;
  - B) FailedRequestsTracingModule;
  - C) ProtocolSupportModule;
  - D) RequestFilteringModule;
  - E) RequestMonitorModule.





## GABARITO

01	02	03			
B	B	D			



## LISTA DE QUESTÕES – FCC

1. (FCC/TRT19/2011) O serviço que faz do Windows 2003 um Servidor Web é o IIS (Internet Information Services). Quando o IIS é instalado da maneira padrão, é disponibilizado um site com uma única página chamada iisstart.htm. Essa página pode ser encontrada no caminho físico
- A) c:\inetpub\wwwroot
  - B) c:\root\
  - C) c:\root\www
  - D) c:\http\www
  - E) c:\net\web
2. (FCC/TRE-SP/2017) Hipoteticamente, o Técnico, responsável pela administração do servidor com Windows Server 2012 do TRE-SP, realizou a instalação do serviço IIS com a configuração padrão de fornecimento. Considerando-se que ele não atribuiu o endereço IP e desligou o servidor da rede de computadores para evitar acesso externo, para que esse profissional realize o teste local do servidor IIS, ele deve utilizar um navegador e acessar, na Barra de endereços do navegador, a URL:
- A) http://127.0.0.1
  - B) ftp://10.0.0.1
  - C) https://192.168.0.1
  - D) http://192.168.0.1
  - E) https://255.255.255.0

## GABARITO

01	02				
A	A				



## LISTA DE QUESTÕES – VUNESP

1. (VUNESP/Pref. de P. Prud.-SP/2016) Por padrão, o servidor web IIS 7.5 armazena o conteúdo do site web no diretório:
- A) %SystemDrive%\var\wwwroot
  - B) %SystemDrive%\inetpub\www
  - C) %SystemDrive%\inetpub\DocumentRoot
  - D) %SystemDrive%\opt\www
  - E) %SystemDrive%\inetpub\wwwroot
2. (VUNESP/Pref. de Registro-SP/2018) Considere o cenário em que você instalou um programa servidor de página web (Apache ou IIS) em um computador. Para testar se o servidor de página web está funcionando, utilizando um navegador web nesse computador, deve-se inserir na barra de endereços do navegador o endereço: http://
- A) 10.0.0.1.
  - B) 127.0.0.1.
  - C) 169.254.0.1.
  - D) 172.16.0.1.
  - E) 192.168.0.1.

## GABARITO

01	02				
E	B				



## LISTA DE QUESTÕES – MULTIBANCAS

- (Quadrix/SERPRO/2014) IIS e Apache são servidores cuja finalidade é:
  - prover serviços de impressão para o Windows e o Linux, respectivamente.
  - prover serviços WEB.
  - prover serviços de WEB e e-mail, respectivamente.
  - prover serviços de WEB para Linux e Windows, respectivamente.
  - viabilizar programas para execução de páginas HTML também conhecidos como browsers.
  
- (UFRJ/UFRJ/2015) Considere as seguintes afirmativas acerca do servidor Web IIS 7:
  - Nesta versão do IIS não é possível mais instalar e configurar o serviço FTP.
  - No IIS 7, os pools de aplicativos podem ser executados dos seguintes modos: integrado ou clássico.
  - No IIS 7, a porta padrão para servir páginas em HTTPS é a 4343.Pode-se afirmar que:
  - apenas II e III estão corretas.
  - apenas I e II estão corretas.
  - apenas II está correta.
  - apenas I está correta.
  - I, II e III estão corretas.
  
- (Colégio Pedro II/Colégio Pedro II/2016) Assinale a alternativa que NÃO apresenta uma característica do servidor de aplicação IIS.
  - Gera páginas HTML dinâmicas.
  - Também é um servidor de aplicativo.
  - Executa códigos PHP, Perl, Javascript e ASP.
  - Usa o protocolo FTP para permitir que proprietários de sites carreguem e baixem arquivos.
  
- (IBFC/EBSERH/2016) Ao ser instalado o IIS (Internet Information Services), no disco rígido C:, por padrão os diretórios que serão criados para hospedagem de páginas e para FTP serão respectivamente:
  - C:/iisserver/wwwroot e C:/iisserver/ftproot
  - C:/inetpub/wwwroot e C:/inetpub/ftproot



- C) C:/iisserver/wwwdir e C:/iisserver/ftplib
- D) C:/inetpub/wwwdir e C:/inetpub/ftplib
- E) C:/inetpub/rootwww e C:/inetpub/rootftp

5. (INAZ do Pará/DPE-PR/2017) No Windows Server 2012, durante a instalação do Servidor WEB (IIS), que opção de segurança deve ser utilizada para analisar as requisições feitas ao servidor Web e impedir os ataques manipulando URLs?

- A) Autenticação Digest, resolve o problema do firewall e de redes internas e externas.
- B) Autenticação de Mapeamento de Certificado de Cliente.
- C) Autenticação do Windows.
- D) Restrições de IP e Domínio.
- E) Filtragem de solicitações.

6. (Quadrix/CRQ 4ª Região-SP/2018) Considerando os conceitos de servidores Web IIS e Apache, julgue o item.

O servidor web IIS é de uso exclusivo em sistemas operacionais Windows e o Apache roda apenas em sistemas que se baseiam em Unix/Linux.

7. (Quadrix/CRQ 4ª Região-SP/2018) Considerando os conceitos de servidores Web IIS e Apache, julgue o item.

O Apache baseia-se em software livre e pode ser usado gratuitamente em servidores web. O IIS é um software proprietário e seu uso é pago.

8. (Quadrix/CRQ 4ª Região-SP/2018) Considerando os conceitos de servidores Web IIS e Apache, julgue o item.

O IIS é configurado exclusivamente pela sua interface web de gerência e as configurações são guardadas internamente, não sendo possível acessá-las com editor de texto.

9. (Quadrix/CRQ 4ª Região-SP/2018) Considerando os conceitos de servidores Web IIS e Apache, julgue o item.

O documento-padrão apresentado por um servidor web, tanto IIS quanto Apache, é o index.html, presente na pasta inicial, caso não seja indicado outro.

10. (Quadrix/CRQ 4ª Região-SP/2018) Considerando os conceitos de servidores Web IIS e Apache, julgue o item.

Apesar do crescimento do uso dos servidores IIS, ele nunca conseguiu superar a quantidade de sites servidos por seu concorrente, Apache, que ainda é o mais usado na Internet.



11.(FUNRIO/Câm. de S. J. de M.-RJ/2018) Um técnico de informática instalou o IIS em um computador usando os parâmetros "default". Nesse contexto, os arquivos html e de imagem relacionados a um site serão armazenados em uma pasta referenciada pelo seguinte caminho:

- A) c:\inetpub\wwwroot
- B) c:\inetpub\webroot
- C) c:\inetpub\sites
- D) c:\inetpub\index
- E) c:\inetpub\default

12.(IBFC/Câm. de Feira de Sant.-BA/2018) Quando comparamos atualmente os servidores IIS (Internet Information Services) e o APACHE pode-se afirmar que:

- A) mundialmente os dois rodam predominantemente em sistemas operacionais Windows
- B) o servidor IIS ao longo do tempo dominou o mercado, deixando o APACHE em 3º lugar
- C) os dois tem o propósito de servir conteúdo baseado em requisições do tipo HTTP
- D) tanto o IIS como o APACHE são servidores desenvolvidos dentro da filosofia Open Source

13.(IBFC/IDAM/2019) Quanto às diferenças entre os servidores Apache e IIS, analise as afirmativas abaixo, dê valores Verdadeiro (V) ou Falso (F).

- ( ) o Apache pode rodar em várias plataformas como o Windows, Unix e Linux.
- ( ) normalmente o servidor IIS utiliza a sua linguagem proprietária o ASP.
- ( ) somente o Apache é capaz de responder as requisições HTTP de máquinas clientes.

Assinale a alternativa que apresenta a sequência correta de cima para baixo.

- A) V, F, F
- B) V, V, F
- C) F, V, V
- D) F, F, V

14.(IBFC/EBSERH/2020) Em relação aos servidores APACHE e IIS, analise as afirmativas abaixo e dê valores Verdadeiro (V) ou Falso (F).

- ( ) ambos os servidores podem executar páginas dinâmicas.
- ( ) tanto o APACHE como o IIS rodam nativamente no Linux.
- ( ) os dois servidores suportam conexões seguras do tipo HTTPS.

Assinale a alternativa que apresenta a sequência correta de cima para baixo.

- A) V, F, F



- B) V, F, V
- C) F, V, V
- D) V, V, F
- E) F, F, V

15. (INSTITUTO MAIS/SETEC/2021) Em um ambiente Windows Server 2016, no qual se tem a necessidade de disponibilizar, por meio de um Servidor Web, páginas Active Server Pages (.ASP), nativamente, é necessário executar o serviço

- A) Apache.
- B) IIS.
- C) DNS.
- D) ZABIX.

16. (CEFET-MG/CEFET-MG/2022) O IIS (Internet Information System), na sua versão 7 ou superior, possui um utilitário que facilita a administração por meio de comandos intuitivos, sem a necessidade de ferramentas com interface gráfica. Esse utilitário pode criar, por exemplo, um backup dos arquivos de configuração do IIS.

O nome desse utilitário é:

- A) Lsass.exe
- B) W3wp.exe
- C) Rundll32.exe
- D) Appcmd.exe
- E) Taskmgr.exe



## GABARITO

01	02	03	04	05	06
B	C	C	B	E	Errado
07	08	09	10	11	12
Certo	Errado	Certo	Certo	A	C
13	14	15	16		
B	B	B	D		





## SERVIDOR WEB NGINX

NGINX (leia-se "enguin écs") é um servidor Web gratuito, de código aberto (*open source*) e de alto desempenho, além de ser proxy reverso, bem como um servidor proxy IMAP/POP3 (e-mail). O NGINX é conhecido por seu alto desempenho, estabilidade, rico conjunto de recursos, configuração simples e baixo consumo de recursos (memória, por exemplo).

O NGINX possui um processo principal e vários processos trabalhadores (*workers*). O objetivo principal do processo principal é ler e avaliar a configuração e manter os processos *workers*. Os processos *workers* realizam o processamento real das solicitações. O NGINX utiliza um **modelo baseado em eventos** e mecanismos dependentes do sistema operacional para distribuir eficientemente as solicitações entre os processos *workers*. O número de processos *workers* é definido no arquivo de configuração e pode ser fixo para uma determinada configuração ou ajustado automaticamente ao número de núcleos de processadores disponíveis.

O NGINX é um servidor Web **multiplataforma**, podendo ser executado nos seguintes sistemas operacionais: Linux (Ubuntu, Debian, Red Hat, entre outras), Unix (FreeBSD, OpenBSD, Solaris, macOS) e Windows. Note que mesmo sendo suportado no Windows, geralmente os manuais de configuração (e as provas de concurso) focam no Linux, principalmente em relação a comandos ou onde fica determinado arquivo de configuração, por isso também focaremos no Linux.

A maneira com que o NGINX e seus módulos funcionam é determinada no arquivo de configuração, que por padrão possui o nome **nginx.conf** e está localizado no diretório **/usr/local/nginx/conf**, **/etc/nginx**, ou **/usr/local/etc/nginx**.

Ao contrário dos servidores tradicionais (Apache, por exemplo), o NGINX não depende de *threads* para lidar com solicitações. Em vez disso, ele usa **uma arquitetura assíncrona orientada a eventos muito mais escalável**. Essa arquitetura usa quantidades pequenas (e previsíveis) de memória.

Em relação ao Apache, o NGINX consome menos memória, pois lida com requisições Web baseadas em eventos, enquanto o Apache é um servidor baseado em processos. Tanto foi falado em eventos, mas o que é uma arquitetura orientada a eventos? É uma arquitetura composta de produtores e consumidores de eventos. Um produtor detecta ou percebe um evento e o representa como uma mensagem. Ele não conhece o consumidor nem o resultado do evento. Após um evento ser detectado, ele é transmitido do produtor para o consumidor por meio de canais, em que uma plataforma processa eventos de maneira assíncrona. O consumidor precisa ser informado quando um evento ocorre. Ele pode processar ou apenas ser afetado pelo evento.

O NGINX pode trabalhar junto com o Apache, passando as requisições Web primeiro no Nginx. Assim, o Apache não precisa servir arquivos estáticos, e pode depender do bom controle de *cache* feito pelo Nginx. um exemplo é o Wikipedia, que utiliza o NGINX como um servidor de terminação SSL, o qual é responsável por receber as requisições e repassar para outros servidores na rede, diminuindo a carga sobre outros servidores.



O Nginx é amplamente utilizado em diferentes sistemas operacionais devido à sua eficiência, escalabilidade e facilidade de configuração.

## Principais Características

Vejamos as principais características de um servidor NGINX abaixo, destacando aqueles que já caíram direta ou indiretamente em provas de concurso:

- Manipulação de arquivos estáticos;
- Auto-indexação de arquivos;
- Cache de descritor de arquivo aberto;
- Permite iniciar **várias instâncias de uma aplicação no mesmo servidor**;
- **Proxy reverso** acelerado com cache;
- **Balanceamento de carga** simples e com tolerância a falhas;
- Suporte acelerado com cache a: FastCGI, uWSGI, SCGI e servidores memcached;
- **Arquitetura modular**;
- Suporta SSL e TLS;
- **Servidores virtuais baseados em nome e baseados em IP**;
- **Conexões keep-alive** e suporte a pipeline;
- **Configuração flexível**;
- Reconfiguração e atualização de um executável sem interrupção do serviço ao cliente;
- Acesso a formatos de log, buffer de escrita de log e de rotação rápida de registro;
- Códigos de erro 3xx-5xx (3xx: Redirecionamento – indica que algo mais precisa ser feito ou precisou ser feito para completar a solicitação; 4xx: Erro do cliente - indica que a solicitação não pode ser concluída ou contém a sintaxe incorreta; 5xx: Erro no servidor – o servidor falhou ao concluir a solicitação);
- Execução de diferentes funções, dependendo do endereço do cliente;
- Controle de acesso baseado em endereço IP do cliente e autenticação HTTP;
- Métodos PUT, DELETE, MKCOL, COPY e MOVE;
- Streaming de FLV e MP4;
- Limitação de taxa de resposta;
- Limitação do número de conexões ou pedidos simultâneas provenientes de um endereço;
- Perl incorporado.

## Iniciando, parando e recarregando a configuração

Para iniciar o NGINX, deve-se executar o servidor. Depois de iniciado, o NGINX pode ser controlado através do executável com o parâmetro -s, com a seguinte sintaxe:

```
nginx -s SINAL
```

Sendo que SINAL pode ser um dos seguintes:

- stop: desligamento rápido;
- quit: desligamento "gentil" (*graceful*);
- reload: recarrega o arquivo de configuração;
- reopen: reabre os arquivos de log.



Por exemplo, para parar os processos do NGINX, aguardando os processos *workers* terminarem de atender às solicitações atuais, deve-se executar o comando:

```
nginx -s quit
```

Importante salientar que esse comando deve ser executado pelo mesmo usuário que iniciou o NGINX.

As alterações realizadas no arquivo de configuração não serão aplicadas até que o comando de recarregar a configuração seja enviado para o NGINX ou até que ele seja reiniciado. Lembrando que para recarregar a configuração o comando é:

```
nginx -s reload
```

## Arquitetura e escalabilidade

Um dos pontos fortes do servidor NGINX é a escalabilidade e como o alto desempenho é mantido, mesmo em grande escala! Vejamos algumas características:

- Um mestre e vários processos de trabalho, executados sob um usuário sem privilégios;
- Keep-alive para dez mil conexões HTTP inativas demandam cerca de 2,5 MB de memória (podemos observar que o baixo consumo de memória RAM é um grande diferencial do NGINX);
- Operações de cópia de dados são mantidos a um mínimo.

## Proxy reverso

O proxy reverso no NGINX é uma função que permite ao servidor atuar como intermediário entre os clientes e outros servidores. Ele recebe as solicitações dos clientes e as redireciona para os servidores backend adequados, processa as respostas desses servidores backend e, em seguida, retorna as respostas aos clientes. Ou seja, o proxy reverso inverte a direção normal do tráfego da Web. Em vez de os clientes fazerem solicitações diretamente aos servidores backend, eles fazem as solicitações para o servidor NGINX, que então encaminha essas solicitações para os servidores backend apropriados. Isso pode ser útil em várias situações, como por exemplo:

- **Balanceamento de carga:** O proxy reverso pode distribuir as solicitações de entrada entre vários servidores backend para melhorar o desempenho e a capacidade de lidar com o tráfego;
- **Cache:** O proxy reverso pode armazenar em cache as respostas dos servidores backend e retorná-las diretamente para os clientes subsequentes, reduzindo a carga nos servidores backend e melhorando o tempo de resposta;
- **Segurança:** O proxy reverso pode atuar como uma camada de proteção (camada adicional) entre os clientes e os servidores backend, ocultando a infraestrutura interna e fornecendo recursos como filtragem de tráfego, autenticação e SSL/TLS;



- **Roteamento avançado:** O proxy reverso pode rotear solicitações com base em diferentes critérios, como o nome de domínio, a URL ou o cabeçalho HTTP, permitindo a hospedagem de vários sites ou aplicativos em um único servidor.

O NGINX é bastante utilizado como um servidor de proxy reverso devido ao seu desempenho, flexibilidade e recursos avançados. Ele fornece várias diretivas e opções de configuração para personalizar o comportamento do proxy reverso.

## Balanceamento de carga

O NGINX pode ser implantado como um balanceador de carga HTTP eficiente para distribuir o tráfego recebido e a carga de trabalho entre um grupo de servidores de aplicativos, sendo que em cada caso a resposta é retornada do servidor selecionado para o cliente apropriado. Os métodos de balanceamento de carga suportados são:

- **round-robin:** distribui solicitações para os servidores de aplicativos de forma round-robin, ou seja, as solicitações são distribuídas igualmente entre os servidores (uma solicitação para cada servidor até chegar no último e recomeça do primeiro). É utilizado por padrão quando nenhum método é especificado;
- **menos conectado (*least connected*):** atribui a próxima solicitação a um servidor menos ocupado (o servidor com o menor número de conexões ativas);
- **ip-hash:** uma função *hash* é utilizada para determinar qual servidor deve ser selecionado para a próxima solicitação com base no endereço IP do cliente. Esse método permite persistência de sessão (vincular um cliente a um servidor de aplicativos específico).

Para configurar o balanceamento de carga, pode-se utilizar a diretiva **upstream** em conjunto com outras diretivas relacionadas. Dentro do bloco `http`, deve-se adicionar um bloco `upstream` para definir os servidores que serão balanceados. Vamos a um exemplo que define três servidores:

```
http {  
  
    upstream teste {  
  
        # least_conn; apenas para mostrar como seria para habilitar o algoritmo "menos conectado"  
  
        # se não tiver nada, o padrão é o round-robin  
  
        server teste1.exemplo.com;  
  
        server teste2.exemplo.com;  
  
        server teste3.exemplo.com;  
  
    }  
}
```



```
...  
}
```

Dentro do bloco server, deve-se definir a configuração para lidar com as solicitações de entrada e encaminhá-las para os servidores balanceados. Aqui está um exemplo que usa a diretiva `proxy_pass`:

```
http {  
  
    ...  
  
    server {  
  
        listen 80;  
  
        location / {  
  
            proxy_pass http://teste;  
  
        }  
  
    }  
  
    ...  
  
}
```

Neste exemplo, o NGINX escutará as solicitações na porta 80 e as encaminhará para o bloco upstream chamado teste. A diretiva `proxy_pass` especifica o endereço dos servidores balanceados.

Após configurar o balanceamento de carga, o NGINX distribuirá automaticamente as solicitações entre os servidores especificados no bloco upstream. Por padrão o balanceamento será round-robin.

## Estrutura do Arquivo de Configuração

O Nginx é composto por **módulos** que são controlados por **diretivas** especificadas no arquivo de configuração (`nginx.conf`). As diretivas são divididas em **diretivas simples** e **diretivas em bloco**. Uma diretiva simples consiste no nome e nos parâmetros separados por espaços e termina com um ponto e vírgula (;). Uma diretiva em bloco tem a mesma estrutura de uma diretiva simples, mas em vez do ponto e vírgula, ela termina com um conjunto de instruções adicionais cercadas por chaves ({ e }). Se uma diretiva em bloco pode ter outras diretivas dentro das chaves, ela é chamada de contexto (exemplos: `events`, `http`, `server` e `location`).

As diretivas colocadas no arquivo de configuração fora de qualquer contexto são consideradas no contexto principal. As diretivas:



- `events` e `http` residem no `contexto principal`;
- `server` reside no contexto `http`;
- `location` reside no contexto `server`.

Os comentários são colocados após o sinal de #.

## Servindo Conteúdo Estático

Um servidor Web serve arquivos (ex.: imagens e páginas HTML estáticas). Dependendo da solicitação, os arquivos são servidos de diferentes diretórios locais: `/data/www` (que pode conter arquivos HTML) e `/data/images` (que contém imagens). Isso exigirá a edição do arquivo de configuração e a configuração de um bloco de servidor dentro do bloco `http` com dois blocos de localização.

Vamos ver um passo a passo de como seria. Primeiro, a criação do diretório `/data/www` e dentro dele deve ser colocado um arquivo `index.html` com qualquer conteúdo de texto, depois a criação do diretório `/data/images` algumas imagens devem ser colocadas dentro dele. Depois, o arquivo de configuração deve ser alterado, iniciando um novo bloco de servidor:

```
http {  
  
    server {  
  
    }  
  
}
```

Geralmente, o arquivo de configuração pode incluir vários blocos de servidor distintos por portas nas quais eles ouvem e por nomes de servidor. Uma vez que o NGINX decide qual servidor processa uma solicitação, ele testa a URI (*Uniform Resource Identifier*) especificada no cabeçalho da solicitação em relação aos parâmetros das diretivas de localização definidas dentro do bloco do servidor. Deve-se adicionar o seguinte bloco de localização ao bloco do servidor:

```
location / {  
  
    root /data/www;  
  
}
```

Este bloco de localização especifica o prefixo `/"` em comparação com a URI da solicitação. Para solicitações correspondentes, a URI será adicionada ao caminho especificado na diretiva `root`, ou seja, `/data/www`, para formar o caminho para o arquivo solicitado no sistema de arquivos local. Se houver vários blocos de localização correspondentes, o NGINX seleciona aquele com o prefixo mais longo. O bloco de localização acima fornece o prefixo mais curto, de comprimento um, e, portanto, somente se todos os outros blocos de localização falharem em fornecer uma correspondência, este bloco será usado. Em seguida, deve-se adicionar o segundo bloco de localização:



```
location /images/ {  
  
    root /data;  
  
}
```

Ele corresponderá a solicitações que começam com /images/ (a localização / também corresponde a tais solicitações, mas tem um prefixo mais curto). A configuração resultante do bloco de servidor deve ficar assim:

```
server {  
  
    location / {  
  
        root /data/www;  
  
    }  
  
    location /images/ {  
  
        root /data;  
  
    }  
  
}
```

Esta já é uma configuração que funciona em um servidor que escuta na porta padrão 80 (HTTP) e é acessível na máquina local em <http://localhost/>. Em resposta a solicitações com URIs que começam com /images/, o servidor enviará arquivos do diretório /data/images. Por exemplo, em resposta à solicitação <http://localhost/images/teste.png>, o NGINX enviará o arquivo /data/images/teste.png. Caso esse arquivo não exista, o nginx enviará uma resposta indicando o erro 404 ("Not found"). Solicitações com URIs que não começam com /images/ serão mapeadas para o diretório /data/www. Por exemplo, em resposta à solicitação <http://localhost/algum/exemplo.html>, o NGINX enviará o arquivo /data/www/algum/exemplo.html.

Para aplicar a nova configuração, o servidor NGINX deve ser iniciado (caso ainda não esteja em execução) ou deve ser enviado o sinal de recarregamento para o processo mestre:

```
nginx -s reload
```

Caso algo não esteja funcionando como esperado, pode-se tentar descobrir o motivo nos arquivos [access.log](#) e [error.log](#) no diretório [/usr/local/nginx/logs](#) ou [/var/log/nginx](#).

## Configurando um host virtual

Uma configuração comum em servidores Web é criar mais de um host para um único servidor, ou seja, um host virtual. Como fazer no NGINX? Primeiro deve-se criar um arquivo de configuração





para o host virtual. Normalmente, esses arquivos são armazenados no diretório `/etc/nginx/conf.d/`. Pode ser criado um novo arquivo com o nome do domínio que se deseja configurar, seguido da extensão `.conf`, como por exemplo, `meusite.conf`. Dentro do arquivo deve ser adicionado o seguinte bloco de configuração:

```
server {  
  
    listen 80;  
  
    server_name dominio_desejado.com www.dominio_desejado.com;  
  
    root /caminho_para_o_diretorio_do_site;  
  
    index index.html index.htm;  
  
    location / {  
  
        try_files $uri $uri/ =404;  
  
    }  
  
}
```

Vamos ver o significado das diretivas utilizadas (junto com argumentos):

- `listen 80;` : especifica a porta em que o NGINX irá ouvir as solicitações HTTP;
- `server_name` : define o nome do servidor, que deve ser o domínio do site desejado. Podem ser adicionados múltiplos nomes de servidor separados por espaços;
- `root` : especifica o diretório raiz onde os arquivos do *site* estão localizados;
- `index` : define a ordem de prioridade dos arquivos a serem servidos quando o NGINX recebe uma solicitação para o diretório raiz;
- `location /` : define as configurações para a localização raiz. A diretiva `try_files` tentará corresponder o URI da solicitação a um arquivo real. Se nenhum arquivo for encontrado, será retornado o código de erro 404 ("Not found").

Após a configuração do host virtual é necessário que o DNS do domínio configurado esteja apontando para o endereço IP do servidor em que o NGINX está sendo executado.

## Configurando um Servidor Proxy Simples

Um dos usos frequentes do NGINX é configurá-lo como um servidor proxy, ou seja, um servidor que recebe solicitações, as encaminha para os servidores proxy, recupera as respostas e as envia para os clientes. Vamos ver como é a configuração de um servidor proxy básico, que atende solicitações de imagens com arquivos do diretório local e envia todas as outras solicitações para um servidor proxy. Neste exemplo, ambos os servidores serão definidos em uma única instância do NGINX.





Primeiro, é necessário definir o servidor proxy adicionando mais um bloco de servidor ao arquivo de configuração com o seguinte conteúdo:

```
server {  
  
    listen 8080;  
  
    root /data/up1;  
  
    location / {  
  
    }  
  
}
```

Este é um exemplo de um servidor simples que "ouve" na porta 8080 (anteriormente, a diretiva listen não tinha sido especificada, pois a porta padrão 80 estava sendo usada) e mapeia todas as solicitações para o diretório /data/up1 no sistema de arquivos local.

Agora, utilizando a configuração do servidor visto anteriormente, vamos modificar para transformar em uma configuração de servidor proxy. No primeiro bloco de localização, deve-se colocar a diretiva proxy\_pass com o protocolo, nome e porta do servidor proxy especificados no parâmetro (no caso mostrado é http://localhost:8080):

```
server {  
  
    location / {  
  
        proxy_pass http://localhost:8080;  
  
    }  
  
    location /images/ {  
  
        root /data;  
  
    }  
  
}
```

Agora vamos modificar o segundo bloco de localização, que atualmente mapeia as solicitações com o prefixo /images/ para os arquivos no diretório /data/images, para que corresponda às solicitações de imagens com extensões de arquivo típicas. O bloco de localização modificado fica assim:

```
location ~ /\.(gif|jpg|png)$ {  
  
    root /data/images;
```



```
}
```

O parâmetro mostrado é uma expressão regular que corresponde a todas as URIs que terminam com .gif, .jpg ou .png. Uma expressão regular deve ser precedida por ~. As solicitações correspondentes serão mapeadas para o diretório /data/images.

Quando o NGINX seleciona um bloco de localização para atender a uma solicitação, ele primeiro verifica as diretivas de localização que especificam prefixos, lembrando a localização com o prefixo mais longo e, em seguida, verifica as expressões regulares. Se houver correspondência com uma expressão regular, o nginx seleciona essa localização; caso contrário, ele seleciona a localização lembrada anteriormente.

A configuração resultante de um servidor proxy fica assim:

```
server {  
    location / {  
        proxy_pass http://localhost:8080/  
    }  
    location ~ \.(gif|jpg|png)$ {  
        root /data/images;  
    }  
}
```

Este exemplo de servidor filtrará as solicitações que terminam com .gif, .jpg ou .png e as mapeará para o diretório /data/images (adicionando a URI ao parâmetro da diretiva root) e encaminhará todas as outras solicitações para o servidor proxy configurado acima. Para aplicar a nova configuração, o sinal de recarregamento deve ser enviado ao NGINX:

```
nginx -s reload
```

## Algumas diretivas

### listen

Embora já tenhamos visto a diretiva listen em nossos exemplos, vale a pena explorar um pouco mais, pois ela tende a ser cobrada em mais detalhes. Essa diretiva é usada para especificar o endereço IP e a porta em que o servidor NGINX deve escutar as solicitações. A sintaxe básica é:

```
listen endereço_IP:porta opções;
```

Sendo que:



- endereço\_IP: é o endereço IP no qual o servidor NGINX deve escutar as solicitações. Pode ser um endereço IP específico ou:
  - o default\_server: indica que o servidor NGINX deve ser o servidor padrão para qualquer endereço IP que não esteja explicitamente configurado;
  - o \* : indica que o servidor NGINX deve escutar em todas as interfaces de rede disponíveis;
  - o localhost: indica que o servidor NGINX deve escutar apenas no endereço IP local (127.0.0.1).
- porta: é o número da porta em que o servidor NGINX deve escutar as solicitações. É comum usar a porta padrão HTTP 80 para as solicitações não criptografadas e a porta padrão HTTPS 443 para solicitações criptografadas usando SSL/TLS;
- opções: parâmetros adicionais. Alguns exemplos comuns ssl para habilitar a criptografia SSL/TLS para conexões seguras e default\_server para configurar o servidor como o servidor padrão quando vários servidores estão configurados.

Vamos ver alguns exemplos:

```
listen 80;
```

faz com que o servidor NGINX escute as solicitações na porta 80 em todas as interfaces de rede disponíveis.

```
listen 192.168.0.1:8080;
```

faz com que o servidor NGINX escute as solicitações no endereço IP 192.168.0.1 e na porta 8080.

```
listen 443 ssl;
```

faz com que o servidor NGINX escute as solicitações na porta 443 e habilite a criptografia SSL/TLS para conexões seguras. Note que foi usada a porta padrão HTTPS (443) para o SSL, mas poderia ser qualquer outra.

A diretiva listen também suporta endereços IPv6. Para especificar um endereço IPv6, deve-se colocar o endereço entre colchetes. Exemplo:

```
listen [::]:80;
```

Neste exemplo o servidor NGINX escutará as solicitações na porta 80 para todos os endereços IPv6 disponíveis.

Para especificar um endereço IPv6 específico, pode-se usar o endereço completo entre colchetes. Ex.:

```
listen [2001:0db8::1]:80;
```



Neste caso, o servidor NGINX escutará as solicitações na porta 80 apenas para o endereço IPv6 específico 2001:0db8::1.

### [worker\\_processes](#)

Diretiva utilizada para definir o número de processos de trabalho que serão criados para manipular as solicitações do servidor. Cada processo de trabalho é responsável por lidar com várias conexões e execução de tarefas simultâneas. Sintaxe:

```
worker_processes número;
```

Onde "número" é o número desejado de processos de trabalho. Exemplo:

```
worker_processes 5;
```

Nesse exemplo o servidor NGINX será configurado para criar e executar cinco processos de trabalho para manipular as solicitações. O número ideal de processos de trabalho varia dependendo do hardware do servidor, tráfego esperado e configuração específica do aplicativo. Geralmente se recomenda ajustar esse valor com base nos recursos do sistema e nas necessidades de desempenho. Um valor muito alto pode sobrecarregar o servidor, enquanto um valor muito baixo pode resultar em baixo desempenho.

Além disso, o NGINX também suporta a utilização de múltiplos processos em diferentes núcleos de CPU com o objetivo de melhorar o desempenho e a capacidade de resposta. Isso pode ser configurado através do uso da diretiva [worker\\_cpu\\_affinity](#).

### [worker\\_connections](#)

Define o número máximo de conexões simultâneas que podem ser abertas por um processo de trabalho (worker). Sintaxe:

```
worker_connections número;
```

O padrão é:

```
worker_connections 512;
```

Importante salientar que esse número inclui todas as conexões (conexões com servidores proxy, entre outros), não apenas conexões com os clientes. Outra consideração importante é que o número real de conexões simultâneas não pode exceder o limite atual do número máximo de arquivos abertos, que pode ser alterado por meio de `worker_rlimit_nofile`.

### [add\\_header](#)

Existe um mecanismo denominado **CORS** (*Cross-Origin Resource Sharing*), utilizado para controlar as políticas de compartilhamento de recursos entre diferentes origens (como o próprio nome sugere, mas nem sempre a banca vai mostrar o significado da sigla!). esse mecanismo permite que um servidor especifique quais origens possuem permissão para acessar recursos



(arquivos, APIs etc.) e quais tipos de requisições são permitidas. Poderíamos dizer que se trata de um filtro de acordo com permissões!

Para configurar o CORS é necessário adicionar alguns cabeçalhos de resposta nas configurações do servidor. Aí entra a diretiva `add_header` (o nome já sugere - adiciona cabeçalho). A sintaxe é:

```
add_header Header-Personalizado Valor;
```

O Header-Personalizado corresponde ao nome do cabeçalho de resposta e o Valor corresponde ao valor que se deseja retornar. Essa diretiva pode ser definida nos blocos `http`, `server` ou `location`.

Vamos a um exemplo, em que há uma seleção (`if`), e caso seja positivo, os cabeçalhos são retornados:

```
server {  
    ...  
    location / {  
        if ($request_method = 'OPTIONS') {  
            add_header 'Access-Control-Allow-Origin' '*';  
            add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS';  
            add_header 'Access-Control-Max-Age' 1728000;  
            add_header 'Content-Type' 'text/plain charset=UTF-8';  
            add_header 'Content-Length' 0;  
            return 204;  
        }  
        add_header 'Access-Control-Allow-Origin' '*';  
        add_header 'Access-Control-Allow-Methods' 'GET, POST';  
        add_header 'Access-Control-Allow-Headers'  
'DNT,X-CustomHeader,Keep-Alive,User-Agent,X-Requested-With,If-Modified-Since,Cache-  
Control,Content-Type';  
        add_header 'Access-Control-Expose-Headers' 'Content-Length,Content-Range';  
    }  
}
```



```
...  
}
```

No exemplo acima estamos configurando o servidor para permitir qualquer origem (\*) e os métodos HTTP GET e POST. Ao invés de qualquer origem (\*) podem ser colocados domínios específicos para permitir apenas essas origens.

## pid

Diretiva utilizada para especificar o caminho e o nome do arquivo onde o processo principal do servidor NGINX irá escrever seu identificador de processo (PID - *Process ID*). Sintaxe:

```
pid caminho/arquivo;
```

Exemplo:

```
pid /var/run/nginx.pid;
```

O arquivo que armazena o PID é útil para várias finalidades, como por exemplo monitoramento, controle e gerenciamento do processo do servidor NGINX. Com isso ele pode ser usado para parar ou reiniciar o servidor NGINX através do comando `nginx -s stop` ou `nginx -s reload`, respectivamente.

## user

Diretiva utilizada para especificar qual usuário e grupo devem ser usados pelo processo principal do servidor NGINX. Ela define as permissões e privilégios do processo para acessar arquivos, diretórios e recursos do sistema. Sintaxe:

```
user nome_usr [nome_grupo];
```

Onde:

- "nome\_usr " é o nome do usuário que será utilizado;
- "nome\_grupo" (opcional) é o nome do grupo ao qual o usuário pertence.

Exemplo:

```
user www-data;
```

Nesse exemplo, o servidor NGINX será executado usando o usuário "www-data" (um usuário comumente usado para servidores Web no sistema operacional Linux).

Por padrão, se a diretiva "user" não for especificada no arquivo de configuração, o processo principal será executado com as permissões do usuário e grupo definidos no momento da compilação do NGINX.



## Recurso de correio

Algumas características relacionadas ao recurso de servidor de correio eletrônico, ainda não observadas em provas de concurso:

- Redirecionamento do usuário para servidor IMAP ou POP3 usando um HTTP externo de autenticação do servidor;
- Autenticação do usuário usando um HTTP externo de autenticação do servidor e redirecionamento de ligação a um servidor SMTP interno;
- Métodos de autenticação: POP3 (USER/PASS, APOP, AUTH LOGIN/PLAIN/CRAM-MD5), IMAP (LOGIN, AUTH LOGIN/PLAIN/CRAM-MD5), SMTP (AUTH LOGIN/PLAIN/CRAM-MD5). Com esse monte de sigla é importante saber que a autenticação pode ser em texto plano (PLAIN) ou com o algoritmo de hásh MD5;
- Suporte a SSL e TLS.



## QUESTÕES COMENTADAS – CEBRASPE

1. (CEBRASPE/ABIN/2018) A respeito de escalabilidade, segurança, balanceamento de carga, fail-over e replicação de estados em sistema gerenciador web, julgue o item subsequente.

Embora, em um sistema gerenciador web, dividir as responsabilidades e a carga que cada tarefa executa torna o sistema mais rápido, mais escalável e de fácil manutenção, tecnologias recentes como o NGINX não permitem iniciar várias instâncias de uma aplicação no mesmo servidor, o que garante maior confiabilidade ao serviço.

Comentários:

Na teoria sobre NGINX, vimos uma grande lista das principais características, sendo que algumas foram destacadas (mostradas abaixo).

- Permite iniciar várias instâncias de uma aplicação no mesmo servidor;
- Proxy reverso acelerado com cache;
- Balanceamento de carga simples e com tolerância a falhas;
- Arquitetura modular;
- Servidores virtuais baseados em nome e baseados em IP;
- Conexões keep-alive e suporte a pipeline;
- Configuração flexível.

Gabarito: Errado

2. (CEBRASPE/ABIN/2018) Julgue o item a seguir, a respeito de sistemas gerenciadores web e de Apache e NGINX.

No NGINX, a configuração de servidores virtuais exige que a diretiva utilizada para especificar o endereço onde o servidor escuta as requisições seja definida usando-se o IPv4.

Comentários:

Pode ser usado também o IPv6, como por exemplo:

```
listen [2001:0db8::1]:80;
```

Neste caso, o servidor NGINX escutará as solicitações na porta 80 apenas para o endereço IPv6 específico 2001:0db8::1.

Gabarito: Errado

3. (CEBRASPE/SERPRO/2021) Tendo como referência servidores Nginx, considere a configuração a seguir.

```
http {  
    upstream appfood {  
        server s1.appfooduat.com;  
        server s2.appfooduat.com;  
        server s3.appfooduat.com;
```





```
}  
server {  
    listen 80;  
    location / {  
        proxy_pass http://appfood;  
    }  
}
```

A partir dessas informações, julgue o item que se segue.

A configuração apresentada é capaz de efetuar balanceamento de carga para acessos ao endereço `http://appfood`.

Comentários:

Seguindo a configuração mostrada na questão, o NGINX escutará as solicitações na porta 80 e as encaminhará para o bloco upstream chamado `appfood`. A diretiva `proxy_pass` especifica o endereço dos servidores balanceados. Após configurar o balanceamento de carga, o NGINX distribuirá automaticamente as solicitações entre os servidores especificados no bloco upstream. Por padrão o balanceamento será round-robin.

Gabarito: Certo

4. (CEBRASPE/SERPRO/2021) Tendo como referência servidores Nginx, considere a configuração a seguir.

```
http {  
    upstream appfood {  
        server s1.appfooduat.com;  
        server s2.appfooduat.com;  
        server s3.appfooduat.com;  
    }  
    server {  
        listen 80;  
        location / {  
            proxy_pass http://appfood;  
        }  
    }  
}
```

A partir dessas informações, julgue o item que se segue.



O método de balanceamento na configuração apresentada será *least-connected*, que indicará as requisições para o servidor com o menor número de conexões ativas.

#### Comentários:

Por padrão, o balanceamento é *round-robin*. Para habilitar o algoritmo "menos conectado", deveria estar assim:

...

```
upstream appfood {  
    least_conn;  
    server s1.appfooduat.com;  
    server s2.appfooduat.com;  
    server s3.appfooduat.com;  
}
```

...

Gabarito: Errado

5. (CEBRASPE/APEX Brasil/2022) No que se refere ao NGINX, assinale a opção que indica a funcionalidade que possibilita fazer proxy de solicitações para um servidor HTTP ou não HTTP, utilizando-se um protocolo especificado.

- A) *gzip\_proxied*
- B) *location*
- C) *nginx\_substitutions\_filter*
- D) *reverse proxy*

#### Comentários:

O proxy reverso no NGINX é uma função que permite ao servidor atuar como intermediário entre os clientes e outros servidores. Ele recebe as solicitações dos clientes e as redireciona para os servidores backend adequados, processa as respostas desses servidores backend e, em seguida, retorna as respostas aos clientes. Ou seja, o proxy reverso inverte a direção normal do tráfego da Web. Em vez de os clientes fazerem solicitações diretamente aos servidores backend, eles fazem as solicitações para o servidor NGINX, que então encaminha essas solicitações para os servidores backend apropriados. Isso pode ser útil em várias situações, como por exemplo:

- **Balanceamento de carga:** O proxy reverso pode distribuir as solicitações de entrada entre vários servidores backend para melhorar o desempenho e a capacidade de lidar com o tráfego;
- **Cache:** O proxy reverso pode armazenar em cache as respostas dos servidores backend e retorná-las diretamente para os clientes subsequentes, reduzindo a carga nos servidores backend e melhorando o tempo de resposta;



- **Segurança:** O proxy reverso pode atuar como uma camada de proteção (camada adicional) entre os clientes e os servidores backend, ocultando a infraestrutura interna e fornecendo recursos como filtragem de tráfego, autenticação e SSL/TLS;
- **Roteamento avançado:** O proxy reverso pode rotear solicitações com base em diferentes critérios, como o nome de domínio, a URL ou o cabeçalho HTTP, permitindo a hospedagem de vários sites ou aplicativos em um único servidor.

Gabarito: D

6. (CEBRASPE/DPE-RO/2022) Na configuração de um servidor web Nginx para prover páginas HTML, é possível adicionar configurações para utilizar HTTPS; assinale a opção que apresenta a diretiva e os argumentos que devem ser utilizados na configuração para ativar esse recurso.

- A) listen 443 ssl;
- B) return 443 ssl;
- C) sslprotocols 443 ssl;
- D) keepalive 443 ssl;
- E) servername 443 ssl;

#### Comentários:

Aqui não tem mistério nenhum! Sabemos que para "ouvir" solicitações existe a diretiva "listen" e a única que possui tal diretiva é a primeira alternativa. É importante lembrar que para o Apache se utiliza "Listen 443 https" e no NGINX se utiliza "listen 443 ssl;". Cuidado!

Gabarito: A

7. (CEBRASPE/DPE-RO/2022) O NGINX open source oferece suporte a diversos métodos de balanceamento de carga, entre eles, inclui-se aquele no qual as solicitações são distribuídas igualmente entre os servidores, sendo levados em consideração os pesos de servidor; esse método de balanceamento de carga é usado por padrão, não havendo diretiva para habilitá-lo. Ele é denominado

- A) IP hash.
- B) least time.
- C) least connections.
- D) round robin.
- E) generic hash.

#### Comentários:

Os métodos de balanceamento de carga suportados são:

- *round-robin*: distribui solicitações para os servidores de aplicativos de forma round-robin, ou seja, as solicitações são distribuídas igualmente entre os servidores (uma solicitação para cada servidor até chegar no último e recomeça do primeiro). É utilizado por padrão quando nenhum método é especificado;



- menos conectado (*least connected*): atribui a próxima solicitação a um servidor menos ocupado (o servidor com o menor número de conexões ativas);
- *ip-hash*: uma função *hash* é utilizada para determinar qual servidor deve ser selecionado para a próxima solicitação com base no endereço IP do cliente. Esse método permite persistência de sessão (vincular um cliente a um servidor de aplicativos específico).

Gabarito: D



## QUESTÕES COMENTADAS – FGV

1. (FGV/Senado Federal/2022) Um analista de TI pretende habilitar o mecanismo CORS para um site hospedado no servidor web nginx 1.18.

O parâmetro correto a ser implantado no arquivo de configuração do site é:

A) `add_header Access-Control-Allow-Origin *`;

B) `add_header Strict-Transport-Security "max-age=31536000; includeSubdomains"`;

C) `add_header Content-Security-Policy "default-src 'self' https: data: 'unsafe-inline' 'unsafe-eval';" always`;

D) `add_header Cache-Control "max-age=0, no-cache, no-store, must-revalidate"`;

E) `add_header X-Frame-Options "SAMEORIGIN" always`;

**Comentários:**

Existe um mecanismo denominado CORS (*Cross-Origin Resource Sharing*), utilizado para controlar as políticas de compartilhamento de recursos entre diferentes origens (como o próprio nome sugere, mas nem sempre a banca vai mostrar o significado da sigla!). esse mecanismo permite que um servidor especifique quais origens possuem permissão para acessar recursos (arquivos, APIs etc.) e quais tipos de requisições são permitidas. Poderíamos dizer que se trata de um filtro de acordo com permissões!

Para configurar o CORS é necessário adicionar alguns cabeçalhos de resposta nas configurações do servidor. Aí entra a diretiva `add_header` (o nome já sugere - adiciona cabeçalho). A sintaxe é:

```
add_header Header-Personalizado Valor;
```

O parâmetro " `Access-Control-Allow-Origin *`" é bastante conhecido e significa que permite qualquer origem, sem determinar domínios específicos.

**Gabarito:** A



## QUESTÕES COMENTADAS – VUNESP

1. (VUNESP/UNICAMP/2022) No Servidor Web Nginx, a diretiva `worker_connections` determina o número

- A) máximo de instâncias que o Nginx pode executar, caso seja necessário escalar.
- B) máximo de conexões que um mesmo cliente pode abrir com o servidor.
- C) de instâncias do Nginx que devem ser executadas quando o serviço for iniciado.
- D) de conexões que cada processo do Nginx deve suportar simultaneamente.
- E) de conexões que um recurso web sendo servido pode abrir para obter recursos de outros servidores.

### Comentários:

A diretiva `worker_connections` define o número máximo de conexões simultâneas que podem ser abertas por um processo de trabalho (`worker`). Sintaxe:

```
worker_connections número;
```

O padrão é:

```
worker_connections 512;
```

É importante salientar que esse número inclui todas as conexões (conexões com servidores proxy, entre outros), não apenas conexões com os clientes. Outra consideração importante é que o número real de conexões simultâneas não pode exceder o limite atual do número máximo de arquivos abertos, que pode ser alterado por meio de `worker_rlimit_nofile`.

Gabarito: D



## QUESTÕES COMENTADAS – MULTIBANCAS

1. (IF-PE/IF-PE/2016) O texto "\_\_\_\_\_ é um servidor proxy HTTP e reverso, bem como um servidor de proxy de email. Segundo a Netcraft em fevereiro de 2013 o \_\_\_\_\_ foi considerado o maior servidor HTTP, responsável por 12,81% dos servidores de HTTP e IMAP/POP3. O \_\_\_\_\_ é um servidor web rápido, leve, e com inúmeras possibilidades de configuração para melhor performance. Dentre as características do \_\_\_\_\_ destacam-se a capacidade de se configurar um proxy reverso acelerado com cache e um balanceamento de carga simples". A expressão que completa corretamente as lacunas é

- A) Apache.
- B) Tomcat.
- C) Nginx.
- D) JBoss.
- E) IIS.

### Comentários:

Conforme vimos no primeiro parágrafo da teoria, NGINX é um servidor Web gratuito, de código aberto (*open source*) e de alto desempenho, além de ser proxy reverso, bem como um servidor proxy IMAP/POP3 (e-mail). O NGINX é conhecido por seu alto desempenho, estabilidade, rico conjunto de recursos, configuração simples e baixo consumo de recursos (memória, por exemplo). Podemos ver em sites de tecnologia que o uso de NGINX está aumentando bastante e figura no topo dos mais usados.

Gabarito: C

2. (CCV-UFC/UFC/2019) Sobre a configuração e execução do servidor web Nginx, assinale a alternativa correta.

- A) O comando "nginx -s reload" é utilizado para reiniciar o servidor.
- B) A directive user define as credenciais de usuário e grupo usadas pelos worker processes.
- C) Quanto mais alto for o valor configurado na directive worker\_processes, melhor é o desempenho do servidor
- D) A directive pid deve ser configurada com um número que define o número que guarda o ID do processo principal do servidor.
- E) Linhas contendo directives podem terminar com ou sem o sinal de ";" (ponto e vírgula). As duas formas irão carregar a configuração sem erro.

### Comentários:

A) "nginx -s SINAL", sendo que SINAL pode ser um dos seguintes:

- stop: desligamento rápido;
- quit: desligamento "gentil" (*graceful*);



- reload: recarrega o arquivo de configuração;
- reopen: reabre os arquivos de log.

B) user: diretiva utilizada para especificar qual usuário e grupo devem ser usados pelo processo principal do servidor NGINX. Ela define as permissões e privilégios do processo para acessar arquivos, diretórios e recursos do sistema. Sintaxe: user nome\_usr [nome\_grupo];

Onde: "nome\_usr " é o nome do usuário que será utilizado e "nome\_grupo" (opcional) é o nome do grupo ao qual o usuário pertence.

C) worker\_processes: utilizada para definir o número de processos de trabalho que serão criados para manipular as solicitações do servidor. Cada processo de trabalho é responsável por lidar com várias conexões e execução de tarefas simultâneas. Sintaxe: worker\_processes número;

Onde "número" é o número desejado de processos de trabalho. Exemplo: worker\_processes 5;

Um valor muito alto pode sobrecarregar o servidor, enquanto um valor muito baixo pode resultar em baixo desempenho.

D) pid: utilizada para especificar o caminho e o nome do arquivo onde o processo principal do servidor NGINX irá escrever seu identificador de processo (PID - *Process ID*). Sintaxe: pid caminho/arquivo;

Exemplo: pid /var/run/nginx.pid;

E) Como vimos em todos os exemplos, as diretivas DEVEM terminar com sinal de ponto e vírgula.

Gabarito: B

3. (UFSC/UFSC/2019) A respeito do servidor nginx, analise as afirmativas abaixo e assinale a alternativa correta.

I. O número de processos de trabalho (worker processes) é fixo e deve ser definido em arquivo de configuração.

II. Frequentemente é utilizado como um servidor indireto (proxy server).

III. Suporta três mecanismos para balanceamento de carga: round-robin, least-connected e ip-hash.

A) Somente a afirmativa II está correta.

B) Somente as afirmativas I e III estão corretas.

C) Somente as afirmativas I e II estão corretas.

D) Somente as afirmativas II e III estão corretas.

E) Todas as afirmativas estão corretas.

Comentários:

I. O número de processos de trabalho (worker processes) é fixo e deve ser definido em arquivo de configuração.





ERRADA - A diretiva "worker\_processes" é usada para definir esse número no arquivo de nginx.conf, portanto pode ser mudado.

II. Frequentemente é utilizado como um servidor indireto (proxy server).

CORRETA - NGINX é um servidor Web gratuito, de código aberto (open source) e de alto desempenho, além de ser proxy reverso, bem como um servidor proxy IMAP/POP3 (e-mail). Frequentemente é usado como proxy.

III. Suporta três mecanismos para balanceamento de carga: round-robin, least-connected e ip-hash.

CORRETA - Os métodos de balanceamento de carga suportados são:

- round-robin: distribui solicitações para os servidores de aplicativos de forma round-robin, ou seja, as solicitações são distribuídas igualmente entre os servidores (uma solicitação para cada servidor até chegar no último e recomeçar do primeiro). É utilizado por padrão quando nenhum método é especificado;
- menos conectado (least connected): atribui a próxima solicitação a um servidor menos ocupado (o servidor com o menor número de conexões ativas);
- ip-hash: uma função hash é utilizada para determinar qual servidor deve ser selecionado para a próxima solicitação com base no endereço IP do cliente. Esse método permite persistência de sessão (vincular um cliente a um servidor de aplicativos específico).

Gabarito: D

4. (FUNDATEC/IF Farroupilha-RS/2023) Na instalação de um servidor de aplicação NGINX, qual porta vem configurada, por padrão, como HTTP listen?

- A) 80
- B) 8084
- C) 22
- D) 2020
- E) 8080

Comentários:

Comentários:

```
server {  
  
    listen 80;  
  
    server_name dominio_desejado.com www.dominio_desejado.com;  
  
    root /caminho_para_o_diretorio_do_site;  
  
    index index.html index.htm;  
  
    location / {
```



```
        try_files $uri $uri/ =404;  
    }  
}
```

Vamos ver o significado das diretivas utilizadas (junto com argumentos):

- **listen 80;** : especifica a porta em que o NGINX irá ouvir as solicitações HTTP (**80 é a porta padrão**);
- **server\_name** : define o nome do servidor, que deve ser o domínio do site desejado. Podem ser adicionados múltiplos nomes de servidor separados por espaços;
- **root** : especifica o diretório raiz onde os arquivos do *site* estão localizados;
- **index** : define a ordem de prioridade dos arquivos a serem servidos quando o NGINX recebe uma solicitação para o diretório raiz;
- **location /** : define as configurações para a localização raiz. A diretiva **try\_files** tentará corresponder o URI da solicitação a um arquivo real. Se nenhum arquivo for encontrado, será retornado o código de erro 404 ("Not found").

Gabarito: A



## LISTA DE QUESTÕES – CEBRASPE

1. (CEBRASPE/ABIN/2018) A respeito de escalabilidade, segurança, balanceamento de carga, fail-over e replicação de estados em sistema gerenciador web, julgue o item subsequente.

Embora, em um sistema gerenciador web, dividir as responsabilidades e a carga que cada tarefa executa torna o sistema mais rápido, mais escalável e de fácil manutenção, tecnologias recentes como o NGINX não permitem iniciar várias instâncias de uma aplicação no mesmo servidor, o que garante maior confiabilidade ao serviço.

2. (CEBRASPE/ABIN/2018) Julgue o item a seguir, a respeito de sistemas gerenciadores web e de Apache e NGINX.

No NGINX, a configuração de servidores virtuais exige que a diretiva utilizada para especificar o endereço onde o servidor escuta as requisições seja definida usando-se o IPv4.

3. (CEBRASPE/SERPRO/2021) Tendo como referência servidores Nginx, considere a configuração a seguir.

```
http {  
    upstream appfood {  
        server s1.appfooduat.com;  
        server s2.appfooduat.com;  
        server s3.appfooduat.com;  
    }  
    server {  
        listen 80;  
        location / {  
            proxy_pass http://appfood;  
        }  
    }  
}
```

A partir dessas informações, julgue o item que se segue.

A configuração apresentada é capaz de efetuar balanceamento de carga para acessos ao endereço `http://appfood`.

4. (CEBRASPE/SERPRO/2021) Tendo como referência servidores Nginx, considere a configuração a seguir.

```
http {
```



```
upstream appfood {  
    server s1.appfooduat.com;  
    server s2.appfooduat.com;  
    server s3.appfooduat.com;  
}  
server {  
    listen 80;  
    location / {  
        proxy_pass http://appfood;  
    }  
}
```

A partir dessas informações, julgue o item que se segue.

O método de balanceamento na configuração apresentada será least-connected, que indicará as requisições para o servidor com o menor número de conexões ativas.

5. (CEBRASPE/APEX Brasil/2022) No que se refere ao NGINX, assinale a opção que indica a funcionalidade que possibilita fazer proxy de solicitações para um servidor HTTP ou não HTTP, utilizando-se um protocolo especificado.

- A) gzip\_proxied
- B) location
- C) nginx\_substitutions\_filter
- D) reverse proxy

6. (CEBRASPE/DPE-RO/2022) Na configuração de um servidor web Nginx para prover páginas HTML, é possível adicionar configurações para utilizar HTTPS; assinale a opção que apresenta a diretiva e os argumentos que devem ser utilizados na configuração para ativar esse recurso.

- A) listen 443 ssl;
- B) return 443 ssl;
- C) sslprotocols 443 ssl;
- D) keepalive 443 ssl;
- E) servername 443 ssl;

7. (CEBRASPE/DPE-RO/2022) O NGINX open source oferece suporte a diversos métodos de balanceamento de carga, entre eles, inclui-se aquele no qual as solicitações são distribuídas



igualmente entre os servidores, sendo levados em consideração os pesos de servidor; esse método de balanceamento de carga é usado por padrão, não havendo diretiva para habilitá-lo. Ele é denominado

- A) IP hash.
- B) least time.
- C) least connections.
- D) round robin.
- E) generic hash.

## GABARITO

01	02	03	04	05	06
Errado	Errado	Certo	Errado	D	A
07					
D					



## LISTA DE QUESTÕES – FGV

1. (FGV/Senado Federal/2022) Um analista de TI pretende habilitar o mecanismo CORS para um site hospedado no servidor web nginx 1.18.

O parâmetro correto a ser implantado no arquivo de configuração do site é:

A) `add_header Access-Control-Allow-Origin *`;

B) `add_header Strict-Transport-Security "max-age=31536000; includeSubdomains"`;

C) `add_header Content-Security-Policy "default-src 'self' https: data: 'unsafe-inline' 'unsafe-eval';" always`;

D) `add_header Cache-Control "max-age=0, no-cache, no-store, must-revalidate"`;

E) `add_header X-Frame-Options "SAMEORIGIN" always`;

## GABARITO

01

A



## LISTA DE QUESTÕES – VUNESP

1. (VUNESP/UNICAMP/2022) No Servidor Web Nginx, a diretiva `worker_connections` determina o número
- A) máximo de instâncias que o Nginx pode executar, caso seja necessário escalar.
  - B) máximo de conexões que um mesmo cliente pode abrir com o servidor.
  - C) de instâncias do Nginx que devem ser executadas quando o serviço for iniciado.
  - D) de conexões que cada processo do Nginx deve suportar simultaneamente.
  - E) de conexões que um recurso web sendo servido pode abrir para obter recursos de outros servidores.

## GABARITO

01

D



## LISTA DE QUESTÕES – MULTIBANCAS

1. (IF-PE/IF-PE/2016) O texto "\_\_\_\_\_ é um servidor proxy HTTP e reverso, bem como um servidor de proxy de email. Segundo a Netcraft em fevereiro de 2013 o \_\_\_\_\_ foi considerado o maior servidor HTTP, responsável por 12,81% dos servidores de HTTP e IMAP/POP3. O \_\_\_\_\_ é um servidor web rápido, leve, e com inúmeras possibilidades de configuração para melhor performance. Dentre as características do \_\_\_\_\_ destacam-se a capacidade de se configurar um proxy reverso acelerado com cache e um balanceamento de carga simples". A expressão que completa corretamente as lacunas é
  - A) Apache.
  - B) Tomcat.
  - C) Nginx.
  - D) JBoss.
  - E) IIS.
  
2. (CCV-UFC/UFC/2019) Sobre a configuração e execução do servidor web Nginx, assinale a alternativa correta.
  - A) O comando "nginx -s reload" é utilizado para reiniciar o servidor.
  - B) A directive user define as credenciais de usuário e grupo usadas pelos worker processes.
  - C) Quanto mais alto for o valor configurado na directive worker\_processes, melhor é o desempenho do servidor
  - D) A directive pid deve ser configurada com um número que define o número que guarda o ID do processo principal do servidor.
  - E) Linhas contendo directives podem terminar com ou sem o sinal de ";" (ponto e vírgula). As duas formas irão carregar a configuração sem erro.
  
3. (UFSC/UFSC/2019) A respeito do servidor nginx, analise as afirmativas abaixo e assinale a alternativa correta.
  - I. O número de processos de trabalho (worker processes) é fixo e deve ser definido em arquivo de configuração.
  - II. Frequentemente é utilizado como um servidor indireto (proxy server).
  - III. Suporta três mecanismos para balanceamento de carga: round-robin, least-connected e ip-hash.
  - A) Somente a afirmativa II está correta.
  - B) Somente as afirmativas I e III estão corretas.





- C) Somente as afirmativas I e II estão corretas.
- D) Somente as afirmativas II e III estão corretas.
- E) Todas as afirmativas estão corretas.

4. (FUNDATEC/IF Farroupilha-RS/2023) Na instalação de um servidor de aplicação NGINX, qual porta vem configurada, por padrão, como HTTP listen?

- A) 80
- B) 8084
- C) 22
- D) 2020
- E) 8080

## GABARITO

01	02	03	04
C	B	D	A



# 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.