# JCA – Conceitos
# Engines – Repositórios de Objetos

- KeyStore
  - Representa um banco de dados de chaves e certificados de entidades acreditadas (trusted)
  - Chaves privadas armazenadas contém uma cadeia de certificados associados a fim de autenticar as chaves públicas correspondentes
  - PoContém certificado
- CertificateFactory
  - Utilizado na criação de certificados de chaves públicas e CRLs (Certification Revocation Lists)
- CertPathBuilder
  - Utilizado na criação de cadeias de certificados, também conhecidas como “Certification Paths”
- CertPathValidator
  - Utilizado na validação de cadeias de certificados
- CertStore
  - Utilizado na recuperação de certificados e CRLs de um repositório

# JCA – Certificados

- A JCA possui as classes que permitem a manipulação de certificados X509
  - X509Certificate
    - Representa um certificado X509
    - Especialização da classe java.security.cert.Certificate
  - CertificateFactory
    - Utilizado na criação de certificados de chaves públicas, CRLs (Certification Revocation Lists)
  - CertPathBuilder
    - Utilizado na criação de cadeias de certificados, também conhecidas como “Certification Paths”
  - CertPathValidator
    - Utilizado na validação de cadeias de certificados

# JCA – Certificados Armazenamento

- Algumas classes representam um repositório de informações de certificados
  - CertStore
    - Utilizado na recuperação de certificados e CRLs de um repositório
  - KeyStore
    - Representa um banco de dados de chaves e certificados de entidades acreditadas (trusted)
    - Contém Certificados (java.security.cert.Certificate)
    - Além disso pode conter outros objetos como chaves: secretas, privadas e públicas
      - Chaves privadas armazenadas contém uma cadeia de certificados associados a fim de autenticar as chaves públicas correspondentes

# JCA Keystore

- Objeto que possui a capacidade de armazenar chaves criptográficas e certificados. Disponível no pacote **java.security.KeyStore**
- Realizar a persistência destas informações e fornece métodos para inserir, pesquisar e alterar as informações
- Consiste de um mapa, onde cada objeto armazenado possui uma chave associada.
  - Esta chave é conhecida como *alias*
- Possui mecanismos para garantir uma segurança adicional no acesso às informações
  - Senha global
  - Senha por entrada
- Armazena diferentes tipos de objetos
  - KeyStore.PrivateKeyEntry – Chave privada (PrivateKey), que pode ser acompanhada por uma cadeia de certificados relacionados com a respectiva chave pública
  - KeyStore.SecretKeyEntry – Chave Secreta (SecretKey) utilizada em criptografia simétrica
  - KeyStore.TrustedCertificateEntry – Certificado (Certificate) pertencente a outra parte.

# JCA
# Keystore - Uso

- Durante seu uso é necessário
  - Obter uma instância da keystore

    ```
    KeyStore ks = KeyStore.getInstance(String type)
    ```
  - Carregar seu conteudo para memória ou iniciliar uma keystore vazia

    ```
    ks.load(InputStream stream, char[] password)
    ks.load(null, char[] password)
    ```
  - Inserir uma chave secreta ou privada

    ```
    ks.setKeyEntry(String alias, Key key,char[] password,Certificate[] chain)
    ```
    - alias – apelido que será utilizado para a chave
    - password - senha que poderá ser associada à chave
    - chain – cadeia de certificados correspodendo à chave pública associada. Somente é necessário quando a entrada armazenar uma chave privada (PrivateKey)
  - Inserir um certificado

    ```
    Key k = ks.setCertificateEntry(String alias, Certificate cert)
    ```
  - Recuperar uma chave secreta ou privada

    ```
    Key k = ks.getKey(String alias, char[] password)
    ```
  - Recuperar um certificado ou cadeia de certificados

    ```
    Certificate c = ks.getCertificate(String alias)
    Certificate[] ch = ks.getCertificateChain(String alias)
    ```
  - Persistir a KeyStore em disco

    ```
    ks.store(OutputStream stream, char[] password)
    ```

# JCA – Keystore Keytool

- A ferramenta keytool é uma ferramenta para gestão de chaves e certificados
- Permite ao usuário administrar suas chaves privadas e públicas e os certificados a elas associados
- Possibilita o armazenado de chaves secretas
- Comandos associados
  - Adicionar chaves e/ou certificados
    - -genkeypair
    - -genseckey
    - -importcert
    - -importkeystore
  - Exportar chaves e/ou certificados
    - -certreq
    - -exportcert
  - Acesso à chaves e/ou ceritificados
    - -list
    - -printcert
  - Gestão da Keystore
    - -storepasswd
    - -keypasswd

# JCA – Keystore
# Keytool – Comandos

- Adicionar chaves e/ou certificados
  - -genkeypair → geração de um par de chaves (pública e a privada)
  - -genseckey → geração de uma chave secreta
  - -importcert → importar o certificado de uma CA ou um Certificate Reply ou ainda uma cadeia de certificados (certificate chain)
  - -importkeystore → importa uma keystore já existente para a atual
- Exportar chaves e/ou certificados
  - -certreq → Gera um Certificate Signing Request (CSR) para ser enviado para uma CA
  - -exportcert → Exporta um valor associado a um alias para um arquivo
- Acesso à chaves e/ou ceritificados
  - -list → lista o conteudo de toda a keystore
  - -printcert → Le um certificado de um arquivo e imprime seu conteúdo
- Gestão da Keystore
  - -storepasswd → altera a senha associada à keystore
  - -keypasswd → altera a senha associada a uma entrada na keystore
  - -delete → apaga uma entrada na keystore a partir de seu alias
  - -changealias → altera o alias de uma entrada na keystore
  - -help → exibe o help

# JCA Keytool
# Geração Par de Chaves

```
-genkeypair   [-v] [-protected]
              [-alias <alias>]
              [-keyalg <keyalg>] [-keysize <keysize>]
              [-sigalg <sigalg>] [-dname <dname>]
              [-validity <valDays>] [-keypass <keypass>]
              [-keystore <keystore>] [-storepass <storepass>]
              [-storetype <storetype>] [-providername <name>]
              [-providerclass <provider_class_name> [-providerarg <arg>]] ...
              [-providerpath <pathlist>]
```

- alias – chave única utilizada na recuperação da chave ou certificado armazenado
- keyalg – algoritmo utilizado para geração do par de chaves. Ex: DSA, RSA, etc.
- keysize – tamanho da chave a ser criada. Ex: 512, 768, 1024, 2048
- validity – validade do par de chaves em dias
- keypass – senha utilizada para protegar a entrada na keystore. Sem esta senha não é possível manipular uma entrada na keystore. Deve possui ao minimo 6 caracteres
- keystore – nome do completo do arquivo onde está a armazenada keystore
- storepass – senha associada a keystore. Somente com esta senha é possível o acesso a keystore
- storetype – tipo associado à keystore. Ex: jceks, jks, pkcs12
- sigalg – Algorimto de assinatura a ser utilizado. Ex: SHA1withDSA para DSA e MD5withRSA para RSA
- dname – Nome distinto (Distinguished Name) associado ao apelido (alias) e também ao certificado (subject)

# JCA Keytool
# Nome Distinto (Distinguished Name)

- Nome definido na especificação X.500 e utilizado para identificar o subject e issuer em ceritifados do tipo X.509
- O nome contém as seguintes partes
  - commonName (cn) – Nome de uma pessoa. Ex. “Flavio Silva"
  - organizationUnit (ou) – Nome de um departamento ou divisão. Ex. “FACOM”
  - organizationName (o) – Nome da organização. Ex. “UFU”
  - localityName (l)– Nome da Cidade. Ex: “Uberlandia"
  - stateName (s) – Nome do estado – Ex. “Minas Gerais"
  - country (c) – Código de duas letras indicando o pais. Ex. “BR”
- Ao utilizar na linha de comando do keytool deve ser utilizado o seguinte formato:
  - "cn=Flavio Silva, ou=FACOM, o=UFU, l=Uberlandia, s=Minas Gerais, c=BR"

# JCA Keytool
# Processo para Importação

- Através da keytool é possível gerar um par de chaves e solicitar que a chave pública seja assinada por uma CA, utilizando o processo abaixo
  - Geração do Par de Chaves
    - keytool -genkeypair
  - Solicitação de um certificado (Certificate Request) a ser assinado por uma CA
    - keytool -certreq
  - Importando o certificado da CA
    - keytool -importcert
  - Importando o certificado devolvido (Certificate Reply) da CA
    - keytool -importcert -trustcacerts

# JCA Keytool Exemplos

- Geração do Par de Chaves

```
keytool -genkeypair -dname "cn=Flavio Silva, ou=FACOM, o=UFU,
l=Uberlandia, s=Minas Gerais, c=BR" -alias cert1 -keypass kps735 -
keystore C:\code\curso\security\ks.keystore -storepass ksp135 -
validity 180
```

- Solicitação de um certificado assinado de uma CA

```
keytool -certreq -alias cert1 -keypass kps735 -keystore
C:\code\curso\security\ks.keystore -storepass ksp135 -file
C:\code\curso\security\cert1.csr
```

- Importando o Certificado da CA

```
keytool -importcert -alias cert2 -keypass kps736 -keystore
C:\code\curso\security\ks.keystore -storepass ksp135 -file
C:\code\curso\security\AC_VALID.crt


keytool -importcert -alias cert3 -keypass kps736 -keystore
C:\code\curso\security\ks.keystore -storepass ksp135 -file
C:\code\curso\security\ICP-Brasilv3.crt
```

# JCA Keytool Exemplos

- Importando o Certificado devolvido (Certificate Reply) da CA

- Exportando um certificado com a chave públic autenticada pela CA

```
keytool -exportcert -alias cert1 -keypass kps735 -keystore
C:\code\curso\security\ks.keystore -storepass ksp135 -file
C:\code\curso\security\cert1.cer

keytool -exportcert -alias cert2 -keypass kps736 -keystore
C:\code\curso\security\ks.keystore -storepass ksp135 -file
C:\code\curso\security\cert2.cer

keytool -exportcert -alias cert3 -keypass kps736 -keystore
C:\code\curso\security\ks.keystore -storepass ksp135 -file
C:\code\curso\security\cert3.cer
```