Protótipo de software para atualização automática de versão de arquivos

Acadêmico:Airison Ambrosi

Orientador: Prof. Francisco Adell Péricas

Roteiro

- Introdução
- Segurança em redes
- Integridade dos dados
- Transferência de dados
- Requisitos de software
- Especificação
- Implementação
- Operacionalidade da implementação
- Resultados e discussão
- Conclusão e extensões

Introdução

- A atualização com controle de versão é eficiente no processo distribuição de um software

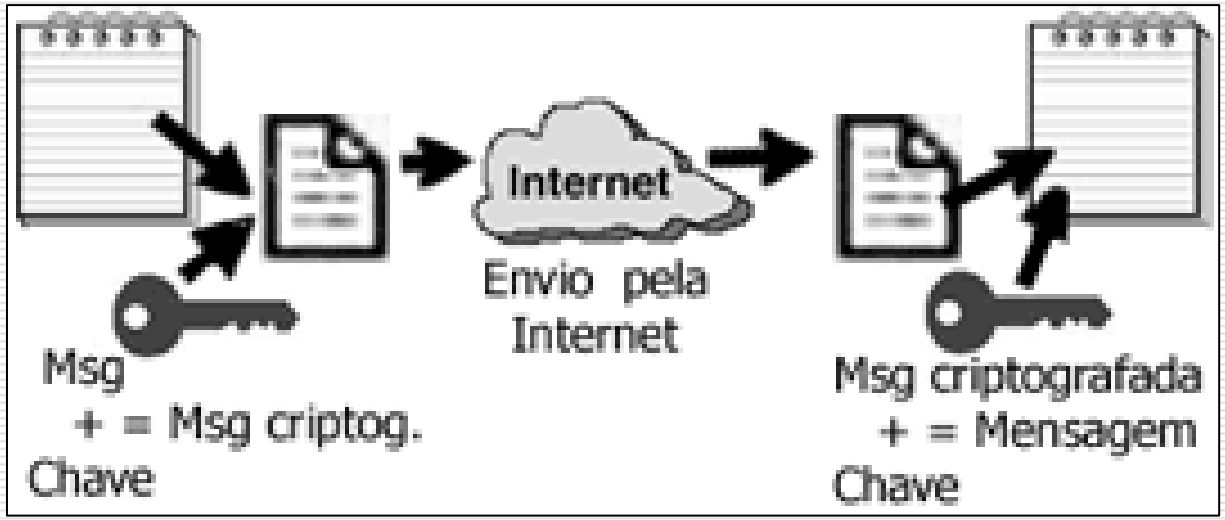
- O uso do algoritmo Hash substitue os algoritmos utilizados para verificacao de erros (CRC) e garantem a integridade de qualquer arquivo atualizado

Segurança em redes

- A segurança preocupa-se que pessoas mal intencionadas não leiam ou modifiquem mensagem enviadas a outros destinatários
- Sigilo: A criptografia é a solução

Segurança em redes: Sigilo

- Criptografia com chave secreta



- Criptografia com chave pública

Autenticação

- Processo de provar a própria identidade a alguém
- Os seres humanos autenticam pessoas atravéz de informações biometricas
- A autenticação deve ser a primeira tarefa a ser executada

Integridade dos dados

- Assinatura digital
- Algoritmo Hash
  - Irreversível
  - Mensagens diferentes não podem produzir resumos iguais
- SHA-1 – (Gera uma saída de 160 bits)
- MD5 – (Gera uma saída de 128 bits)

Exemplo simples de cálculo do valor "*hash*"

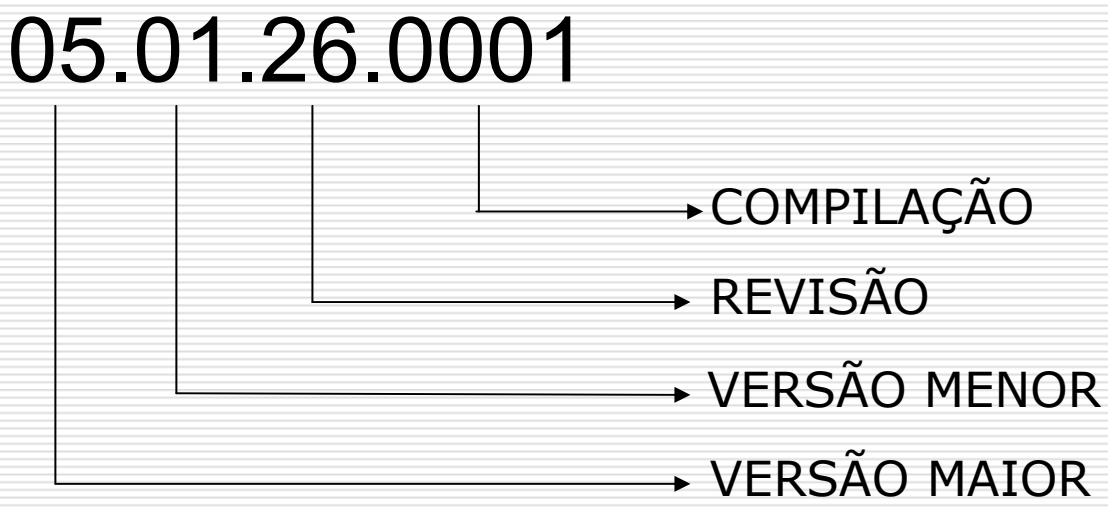
	Caractere	**Código ASCII (em decimal)**	**Código ASCII (em binário)**	**XOR**
1º Caractere	H	72	01001000	-
2º Caractere	O	79	01001111	00000111
3º Caractere	J	74	01001010	01001101
4º Caractere	E	69	01000101	**00001000**

Transferência de dados

- *File Transfer Protocol* (FTP) é o serviço padrão da Internet para a transferência de arquivos entre computadores.
- Trivial File Transfer (TFTP) sem mecanismos de autenticação e utiliza a UDP
- Secure File Transfer (SFTP) utiliza mecanismos de autenticação segura

Controle de versão

- Padronização do número de versão



Padrão XML

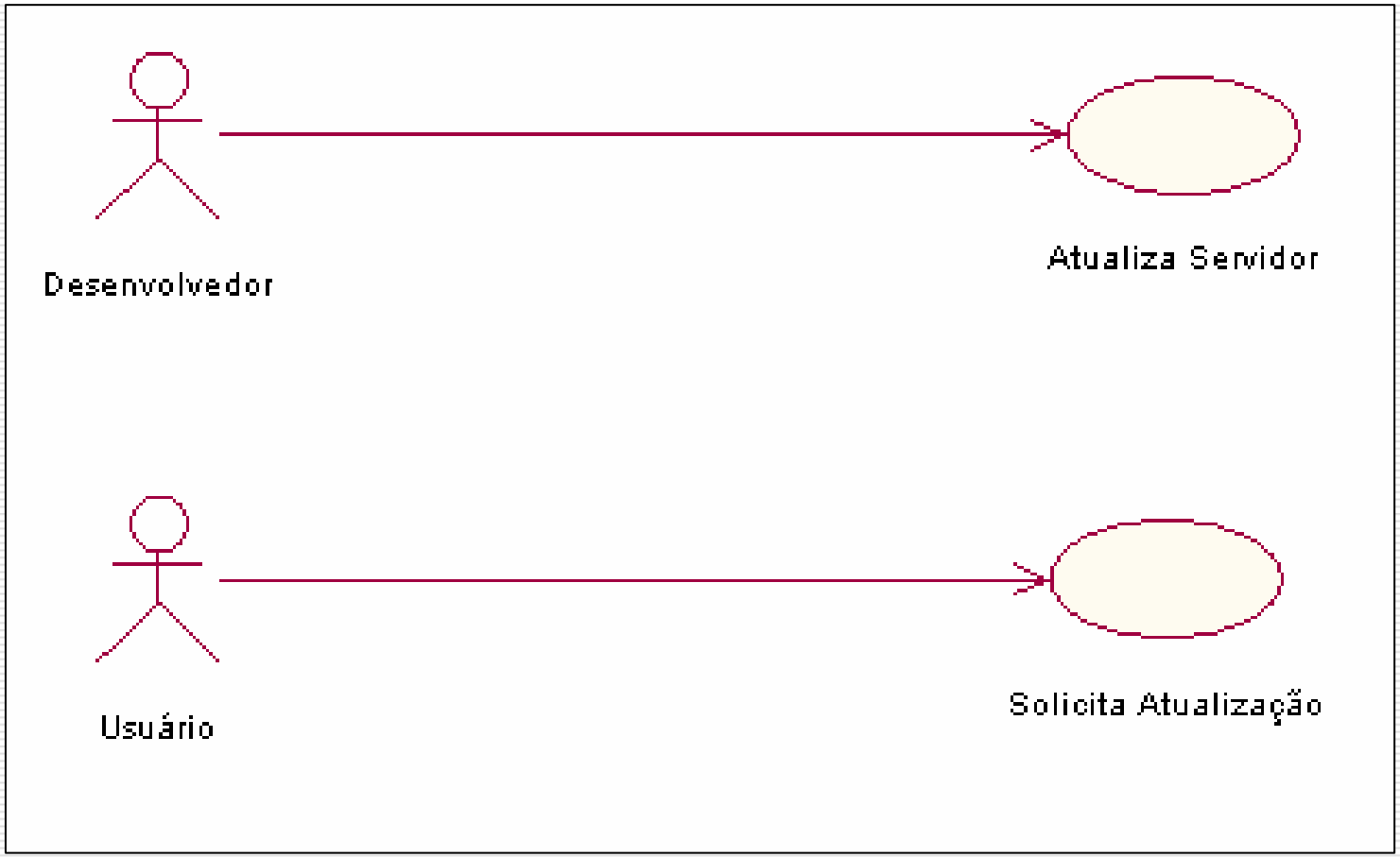
- *eXtensible Markup Language*
- Baseada na linguagem HTML
- Usa marcadores (*tags)* para descrever os dados

```
<?xml version="1.0" standalone="yes" ?>
<METADATA>
  <FIELDS>
     <FIELD attrname="Nome" fieldtype="string" WIDTH="100" />
  </FIELDS>
</METADATA>
```

Requisitos de software

- Transferência de arquivos (FTP)
- Utilização de criptografia (SSL)
- Controle de Versão (XML)
- Integridade dos dados (Hashing)
- Verificação da necessidade de atualização

Especificação: Diagrama de casos de uso



Especificação: Diagrama de classes

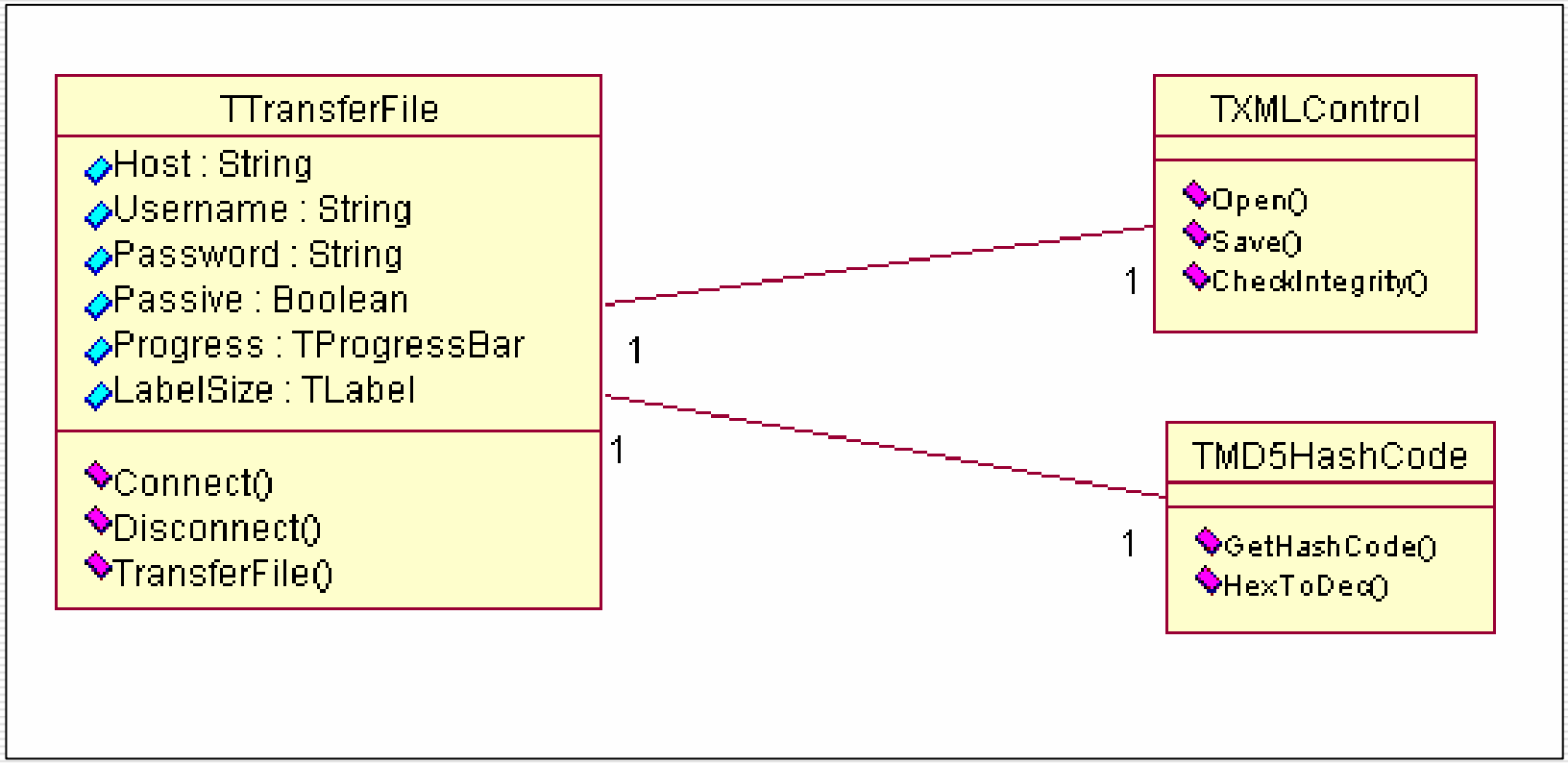


Diagrama de seqüência: Gerar Atualização

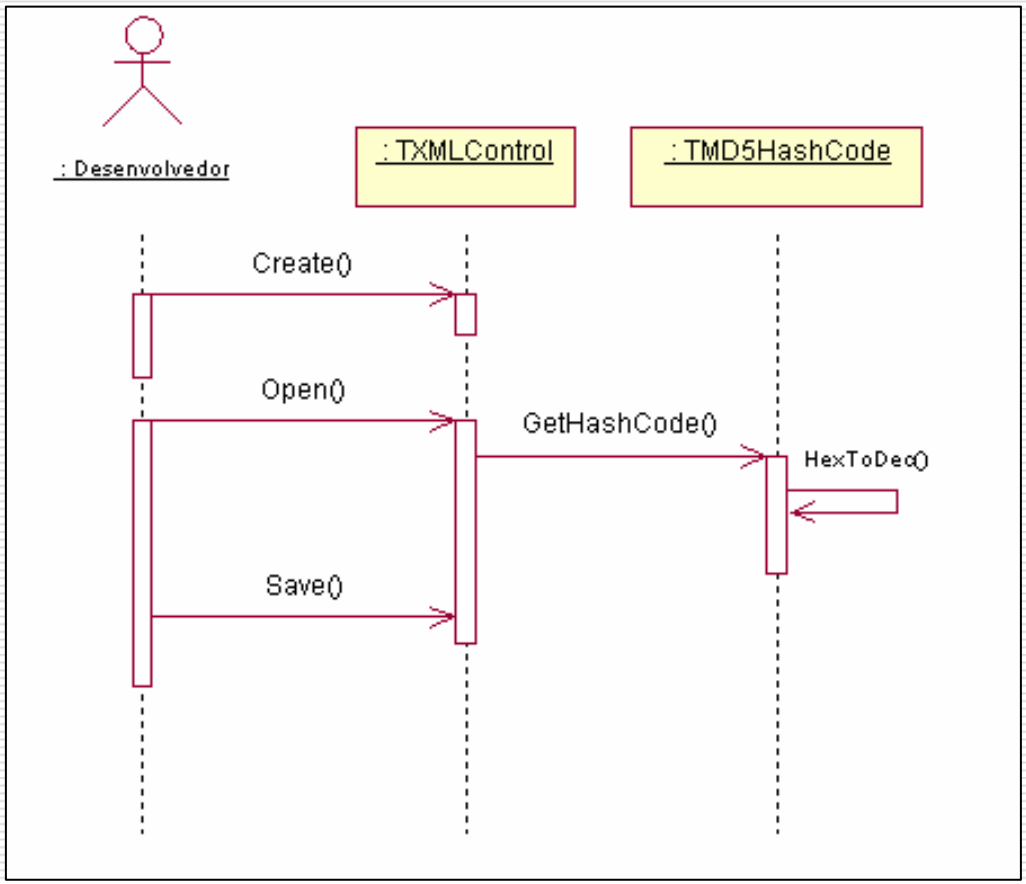
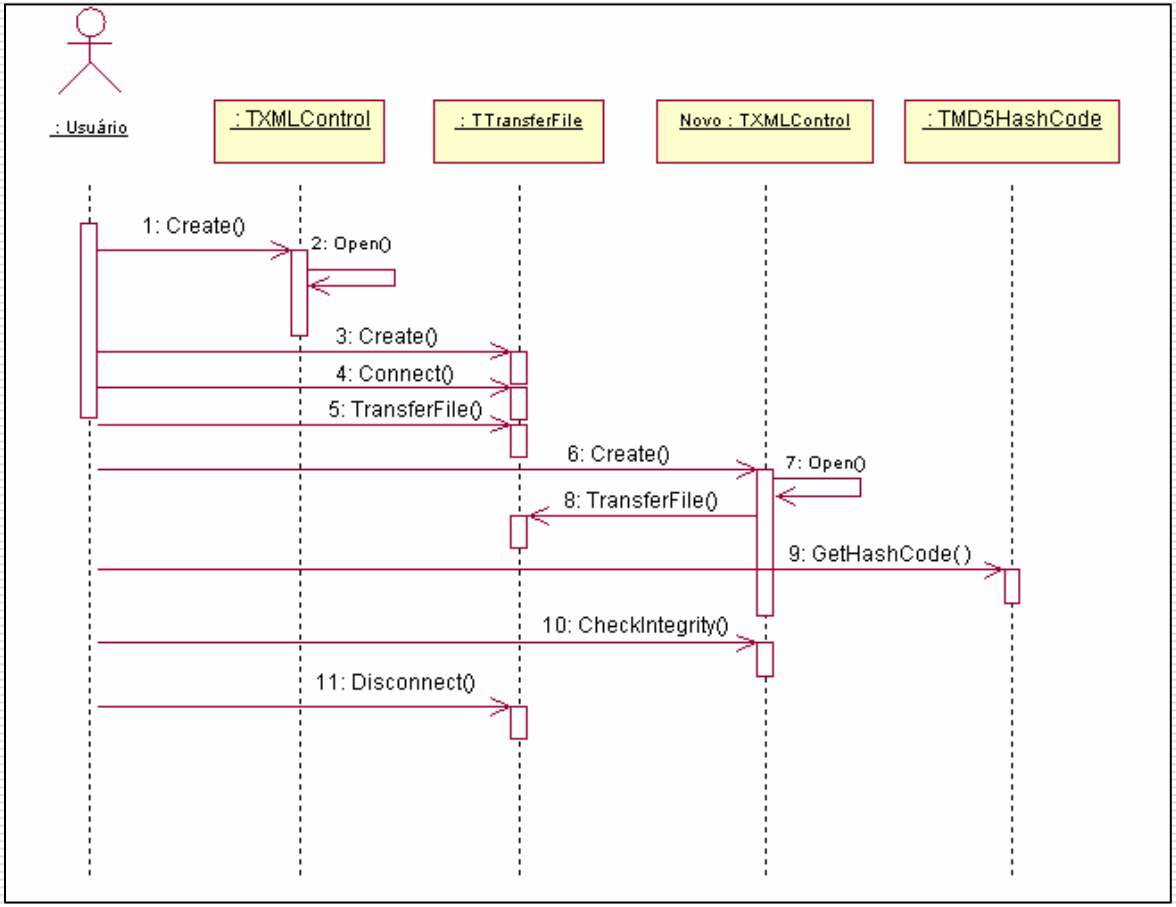


Diagrama de seqüência: Solicitar Atualização



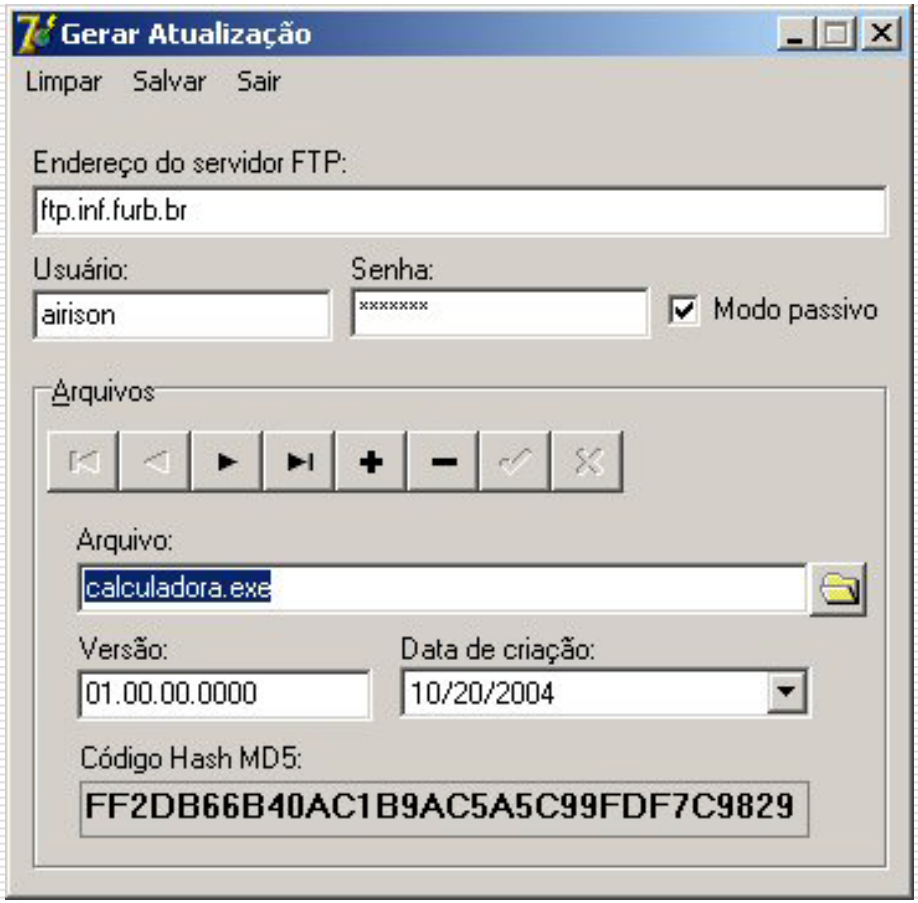
Implementação: Protótipo Gerar

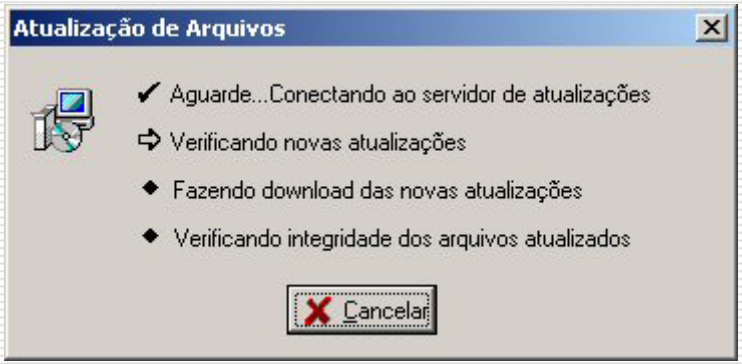
- Utilização de um arquivo no formato XML para controle de versão
- Gerar Hash do arquivo disponível para atualização

Implementação: Protótipo Atualizar

- Estabelecer conexão via protocolo FTP com suporte a SSL
- Verificação de necessidade de atualização a partir do controle de versão contido arquivo XML
- Transferência das atualizações
- Verificação de integridade via Hash

Operacionalidade da implementação

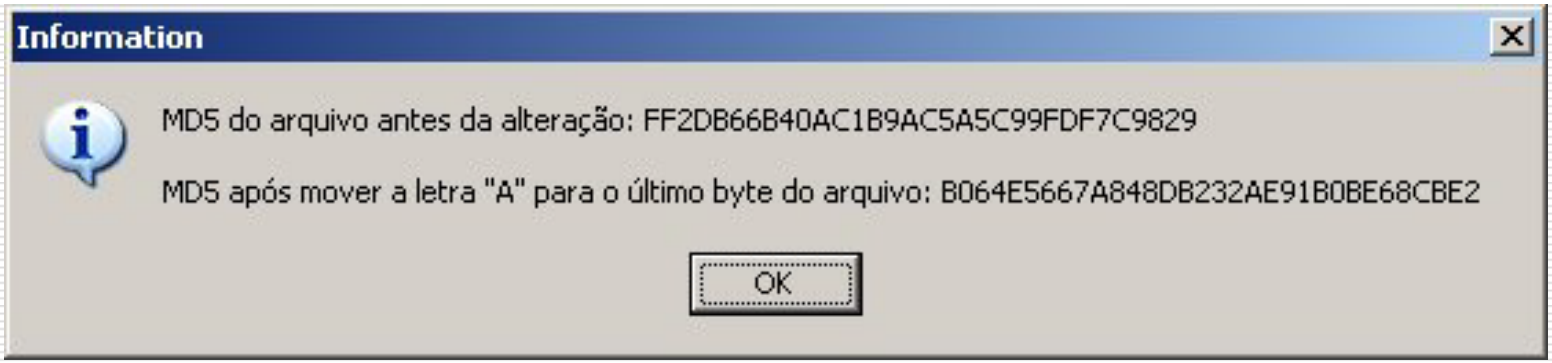




Resultados e Discussão

- Qualquer alteração por menor que seja no arquivo, gera um código Hash totalmente diferente

- O protótipo demonstra essa situação utilizando o parametro “teste”



Conclusão

- A conexão segura e a verificação de integridade tornam a atualização de arquivos muito segura
- Não foi utilizado CRC, pois o verficação atravéz do Hash é muito segura e eficiente
- A biblioteca de componentes INDY é uma ferramenta completa para soluções de rede e internet

Extensões

- Utilização de outros algoritmos como o SHA-1
- Protótipo de sincronização entre servidores espelhos (*mirrors)*
- Criação de um novo protocolo de transferência de arquivo como uma extensão ao FTP, incluindo verificação de integridade dos arquivos