Protótipo de um sistema para licenciamento de aplicativos Microsoft .NET baseado em assinatura digital XML

Acadêmico: Leonardo Chagas D'Ippolito
Orientador: Prof. Marcel Hugo

Blumenau, Dezembro de 2004

Roteiro

- Introdução
  - Objetivos
- Fundamentação teórica
  - Propriedade intelectual e mecanismos de proteção
  - Tecnologias utilizadas
  - Criptografia e assinaturas digitais
  - Trabalhos correlatos

Roteiro

- Desenvolvimento do sistema
  - Visão geral do sistema
  - Requisitos
  - Especificação
  - Implementação
  - Operacionalidade da implementação
  - Resultados e discussão
- Conclusões
  - Extensões

Introdução

Introdução

- Para produzir um *software* de qualidade são necessários consideráveis investimentos financeiros e uma equipe técnica qualificada
- Folha de São Paulo: 56% dos aplicativos utilizados atualmente no Brasil são piratas
- Pirataria de *software* significa reproduzir ilegalmente um programa de computador ou utilizá-lo sem a devida licença de uso
- Prejuízos decorrentes: menos empregos são gerados e evolução mais lenta do segmento tecnológico
- Um mecanismo de gerenciamento eletrônico de licenças é um instrumento técnico que representa uma maneira de reduzir ou dificultar a pirataria de *software*

Introdução

- Desafios para um mecanismo de proteção contra cópias:
  - Ter facilidade na obtenção de uma licença de uso
  - Poder confiar no arquivo de licença
  - Impedir que uma licença seja copiada para máquinas não autorizadas

Objetivos

- Desenvolver um sistema para proteger aplicativos Microsoft .NET contra cópias ilegais
- Composição do sistema:
  - Um módulo de licenciamento para ser integrado a um aplicativo .NET
  - Um módulo de gerenciamento em forma de aplicação *web* para expedição de licenças

Fundamentação teórica

Propriedade intelectual e mecanismos de proteção contra cópia

- A lei brasileira nº. 9609 de Fev./1998 define um programa de computador e estipula a pena para quem viola os seus direitos autorais
- Apenas a legislação não garante que as empresas produtoras de software estejam livres de prejuízos, porque a fiscalização é deficiente
- Que mecanismos técnicos podem ser usados para proteger um programa de computador?

Propriedade intelectual e mecanismos de proteção contra cópia

- Desbloqueio por senha consultada em manual
- Verificação de números seriais
- Ativação *online*
- Proteção com *hardware*
- Proteção com identificação do *host*
- Licenciamento com assinatura digital

Propriedade intelectual e mecanismos de proteção contra cópia

- Considerações sobre licenciamento de software:
  - Assim como em outras tecnologias no campo da segurança da informação, os mecanismos de proteção fazem parte do popularmente chamado "jogo de gato e rato"
  - Fazer o possível dentro dos limites de tecnologia e orçamento para proteger o *software*
  - Seebach(2003) alerta: *The point is to be able to use the software you purchase* (o propósito é poder usar o software que você comprou)

Tecnologias utilizadas

- Microsoft .NET Framework
  - Nova interface de programação para os serviços de API do Windows
  - Aceita diferentes linguagens que compilam para um mesmo módulo gerenciado *(assembly* .NET)
  - *Common Language Runtime* (CLR) : gerencia e executa o código intermediário embutido no *assembly*
  - Projeto Mono: um projeto de código aberto que implementa o .NET Framework nas plataformas Linux e Unix

Tecnologias utilizadas

- Tecnologia ASP.NET
  - Permite a construção de aplicações para a Internet
  - Suporta todas as linguagens do .NET Framework
  - Trabalha com o modelo *code behind*, onde a apresentação fica separada da lógica de processamento
  - A aplicação é compilada

Tecnologias utilizadas

- XML
  - Atribui significado à informação
  - Facilidade de manipulação das informações no arquivo de licença
  - Uso de schemas para enviar, receber e processar informações de forma padronizada
  - Baixo custo

Tecnologias utilizadas

- Web services
  - Qualquer funcionalidade acessível a partir da Internet
  - Frequentemente usa o formato XML empacotado em envelopes SOAP
  - Permite o desenvolvimento baseado em componentes que ficam expostos na Internet
  - Utiliza um canal de comunicação HTTP
  - Conecta plataformas heterogêneas

Criptografia e assinaturas digitais

- A criptografia é a preocupação em tornar uma comunicação privada (RSA Laboratories)
- Categorias:
  - De chave secreta ou simétrica
  - De chaves públicas ou assimétrica

Criptografia e assinaturas digitais

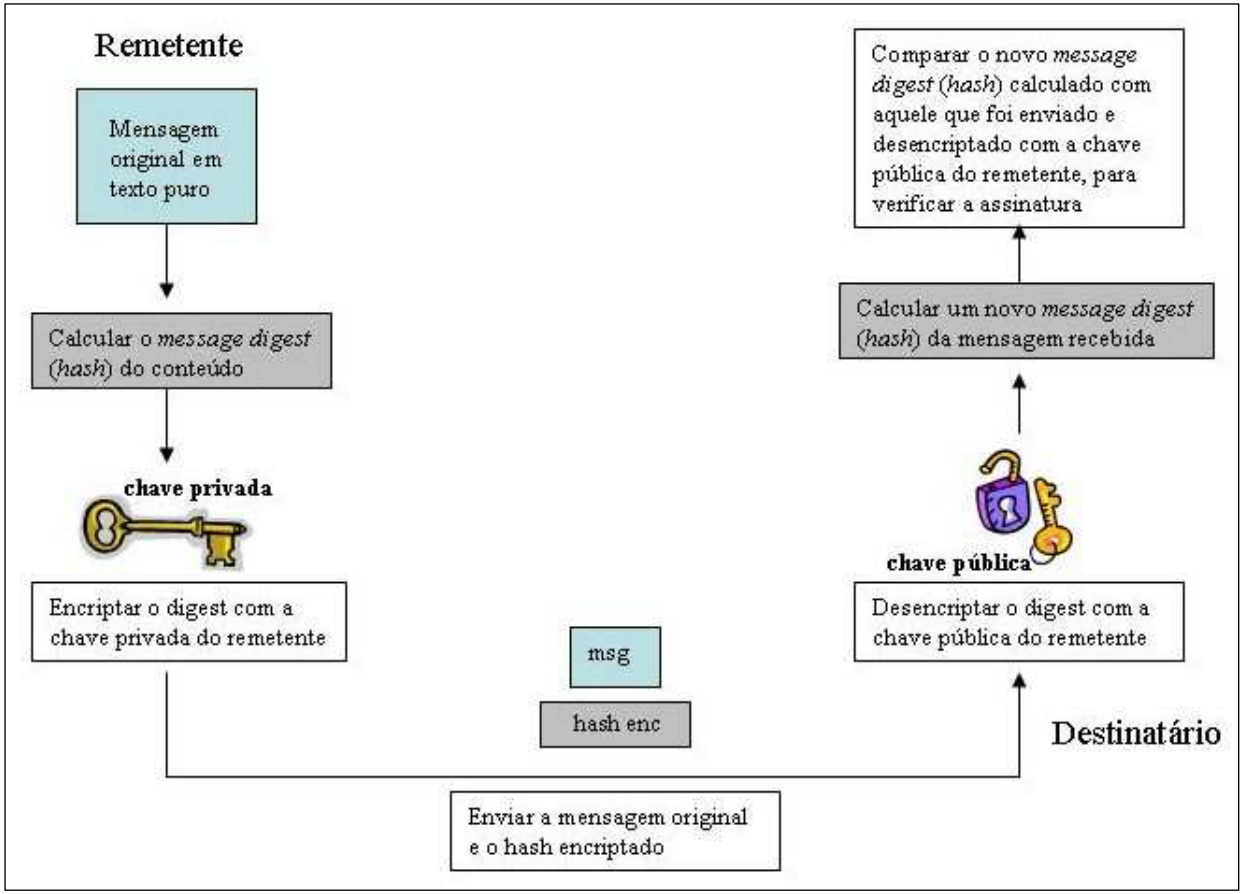
- Algoritmo de Rijndael
  - Criptografia simétrica
  - A partir de um vetor de inicialização e uma chave criptográfica, realiza transformações no texto original (em formato binário), para produzir o texto encriptado
  - O texto encriptado pode ser convertido de volta para o texto original com o uso do mesmo vetor de inicialização e mesma chave criptográfica usados para encriptá-lo

Criptografia e assinaturas digitais

- Assinatura digital
  - Funciona com base na criptografia de chaves públicas
  - Chave privada: mantida secreta; chave pública: divulgada ao público
  - Possibilita ter autenticidade e integridade dos dados que são assinados

Criptografia e assinaturas digitais

- Visão geral da assinatura digital



Criptografia e assinaturas digitais

- Assinatura digital XML
  - Da recomendação formal do W3C "*XML Signature Syntax and Processing*", de fevereiro de 2002
  - Por que surgiu essa especificação?
  - Problemas com os padrões de assinatura digital usados com XML antes desta recomendação:
    - Possuíam formato binário, com sintaxe própria
    - A aplicação que desejava assinar um documento precisava enviar, para aquela que iria verificar a assinatura, as informações sobre a assinatura digital, como por exemplo, o algoritmo utilizado

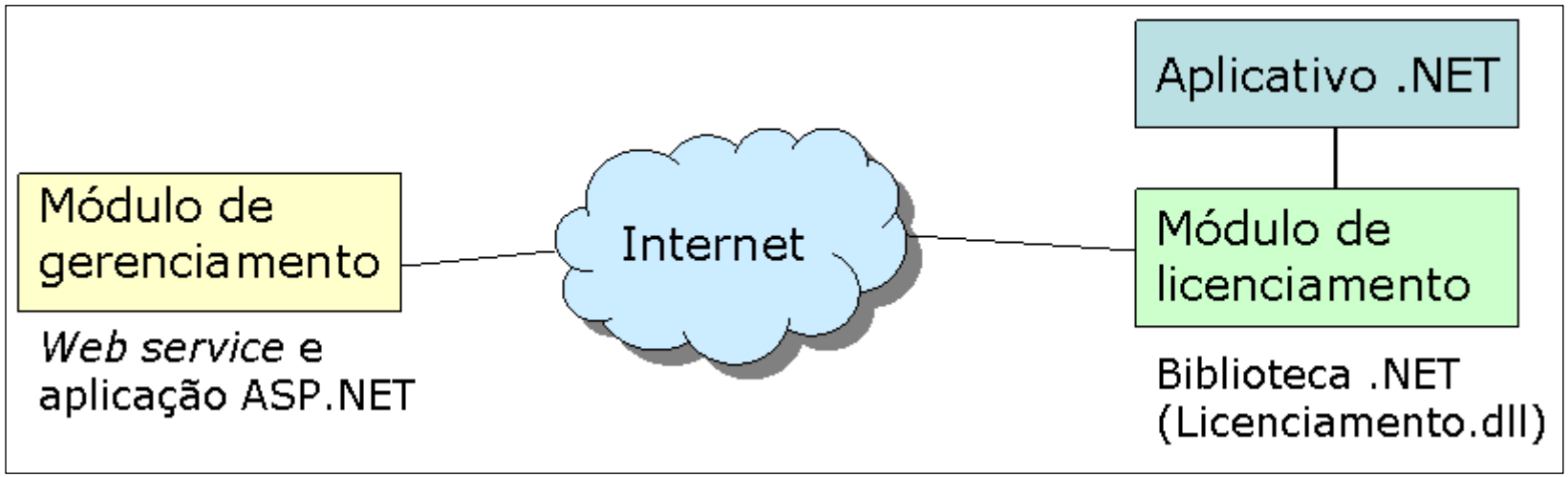
Criptografia e assinaturas digitais

- Sintaxe da assinatura digital XML

```
<?xml version="1.0" encoding="utf-8"?>
<pedido>
  <compra>
    <item qtde="1">Jornal Nacional: a Notícia Faz História</item>
    <item qtde="1">Pensar é Transgredir</item>
  </compra>
  <entrega>
    <para>Jose da Silva</para>
    <cep>89012-510</cep>
    <complemento>apto 403</complemento>
  </entrega>
  <pagamento>
    <cartao tipo="visa">1234-5678-9012-3456</cartao>
  </pagamento>
  <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
    <SignedInfo>
      <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
      <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
      <Reference URI="">
        <Transforms>
          <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
        </Transforms>
        <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
        <DigestValue>XKbVVUWzTz7eTHye7+dunlwVWAQ=</DigestValue>
      </Reference>
    </SignedInfo>
    <SignatureValue>DVIDHAB5eey9yr7s6M/YUPzCBeNAfQ5BMyOnukqRFGIeWYwFYf1FLZ7TjwSk6JpM1DV92
    +vmGaZlr8Nev3PtoYpA+ASgULI+LYTGMBWO=</SignatureValue>
  </Signature>
</pedido>
```

Desenvolvimento

Visão geral do sistema



Requisitos

- Funcionais
  - Permitir a requisição de licenças através da Internet
  - Embutir na requisição de licença informações sobre o *hardware* do *host* que a enviou
  - Permitir que seja especificado na requisição o nível de acesso e tempo de uso desejados
  - Permitir acompanhar a situação do pedido de licença
  - Permitir a transferência do arquivo de licença, quando disponível

Requisitos

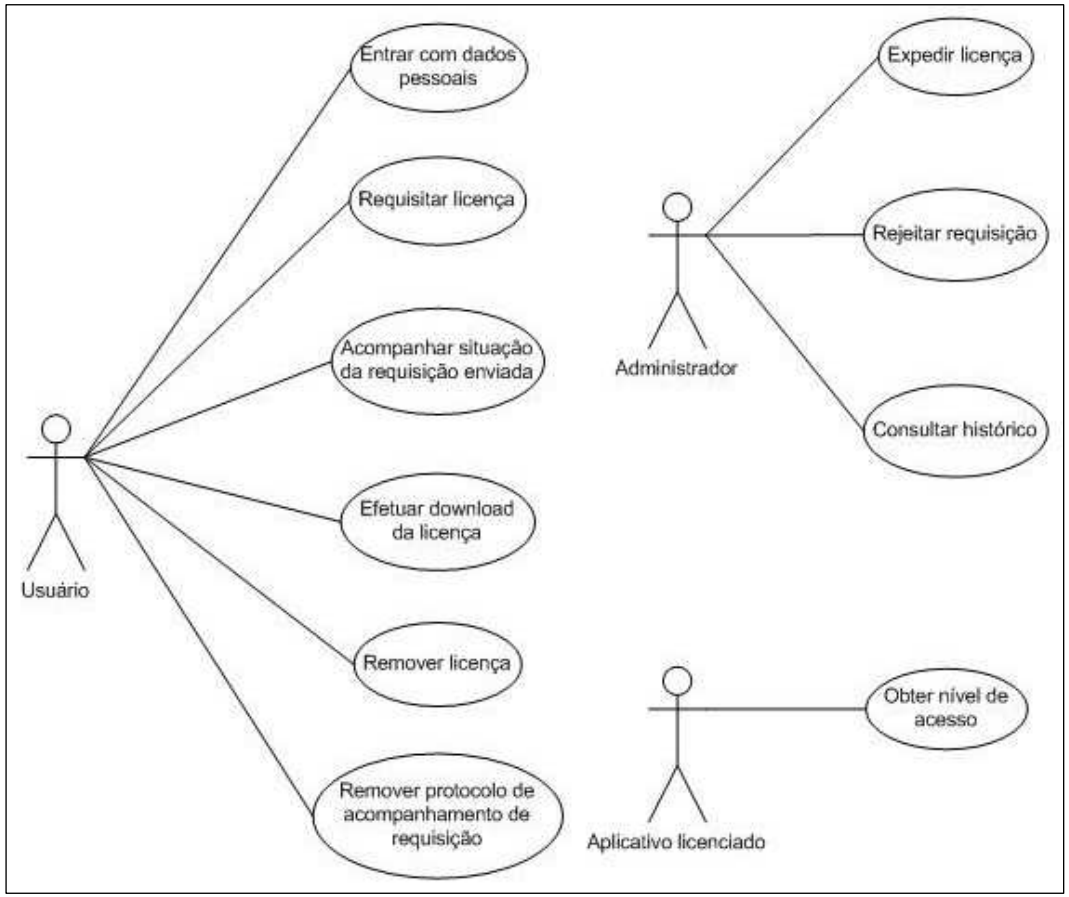
- Funcionais
  - Permitir a assinatura e rejeição dos pedidos de licença a partir da Internet
  - Enviar um *email* para o requerente, informando da disponibilidade da licença ou rejeição do pedido
  - Permitir a consulta do histórico de assinatura e solicitação de licenças
  - Verificar os arquivos de licença, no que diz respeito a autenticidade, integridade dos dados, data de expiração e o hardware onde o aplicativo está sendo executado
  - Retornar informações sobre a validade da licença e o nível de acesso que a licença autoriza para o aplicativo

Requisitos

- Não funcionais
  - Permitir uma fácil integração do sistema com aplicativos existentes da plataforma .NET
  - Gerenciar os aspectos de segurança envolvidos para que o sistema de licenciamento não seja quebrado

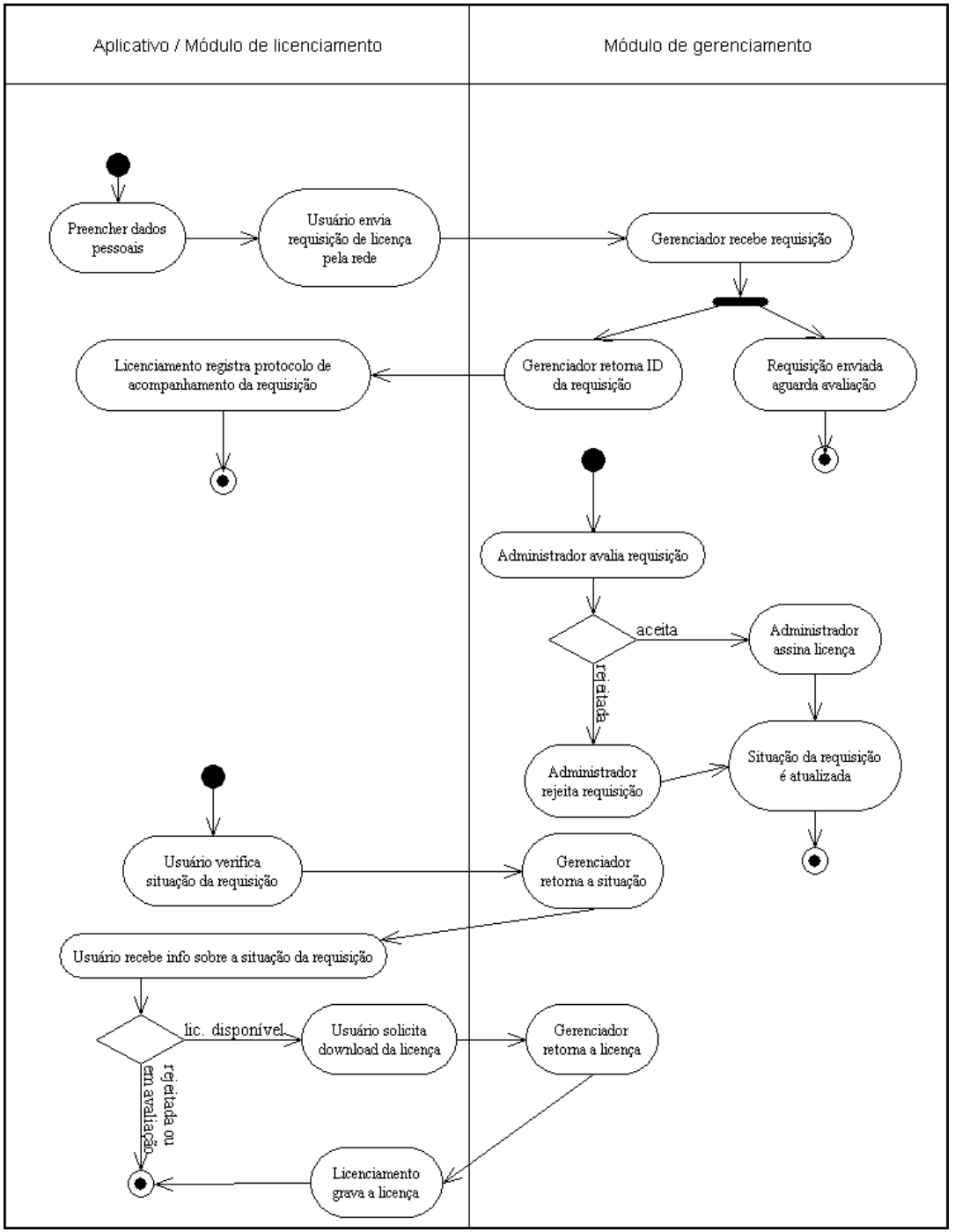
Especificação

- Casos de uso



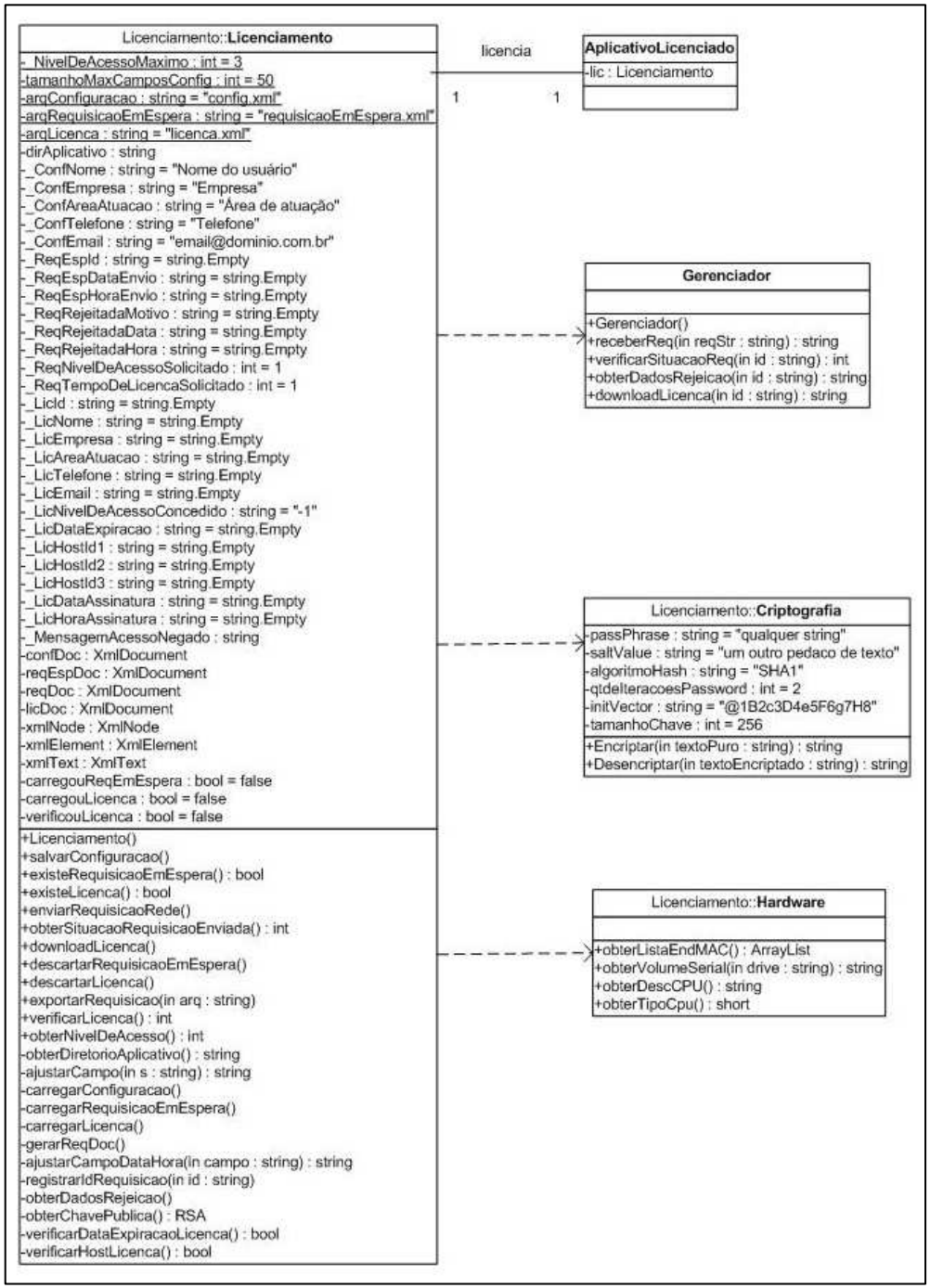
Especificação

- **Diagrama de atividades**



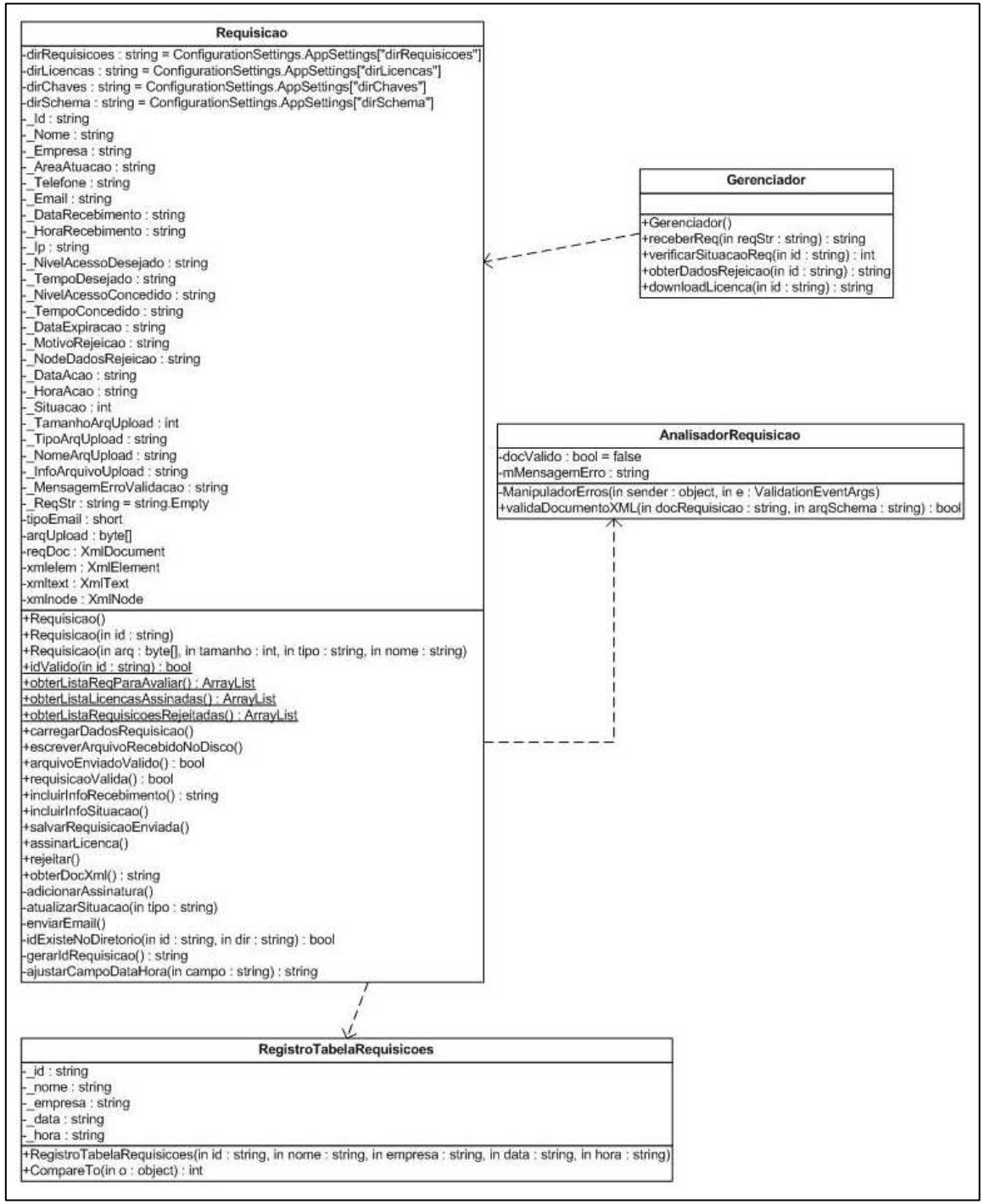
Especificação

- **Classes de análise para o módulo de licenciamento**



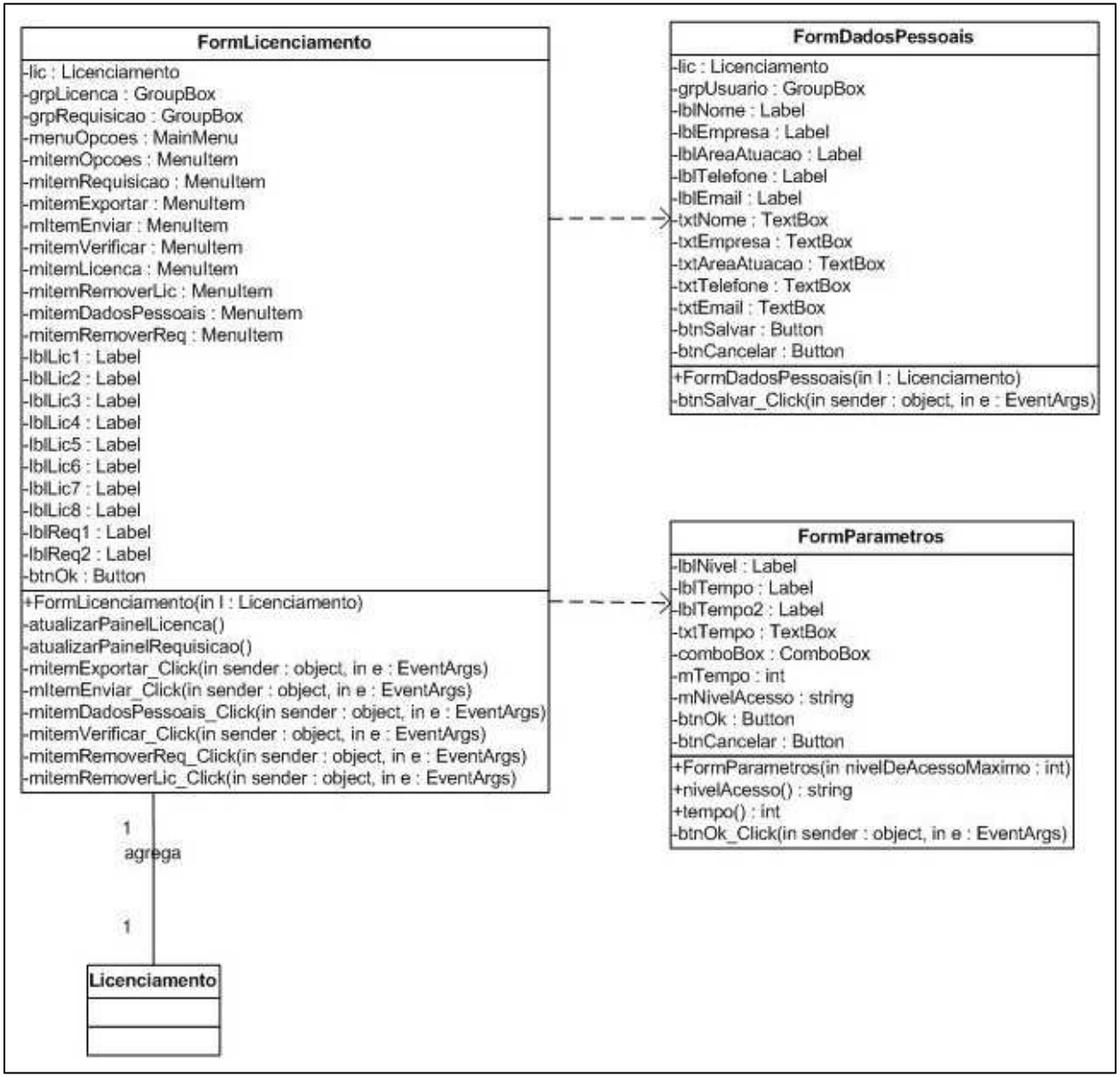
Especificação

- **Classes de análise para o módulo de gerenciamento**



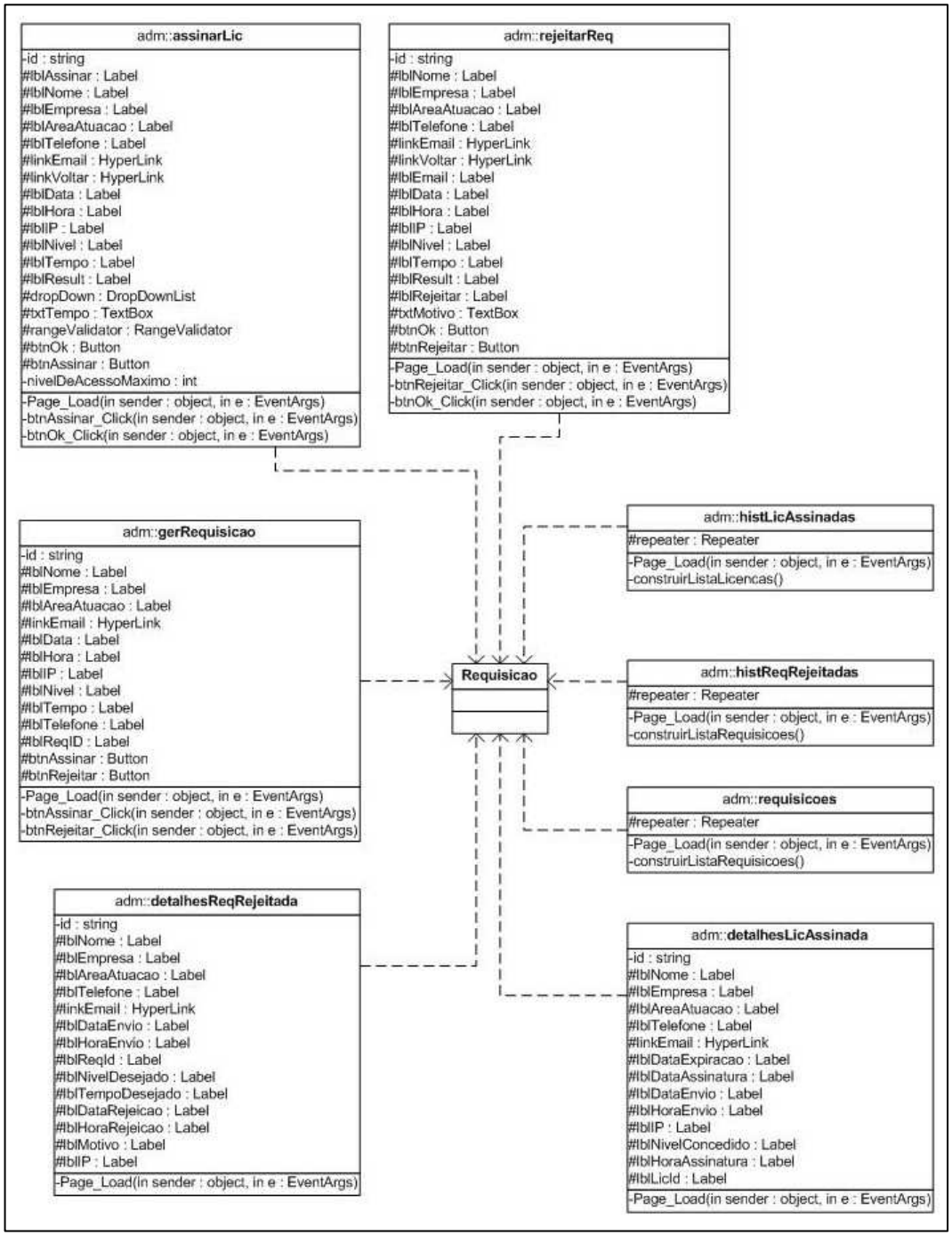
Especificação

- **Classes de projeto para o módulo de licenciamento**



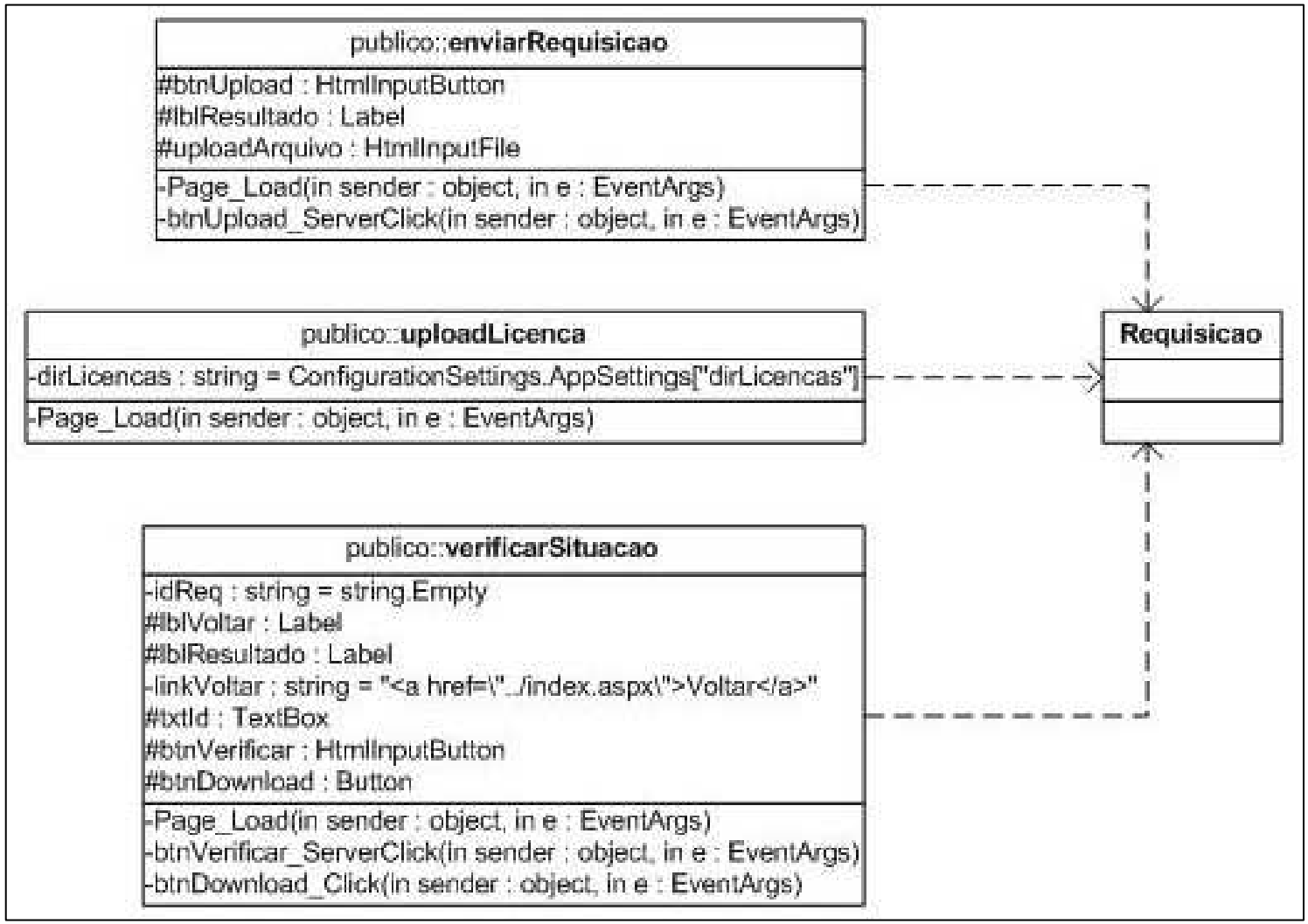
Especificação

- **Classes de projeto para o módulo de gerenciamento (administrador)**



Especificação

- **Classes de projeto para o módulo de gerenciamento (público)**



Implementação

- Ferramentas utilizadas:
  - Visual Studio .NET
  - Microsoft Visio
  - SmartDraw

Implementação

- **Documento de requisição de licença**

```
<?xml version="1.0" ?>
<requisicao>
    <requerente>
      <nome>Leonardo D'Ippolito</nome>
      <empresa>FURB</empresa>
      <areaAtuacao>Estudante</areaAtuacao>
      <telefone>555-5555</telefone>
      <email>leodippolito@terra.com.br</email>
    </requerente>
    <software>
      <nivelDeAcesso>2</nivelDeAcesso>
      <tempoDeLicenca>5</tempoDeLicenca>
    </software>
    <host>
        <id1>NxyE7xjCMXPbs25MyLo6nT231ObITvtKV3efFaOPLpE=</id1>
        <id2>FUuVS3GQZySIfvuvtsFKfA==</id2>
        <id3>hVCjOTczli7Tn7TARZEWDIriGIpdm169LLNtTcwvbzfc2kspH1jUPaFLfOheTn4</id3>
    </host>
</requisicao>
```

Implementação

- *Platform interop*

```
// chama API do Windows para obter informacoes de uma particao do HD
[DllImport("kernel32.dll")]
private static extern long GetVolumeInformation(
    string PathName,
    StringBuilder VolumeNameBuffer,
    UInt32 VolumeNameSize,
    ref UInt32 VolumeSerialNumber,
    ref UInt32 MaximumComponentLength,
    ref UInt32 FileSystemFlags,
    StringBuilder FileSystemNameBuffer,
    UInt32 FileSystemNameSize);

// ------------------------------------------------------------------------

// obtem o 'volume serial number' de uma particao do HD
public string obterVolumeSerial(string drive)
{
    uint serNum = 0;
    uint maxCompLen = 0;
    StringBuilder VolLabel = new StringBuilder(256); // label
    UInt32 VolFlags = new UInt32();
    StringBuilder FSName = new StringBuilder(256);  // file system name
    drive = drive + ":\\";

    long Ret = GetVolumeInformation(
        drive,
        VolLabel,
        (UInt32)VolLabel.Capacity,
        ref serNum,
        ref maxCompLen,
        ref VolFlags,
        FSName,
        (UInt32)FSName.Capacity);

    return Convert.ToString(serNum);
}
```

Implementação

- **Documento de requisição de licença recebido**

```
<?xml version="1.0" ?>
<requisicao>
    <requerente>
      <nome>Leonardo D'Ippolito</nome>
      <empresa>FURB</empresa>
      <areaAtuacao>Estudante</areaAtuacao>
      <telefone>555-5555</telefone>
      <email>leodippolito@terra.com.br</email>
    </requerente>
    <software>
      <nivelDeAcesso>2</nivelDeAcesso>
      <tempoDeLicenca>5</tempoDeLicenca>
    </software>
    <host>
      <id1>NxyE7xjCMXPbs25MyLo6nT23lOblTvtKV3efFaOPLpE=</id1>
      <id2>FUuVS3GQZySIfvuvtsFKfA==</id2>
      <id3>hVCjiGNtTcwvbzfc2kspH1jUPaFLfOheTNGmSn4+dDV/67GLqJyuUW9uNw==</id3>
    </host>
    <recebimento>
      <idRequisicao>W3P4S9Q7S1A3N7</idRequisicao>
      <ipChamador>127.0.0.1</ipChamador>
      <data dia="07" mes="11" ano="2004" />
      <horario horas="22" minutos="13" />
    </recebimento>
    <situacao>
      <desc>em avaliacao</desc>
      <data />
      <horario />
    </situacao>
</requisicao>
```

Implementação

- **Assinatura de licença**

```
private void adicionarAssinatura()
{
    // instancia objeto do algoritmo RSA
    RSA chave = RSA.Create();

    // obtem a chave privada do disco
    StreamReader sr = System.IO.File.OpenText(dirChaves + "chavePrivada.xml");
    chave.FromXmlString(sr.ReadToEnd());

    // cria uma assinatura
    SignedXml assinatura = new SignedXml(reqDoc);

    // define a chave para a assinatura
    assinatura.SigningKey = chave;

    // define o metodo de canonicalizacao
    assinatura.SignedInfo.CanonicalizationMethod =
        SignedXml.XmlDsigCanonicalizationWithCommentsUrl;

    // instancia uma nova referencia para a assinatura
    Reference r = new Reference("");

    // define um transform para DsigEnvelopedSignature
    r.AddTransform(new XmlDsigEnvelopedSignatureTransform(false));

    // adiciona a referencia no documento de assinatura
    assinatura.AddReference(r);

    // calcula a assinatura digital
    assinatura.ComputeSignature();

    // anexa a assinatura no documento original de requisicao
    XmlElement elemAssina = assinatura.GetXml();
    reqDoc.DocumentElement.AppendChild(elemAssina);
}
```

Implementação

- Licença assinada

```
<?xml version="1.0" ?>
<requisicao>
    <requerente>
      <nome>Samantha</nome>
      <empresa>Casa</empresa>
      <areaAtuacao>Psicologia</areaAtuacao>
      <telefone>3203545</telefone>
      <email>samsabel@terra.com.br</email>
    </requerente>
    <software>
      <nivelDeAcessoConcedido>2</nivelDeAcessoConcedido>
      <dataDeExpiracao dia="26" mes="12" ano="2004" />
    </software>
    <host>
      <id1>vUAHpI3dA6jt/BGKA+o4jA==</id1>
      <id2>zhphTArDcxP3oLU3ALBw1Q==</id2>
      <id3>+Yhvg3BMgRmFJz6hiCIBHRMdrMZfnUaa/qHwAtShUaAUF5BUwt7w==</id3>
    </host>
    <recebimento>
      <idRequisicao>Q8T3R6D2R2C2M7</idRequisicao>
      <ipChamador>192.168.0.138</ipChamador>
      <data dia="13" mes="11" ano="2004" />
      <horario horas="13" minutos="51" />
    </recebimento>
    <situacao>
      <desc>licenca assinada</desc>
      <data dia="15" mes="11" ano="2004" />
      <horario horas="16" minutos="20" />
    </situacao>
    <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
        <SignedInfo>
            <CanonicalizationMethod
                Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments" />
            <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
            <Reference URI="">
                <Transforms>
                    <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
                </Transforms>
                <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
                <DigestValue>yupwcuqsNmL21GDgPGQ/NkOaAIA=</DigestValue>
            </Reference>
        </SignedInfo>
        <SignatureValue>L17C5uqhgm8aMnpj3Eg5A7OcSOkwmhoiI8+OhyF9tLtR8vBC5DuQ4n/BMg=</SignatureValue>
    </Signature>
</requisicao>
```

Implementação

- Verificação da assinatura

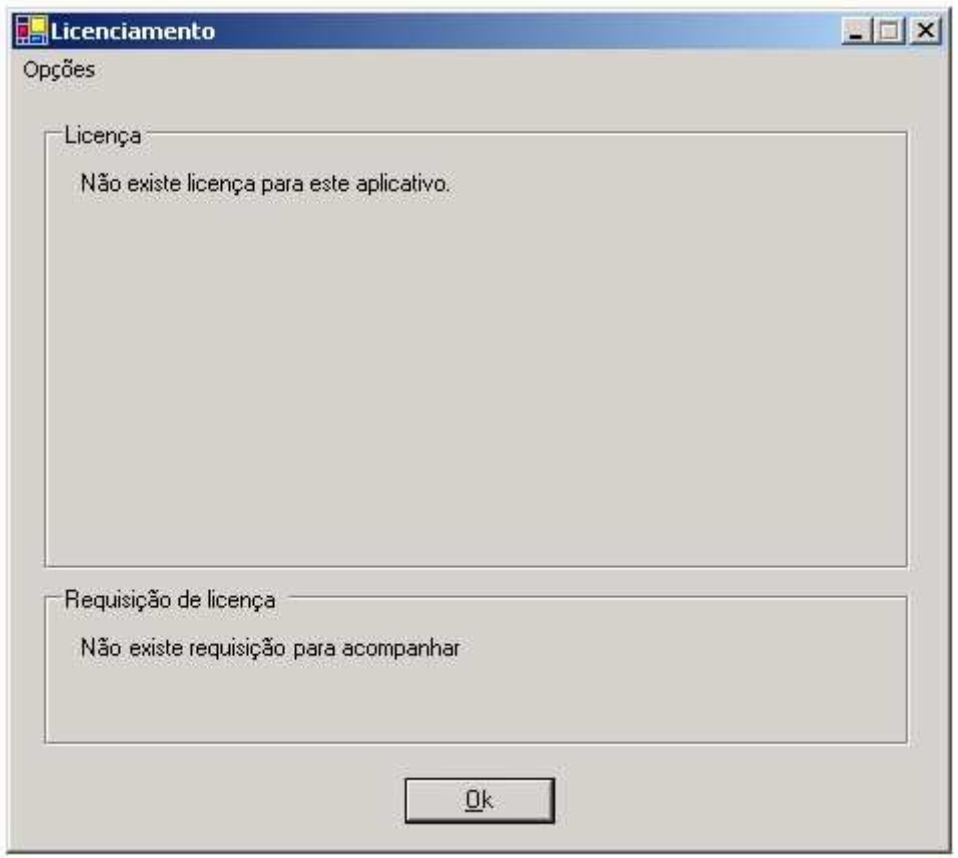
```
// Possíveis retornos:
// 0 - "licenca expirou"      1 - "licenca ok"
// 2 - "licenca invalida"     3 - "nao existe licenca"
// 4 - "host errado"

public int verificarLicenca()
{
    if(existeLicenca() == false) {return 3;}
    else
    {
        if(carregouLicenca == false) {carregarLicenca();}

        SignedXml assinatura = new SignedXml(licDoc);

        XmlNode nodeAssinatura = licDoc.GetElementsByTagName("Signature",
            SignedXml.XmlDsigNamespaceUrl)[0];

        assinatura.LoadXml((XmlElement) nodeAssinatura);
        RSA chave = obterChavePublica();

        if (assinatura.CheckSignature(chave) == false)
        {
            return 2;
        }
        else
        {
            // licenca existe e assinatura esta OK
            if(verificarHostLicenca() == false) {return 4;}
            else
            {
               if(verificarDataExpiracaoLicenca() == false) {return 0;}
               else {return 1;}
            }
        }
    }
}
```

Operacionalidade da implementação

- Aplicativo .NET

Operacionalidade da implementação

- Menu de licenciamento



Operacionalidade da implementação

- Módulo de gerenciamento

Resultados e discussão

- Atendeu a todos os requisitos funcionais
- Reconheceu corretamente licenças válidas e inválidas
- Fez a verificação correta do hardware que executa o aplicativo
- Atendeu ao requisito não funcional de facilidade de integração

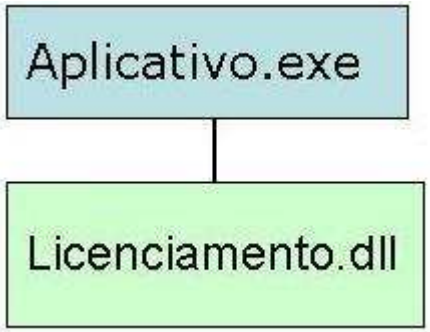
Resultados e discussão

- Vulnerabilidade identificada
  - Sistema escrito em .NET, de código gerenciado (MSIL)
  - Possibilidade de descompilação e modificação do *assembly* .NET (Licenciamento.dll) através de ferramentas como ILDASM e ILASM
  - Identificar o módulo de licenciamento com um *strong name* (assinatura digital no *assembly*) evita que ele seja trocado por uma versão falsa
  - Porém, é essencial que o aplicativo licenciado seja também protegido. Caso contrário, ele pode ser descompilado e recompilado para funcionar sem fazer verificações de licença
  - Técnicas para proteger o aplicativo: *obfuscation* e compilação parcialmente nativa (Visual C++ .NET)

Resultados e discussão

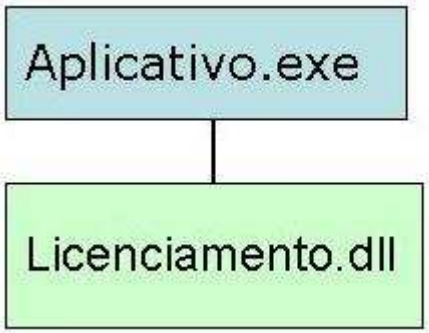
- Proteção dos *assemblies*

Sem proteção



Reconhece a biblioteca pelo nome

Com proteção



Obfuscation e compilação para código nativo

Strong name

Resultados e discussão

- Troca do Licenciamento.dll, assinado com *strong name,* por uma versão falsa

Resultados e discussão

- Vulnerabilidade da data de expiração
  - Como evitar que o usuário atrase o calendário do sistema operacional, para que sua licença não expire?

Conclusões

Conclusões

- Sistema de licenciamento de bom nível de segurança, quando considerada a proteção dos *assemblies*
- Equivalente em funcionalidade aos sistemas comerciais observados nos trabalhos correlatos
- Integra diferentes tecnologias
- A segurança com assinatura digital XML pode ser aplicada aos diferentes sistemas que trabalham com XML na Internet

Sugestões de extensões

- Desenvolver um módulo para gerenciar uma fila de utilização de licenças. A empresa tem 50 usuários mas quer obter apenas 10 licenças para o aplicativo. Esse módulo faria o gerenciamento de utilização simultânea
- Adicionar um componente de *hardware* para fornecer uma camada adicional de segurança
- Elaborar técnicas de *obfuscation* e proteção de *assembly* para impedir que eles sejam modificados
- Desenvolver um módulo para alertar e fazer o *download* de atualizações do aplicativo de forma segura
- Adaptar o sistema para outras plataformas, como Java ou Win32 nativo

Apresentação do protótipo