**Projeto de Segurança Informática**

**Trabalho realizado por Grupo 10:**

Tiago Rodrigues 84040

João Mendes 86093

**Licenciatura em Tecnologias da Informação**

**1º Semestre**

**2018/2019**

# Resumo

O objetivo deste projeto é criar um programa que depois de iniciado durante o registo emita um pedido de licença com as informações do utilizador e da maquina onde se encontra, esse pedido de deverá ser autenticado e enviado novamente para o programa que a partir desse momento vai permitir que o utilizador utilize o tal.

# Estrutura do projeto

O projeto foi dividido em duas partes, *user* e *owner*, sendo o *user* o utilizador do programa e o *owner* o dono que irá emitir as licenças.

## User

|  |  |
| --- | --- |
| Classe | Tarefa |
| Criptografia | Este método está encarregue de criar uma chave simétrica com o algoritmo AES, encriptar e desencriptar com essa chave. |
| GenerateKeys | Este método cria um par de chaves publica/privada, escreve e lê de um ficheiro. |
| Identification | Este método inicializa o programa atribuindo-lhe um id e uma versão. Com os respetivos getters e setters |
| MachineInfo | Este método adquire as informações necessárias para a identificação da máquina, sendo estas o número da Bios e o UUID (identificador único universal), este método retorna também o Unix time |
| Main | Este método corre os métodos de todas as classes. |
| ReadKeys | Este método permite ler as chaves publica e privada, e também encriptar e desencriptar com as tais. |
| CartaoCidadao | Este método recolhe toda a informação relevante ao cartão do cidadão, tais como os providers, a chave publica o certificado, |

## Owner

|  |  |
| --- | --- |
| Classe | Tarefa |
| GenerateKeys | Este método cria um par de chaves publica/privada, escreve e lê de um ficheiro. |
| Main | Este método corre os métodos de todas as classes. |
| VerifyRequest | Verifica o certificado e a assinatura. |
| SendAutentication | Envia encripta a informação e envia a autenticação para o programa. |

# Problemas e Soluções

# Conclusão

**Aplicação**

void init(string nomeDaApp, string versão): Construtor e vai inicializar o isRegistered.

bool isRegistered(): Vai verificar se existe uma licença e a sua autenticidade.

bool startRegistration(): Ca so não exista uma licença, vai iniciar um registo que vai pedir as informações do Pc e CC, e enviar um pedido de licença a biblioteca.

void showLicenseInfo(): Mostrar informações da licença.

**Biblioteca**

Void init()

Void receberPedido(): Verificar a legitimidade do pedido

Void emitirLicença()

**Ficheiro de licença**

Informações que identifiquem o utilizador

* Nome
* Endereço de email
* NIC (número de identificação civil)
* PK+ do cartão do cidadão

Informações que identifiquem o sistema

* Número e tipo de cpu
* Placas de rede (endereço MAC)
* Número de series dos discos
* Identificadores da Bios

Informação que identifique a aplicação

* Nome da aplicação
* Versão atual
* Valor da síntese do ficheiro principal?

Identificação que identifique o intervalo temporal

* Data de início
  + Data de expiração

**Segurança**

Proteção

Integridade:

Confidencialidade: Encriptação com a chave publica do destinatário

Autenticação:

Não repudiação: Saber quem fez o pedido (assinatura)

PasswordEncryption – pesquisar por pbkdf2

Problemas com o tamanho da password

https://www.oracle.com/technetwork/java/javase/downloads/jce8-download-2133166.html