Faculdades Integradas PROMOVE
DE BRASÍLIA

CURSO DE TECNOLOGIA EM SEGURANÇA DA INFORMAÇÃO

ISRAELL BARRETO PACCAMICIO

APLICATIVO DE CRIPTOGRAFIA SIMÉTRICA ACEcrypt

Brasília-DF

2015

Israell Barreto Paccamicio

Aplicativo de Criptografia Simétrica ACEcrypt

Trabalho de Conclusão de Curso apresentada à Faculdades Integradas Promove de Brasília como requisito parcial para obtenção do título de Tecnólogo em Segurança da Informação.

Orientador: Prof. MSc. Cid Bendahan Coelho Cintra

Aprovado em __/__/2015

Brasília-DF

2015

Israell Barreto Paccamicio

Aplicativo de Criptografia Simétrica ACEcrypt

Trabalho de Conclusão de Curso apresentada à Faculdades Integradas Promove de Brasília como requisito parcial para obtenção do título de Tecnólogo em Segurança da Informação.

Orientador: Prof. MSc. Cid Bendahan Coelho Cintra

Aprovado em __/__/2015

Aprovado por :

_____

Prof. Avaliador.

_____

Prof. Avaliador.

Brasília-DF

2015

AGRADECIMENTOS

Agradeço ao criador do universo por gerar toda a existência, a minha avó Tereza, grande mãe por sempre estar comigo, ao meu pai Oswaldo, pelos sábios conselhos, aos meus poucos amigos, por sempre trilharem a estrada da vida comigo, ao meu longínquo amor pela força dada e ao meu orientador pela paciência e eficiência para a conclusão deste trabalho.

DEDICATÓRIA

“A comunidade que sempre me ajudou mesmo sem dizer uma palavra.”

EPÍGRAFE

“Levante-se. Se você ainda tem tempo para pensar em uma bela morte, porque não pensa em como viver belamente até o fim? ”

Sakata Gintoki

RESUMO

Este trabalho tem como foco o desenvolvimento de um aplicativo que será utilizado diretamente para a segurança de arquivos e pastas no sistema operacional Windows. O aplicativo em questão leva em consideração a confidencialidade e sigilo de dados privados e é composto para suprir as necessidades básicas de um usuário que necessita armazenar arquivos ou pastas com mais segurança. Após diversas pesquisas, foram utilizados métodos muito bem reconhecidos no mercado para a criação do aplicativo, como o PRISM e a linguagem C#. Por fim, o arquivo gerado pelo aplicativo pode ser usado como um arquivo seguro.

Palavras chaves: criptografia, segurança da informação, aplicativo.

ABSTRACT

This work focuses on the development of an application that will be used directly for the safety of files and folders in Windows operating system. The application in question takes into consideration the confidentiality and secrecy of private data and is made to meet the basic needs of a user who needs to store files or folders with more security. After various studies, well recognized in the market methods were used for creating the application, as the PRISM and the C # language. Finally, the file generated by the application can be used as a secure file.

Keywords: cryptography, information security, application.

GLOSSÁRIO DE TERMOS E SIGLAS

Aplicação - Programas ou ferramentas que estão instalados no computador.

C# - Linguagem de programação usada para desenvolvimento de aplicativos.

Criptografia - Metodologia matemática para cifrar dados.

Hardware - Partes físicas do computador.

PRISM - *Practical Software Development Model* (Metodologia de desenvolvimento de software).

UML - Sigla que significa *Unified Modeling Language (*Linguagem de Modelagem Unificada).

Visual Studio - Ferramenta proprietária, porém gratuita para desenvolvimento de aplicações.

Lista de Ilustrações

Lista de Quadros

SUMÁRIO

CAPÍTULO I - APRESENTAÇÃO

1.1. Introdução

Desde a mais antiga das civilizações, tudo que possui valor necessita de proteção. Isso se deve ao fato de qualquer um desses ativos estar sujeito a ataques de agentes externos. A proteção desses ativos torna-se característica essencial para o cumprimento de objetivos finais.

Hoje, a computação é o maior e mais ágil meio para transmissão e armazenamento de dados. A junção desses dados transmitidos ou armazenados podem gerar todo tipo de ativo digital, sejam eles textuais, visuais ou sonoros.

A transmissão de dados via sistemas digitais é de escala mundial. Devido ao advento da Internet e a liberdade da mesma, qualquer tipo de agente externo pode ter contato com o caminho por onde percorrem os dados trazendo assim diversos riscos aos mesmos.

Os riscos gerados a esses ativos são: acesso indevido a informações confidenciais, quebra de autenticidade, exclusão, comércio, exposição e diversos outros. Por isso, assim como qualquer ativo físico, os dados digitais necessitam de proteção.

Para ajudar a resolver esse tipo de problema existe a criptografia. A criptografia pode ser definida como o estudo de técnicas que embaralham e tornam os dados ilegíveis. Utilizada por milênios desde as antigas civilizações, hoje as técnicas de criptografia estão aprimoradas e adaptadas para o mundo digital, sendo consideravelmente mais segura e eficaz.

Perante este contexto, este trabalho tem como objetivo apresentar a análise e o desenvolvimento de um aplicativo que aplica a técnica de criptografia simétrica, fazendo com que todo tipo de dado passado por ele seja ilegível aos agentes externos garantindo assim a confidencialidade dos dados submetidos a aplicação.

1.2. Justificativa

Para proteger o usuário de acesso indevido a arquivos e pastas com informações sigilosas é necessário um aplicativo com a capacidade de transformar esses arquivos e pastas em dados ilegíveis a agentes externos. Assegurando assim a confidencialidade e segurança do mesmo.

1.3. Objetivos

1.3.1. Objetivo Geral

Desenvolvimento de um aplicativo para realizar a técnica de criptografia simétrica em arquivos e pastas.

1.3.2. Objetivos Específicos

- Pesquisar sobre aplicativos de segurnaça.
- Pesquisar sobre o desenvolvimento de aplicativos.
- Elaborar a documentação necessária para o aplicativo.
- Desenvolver e homologar o protótipo do aplicativo proposto.

1.4. Metodologia

Foi realizada uma longa pesquisa e por meio dela foram levantados todos os materiais bibliográficos necessários para explicar, analisar e desenvolver o aplicativo, os principais temas a cerca da pesquisa foram, criptografia simétrica, aplicações e linguagem C#. O desenvolvimento do aplicativo foi realizado a partir da viabilidade encontrada na pesquisa.

CAPÍTULO II - REFERENCIAL TEÓRICO

2.1. Informação

A informação é um elemento intangível e essencial para a vida do ser humano e de instituições.

Aurélio (2015), apresenta uma visão do que vem a ser a informação e a define como:

> [...] um conjunto organizado de dados, que constitui uma mensagem sobre um determinado fenômeno ou evento. A informação permite resolver problemas e tomar decisões, tendo em conta que o seu uso racional é a base do conhecimento.

Ferreira (2003, p. 2), afirma que a " informação se tornou ponto crucial para a sobrevivência das organizações. "

A norma ABNT NBR ISO/IEC 27002 (2005, p. 10.), define que a informação pode ser "impressa ou escrita em papel, armazenada eletronicamente, transmitida pelo correio ou por meios eletrônicos, apresentada em filmes ou falada em conversas".

Verifica-se, portanto, que a informação é um item extremamente necessário na vida pessoal e em todos os campos de uma organização.

2.2. Segurança da Informação

É um conceito aplicado diretamente a proteção dos dados, principais elementos para a produção de informações pessoais ou institucionais.

Lyra (2008, p. 3 e 4) define para a segurança da informação os elementos que são necessários para considerar que as informações possuem segurança, sendo eles:

> **Confidencialidade:** capacidade de um sistema de permitir que alguns usuários acessem determinadas informações ao mesmo tempo em que impede que outros, não autorizados, a vejam.
>
> **Integridade:** a informação deve estar correta, ser verdadeira e não estar corrompida.
>
> **Disponibilidade:** a informação deve estar disponível para todos que precisarem dela para a realização dos objetivos [...]
>
> **Autenticação:** garantir que um usuário é de fato quem alega ser.
>
> **Legalidade:** garantir que o sistema esteja aderente à legislação pertinente.
>
> **Privacidade:** capacidade de um sistema de manter anônimo um usuário e suas ações[...]
>
> **Auditoria:** capacidade do sistema de auditar tudo o que foi realizado pelos usuários, detectando fraudes ou tentativas de ataque.

2.3. Orientação a Objetos

Eddy, et. al (1994, p. 2) afirma que na orientação a objetos “o software é organizado como uma coleção de objetos separados que incorporam tanto a estrutura quanto o comportamento dos dados. ”

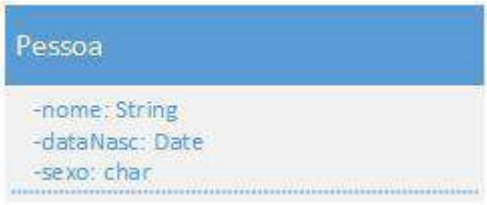
Furlan (1998 apud Monturil, 2014, p. 18) explica que essa forma de análise proporciona a modularidade de seus elementos podendo-se tomar um subconjunto existente e integrá-lo de maneira diferente em outra parte do sistema.

A análise orientada a objetos pode ser definida com os elementos e classificações a seguir:

Classe: uma classe é uma definição, um modelo, que descreve como criar uma representação precisa de um tipo de objeto específico. Cada objeto é instanciado

(criado, tornando real) pelo uso da definição de classe como um modelo. (DIAS, 2014).

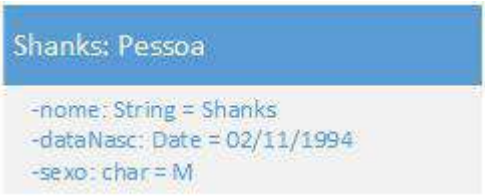
Figura 1 – Classe em notação UML.



Fonte: O autor.

Objeto: qualquer coisa que você possa descrever pode ser representada como um objeto, e essa representação pode ser criada, manipulada e destruída para representar como usar o objeto real que ela modela. (DIAS, 2014).

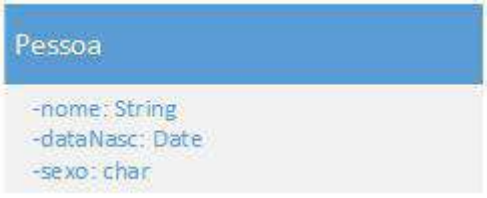
Figura 2 – Objeto em notação UML.



Fonte: O autor.

Atributos: são as propriedades de um objeto, também são conhecidos como variáveis ou campos. Essas propriedades definem o estado de um objeto, fazendo com que esses valores possam sofrer alterações. (PALMEIRA, 2013).

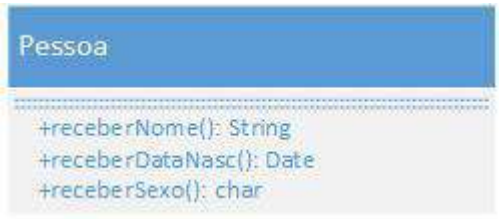
Figura 3 – Atributos em notação UML.



Fonte: O autor.

Métodos: são as ações ou procedimentos, onde podem interagir e se comunicarem com outros objetos. A execução dessas ações se dá através de mensagens, tendo como função o envio de uma solicitação ao objeto para que seja efetuada a rotina desejada. (PALMEIRA, 2013).

Figura 4 – Métodos em notação UML.



Fonte: O autor.

2.3.1. The Unified Modeling Language (UML)

A *Unified Modeling Language (UML)* hoje, é o principal padrão para documentação e modelagem de um sistema orientado a objetos.

A UML é uma linguagem gráfica rica de recursos. Não sendo uma linguagem de programação, a UML é um padrão para entender o software, fornecendo desenhos ou imagens com textos para o entendimento de todos. (DEITEL, 2005).

Fowler (2005, p. 8) explica que:

> "UML é uma família de notações gráficas, apoiada por um metamodelo único, que ajuda na descrição e no projeto de sistemas de software, particularmente daqueles construídos utilizando o estilo orientado a objetos (OO). "

Fowler (2005, p. 13) também informa que a "UML foi criada a partir de várias outras linguagens gráficas de modelagem OO que floresceram no final dos anos oitenta e que sua aparição foi feita em 1997. "

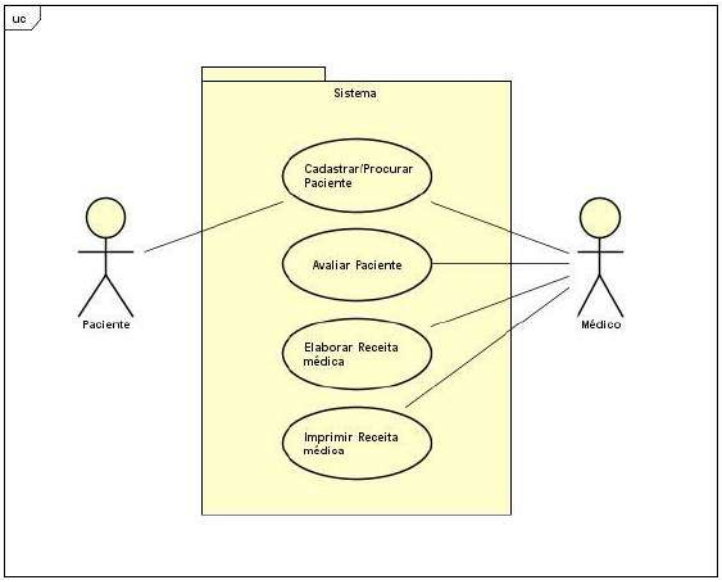
2.3.1. Os Diagramas da UML

A UML 2.0 possui 13 diagramas oficiais sendo utilizados de acordo com cada necessidade. Para as necessidades do sistema proposto os diagramas de OO a serem utilizados são: **Diagrama de Casos de Uso**, **Diagrama de Classes**, **Diagrama de Sequência e Diagrama de estados**. (FOWLER, 2005, p. 33). Abaixo segue a definição de cada um deles.

2.3.1.1. Diagrama de Casos de Uso

Deboni (1995, p. 71) explica que o diagrama de casos de uso é o ato de "Descrever um modelo funcional de alto nível do sistema ou projeto. Esse diagrama procura identificar os usuários e representar o sistema segundo a sua visão. " E cita que Jacobson (1992) afirma que um conjunto de descrições de casos de uso deve especificar completamente a funcionalidade do sistema, assim os desenvolvedores devem procurar junto aos usuários de cada subsistema formar este conjunto de casos de uso.

Figura 5 – Diagrama de Caso de Uso



Fonte: O autor.

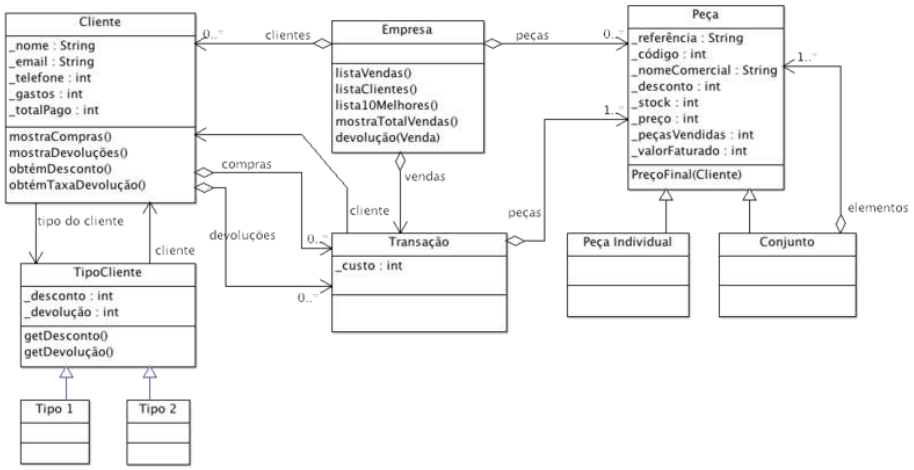
2.3.1.2. Diagrama de Classe

Sobre o diagrama de classes Deboni (1995, p. 92) define que “Classes são matrizes de objetos, elas identificam grupos de elementos do sistema que compartilham as mesmas propriedades. ”

Deboni (1995, p. 92) também explica que:

> “UML incorporou a representação de classes utilizada na OMT de Rumbaugh (Rumbaugh et al., 1994) com pequenas alterações na notação. As classes são identificadas por retângulos divididos horizontalmente em três porções, como mostra a figura. No terço superior encontra-se o nome da classe, no terço médio a lista de atributos e no terço inferior a lista de operações que esta classe pode realizar. ”

Figura 6 – Diagrama de Classes



Fonte: INESC[1]

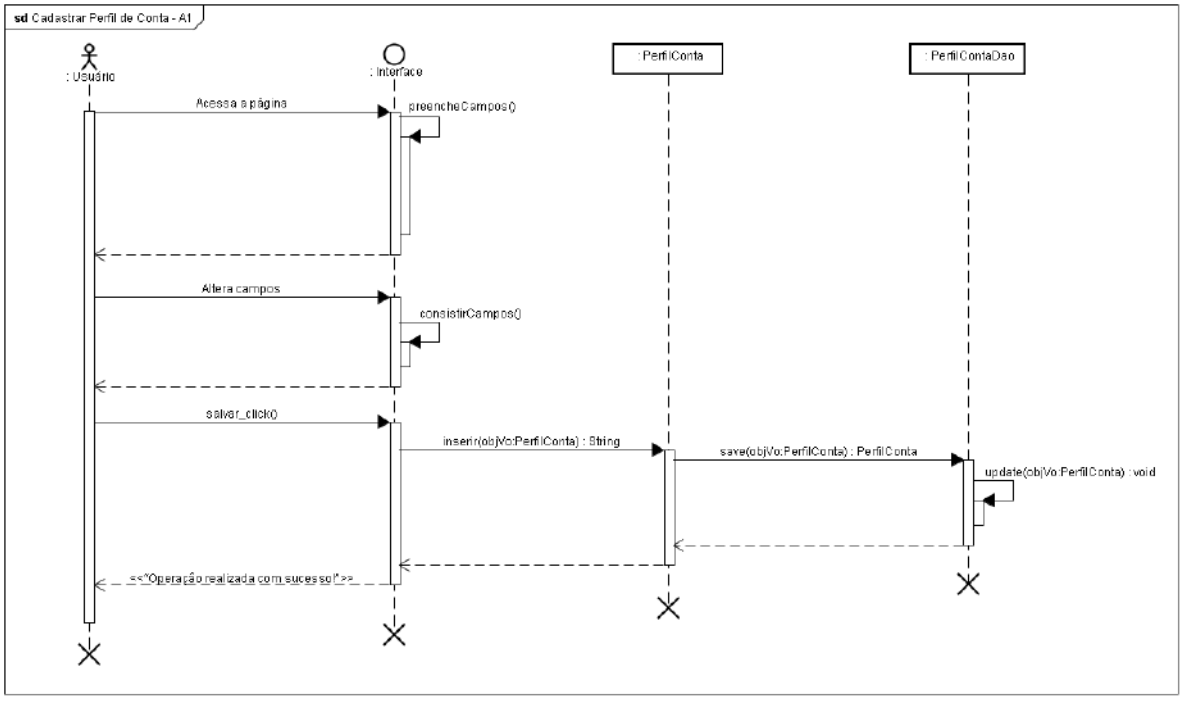
2.3.1.3. Diagrama de Sequência

[1] Disponível em: <https://www.l2f.inesc-id.pt/~david/w/pt/Introdu%C3%A7%C3%A3o_%C3%A0_Modela%C3%A7%C3%A3o_com_UML/Empresa_de_Mobili%C3%A1rio > Acesso: 10/11/2015.

Nos diagramas de sequência Deboni (1995, p. 163), diz que:

> “Os diagramas de sequência descrevem o comportamento dos objetos do sistema, que se relacionam pela troca de mensagens em interações sequenciais no tempo. Cada diagrama mostra um cenário, isto é, um conjunto de mensagens, ordenadas no tempo, com um determinado objetivo. ”

Figura 7 – Diagrama de Sequência



Fonte: Google Code[2]

2.3.1.4. Diagrama de Estados

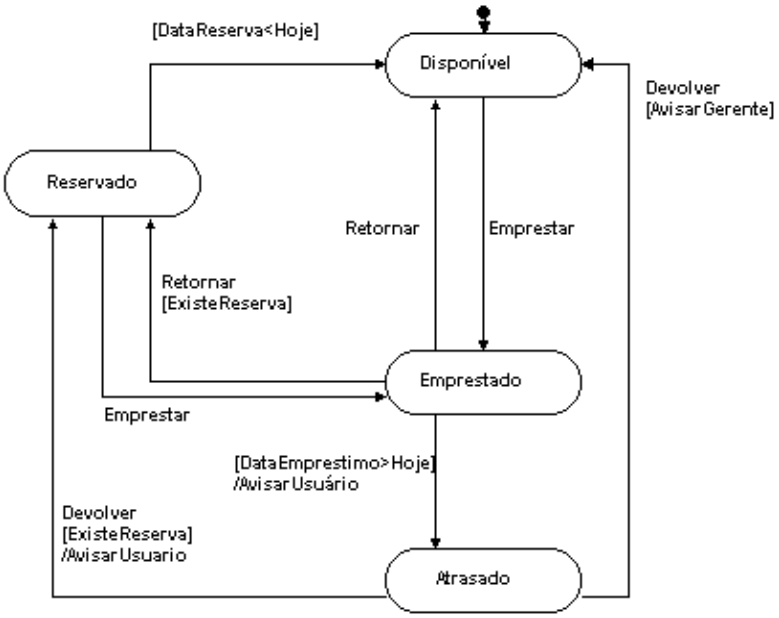
Sobre os diagramas de estados Deboni (1995, p. 168), diz que:

> “A descrição da dinâmica interna de uma classe é feita pelo diagrama de estados, ele reflete o ciclo de vida dos objetos da classe, desde o momento em que este objeto é criado até o seu término quando ele não é mais utilizado

[2] Disponível em: < http://contasnet.googlecode.com/svn/trunk/_docs/documentacao-logica/diagrama-de_classes%20e%20de%20sequencia/png/Diagramas%20de%20Sequencia/Diagramas%20de%20Sequencia/Cadastrar%20Perfil%20de%20Conta/ > Acesso: 10/11/2015.

pelo sistema, passando por todas as situações em que o objeto poderia se encontrar. ”

Figura 8 – Diagrama de Estados



Fonte: Kleber Mota[3]

A diversidade que os padrões da UML proporcionam para documentar junto a todos os tipos de diagramas demonstram que a UML garante especificações com alta organização mesmo em sistemas com arquiteturas muito complexas.

2.4. Processo de Desenvolvimento de Software

O processo de desenvolvimento de um software tornou-se ao longo do tempo um processo colaborativo, interativo e dinâmico, Pressman (2011, p. 38) afirma que:

> “Quando uma aplicação ou um sistema embutido estão para ser desenvolvidos, muitas vozes devem ser ouvidas. E, algumas vezes, parece que cada uma delas possui uma ideia ligeiramente diferente de quais funções ou recursos o software deve oferecer. Depreende-se, portanto, que se deve

[3] Disponível em: < http://www.klebermota.eti.br/2011/11/22/uml-unified-modeling-language-a-linguagem-unificada-de-modelagem/> Acesso 10/11/2015.

> fazer um esforço concentrado para compreender o problema antes de desenvolver uma solução de software. ”

Pressman (2011, P. 39) afirma que o “software, em todas suas formas e em todos os seus campos de aplicação, deve passar pelos processos de engenharia. ”

Processo é um conjunto de atividades, ações e tarefas realizadas com o propósito de criar algum produto de trabalho. Entretanto um processo no contexto de engenharia do software não é uma prescrição rígida sobre como desenvolver um software. É uma abordagem adaptável que possibilita a equipe responsável pelo desenvolvimento escolher o conjunto apropriado de ações e tarefas. (PRESSMAN, 2011).

Pressman (2011, P. 40) define que:

> “Uma metodologia de processo genérica para engenharia de software compreende cinco atividades:
>
> **Comunicação**: Antes de iniciar qualquer trabalho técnico é de vital importância comunicar-se com as partes interessadas (stakeholders).
>
> **Planejamento**: Qualquer jornada complexa pode ser simplificada caso exista um mapa.
>
> **Modelagem**: Um engenheiro de software cria modelos para melhor entender as necessidades do software e o projeto que irá atender a essas necessidades.
>
> **Construção**: Essa atividade combina geração de código (manual ou automatizado) e testes necessários para revelar erros na codificação.
>
> **Emprego**: O software (como entidade completa ou como incremento parcialmente efetivado) é entregue ao cliente, que avalia o produto entregue e fornece feedback, baseado na avaliação. ”

Essas atividades metodológicas são genéricas e podem ser utilizadas para o desenvolvimento de softwares pequenos e simples. Para se criar softwares mais complexos os detalhes serão bem diferentes para cada caso embora a essência se mantenha a mesma. (PRESSMAN, 2011).

2.5. Modelo Prático para Desenvolvimento de Software

Trata-se de um modo de desenvolver softwares de maneira interativa, o Modelo prático para desenvolvimento de software (PRISM), visualiza os diagramas da UML e faz deles a base central para facilitar o desenvolvimento, utilizando-os como ferramentas de implementação, ajudando no processo de engenharia de software e monitorando toda a fase de documentação e desenvolvimento. (CARDOSO, 2003).

A proposta do PRISM é criar uma forma de desenvolvimento dinâmica de acordo com a prisma do desenvolvedor, apresentado de forma simples onde se começa o trabalho quais etapas a serem seguidas e onde se termina. (CARDOSO, 2003).

A metodologia propõe que existam 12 etapas a serem seguidas em ordem. Elas permitem que o software seja desenvolvido com prazo, custo e qualidade. (CARDOSO, 2003).

As etapas são dispostas da seguinte maneira:

- Problema;
- Levantamento e análise de requisitos;
- Modelagem de use case;
- Escolha das use cases a serem trabalhadas;
- Modelagem de classes de análise;
- Modelagem dinâmica utilizando o diagrama de sequência;
- Modelagem de classes de projeto;
- Geração de código;
- Testes;
- Sistema;
- Modelagem de sistemas de tempo real;

- Modelagem de pacotes/componentes.

2.6. A linguagem de programação C#

C# é uma linguagem de programação criada com influencias de diversas linguagens mundialmente reconhecidas como por exemplo, JAVA e C++. Ela foi criada do zero usando padrões atuais e sua sintaxe é simples e de fácil aprendizagem, muito similar com a sintaxe de JAVA e C. Além de simplificar muito as complexidades do C++. (ARAÚJO, 2015).

Araújo (2015) afirma que:

> "A linguagem C# é uma linguagem que visa facilitar muito o desenvolvimento, e possui uma vasta gama de recursos que podem proporcionar uma grande produtividade para desenvolvedores que a utilizam. Além dos recursos como sua sintaxe e programação orientada a objetos, que fazem dela uma linguagem poderosa para se trabalhar. "

2.7. Criptografia

Microsoft, explica que:

> "Criptografia é um meio de aprimorar a segurança de uma mensagem ou arquivo embaralhando o conteúdo de modo que ele só possa ser lido por quem tenha a chave de criptografia correta para desembaralhá-lo. "

2.7.1. Criptografia Simétrica

Ferreira (2012), explica que:

> "Criptografia simétrica nada mais é do que um algoritmo ("programa") que embaralha as informações tonando estas ilegíveis. Para que a informação volte a ser legível é preciso inserir a senha criada no processo de encriptação. "

2.7.2 *Advanced Encryption Standard* (AES)

Gonçalves (2015), apresenta que:

> "O AES ou *Advanced Encryption Standard* (do inglês), também chamado de Rijndael, é um algoritmo de cifragem de blocos adotada pelo Governo dos Estados Unidos desde 2001 e foi criado para se tornar um padrão em criptografia. Por se tratar de um algoritmo de criptografia simétrica, a mesma chave usada para cifrar, também é usada para decifrar uma informação. "

Mathias (2005, p. 1) explica que o AES nasceu de uma

> "[...] proposta para substituir o algoritmo criptográfico DATA ENCRIPTION STANDART (DES) " e apresenta as especificações seguintes: "o DES é publicamente definido, é uma cifra simétrica de bloco, projetado para que o tamanho da chave possa ser expansível, implementável tanto em hardware quando software, disponibilizado de maneira livre. " Ele também revela que a chave para cifrar e decifrar age com "uma chave criptografada e blocos, ambos de tamanhos com 128, 192 e 256 bits. "

2.9 Visual Studio

Visual Studio é o ambiente de desenvolvimento integrado da Microsoft que permite criar aplicativos para Web, Windows, Mac e Linux. A ferramenta é voltada para desenvolvedores que trabalham com a linguagem de programação C# e com o framework .NET. (BRITO, 2015).

CAPÍTULO III - ANÁLISE DO APLICATIVO

3.1. Documento de Visão

3.1.1. Introdução

3.1.1.1. Finalidade

Esse documento possui como finalidade a coleta, análise e definição das necessidades do aplicativo ACEcrypt. Os detalhes de como este atinge as necessidades estão descritos nos casos de uso e nas especificações auxiliares.

3.1.1.2. Definições, Acrônimos e Abreviações

ACEcrypt – Aplicativo de Criptografia Simétrica

PRISM – *Practical Software Development Model*

RUP – *Rational Unified Process*

Aplicativo – Programa computacional

3.1.2. Posicionamento

3.1.2.1. Oportunidade de Negócios

Com o avanço da tecnologia da informação em geral, a privacidade de informações a serem guardadas ou tramitadas aumentou consideravelmente. Faz-se necessário assegurar ao usuário que ele está devidamente resguardado com seus dados e transmissões, e que existe a proteção necessária com os riscos envolvidos.

É fato que a maioria dos usuários não possuem consciência que seus dados estão seguros ou não, implicando em riscos na privacidade de seus dados. Por isso métodos se segurança devem estar sustentando a informação para que ela seja protegida de acesso não autorizado.

No meio digital existe um método eficaz para a proteção das informações, sendo denominada Criptografia.

3.1.2.2. Descrição do Problema

Quadro 1 – Descrição do problema

O problema de	Roubo de informações, perda de confidencialidade em arquivos.
Afeta	Pessoas que contêm dados sigilosos.
Cujo impacto é	Perda de privacidade, vazamento, exposição e perda de confidencialidade dos dados.
Uma boa solução seria	Desenvolver um aplicativo para criptografar e descriptografar os arquivos tornando-os ilegíveis aos olhos humanos e ferramentas computacionais.

Fonte: O autor.

3.1.2.3. Sentença de Posição do Produto

Quadro 2 – Sentença de posição do produto

Para	Criptografar e descriptografar arquivos.
Quem	Necessita de sigilo em armazenamento de arquivos.
O ACEcrypt	É um Aplicativo.

Que	Permitirá criptografar e descriptografar arquivos tornando os dados ilegíveis a agentes externos.
Diferente de	Ter seus dados expostos ou espionados por agentes externos.
Nosso produto	Atuará na privacidade e confidencialidade da informação.

Fonte: O autor.

3.1.2.4. Sentença de Posição do Produto

O produto proposto é direcionado ao sigilo durante o armazenamento de arquivos digitais. Contemplará funcionalidade que fará a criptografia simétrica do arquivo, garantindo assim a privacidade e confidencialidade.

O aplicativo realizará o método de criptografia simétrica que fará a “cifra” dos dados baseado em uma chave única.

O aplicativo processará os dados e seu resultado irá garantir a privacidade e confidencialidade da informação.

3.1.3. Descrições dos Envolvidos e dos Clientes

3.1.3.1. Demografia do Mercado

A necessidade da segurança da informação em todos os âmbitos digitais é uma realidade, devido diversos escândalos e demais incidentes causados por dados privados sendo acessados por pessoas não autorizadas.

O ACEcrypt visa evitar e dificultar o acesso de pessoas não autorizadas aos arquivos assegurando a confidencialidade e a privacidade dos arquivos.

3.1.3.2. Resumo dos Envolvidos

Quadro 3 – Resumo dos Envolvidos

Nome	Descrição	Responsabilidades
Analista de Desenvolvimento	Responsável pelo desenvolvimento e análise sobre o sistema e sobre o método de criptografia a ser usado no sistema.	Manter o sigilo dos arquivos. Documentar as metodologias utilizadas e o fluxo de trabalho do sistema.
Usuário	Interação com o sistema.	Inserção de dados no sistema.

Fonte: O autor.

3.1.3.3. Resumo dos Clientes

Quadro 4 – Resumo dos Clientes

Nome	Descrição	Responsabilidades	Envolvido
Usuário	Interação com o sistema.	Inserção de dados no sistema.	Não se aplica.

Fonte: O autor.

3.1.3.4. Ambiente dos Clientes

Será utilizado exclusivamente o sistema operacional Windows.

3.1.3.5. Perfis dos Envolvidos

3.1.3.5.1. Analista de Desenvolvimento

Quadro 5 – Perfil de Envolvido Analista de Desenvolvimento

Representante	Analistas de desenvolvimento do sistema.
Descrição	Análise, documentação, criação e desenvolvimento das funcionalidades do aplicativo.
Tipo	Representantes do produto.
Responsabilidades	Desenvolver o produto.
Critérios de Sucesso	Criptografia dos arquivos, que garante a privacidade e confidencialidade dos mesmos.
Envolvimento	Total com o planejamento e desenvolvimento do produto.
Produtos Liberados	Não se aplica.
Comentários/Problemas	Não se aplica.

Fonte: O autor.

3.1.3.5.2. Usuário

Quadro 6 – Perfil de Envolvido Usuário

Representante	Usuários do sistema.
Descrição	Entidade que interage com o sistema.
Tipo	Usuário do sistema.
Responsabilidades	Alimentar o sistema com arquivos.
Critérios de Sucesso	Capacidade de inserir arquivos no sistema.
Envolvimento	Interatividade com o aplicativo.
Produtos Liberados	Não se aplica.

Comentários/Problemas	Não conseguir inserir arquivos no aplicativo.

Fonte: O autor.

3.1.3.5.2. Principais necessidades dos envolvidos ou dos clientes

Quadro 7 - Principais necessidades dos envolvidos ou dos clientes

Necessidade	Prioridade	Preocupações	Solução atual	Soluções Propostas
Sigilo no armazenamento de dados digitais.	Alta	A privacidade e confidencialidade dos dados digitais.	Esconder arquivos no sistema operacional.	Uso da criptografia para impedir acessos não autorizados aos dados.

Fonte: O autor.

3.1.3.5.3. Alternativas e Concorrência

O aplicativo proposto possui o diferencial de possuir uma usabilidade mais simples, enquanto os aplicativos alternativos são mais complexos para se utilizar.

3.1.4. Restrições

Será utilizado somente no sistema operacional Windows.

3.2. Especificações de Requisitos

3.2.1. ER aF ACEcrypt.001

Quadro 8 – Quadro de especificação de Requisito ER aF ACEcrypt.001

ER aF ACEcrypt.001	**Selecionar arquivo ou pasta**	
Descrição	O sistema deverá permitir ao usuário selecionar um arquivo.	
Descrição do risco	**Risco**	**Prioridade**
- Não escolher um arquivo.	Baixo	Baixo
- Escolher arquivo com tamanho maior que 600MB(Mega bytes).	Alto	Alto
Usuário Envolvido	Usuário/Desenvolvedor	

Fonte: O autor.

3.2.2. ER aF ACEcrypt.002

Quadro 9 – Quadro de especificação de Requisito ER aF ACEcrypt.002

ER aF ACEcrypt.002	**Inserir senha**	
Descrição	O sistema deverá solicitar a inserção de uma senha pessoal para ser processada com o arquivo.	
Descrição do risco	**Risco**	**Prioridade**
- Não inserir senha	Baixo	Baixo
Usuário Envolvido	Usuário/Desenvolvedor	

Fonte: O autor.

3.2.3. ER aF ACEcrypt.003

Quadro 10 – Quadro de especificação de Requisito ER aF ACEcrypt.003

ER aF ACEcrypt.003	**Criptografar arquivo ou pasta**	
Descrição	O sistema processará os dados digitais para gerar o arquivo ou pasta criptografado.	
Descrição do risco	**Risco**	**Prioridade**
- Gerar o arquivo final com erros.	Alto	Alta

- Não gerar o arquivo final.	Alto	Alta
Usuário Envolvido	Usuário/Desenvolvedor	

Fonte: O autor.

3.2.4. ER aF ACEcrypt.004

Quadro 11 – Quadro de especificação de Requisito ER aF ACEcrypt.004

ER aF ACEcrypt.004	**Descriptografar arquivo ou pasta**	
Descrição	O sistema processará os dados digitais para gerar o arquivo ou pasta descriptografado.	
Descrição do risco	**Risco**	**Prioridade**
- Gerar o arquivo final com erros.	Alto	Alta
Usuário Envolvido	Usuário/Desenvolvedor	

Fonte: O autor.

3.3. Descrição dos Casos de Uso e Atores

3.3.1 Casos de Uso

3.3.1.1. Selecionar arquivo

Este caso de uso é o responsável pela seleção do arquivo que será criptografado no sistema.

3.3.1.2. Inserir senha

Este caso de uso é o responsável pela inserção da senha que será usada para criptografar e descriptografar o arquivo.

3.3.1.3. Criptografar

Este caso de uso é o responsável pela criptografia do arquivo recebido pelo aplicativo.

3.3.1.4. Descriptografar

Este caso de uso é o responsável por descriptografar o arquivo criptografado.

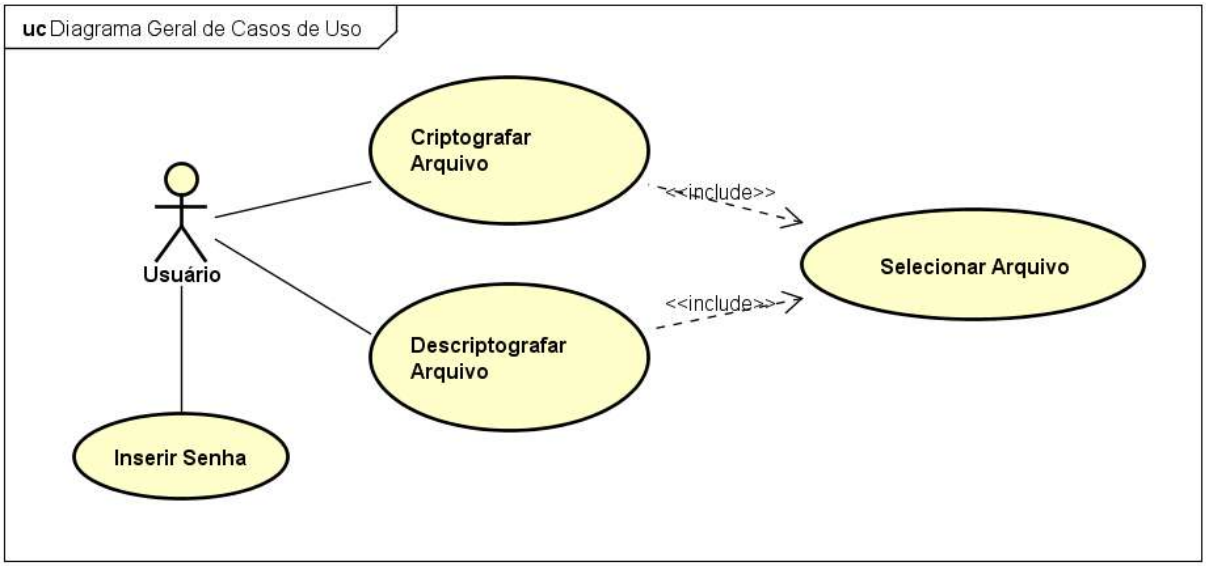
3.3.2. Descrição dos Atores

3.3.2.1 Usuário

Este ator é uma pessoa que atua no sistema inserindo o arquivo e a senha para o processo de criptografia.

3.4. Diagrama Geral de Casos de Uso

Figura 9 – Diagrama de Casos de Uso



Fonte: O autor.

3.5. Detalhamento dos casos de Uso

3.5.1 UC[01] – Criptografar Arquivo

Quadro 12 – Fluxo de Eventos do Caso de Uso Criptografar Arquivo

Nome do Caso de Uso	Criptografar Arquivo.
Descrição	Executado na tela inicial, inicia o processo de criptografia de arquivo.
Requisitos Associados	ER aF ACEcrypt.001, ER aF ACEcrypt.002 e ER aF ACEcrypt.003.
Pré-Condições	Aplicativo aberto.
Pós-Condições	O aplicativo deve processar os dados inseridos e indicar que o arquivo criptografado foi gerado com sucesso.
Atores	Usuário
Fluxo Principal	
Ações Recebidas	Ações Realizadas
1. O usuário inicia o aplicativo.	2. O aplicativo exibe as opções de criptografar e descriptografar.
3. O usuário opta por criptografar.	4. O aplicativo exibe uma tela para selecionar o arquivo a ser criptografado.
5. O usuário seleciona o arquivo.	6. O aplicativo recebe o arquivo a ser criptografado e exibe uma tela solicitando a entrada de uma senha.
7. O usuário insere a senha.	7, O aplicativo recebe a senha, processa os dados, salva o arquivo criptografado e emite uma mensagem de processo concluído.
Fluxo de Excessão A	
Ações Recebidas	Ações Realizadas
1. O usuário inicia o aplicativo.	2. O aplicativo exibe as opções de criptografar e descriptografar.
3. O usuário opta por criptografar.	4. O aplicativo exibe uma tela para selecionar o arquivo.
5. O usuário fecha a tela de seleção de arquivo.	6. O aplicativo exibe uma mensagem de erro e encerra.
Fluxo de Excessão B	
Ações Recebidas	Ações Realizadas
1. O usuário inicia o aplicativo.	2. O aplicativo exibe as opções de criptografar e descriptografar.
3. O usuário opta por criptografar.	4. O aplicativo exibe uma tela para selecionar o arquivo.
5. O usuário seleciona o arquivo.	6. O aplicativo recebe o arquivo e exibe uma tela solicitando a inserção de uma senha.

7. O usuário fecha a tela de inserção de senha.	8. O aplicativo exibe uma mensagem de erro e encerra.

Fonte: O autor.

3.5.1 UC[02] – Selecionar Arquivo

Quadro 13 – Fluxo de Eventos do Caso de Uso Selecionar Arquivo

Nome do Caso de Uso	Selecionar Arquivo.
Descrição	O usuário seleciona o arquivo a ser processado pelo aplicativo.
Requisitos Associados	ER aF ACEcrypt.001, ER aF ACEcrypt.003 e ER aF ACEcrypt.004.
Pré-Condições	O usuário ter selecionado a opção criptografar ou a opção descriptografar.
Pós-Condições	O aplicativo deve receber o arquivo.
Atores	Usuário
Fluxo Principal	
Ações Recebidas	Ações Realizadas
1.O usuário inicia o aplicativo.	2. O aplicativo exibe as opções de criptografar e descriptografar.
3. O usuário opta por criptografar ou descriptografar.	4. O aplicativo exibe uma tela para selecionar o arquivo.
5. O usuário seleciona o arquivo.	6. O aplicativo recebe o arquivo.
Fluxo de Excessão A	
Ações Recebidas	Ações Realizadas
1. O usuário inicia o aplicativo.	2. O aplicativo exibe as opções de criptografar e descriptografar.
3. O usuário opta por criptografar ou descriptografar.	4. O aplicativo exibe uma tela para selecionar o arquivo.
5. O usuário fecha a tela de seleção de arquivo.	6. O aplicativo exibe uma mensagem de erro e encerra.

Fonte: O autor.

3.5.2 UC[03] – Inserir senha

Quadro 14 – Fluxo de Eventos do Caso de Uso Inserir senha

Nome do Caso de Uso	Inserir senha.
Descrição	Recebe do usuário a senha desejada para usar como parâmetro de criptografia do arquivo.
Requisitos Associados	ER aF ACEcrypt.001, ER aF ACEcrypt.002, ACEcrypt.003, ACEcrypt.004.
Pré-Condições	Arquivo inserido no aplicativo.

Pós-Condições	O aplicativo receber a senha inserida.
Atores	Usuário
Fluxo Principal	
Ações Recebidas	Ações Realizadas
1.O usuário inicia o aplicativo.	2. O aplicativo exibe as opções de criptografar e descriptografar.
3. O usuário opta por criptografar ou descriptografar	4. O aplicativo exibe uma tela para selecionar o arquivo.
5. O usuário insere o arquivo.	6. O aplicativo recebe o arquivo e emite uma tela solicitando a senha.
7. O usuário insere a senha.	8. O aplicativo recebe a senha.
Fluxo de Excessão A	
Ações Recebidas	Ações Realizadas
1. O usuário inicia o aplicativo.	2. O aplicativo exibe as opções de criptografar e descriptografar.
3. O usuário opta por criptografar ou descriptografar.	4. O aplicativo exibe uma tela para selecionar o arquivo.
5. O usuário seleciona o arquivo.	6. O aplicativo recebe o arquivo e exibe uma tela solicitando a inserção de uma senha.
7. O usuário fecha a tela de inserção de senha.	8. O aplicativo exibe uma mensagem de erro e encerra.

Fonte: O autor.

3.5.1 UC[04] – Descriptografar

Quadro 15 – Fluxo de Eventos do Caso de Uso Descriptografar arquivo.

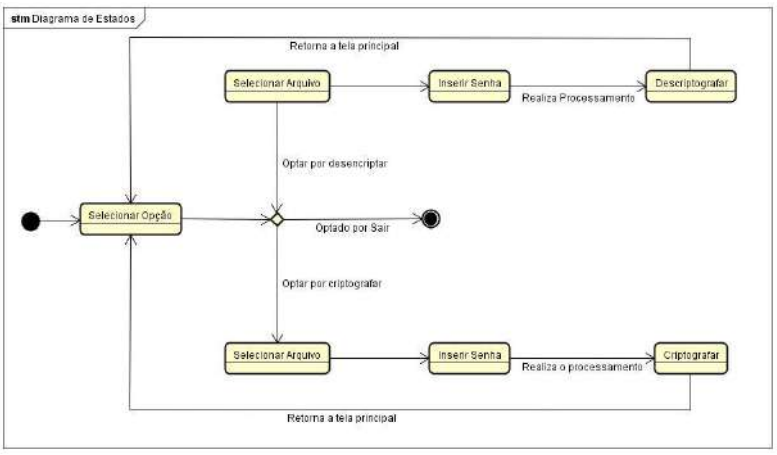
Nome do Caso de Uso	Descriptografar Arquivo.
Descrição	Executado na tela inicial, inicia o processo de descriptografar de arquivo.
Requisitos Associados	ER aF ACEcrypt.001, ER aF ACEcrypt.002 e ER aF ACEcrypt.003.
Pré-Condições	Aplicativo aberto.
Pós-Condições	O aplicativo deve processar os dados inseridos e indicar que o arquivo descriptografado foi gerado com sucesso.
Atores	Usuário
Fluxo Principal	
Ações Recebidas	Ações Realizadas
1. O usuário inicia o aplicativo.	2. O aplicativo exibe as opções de criptografar e descriptografar.
3. O usuário opta por descriptografar.	4. O aplicativo exibe uma tela para selecionar o arquivo a ser descriptografado.

5. O usuário seleciona o arquivo.	6. O aplicativo recebe o arquivo a ser descriptografado e exibe uma tela solicitando a entrada de uma senha.
7. O usuário insere a senha.	7, O aplicativo recebe a senha, processa os dados, salva o arquivo descriptografado e emite uma mensagem de processo concluído.
Fluxo de Excessão A	
Ações Recebidas	Ações Realizadas
1. O usuário inicia o aplicativo.	2. O aplicativo exibe as opções de criptografar e descriptografar.
3. O usuário opta por descriptografar.	4. O aplicativo exibe uma tela para selecionar o arquivo.
5. O usuário fecha a tela de seleção de arquivo.	6. O aplicativo exibe uma mensagem de erro e encerra.
Fluxo de Excessão B	
Ações Recebidas	Ações Realizadas
1. O usuário inicia o aplicativo.	2. O aplicativo exibe as opções de criptografar e descriptografar.
3. O usuário opta por criptografar.	4. O aplicativo exibe uma tela para selecionar o arquivo.
5. O usuário seleciona o arquivo.	6. O aplicativo recebe o arquivo e exibe uma tela solicitando a inserção de uma senha.
7. O usuário fecha a tela de inserção de senha.	8. O aplicativo exibe uma mensagem de erro e encerra.

Fonte: O autor.

3.6. Diagrama Geral de Estados

Figura 10 – Diagrama Geral de Estados



Fonte: o autor.

3.7. Classes de Análise

3.7.1. Detalhamento das Classes de Análise

3.7.1.1. Classe de Fronteira 1 – Tela Principal

Tela responsável por apresentar as opões de Criptografar e Descriptografar arquivo.

3.7.1.2. Classe de Fronteira 2 – Tela Selecionar arquivo

Tela responsável por apresentar as opções para selecionar arquivo.

3.7.1.4. Classe de Controle 1 – Criptografia de Arquivo

Responsável por receber o arquivo selecionado e executar as ações de Criptografar e Descriptografar o arquivo.

3.7.1.5. Classe de Controle 2 – Selecionar arquivo

Responsável por selecionar o arquivo no aplicativo.

3.7.1.6. Classe de Controle 3 – Receber senha

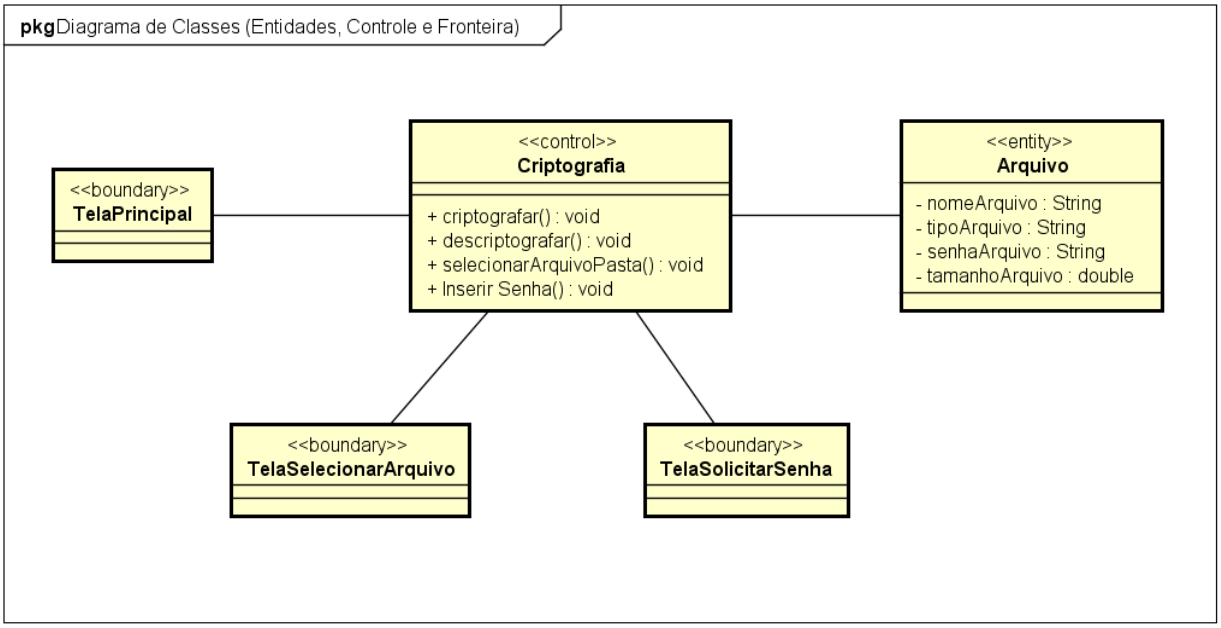
Responsável por receber a senha usada para criptografar o arquivo.

3.7.1.6. Classe de Entidade – Arquivo

Responsável por armazenar as informações do arquivo durante o tempo de processamento do aplicativo e informar o valor de seus atributos para as classes de controle.

3.7.2. Diagrama de Classes com fronteiras e entidade

Figura 11 – Diagrama Geral de Classes com fronteiras e entidade

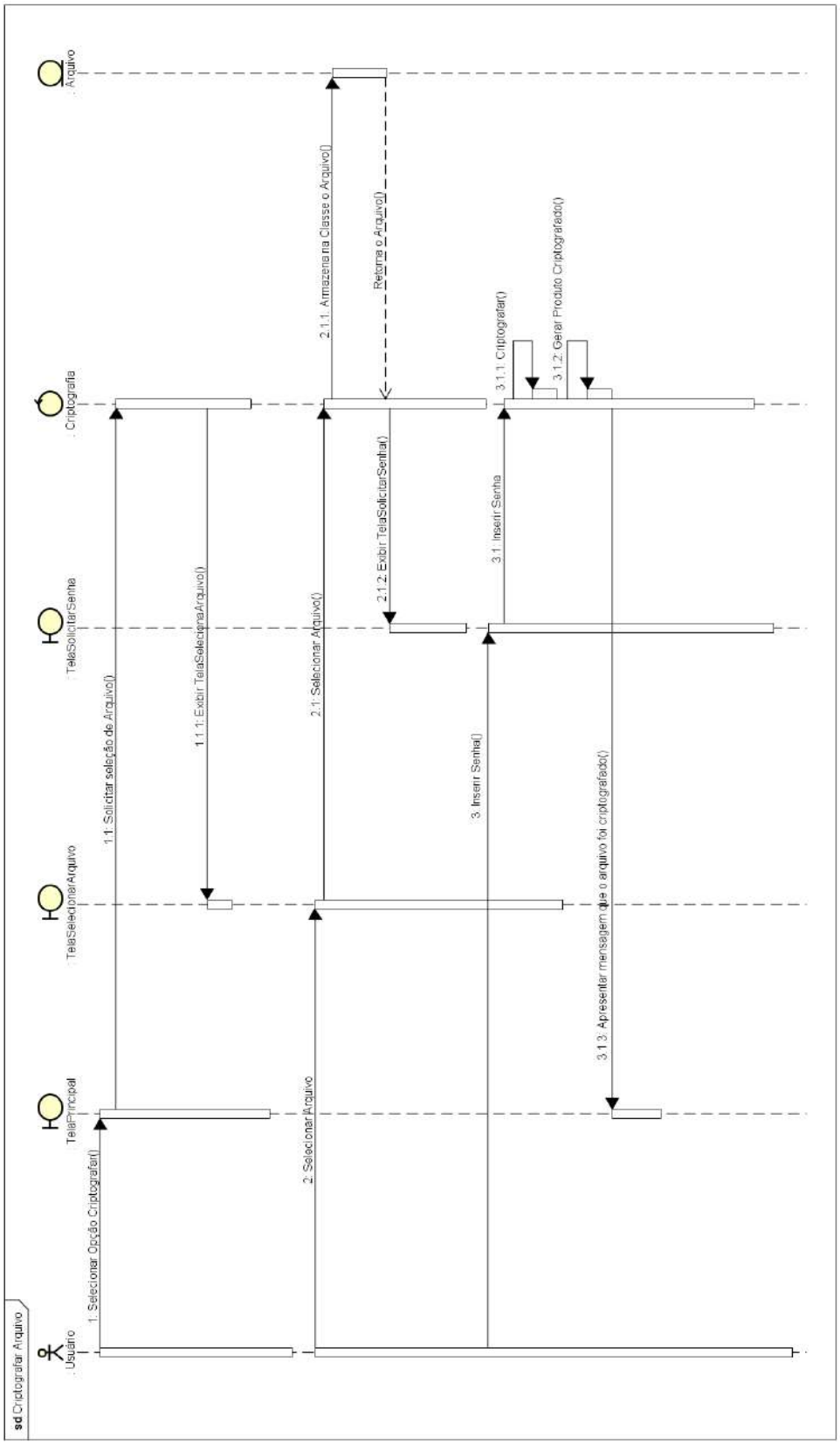


Fonte: O autor.

3.8. Diagramas de sequência

3.8.1 Diagrama de sequência fluxo criptografar arquivo

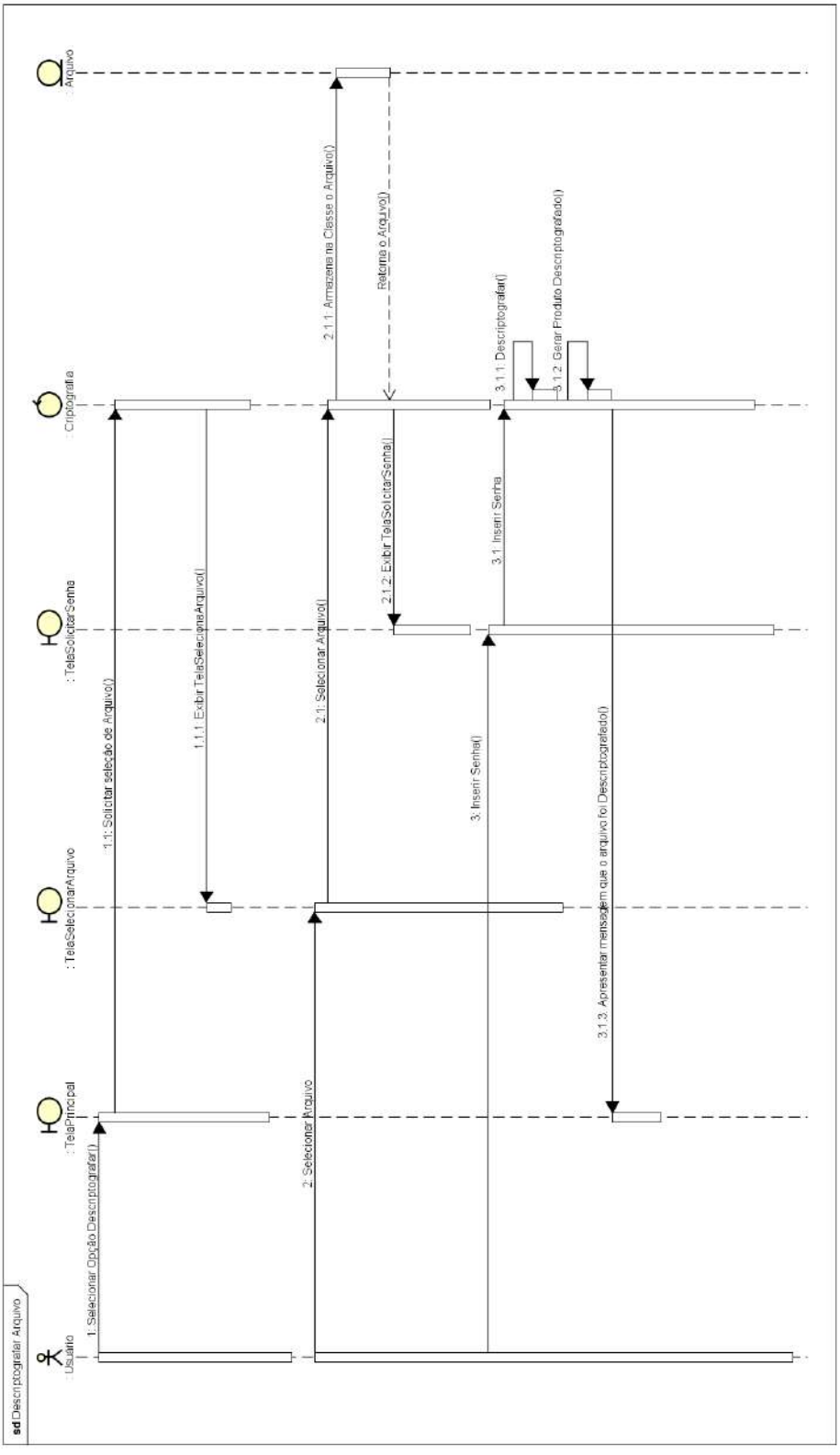
Figura 12 – Diagrama de sequência fluxo criptografar arquivo



Fonte: O autor.

3.8.2 Diagrama de sequência fluxo descriptografar arquivo.

Figura 13 – Diagrama de sequência fluxo descriptografar arquivo



Fonte: O autor

CAPÍTULO IV – CONCLUSÃO

A segurança da informação tornou-se essencial nesta era digital, pois cada vez mais tecnologias computacionais estão presentes na vida das pessoas. Atualmente informações pessoais e sigilosas estão a cada dia mais vinculadas a esse meio.

A criptografia é uma das formas existentes para a proteção de informações, porém o estudo e uso desta técnica proporciona diversos esclarecimentos sobre a necessidade do sigilo das informações. A criação e implementação do aplicativo proporcionou a compreensão da adaptação digital desta técnica, da orientação a objetos e dá análise de documentação de sistemas, além de garantir um sólido conhecimento sobre a segurança da informação e de uma das suas ferramentas. Conhecimento esse, essencial para profissionais de Tecnologia da Informação.

Conclui-se então, que o uso de um aplicativo para criptografar arquivos é uma maneira relativamente eficiente para proteger as informações pessoais contra um agente não autorizado. Utilizando a criptografia simétrica, é possível proteger os arquivos de maneira mais confiável e robusta, pois somente a pessoa com a senha reverte o processo e acessa o arquivo.

REFERÊNCIAS

AURÉLIO, Marco. **Quanto vale sua informação.** 2015. Disponível em:< http://www.administradores.com.br/mobile/artigos/tecnologia/quanto-vale-sua-informacao/90651/ > Acesso em 5 out. 2015.

ARAÚJO, Marcelo. **Análise Orientada a Objetos.** Disponível em:< http://www.devmedia.com.br/artigo-engenharia-de-software-2-analise-orientada-a-objetos/9150 > Acesso em 9 nov. 2015.

ARAÚJO, **Everton Coimbra de. Introdução à linguagem C#.** Disponível em:< http://www.devmedia.com.br/introducao-a-linguagem-c/27711 > Acesso em 10 nov. 2015.

BRITO, Edivaldo. **Visual Studio: crie apps em C# e .NET e exporte para múltiplas plataformas.** Disponível em:< http://www.techtudo.com.br/tudo-sobre/visual-studio.html > Acesso em 10 nov. 2015.

ASSOCIAÇÃO BRASILEIRA DE NORMAS E TÉCNICAS: **ABNT. NBR ISO/IEC 27002:2005** - Tecnologia da informação – Técnicas de segurança – Código de prática para a gestão da segurança da informação. Rio de Janeiro: ABNT, 2005.

CARDOSO, Caíque. **UML na Prática: do Problema ao Sistema.** Ciência Moderna. Rio de Janeiro, Brasil: 2003.

DEITEL, H. M.; **Java Como Programar.** 6. ed. São Paulo: Pearson Prentice Hall, 2005.

DIAS, Jorge Luis Ferreira. **Utilizando UML e Padrões - Parte I.** 2014. Disponível em:< http://www.devmedia.com.br/utilizando-uml-e-padroes-parte-i/4046 > Acesso em 9 nov. 2015.

DEBONI, José Eduardo Zindel. **Modelagem orientada a objetos com a UML.** Rio de Janeiro: Futura, 1995.

FERREIRA, Fernando Nicolau Freitas. **Segurança da Informação.** Rio de Janeiro: Ciência Moderna LTDA, 2003.

FOWLER, Martin. **UML essencial: Uma breve guia para a linguagem padrão de modelagem de objetos.** 3. ed. Porto Alegre: Bookman, 2013.

EDDY, Frederick et al. **Modelagem e projetos baseados em objetos.** 6. ed. Rio de Janeiro: Campus, 1994.

GONÇALVES, Paulo Alessandro Del Bianco. **Criptografia AES com .NET.** Disponível em:<http://www.devmedia.com.br/criptografia-aes-com-net/15903 > Acesso em 10/11/2015.

LYRA, Maurício Rocha. **Segurança e auditoria em sistemas de informação.** Rio de Janeiro: Ciência Moderna, 2008.

MATHIAS, Leopoldo A. P. **Algoritmo de criptografia AES.** 2005. Disponível em :< http://www.gta.ufrj.br/grad/05_2/aes/ > Acesso em 10/11/2015

MICROSOFT. **O que é criptografia?** Disponível em:< http://windows.microsoft.com/pt-br/windows/what-is-encryption#1TC=windows-7 > Acesso em 10/11/2015.

MONTURIL, Caio. **Desenvolvimento de aplicativo esteganográfico para dispositivos móveis.** 2014. 78 f. TCC (Graduação em Segurança da Informação) - Faculdades Integradas ICESP Promove de Brasília, Brasília.

PALMEIRA, Thiago Vinícios Varallo. **Introdução à Programação Orientada a Objetos em Java.** 2013. Disponível em:< http://www.devmedia.com.br/introducao-a-programacao-orientada-a-objetos-em-java/26452 > Acesso em 10/11/2015.

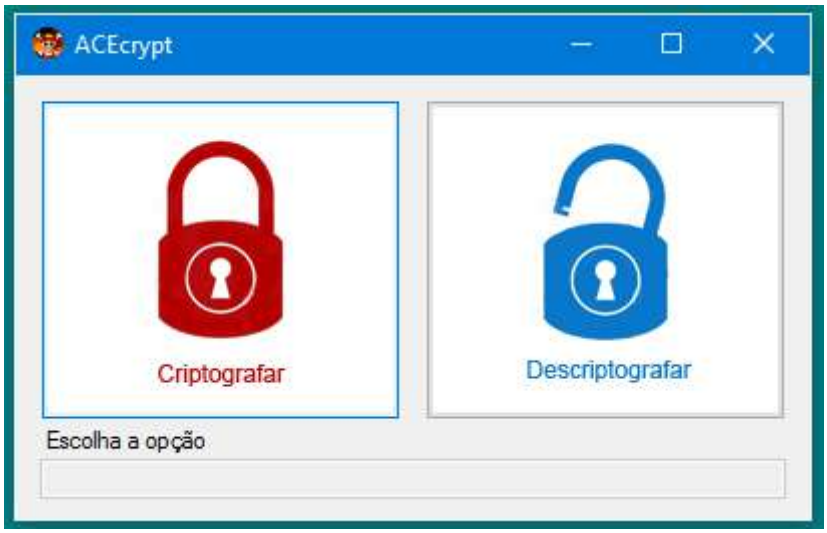
PRESSMAN, Roger S**. Engenharia de Software: uma abordagem profissional.** 7. ed. 2011. São Paulo AMGH.

STELLMAN, Andrew; GREENE, Jennifer. **Use a cabeça C#.** 2. ed. São Paulo: Alta Books, 2011.

UML DIAGRAMS**: UML 2.5 Diagrams Overview.** Disponível em:<http://www.uml-diagrams.org/uml-25-diagrams.html> Acesso em: 12/11/2015.

APÊNDICE A – Tela Principal

Figura 14 – Tela Principal

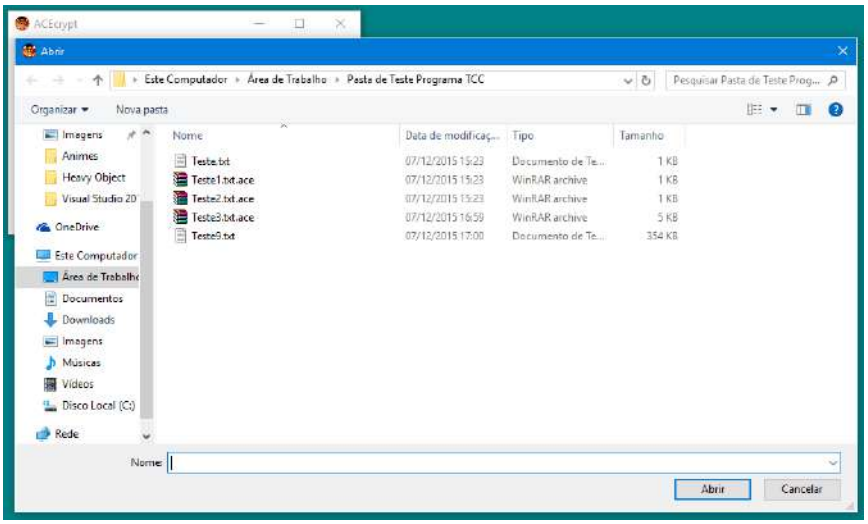


Fonte: O autor.

A tela principal, prove ao usuário as duas opções para utilização do aplicativo: criptografar e descriptografar.

APÊNDICE B – Tela Selecionar arquivo

Figura 15 – Tela Selecionar arquivo

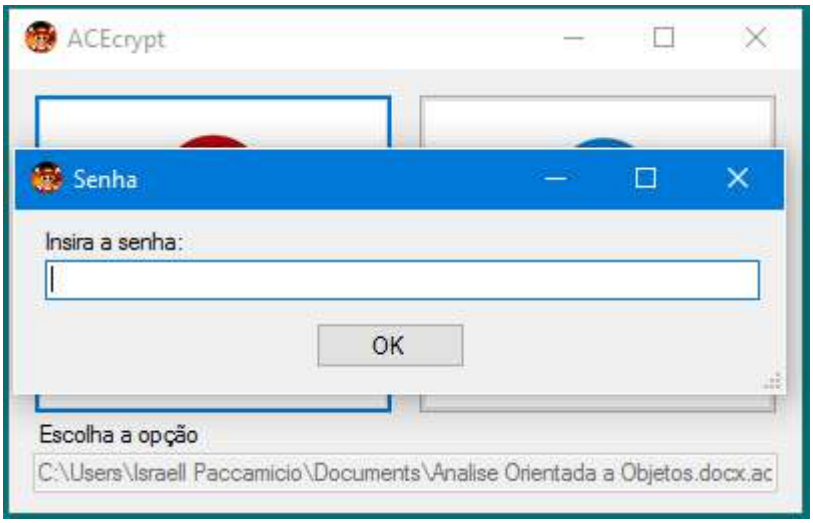


Fonte: O autor.

A tela Selecionar arquivo, contém a funcionalidade de seleção de arquivos ou pastas por intermédio do usuário.

APÊNDICE C – Tela Inserir Senha

Figura 16 – Inserir senha



Fonte: O autor.

A tela Inserir senha, apresenta uma caixa de texto, para que o usuário insira a senha desejada, a qual será usada para criptografar e descriptografar o arquivo.