



UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
PRÓ-REITORIA DE GRADUAÇÃO
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIA DA COMPUTAÇÃO

ANTONIO QUEIRÓZ NETO

**SISTEMA MOBILE PARA UNIDADES PRISIONAIS DO ESTADO DO RIO
GRANDE DO NORTE – INFOSISP MOBILE**

MOSSORÓ

2017

ANTONIO QUEIRÓZ NETO

**SISTEMA MOBILE PARA UNIDADES PRISIONAIS DO ESTADO DO RIO
GRANDE DO NORTE – INFOSISP MOBILE**

Monografia apresentada a Universidade Federal Rural do Semi-Árido como requisito para obtenção do título de Bacharel em Ciência da Computação.

Orientador: Angélica Félix de Castro, Prof. Dr.

MOSSORÓ

2017

© Todos os direitos estão reservados a Universidade Federal Rural do Semi-Árido. O conteúdo desta obra é de inteira responsabilidade do (a) autor (a), sendo o mesmo, passível de sanções administrativas ou penais, caso sejam infringidas as leis que regulamentam a Propriedade Intelectual, respectivamente, Patentes: Lei nº 9.279/1996 e Direitos Autorais: Lei nº 9.610/1998. O conteúdo desta obra tomar-se-á de domínio público após a data de defesa e homologação da sua respectiva ata. A mesma poderá servir de base literária para novas pesquisas, desde que a obra e seu (a) respectivo (a) autor (a) sejam devidamente citados e mencionados os seus créditos bibliográficos.

Q3s Queiróz Neto, Antonio.
 Sistema Mobile para Unidades Prisionais do
 Estado do Rio Grande do Norte / Antonio Queiróz
 Neto. - 2017.
 52 f. : il.

 Orientadora: Angélica Félix de Castro.
 Monografia (graduação) - Universidade Federal
 Rural do Semi-árido, Curso de Ciência da
 Computação, 2017.

 1. Sistema Penitenciário. 2. Engenharia de
 Software. 3. Modelo de Aceitação. 4. INFOSISP
 Mobile. I. Castro, Angélica Félix de, orient. II.
 Título.

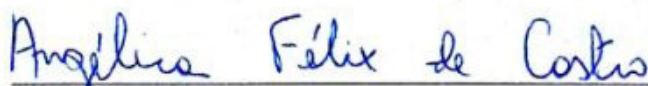
ANTONIO QUEIRÓZ NETO

**SISTEMA MOBILE PARA UNIDADES PRISIONAIS DO ESTADO DO RIO
GRANDE DO NORTE - INFOSISP MOBILE**

Monografia apresentada a Universidade
Federal Rural do Semi-Árido como requisito
para obtenção do título de Bacharel em
Ciência da Computação.

Defendida em: 22 /05/2017.

BANCA EXAMINADORA



Angélica Felix de Castro, Prof. Dr^a. (UFERSA)
Presidente



Leonardo Augusto Casillo, Prof. Dr. (UFERSA)
Membro Examinador



Danielle Simone da Silva Casillo, Prof. Dr. (UFERSA)
Membro Examinador

AGRADECIMENTOS

Agradeço infinitamente aos meus pais Cosma e Francisco Queiroz, por terem me ensinado tantas lições simplesmente sendo quem são. Em especial, a minha mãe, que sempre me proporcionou um ambiente sem restrições a carinho, risadas e livros.

Agradeço à minha família, que sempre apoiou e torceu por mim. À minha irmã Camila, ao meu irmão Arthur. Às minhas tias Jacinta, Écia, Lúcia e Francisca. Às minhas primas Cássia, Rúbia, Anny, Carol, Clara, Cecília, Candice, Tarsila e Ilana. Aos meus primos Heitor e Bernardo.

Agradeço aos meus amigos Andrezza Kharla, Jefferson Lins e Maria Luzia. Minhas palavras parecem nunca ser suficientes ou significativas o bastante para dizer o quanto vocês significam para mim, agradeço todos os dias por tê-los como amigos. Saibam que eu não teria chegado aqui sem o apoio e incentivo de vocês, leia “puxadas de orelha”.

Agradeço a todos os meus amigos, que fiz dentro e fora da universidade. Em especial a Estefânia Rifani, você me mima com tantos abraços; Isabela Sousa, minha melhor companhia nos rituais de magia negra; Igo Marcel e Jéssica Brito, ainda guardo todas as nossas conversas; Guacira Costa, Damirys Lucema e Aline Bezerra, as melhores caminhadas depois da escola foram com vocês; Alisson Maurício, você entende todas as referências; Gleice Medeiros, seremos *dark metals* para sempre; Amanda Costa, a gente se entende em toda casa do mapa astral; Ildérica Lopes, Adalcina Fernandes e Maxsuel Castro, nunca esperei ter amigos tão leais quanto vocês.

Agradeço a todos os meus professores por dividirem seus conhecimentos comigo. Agradeço aos professores Gabriel Gadelha e Amanda Gondim, por terem confiado em mim o bastante para ser seu monitor. Agradeço, em especial, à minha queridíssima orientadora Angélica Félix, pela disponibilidade e, principalmente, pela paciência. Agradeço a Banca Examinadora, Leonardo e Danielle Casillo por terem concordado em avaliar este trabalho.

*“So the means to the end was the peace
that was meant to be ours since it all began.
We fight to win.”*

Our destiny por Epica.

RESUMO

No Sistema Prisional do Rio Grande do Norte, a contagem de apenados, uma das atividades mais corriqueiras em uma unidade prisional, e à primeira vista simples, pode se mostrar dispendiosa quando as ferramentas apropriadas não são fornecidas. Partindo dessa problemática, foi proposto o desenvolvimento de um aplicativo para dispositivos móveis, utilizando as devidas técnicas de engenharia de *software*, que auxiliará os agentes penitenciários nessa atividade. Esse trabalho mostrará o desenvolvimento e o procedimento para a validação de uma ferramenta para melhorar o controle na movimentação interna dos apenados, assim como os resultados obtidos. O sistema foi testado na Cadeia Pública de Caraúbas por agentes penitenciários reais que podem efetivamente vir a utilizá-lo.

Palavras-chave: Sistema Penitenciário, Engenharia de *Software*, Modelo de Aceitação, INFOSISP *Mobile*.

ABSTRACT

In the Penitentiary System of the Rio Grande do Norte State, counting the inmates, one of the most trivial activities in a prison unity, and at a first glance simple, may unfold costly when the proper tools are not provided. Based on this problem, it was proposed the development of an application for mobile devices, using the appropriate software engineering techniques, which will help the prison agents in such activity. This paper will present the development and validation proceedings for a tool that will improve the control in the inner movement of the inmates, and so the results obtained. The system was tested on the Cadeia Pública de Caraúbas by real prison agents who may effectively come to use it.

Keywords: Penitentiary System, Softer Engineering, Acceptance Model, INFOSISP *Mobile*.

LISTA DE FIGURAS

Figura 1 - O ciclo de vida do <i>software</i>	18
Figura 2 - Ciclo de vida de uma <i>Activity</i>	22
Figura 3 - Exemplo do Modelo de Aceitação da Tecnologia (TAM).	25
Figura 4 - Diagrama de Arquitetura de <i>Software</i>	30
Figura 5 - Diagrama de Casos de Uso.	31
Figura 6 - Diagrama de Classes.	33
Figura 7 - Diagrama de Entidade e Relacionamento.	35
Figura 8- Tela inicial do INFOSISP Mobile.	36
Figura 9 - Tela home do INFOSISP Mobile.....	37
Figura 10 - Telas para mudança de cela no INFOSISP <i>Mobile</i>	38
Figura 11 - Tela para escolha de uma nova cela no INFOSISP Mobile.....	39
Figura 12 - Tela para listagem de apenados no INFOSISP Mobile.	39
Figura 13 - Aviso de novos apenados na unidade.	40
Figura 14 - Tela para listagem dos novos apenados na unidade.	40
Figura 15 - Perguntas sobre melhorias e sugestões para o INFOSIP <i>Mobile</i>	42
Figura 16- Tabelas de Diagrama de Entidade e Relacionamento detalhadas.....	51
Figura 17- Construtos de UP e FUP.	52

LISTA DE TABELAS

Tabela 1- Questionário 1- Construto sobre a Utilidade Percebida.	26
Tabela 2 - Questionário 2- Construto sobre a Facilidade de Uso Percebida.	27
Tabela 3 - Análise do fator Utilidade Percebida.....	43
Tabela 4 - Análise do fator Facilidade de Uso Percebida.....	44
Tabela 5 - Caso de uso Mover Apenado.	50
Tabela 6 - Caso de uso Listar Apenados.	50

LISTA DE ABREVIATURAS E SIGLAS

BD	Banco de Dados
BDR	Banco de Dados Relacional
BDOO	Banco de Dados Orientado a Objetos
BDOR	Banco de Dados Objeto-Relacional
CASE	<i>Computer-Aided Software Engineering</i>
CDP	Centro de Detenção Provisória
COAPE	Coordenado pela Coordenadoria de Administração Penitenciária
CPC	Cadeia Pública de Caraúbas
DC	Diagrama de Classes
DER	Diagrama de Entidade e Relacionamento
Dr	Doutor
FDD	<i>Feature-driven development</i>
FUP	Facilidade de Uso Percebida
INFOSISP	Informação do Sistema Penitenciário
MER	Modelo de Entidade e Relacionamento
RN	Rio Grande do Norte
SEJUC	Secretaria do Estado da Justiça e da Cidadania
SGBD	Sistema Gerenciador de Banco de Dados
SI	Sistema de Informação
SQL	<i>Structured Query Language</i>
TAM	<i>Technology Acceptance Model</i>
TI	Tecnologia da Informação
UP	Utilidade Percebida
UML	<i>Unified Modeling Language</i>

LISTA DE SÍMBOLOS

@	Arroba
©	Copyright
®	Marca registrada

SUMÁRIO

1	INTRODUÇÃO.....	15
1.1	Justificativa.....	15
1.2	Objetivos.....	16
1.2.1	Objetivo Geral.....	16
1.2.2	Objetivos Específicos	16
1.3	Estrutura do Trabalho	16
2	REFERENCIAL TEÓRICO.....	17
2.1	Processos e Metodologias de Desenvolvimento de <i>Software</i>	17
2.1.1	Modelo Sequencial Linear.....	17
2.2	Banco de Dados	19
2.2.1	Bancos de Dados Objeto-Relacional (BDOR)	19
2.3	<i>Android</i>	20
2.3.1	<i>Activity</i>	21
2.3.2	SQLite.....	23
2.4	O Sistema Penitenciário do Rio Grande do Norte.....	23
2.4.1	O Agente Penitenciário.....	23
2.4.2	Cadeia Pública de Caraúbas (CPC)	24
2.5	O processo de Testes e Validação	24
2.5.1	Modelo de Aceitação TAM	24
3	DESENVOLVIMENTO DO SISTEMA.....	28
3.1	Especificação de Requisitos	28
3.2.1	Requisitos do Sistema	29
3.2.1	Diagramas de Arquitetura de <i>Software</i>	30
3.2.1	Diagramas de Caso de Uso	31
3.2	Elaboração de Projeto.....	32
3.2.2	Diagrama de Entidade e Relacionamento	34
3.3	Implementação.....	35
3.3.1	O Sistema de Informação para Unidades Prisionais <i>Mobile</i> (INFOSISP <i>Mobile</i>).....	36
4	TESTES E VALIDAÇÃO.....	42
4.1	Instrumento de Procedimento de Coleta de dados.....	42
4.2	Análise dos Dados	43
4.3	Análise dos Resultados.....	43
4.4.1	Utilidade Percebida (UP)	43

4.4.2	Facilidade de Uso Percebida (FUP).....	44
5	CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS	45
5.1	Trabalhos futuros.....	45
	REFERÊNCIAS	47

1 INTRODUÇÃO

Com o avanço da tecnologia e a grande difusão de computadores, é, no mínimo, curioso pensar em um sistema que não aproveite as facilidades oferecidas por tais. Um sistema desenvolvido com funcionalidades específicas para a resolução de um problema é, sem dúvidas, a maneira mais eficaz de fazê-lo, implicando em uma série de vantagens, como a agilidade, confiabilidade, segurança, entre outras.

Um dos grandes avanços tecnológicos é a mobilidade incorporada aos computadores. Um *smart phone*, por exemplo, não mais tem como único objetivo realizar chamadas; agora é uma ferramenta poderosa capaz de realizar diversos tipos de tarefas, com um potencial tão grande quanto o de qualquer computador de mesa.

Apesar de todas as vantagens oferecidas, existem instituições que não se beneficiam de ferramentas tecnológicas por diversas razões. O Sistema Penitenciário do Estado do Rio Grande do Norte é uma delas, não possuindo informatização integrada de seus dados. Suas atividades são feitas através de uso de papel e caneta ou de tecnologias básicas, como exemplo, ferramentas da suíte de aplicativos para escritório da Microsoft®, o *Microsoft Office*, que não atendem a todas as necessidades, deixando as informações sujeitas a inconsistências e redundância.

1.1 Justificativa

A escolha do tema se deu pela necessidade de ser desenvolvido um aplicativo para dispositivos móveis que irá auxiliar os agentes penitenciários da Cadeia Pública de Caraúbas a realizarem as movimentações internas dos apenados, que é uma das atividades mais corriqueiras das unidades prisionais, e também uma das mais importantes. Essa movimentação é feita mais de uma vez por dia.

Para que a contagem de apenados seja realizada, é necessária uma lista de todos os apenados da unidade e da respectiva cela em que se encontram. Além da chegada de novos apenados e da saída, seja por término da pena ou transferência para outra unidade, há internamente uma troca de celas constante, por exemplo, para isolamento e triagem.

Atualmente, os agentes registram as mudanças de celas por meio de listas com os nomes dos apenados e suas respectivas celas impressas em folhas de papel ofício; em seguida, o agente responsável se desloca até o bloco de celas e deve, manualmente com uma caneta esferográfica, rabiscar todas as modificações com base na última contagem feita. Então, a lista é refeita no computador e reimpressa para, finalmente, ser feita a contagem oficial.

O tempo gasto no processo descrito pode ser diminuído com a ajuda de um aplicativo para dispositivos móveis no qual o agente consiga registrar essas mudanças de celas, acrescentando agilidade e eficiência ao processo e uma maior garantia da integridade dos dados.

1.2 Objetivos

O objetivo deste trabalho é projetar um aplicativo que facilite o gerenciamento de movimentações internas dos presos nas Unidades Prisionais.

1.2.1 Objetivo Geral

O objetivo geral deste trabalho é desenvolver um aplicativo para dispositivos móveis com sistema operacional *Android* para auxiliar os agentes penitenciários da Cadeia Pública de Caraúbas.

1.2.2 Objetivos Específicos

O desenvolvimento deste trabalho considera, ainda, os seguintes objetivos:

- Modelar o aplicativo seguindo as práticas da Engenharia de *Software*;
- Desenvolver o aplicativo de acordo com as especificações;
- Testar e validar o aplicativo com funcionários da Cadeia Pública de Caraúbas.

1.3 Estrutura do Trabalho

O Capítulo 2 apresenta uma listagem dos principais conceitos que serão abordados, são eles: (I) Processos e Metodologias de Desenvolvimento de *Software*; (II) Banco de Dados; (III) *Android*; (IV) O Sistema Penitenciário do Rio Grande do Norte; e (V) O Processo de Testes e Validação.

O Capítulo 3 mostrará como se deu o desenvolvimento do sistema, onde são apresentadas suas fases. (I) Especificação de requisitos; (II) Elaboração do Projeto; e (III) Implementação.

O Capítulo 4 mostrará os resultados do projeto e o processo de validação. (I) Instrumento de Procedimento e Coleta de Dados; (II) Universo da Amostra; (III) Análise de Dados; e (IV) Análise dos Resultados.

O Capítulo 5 fará as considerações finais para esse trabalho e falará sobre trabalhos futuros.

2 REFERENCIAL TEÓRICO

Neste capítulo, serão abordados os conceitos sobre processos e metodologias de desenvolvimento de *software*, em especial o Modelo Sequencial Linear; o processo de testes e validação, alguns aspectos relevantes relacionados ao Sistema Penitenciário do Estado do Rio Grande do Norte e, ainda, alguns aspectos do Sistema *Android* para dispositivos móveis.

2.1 Processos e Metodologias de Desenvolvimento de *Software*

De acordo com Sommerville (2007), um processo de *software* pode ser definido como um conjunto de atividades e resultados associados que levam à produção de um produto de *software*.

Diferentes processos de *software* organizam essas atividades de maneiras diversas (AMBLER, 2004). Segundo (PMBOK, 2008), as atividades básicas de todo e qualquer processo são: (i) entender as necessidades do cliente, (ii) planejar a solução, (iii) implementar a solução, (iv) validar esta solução e (v) entregar o produto ao cliente.

O conjunto de atividades recomendadas para o desenvolvimento de *software* denomina-se metodologia de desenvolvimento. Existem os modelos tradicionais, chamados “pesados” e os modelos ágeis, conhecidos como “leves”.

Os modelos tradicionais ainda são amplamente utilizados nas empresas e fazem parte da história da Engenharia de *Software*. Para Souza Neto (2004), as metodologias tradicionais, por vezes chamadas de “pesadas”, têm como característica principal a divisão em etapas e/ou fases. Essas fases são bem definidas e englobam atividades como Análise, Modelagem, Desenvolvimento e Testes.

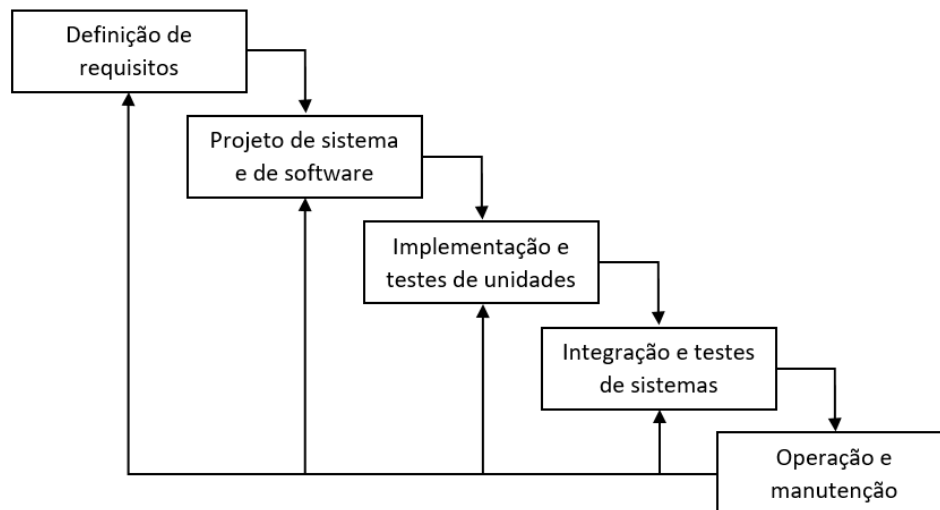
O tópico a seguir fala sobre o Modelo Sequencial Linear, escolhido para este projeto por se mostrar eficaz e atender as necessidades.

2.1.1 Modelo Sequencial Linear

O Modelo Sequencial Linear, proposto em 1970, é também conhecido como Modelo em Cascata, no qual as fases definidas são sistematicamente seguidas de maneira linear (PRESSMAN, 2006).

A Figura 1 representa as fases do Modelo Sequencial Linear.

Figura 1 - O ciclo de vida do *software*.



Fonte: Sommerville (2007).

As fases do Modelo em Cascata são definidas a seguir, de acordo com PRESSMAN (2006).

- *Análise e definição de requisitos:* consultam-se os usuários do sistema a fim de estabelecer funções, objetivos e restrições que, em seguida, são definidos em detalhes e servirão como especificação da aplicação.
- *Projeto de sistemas e de software:* o projeto de sistemas agrupa os requisitos de *hardware* ou de *software* e estabelece uma arquitetura da aplicação geral. O projeto de *software* identifica e descreve as abstrações fundamentais do sistema a ser desenvolvido e suas relações.
- *Implementação e teste de unidades:* neste estágio, o projeto de *software* pode ser compreendido como um conjunto de programas ou unidades de programa, enquanto o teste de unidades envolve verificar se cada unidade está atendendo à sua especificação.
- *Integração e teste de sistemas:* as unidades de programa são integradas e testadas como um sistema completo, com o intuito de verificar se os requisitos de *software* foram atendidos. Após esta fase, o sistema de *software* é entregue ao cliente.
- *Operação e manutenção:* geralmente, esta é a fase mais longa do ciclo de vida. O sistema é instalado e colocado em operação. Caso apareçam erros que não foram descobertos em estágios anteriores, a manutenção os corrige e, à medida que se criam novos requisitos, as funções do sistema são aumentadas.

2.2 Banco de Dados

Banco de Dados (BD) é uma coleção de informações, também chamados de dados, devidamente relacionadas e que têm sentido. Os dados podem ser armazenados e possuem um significado implícito. Machado (2008) afirma que um Banco de Dados possui as seguintes propriedades:

- É uma coleção lógica coerente de dados com um significado inerente; deste modo, uma disposição desordenada dos dados não pode ser referenciada como Banco de Dados;
- É projetado, construído e preenchido com valores de dados para um propósito específico; um Banco de Dados possui um conjunto predefinido de usuários e de aplicações;
- Apresenta algum aspecto do mundo real, o qual é chamado de minimundo; qualquer alteração efetuada no minimundo é automaticamente refletida no Banco de Dados.

Os bancos de dados computadorizados são normalmente criados e mantidos pelos Sistemas Gerenciadores de Bancos de Dados (SGBDs). O SGBD é um sistema de *software* de uso geral que facilita o processo de *definição, construção, manipulação e compartilhamento de bancos de dados* entre diversos usuários e aplicações (DATE, 2003).

Definir um banco de dados envolve especificar os tipos, estruturas e restrições dos dados a serem armazenados (ELMASRI E NAVATHE, 2011).

De maneira semelhante à engenharia de *software*, existem modelos para a criação de Bancos de Dados, chamados paradigmas. Dentre esses paradigmas, pode ser destacado o objeto-relacional, um híbrido entre o modelo mais tradicional relacional e o modelo orientado a objetos, que será utilizado nesse projeto e explicado no próximo tópico.

2.2.1 Bancos de Dados Objeto-Relacional (BDOR)

Enquanto em um de Banco de Dados Relacional (BDR) modela os dados de forma que sejam percebidos como tabelas pelo usuário, o Banco de Dados Orientado a Objetos (BDOO) armazena os dados na forma de objetos.

O Banco de Dados Objeto-Relacional utiliza uma técnica, chamada mapeamento objeto-relacional, na qual é realizada uma correspondência entre as relações de um Banco de Dados Relacional com as classes do programa.

Informalmente, cada relação de um BDR é semelhante a uma tabela de valores. Para que o usuário tenha uma melhor visão dessa abordagem relacional, o banco de dados é mostrado como um conjunto de tabelas bidimensionais, originadas em *linhas e colunas*.

Quando uma relação é considerada uma tabela de valores, cada linha na tabela representa uma coleção de dados relacionais; uma linha representa um fato que normalmente corresponde a uma entidade ou relacionamento do mundo real; os nomes da tabela são usados para ajudar a interpretar o significado dos valores em cada linha; os nomes de coluna representam os valores de dados em cada linha, com base na coluna em que cada valor se encontra; e todos os valores em uma coluna são do mesmo tipo de dado (OLIVEIRA, 2013).

Segundo Machado (2008), a Teoria Relacional possui premissas que definem uma tabela de dados:

- Cada uma das tabelas é chamada de relação;
- Cada coluna dessa tabela tem um nome e representa um domínio da tabela;
- A ordem das linhas é irrelevante;
- Não há duas linhas iguais;
- A ordem das colunas também é irrelevante.

É importante a compreensão de que cada linha de uma tabela representa um objeto, um assunto que é descrito pelos valores de cada uma dessas colunas.

De acordo com Oliveira (2013), existem quatro operações que podem ser aplicadas nos estados das relações de um banco de dados: inserir, utilizada para adicionar novos dados à tabela; selecionar, utilizada para resgatar as informações de uma única tupla ou de várias ao mesmo tempo; excluir, para excluir uma linha da tabela; e alterar, utilizada para alterar as informações de uma tupla.

Para termos acesso a esses comandos, é utilizada a linguagem para Banco de Dados Relacional. A SQL (*Structured Query Language*), ou Linguagem de Consulta Estruturada, foi desenvolvida originalmente no início dos anos 70 e é bastante popular. A linguagem SQL é um dos principais motivos para o sucesso dos Bancos de Dados relacionais comerciais, pois, dentre outros benefícios, proporciona a padronização (OLIVEIRA, 2013).

2.3 Android

Tendo milhões de usuários em mais de 190 países, o *Android* é o sistema operacional para dispositivos móveis mais popular do mundo segundo Android (2017). O núcleo do *Android* é baseado no sistema operacional *Linux*. Atualmente, esse sistema operacional é continuado pela empresa de tecnologia *Google*. A linguagem de programação utilizada para desenvolvimento de aplicações em *Android* é o Java.

Para o desenvolvedor, o *Android* disponibiliza quatro componentes que servirão de base para o desenvolvimento de aplicações: *Activity*, *Service*, *Content Provider* e *Broadcast Reciever*.

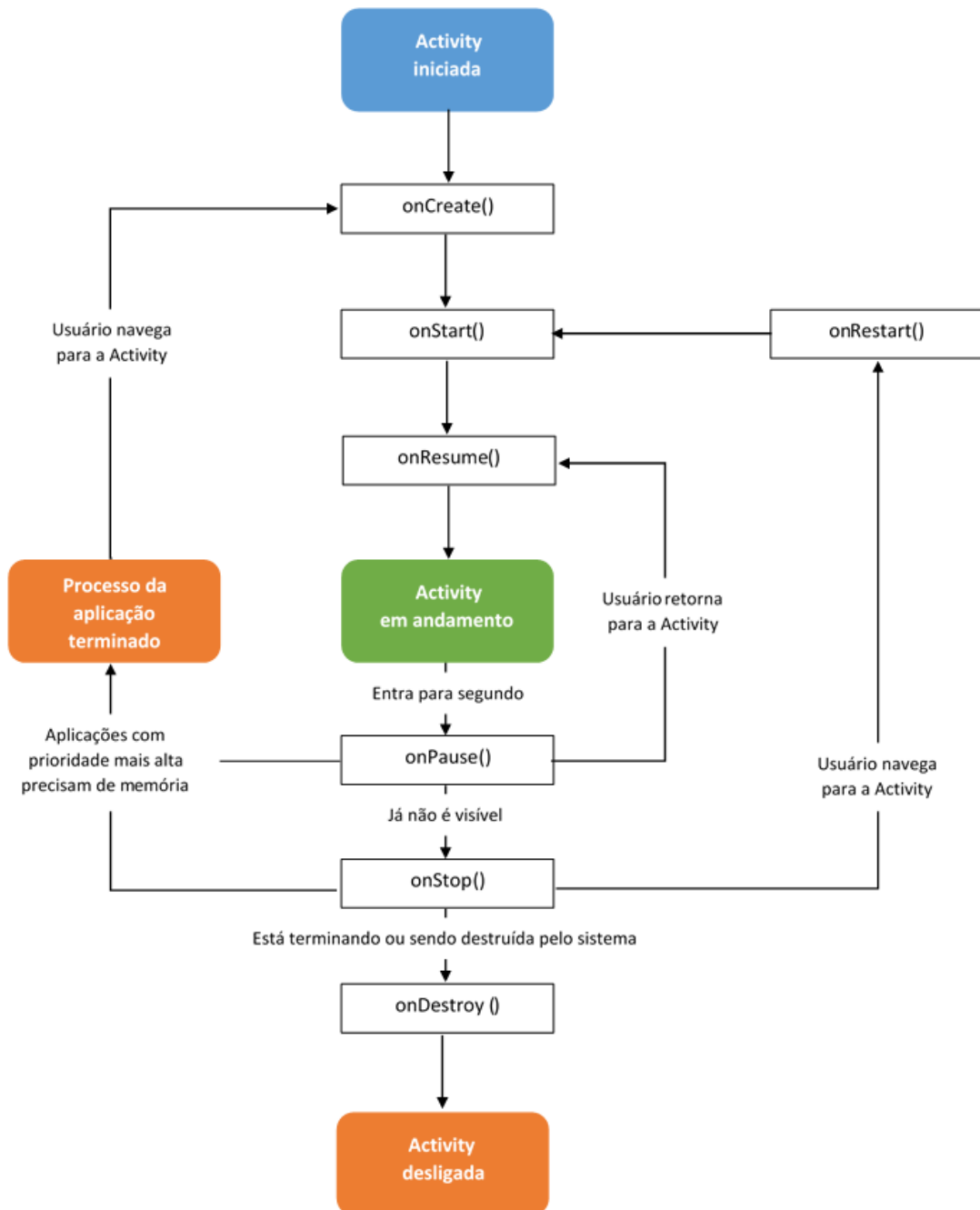
2.3.1 *Activity*

Podemos pensar nesse componente como sendo a tela de uma aplicação. De fato, uma *Activity* é responsável pelo gerenciamento de uma tela e pela exibição de sua interface gráfica. Normalmente, uma aplicação *Android* é formada por várias telas e, por conseguinte, várias *Activities*.

Uma *Activity* tem, essencialmente, quatro estados: *Em andamento*, quando está no *foreground* da tela; *Pausada*, quando não está em foco, mas ainda é visível; *Parada*, quando é completamente coberta por outra *Activity*; e *Desligada*, quando está terminando ou sendo destruída pelo sistema.

Na Figura 2, vemos os métodos padrões que estão presentes em cada *Activity*. Cada um desses métodos representa uma fase na vida da *Activity* e é chamado pelo próprio *Android* no momento que lhe é pertinente. Quando a *Activity* é criada, automaticamente o sistema chama o método *onCreate()*; o método *onStart()* indica que a *Activity* deve se preparar para ser mostrada na tela; o *onResume()* indica que a *Activity* é a que tem o foco no momento; o método *onPause()* é chamado para parar algum componente de *Activity* que não tenha mais uso em segundo plano; para parar uma *Activity* é chamado o método *onStop()*; e o método *onDestroy()* caso a *Activity* seja encerrada.

Figura 2 - Ciclo de vida de uma Activity.



Fonte: Adaptado de ANDROID (2017).

Existem duas maneiras de criar interfaces gráficas no *Android*: a declarativa e a programática. Na maneira declarativa, as propriedades da interface gráfica são declaradas em um arquivo XML (*eXtensible Markup Language*), ou Linguagem de Marcação Extensível, e então são carregadas na aplicação a partir desse arquivo; na maneira programática, os componentes são descritos em tempo de execução. As duas maneiras interagem muito bem e se completam.

2.3.2 SQLite

O Sistema *Android* tem um pacote de *software* que provê um sistema de gerenciamento de bancos de dados relacionais, o SQLite. Mesmo seu tamanho total não passando de 1 *megabyte* e seu consumo de memória também sendo bem baixo, o SQLite não perde em desempenho a outros SGBDs.

Os comandos padrões do SQLite para interação com o BD são similares aos da linguagem SQL. Eles são *create*, *select*, *insert*, *update*, *delete* e *drop*.

2.4 O Sistema Penitenciário do Rio Grande do Norte

O Sistema Penitenciário do Estado do Rio Grande do Norte pertence à Secretaria de Estado da Justiça e Cidadania (SEJUC) e é Coordenado pela Coordenadoria de Administração Penitenciária (COAPE). Que tem como missão, de acordo com SEJUC-RN (2017).

Missão: Gerenciar o Sistema Penitenciário do Estado propiciando uma melhor estrutura às unidades, para acolher e proporcionar ao preso provisório, condenado ou interno em geral, condições dignas para o cumprimento da pena ou decisão criminal, nos termos dos direitos e deveres contidos na Lei de Execução Penal.

Atualmente (2017) o Sistema Penitenciário do RN é constituído por trinta e três unidades prisionais, distribuídas em Cadeias, Centros de Detenção Provisória (CDP), Penitenciárias, Complexos Penais e Unidade Psiquiátrica de Custódia.

2.4.1 O Agente Penitenciário

O agente penitenciário do Estado do Rio Grande do Norte tem uma jornada de trabalho de quarenta horas semanais, com escala, em média, de vinte e quatro horas de trabalho com setenta e duas horas de folga. Em seu dia de trabalho desempenha várias atividades como: assistência, fiscalização, funções ostensivas e atribuições administrativas, conforme normas vigentes. Dentre tantas atividades, será dado ênfase apenas à parte administrativa, que é o foco deste trabalho.

Ao desempenhar tantos papéis em seu dia de trabalho, o agente penitenciário não conta com nenhuma ferramenta que o ajude na realização de algumas tarefas as quais o auxiliaria muito nesses processos.

2.4.2 Cadeia Pública de Caraúbas (CPC)

A CPC fica localizada na região oeste potiguar na cidade de Caraúbas a 296 quilômetros da capital do Estado do Rio Grande do Norte (RN), Natal e foi inaugurada em dezembro de 2004. A CPC foi projetada com capacidade para noventa e seis (96) presos e é destinada ao recolhimento de presos provisórios.

A carceragem da CPC é formada por dois pavilhões: A e B, onde cada um contém seis celas com capacidade para oito presos cada uma. Em cada pavilhão existe, ainda, uma área com mais quatro celas, chamadas “parlatório”, que são celas especiais que são utilizadas como isolamento de apenados. Existem também, três celas, chamadas de “triagem”, que são celas especiais onde são recolhidos os apenados que chegam à unidade e lá ficam, inicialmente, para o acompanhamento do comportamento do mesmo por parte dos agentes penitenciários. Por fim, existem mais duas celas onde ficam os apenados que trabalham na área administrativa da unidade.

2.5 O processo de Testes e Validação

Baseando-se no estudo realizado por Oliveira (2015), no qual foi avaliado os principais modelos de validação de uma tecnologia, optou-se pelo Modelo de Aceitação da Tecnologia ou *Technology Acceptance Model* (TAM) para validar o sistema proposto neste trabalho.

Esta metodologia foi escolhida pela facilidade de uso e por seu objetivo ser avaliar a intenção de uso de uma tecnologia através da observação da utilidade e da facilidade de uso de um sistema.

2.5.1 Modelo de Aceitação TAM

O Modelo de Aceitação da Tecnologia, do inglês *Technology Acceptance Model* (TAM), foi proposto por Davis (1989). Este modelo é útil para identificar o porquê da não aceitação de um sistema ou tecnologia em particular pelos usuários, e, consequentemente, implementar os passos corretivos adequados (DAVIS; BAGOZZI; WARSHAW, 1989).

De acordo com Davis (1989), a teoria do TAM tem como finalidade entender o comportamento do indivíduo a partir de duas crenças: utilidade percebida e a facilidade de uso percebida.

Conforme Davis (1989), os principais determinantes do modelo TAM são:

- *Utilidade percebida (UP)*: é o grau em que uma pessoa acredita que o uso de um sistema particular pode melhorar o seu desempenho;

- *Facilidade de uso percebida (FUP)*: é o grau em que uma pessoa acredita que o uso de um sistema de informação será livre de esforço.

Para Pires (2008), o principal objetivo do modelo TAM é disponibilizar uma base para mapear o impacto de fatores externos sobre aqueles internos ao indivíduo. Considerando-se essa proposição, o grau com que uma pessoa acredita que, ao utilizar um sistema específico, irá melhorar seu desempenho no trabalho corresponde ao construto utilidade percebida, e o grau com que o sujeito crê que, ao utilizar um sistema específico, não despenderá esforço corresponde à facilidade de uso.

O Modelo TAM é representado como mostra a Figura 3, segundo a visão de Davis (1989).

Figura 3 - Exemplo do Modelo de Aceitação da Tecnologia (TAM).



Fonte: Adaptado de Davis (1989).

O modelo só pode referir-se às questões diretamente relacionadas ao usuário e suas percepções sobre o uso do sistema. Conforme Saleh (2004), os construtos devem ser desenvolvidos de modo a captar opiniões pessoais e tratar suposições a respeito de terceiros (pessoas ou instituições).

2.5.1.1 Utilidade Percebida (UP)

Para Davis, Bagozzi e Warshaw (1989), a utilidade percebida mede o quanto uma pessoa acredita que o uso de um determinado sistema aumenta seu desempenho no trabalho, ou seja, é o grau que o usuário acredita que, o uso do sistema irá melhorar seu desempenho.

Davis (1989) concluiu com a Utilidade Percebida, que um sistema com alta avaliação pelo usuário pode gerar confiança, influenciando no seu relacionamento ou na opção de uso.

Ainda de acordo Davis (1989), a utilidade percebida é quem ajuda na intenção de uso das tecnologias pelos usuários.

A Tabela 1 mostra o questionário formulado para este trabalho a partir das adaptações de Davis (1989).

Tabela 1- Questionário 1- Construto sobre a Utilidade Percebida.

Variável	Questões
UP1	Utilizar o INFOSISP <i>Mobile</i> melhora meu desempenho no trabalho?
UP2	Utilizar o INFOSISP <i>Mobile</i> aumenta minha produtividade no trabalho?
UP3	Utilizar o INFOSISP <i>Mobile</i> melhora minha eficiência no meu trabalho?
UP4	Utilizar o INFOSISP <i>Mobile</i> torna o meu trabalho mais fácil?
UP5	Utilizar o INFOSISP <i>Mobile</i> reduz o tempo que gasto em atividades improdutivas?
UP6	Utilizar o INFOSISP <i>Mobile</i> me poupa tempo para outras atividades?
UP7	Utilizar o INFOSISP <i>Mobile</i> supre as necessidades relacionadas ao meu trabalho?
UP8	Utilizar o INFOSISP <i>Mobile</i> me dá maior controle sobre os dados referentes ao meu trabalho?
UP9	De forma geral, o INFOSISP <i>Mobile</i> é útil para o meu trabalho?

Fonte: Adaptado de Davis (1989).

2.5.1.2 Facilidade de Uso Percebida (FUP)

No construto Facilidade de Uso Percebida, é questionado o quanto um indivíduo confia que o uso de um determinado sistema é fácil e, portanto, livre de esforço (DAVIS, 1989). Tal fator tem sido amplamente utilizado em estudos sobre aceitação de modo geral.

A Tabela 2 mostra o questionário formulado para este trabalho a partir das adaptações de Davis (1989).

Tabela 2 - Questionário 2- Construto sobre a Facilidade de Uso Percebida.

Variável	Questões
FUP1	Utilizar o INFOSISP <i>Mobile</i> é agradável?
FUP2	No INFOSISP <i>Mobile</i> eu sempre sei onde estou e onde quero chegar?
FUP3	O INFOSISP <i>Mobile</i> tem uma interação compreensível e clara?
FUP4	No INFOSISP <i>Mobile</i> é fácil encontrar a informação que desejo?
FUP5	O INFOSISP <i>Mobile</i> possui uma interface atraente?
FUP6	Não é necessário treinamento para utilizar o INFOSISP <i>Mobile</i> ?
FUP7	Os recursos de navegação no INFOSISP <i>Mobile</i> são fáceis de encontrar?
FUP8	Acho necessário pouco esforço para utilizar o INFOSISP <i>Mobile</i> ?
FUP9	De forma geral, eu acho o INFOSISP <i>Mobile</i> fácil de utilizar?

Fonte: Adaptado de Davis (1989).

Os conceitos abordados no decorrer deste capítulo se mostram fundamentais para o completo entendimento deste trabalho, como os que foram apresentados na seção de Banco de Dados ou as informações sobre o funcionamento do Sistema Penitenciário do Rio Grande do Norte, ou foram utilizados em algum momento no projeto ou validação do sistema, como os conceitos da engenharia de *software* e o modelo de aceitação TAM.

3 DESENVOLVIMENTO DO SISTEMA

O desenvolvimento do aplicativo foi definido com base nas práticas de Engenharia de *Software*. O objetivo de usar essas práticas é, tomando como base as necessidades do cliente, chegar a um produto de *software* por meio de atividades e um ciclo de vida bem definidos.

Em Sommerville (2007) é encontrada a classificação de diversos modelos de processo de *software* segundo seu ciclo de vida. Para o processo de desenvolvimento do aplicativo, será utilizado o modelo em cascata, pois este se mostrou adequado para o desenvolvimento e os requisitos do projeto terem sido reconhecidos e definidos.

Utilizando uma abordagem baseada em disciplinas, o desenvolvimento do projeto será guiado por casos de uso, seu ciclo de vida é iterativo e incremental, projetado e documentado utilizando a notação UML (Unified Modeling Language), ou Linguagem de Modelagem Unificada.

Nesse contexto, de acordo com Sommerville (2007), as fases de desenvolvimento deste trabalho serão basicamente:

- *Especificação de requisitos*: esta é a fase na qual serão conhecidas as partes interessadas, também chamados *stakeholders*, e suas necessidades para o sistema, gerando artefatos de especificação de requisitos, casos de uso, entre outros.
- *Elaboração do projeto*: é feita a análise de forma mais detalhada sobre o domínio do problema, assim como a modelagem da arquitetura do projeto. Nesta fase, são gerados artefatos como diagramas e modelagem do banco de dados.
- *Implementação*: esta é a fase da concretização dos requisitos, arquitetura e modelagem propostos, gerando um produto de *software*.
- *Testes e validação*: nessa fase é percebido se foram solucionados erros de “comportamento” do *software* assegurando se as entradas definidas produzem resultados reais que coincidem com os requisitos que atendam a especificação e a validação do sistema pelos usuários.

As próximas seções apresentam detalhadamente cada fase de desenvolvimento deste trabalho.

3.1 Especificação de Requisitos

Nessa fase da especificação de requisitos, o objetivo é conhecer os *stakeholders*, ou seja, as partes interessadas, e, ainda, analisar as necessidades, avaliar a possibilidade de

desenvolvimento computacional, especificar a solução de modo não ambíguo, negociar uma condição razoável e validar a especificação.

Os *stakeholders* são pessoas que têm interesse ou são, de alguma forma, afetadas pelo projeto. Os *stakeholders* desse projeto são:

- *Agente Penitenciário*: funcionário de uma unidade prisional que gerencia os dados dos apenados e provê informações necessárias para o desenvolvimento da aplicação.

3.2.1 Requisitos do Sistema

O *software* desta pesquisa tem por objetivo gerenciar movimentações internas de apenados de uma unidade prisional, permitindo acesso aos dados por meio de um aparelho com sistema *Android*. Com este sistema, é esperada a otimização dos processos relativos ao sistema penitenciário do estado do Rio Grande do Norte, beneficiando, inicialmente e especificamente, a Cadeia Pública de Caraúbas - RN.

O sistema *INFOSISP mobile*, Sistema de Informação *Mobile* para Unidades Prisionais do Rio Grande do Norte, deve ser acessível às unidades prisionais, bastando para isso a disponibilização do arquivo *.apk*, que serve como instalador do aplicativo para os dispositivos com sistema *Android*, e acesso à *internet* para comunicação com o servidor onde estarão as informações necessárias ao aplicativo, como os dados das unidades prisionais e os dados dos apenados.

Para identificação dos requisitos necessários, foram realizadas reuniões presenciais com agentes penitenciários da CPC e com a equipe de desenvolvimento. No decorrer das reuniões, foram levantados os requisitos indispensáveis, considerando os processos da CPC realizados pelos agentes penitenciários. Dessa forma, o sistema terá um único tipo de usuário: o Agente Penitenciário, que poderá gerenciar a localização interna dos apenados na unidade em que está lotado.

O cadastro de novos apenados será feito em outro sistema, proposto por Penha (2017), assim como a exclusão dos mesmos, deixando o Agente Penitenciário somente com permissão de mudar o apenado de cela. Casos de transferência, fuga, término da pena, ou qualquer outra movimentação que requer a saída do apenado da Unidade Prisional também serão feitas em outro sistema. Além da movimentação, o usuário terá acesso à listagem e busca de apenados cadastrados na unidade prisional em que ele seja lotado

O *software* deste trabalho tem como requisitos funcionais para o Agente Penitenciário:

- Listagem de apenados por unidade de lotação;

- Mudança de cela do apenado.

Os requisitos não funcionais para o usuário são: usabilidade, segurança e acesso remoto.

3.2.1 Diagramas de Arquitetura de *Software*

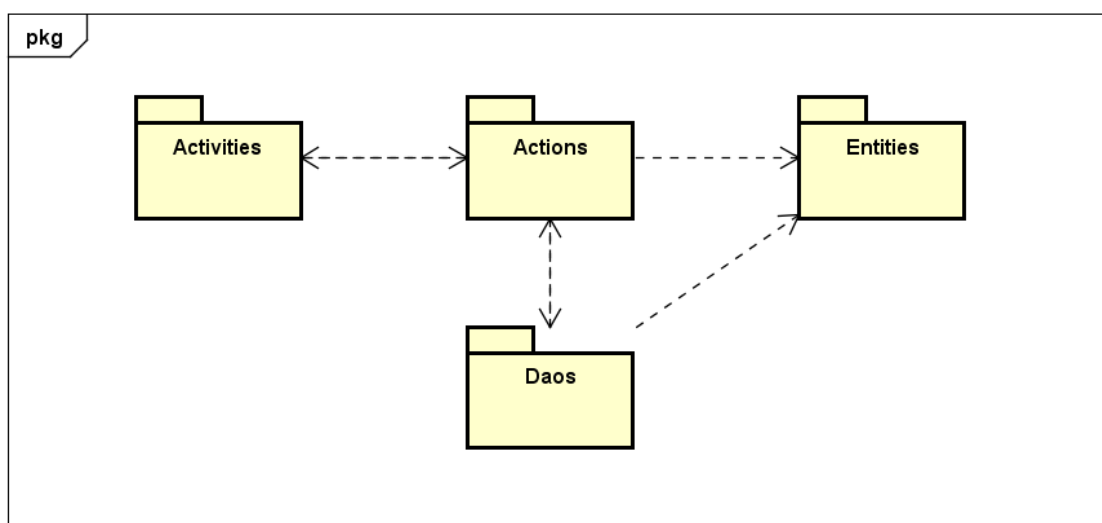
A arquitetura consiste em um modelo de alto nível que possibilita um entendimento e uma análise mais fácil do *software* a ser desenvolvido. O uso de arquitetura para representar soluções de *software* foi incentivado principalmente por duas tendências (GARLAN e PERRY, 1995):

- O reconhecimento por parte dos projetistas que o uso de abstrações facilita a visualização e o entendimento de certas propriedades do *software*;
- A exploração cada vez maior de *frameworks* visando diminuir o esforço de construção de produtos através da integração de partes previamente desenvolvidas.

Outra propriedade da arquitetura é a possibilidade de usá-la como ferramenta para comunicar a solução projetada aos diversos *stakeholders* que participam do processo de desenvolvimento do *software* (GARLAN, 2000). Contudo, para que essa comunicação seja possível, a arquitetura deve ser representada através de um documento, conhecido como documento arquitetural.

A Figura 4 apresenta o Diagrama de Arquitetura de *Software* para o sistema desta pesquisa.

Figura 4 - Diagrama de Arquitetura de *Software*.



Fonte: Autoria Própria.

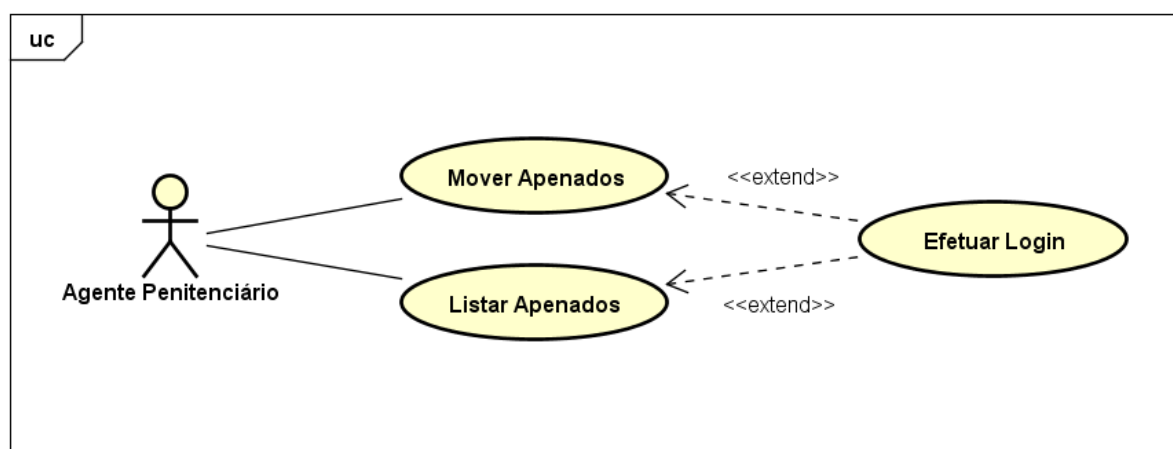
O Diagrama de Arquitetura de *Software* da Figura 4 apresenta quatro pacotes que são: *Activities*, *Actions*, *Entities*, *Daos*. No pacote *Activities* estão as telas do aplicativo que interagem com as instâncias das classes do pacote *Actions*. As instâncias das classes desse pacote interagem com as do pacote *Daos*, que por sua vez, realiza a interação com o Banco de Dados associado. O pacote *Entities* possui as classes que realizam todo o mapeamento das entidades do Banco de Dados.

3.2.1 Diagramas de Caso de Uso

Os Casos de Uso descrevem as funcionalidades que serão construídas no sistema proposto e também a sequência de eventos de um ator que usa o sistema para completar um dos processos. Cada requisito funcional foi formalizado em um Caso de Uso e pode ser encontrado no Apêndice A deste trabalho.

A Figura 5 - Diagrama de Casos de Uso. Figura 5 apresenta o Diagrama de Casos de Uso relacionados ao ator Agente Penitenciário.

Figura 5 - Diagrama de Casos de Uso.



Fonte: Autoria Própria.

Esse diagrama foi modelado utilizando a notação UML e a ferramenta CASE *Astah Community*, tem por objetivo descrever uma visão externa do sistema e representar o ator e os casos de uso a ele associados. O ator é representado pelo boneco localizado na lateral esquerda da Figura 5, os casos de uso são representados pelas elipses ao centro da figura e as linhas que ligam o ator aos casos de uso representam a comunicação entre eles, mostrando o que e como o ator interage com o sistema. As setas tracejadas indicam que para o ator ter acesso a essas funcionalidades, precisa primeiro ter efetuado *login* no sistema.

As funcionalidades propostas pelos casos de uso foram implementadas e são apresentadas no Capítulo 4.

3.2 Elaboração de Projeto

A fase de elaboração do projeto é a parte da engenharia de *software* que se encarrega de transformar os resultados da análise de requisitos em um documento ou conjunto de documentos e/ou diagramas, capazes de serem interpretados diretamente pelo engenheiro de *software*. Foram elaborados o diagrama de classes e o diagrama de entidade e relacionamento, apresentados nas próximas seções.

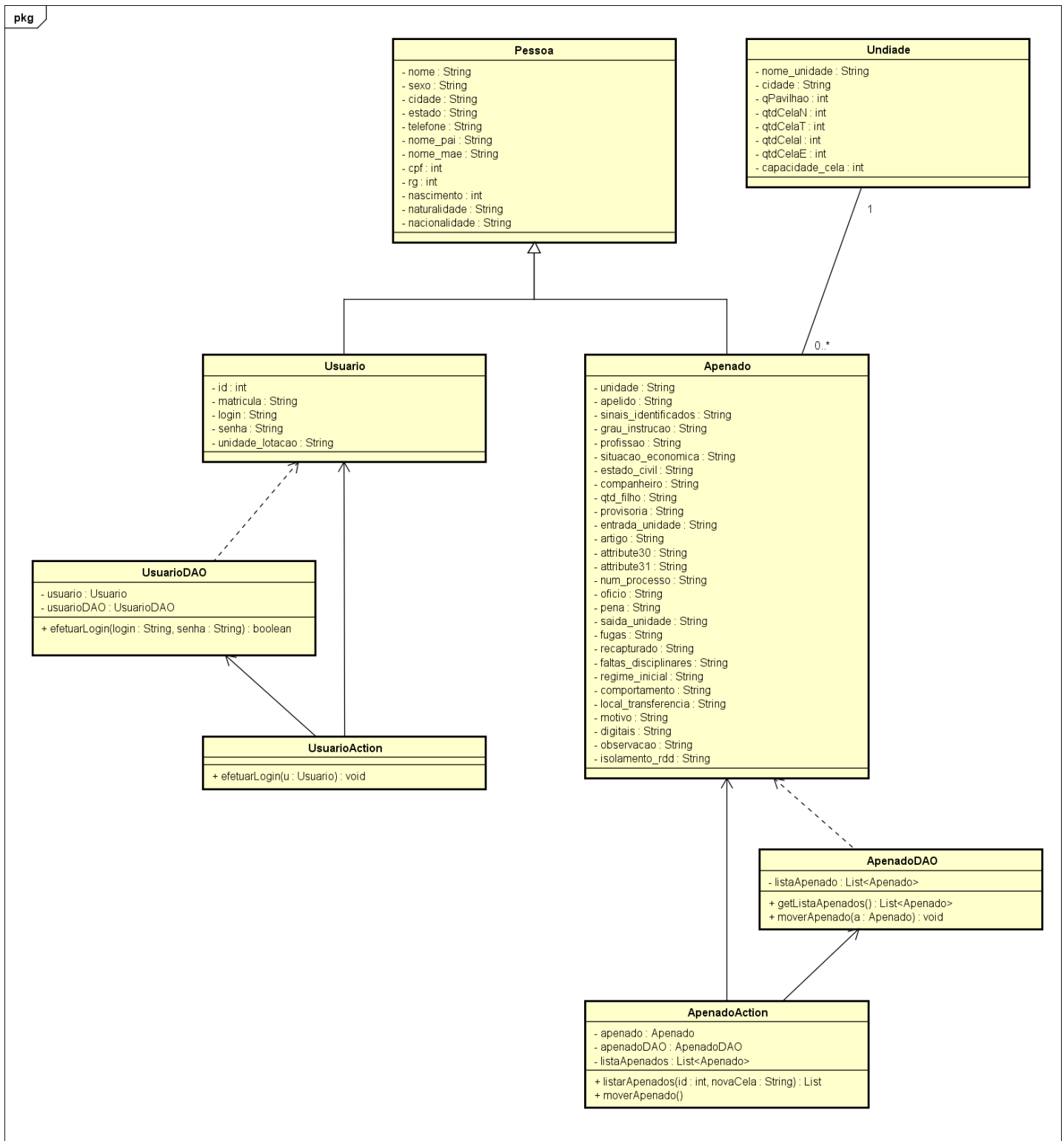
3.2.1 Diagrama de Classes

O Diagrama de Classes é uma modelagem útil e significativa para o desenvolvimento de sistemas. Este diagrama, também modelado utilizando a UML, deve ser compreendido pela equipe de desenvolvimento e define todas as classes com seus atributos e operações necessárias para o sistema. Neste Diagrama de Classes, as classes são representadas por retângulos; os relacionamentos entre classes são representados por linhas; e a conectividade de associação entre classes é indicada pelos números acima das linhas

Para uma melhor organização e estruturação do projeto, as classes foram separadas por pacotes, sendo os principais: *Entities*, *Daos* e *Actions*. O pacote *Entities* conterá as classes que fazem o mapeamento com as tabelas do Banco de Dados. Já o pacote *Daos* conterá as classes que interagem diretamente com o BD, de acordo com a *Entity* equivalente. E o pacote *Actions* conterá as classes que fazem o mapeamento com as telas, também de acordo com a *Entity* equivalente.

A Figura 6 apresenta o Diagrama de Classes no qual as classes são representadas por retângulos; os relacionamentos entre classes são representados por linhas; e a conectividade de associação entre classes é indicada pelos números acima das linhas. As relações de *Usuário* e *Apenado* com *Pessoa* são de herança, representadas pelos segmentos de retas com setas de cor branca. Os segmentos de retas com setas representam uma relação de associação, que é formada por instâncias de objetos durante a execução do sistema. Assim, tem-se que *UsuárioAction* está associada aos objetos das classes *Usuário* e *UsuárioDAO*, como também, *ApenadoAction* está associado com objetos das classes *Apenado* e *ApenadoDAO*. O fragmento de reta tracejada que liga *ApenadoDAO* e *Apenado* significa que a primeira possui uma relação de dependência com a segunda, assim como, o fragmento que liga *UsuárioDAO* e *Usuário*.

Figura 6 - Diagrama de Classes.



Fonte: Autoria Própria.

3.2.2 Diagrama de Entidade e Relacionamento

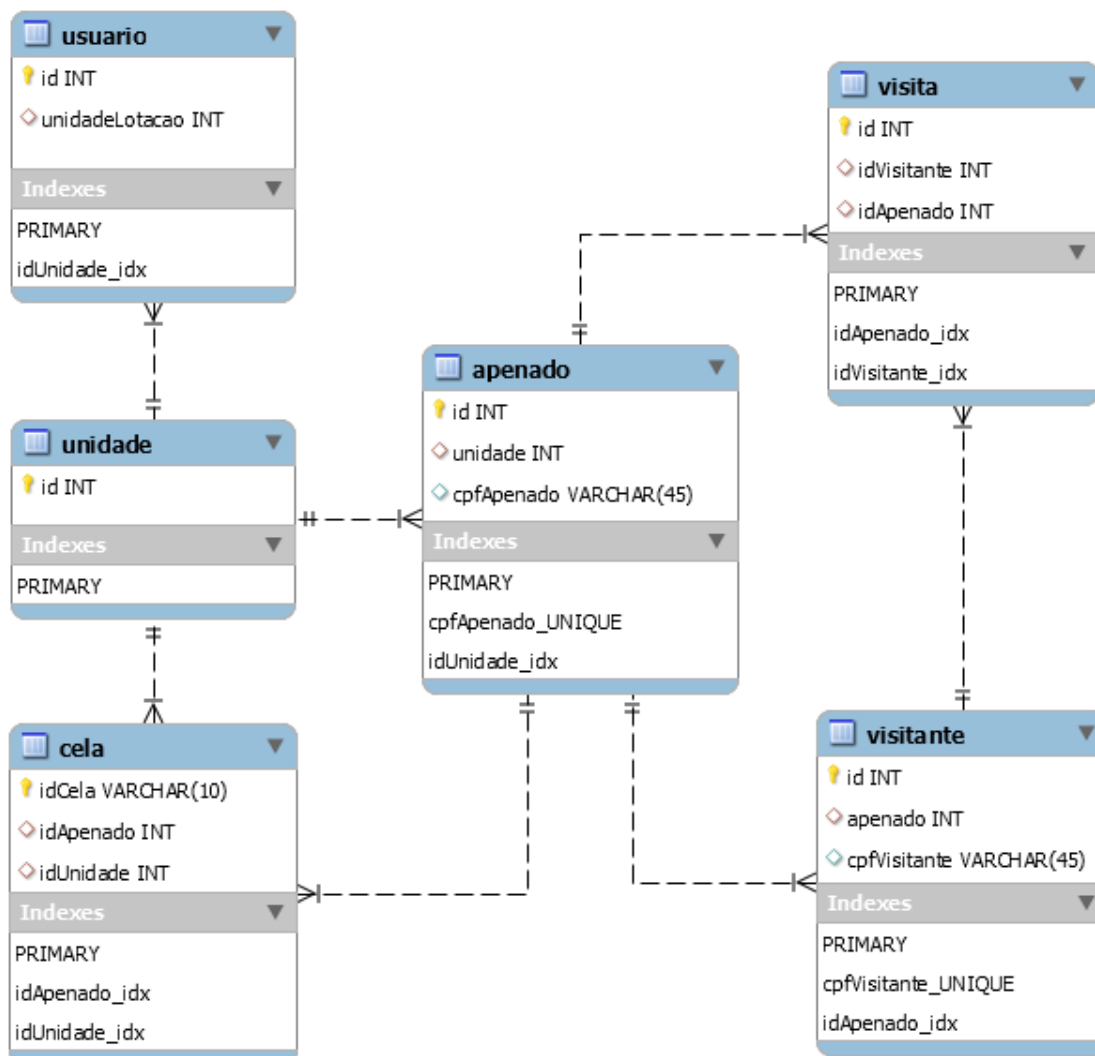
Um Diagrama de Entidade e Relacionamento é a representação gráfica do Modelo de Entidade e Relacionamento. O Modelo de Entidade e Relacionamento é um modelo abstrato, cuja finalidade é descrever, de maneira conceitual, os dados a serem utilizados em um sistema de informação. Este modelo é uma representação da realidade e é representado por entidades, relacionamentos e atributos.

Nesse diagrama, as entidades são representadas pelos retângulos, onde na parte superior é descrito o nome da entidade e na parte interna, são descritos os seus atributos. As linhas que ligam as entidades representam os relacionamentos entre elas.

Em termos de bancos de dados, cada entidade será uma tabela e os seus atributos as colunas da tabela. Os relacionamentos determinam o modo como cada registro de uma tabela se associa a registros de outras tabelas, como quando uma Unidade pode possuir vários usuários, por exemplo.

A Figura 7 apresenta o diagrama de entidade e relacionamento do sistema deste trabalho.

Figura 7 - Diagrama de Entidade e Relacionamento.



Fonte: Autoria Própria.

A representação detalhada, com atributos, de cada classe é apresentada no Apêndice B.

3.3 Implementação

A linguagem de programação escolhida para o desenvolvimento do sistema foi o JAVA, por ser uma linguagem comercial e reconhecida pela maioria dos programadores. A interface, como é padrão para aplicações *Android*, foi desenvolvida utilizando a Linguagem de Marcação *eXtensible Markup Language* (XML) junto com o próprio JAVA.

No que se refere ao armazenamento dos dados, a aplicação foi desenvolvida usando o *SQLite*, um pacote de *software* que implementa um Sistema Gerenciador de Banco de Dados (SGBD).

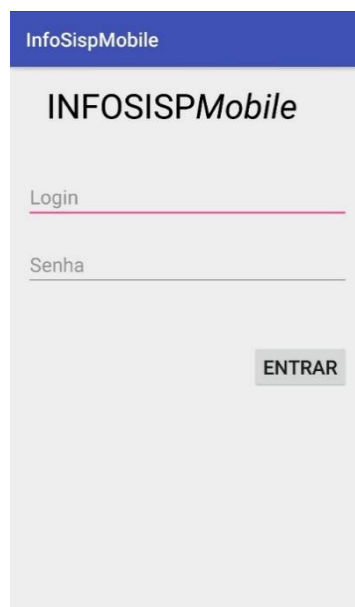
3.3.1 O Sistema de Informação para Unidades Prisionais *Mobile* (INFOSISP *Mobile*)

O Sistema de Informação para Unidades Prisionais *Mobile* (INFOSISP *Mobile*) se refere a uma aplicação para *Android*, a qual será utilizada para ter acesso às informações de maneira fácil e ágil. Este sistema irá auxiliar nos processos de trabalho dos funcionários que irão utilizá-lo, para este caso inicialmente, os Agentes Penitenciários da CPC-RN.

O INFOSISP *Mobile* é um sistema de informação integrado que auxiliará os usuários da CPC na movimentação interna dos apenados. O INFOSISP *Mobile* abrange um único tipo de usuário, que é o próprio agente penitenciário.

Na Figura 8, é apresentada a tela inicial do INFOSISP *Mobile*, na qual os usuários realizam o *login* para ter acesso às funcionalidades do *software*.

Figura 8- Tela inicial do INFOSISP Mobile.



Fonte: Autoria Própria.

O usuário, após efetuar o *login*, terá acesso a um menu, no qual poderá acessar as funcionalidades disponíveis. Além do menu, é mostrada o nome da unidade em que o usuário está lotado e a quantidade de apenados encarcerados. Essa tela é mostrada na Figura 9.

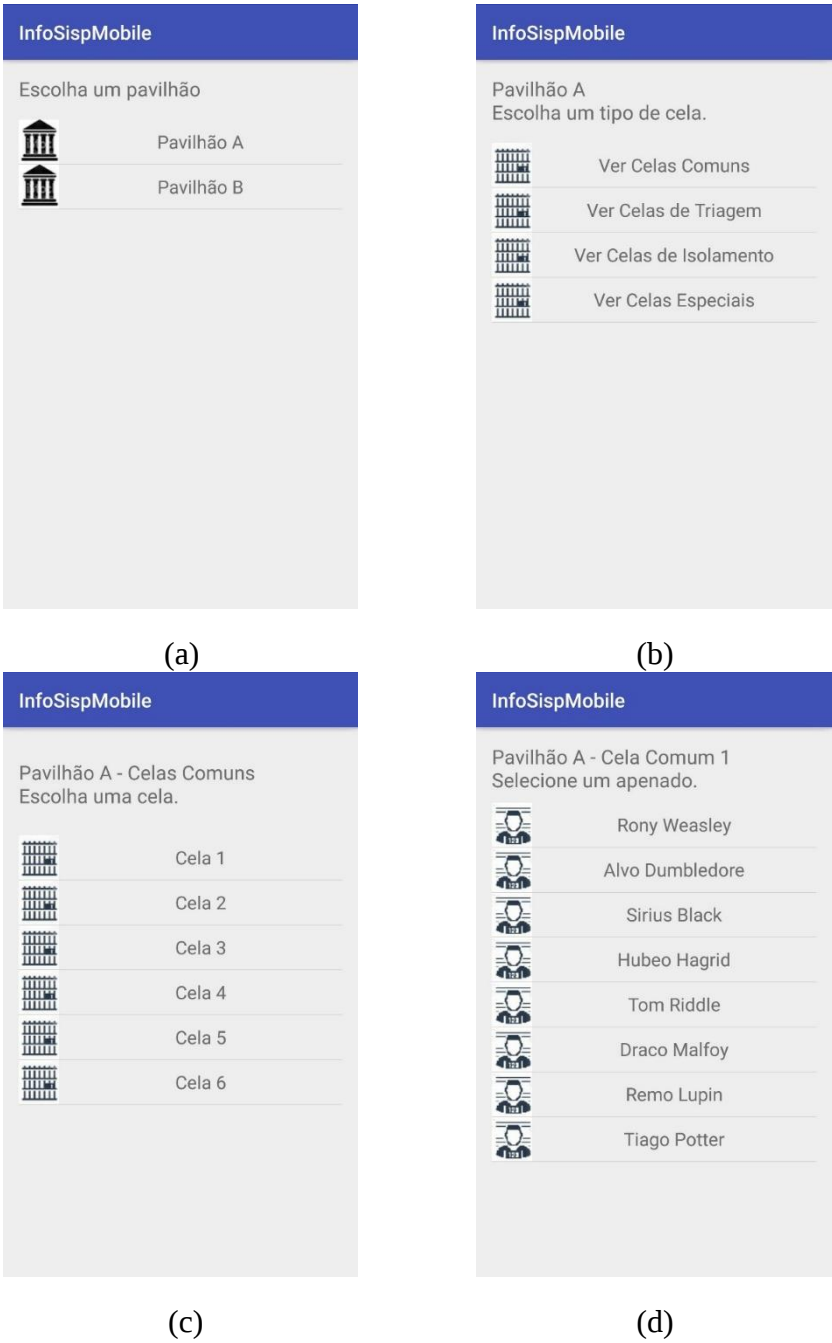
Figura 9 - Tela home do INFOSISP Mobile.



Fonte: Autoria Própria.

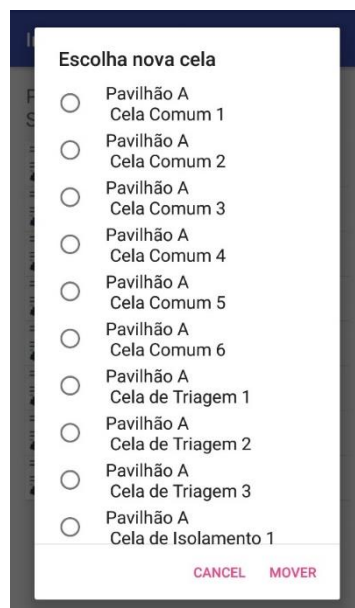
Para mover um apenado para outra cela, devemos seguir os seguintes passos: selecionar o pavilhão em que o apenado se encontra, a tela é mostrada na Figura 10 (a); em seguida selecionar o tipo de cela, a tela é mostrada na Figura 10 (b); selecionar a cela, a tela é mostrada na Figura 10 (c); selecionar o apenado que se deseja mover, como na Figura 10 (d); e, por último, selecionar a cela para onde deseja que o apenado se mude, como mostrado na Figura 11.

Figura 10 - Telas para mudança de cela no INFOSISP *Mobile*.



Fonte: Autoria Própria

Figura 11 - Tela para escolha de uma nova cela no INFOSISP Mobile.



Fonte: Autoria Própria.

Outra funcionalidade disponível no *INFOSISP Mobile* é a listagem de todos os apenados da unidade prisional em que o agente está lotado. Para ter acesso à lista, basta o usuário, na tela *home*, escolher o menu “Listar apenados”. A tela com a essa lista é mostrada na Figura 12.

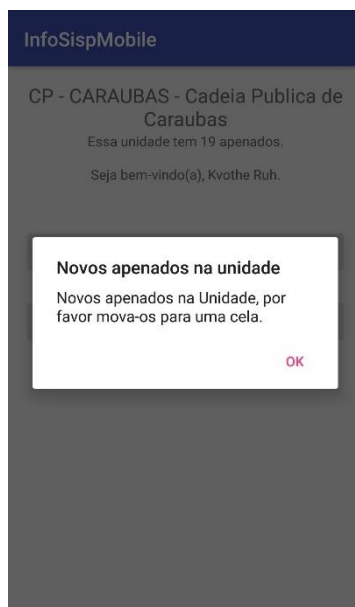
Figura 12 - Tela para listagem de apenados no INFOSISP Mobile.



Fonte: Autoria Própria.

Quando novos apenados chegam à unidade, um aviso é mostrado assim que o usuário acessa o sistema. Esse aviso é mostrado na Figura 13.

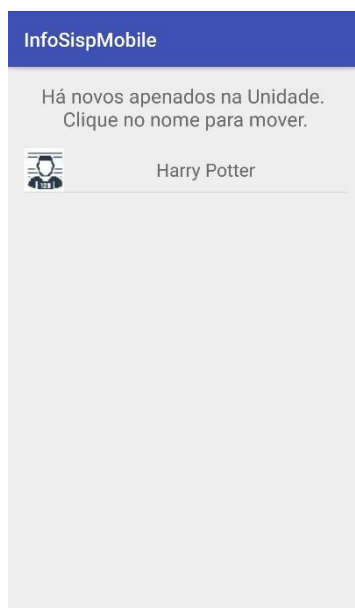
Figura 13 - Aviso de novos apenados na unidade.



Fonte: Autoria Própria.

Ao apertar o botão *ok*, o usuário é levado a uma tela com a lista dos apenados que chegaram recentemente à unidade, como mostra a Figura 14. Ao escolher o apenado que deseja mover, uma janela semelhante à Figura 11 aparecerá.

Figura 14 - Tela para listagem dos novos apenados na unidade.



Fonte: Autoria Própria.

Foram apresentadas, neste capítulo, as telas e funcionalidades do INFOSISP *Mobile*. No capítulo seguinte, serão apresentados os testes e a validação do sistema, de acordo com o modelo escolhido para este.

4 TESTES E VALIDAÇÃO

Para a coleta de dados, foi proposto um questionário composto por perguntas feitas baseadas no modelo TAM. O questionário foi realizado a partir de uma ferramenta para pesquisas *on-line*. A análise dos dados foi feita com base em estatística descritiva que também será comentada das próximas seções.

A pesquisa está fundamentada em perguntas, direcionadas ao usuário final do sistema, criadas com base no modelo TAM, onde o usuário terá duas opções de resposta: “sim” ou “não”.

4.1 Instrumento de Procedimento de Coleta de dados

O trabalho se propõe a validar o *INFOSISP Mobile*, um aplicativo de apoio às Unidades Prisionais do RN. O questionário foi organizado através da ferramenta *on-line Survey Monkey* (SURVEY, 2017). Com a ferramenta criou-se um questionário, sendo impresso e distribuído entre os usuários testadores para que pudessem respondê-lo. Os usuários serão questionados dezoito vezes, onde nove dessas questões são para avaliar a Utilidade Percebida e as outras nove para avaliar a Facilidade de Uso Percebida.

Os questionários referentes à Utilidade Percebida e a Facilidade de Uso Percebida, utilizados nesse trabalho, estão disponibilizados no Apêndice C.

Ao final das questões que envolvem o modelo TAM, foram solicitadas outras duas respostas, onde de forma livre, os usuários poderiam descrever melhorias e sugestões para o *INFOSISP*, uma de Utilidade Percebida e outra sobre Facilidade de Uso Percebida, conforme mostra a Figura 15.

Figura 15 - Perguntas sobre melhorias e sugestões para o *INFOSISP Mobile*.

10. Quais são suas recomendações para facilitar o uso do *INFOSISP Mobile*?



20. Quais são suas recomendações para melhorar a utilização do *INFOSISP Mobile*?



Fonte: Autoria Própria.

4.2 Análise dos Dados

Dos vinte e dois agentes penitenciários que trabalham na Cadeia Pública de Caraúbas (CPC), dez utilizaram o aplicativo e responderam ao questionário.

Apesar de a ferramenta *online Survey Monkey* (2017) fornecer a opção de análise de resultados, para a análise desse trabalho, foi decidido coletar e fazer a própria análise.

Para a análise dos dados coletados, foi utilizada uma regra de três simples, como mostra a Equação 1, para saber a porcentagem de concordância em cada pergunta.

Equação 1 - Cálculo da Porcentagem de Concordância.

$$Porcentagem = \frac{Quantidade\ de\ respostas\ "sim"}{Quantidade\ total\ de\ respostas} \times 100$$

Fonte: Autoria Própria.

4.3 Análise dos Resultados

Neste tópico será apresentado a análise do tratamento realizado a partir dos resultados obtidos com a pesquisa.

4.4.1 Utilidade Percebida (UP)

A Tabela 3 apresenta a análise do fator Utilidade Percebida.

Tabela 3 - Análise do fator Utilidade Percebida.

VARIÁVEL	Quantidade de respostas "Sim"	Quantidade de respostas "Não"	Quantidade de respostas em branco	Porcentagem
UP1	10	0	0	100%
UP2	10	0	0	100%
UP3	9	0	1	90%
UP4	10	0	0	100%
UP5	10	0	0	100%
UP6	10	0	0	100%
UP7	9	1	0	90%
UP8	10	0	0	100%
UP9	10	0	0	100%
Total	88	1	1	97,78%

Fonte: Autoria Própria.

A partir da Tabela 3, pode-se afirmar que existe uma grande concordância quanto a Utilidade Percebida.

4.4.2 Facilidade de Uso Percebida (FUP)

A Tabela 4 apresenta a análise do fator Facilidade de Uso Percebida.

Tabela 4 - Análise do fator Facilidade de Uso Percebida.

VARIÁVEL	Quantidade de respostas “Sim”	Quantidade de respostas “Não”	Quantidade de respostas em branco	Porcentagem
FUP1	10	0	0	100%
FUP2	10	0	0	100%
FUP3	10	0	0	100%
FUP4	10	0	0	100%
FUP5	9	1	0	90%
FUP6	4	6	0	40%
FUP7	10	0	0	100%
FUP8	10	0	0	100%
FUP9	10	0	0	100%
Total	83	7	0	92,22%

Fonte: Autoria Própria.

A partir da Tabela 4, pode-se afirmar que existe uma boa concordância quanto a Utilidade Percebida. Percebe-se que apenas na questão UP6 a maioria dos testadores tem uma opinião negativa.

Analisando os resultados obtidos, podemos afirmar que o aplicativo *INFOSISP Mobile* teve uma boa aceitação pelos usuários que o testaram tanto quanto ao uso quanto à sua facilidade de uso percebida.

5 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

O objetivo de desenvolver um aplicativo para dispositivos móveis que auxiliasse nas atividades dos agentes penitenciários do Sistema Penitenciário do Rio Grande do Norte, com os testes feitos na Cadeia Pública de Caraúbas, foi alcançado. O sistema proporciona aos seus usuários a facilidade e a agilidade pretendidas para a movimentação internas dos apenados.

O uso de um processo de desenvolvimento de *software* adequado se mostrou eficiente para o desenvolvimento de um *software* funcional, que atendesse aos requisitos e entregasse um bom produto aos usuários finais. Para satisfazer as necessidades desse projeto, foi utilizado o Modelo em Cascata, já que os requisitos estavam bem definidos desde o começo do projeto e o tempo para entrega era limitado.

Os usuários testadores se mostraram interessados pelo *software* e, até, ávidos para passar a utilizá-lo nas atividades do que desempenham. Com base nestes resultados, pode-se afirmar que o objetivo do INFOSISP *Mobile* foi alcançado.

5.1 Trabalhos futuros

Como trabalho futuro podemos citar a integração com o sistema proposto por Penha (2017), compartilhando o banco de dados e, por conseguinte, as informações relevantes a ambos os *softwares*.

Com relação às funcionalidades do sistema, os usuários testadores sugeriram que fosse implementado uma opção para buscar um apenado pelo nome e poder visualizar suas informações. Também foi sugerido o incremento de algumas listagens como o de apenados transferidos da unidade e o local de transferência de apenados que receberam alvará de soltura e dos que vieram a óbito.

REFERÊNCIAS

AMBLER, S.W. **Modelagem Ágil: Práticas eficazes para a Programação eXtrema e o Processo Unificado**. Bookman. 2004.

ANDROID. **Activity**. Disponível em <
<https://developer.android.com/reference/android/app/Activity.html>>. Data de acesso: 11 maio 2017

BOEHM, B. W. . **A spiral modelo f software development and enhancement**. IEEE Computer. DAVIS, F. D. **Perceived usefulness, perceived ease of use, and user acceptance of information technology**. MIS Quarterly, Minneapolis (MN), 1989.

DATE, C. J. **Uma Introdução a Sistemas de Banco de Dados**. 8. ed. Traduzido por Daniel Vieira. Rio de Janeiro: Elsevier, 2003.

DAVIS, F. D. **Perceived usefulness, perceived ease of use, and user acceptance of information technology**. MIS Quarterly, Minneapolis (MN), v.13, n.3, p. 319-339, 1989.

DAVIS, F. D.; BAGOZZI, R. P.; WARSHAW, P. R. **User Acceptance of computer technology: a comparison of two theoretical models**. Manage Sci., New York, v. 35, n. 8, 1989.

ELMASRI, Ramez; NAVATHE, Ehamkant. **Sistemas de Banco de Dados**. 6. ed. Traduzido por Daniel Vieira. São Paulo: Pearson Addison Wesley, 2011.

GARLAN, D. **Software architecture: a roadmap**. In: Proceedings of The Conference on The Future of Software Engineering, 2000.

GARLAN, D., PERRY, D. **Introduction to the Special Issue on Software Architecture**. In: IEEE Transactions on Software Engineering, v. 21, April 1995.

MACHADO, F. N. R. **Banco de Dados Projeto e Implementação**. Editora Érica, 2ª Edição, 2008.

OLIVEIRA, Manoel Marisergio Alves de. **Um estudo comparativo de desempenho entre técnicas de persistências de dados, no contexto de uma aplicação web**. Monografia (Graduação em Ciência da Computação), Departamento de Ciências Exatas e Naturais, Universidade Federal Rural do Semi-árido, Mossoró-RN, 2013.

OLIVEIRA, P. R. D. **Uma avaliação de metodologias para validação de software**. 2015. 68 f. Monografia (Graduação em Ciência da Computação) - Universidade Federal Rural do Semi-Árido, Mossoró, 2015.

PENHA, A.K.C. **Sistema De Informação Para Unidades Prisionais Do Estado Do Rio Grande Do Norte – INFOSISP**. 2017. Monografia (Graduação em Ciência da Computação) - Universidade Federal Rural do Semi-Árido, Mossoró, 2017.

PIRES, P. J; COSTA FILHO, B. A. **Fatores do Índice de Prontidão à Tecnologia (TRI) como Elementos Diferenciadores entre Usuários e Não Usuários de Internet Banking e como Antecedentes do Modelo de Aceitação de Tecnologia (TAM)**. RAC, Curitiba, v. 12, n. 2, p. 429-456, Abr./Jun. 2008

PMBOK 2008, **Project Management Body of Knowledge**. Project Management Institute, 4ª edição, 2008.

PRESSMAN, R.S. **Engenharia de Software**, Uma abordagem Profissional. Editora McGraw-Hill, 6ª Edição, 2006.

SALEH, A. M. **Adoção de tecnologia: um estudo sobre a adoção de software livre nas empresas**. 2004. 149 f. Dissertação (Mestrado em Administração) - Faculdade de economia, administração e contabilidade, Universidade de São Paulo, São Paulo. 2004.

SECRETARIA DE ESTADO JUSTIÇA E CIDADANIA (SEJUC). Disponível em: <<http://www.sejuc.rn.gov.br/Index.asp>> Acesso em: 20 de abril de 2017.

SOMMERVILLE, I. **Engenharia de Software**. Pearson Addison Wesley. 2007.

SOUZA NETO, Oscar Nogueira de. **Análise comparativa das metodologias de desenvolvimento de softwares tradicionais e ágeis**. 2004. 74 f. Monografia (Graduação em Ciência da Computação), Centro de Ciências Exatas e Tecnologia, Universidade da Amazônia, Belém-PA, 2004.

SURVEY. <https://pt.surveymonkey.com/home/>, **Pesquisa de Avaliação do INFOSISP**, realizada em Abril, 2017.

APÊNDICE A – ESPECIFICAÇÃO DE CASOS DE USO

Tabela 5 - Caso de uso Mover Apenado.

Nome do caso de uso	Cadastrar Unidade Prisional
Objetivo	O usuário move o apenado de cela.
Ator Principal	Agente Penitenciário
Pré-condições	O usuário deve ter efetuado o <i>login</i> no sistema.
Fluxo Principal	P1.1. O usuário escolhe a opção “MOVER APENADO”. P1.2. O sistema mostra uma lista com os pavilhões da unidade. P1.3. O usuário clica sobre o pavilhão que o apenado está atualmente. P1.4. O sistema mostra uma lista com os tipos de cela. P1.5. O usuário clica sobre o tipo de cela em que o apenado está atualmente. P1.6. O sistema mostra uma lista com as celas do tipo escolhido em < P1.5 >. P1.7. O usuário clica sobre a cela em que o apenado está atualmente. P1.8. O sistema mostra uma lista com todos os apenados que se encontram na sela escolhida em < P1.7 >. P1.9. O usuário clica sobre o nome do apenado que deseja mover. P1.10. O sistema mostra uma lista com todas as celas da unidade. P1.11. O usuário escolhe uma das celas e aperta em “OK”.
Pós-condições	Ao término desta ação, a mudança de cela do apenado deve ter sido efetuada com sucesso.

Fonte: Autoria Própria

Tabela 6 - Caso de uso Listar Apenados.

Nome do caso de uso	Editar Unidade Prisional
Objetivo	O usuário lista os apenados já cadastrados no sistema.
Ator Principal	Agente Penitenciário
Pré-condições	O usuário deve ter efetuado o <i>login</i> no sistema.
Fluxo Principal	P1.1. O usuário escolhe a opção “LISTAR APENADOS”. P1.2. O sistema mostra uma tela que contém a listagem dos apenados já cadastrados.
Fluxo de Exceção	Ao término desta ação, o usuário poderá ter acesso a todos os apenados cadastrados do sistema de acordo com sua unidade de lotação.

Fonte: Autoria Própria.

APÊNDICE B – ESPECIFICAÇÃO DAS TABELAS DO DIAGRAMA DE ENTIDADE E RELACIONAMENTO

Figura 16- Tabelas de Diagrama de Entidade e Relacionamento detalhadas.

usuario - id INT - matricula VARCHAR(20) - nome VARCHAR(50) - login VARCHAR(15) - senha VARCHAR(35) - unidadeLotacao INT - permissoes VARCHAR(10) - fotoUsuario VARCHAR(100) Indexes 	apenado - id INT - nomeApenado VARCHAR(50) - apelido VARCHAR(50) - sexo VARCHAR(15) - endereco VARCHAR(50) - cidade VARCHAR(50) - estado VARCHAR(50) - telefone VARCHAR(20) - sinais_identificados VARCHAR(100) - nome_pai VARCHAR(50) - nome_mae VARCHAR(50) - cpfApenado VARCHAR(20) - rgApenado VARCHAR(15) - nascimentoApenado DATE - naturalidade VARCHAR(30) - nacionalidade VARCHAR(30) - grau_instrucao VARCHAR(20) - profissao VARCHAR(20) - situacao_economica VARCHAR(20) - estado_civil VARCHAR(20) - companheiro VARCHAR(50) - qtd_filhos VARCHAR(10) - provisoria VARCHAR(20) - entrada_unidade DATE - artigo VARCHAR(20) - local_delito VARCHAR(20) - num_processo VARCHAR(20) - oficio VARCHAR(20) - pena VARCHAR(20) - saida_unidade DATE - fugas VARCHAR(200) - recapturado VARCHAR(200) - faltas_disciplinares VARCHAR(300) - regime_inicial VARCHAR(45) - comportamento VARCHAR(30) - local_transferencia VARCHAR(30) - motivo VARCHAR(100) - digitais VARCHAR(100) - observacao VARCHAR(500) - isolamento_rdd VARCHAR(20) - fotoApenado VARCHAR(50) - cela VARCHAR(10) Indexes PRIMARY cpfApenado_UNIQUE idUnidade_idx	unidade - id INT - nomeUnidade VARCHAR(100) - cidade VARCHAR(50) - qtPavilhao VARCHAR(10) - qtdCelaN VARCHAR(10) - qtdCelaT VARCHAR(10) - qtdCelaI VARCHAR(10) - qtdCelaE VARCHAR(10) - capacidadeCela VARCHAR(10) Indexes PRIMARY
visitante - id INT - nomeVisitante VARCHAR(100) - cpfVisitante VARCHAR(45) - rgVisitante VARCHAR(45) - nascimentoVisitante DATE - apenado INT - parentesco VARCHAR(45) - data_emissao DATE - nome_pai VARCHAR(45) - nome_mae VARCHAR(45) - naturalidade VARCHAR(45) - endereco VARCHAR(45) - cidade VARCHAR(45) - estado VARCHAR(45) - fotoVisitante VARCHAR(50) Indexes PRIMARY cpfVisitante_UNIQUE idApenado_idx		cela - idCela VARCHAR(10) - idApenado INT - idUnidade INT Indexes idApenado_idx idUnidade_idx
visita - id INT - idVisitante INT - idApenado INT - data DATE - tipoVisita VARCHAR(15) - observacao VARCHAR(100) Indexes PRIMARY idApenado_idx idVisitante_idx		

Fonte: Autoria Própria

APÊNDICE C – QUESTIONÁRIOS UTILIDADE PERCEBIDA (UP) E FACILIDADE DE USO PERCEBIDA (FUP).

Figura 17- Construtos de UP e FUP.

Avaliação do InfoSisp Mobile	Avaliação do InfoSisp Mobile
Construto sobre a Utilidade Percebida	Construto sobre a Facilidade de Uso Percebida
1. Utilizar o InfoSisp Mobile melhora meu desempenho no trabalho?	1. Utilizar o InfoSisp Mobile é agradável?
☐ Sim	☐ Sim
☐ Não	☐ Não
2. Utilizar o InfoSisp Mobile aumenta minha produtividade no trabalho?	2. No InfoSisp Mobile eu sempre sei onde estou e onde quero chegar?
☐ Sim	☐ Sim
☐ Não	☐ Não
3. Utilizar o InfoSisp Mobile melhora minha eficiência no trabalho?	3. O InfoSisp Mobile tem uma interação compreensível e clara?
☐ Sim	☐ Sim
☐ Não	☐ Não
4. Utilizar o InfoSisp Mobile torna o meu trabalho mais fácil?	4. No InfoSisp Mobile é fácil encontrar a informação que desejo?
☐ Sim	☐ Sim
☐ Não	☐ Não
5. Utilizar o InfoSisp Mobile reduz o tempo gasto em atividades improdutivas?	5. O InfoSisp Mobile possui uma interface atraente?
☐ Sim	☐ Sim
☐ Não	☐ Não
6. Utilizar o InfoSisp Mobile me poupa tempo para outras atividades?	6. É necessário treinamento para utilizar o InfoSisp Mobile?
☐ Sim	☐ Sim
☐ Não	☐ Não
7. Utilizar o InfoSisp Mobile supre as necessidades relacionadas ao trabalho?	7. Os recursos de navegação são fáceis de encontrar?
☐ Sim	☐ Sim
☐ Não	☐ Não
8. Utilizar o InfoSisp Mobile me dá maior controle sobre os dados referentes ao meu trabalho?	8. Acho necessário muito esforço para utilizar o InfoSisp Mobile?
☐ Sim	☐ Sim
☐ Não	☐ Não
9. De forma geral, o InfoSisp Mobile é útil para o meu trabalho?	9. De forma geral, eu acho o InfoSisp Mobile fácil de utilizar?
☐ Sim	☐ Sim
☐ Não	☐ Não
10. Quais são as suas recomendações para facilitar o uso do InfoSisp Mobile?	10. Quais são as suas recomendações para facilitar o uso do InfoSisp Mobile?

Fonte: Autoria Própria.