

UNIVERSIDADE VEIGA DE ALMEIDA - UVA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

**COMPUTAÇÃO EM NUVEM – ANÁLISE PRÁTICA DE UMA
BIBLIOTECA VIRTUAL NA NUVEM**

MAYARA DE LACERDA CABRAL

RIO DE JANEIRO

2015

UNIVERSIDADE VEIGA DE ALMEIDA - UVA

MAYARA DE LACERDA CABRAL

Monografia apresentada ao curso de ciência da computação da Universidade Veiga de Almeida, como requisito parcial para obtenção do título de bacharel em ciência da computação.

Orientador: Adriana Aparício Sicsu Ayres do Nascimento

**COMPUTAÇÃO EM NUVEM – ANÁLISE PRÁTICA DE UMA
BIBLIOTECA VIRTUAL NA NUVEM**

RIO DE JANEIRO

2015



**UNIVERSIDADE VEIGA DE ALMEIDA - UVA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

MAYARA DE LACERDA CABRAL

**COMPUTAÇÃO EM NUVEM – ANÁLISE PRÁTICA DE UMA
BIBLIOTECA VIRTUAL NA NUVEM**

Monografia apresentada como
requisito final à conclusão do curso em
Bacharel em Ciência da Computação.

APROVADA EM:

CONCEITO: _____

BANCA EXAMINADORA

PROF. ADRIANA APARÍCIO SICSU AYRES DO NASCIMENTO
ORIENTADOR

PROF.

PROF.

Coordenação de Ciência da Computação

Rio de Janeiro

"Ninguém é tão sábio que nada tenha para aprender, nem tão tolo que nada tenha pra ensinar."

Blaise Pascal

AGRADECIMENTOS

Agradeço primeiramente aos meus pais, Alexandre e Nádia Cabral, que sempre me apoiaram e não mediram esforços para que eu chegasse até esta etapa da minha vida, e ao meu noivo Thiago Felix pela compreensão e paciência nesse período tumultuado da minha vida e por estar sempre ao meu lado me ajudando e dando forças .

RESUMO

O mercado de dispositivos móveis vem apresentando um forte crescimento nos últimos anos. Aliado a esse fato está a evolução das tecnologias de comunicação, permitindo o acesso à internet móvel de forma mais rápida e barata. Em virtude disso, a computação em nuvem está cada vez mais presente no cotidiano das pessoas e empresas. Muitas atividades que dependiam de aplicações e/ou infraestrutura de alto custo de implantação e manutenção, agora podem ser acessadas de forma mais simples, transparente e a um custo reduzido. Uma dessas finalidades é a leitura de livros em formato digital, que está se tornando cada vez mais comum graças, também, à conscientização ecológica. Atualmente, existem diversas aplicações que disponibilizam esse tipo de serviço, porém observou-se que nenhuma delas oferecia a possibilidade de armazenamento remoto de arquivos, sincronização entre dispositivos, organização dos mesmos em categorias e, mais importante, que fosse gratuita. Dessa forma, a proposta deste trabalho é o desenvolvimento de uma aplicação de biblioteca virtual que utilize os conceitos de computação em nuvem, bem como a análise e o entendimento dos seus conceitos, empregabilidade, custos e possibilidades.

Palavras chave: Computação em nuvem, biblioteca virtual, dispositivos móveis, livros digitais, custos, possibilidades.

ABSTRACT

The market of mobile devices has been growing up quickly in the last few years. Allied to that fact is the evolving of communication technologies, which allows mobile access to the internet in a cheaper and faster way. Because of that, cloud computing is getting even more present in the people's and companies' day life activities. Many activities that depended on high cost applications and/or infrastructure can now be accessed in a simpler, more transparent and cheaper way. One of these functionalities is the ability to read digital books, which is becoming even more common thanks to, among others, the ecological consciousness. Nowadays, there are several applications that provide this kind of service. However none of them offer the possibility of remote storage, device synchronization, organization in categories and, more important, free access. Therefore, the goal of this work is the development of a virtual library application that relies on cloud computing concepts, as well as the analysis and understanding of its concepts, impregnability, costs and possibilities.

Keywords: Cloud computing, virtual library, mobile devices, digital books, costs, possibilities.

LISTA DE ILUSTRAÇÕES

| | |
|--|----|
| Figura 1 - A computação em nuvem e os tipos de usuários e serviços. | 13 |
| Figura 2 - Ambiente de trabalho em uma sala de mainframe na década de 60. | 16 |
| Figura 3 - Exemplo de conteúdos na nuvem que podem ser acessadas através de um dispositivo móvel. | 18 |
| Figura 4 - Modelos de Implantação. | 19 |
| Figura 5 - Nuvem pública. | 19 |
| Figura 6 - Nuvem privada. | 20 |
| Figura 7 - Nuvem comunitária. | 21 |
| Figura 8 - Nuvem híbrida. | 21 |
| Figura 9 - Modelos de serviço na nuvem. | 23 |
| Figura 10 - Exemplo de modelo IaaS. | 24 |
| Figura 11 - Exemplo de uma infraestrutura de PaaS. | 26 |
| Figura 12 - Estrutura do modelo SaaS. | 27 |
| Figura 13 - Abrir mão do controle total dos seus dados em busca de novas oportunidades? .. | 29 |
| Figura 14 - Privacidade de dados. | 31 |
| Figura 15 - Segurança na nuvem. | 32 |
| Figura 16 - Desafio para o futuro. | 33 |
| Figura 17 - Diagrama de entidade-relacionamento. | 37 |
| Figura 18 - Diagrama de caso de uso da interface web. | 38 |
| Figura 19 - Diagrama de caso de uso do aplicativo móvel. | 50 |
| Figura 20 - Diagrama de classes. | 58 |
| Figura 21 - Gráfico de participação no mercado de sistemas operacionais mobile. | 60 |
| Figura 22 - Interface do Android Studio. | 61 |
| Figura 23 - Emulador do Android Studio. | 62 |
| Figura 24 - Interface do NetBeans 8.0.2. | 63 |
| Figura 25 - Interface do MySQL Workbench. | 64 |
| Figura 26 - Interface do MySQL. | 64 |
| Figura 27 - Esquema do banco de dados. | 65 |
| Figura 28 - Permissões de usuário do MySQL. | 65 |
| Figura 29 - Redirecionamento de portas TCP/UDP. | 66 |
| Figura 30 - Trecho do código do ServCadUsuario. | 67 |
| Figura 31 - Trecho do código do ServLogin. | 68 |
| Figura 32 - Trecho do código do ServLoginWeb. | 68 |
| Figura 33 - Trecho do código do ServCadLivro. | 69 |
| Figura 34 - Rascunho da tela “Meu Perfil”. | 70 |
| Figura 35 - Rascunho da tela “Meus Livros”. | 70 |
| Figura 36 - Rascunho da tela “Minhas Categorias”. | 71 |
| Figura 37 - Rascunho da tela “Novo Livro”. | 71 |
| Figura 38 - Rascunho da tela “Nova Categoria”. | 72 |
| Figura 39 - Trecho de código da tela de login do aplicativo móvel. | 73 |
| Figura 40 - Layout da tela de login, dentro do Android Studio. | 74 |
| Figura 41 - Método onCreateView da tela de categorias. | 75 |
| Figura 42 - Função que carrega o livro selecionado. | 76 |
| Figura 43 - Tela inicial do aplicativo móvel. | 77 |
| Figura 44 - Tela de cadastro de usuário do aplicativo móvel. | 78 |
| Figura 45 - Tela do menu principal. | 79 |
| Figura 46 - Tela de lista de categorias. | 80 |

| | |
|---|----|
| Figura 47 - Tela de lista de livros da categoria “Para Ler” . | 81 |
| Figura 48 - Tela de leitura do livro em formato epub. | 82 |
| Figura 49 - Tela inicial da interface web. | 83 |
| Figura 50 - Tela de perfil do usuário. | 84 |
| Figura 51 - Tela de livros do usuário. | 85 |
| Figura 52 - Tela de cadastro de livros. | 86 |
| Figura 53 - Tela de categorias do usuário. | 87 |
| Figura 54 - Tela de cadastro de categorias. | 88 |

LISTA DE TABELAS

| | |
|--|----|
| Tabela 1 - Caso de Uso Cadastrar Usuário..... | 39 |
| Tabela 2 - Caso de Uso Alterar Dados Pessoais..... | 40 |
| Tabela 3 - Fazer Login..... | 41 |
| Tabela 4 - Caso de Uso Consultar Livros..... | 42 |
| Tabela 5 - Caso de Uso Excluir Livro. | 43 |
| Tabela 6 - Caso de Uso Upload de Livro. | 44 |
| Tabela 7 - Caso de Uso Editar Livro. | 45 |
| Tabela 8 - Caso de Uso Consultar Categorias. | 46 |
| Tabela 9 - Caso de Uso Excluir Categoria. | 47 |
| Tabela 10 - Caso de Uso Criar Categoria. | 48 |
| Tabela 11 - Caso de Uso Consultar Livros de uma Categoria. | 49 |
| Tabela 12 - Caso de Uso Web Fazer Login..... | 51 |
| Tabela 13 - Caso de Uso Web Cadastrar Usuário. | 52 |
| Tabela 14 - Caso de Uso Web Visualizar Perfil..... | 53 |
| Tabela 15 - Caso de Uso Web Consultar Livros. | 54 |
| Tabela 16 - Caso de Uso Web Consultar Categorias..... | 55 |
| Tabela 17 - Caso de Uso Web Consultar Livros de uma Categorias. | 56 |
| Tabela 18 - Caso de Uso Web Ler Livro..... | 57 |

LISTA DE SIGLAS

HD – Disco rígido (do inglês, *hard disk*)

IDE – Ambiente de desenvolvimento integrado (do inglês, *integrated development environment*)

TI – Tecnologia da informação

SLA – Acordo de nível de serviço (do inglês, *service level agreement*)

IaaS – Infraestrutura como serviço (do inglês, *infrastructure as service*)

PaaS – Plataforma como serviço (do inglês, *platform as service*)

SaaS – *Software* como serviço (do inglês, *software as service*)

SaaS – *Software* como serviço (do inglês, *software as service*)

APM – Plataforma de gerenciamento de aplicação (do inglês, *application platform management*)

SOA – Arquitetura orientada a serviço (do inglês, *service-oriented architecture*)

ASP – Provedor de serviços de aplicação (do inglês, *application service provider*)

NSA – Agência de segurança nacional (do inglês, *national security agency*)

RF – Requisitos funcionais

RNF – Requisitos não funcionais

SDK – Kit de desenvolvimento de *software* (do inglês, *software development kit*)

IP – Protocolo de internet (do inglês, *internet protocol*)

TCP – Protocolo de controle de transmissão (do inglês, *transmission control protocol*)

UDP – Protocolo de datagrama de usuário (do inglês, *user datagram protocol*)

SUMÁRIO

| | | |
|----------|---|----|
| 1. | INTRODUÇÃO..... | 13 |
| 1.1. | OBJETIVO | 14 |
| 1.2. | JUSTIFICATIVA..... | 14 |
| 1.3. | ORGANIZAÇÃO DO TRABALHO | 14 |
| 2. | COMPUTAÇÃO EM NUVEM | 16 |
| 2.1. | MODELOS DE IMPLANTAÇÃO | 18 |
| 2.1.1. | NUVEM PÚBLICA | 19 |
| 2.1.2. | NUVEM PRIVADA..... | 20 |
| 2.1.3. | NUVEM COMUNITÁRIA | 20 |
| 2.1.4. | NUVEM HÍBRIDA | 21 |
| 2.1.5. | APLICAÇÃO..... | 21 |
| 2.2. | VANTAGENS E DESVANTAGENS | 22 |
| 2.2.1. | VANTAGENS | 22 |
| 2.2.2. | DESVANTAGENS | 22 |
| 2.3. | MODELOS DE SERVIÇOS | 23 |
| 2.3.1. | INFRAESTRUTURA COMO SERVIÇO (IAAS) | 23 |
| 2.3.2. | PLATAFORMA COMO SERVIÇO (PAAS)..... | 25 |
| 2.3.3. | SOFTWARE COMO SERVIÇO (SAAS) | 27 |
| 2.3.4. | APLICAÇÃO..... | 28 |
| 2.4. | SEGURANÇA DA INFORMAÇÃO..... | 28 |
| 2.4.1. | AMEAÇAS E OPORTUNIDADES | 29 |
| 2.4.2. | PRIVACIDADE | 30 |
| 2.4.2.1. | DIVULGAÇÃO DE INFORMAÇÕES | 31 |
| 2.4.2.2. | SOLUÇÕES DE PRIVACIDADE | 32 |
| 2.4.3. | SEGURANÇA | 32 |
| 2.4.4. | SUSTENTABILIDADE | 33 |
| 3. | MODELAGEM DO PROJETO..... | 35 |
| 3.1. | REQUISITOS | 35 |
| 3.1.1. | MÓDULO CLIENTE (APLICATIVO <i>MOBILE</i>)..... | 35 |
| 3.1.2. | INTERFACE WEB | 36 |
| 3.2. | ESPECIFICAÇÃO..... | 37 |
| 3.2.1. | DIAGRAMA DE ENTIDADE-RELACIONAMENTO..... | 37 |
| 3.2.2. | DIAGRAMA DE CASO DE USO | 38 |
| 3.2.2.1. | INTERFACE WEB | 38 |
| 3.2.2.2. | APLICATIVO <i>MOBILE</i> | 50 |
| 3.2.3. | DIAGRAMA DE CLASSES..... | 58 |
| | DESENVOLVIMENTO | 59 |
| 3.3. | PREMISSAS..... | 59 |
| 3.4. | TÉCNICAS E FERRAMENTAS | 60 |
| 3.4.1. | ANDROID STUDIO | 60 |
| 3.4.2. | NETBEANS..... | 62 |
| 3.4.3. | MYSQL..... | 63 |
| 3.4.4. | GLASSFISH..... | 64 |
| 3.5. | DESENVOLVIMENTO DA APLICAÇÃO | 65 |
| 3.5.1. | BANCO DE DADOS | 65 |
| 3.5.2. | SERVLET..... | 66 |
| 3.5.3. | INTERFACE WEB | 69 |
| 3.5.4. | APLICATIVO <i>MOBILE</i> | 72 |
| 3.6. | RESULTADOS OBTIDOS | 77 |
| 3.6.1. | APLICATIVO <i>MOBILE</i> | 77 |
| 3.6.2. | INTERFACE WEB | 83 |
| 4. | CONCLUSÃO | 89 |
| 4.1. | TRABALHOS FUTUROS | 89 |

| | |
|------------------------------------|-----------|
| BIBLIOGRAFIA | 90 |
| APÊNDICE - DEFINIÇÕES | 92 |

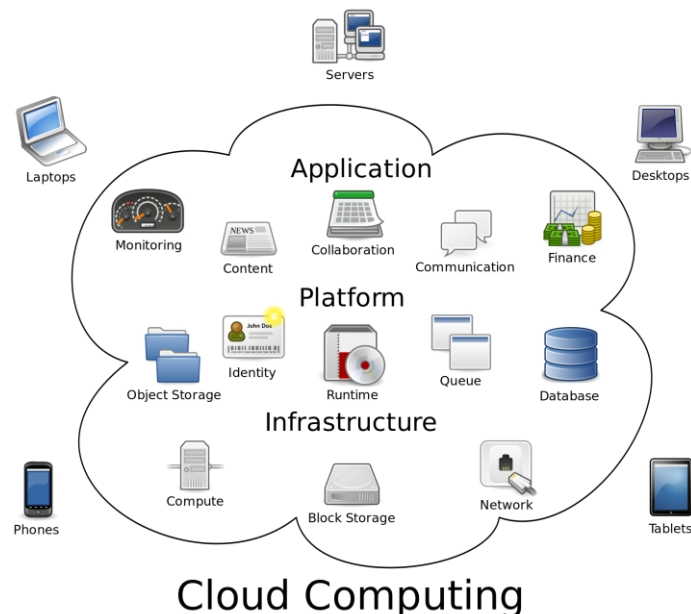
1. INTRODUÇÃO

Até pouco tempo atrás, o custo do armazenamento de arquivos remotamente e o acesso a aplicações remotas era proibitivo para o usuário doméstico, tornando esses serviços restritos às grandes empresas. Porém, com o avanço constante da tecnologia, cada vez mais vemos empresas que oferecendo os mais variados tipos de serviços na nuvem a preços cada vez mais acessíveis.

Ao armazenar fotos em serviços online, como Dropbox ou até mesmo no Facebook, ao invés de guarda-las no computador, ao utilizar um *webmail* ou ao acessar qualquer tipo de rede social, você está utilizando um serviço de computação em nuvem.

O termo computação em nuvem se refere à entrega de recursos computacionais através da Internet. Ao invés de manter arquivos no seu próprio HD ou instalar aplicativos novos sempre que houver necessidade, utilizam-se serviços *online*, de outras localidades, sem a necessidade de instalação, para armazenar seus dados e fazer uso dessas aplicações. Porém, ao expor arquivos pessoais em serviços como esses, o usuário passa a se questionar a respeito da segurança, privacidade e confiabilidade.

Figura 1 - A computação em nuvem e os tipos de usuários e serviços.
 Fonte: Wikipedia. http://en.wikipedia.org/wiki/Cloud_computing, 2015



1.1. OBJETIVO

O foco deste trabalho é a análise prática de um sistema de biblioteca virtual. Para isso, foi desenvolvida uma aplicação que consiste em uma interface *web* e um aplicativo para *smartphone*, utilizando as IDE's Android Studio e NetBeans, tendo como base o sistema operacional Android. Também foram desenvolvidos diversos recursos necessários para o correto funcionamento da aplicação em toda a sua cadeia estrutural (desde a comunicação dos dados até o armazenamento na nuvem). Para isso, também foram utilizados o banco de dados MySQL e o servidor web Glassfish.

Seu objetivo principal é demonstrar a utilização de um serviço na nuvem para o gerenciamento de uma biblioteca virtual e leitura de livros em formato digital. Nele, o usuário poderá enviar arquivos para um servidor (que também será configurado nesse projeto) e acessá-los de qualquer outro dispositivo através da sua conta.

Além disso, também são verificadas as soluções empregadas nos serviços de computação em nuvem em geral por meio de um levantamento histórico e análise das tecnologias e serviços disponíveis atualmente.

1.2. JUSTIFICATIVA

Os aplicativos disponíveis atualmente no mercado, que armazenam os arquivos dos livros na nuvem e possuem sincronização entre dispositivos, não possuem uma interface simples e amigável, bem como não permitem que o usuário organize seus livros em forma de categorias. Dessa forma, viu-se a necessidade de uma aplicação gratuita que contemplasse todas essas funcionalidades aliadas a uma interface de simples e intuitiva, de forma a fornecer uma experiência única e inovadora ao usuário.

1.3. ORGANIZAÇÃO DO TRABALHO

Este trabalho é dividido, além desta introdução, em outros quatro capítulos.

O capítulo 2 faz uma revisão histórica da origem da computação em nuvem, seus modelos de implantação, suas vantagens e desvantagens e seus modelos de serviço, além de falar também sobre a questão da segurança da informação.

No capítulo 3, é apresentada a modelagem do projeto de biblioteca virtual, bem como seus requisitos e diagramas.

No capítulo 4 é abordada a etapa de desenvolvimento do sistema, com as suas premissas, técnicas e ferramentas utilizadas e o desenvolvimento da aplicação em si.

O capítulo 5 traz a conclusão baseada em tudo que foi apresentado, bem como as sugestões para trabalhos futuros.

São utilizadas, nesta análise, termos técnicos (em português ou não) e citações diversas que tem relação direta ou indireta com o assunto abordado. Uma lista com as definições pode ser consultada no apêndice.

2. COMPUTAÇÃO EM NUVEM

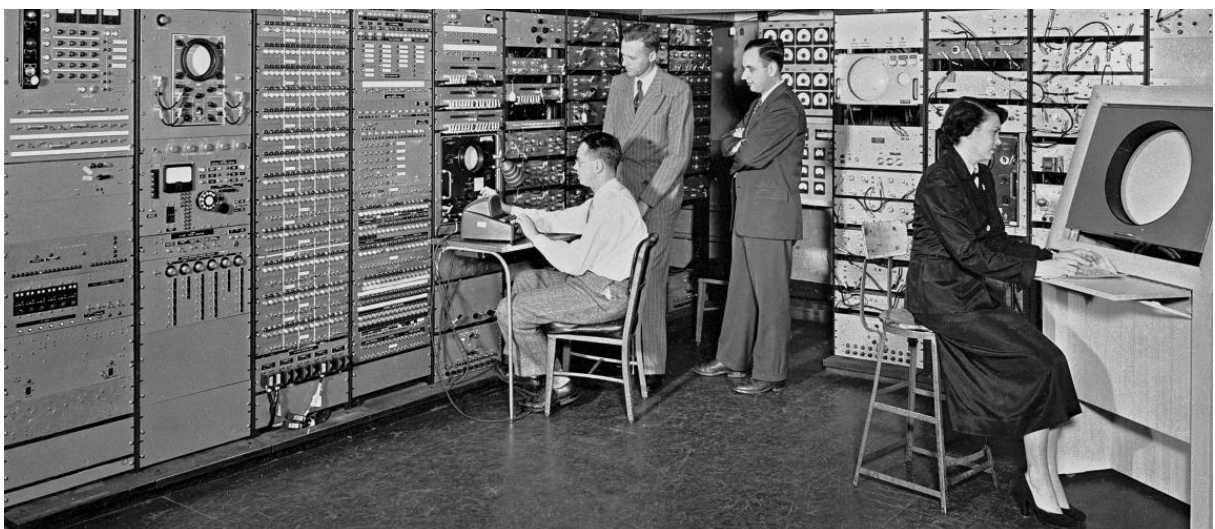
O termo computação em nuvem, do inglês *cloud computing*, baseia-se na premissa de que se deve ter a possibilidade de acessar arquivos ou mesmo utilizar ferramentas de qualquer lugar, sem a necessidade de se instalar qualquer *software* para isso, através da internet.

De acordo com o Instituto Nacional de Padrões e Tecnologia do Departamento de Comércio norte-americano:

Computação em nuvem é um modelo para permitir acesso ubíquo, conveniente e sob demanda via rede a um agrupamento compartilhado e configurável de recursos computacionais (por exemplo, redes, servidores, equipamentos de armazenamento, aplicações e serviços), que pode ser rapidamente fornecido e liberado com esforços mínimos de gerenciamento ou interação com o provedor de serviços. (NIST, 2015)

O conceito de computação em nuvem já era discutido por volta da década de 60, mas havia sido abandonado, pois as tecnologias existentes naquela época não estavam aptas para sustentar esse modelo computacional, que na época era considerado “futurista”. Na época, as empresas precisavam de uma infraestrutura de grande porte para sustentar os *mainframes* onde eram armazenados os dados e realizado todo o processamento necessário. Tal arquitetura gerava um alto custo de manutenção, tanto na infraestrutura como na mão-de-obra.

Figura 2 - Ambiente de trabalho em uma sala de mainframe na década de 60.
Fonte: Mitre. <http://www.mitre.org/about/our-history>, 2015



Apenas em 1999 a Salesforce.com desenvolveu aplicações que eram disponibilizadas através da internet. Esta foi a primeira empresa a desenvolver para nuvem. Depois do seu

grande sucesso, outras empresas passaram a investir nessa área, como a Amazon, Google, IBM e Microsoft.

No início, a grande maioria dos usuários de serviços na nuvem eram, em geral, grandes empresas que, executando aplicações em servidores na nuvem (remotamente) não seriam necessários grandes investimentos em hardware das máquinas de seus funcionários, visto que todo o processamento ocorria nos servidores. Elas também passaram a gastar menos com licenças de software, não sendo mais necessária a compra de uma licença por usuário, uma vez que eles compartilham a aplicação da nuvem. Utilizando o caso do Microsoft Office 365 para empresas, a versão para vários usuários pode ser até 35% mais barata que a versão individual (versão *Business* para um único usuário custa U\$12,50 por mês, enquanto que a versão *Enterprise E1* custa U\$8,00 por mês para cada usuário – para empresas com mais de 50 funcionários).

Com a popularização da internet e dos *smartphones*, diferentes vertentes da computação em nuvem puderam ser exploradas. Diversas empresas passaram a investir no desenvolvimento de aplicações voltadas para o usuário doméstico que até então não eram cogitadas para o mundo corporativo, como aplicações voltadas para o entretenimento. Hoje através de dispositivos móveis conectados à internet, é possível acessar e-mails, editar arquivos de texto, editar imagens, armazenar arquivos remotamente e até mesmo assistir vídeos sob demanda em qualquer lugar. Algumas empresas que oferecem esses serviços são: Youtube, que disponibiliza o envio e a visualização de vídeos através de diversos dispositivos, o Google Docs, onde é possível criar e editar arquivos de textos e planilhas através do navegador e o Sportify, que possui uma coletânea diversificada de músicas à disposição do usuário.

Figura 3 - Exemplo de conteúdos na nuvem que podem ser acessadas através de um dispositivo móvel.
Fonte: Zoom. <http://www.zoom.com.br/tablet-ipad/deumzoom/para-que-serve-como-usar-armazenamento-em-nuvem>, 2015



2.1. MODELOS DE IMPLANTAÇÃO

É possível implantar soluções utilizando computação em nuvem de maneiras diferentes, dependendo de vários fatores, tais como:

- Onde os serviços serão hospedados;
- Requisitos de segurança;
- Necessidade de compartilhar serviços;
- Necessidade de gerenciamento dos serviços;
- Nível de acesso;
- Capacidade de customização;

Existem quatro modelos principais que são determinados pelo nível de acesso desejado ao serviço: Nuvem Pública, Nuvem Privada, Nuvem Comunitária e Nuvem Híbrida.

Figura 4 - Modelos de Implantação.

Fonte: UTIC-SP. http://www.utic.sp.gov.br/paginas/coetic/Computacao_Nuvem.html, 2015



2.1.1. Nuvem Pública

Uma infraestrutura de nuvem pública é disponibilizada para o público geral e é de propriedade de um provedor de serviços terceirizado. Em uma nuvem pública, os recursos de computação são disponibilizados dinamicamente através do provedor e são fornecidos para outras organizações. Assim como ocorre em uma conta de energia, onde o “provedor” cobra pela quantidade de recursos que o “cliente” utiliza.

Esse modelo é o que possui maior custo/benefício para o cliente, uma vez que oferece a flexibilidade de disponibilizar apenas os recursos necessários e entregar todos os serviços com uma consistência de disponibilidade, resiliência, segurança e facilidade de gerenciamento. Sendo assim, para se beneficiar de uma nuvem pública, o cliente precisa aceitar o controle reduzido e o monitoramento do provedor sobre a governança e segurança da infraestrutura.

Figura 5 - Nuvem pública.

Fonte: Cloud.cio. <http://cloud.cio.gov/topics/cloud-computing-deployment-models>, 2015



2.1.2. Nuvem Privada

Uma infraestrutura de nuvem privada é operada apenas para um único cliente: o provedor dedica serviços específicos para aquele cliente, e para nenhum outro. O cliente especifica, arquiteta e controla toda a gama de recursos computacionais que o provedor fornece. Um motivo comum que leva os clientes a procurarem um serviço de nuvem privado é a capacidade de controlar e garantir seus próprios padrões de segurança dos dados.

Um cliente normalmente fará uso de uma nuvem pública utilizando conexões através de links privados e esses recursos apenas serão compartilhados internamente. Como os recursos não são compartilhados entre várias organizações, o cliente, que contratou o serviço, paga o valor total pelos recursos da nuvem, independente da quantidade que foi utilizada. Sendo assim, a organização contratante pode realocar os recursos para subáreas da mesma empresa conforme suas necessidades.

Figura 6 - Nuvem privada.

Fonte: Cloud.cio. <http://cloud.cio.gov/topics/cloud-computing-deployment-models>, 2015



2.1.3. Nuvem Comunitária

Uma infraestrutura de nuvem comunitária é contratada por um grupo de organizações em conjunto ou programas que compartilham necessidades específicas, como segurança e aspectos legais. O controle da nuvem pode ser feito pelo cliente ou pelo provedor, de acordo com o que foi combinado no contrato.

Quando organizações possuem o mesmo conjunto de requisitos e clientes, a nuvem comunitária permite a eles combinarem ferramentas e compartilharem recursos computacionais, dados e capacidades. Ao eliminar a duplicidade de sistemas similares, as organizações podem economizar dinheiro e alocar seus recursos de maneira mais eficiente.

Figura 7 - Nuvem comunitária.

Fonte: Cloud.cio. <http://cloud.cio.gov/topics/cloud-computing-deployment-models>, 2015



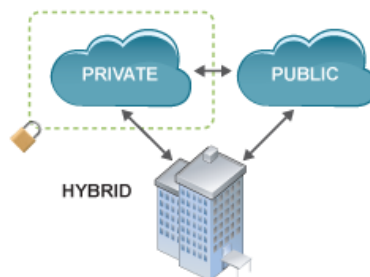
2.1.4. Nuvem Híbrida

Uma infraestrutura de nuvem híbrida abrange duas ou mais nuvens, podendo estas serem nuvens públicas, comunitárias ou privadas, com o conjunto dos serviços que são hospedados internamente ou externamente.

Os clientes geralmente não se limitam a um único modelo de implantação, mas sim preferem incorporar serviços de nuvem diferentes e sobrepostos para atingir seus requisitos específicos. Modelos de implantação híbridos são complexos e requerem um planejamento cuidadoso para serem executados e gerenciados especialmente quando é necessária a comunicação entre dois tipos diferentes de implantações em nuvem.

Figura 8 - Nuvem híbrida.

Fonte: Cloud.cio. <http://cloud.cio.gov/topics/cloud-computing-deployment-models>, 2015



2.1.5. Aplicação

A aplicação desenvolvida neste trabalho se encaixa no conceito de nuvem pública, uma vez que está disponível para o público em geral, não havendo nenhum tipo de restrição quanto à exclusividade.

2.2. VANTAGENS E DESVANTAGENS

2.2.1. Vantagens

A computação em nuvem permite que o usuário adquira apenas os serviços que são necessários a ele, a qualquer momento, eliminando o custo inicial de infraestrutura. Dessa forma, evitam-se problemas comuns da área de TI, como equipamentos se tornando obsoletos, garantir a segurança e confiabilidade do sistema. É possível adquirir ou eliminar serviços sempre que for necessário com rapidez e agilidade.

De forma resumida, podemos citar algumas dessas vantagens:

- Baixo custo de investimento inicial e de infraestrutura.
- Fácil escalabilidade de serviços e recursos computacionais.
- Adquirir apenas o necessário.
- Gerenciamento sob SLAs (Acordo de Nível de Serviço, do inglês, *Service Level Agreement*).
- Benefício ambiental (redução de emissão de gás carbono) devido ao compartilhamento eficiente de recursos computacionais por vários usuários.
- A produtividade pode ser aumentada uma vez que mais de um usuário pode trabalhar no mesmo dado simultaneamente, ao invés de esperar que ele seja salvo e liberado.
- Alta confiabilidade, devido ao uso de ambientes redundantes que torna possível a recuperação de dados em caso de um desastre.

2.2.2. Desvantagens

No entanto, tais facilidades podem custar caro. Ao invés de adquirir *hardware* e *software*, adquirem-se serviços. Dessa forma, os custos de implantação se transformam em custos operacionais constantes. Essa abordagem pode se tornar mais cara em longo prazo.

Podemos listar algumas desvantagens abaixo:

- Custos operacionais mais elevados. A computação em nuvem seria mais cara com o passar do tempo?
- Maior dependência dos provedores de serviços. É possível resolver problemas de forma rápida, mesmo com as restrições do SLA?
- Risco de se prender ao sistema ou provedor contratado. Com que facilidade você pode migrar para um outro sistema ou provedor, se for necessário?

- Seu provedor pode simplesmente decidir parar o suporte a um produto ou sistema de qual você é dependente.
- Risco potencial de segurança e privacidade ao armazenar informações em servidores de terceiros.
- Dependência de uma conexão de internet confiável.

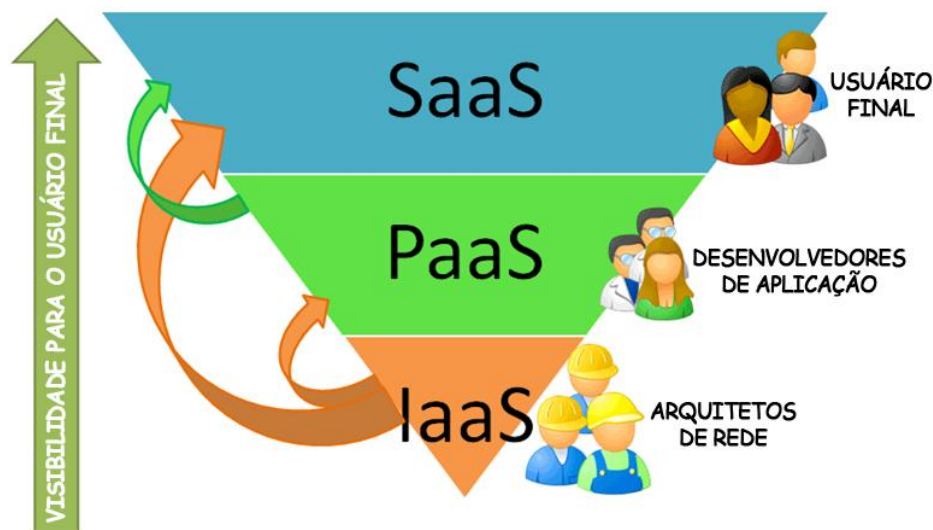
2.3. MODELOS DE SERVIÇOS

Os provedores podem oferecer seus serviços de acordo com diversos modelos fundamentais. Neste capítulo abordaremos os três modelos principais: Infraestrutura como Serviço (IaaS), Plataforma como Serviço (PaaS) e Software como Serviço (SaaS).

Figura 9 - Modelos de serviço na nuvem.

Fonte: Vitor Meriat. <http://vitormeriat.com.br/2011/07/08/modelos-de-servio-na-nuvem-iaas-paas-e-saas>,

2015



2.3.1. INFRAESTRUTURA COMO SERVIÇO (IaaS)

É o modelo de computação em nuvem que oferece recursos computacionais virtualizados através da internet. Nesse modelo, um provedor terceirizado hospeda *hardware*, *software*, servidores, armazenamento de dados e outros equipamentos de infraestrutura para os seus usuários. O provedor também hospeda as aplicações e lida com tarefas como manutenção do sistema, *backup* e planejamento de resiliência.

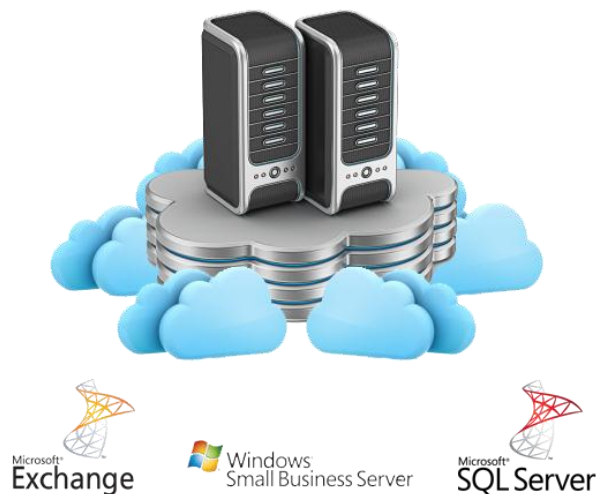
Uma plataforma de IaaS oferece alta escalabilidade de recursos que podem ser ajustados sob demanda. Isso a torna ideal para cargas de trabalho que são temporárias, experimentais ou que mudam inesperadamente.

O IaaS também possui outras características, como a automação de tarefas administrativas, escalabilidade dinâmica, virtualização de áreas de trabalho e serviços baseados em políticas específicas.

Na imagem abaixo podemos ver uma referência ao Microsoft Exchange, um exemplo de serviço que usa o modelo IaaS.

Figura 10 - Exemplo de modelo IaaS.

Fonte: Yunion. <http://www.yunion.net/en/home/infrastructure-as-a-service.html>, 2015



Os usuários são cobrados de acordo com os recursos usados. Essa cobrança normalmente é feita com base em horas, semanas ou meses. Alguns provedores também tarifam o usuário de acordo com a quantidade de espaço de máquina virtual utilizada. Esse modelo elimina o custo de implantação interna de *hardware* e *software*. Entretanto, os usuários devem monitorar seus ambientes de IaaS constantemente para evitar serem cobrados por serviços que não foram autorizados.

Devido ao fato do provedor ser dono da infraestrutura, o monitoramento e gerenciamento do sistema pode se tornar mais difícil para o usuário. Além disso, se o provedor ficar inoperante, o trabalho do usuário vai ser afetado.

Por exemplo, se um novo produto de software está sendo desenvolvido, pode ser mais eficiente em termos de custo hospedar e testar a aplicação em um provedor IaaS e assim que o software estiver testado e pronto para o lançamento, ele pode ser removido do ambiente IaaS e alocado em um servidor local para economizar ou para liberar o ambiente de desenvolvimento para outra aplicação.

Alguns dos provedores de IaaS mais conhecidos são:

- Amazon Web Services (AWS)
- Windows Azure
- Microsoft Exchange
- Google Compute Engine
- Rackspace Open Cloud
- IBM SmartCloud Enterprise
- AT&T Cloud Computing Services
- BlueBlock
- Datapipe
- HP BladeSystem Matrix

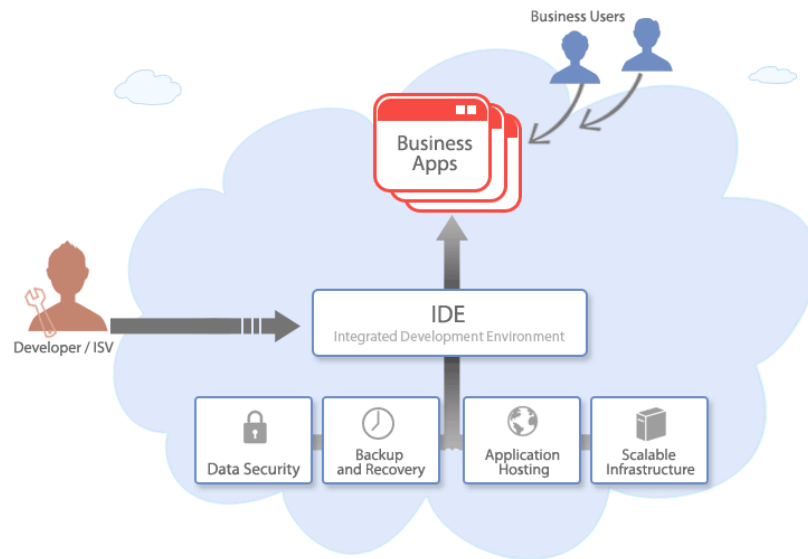
2.3.2. Plataforma como Serviço (PaaS)

É o modelo de computação em nuvem que entrega aplicações através da internet. Nesse modelo, o provedor oferece hardware e ferramentas de software, necessários para o desenvolvimento de aplicações, para seus usuários, como um serviço. Ele hospeda o hardware e software em sua própria infraestrutura. Dessa forma os usuários não precisam instalar os softwares e hardwares necessários para o desenvolvimento de novas aplicações.

PaaS nem sempre substitui toda a infraestrutura de um negócio. Ao invés disso, o usuário contrata o provedor para serviços específicos, como desenvolvimento Java ou hospedagem de aplicações. Por exemplo, implantar uma ferramenta de negócio localmente pode requerer uma equipe de TI para adquirir e instalar *hardware*, sistemas operacionais, *middleware* (como banco de dados, servidores *web*, etc), a aplicação em si, definir o acesso dos usuários ou níveis de segurança, e após isso adicionar a aplicação em um sistema de gerenciamento existente ou em uma ferramenta de gerenciamento de desempenho da aplicação (APM). A equipe de TI deve então manter todos esses recursos com o passar do tempo. Um provedor PaaS, entretanto, dá suporte a toda essa base computacional e o usuário precisa apenas iniciar uma sessão, normalmente via navegador web, e usar a plataforma de desenvolvimento.

Na imagem abaixo, podemos ver como um serviço na nuvem pode ser estruturado para servir como plataforma de desenvolvimento de aplicações.

Figura 11 - Exemplo de uma infraestrutura de PaaS.
Fonte: Zoho. <https://www.zoho.com/creator/paas.html>, 2015



A maioria das plataformas PaaS é voltada para o desenvolvimento de *software* e elas normalmente oferecem diversas vantagens aos desenvolvedores. Por exemplo, PaaS permite que os desenvolvedores façam mudanças frequentes ou atualizem funcionalidades do sistema operacional. Também ajuda no aspecto colaborativo dos projetos.

Os usuários normalmente acessam PaaS através de navegadores *web*. Os provedores então cobram por este acesso de acordo com o uso. Alguns provedores cobram um valor fixo por mês para o acesso da plataforma e os aplicativos disponíveis. É importante discutir como será realizada a cobrança com o provedor antes de adquirir o serviço, para escolher um plano de serviço que se adapte às suas necessidades.

Uma vez que os usuários passam a depender da infraestrutura e software do provedor, suas restrições podem ser um problema nos ambientes PaaS. Outros riscos associados ao PaaS são indisponibilidade do provedor, se o provedor parar de suportar determinada linguagem de desenvolvimento os usuários podem ser forçados a trocar de provedor ou da linguagem escolhida para o projeto.

Alguns dos provedores de PaaS mais conhecidos são:

- Salesforce.com
- Appcar IQ
- Mendix
- Amazon Web Services (AWS) Elastic Beanstalk

- Google App Engine
- Heroku
- CloudBees
- Cloud Foundry
- Red Hat OpenShift

2.3.3. Software como Serviço (SaaS)

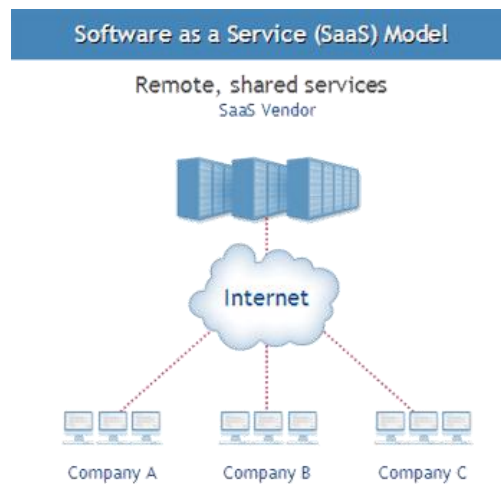
É o modelo de computação em nuvem que disponibiliza aplicações, que são hospedadas no provedor e disponibilizadas para os clientes, através da internet.

SaaS está se tornando um modelo de entrega em constante crescimento de tecnologias que dão suporte a serviços web e a arquitetura orientada a serviços (SOA). Enquanto isso, serviços de banda larga tem se tornado cada vez mais disponíveis aos usuários, dessa forma permitindo que o SaaS seja acessado em mais localidades ao redor do mundo.

Na imagem abaixo podemos ver como funciona um modelo de SaaS cujo produto (software em si) é distribuído para várias empresas.

Figura 12 - Estrutura do modelo SaaS.

Fonte: Wolters Kluwer. <http://www.wolterskluwer.com/Innovation/Innovation/Pages/saas.aspx>, 2015.



SaaS está fortemente relacionado a ASP e a modelos de entrega de software sob demanda. IDC identifica dois modelos de SaaS ligeiramente diferentes. O modelo de gerenciamento de aplicações hospedadas é similar ao ASP: o provedor hospeda *softwares* disponíveis no mercado e os disponibiliza para os usuários através da internet. No modelo de

software sob demanda, o provedor dá aos usuários acesso via rede a uma única cópia de uma aplicação criada especificamente para uma distribuição SaaS.

Os benefícios do modelo SaaS incluem:

- Fácil administração
- Atualizações automáticas e gerenciamento de modificações
- Compatibilidade: Todos os usuários estarão usando a mesma versão do software
- Fácil colaboração
- Acesso global

Alguns dos provedores de SaaS mais conhecidos são:

- Netflix
- Paypal
- McAfee
- Adobe Creative Cloud
- Salesforce
- Google Docs
- Microsoft Office 365

2.3.4. Aplicação

A aplicação desenvolvida neste trabalho se encaixa no conceito de SaaS, uma vez que trata-se de um *software* que utiliza recursos da nuvem para o seu funcionamento, incluindo o armazenamento do seu conteúdo e gerenciamento de usuários.

2.4. SEGURANÇA DA INFORMAÇÃO

A computação em nuvem tem se tornado um fenômeno social que é utilizado pela maioria das pessoas durante o seu dia a dia. E assim como todo fenômeno social importante, existem preocupações que limitam sua adoção em grande escala.

A maioria das preocupações deriva do fato de que o usuário perde o controle total dos seus dados, pois eles ficam armazenados em servidores de terceiros (no provedor do serviço). Isso ocorre quando o dono dos servidores remotos é uma pessoa ou organização que não o usuário, pois seus interesses podem apontar em direções diferentes. Por exemplo, o usuário

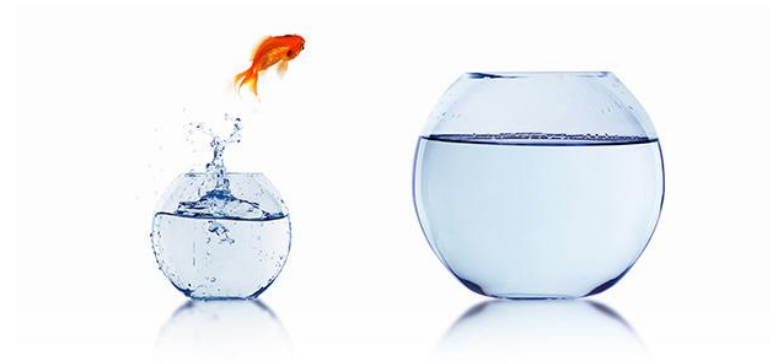
pode querer que sua informação seja mantida em sigilo, porém o dono do servidor pode querer tirar vantagem delas para seu próprio interesse.

Neste capítulo iremos abordar algumas das principais questões relacionadas à segurança e privacidade na computação em nuvem.

2.4.1. Ameaças e Oportunidades

Figura 13 - Abrir mão do controle total dos seus dados em busca de novas oportunidades?

Fonte: The Bunker. <http://www.thebunker.net/solutions/cloud-migration>, 2015



Embora muitas autoridades do assunto, como Richard Stallman (líder do projeto GNU) e Larry Ellison (fundador da Oracle), tenham alertado quanto à privacidade e segurança dos dados, a computação em nuvem continuou ganhando folego. As principais empresas de tecnologia da Europa estimaram que a nuvem foi uma prioridade em 2013 e 2014 e que o seu orçamento pode ter alcançado 30% do orçamento total previsto para a área de TI.

De acordo com o relatório de 2013 do TechInsights, as implementações na nuvem geralmente atendem ou excedem as expectativas na maioria dos modelos de serviço, como na Infraestrutura como Serviço (IaaS), Plataforma como Serviço (PaaS) e Software como Serviço (SaaS).

Muitos impeditivos da adoção em grande escala da computação em nuvem se mantêm. Dentre eles estão:

- Confiança;
- Disponibilidade dos serviços e dados;
- Segurança;
- Complexidade;
- Custos;
- Regulamentações e questões legais;

- Desempenho;
- Migração;
- Reversibilidade;
- Falta de padronização;
- Customização limitada;
- Privacidade.

A computação em nuvem também oferece muitos pontos positivos, tais como:

- Flexibilidade de infraestrutura;
- Agilidade na implantação de aplicações e dados;
- Controle de custos;
- Adaptação dos serviços às necessidades do cliente;
- Aumento da produtividade.

No início de 2010 o mercado da nuvem era dominado por softwares e serviços nos modos SaaS e IaaS, especialmente a nuvem privada. PaaS e a nuvem pública perderam espaço.

2.4.2. Privacidade

O aumento no uso de serviço de nuvem como o *Gmail* e o *Google Docs* tem trazido à tona questões de privacidade dos serviços de computação em nuvem. O provedor destes serviços se encontra em uma posição na qual, com o grande uso de seus serviços, ele detém acesso a uma multiplicidade de dados. Com isso, há o risco iminente dos dados serem expostos tanto acidentalmente quanto intencionalmente. Os defensores da privacidade criticam o modelo de nuvem por darem facilidade de controle aos servidores de hospedagem (e assim, a possibilidade de monitoramento das informações do usuário sem a permissão do mesmo).

Figura 14 - Privacidade de dados.

Fonte: Can I Retire Yet?. <http://www.caniretireyet.com/protecting-your-assets-in-a-digital-world/>, 2015.



Episódios como o que ocorreu entre a Agência de Segurança Nacional norte-americana (NSA) em conjunto com a AT&T e a Verizon, que gravaram mais de 10 milhões de ligações telefônicas entre cidadãos americanos, geraram incertezas entre os defensores da privacidade sobre os grandes poderes que são dados às companhias de telecomunicações para monitorarem as atividades dos usuários. Um provedor de serviços na nuvem pode dificultar a privacidade dos dados por causa da extensão da virtualização (máquinas virtuais) e do armazenamento de dados usado para implementar o serviço na nuvem. As operações do provedor e do usuário podem não permanecer no mesmo sistema, ou no mesmo *data center* ou até mesmo no mesmo provedor de nuvem. Isso pode levantar preocupações a respeito de jurisdição. Mesmo havendo esforços no sentido de harmonizar esta questão, provedores como a Amazon ainda oferecem ao cliente a possibilidade de escolher a região onde os seus dados ou serviços serão armazenados.

A computação em nuvem apresenta preocupações a respeito da privacidade pois o provedor dos serviços pode acessar os dados do usuário a qualquer momento. Dessa forma, é possível alterar ou até mesmo excluir dados de forma acidental ou intencional. Isso se torna uma grande preocupação entre os provedores de serviço, cujos funcionários podem divulgar informações indesejáveis de seus clientes.

2.4.2.1. Divulgação de Informações

Muitos provedores de serviços podem compartilhar informações com terceiros, caso necessário, para propósitos legais, mesmo sem a necessidade de um mandado oficial. Isso está contemplado na sua política de privacidade, que o usuário deve aceitar antes de começar a utilizar seus serviços.

Em situações onde há risco de vida e não há tempo para se esperar pela emissão de um mandado, muitos provedores de nuvem podem divulgar as informações imediatamente para a polícia. Um exemplo de política de privacidade que permite isso é a do *Dropbox*:

Fins legais. Podemos divulgar suas informações a terceiros caso determinemos que a divulgação é razoavelmente necessária para (a) cumprir a lei, (b) proteger qualquer pessoa de morte ou lesão corporal grave; (c) evitar fraude ou abuso do Dropbox ou dos nossos usuários, ou (d) proteger os direitos de propriedade do Dropbox (Dropbox, 2015).

2.4.2.2. Soluções de Privacidade

Soluções para a privacidade na computação em nuvem incluem políticas e legislações específicas, bem como a escolha do usuário de como o dado é armazenado. O provedor do serviço precisa estabelecer políticas claras e relevantes que descrevem como a informação de cada usuário da nuvem será acessada e utilizada. Além disso, os usuários podem criptografar a informação que é armazenada ou processada nos servidores, de modo a prevenir o acesso não autorizado.

2.4.3. Segurança

A efetividade e eficiência dos mecanismos de proteção tradicionais estão sendo reconsiderados, pois as características do modelo de implantação da computação em nuvem podem ser consideravelmente diferentes das arquiteturas tradicionais. Uma perspectiva alternativa sobre o tema da segurança da nuvem é que este é apenas outro caso de segurança aplicada cujos princípios que se aplicam aos modelos utilizados em *mainframes* podem ser aplicados na segurança da nuvem.

Figura 15 - Segurança na nuvem.

Fonte: Big Data Cloud Help. <http://www.bigdatacloudhelp.com/cloud-computing-security>, 2015.



A segurança relativa dos serviços de computação em nuvem é uma questão polêmica que pode dificultar sua disseminação. O controle físico do equipamento da Nuvem Privada é mais seguro do que ter o equipamento fora do seu patrimônio e sob o controle de outra pessoa. O controle físico e a habilidade de inspecionar *links* de dados e acessar portas é necessário para que se garanta que os mesmos não estão comprometidos. As preocupações que barram a adoção da computação em nuvem se dão por conta de que grande parte do setor público e privado não se sente confortável em deixar que outra empresa assuma o gerenciamento e a segurança dos seus dados, e esse é o princípio dos serviços baseados na computação em nuvem, tanto os privados quanto os públicos.

2.4.4. Sustentabilidade

Embora a computação em nuvem seja considerada uma forma de computação verde, não existe atualmente uma forma de medir quão “verde” uma metodologia é.

Figura 16 - Desafio para o futuro.

Fonte: Cloud People. <http://www.cloudpeople.com.au/cloud-computing/>, 2015.



O principal problema ambiental relacionado com a computação em nuvem é o uso da energia. De acordo com um dos diretores executivos do Greenpeace:

Estamos preocupados que esse grande aumento no uso de energia elétrica possa nos prender em fontes de energias antigas e poluentes ao invés das novas e mais limpas fontes de energias que estão disponíveis atualmente. (Phil Radford, 2013).

O Greenpeace fez uma lista com as dez maiores empresas de computação em nuvem e conseguiu convencer a grande maioria a mudar para fontes limpas de energia. No dia 15 de dezembro de 2011, o Greenpeace e o Facebook anunciaram que este último estaria adotando o uso de energia limpa em suas operações. Logo após, a Apple concordou em abolir o uso de carvão em seus datacenters até o final de 2013 e dobrou a quantidade de energia solar

produzida no seu datacenter de Maiden. Em seguida, a Salesforce também concordou em alterar 100% do seu uso para energia limpa até o final de 2020.

O uso de computação em nuvem em regiões com clima mais frio e com disponibilidade de energia renovável reduz significativamente os impactos ambientais. Por esse motivo, países que se encaixam nessa descrição, como a Finlândia, Suécia e Suíça estão atraindo a instalação de datacenters.

3. MODELAGEM DO PROJETO

O sistema possui quatro módulos básicos:

- **Cliente**

Aplicativo voltado para o usuário final. Nele, é possível criar uma conta, acessar os livros do usuário que estão armazenados no servidor, criar categorias para os livros e exibir o conteúdo dos mesmos na tela.

- **Web**

Interface *web* pela qual o usuário envia os arquivos dos livros para o servidor. Nela, o usuário também realiza o cadastro das categorias e organiza os seus livros em categorias, para facilitar o acesso à informação no módulo cliente (aplicativo *mobile*).

- **Banco de dados**

Servidor onde são armazenadas todas as informações dos usuários, bem como os arquivos dos livros digitais.

- **Web service**

Responsável pela comunicação entre o cliente (aplicativo final ou interface web) e o banco de dados.

3.1. REQUISITOS

Cada módulo do sistema possui uma série de requisitos funcionais (RF) e requisitos não funcionais (RNF).

3.1.1. Módulo Cliente (Aplicativo *Mobile*)

- **Utilizar a IDE Android Studio (RNF – implementação)**

Trata-se de uma suíte de desenvolvimento oficial do Google com total suporte ao SDK do Android e possibilidade de executar as aplicações desenvolvidas em um emulador.

- **Permitir o acesso somente de usuários registrados (RF)**

A aplicação deve oferecer ao usuário a opção de fazer *login* com a sua conta ou criar uma nova caso ele não a possua.

- **Limitar o acesso aos arquivos (RF)**

A aplicação deve restringir o acesso dos usuários somente aos arquivos próprios, não sendo possível acessar nenhum arquivo de outro usuário.

- **Fazer *download* dos livros armazenados (RF)**

A aplicação deve fazer *download* de todos os livros armazenados no servidor (que estiverem associados ao perfil do usuário) diretamente para o dispositivo.

- **Gerenciar categorias de livros (RF)**

A aplicação deve oferecer ao usuário uma forma de organizar os livros em categorias que podem ser criadas e editadas na própria aplicação. Dessa forma, deve ser possível filtrar a exibição dos livros baseada nessas categorias.

- **Exibir o conteúdo dos livros na tela (RF)**

A aplicação deve ser capaz de ler e exibir o conteúdo dos arquivos dos livros digitais na tela.

3.1.2. Interface WEB

- **Visualizar os dados de cadastro do usuário (RF)**

A interface web deve ser capaz de exibir os dados de cadastro do usuário como nick, nome, e-mail e senha. Dessas, as três últimas poderão ser editadas pela interface.

- **Visualizar lista de livros (RF)**

A interface web deve ser capaz de exibir uma lista com os livros cadastrados pelo usuário.

- **Visualizar lista de categorias (RF)**

A interface web deve ser capaz de exibir uma lista com as categorias cadastradas pelo usuário.

- **Cadastrar novos livros (RF)**

A interface web deve permitir que o usuário cadastre novos livros e faça o envio dos arquivos dos mesmos para um servidor.

3.2. ESPECIFICAÇÃO

A seguir, são ilustrados os diagramas de modelo de dados (Modelo Entidade-Relacionamento), de caso de uso e de classes.

O modelo de dados foi extraído diretamente do MySQL Workbench, já os diagramas UML foram feitos na ferramenta Astah Community (versão gratuita) e no site Draw.io.

3.2.1. Diagrama de Entidade-Relacionamento

Figura 17 - Diagrama de entidade-relacionamento.
Fonte: Própria.



- **ebk_usuario**

Armazena as informações de cadastro do usuário.

- **ebk_livro**

Armazena informações do livro, bem como o arquivo do livro digital.

- **ebk_categoria**

Armazena informações referentes às categorias criadas pelo usuário.

- **ebk_livro_categoria**

Armazena informações das relações entre livros e categorias.

3.2.2. Diagrama de Caso de Uso

3.2.2.1. Interface Web

O diagrama de caso de uso a seguir apresenta as ações que os usuários podem executar ao utilizarem a interface *web*, e são detalhados nos quadros 1 a 11.

Figura 18 - Diagrama de caso de uso da interface web.
Fonte: Própria.

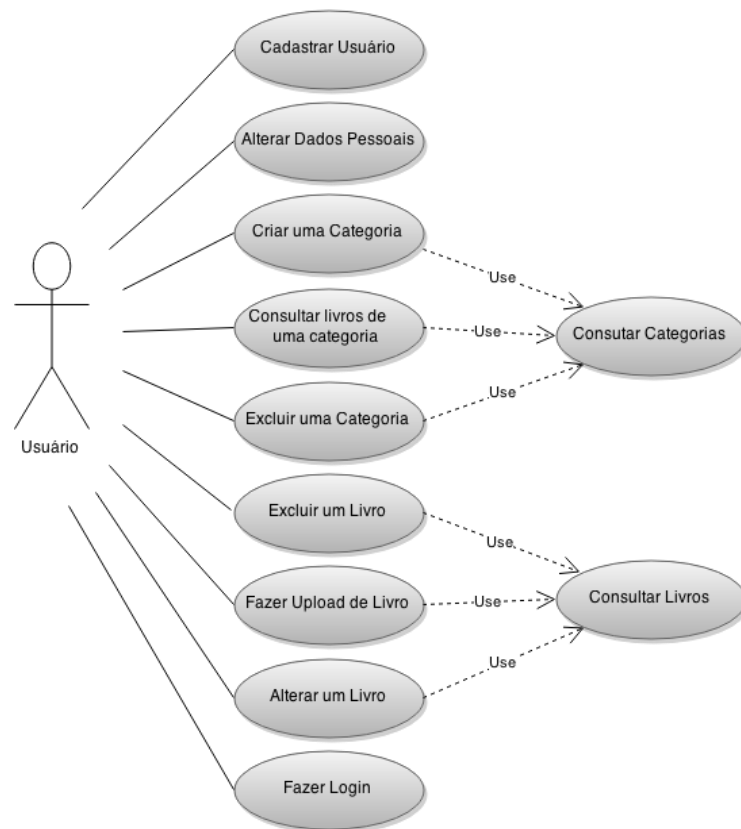


Tabela 1 - Caso de Uso Cadastrar Usuário.
Fonte: Própria.

| Identificação | |
|--|--|
| Sistema | NT - NuvemTeca - Módulo WEB |
| Código | NT_WEB – UC-01 |
| Nome do Caso de Uso | Cadastro de Usuário |
| Descrição do Caso de Uso | Permitir que um usuário realize seu cadastro no sistema. |
| Informações adicionais | |
| Atores | Usuários |
| Pré-Condições | Estar conectado à internet |
| Pós-Condições | Usuário cadastrado na aplicação. |
| Inclusões | |
| Extensões | |
| Fluxo Principal | |
| 1. O usuário acessa a página de login. | |
| 2. O usuário clica no botão “Cadastre-se”. | |
| 3. A aplicação irá mostrar uma tela contendo um formulário com os campos necessário para a realização do cadastro do usuário. | |
| 4. O usuário preencher o formulário de cadastro. | |
| 5. O usuário clica em “Cadastrar”. | |
| 6. A aplicação valida as informações inseridas pelo usuário | |
| 7. A aplicação cadastra o novo usuário. | |
| 8. A aplicação exibe a mensagem: “Usuário Cadastrado com sucesso”. | |
| Fluxo de Exceção | |
| FE01 – Nome de Usuário inválido Caso o nome de usuário informado já esteja sendo usado por outro usuário, a aplicação irá exibir a seguinte mensagem: “O nome de usuário informado já está sendo utilizado”. | |
| Restrições/Validações | |
| <ul style="list-style-type: none"> Os campos nome de usuário, nome, e-mail e senha devem ser preenchidos obrigatoriamente. O campo Nome de Usuário deve ser único | |

Tabela 2 - Caso de Uso Alterar Dados Pessoais.
Fonte: Própria.

| Identificação | |
|--|---|
| Sistema | NT - NuvemTeca - Módulo WEB |
| Código | NT_WEB – UC-02 |
| Nome do Caso de Uso | Alterar Dados Pessoais |
| Descrição do Caso de Uso | Alterar os dados do cadastro do usuário. |
| Informações adicionais | |
| Atores | Usuários |
| Pré-Condições | Estar autenticado no sistema. |
| Pós-Condições | Dados do usuário alterados no banco de dados. |
| Inclusões | |
| Extensões | |
| Fluxo Principal | |
| 1. O usuário acessa o menu Meu Perfil. | |
| 2. A aplicação irá mostrar uma tela contendo um formulário com os dados do usuário. | |
| 3. O usuário realiza as alterações desejadas e clicar no botão Salvar. | |
| 4. A aplicação salva as alterações feitas no cadastro do usuário. | |
| 5. A aplicação irá mostrar a mensagem “Alterações realizadas com sucesso”. | |
| Fluxo de Exceção | |
| | |
| Restrições/Validações | |
| 1. O campo “Nome de Usuário” não deve ser editável. 2. Para realizar a alteração da senha, o usuário deve preencher o campo “Senha Anterior” com a sua senha atual e o campo “Nova Senha” com a senha desejada. | |

Tabela 3 - Fazer Login.
Fonte: Própria.

| Identificação | |
|--|---------------------------------------|
| Sistema | NT - NuvemTeca - Módulo WEB |
| Código | NT_WEB – UC-09 |
| Nome do Caso de Uso | Fazer Login |
| Descrição do Caso de Uso | O usuário realiza o login no sistema. |
| Informações adicionais | |
| Atores | Usuários |
| Pós-Condições | Usuário autenticado no sistema. |
| Inclusões | |
| Extensões | |
| Fluxo Principal | |
| 1. O usuário acessa a página de login. | |
| 2. A aplicação irá mostrar uma tela com dados necessários para efetuar o login. | |
| 3. O usuário preenche os dados e clica no botão Login. | |
| 4. A aplicação valida os dados inseridos pelo usuário. | |
| 5. A aplicação redireciona o usuário para a tela inicial. | |
| Fluxo de Exceção | |
| <p>FE01 – Usuário Não Encontrado Caso o usuário não seja cadastrado no sistema, a aplicação informará a mensagem “O nome de usuário informado não foi cadastrado. Verifique se informou o nome de usuário correto ou clique no botão Cadastre-se para realizar o cadastro”.</p> <p>FE02– Dados inválidos Caso o usuário exista no banco de dados, mas a senha digitada seja diferente da senha armazenada nas informações do usuário, a aplicação web deve informar a mensagem “Senha inválida”.</p> | |
| Restrições/Validações | |
| 1. Os campos Nome de Usuário e Senha são obrigatórios para realizar o login. | |

Tabela 4 - Caso de Uso Consultar Livros.
Fonte: Própria.

| Identificação | |
|---|---|
| Sistema | NT - NuvemTeca - Módulo WEB |
| Código | NT_WEB – UC-04 |
| Nome do Caso de Uso | Consultar Livros |
| Descrição do Caso de Uso | Exibe os livros cadastrados pelo usuário. |
| Informações adicionais | |
| Atores | Usuários. |
| Pré-Condições | Estar autenticado no sistema. |
| Pós-Condições | |
| Inclusões | |
| Extensões | |
| Fluxo Principal | |
| 1. O usuário acessar o menu Meus Livros. | |
| 2. A aplicação mostra uma tela com a lista dos livros que foram adicionados pelo usuário, com opção de excluir e editar os livros listados. | |
| Fluxo de Exceção | |
| | |
| Restrições/Validações | |
| | |

Tabela 5 - Caso de Uso Excluir Livro.
Fonte: Própria.

| Identificação | |
|--|---|
| Sistema | NT - NuvemTeca - Módulo WEB |
| Código | NT_WEB – UC-05 |
| Nome do Caso de Uso | Excluir Livro |
| Descrição do Caso de Uso | Exclui um livro selecionado pelo usuário. |
| Informações adicionais | |
| Atores | Usuários |
| Pré-Condições | Estar autenticado no sistema. |
| Pós-Condições | Livro excluído do banco de dados. |
| Inclusões | Consultar Livros (UC-04) |
| Extensões | |
| Fluxo Principal | |
| 1. O usuário clica no botão de exclusão da tela de consulta de livros. | |
| 2. O aplicação exibe a mensagem “Confirma a exclusão do livro?” | |
| 3. O usuário clica na opção “Sim”. (FA01) | |
| 4. O aplicação exclui o livro . | |
| 5. O sistema exibe a mensagem “Livro excluído com sucesso”. | |
| 6. Volta para a etapa 2 do caso de uso Consultar Livros (UC-04). | |
| Fluxo Alternativo (FA01) | |
| 1. O usuário clica na opção “não”. | |
| 2. Volta para a etapa 2 do caso de uso Consultar Livros (UC-04). | |
| Restrições/Validações | |
| | |

Tabela 6 - Caso de Uso Upload de Livro.
Fonte: Própria.

| Identificação | |
|---|---|
| Sistema | NT - NuvemTeca - Módulo WEB |
| Código | NT_WEB – UC-06 |
| Nome do Caso de Uso | Upload de Livro |
| Descrição do Caso de Uso | Cadastrar um livro e realizar o upload do seu arquivo digital |
| Informações adicionais | |
| Atores | Usuários |
| Pré-Condições | Estar autenticado no sistema. |
| Pós-Condições | Livro cadastrado no sistema |
| Inclusões | |
| Extensões | |
| Fluxo Principal | |
| Ações do Ator | Ações do Sistema |
| 1. O usuário clica no botão Adicionar Livro na tela de consulta de livros. | |
| 2. A aplicação mostra uma tela com os as informações necessárias para o cadastro de um novo livro. | |
| 3. O usuário preenche as informações do livro. | |
| 4. O usuário clica no botão “Adicionar Arquivo”. | |
| 5. O usuário seleciona o arquivo do livro no formato epub e clica em Ok. | |
| 6. O usuário clica em “Salvar Livro” | |
| 7. A aplicação salva o livro. | |
| 8. A aplicação exibe a mensagem “Livro Cadastrado com Sucesso”. | |
| Fluxo de Exceção | |
| FE01 – Formato incorreto. Caso o usuário adicione o arquivo de um formato diferente de epub, a aplicação exibirá a mensagem: “Formato incorreto, selecione um arquivo no formato epub” | |
| FE02 – Campos Obrigatórios não preenchidos. Caso o usuário não preencha algum campo obrigatório, a aplicação exibirá a mensagem: “Um campo obrigatório não foi preenchido, verifique as informações inseridas e tente novamente”. | |
| Restrições/Validações | |
| 1. Os campos Título e Autor são obrigatórios. 2. O campo Arquivo é obrigatório. 3. O arquivo selecionado deve possuir da extensão .epub. | |

Tabela 7 - Caso de Uso Editar Livro.
Fonte: Própria.

| Identificação | |
|--|------------------------------------|
| Sistema | NT - NuvemTeca - Módulo WEB |
| Código | NT_WEB – UC-07 |
| Nome do Caso de Uso | Editar Livro |
| Descrição do Caso de Uso | Editar as informações de um livro. |
| Informações adicionais | |
| Atores | Usuários |
| Pré-Condições | Estar autenticado no sistema. |
| Pós-Condições | Informações do livro alteradas. |
| Inclusões | |
| Extensões | |
| Fluxo Principal | |
| Ações do Ator | Ações do Sistema |
| 1. O usuário clica no botão de edição de um livro na tela de consulta de livros. | |
| 2. A aplicação mostra uma tela com os as informações do livro selecionado. | |
| 3. O usuário realiza as alterações desejadas. | |
| 4. O usuário clica no botão Salvar | |
| 5. A aplicação salva as alterações. | |
| 6. A aplicação exibe a mensagem “Livro alterado com Sucesso”. | |
| Fluxo de Exceção | |
| <p>FE01 – Formato incorreto. Caso o usuário adicione o arquivo de um formato diferente de epub, a aplicação exibirá a mensagem: “Formato incorreto, selecione um arquivo no formato epub”</p> <p>FE02 – Campos Obrigatórios não preenchidos. Caso o usuário não preencha algum campo obrigatório, a aplicação exibirá a mensagem: “Um campo obrigatório não foi preenchido, verifique as informações inseridas e tente novamente”.</p> | |
| Restrições/Validações | |
| <ol style="list-style-type: none"> 1. Os campos Título e Autor são obrigatórios. 2. O campo Arquivo é obrigatório. 3. O arquivo selecionado deve possuir da extensão .epub. | |

Tabela 8 - Caso de Uso Consultar Categorias.
Fonte: Própria.

| Identificação | |
|---|--|
| Sistema | NT - NuvemTeca - Módulo WEB |
| Código | NT_WEB – UC-08 |
| Nome do Caso de Uso | Consultar Categorias |
| Descrição do Caso de Uso | Exibe as categorias criadas pelo usuário |
| Informações adicionais | |
| Atores | Usuários |
| Pré-Condições | Estar autenticado no sistema. |
| Pós-Condições | |
| Inclusões | |
| Extensões | |
| Fluxo Principal | |
| 1. O usuário acessa o menu Minhas Categorias | |
| 2. A aplicação exibe uma tela com a lista de categorias criadas pelo usuário, com opção de excluir e editar as categorias listadas. | |
| Fluxo Alternativo (FA01) | |
| | |
| Fluxo de Exceção | |
| | |
| Restrições/Validações | |
| | |

Tabela 9 - Caso de Uso Excluir Categoria.
Fonte: Própria.

| Identificação | |
|--|---|
| Sistema | NT - NuvemTeca - Módulo WEB |
| Código | NT_WEB – UC-09 |
| Nome do Caso de Uso | Excluir Categoria |
| Descrição do Caso de Uso | Exclui a categoria selecionada pelo usuário |
| Informações adicionais | |
| Atores | Usuários |
| Pré-Condições | Estar autenticado no sistema. |
| Pós-Condições | |
| Inclusões | |
| Extensões | |
| Fluxo Principal | |
| 1. O usuário clica no botão de excluir categoria na tela de listar categorias. | |
| 2. A aplicação mostrar a mensagem “Confirma a exclusão da categoria?” | |
| 3. O usuário clica na opção “Sim”. (FA01). | |
| 4. O sistema exclui a categoria. | |
| 5. A aplicação Mostra a mensagem “Categoria excluída com sucesso”. | |
| 6. A aplicação retorna para a tela de Consulta de Categorias (UC-08). | |
| Fluxo Alternativo (FA01) | |
| Ações do Ator | Ações do Sistema |
| 1. O usuário clica na opção “Não”. | |
| 2. A aplicação retorna para a tela de Consulta de Categorias (UC-08) | |
| Fluxo de Exceção | |
| | |
| Restrições/Validações | |
| | |

Tabela 10 - Caso de Uso Criar Categoria.
Fonte: Própria.

| Identificação | |
|---|--|
| Sistema | NT - NuvemTeca - Módulo WEB |
| Código | NT_WEB – UC-10 |
| Nome do Caso de Uso | Criar Categoria |
| Descrição do Caso de Uso | Cria uma categoria e adiciona livros a ela |
| Informações adicionais | |
| Atores | Usuários |
| Pré-Condições | Estar autenticado no sistema. |
| Pós-Condições | Categoria criada. |
| Inclusões | |
| Extensões | |
| Fluxo Principal | |
| 1. O usuário clica no botão Adicionar Categoria na tela de Consulta de Categorias. | |
| 2. A aplicação mostrar uma tela com as informações para o cadastro da categoria. | |
| 3. O usuário preenche o nome da categoria. | |
| 4. O usuário clica no botão de Adicionar Livros à Categoria. | |
| 5. O usuário seleciona os livros desejados, que foram cadastrados no sistema previamente. | |
| 6. O usuário clica em Salvar | |
| 7. A aplicação salva a categoria. | |
| 8. A aplicação exibe a mensagem “Categoria adicionada com sucesso”. | |
| Fluxo Alternativo (FA01) | |
| | |
| Fluxo de Exceção | |
| | |
| Restrições/Validações | |
| 1. O campo Nome da Categoria é obrigatório. | |
| 2. O campo Nome da Categoria deve ser único por usuário. | |

Tabela 11 - Caso de Uso Consultar Livros de uma Categoria.
Fonte: Própria.

| Identificação | |
|---|--|
| Sistema | NT - NuvemTeca - Módulo WEB |
| Código | NT_WEB – UC-11 |
| Nome do Caso de Uso | Consultar Livros de uma Categoria |
| Descrição do Caso de Uso | Cria uma categoria e adiciona livros a ela |
| Informações adicionais | |
| Atores | Usuários |
| Pré-Condições | Estar autenticado no sistema. |
| Pós-Condições | |
| Inclusões | |
| Extensões | |
| Fluxo Principal | |
| 1. O usuário clica no botão de visualizar informações da categoria na tela de Consulta de Categoria. | |
| 2. A aplicação mostrar uma tela com as informações da categoria e os livros que foram associados a ela. | |
| Fluxo Alternativo (FA01) | |
| | |
| Fluxo de Exceção | |
| | |
| Restrições/Validações | |
| | |

3.2.2.2. Aplicativo Mobile

O diagrama de caso de uso a seguir apresenta as ações que os usuários podem executar ao utilizarem o aplicativo móvel, e são detalhados nos quadros 12 a 18.

Figura 19 - Diagrama de caso de uso do aplicativo móvel.
Fonte: Própria

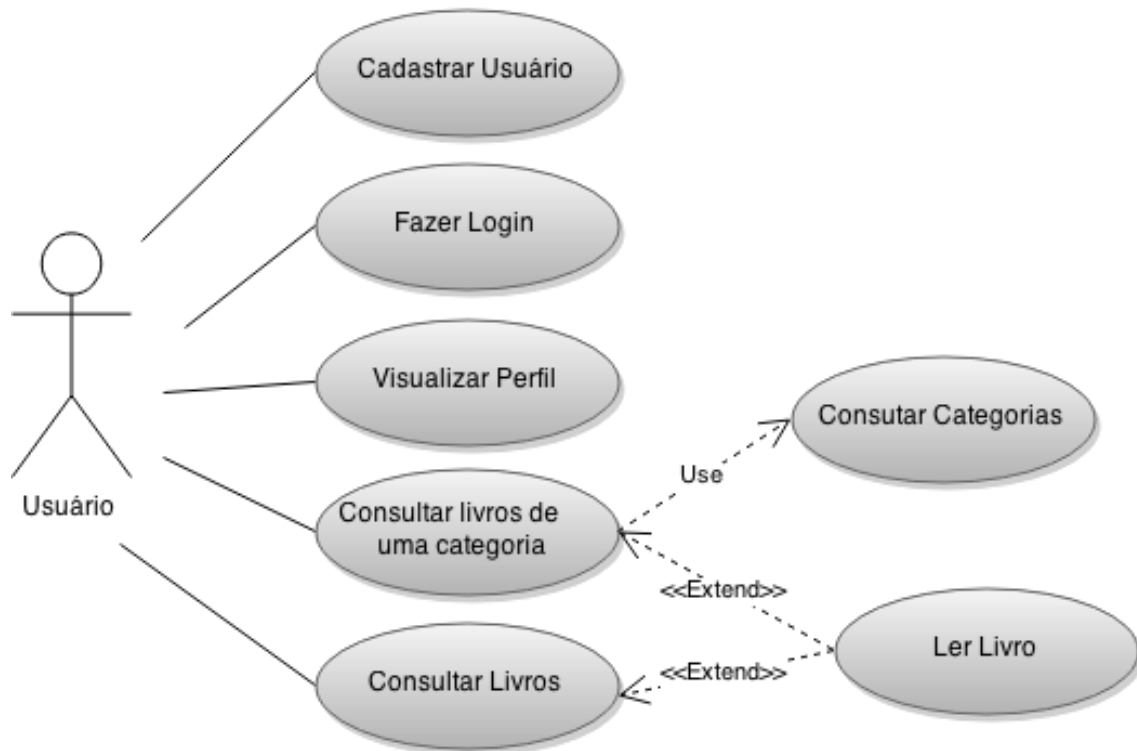


Tabela 12 - Caso de Uso Web Fazer Login.
Fonte: Própria.

| Identificação | |
|---|---------------------------------------|
| Sistema | NT - NuvemTeca – Aplicativo Mobile |
| Código | NT_MOBILE – UC-01 |
| Nome do Caso de Uso | Fazer Login |
| Descrição do Caso de Uso | O usuário realiza o login no sistema. |
| Informações adicionais | |
| Atores | Usuários |
| Pós-Condições | Usuário autenticado no sistema. |
| Inclusões | |
| Extensões | |
| Fluxo Principal | |
| 1. O usuário abre o aplicativo. | |
| 2. O aplicativo irá mostrar uma tela com dados necessários para efetuar o login. | |
| 3. O usuário preenche os dados e clica no botão Login. | |
| 4. O aplicativo valida os dados inseridos pelo usuário. | |
| 5. O aplicativo redireciona o usuário para a tela inicial. | |
| Fluxo de Exceção | |
| FE01 – Usuário Não Encontrado Caso o usuário não seja cadastrado no sistema, o aplicativo informará a mensagem “O nome de usuário informado não foi encontrado”. | |
| FE02– Dados inválidos Caso o usuário exista no banco de dados, mas a senha digitada seja diferente da senha armazenada nas informações do usuário, a aplicação web deve informar a mensagem “Senha inválida”. | |
| Restrições/Validações | |
| 2. Os campos Nome de Usuário e Senha são de preenchimento obrigatório. | |

Tabela 13 - Caso de Uso Web Cadastrar Usuário.
Fonte: Própria.

| Identificação | |
|--|---|
| Sistema | NT - NuvemTeca - Aplicativo Mobile |
| Código | NT_ MOBILE – UC-02 |
| Nome do Caso de Uso | Cadastro de Usuário |
| Descrição do Caso de Uso | Permitir que um usuário realize seu cadastro no sistema,. |
| Informações adicionais | |
| Atores | Usuários |
| Pré-Condições | Estar conectado à internet |
| Pós-Condições | Usuário cadastrado na aplicação. |
| Inclusões | |
| Extensões | |
| Fluxo Principal | |
| 1. O usuário abre o aplicativo. | |
| 2. O usuário clica no botão “Cadastre-se”. | |
| 3. O aplicativo irá mostrar uma tela contendo um formulário com os campos necessário para a realização do cadastro do usuário. | |
| 4. O usuário preencher o formulário de cadastro. | |
| 5. O usuário clica em “Cadastrar”. | |
| 6. O aplicativo valida as informações inseridas pelo usuário | |
| 7. O aplicativo cadastra o novo usuário. | |
| 8. O aplicativo exibe a mensagem: “Usuário Cadastrado com sucesso”. | |
| 9. O aplicativo redireciona o usuário para a tela de login. | |
| Fluxo de Exceção | |
| FE01 – Nome de Usuário inválido Caso o nome de usuário informado já esteja sendo usado por outro usuário, a aplicação irá exibir a seguinte mensagem: “O nome de usuário informado já está sendo utilizado”. | |
| Restrições/Validações | |
| <ul style="list-style-type: none"> Os campos nome de usuário, nome, e-mail e senha devem ser preenchidos obrigatoriamente. O campo Nome de Usuário deve ser único | |

Tabela 14 - Caso de Uso Web Visualizar Perfil.
Fonte: Própria.

| Identificação | |
|---|--|
| Sistema | NT - NuvemTeca - Aplicativo Mobile |
| Código | NT_ MOBILE – UC-03 |
| Nome do Caso de Uso | Visualizar Perfil |
| Descrição do Caso de Uso | Permitir que um usuário visualize as informações do seu perfil |
| Informações adicionais | |
| Atores | Usuários |
| Pré-Condições | Estar autenticado no sistema. |
| Pós-Condições | |
| Inclusões | |
| Extensões | |
| Fluxo Principal | |
| 1. O usuário acessa o menu Meu Perfil | |
| 2. O aplicativo exibe uma tela com as informações do cadastro do usuário. | |
| Fluxo de Exceção | |
| | |
| Restrições/Validações | |
| | |

Tabela 15 - Caso de Uso Web Consultar Livros.
Fonte: Própria.

| Identificação | |
|---|--|
| Sistema | NT - NuvemTeca - Aplicativo Mobile |
| Código | NT_ MOBILE – UC-04 |
| Nome do Caso de Uso | Consultar Livros |
| Descrição do Caso de Uso | Permitir que um usuário visualize seus livros. |
| Informações adicionais | |
| Atores | Usuários |
| Pré-Condições | Estar autenticado no sistema. |
| Pós-Condições | |
| Inclusões | |
| Extensões | Ler Livro (UC-07) |
| Fluxo Principal | |
| 1. O usuário acessa o menu Meus Livros | |
| 2. O aplicativo exibe uma tela com a lista dos livros do usuário. | |
| Fluxo Alternativo | |
| 1. O usuário seleciona um dos livros listados. | |
| 2. Executa o caso de uso Ler Livro (UC-07) | |
| Restrições/Validações | |
| | |

Tabela 16 - Caso de Uso Web Consultar Categorias.
Fonte: Própria.

| Identificação | |
|---|---|
| Sistema | NT - NuvemTeca - Aplicativo Mobile |
| Código | NT_ MOBILE – UC-05 |
| Nome do Caso de Uso | Consultar Categorias |
| Descrição do Caso de Uso | Permitir que um usuário visualize suas categorias |
| Informações adicionais | |
| Atores | Usuários |
| Pré-Condições | Estar autenticado no sistema. |
| Pós-Condições | |
| Inclusões | |
| Extensões | |
| Fluxo Principal | |
| 1. O usuário acessa o menu Minhas Categorias | |
| 2. O aplicativo exibe uma tela com a lista das categorias do usuário. | |
| Fluxo de Exceção | |
| | |
| Restrições/Validações | |
| | |

Tabela 17 - Caso de Uso Web Consultar Livros de uma Categorias.
Fonte: Própria.

| Identificação | |
|---|---|
| Sistema | NT - NuvemTeca - Aplicativo Mobile |
| Código | NT_ MOBILE – UC-06 |
| Nome do Caso de Uso | Consultar Livros de uma Categorias |
| Descrição do Caso de Uso | Permitir que um usuário visualize os livros de uma categorias |
| Informações adicionais | |
| Atores | Usuários |
| Pré-Condições | Estar autenticado no sistema. |
| Pós-Condições | |
| Inclusões | Consultar Categorias (UC-05) |
| Extensões | Ler Livro (UC-07) |
| Fluxo Principal | |
| 1. O usuário seleciona uma categoria na tela de Consulta de Categorias. | |
| 2. O aplicativo exibe uma tela com a lista dos livros que pertencem a essa categoria. | |
| Fluxo Alternativo | |
| 3. O usuário seleciona um dos livros listados. | |
| 4. Executa o caso de uso Ler Livro (UC-07) | |
| Restrições/Validações | |
| | |

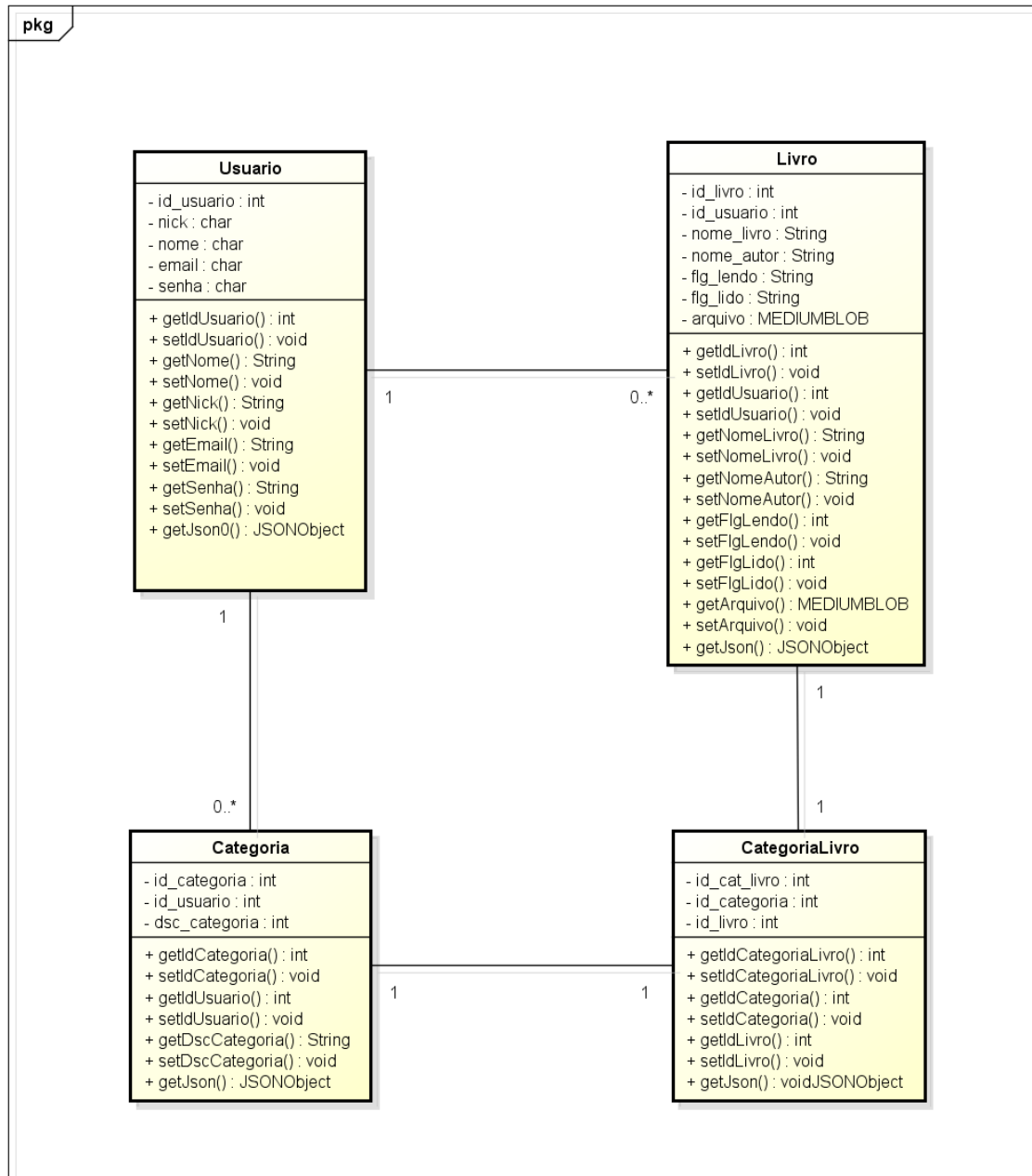
Tabela 18 - Caso de Uso Web Ler Livro.
Fonte: Própria.

| Identificação | |
|--|--------------------------------------|
| Sistema | NT - NuvemTeca - Aplicativo Mobile |
| Código | NT_ MOBILE – UC-07 |
| Nome do Caso de Uso | Ler Livro |
| Descrição do Caso de Uso | Permite que o usuário leia um livro. |
| Informações adicionais | |
| Atores | Usuários |
| Pré-Condições | Estar autenticado no sistema. |
| Pós-Condições | |
| Inclusões | |
| Extensões | |
| Fluxo Principal | |
| 1. O aplicativo exibe o livro digital para que o usuário inicie a leitura. | |
| Fluxo Alternativo | |
| | |
| Restrições/Validações | |
| | |

3.2.3. Diagrama de Classes

A imagem abaixo representa o diagrama de classes da aplicação:

Figura 20 - Diagrama de classes.
Fonte: Própria



DESENVOLVIMENTO

Com base no modelo de projeto do capítulo anterior, foi desenvolvida uma aplicação de biblioteca virtual que faz uso dos princípios da computação em nuvem que foram apresentados no início deste trabalho.

Este capítulo irá abordar as premissas do projeto, as principais técnicas e ferramentas utilizadas, o desenvolvimento da aplicação e os resultados obtidos.

3.3. PREMISSAS

Para a concepção da aplicação, seguiu-se uma série de premissas pré-definidas, tais como:

- **Tipo de licença**

Definiu-se que todas as ferramentas utilizadas durante o desenvolvimento deveriam oferecer licenças gratuitas.

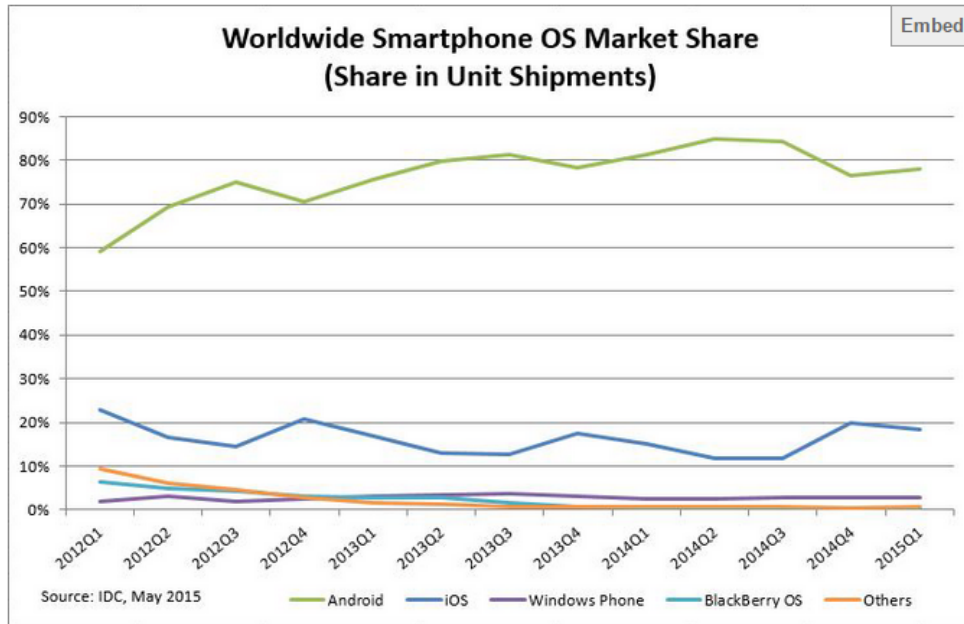
- **Portabilidade**

Optou-se pelo desenvolvimento de uma aplicação *mobile* pela portabilidade que os *smartphones* oferecem aos seus usuários. Restringir a utilização de uma aplicação de biblioteca virtual a um computador não fazia muito sentido nos dias atuais, uma vez que os dispositivos móveis estão ganhando cada vez mais espaço. Além disso, as pessoas estão cada vez com menos tempo disponível para se dedicarem à leitura devido ao ritmo de vida acelerado das grandes cidades, portanto é importante ter acesso ao aplicativo e todo o seu conteúdo em qualquer lugar, a qualquer momento.

- **Sistema operacional Android**

A possibilidade e facilidade de instalação de aplicativos de terceiros em *smartphones* com o sistema operacional Android diretamente, sem a necessidade de publicação do mesmo em uma loja virtual (como é o caso do iOS), ou a criação de um perfil de desenvolvedor oficial. Além disso, o Android é o sistema operacional de maior abrangência da atualidade (conforme podemos ver na imagem abaixo), permitindo que um maior número de pessoas pudesse ter acesso ao aplicativo.

Figura 21 - Gráfico de participação no mercado de sistemas operacionais mobile.
Fonte: IDC. <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>, 2015



| Period | Android | iOS | Windows Phone | BlackBerry OS | Others |
|---------|---------|-------|---------------|---------------|--------|
| Q1 2015 | 78.0% | 18.3% | 2.7% | 0.3% | 0.7% |
| Q1 2014 | 81.2% | 15.2% | 2.5% | 0.5% | 0.7% |
| Q1 2013 | 75.5% | 16.9% | 3.2% | 2.9% | 1.5% |
| Q1 2012 | 59.2% | 22.9% | 2.0% | 6.3% | 9.5% |

3.4. TÉCNICAS E FERRAMENTAS

O desenvolvimento da aplicação empregou técnicas e paradigmas da orientação a objeto, utilizando a linguagem de programação Java e as IDE's Netbeans 8.0.2, para o desenvolvimento do *web service* e a aplicação *web*, e o Android Studio 1.1, para o desenvolvimento do aplicativo *mobile*. Além disso, foi utilizado o banco de dados MySQL 5.6 e o servidor Glassfish 4.

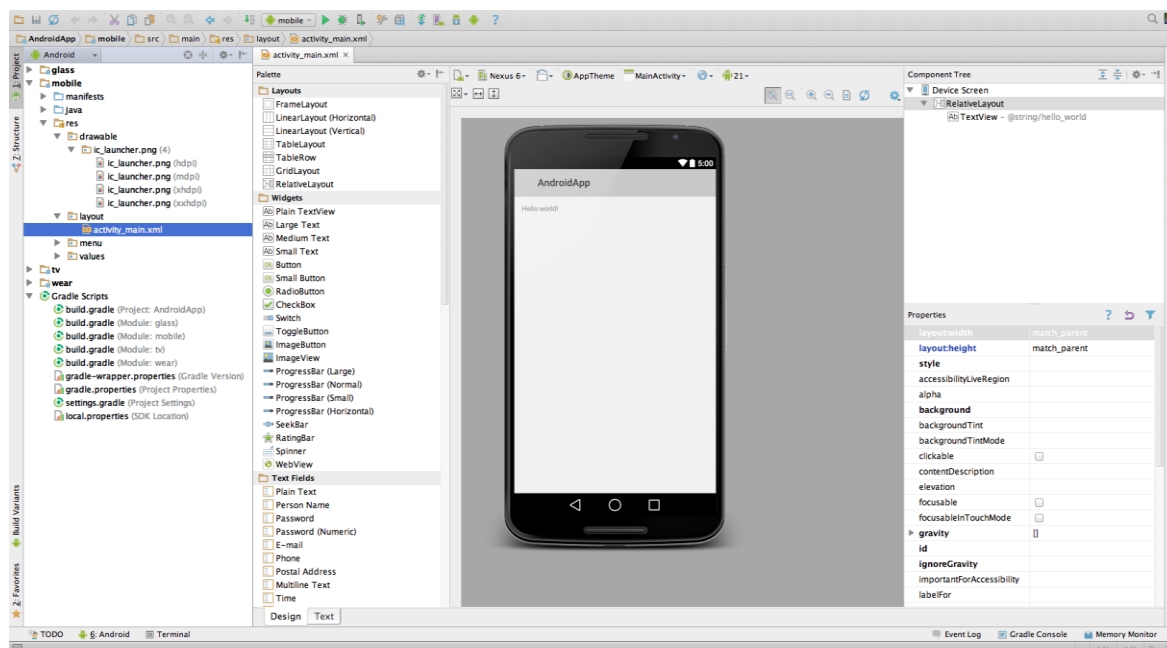
3.4.1. Android Studio

Anunciado durante a Google I/O em 2013, o Android Studio é o ambiente de desenvolvimento integrado (IDE, do inglês *Integrated Development Environment*) oficial do Google para a plataforma Android. As principais funções deste *software* incluem a edição inteligente de códigos, recursos para design de interface de usuário e análise de performance, ambiente de depuração de erros, entre outros.

Na imagem abaixo, podemos ver a interface básica do Android Studio.

Figura 22 - Interface do Android Studio.

Fonte: Android Developer. <http://developer.android.com/tools/studio/index.html>, 2015



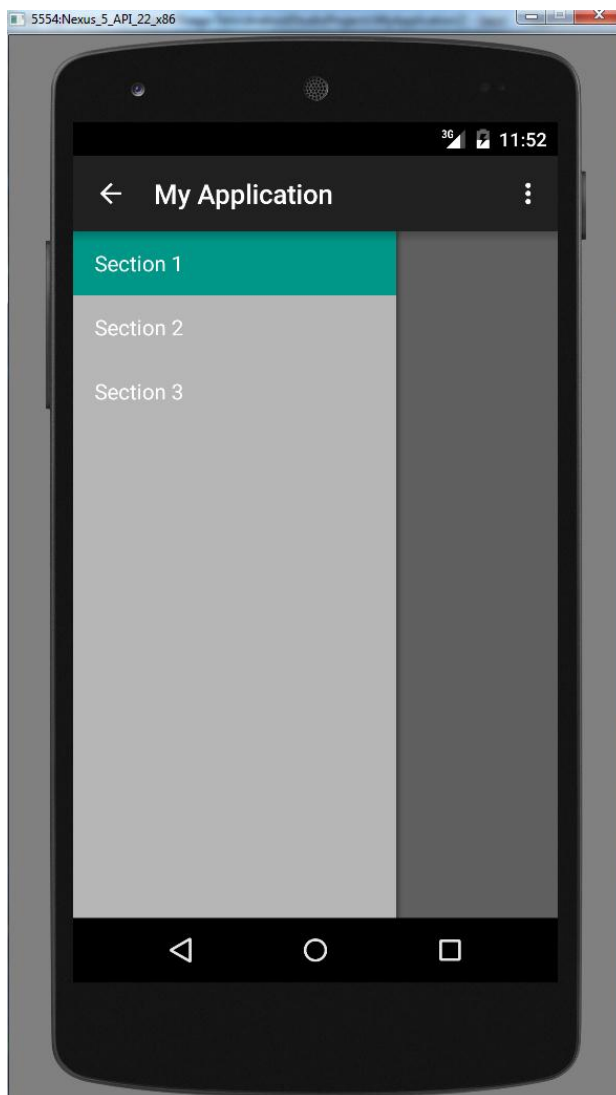
A versão utilizada neste trabalho é a 1.1 que já oferece suporte nativo para o Android 5.0 (Lollipop). Ela é disponibilizada de forma gratuita no site oficial para as plataformas Windows, Mac OS X e Linux.

Também é possível obter apenas o pacote com o SDK do Android para ser utilizado em outras IDE's. Este pacote também está disponível para Windows, Mac OS X e Linux.

Dentre as muitas ferramentas do Android Studio, vale destacar o emulador virtual de dispositivos móveis. Trata-se de uma ferramenta que roda o sistema operacional Android (é possível escolher qual a versão desejada) simulando um dispositivo real. Nela, é possível executar as aplicações em desenvolvimento em um ambiente virtual. Também é possível personalizar o hardware do dispositivo que se deseja simular, de modo que é possível testar a performance da aplicação em um hardware específico.

Na imagem abaixo, podemos ver a utilização desse emulador rodando o projeto desenvolvido neste trabalho dentro do Android Studio.

Figura 23 - Emulador do Android Studio.
Fonte: Própria.



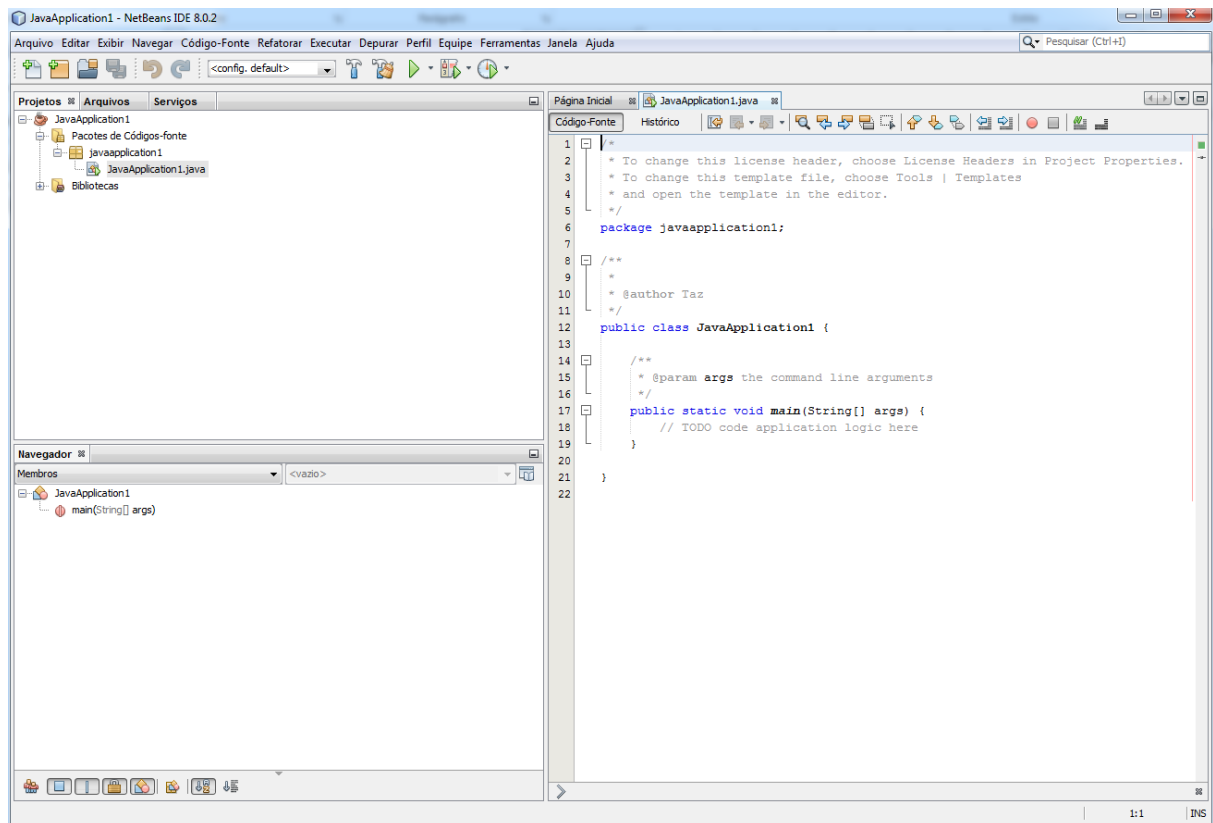
3.4.2. Netbeans

O NetBeans é a suíte de desenvolvimento oficial para o Java 8. No entanto, nele, é também é possível desenvolver em muitas outras linguagens. Ele também da suporta o desenvolvimento de aplicações mobile, inclusive para Android. Porém, optou-se pela escolha do Netbeans para o desenvolvimento dos *servlets* utilizados na comunicação entre a aplicação cliente e o servidor, e para o desenvolvimento da aplicação web.

A versão 8.0.2, utilizada neste trabalho, foi lançada em novembro de 2014 e está disponível gratuitamente em seu site oficial. Optou-se pela utilização dessa ferramenta por se tratar de um ambiente de desenvolvimento bastante robusto e conceituado no mercado, além da facilidade operacional e vasto material de suporte (inclusive em português).

Na imagem abaixo, podemos ver a interface básica do NetBeans.

Figura 24 - Interface do NetBeans 8.0.2.
Fonte: Própria



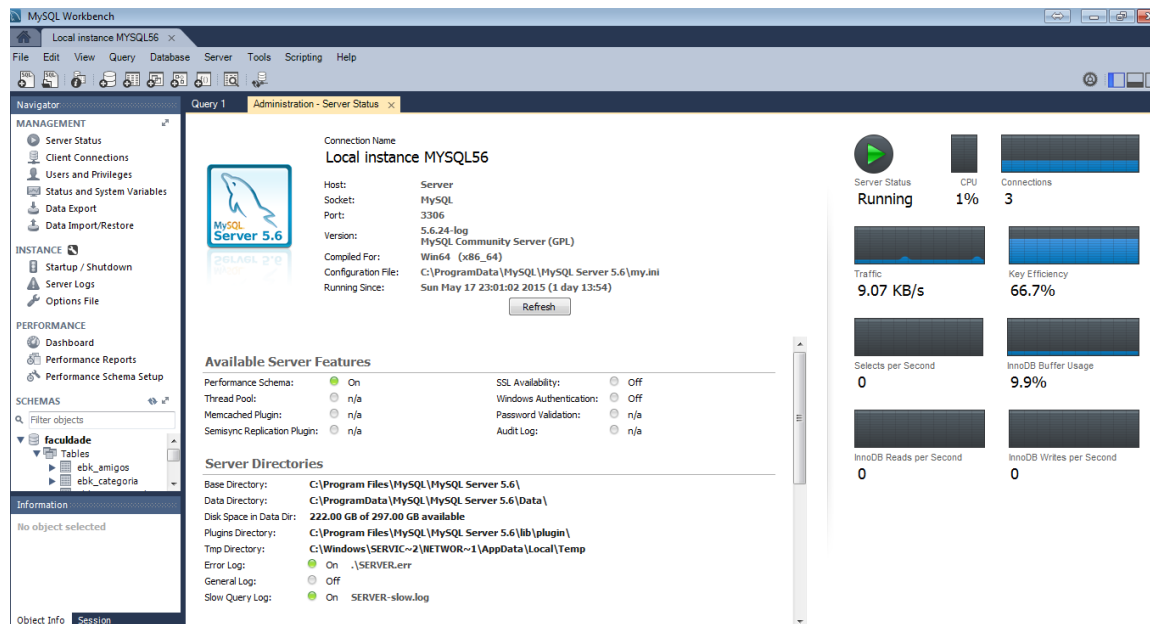
A suíte NetBeans está disponível para Windows, Mac OS X e Linux.

3.4.3. MySQL

O MySQL foi o banco de dados escolhido neste trabalho por ser o sistema de código aberto mais utilizado no mundo. A versão adotada foi a 5.6 para Windows que está disponível gratuitamente no site oficial da Oracle (desenvolvedora do sistema). Além desta, existem muitas outras versões que atendem a todo tipo de necessidade, desde um usuário doméstico até grandes empresas (Enterprise Edition).

A imagem abaixo ilustra a central de controle do banco de dados MySQL.

Figura 25 - Interface do MySQL Workbench.
Fonte: Própria

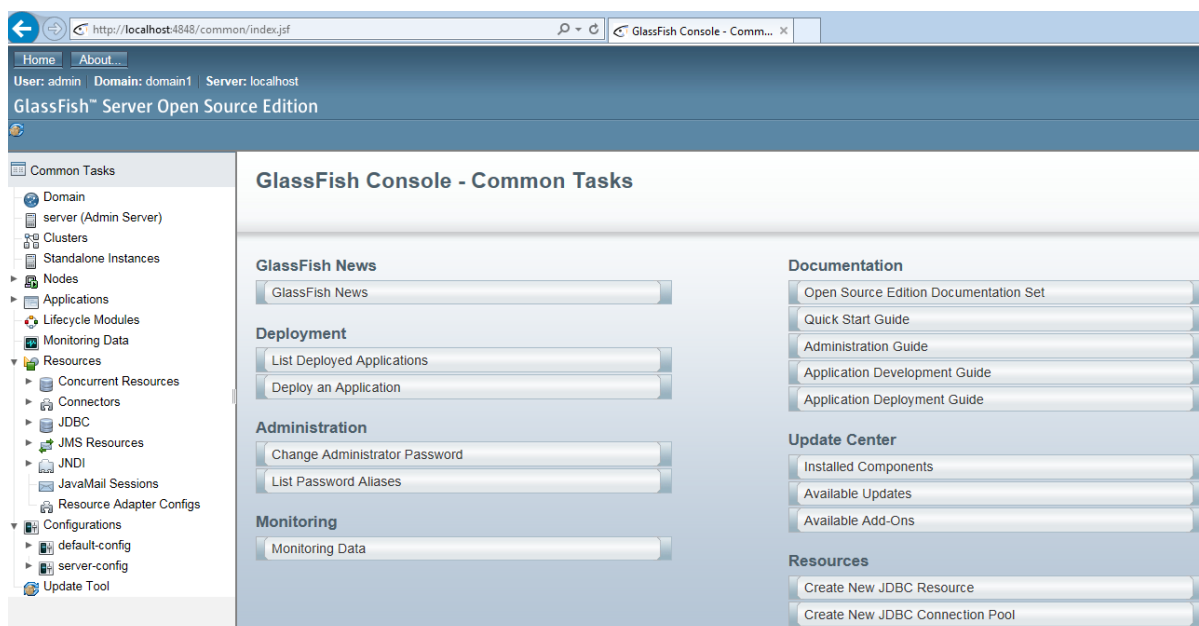


3.4.4. Glassfish

O Glassfish é um servidor de aplicação de código aberto. Por padrão, ele é instalado em conjunto com o NetBeans para rodar aplicações Java. Optou-se pela utilização dele por se tratar de um servidor relativamente leve e com vasto material de suporte. O mesmo está disponível gratuitamente no site oficial da Oracle para Windows e Linux.

Na imagem abaixo, podemos ver o servidor Glassfish em execução.

Figura 26 - Interface do MySQL.
Fonte: Própria



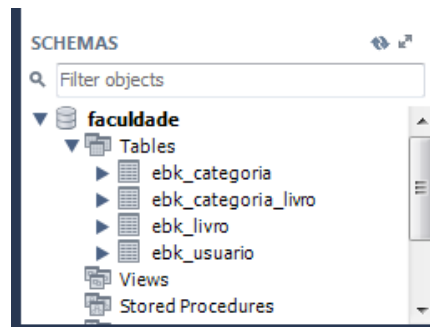
3.5. DESENVOLVIMENTO DA APLICAÇÃO

O desenvolvimento da aplicação foi dividido em quatro fases principais: banco de dados, servlet, interface web e aplicativo mobile.

3.5.1. Banco de Dados

A criação da estrutura do banco de dados foi feita com o auxílio da ferramenta MySQL Workbench. Nela, foi definido um novo esquema chamado “faculdade”, onde foram criadas todas as tabelas definidas no capítulo 5.

Figura 27 - Esquema do banco de dados.
Fonte: Própria



Uma vez criado o esquema, o banco de dados já está funcionando. No entanto, como a aplicação irá se conectar através da internet, foi necessário configurar as permissões de acesso que, por padrão, não permitem acesso de IP's não cadastrados.

No menu “Users and privileges”, foi definida a tag “%” para o usuário root no campo onde se restringe o IP de origem para o acesso ao banco. Dessa forma é possível se conectar como root de qualquer lugar.

Figura 28 - Permissões de usuário do MySQL.
Fonte: Própria

Local instance MySQL56
Users and Privileges

User Accounts

| User | From Host |
|-----------------|-----------|
| (!) <anonymous> | % |
| admin | % |
| root | localhost |
| root | 127.0.0.1 |
| root | ::1 |
| root | % |

Details for account root@%

Login Account Limits Administrative Roles Schema Privileges

Login Name: You may create multiple accounts with the same name to connect from different hosts.

Authentication Type: For the standard password and/or host based authentication, select 'Standard'.

Limit to Hosts Matching: % and _ wildcards may be used

Password: Type a password to reset it.

Consider using a password with 8 or more characters with mixed case letters, numbers and punctuation marks.

Confirm Password: Enter password again to confirm.

Mesmo configurando as permissões de usuário, ainda não era possível acessar o banco de dados pela internet. Pelo fato do servidor estar hospedado em uma rede residencial, foi necessário configurar o roteador para fazer o redirecionamento das portas necessárias para o IP local do servidor. Nesse caso, as portas 3306 TCP e 3306 UDP foram redirecionadas para o IP 192.168.0.200 através do menu “Virtual Servers”.

Figura 29 - Redirecionamento de portas TCP/UDP.
Fonte: Própria

LIVE™
Viva essa experiência.

DSL ☒ Downstream 57481 kbps
Upstream 34495 kbps refresh
Internet ☒ Connected. reboot

Device Info
Advanced Setup
3G Config
LAN
NAT
Virtual Servers
Port Triggering
DMZ Host
Security
Parental Control
DNS
UPnP
Power Management
Wireless
Multimedia
Diagnostics
Management
Logout

NAT -- Virtual Servers Setup

Virtual Server allows you to direct incoming traffic from WAN side (identified by Protocol and External port) to the Internal server with private IP address on the LAN side. The Internal port is required only if the external port needs to be converted to a different port number used by the server on the LAN side. A maximum 32 entries can be configured.

| Server Name | External Port Start | External Port End | Protocol | Internal Port Start | Internal Port End | Server IP Address | WAN Interface | Remove |
|-------------------------|---------------------|-------------------|----------|---------------------|-------------------|-------------------|---------------|--------------------------|
| IIS | 1234 | 1234 | TCP/UDP | 1234 | 1234 | 192.168.0.200 | ppp1.1 | <input type="checkbox"/> |
| Minecraft | 25565 | 25565 | TCP/UDP | 25565 | 25565 | 192.168.0.200 | ppp1.1 | <input type="checkbox"/> |
| Apache | 90 | 90 | TCP/UDP | 90 | 90 | 192.168.0.200 | ppp1.1 | <input type="checkbox"/> |
| OpenTTD | 3979 | 3979 | TCP/UDP | 3979 | 3979 | 192.168.0.200 | ppp1.1 | <input type="checkbox"/> |
| Teamspeak | 9987 | 9987 | TCP/UDP | 9987 | 9987 | 192.168.0.200 | ppp1.1 | <input type="checkbox"/> |
| Teamspeak File Transfer | 30033 | 30033 | TCP/UDP | 30033 | 30033 | 192.168.0.200 | ppp1.1 | <input type="checkbox"/> |
| Filezilla | 2050 | 2050 | TCP/UDP | 2050 | 2050 | 192.168.0.200 | ppp1.1 | <input type="checkbox"/> |
| Filezilla | 50000 | 51000 | TCP/UDP | 50000 | 51000 | 192.168.0.200 | ppp1.1 | <input type="checkbox"/> |
| MySQL | 3306 | 3306 | TCP/UDP | 3306 | 3306 | 192.168.0.200 | ppp1.1 | <input type="checkbox"/> |

3.5.2. Servlet

Dentre os *servlets* criados para a aplicação, podemos destacar os seguintes:

- **Cadastro de usuário (ServCadUsuario)**

Recebe, através de parâmetros, os dados de usuário do aplicativo móvel ou da interface *web*. Em seguida, verifica se o nome de usuário já existe no banco de dados. Caso ele não exista, o *servlet* chama a classe de manipulação de dados de usuário (“DaoUsuario”) passando os dados do usuário para serem inseridos no banco.

Na imagem a seguir, podemos ver o trecho de código que realiza uma consulta ao banco de dados para saber se o usuário já existe.

Figura 30 - Trecho do código do ServCadUsuario.

Fonte: Própria

```
public boolean verificaUsuarioDuplicado ( String p_nick ) throws SQLException
{
    conectar();

    String sql = "select 1 from ebk_usuario where nick = '@nick'";
    sql = sql.replace("@nick", p_nick);

    try
    {
        //Executa a query
        ResultSet rs = comando.executeQuery(sql);

        // Verifica se ele encontrou o nick, se encontrar ele retorna true
        return rs.next();
    }
    finally
    {
        fechar();
    }
}
```

- **Login (ServLogin e ServLoginWeb)**

Foram criados dois *servlets* de *login*, um para o aplicativo móvel e outro para a interface *web*. Em ambos os casos, é verificado se o nome de usuário existe no banco de dados e se a senha corresponde à cadastrada. Caso o *login* seja bem sucedido, no caso do aplicativo móvel, todas as informações do usuário são recuperadas (dados do usuário, livros e categorias), convertidas em formato JSON e enviadas para o dispositivo, para que não seja necessário que a conexão com a internet seja mantida durante toda a utilização. Já na interface *web*, são recuperados os dados conforme as solicitações do usuário.

Na imagem a seguir, podemos ver o trecho de código do ServLogin (versão para aplicativo móvel) e do ServLoginWeb (versão para interface *web*).

Figura 31 - Trecho do código do ServLogin.
Fonte: Própria.

```
// Recupera os parâmetros
v_nick = request.getParameter("u");
v_senha = request.getParameter("p");

try
{
    // 1 - Verifica se o login está correto
    login = daoUsuario.validaLoginUsuario(v_nick, v_senha);

    if (login)
    {
        json.put("info", "success");

        // 2 - Inclui no JSON de resposta as informações do usuário
        recuperaInformacoesUsuario();

        // 3 - Inclui no JSON de resposta as informações das categorias do usuário
        recuperarCategoriasUsuario();

        // 4 - Inclui no JSON de resposta as informações dos livros do usuário
        recuperarLivrosUsuario();

        // 5 - Inclui no JSON de resposta as informações da relação livro x categoria
        recuperarRelacaoLivroCat();

    }
    else
    {
        json.put("info", "fail");
    }
}
```

Figura 32 - Trecho do código do ServLoginWeb.
Fonte: Própria

```
// Etapa 01 : Recupera os parâmetros
String v_nick = request.getParameter("p_nome_usuario");
String v_senha = request.getParameter("p_senha");

// Etapa 02 : Verifica se o login está correto
boolean login = daoUsuario.validaLoginUsuario(v_nick, v_senha);

// Etapa 03: Se o login estiver correto, abre a tela inicial
if (login)
{
    try
    {
        // Recupera os dados do usuário
        DadosUsuario.usuario = daoUsuario.recuperaUsuario(v_nick);

        // Salva o id do usuário na sessão
        HttpSession session = request.getSession(true);
        session.setAttribute("id_usuario", DadosUsuario.usuario.getIdUsuario());

        //Redireciona o usuário para a tela principal
        response.sendRedirect("home.jsp");
    }
    catch(SQLException se)
    {
    }
}
```

- **Cadastro de livros (ServCadLivro)**

Recebe os dados do livro: nome do livro, nome do autor e arquivo no formato epub. Este último é enviado através de um método `InputStream` da classe Java e armazenado no banco de

dados em formato mediublob. Caso a inserção seja bem sucedida, é retornada uma mensagem “Livro cadastrado com sucesso”.

Na imagem abaixo, podemos ver um trecho de código referente à inclusão de um livro no banco de dados.

Figura 33 - Trecho do código do ServCadLivro.
Fonte: Própria

```
public boolean insertLivroComArquivo(Livro livro) throws SQLException
{
    conectar();

    try
    {
        String sql = "insert into ebk_livro (id_livro, id_usuario, nome_livro, nome_autor, arquivo) "
            + " values (?, ?, ?, ?, ?)";

        PreparedStatement statement;
        statement = con.prepareStatement(sql);

        statement.setString(1, livro.getIdLivroString());
        statement.setString(2, livro.getIdUsuarioString());
        statement.setString(3, livro.getNomeLivro());
        statement.setString(4, livro.getNomeAutor());
        statement.setBlob(5, livro.getArquivo());

        int row = statement.executeUpdate();

        // Retorna true se a operação for bem sucedida
        return (row > 0);
    }
    finally
    {
        fechar();
    }
}
```

3.5.3. Interface Web

A interface web é o canal de acesso do usuário para o envio dos arquivos no formato epub, além de poder consultar e editar diversas informações do perfil, dos livros e das categorias.

- **Meu Perfil**

Nesta tela, o usuário pode visualizar e editar todos os seus dados de cadastro.

Figura 34 - Rascunho da tela “Meu Perfil”.
Fonte: Própria

Logo

Meu Perfil

Meus Livros

Minhas Categorias

Sair

Meu Perfil

Nome de Usuário
user

Nome
Usuario

Email
usuario@email.com

Senha Atual

Nova Senha

Salvar

- **Meus Livros**

Nesta tela, o usuário pode visualizar todos os seus livros, bem como editá-los e excluí-los também.

Figura 35 - Rascunho da tela “Meus Livros”.
Fonte: Própria

Logo

Meu Perfil

Meus Livros

Minhas Categorias

Sair

Meus Livros

Título Livro 00 ✎ ✕

Título Livro 01 ✎ ✕

Título Livro 02 ✎ ✕

Título Livro 03 ✎ ✕

Título Livro 04 ✎ ✕

Adicionar Livro

- **Minhas Categorias**

Nesta tela, o usuário pode visualizar, adicionar, editar e excluir todas as suas categorias.

Figura 36 - Rascunho da tela “Minhas Categorias”.
Fonte: Própria



- **Novo Livro**

Nesta tela, o usuário adiciona um novo livro, entrando com as informações do mesmo e selecionando o arquivo no formato epub.

Figura 37 - Rascunho da tela “Novo Livro”.
Fonte: Própria



- **Nova Categoria**

Nesta tela, o usuário adiciona um novo livro, entrando com as informações do mesmo e selecionando o arquivo no formato epub.

Figura 38 - Rascunho da tela “Nova Categoria”.
Fonte: Própria

The wireframe shows a web interface for creating a new category. On the left is a sidebar with a logo and navigation links: 'Meu Perfil', 'Meus Livros', 'Minhas Categorias', and 'Sair'. The main content area is titled 'Nova Categoria'. It includes a text input for 'Nome da Categoria'. Below it is a section 'Livros da Categoria' with a plus icon and a list of three existing books, each with a delete icon. To the right is a 'Seleção de Livro' panel containing a list of four books with checkboxes. The first three are checked, and the fourth is not. A 'Salvar' button is located at the bottom of the main area.

3.5.4. **Aplicativo Mobile**

O aplicativo móvel é onde o usuário pode acessar as informações que foram inseridas na interface *web* e ler os livros no formato epub.

A imagem abaixo mostra um trecho de código referente à função de *login*. Nele, podemos ver como é feita a inclusão dos parâmetros, a chamada do servidor e a recuperação dos dados. Outro detalhe importante é a necessidade de executar essa função de forma assíncrona, ou seja, em paralelo à execução da atividade principal. Isso se deve ao fato que esta função transmite uma grande quantidade de dados e por isso seu tempo de execução pode ser longo. Dessa forma, caso a execução fosse linear, a aplicação iria travar enquanto a função estivesse em andamento.

Figura 39 - Trecho de código da tela de login do aplicativo móvel.
Fonte: Própria

```
private class Login extends AsyncTask<String, String, String>
{
    @Override
    protected String doInBackground(String... args)
    {
        // Recupera o nome de usuário e senha da tela de login
        String username = uname.getText().toString();
        String pass = password.getText().toString();

        // Monta os parâmetros
        List<NameValuePair> params = new ArrayList<>();
        params.add(new BasicNameValuePair("u",username));
        params.add(new BasicNameValuePair("p",pass));

        // Faz o request para o servlet
        json = jParser.makeHttpRequest(url_login, "GET", params);

        // Verifica o retorno
        if ( json.length() != 0 )
        {
            try {
                String retorno = json.getString("info");

                Log.d("Msg", json.getString("info"));

                if (retorno.equals("success"))
                {
                    // Recupera os dados do usuário
                    DadosUsuario.recuperaDadosDoJason(json);

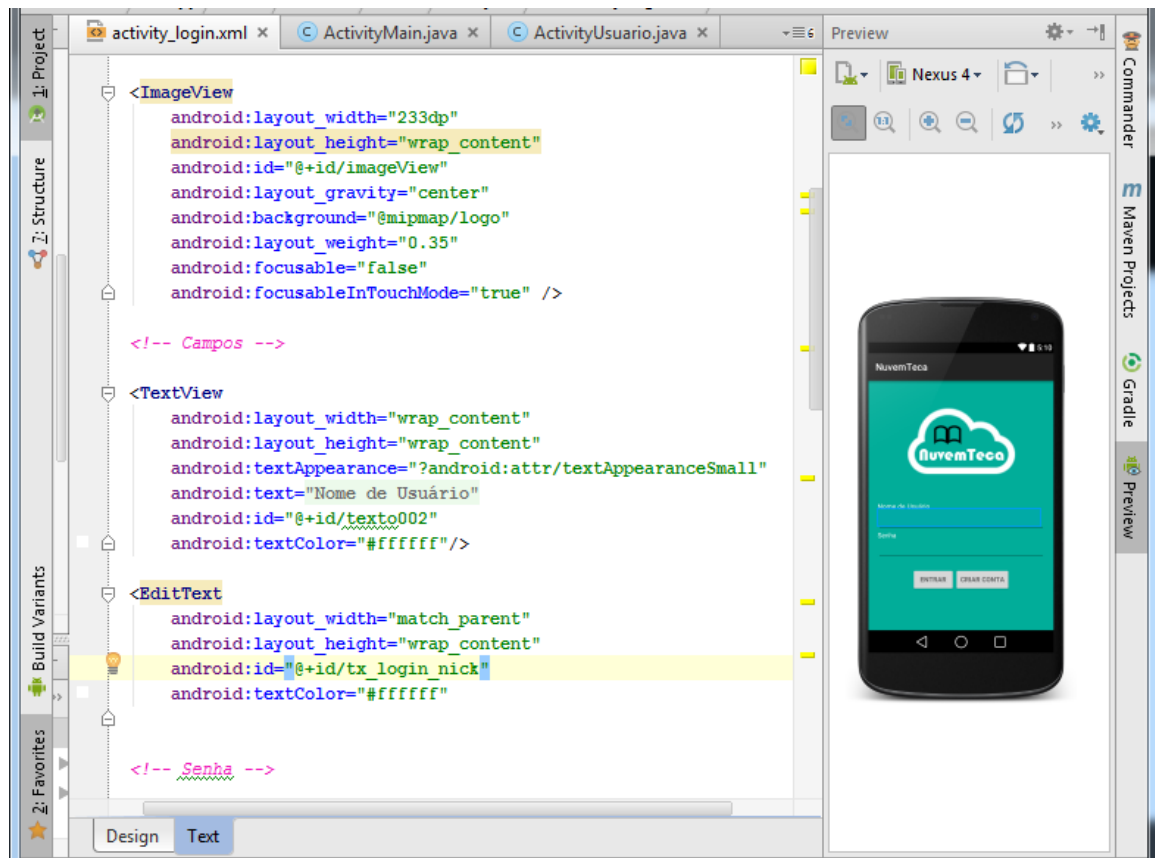
                    // Redireciona o usuário para a tela inicial
                    Intent login = new Intent(getApplicationContext(), ActivityMain.class);
                    login.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                    startActivity(login);
                    finish();
                }
            }
            else

```

No Android Studio, todas as telas de interface são escritas em linguagem XML. Associado ao XML existe uma classe de controle dessa tela, que define o seu tipo (fragmento, atividade, lista, entre outros).

Na imagem abaixo, podemos ver o *layout* da tela de *login*, com o seu respectivo código XML.

Figura 40 - Layout da tela de login, dentro do Android Studio.
Fonte: Própria



A transmissão de dados entre fragmentos é feita através da classe *Bundle*, que é um conjunto de objetos que podem ser definidos e adicionados ao fragmento utilizando a função *setArgument()* (função nativa da classe *Fragment*).

Na imagem a seguir, podemos ver o método *onCreateView* da tela de categorias. Essa tela mostra a lista de categorias do usuário e, quando o usuário clica em uma das categorias, ele é redirecionado para uma tela com a lista de livros pertencentes à essa categorias.

No método *onCreateView*, podemos ver como é feita a recuperação do identificador da categoria selecionada e como o mesmo é transmitido para a tela de livros, para ser usado como filtro de exibição dos livros.

**Figura 41 - Método onCreateView da tela de categorias.
Fonte: Própria**

```

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)
{
    rootView = inflater.inflate(R.layout.fragment_categoria, container, false);
    final FragmentManager fragmentManager = getActivity().getSupportFragmentManager();

    // Recupera a lista
    list = (ListView) rootView.findViewById(R.id.lista_categorias);

    // Popula a lista de categorias
    PopulateListView();

    list.setOnItemClickListener( (parent, view, position, id) -> {

        // Recupera o ID da categoria selecionada
        String id_categoria_selecionada = recuperaElementoLista(Integer.toString(position));

        // Cria Argumentos para o Fragmento
        Bundle bundle = new Bundle();
        bundle.putString("id_categoria", id_categoria_selecionada);

        // Cria um fragmento de livros e adiciona os argumentos
        FragmentLivros fragmento = new FragmentLivros();
        fragmento.setArguments(bundle);

        // Redireciona para a tela de livros passando o id da categoria para realizar o filtro
        fragmentManager.beginTransaction().replace(R.id.container, fragmento).commit();

    });

    return rootView;
}

```

A leitura dos arquivos .epub foi realizada com o auxílio da biblioteca Epublib. Ela está disponível para Android e JRE. Ela possui uma função pública chamada *ReadEpub(InputStream steam)*, que recebe um *InputStream* contendo o arquivo no formato epub como parâmetro. Podemos ver isso exemplificado na imagem abaixo.

Figura 42 - Função que carrega o livro selecionado.
Fonte: Própria

```
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)
{
    //Recupera o fragmento
    rootView = inflater.inflate(R.layout.fragment_livro, container, false);

    // Recupera a lista
    list = (ListView) rootView.findViewById(R.id.lista_de_livros);

    // Recupera o parâmetro
    Boolean possuiIdCategoria = verificaParametros();

    //popula a lista de livros
    PopulateListView( possuiIdCategoria );

    list.setOnItemClickListener( (parent, view, position, id) -> {

        // Recupera o ID do livro selecionado
        String id_livro_selecionado = recuperaElementoLista(Integer.toString(position));

        // Recupera o caminho do arquivo epub
        String caminho_livro = recuperaArquivo(id_livro_selecionado);

        EpubReader epubReader = new EpubReader();

        try
        {
            Book livro = epubReader.readEpub(new FileInputStream(caminho_livro));
        }
    }
}
```

3.6. RESULTADOS OBTIDOS

3.6.1. Aplicativo *Mobile*

- **Tela inicial**

Na tela abaixo, o usuário pode inserir seu nome de usuário e senha, para realizar o *login*, ou clicar na opção “Criar conta” caso não possua uma.

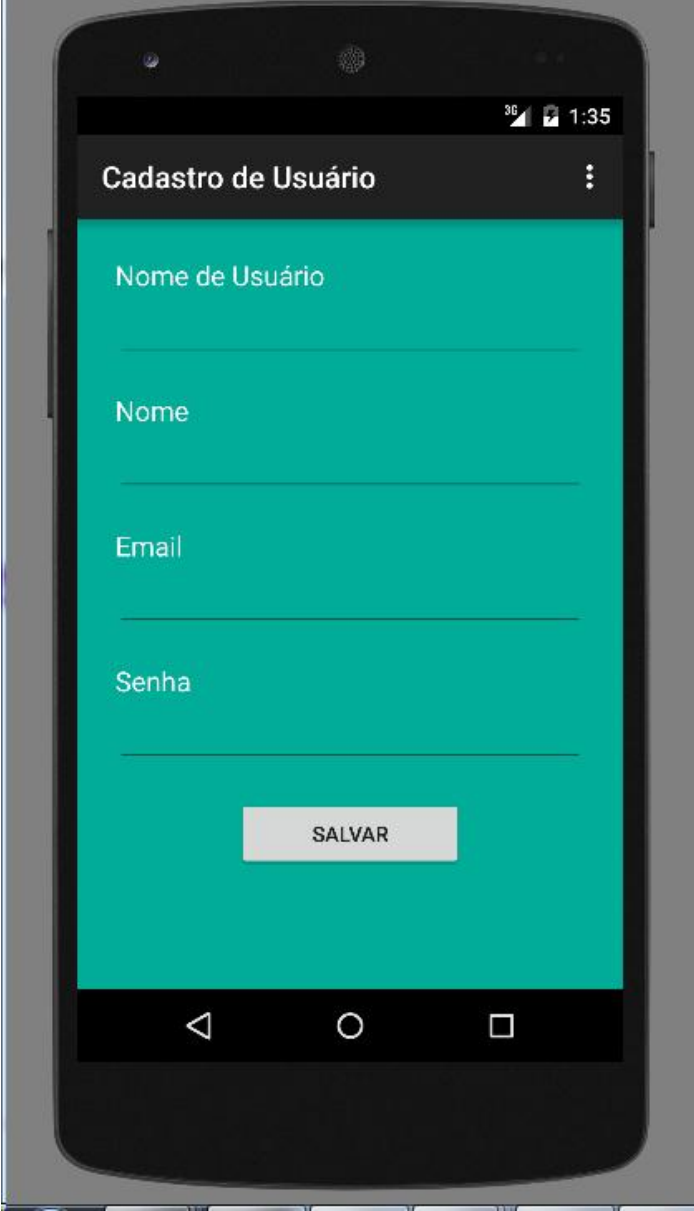
Figura 43 - Tela inicial do aplicativo móvel.
Fonte: Própria



- **Cadastro de usuário**

Na tela abaixo, o usuário insere seus dados e clica no botão “Salvar” para realizar um novo cadastro.

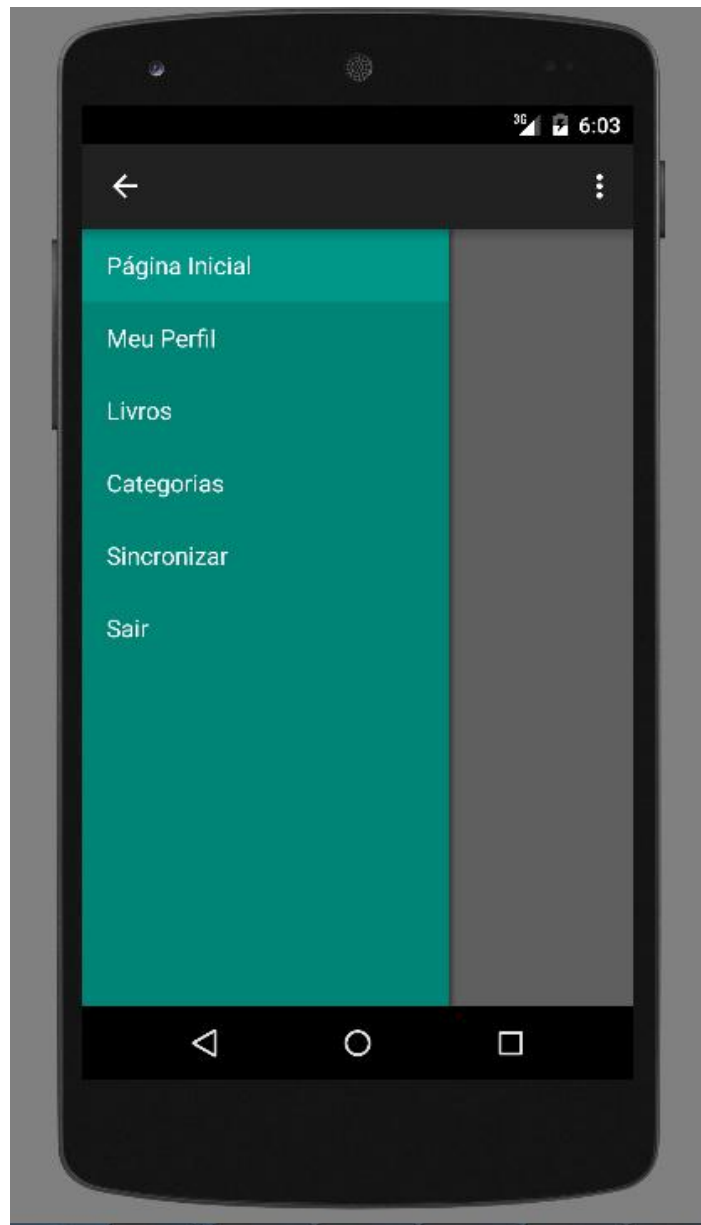
Figura 44 - Tela de cadastro de usuário do aplicativo móvel.
Fonte: Própria

A imagem mostra a interface de usuário de um aplicativo móvel para cadastro. O título da tela é "Cadastro de Usuário". O formulário contém quatro campos de entrada: "Nome de Usuário", "Nome", "Email" e "Senha". Cada campo é precedido por um ícone de lupa para busca. Abaixo dos campos, há um botão cinza com o texto "SALVAR". O status bar no topo indica 3G, uma bateria e o horário 1:35. O rodapé do aplicativo mostra os ícones de navegação padrão do Android (seta para trás, círculo e quadrado).

- **Menu principal**

Tela contendo as opções do menu principal.

Figura 45 - Tela do menu principal.
Fonte: Própria

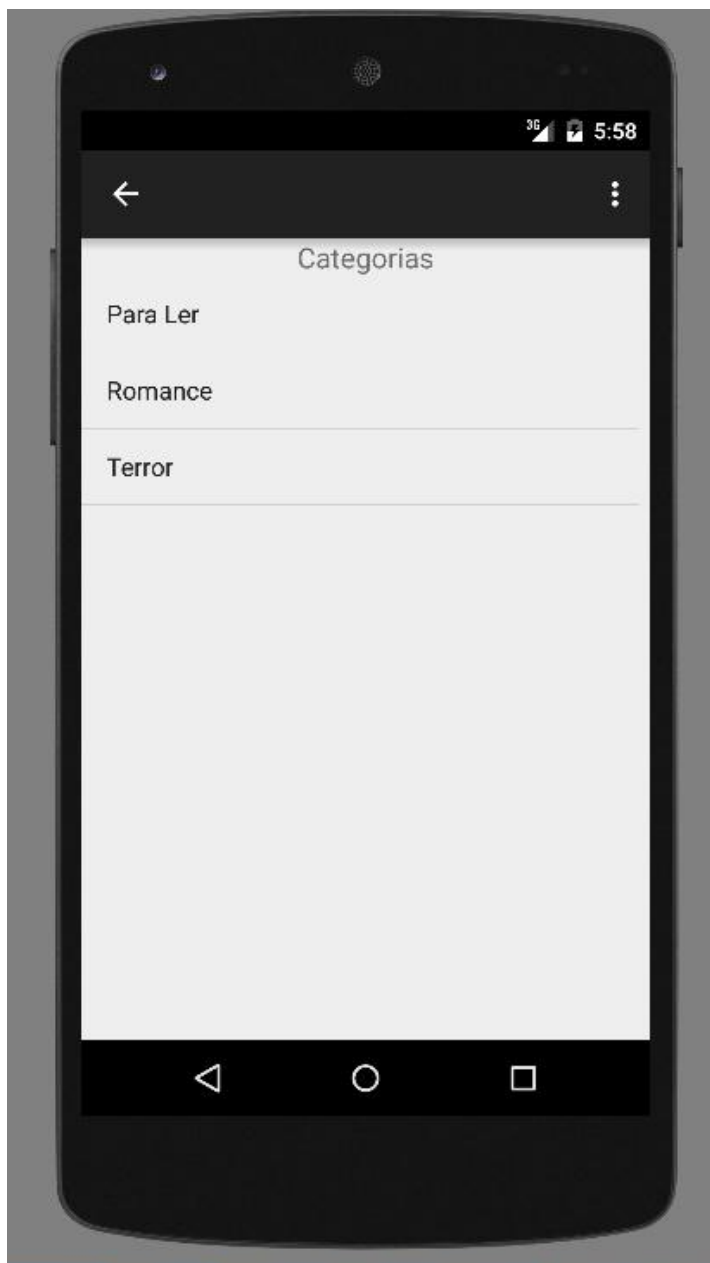


- **Lista de categorias**

Na tela abaixo, é exibida a lista com as categorias criadas pelo usuário.

Figura 46 - Tela de lista de categorias.

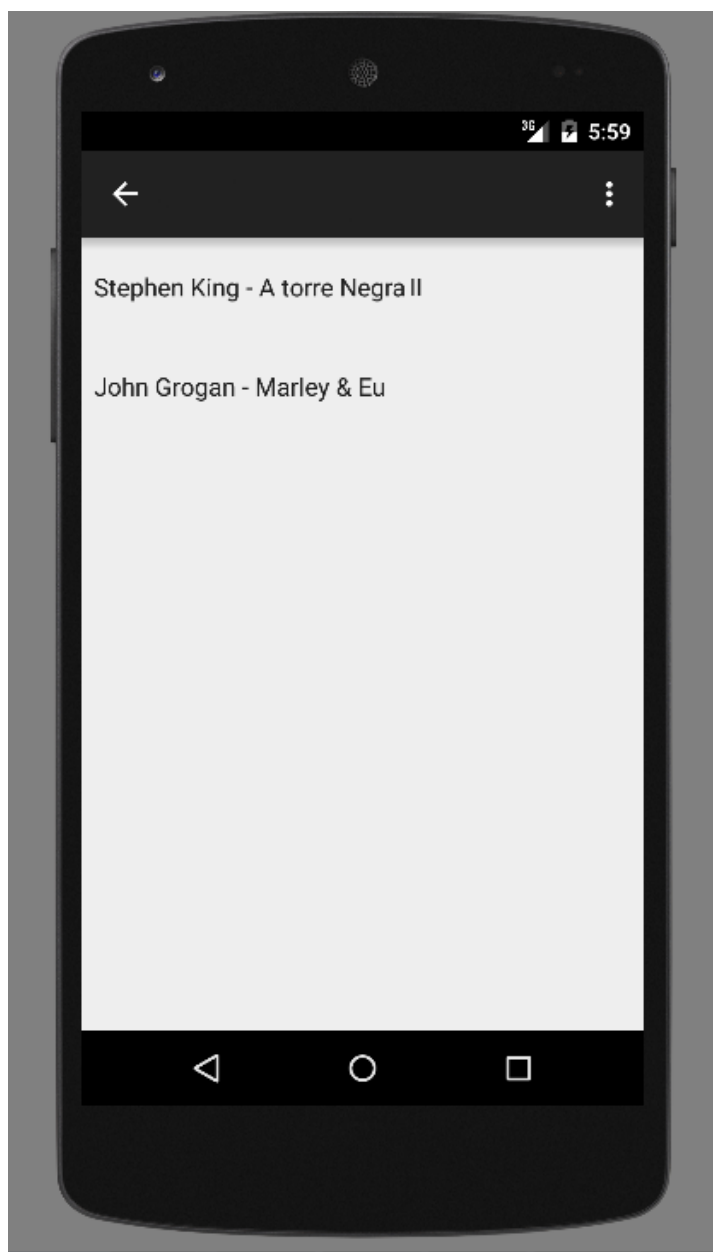
Fonte: Própria



- **Lista de livros**

Na tela abaixo, é exibida a lista com os livros do usuário, pertencentes à categoria “Para Ler”.

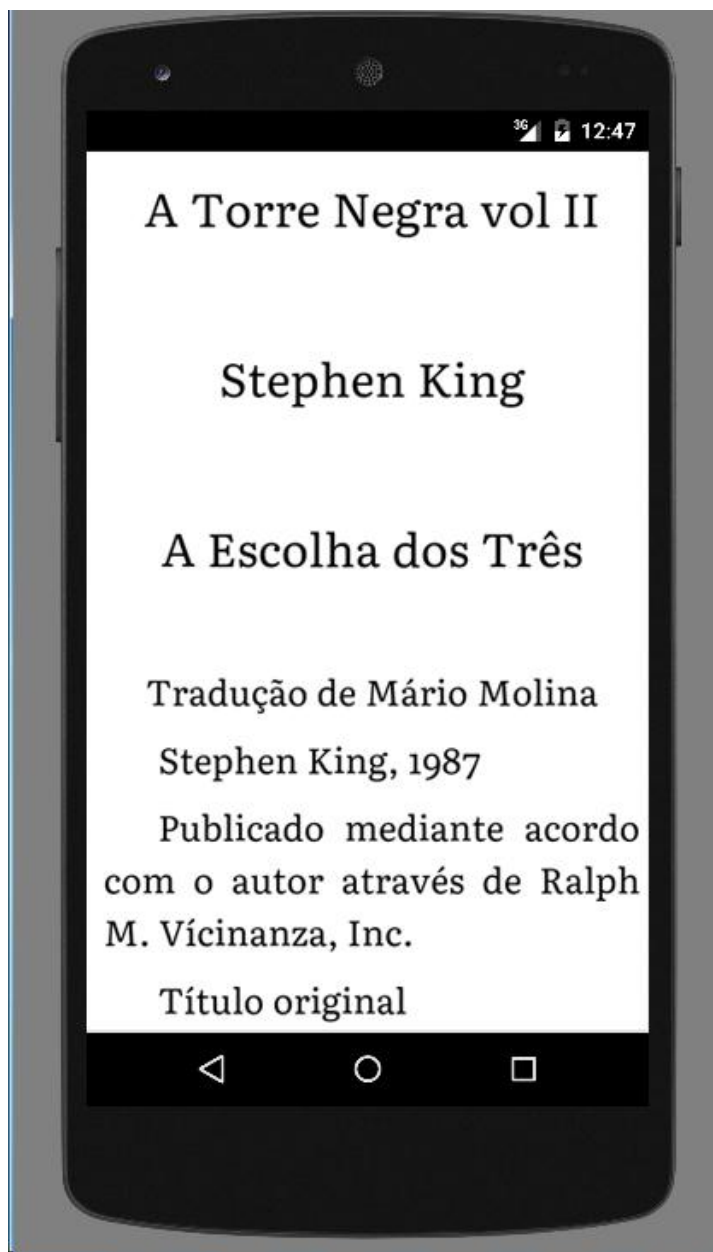
Figura 47 - Tela de lista de livros da categoria “Para Ler”.
Fonte: Própria



Leitura do livro

Na tela abaixo, é feita a leitura dos arquivos no formato epub.

Figura 48 - Tela de leitura do livro em formato epub.
Fonte: Própria

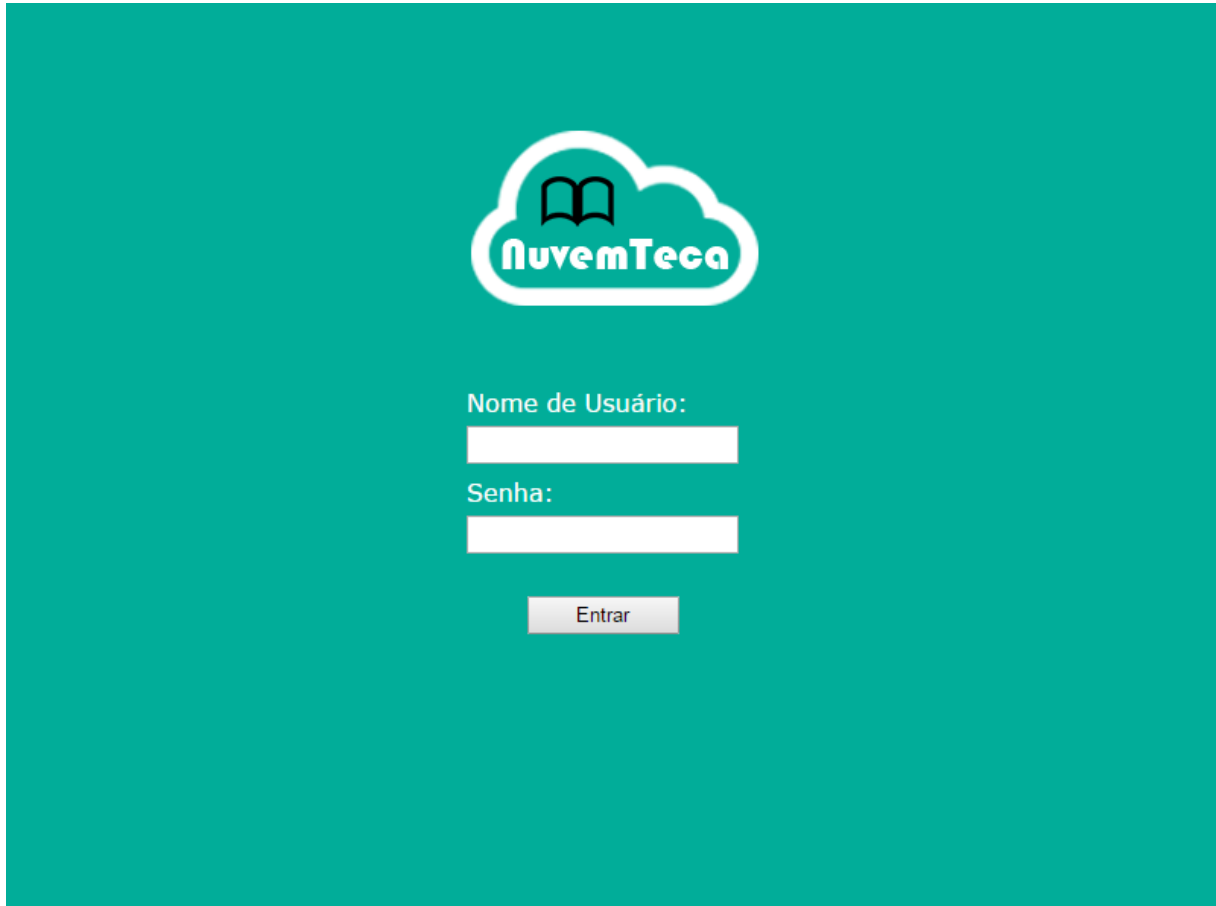


3.6.2. INTERFACE WEB

- **Tela inicial**

Na tela abaixo, o usuário realiza o *login* no sistema.

Figura 49 - Tela inicial da interface web.
Fonte: Própria



A imagem mostra a tela inicial da interface web do NuvemTeca. O fundo é verde-água. No topo central, há um logotipo branco que consiste em uma nuvem com um ícone de livro aberto dentro dela e o texto "NuvemTeca" abaixo. Abaixo do logotipo, há dois campos de entrada de texto brancos. O primeiro campo é precedido pelo rótulo "Nome de Usuário:" e o segundo pelo rótulo "Senha:". Abaixo dos campos, há um botão retangular branco com o texto "Entrar" em preto.

- **Tela de perfil do usuário**

Na tela abaixo, o usuário pode alterar seus dados cadastrais.

Figura 50 - Tela de perfil do usuário.
Fonte: Própria

NuvemTeca

Meu Perfil

Meus Livros

Minhas Categorias

Sair

Meu Perfil

Nick: maylc

Nome: Mayara

Email: mayaralcabral@gmail.com

Senha: ...

Salvar Alterações

- **Tela de livros de usuário**

Na tela abaixo, o usuário pode incluir, editar ou excluir os seus livros.

Figura 51 - Tela de livros do usuário.
Fonte: Própria



- **Tela de cadastro de livros**

Na tela abaixo, o usuário insere os dados do livro e escolhe o arquivo .epub correspondente:

Figura 52 - Tela de cadastro de livros.

Fonte: Própria



The image shows a web interface for registering a new book. On the left is a dark teal sidebar with the 'NuvemTeca' logo (a cloud with an open book) and a menu with the following items: 'Meu Perfil', 'Meus Livros', 'Minhas Categorias', and 'Sair'. The main area has a teal background and is titled 'Novo Livro'. It contains three input fields: 'Título:', 'Autor:', and 'Arquivo:'. The 'Arquivo:' field is a file selector showing 'Escolher arquivo' and 'Nenhum arquivo selecionado'. Below these fields is a button labeled 'Cadastrar Livro'.

- **Tela de categorias do usuário**

Na tela abaixo, o usuário pode incluir, visualizar, editar ou excluir categorias.

Figura 53 - Tela de categorias do usuário.


Fonte: Própria



- **Tela de cadastro de categorias**

Na tela abaixo, o usuário cria uma nova categoria e associa livros à ela. Caso ele não quera associar os livros no momento da criação, ele pode acessar a opção de editar categoria na tela de visualização de categorias e vincular livros a ela a qualquer momento.

Figura 54 - Tela de cadastro de categorias.
Fonte: Própria



A interface de cadastro de categorias do NuvemTeca apresenta um layout dividido em duas colunas. A coluna da esquerda, com fundo verde escuro, contém o logo 'NuvemTeca' (uma nuvem branca com um ícone de livro aberto) e um menu de navegação com os itens: 'Meu Perfil', 'Meus Livros', 'Minhas Categorias' (destacado em branco) e 'Sair'. A coluna da direita, com fundo verde claro, é o formulário principal. No topo, o título 'Nova Categoria' é seguido por uma linha decorativa. Abaixo, há um campo 'Nome:' com o texto 'Nova Categoria' inserido. Segue-se o campo 'Livros da Categoria:' com um ícone de livro e uma seta verde. Um container de seleção de livros exibe duas opções: 'Stephen King - A torre Negra' e 'John Grogan - Marley & Eu', cada uma com um ícone de documento e uma seta vermelha. No final, há um botão 'Cadastrar Categoria' e uma linha decorativa.

4. CONCLUSÃO

A escolha do tema abordado neste trabalho se deu através de uma necessidade pessoal, pois tenho o costume de ler livros em formato digital há bastante tempo. Porém, a falta de uma aplicação que atendesse a todas as minhas necessidades acabou me fazendo comprar um dispositivo específico para isso: o Kindle. No entanto, a necessidade de um sistema gratuito, de fácil utilização e com sincronização entre diversos dispositivos continuou e acabou motivando este projeto.

Seguindo a tendência observada há algum tempo, as aplicações que fazem uso de recursos na nuvem estão se tornando cada vez mais populares. Com o passar do tempo, grandes empresas como a Microsoft enxergaram nesse tipo de abordagem uma ótima oportunidade de negócio, capaz de atrair maiores parcelas do mercado que antes dependiam de uma infraestrutura de base para o seu funcionamento e agora podem “terceirizar” esse serviço.

A aplicação desenvolvida mostrou-se muito eficiente, pois cumpriu com todos os objetivos definidos e apresentou resultados dentro do esperado. A intenção de disponibilizá-la comercialmente ainda será avaliada, uma vez que cada dia que passa, devido à facilidade do desenvolvimento, novas aplicações surgem no mercado. Isso acaba tornando a competitividade extremamente acirrada e exigindo um tempo elevado de dedicação para manter um produto.

4.1. TRABALHOS FUTUROS

Como sugestão para trabalhos futuros, seria interessante a possibilidade de integrar todo o sistema de cadastro e *login* com as redes sociais. Dessa forma, seria possível compartilhar livros e categorias com os amigos, sugerir novas leituras, acompanhar o andamento da leitura de uma pessoa específica, entre outros.

Além disso, é importante também expandir as funcionalidades de inserção e edição de livros e categorias para o aplicativo móvel, além da interface *web*.

BIBLIOGRAFIA

Office of the Privacy Commissioner of Canada, Introduction to Cloud Computing. Disponível em: <https://www.priv.gc.ca/resource/fs-fi/02_05_d_51_cc_e.pdf>. Acessado em: 03/2015.

Cloud Computing Deployment Models. Disponível em: <<http://cloud.cio.gov/topics/cloud-computing-deployment-models>>. Acessado em: 03/2015.

MELL, Peter, GRANCE, Timothy. The NIST Definition of Cloud Computing. Disponível em: <<http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>>. Acessado em: 03/2015.

RYAN, Mark. Cloud Computing Privacy Concerns on Our Doorstep. Disponível em: <<http://cacm.acm.org/magazines/2011/1/103200-cloud-computing-privacy-concerns-on-our-doorstep/fulltext>>. Acessado em: 03/2015.

Serviços de cloud computing PaaS: um guia para desenvolvedores Java. Disponível em: <http://www.infoq.com/br/articles/paas_comparison>. Acessado em: 04/2015.

ROUSE, Margare. Platform as a Service (PaaS). Disponível em: <<http://searchcloudcomputing.techtarget.com/definition/Platform-as-a-Service-PaaS>>. Acessado em: 04/2015.

SaaS – Software as a Service. Disponível em: <<https://blog.vindi.com.br/saas-software-as-a-service>>. Acessado em: 04/2015.

What is Cloud Computing? Disponível em: <<http://www.cloudpeople.com.au/cloud-computing>>. Acessado em: 04/2015.

Cloud Computing Security – Risks to Cloud Computing Security and Possible Solutions. Disponível em: <<http://www.bigdatacloudhelp.com/cloud-computing-security>>. Acessado em: 04/2015.

Android Developer, Get Started with Android Studio. Disponível em: <<http://developer.android.com>>. Acessado em: 05/2015.

Epublib, Epublib – a java epub library. Disponível em: <<http://siegmänn.nl/epublib>>. Acessado em: 05/2015.

APÊNDICE - DEFINIÇÕES

Android

Sistema operacional para *smartphones* do Google.

Aplicativo mobile

Aplicativo desenvolvido para dispositivos móveis (*smartphones*, *tablets*, etc).

Backup

Cópias de segurança de determinados arquivos ou dados.

Banco de dados

Estrutura onde são armazenados dados e arquivos.

Cliente

Entidade que solicita e utiliza recursos computacionais do servidor e/ou provedor.

Epub

Formato de arquivo para livros digitais.

Hardware

Componentes físicos de um sistema de computação.

Java

Linguagem de programação.

Linux

Sistema operacional de código aberto para computadores pessoais.

Mac OSX

Sistema operacional para computadores pessoais, desenvolvido pela Apple.

Mainframe

Computadores de alto desempenho, geralmente utilizados por grandes organizações e/ou governos para aplicações críticas.

Provedor

Entidade fornecedora de recursos

Roteador

Dispositivo que encaminha pacotes de dados entre redes de computadores.

Servidor

Estrutura onde são armazenados e fornecidos recursos computacionais.

Servlet

Classe Java usada para estender as funcionalidades de um servidor.

Smartphones

Aparelhos de telefonia móvel com recursos computacionais avançados.

Software

Programas de computador.

Web Service

Solução utilizada na integração de sistemas e na comunicação entre aplicações diferentes.

Webmail

Cliente de e-mail que utiliza o navegador como estrutura.

Windows

Sistema operacional para computadores pessoais, desenvolvido pela Microsoft.

XML

Formato de arquivo.