

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ – UTFPR  
DEPARTAMENTO ACADÊMICO DE COMPUTAÇÃO  
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

**FELIPE ANTONIO ZADINELLO**

**PROPOSTA DE APLICATIVO REPOSITÓRIO DE BIBLIOTECAS  
VIRTUAIS APLICANDO AVALIAÇÃO HEURÍSTICA.**

**TRABALHO DE CONCLUSÃO DE CURSO**

**MEDIANEIRA**

**2015**

FELIPE ANTONIO ZADINELLO

**PROPOSTA DE APLICATIVO REPOSITÓRIO DE BIBLIOTECAS VIRTUAIS  
APLICANDO AVALIAÇÃO HEURÍSTICA.**

Trabalho de Conclusão de Curso de Graduação do Curso Superior de Bacharelado em Ciência da Computação do Departamento Acadêmico de Computação – DACOM – da Universidade Tecnológica Federal do Paraná – UTFPR - Câmpus Medianeira, como requisito para obtenção do Título de Bacharel.

Orientador: Prof. Ricardo Sobjak  
Co-Orientadora: Prof.<sup>a</sup> Alessandra Bortoletto Garbelotti Hoffmann

MEDIANEIRA

2015



---

## TERMO DE APROVAÇÃO

### PROPOSTA DE APLICATIVO REPOSITÓRIO DE BIBLIOTECAS VIRTUAIS APLICANDO AVALIAÇÃO HEURÍSTICA.

Por

**FELIPE ANTONIO ZADINELLO**

Este Trabalho de Conclusão de Curso (TCC) foi apresentado às 15:50 h do dia 19 de novembro de 2015 como requisito para a obtenção do título de Bacharel no Curso Superior de Ciência da Computação, da Universidade Tecnológica Federal do Paraná, *Campus* Medianeira. O acadêmico foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

---

Prof. Msc. Ricardo Sobjak  
UTFPR – *Campus* Medianeira  
(Orientador)

---

Prof. MSc Juliano Rodrigo Lamb  
UTFPR – *Campus* Medianeira  
(Convidado)

---

Prof. Dr. Arnaldo Cândido Júnior  
UTFPR – *Campus* Medianeira  
(Convidado)

---

Prof. MSc Jorge Aikes Junior  
UTFPR – *Campus* Medianeira  
(Responsável pelas atividades de TCC)

## **AGRADECIMENTOS**

Agradeço a meus pais, Alceu Antonio Zadinello e Salete Zamo Vargas Zadinello, que sempre acreditaram em mim e estiveram ao meu lado me apoiando e dando forças para seguir em frente.

Aos professores Ricardo Sobjak e Alessandra Bortoletto Garbelotti Hoffmann, pelo conhecimento, orientação, incentivo, críticas e sugestões que ajudaram a construir este trabalho.

Aos meus amigos de curso que tive a honra de conhecer e que contribuíram para a conclusão deste trabalho.

## RESUMO

ZADINELLO, Felipe Antonio. PROPOSTA DE APLICATIVO REPOSITÓRIO DE BIBLIOTECAS VIRTUAIS APLICANDO AVALIAÇÃO HEURÍSTICA. Trabalho de conclusão do curso de superior de Ciência da Computação. Universidade Tecnológica Federal do Paraná. Medianeira, 2015, 41p.

As bibliotecas virtuais dispõem de conteúdo através da Internet, porém diversas bibliotecas trazem aos usuários diferentes maneiras de realizar o mesmo objetivo, deste modo o presente trabalho tem como principal objetivo a modelagem e implementação de um protótipo de aplicativo de um repositório de bibliotecas virtuais para dispositivos móveis para tornar-se apenas um único ambiente de diversas bibliotecas. Para elaboração do protótipo utilizou-se o método de avaliações heurísticas de *design* em bibliotecas existentes e no desenvolvimento do protótipo utilizou-se Android. O estudo compõe-se de revisão da literatura de Interface Humano Computador, *Web Services* e Dados. O protótipo desenvolvido é compatível com os *smartphones* equipados com o sistema Android.

**Palavras-chaves:** *design*, interface humano computador, Android

## ABSTRACT

ZADINELLO, Felipe Antonio. PROPOSED APPLICATION REPOSITORY LIBRARIES VIRTUAL ASSESSMENT APPLYING HEURISTICS. Trabalho de conclusão do curso de superior de Ciência da Computação. Universidade Tecnológica Federal do Paraná. Medianeira, 2015, 41p.

Virtual libraries it offers content through the Internet, but many libraries brings to the users many ways to do the same thing, so the current work see as main objective the modeling and implementation of a prototype of a virtual library repository for mobile devices to be the only platform to many virtual libraries. To elaborate this prototype was used *design* avaliation in existing virtual libraries and on development of the prototype was used Android. This study consists in a literature review of Interface Human Computer, Web Services and Data. The prototype developed is compatible with smartphones equipped with Android Operational System.

**Keywords:** *design*, virtual libraries virtuais, human computer interaction, android

## LISTA DE FIGURAS

Figura 1 - Exemplo de Web Service. ....	13
Figura 2 - Atividades do <i>Design</i> . ....	16
Figura 3 - Página principal da biblioteca virtual Arquivo Público do Estado de São Paulo. ....	23
Figura 4 – Página principal da biblioteca virtual Bibliomania. ....	24
Figura 5 - Página principal da biblioteca virtual Domínio Público. ....	24
Figura 6 - Processo de <i>design</i> de IHC. ....	25
Figura 7 - Página principal da biblioteca virtual Domínio público acessada a partir de um navegador de dispositivo móvel (smartphone). ....	25
Figura 8 - Página principal da biblioteca virtual Bibliomania acessada a partir de um navegador de dispositivo móvel (smartphone) ....	25
Figura 9 - Página principal da biblioteca virtual Arquivo Público do Estado de São Paulo acessada a partir de um navegador de dispositivo móvel (smartphone). ....	26
Figura 10 - Página principal da biblioteca virtual Arquivo Público do Estado de São Paulo acessada a partir de um navegador de dispositivo móvel (smartphone). ....	26
Figura 11 - Arquitetura do Projeto ....	28
Figura 12 - Lista de Bibliotecas. ....	32
Figura 13 - Lista de Bibliotecas. ....	32
Figura 14 - Lista de Documentos.....	33
Figura 15 - Busca na biblioteca ....	33

## LISTA DE QUADROS

Quadro 1 - Exemplo de objeto JSON. ....	13
Quadro 2 - Exemplo de lista de objetos JSON. ....	14
Quadro 3 - Interface inicial de <i>Web Service</i> . ....	30
Quadro 4 - Modelo JSON. ....	31
Quadro 5 - Objeto Java pelo GSON. ....	33
Quadro 6 - Processo de requisição ao <i>WebService</i> . ....	34



## LISTA DE TABELAS

Tabela 1 - Comparativo entre Heurísticas de Nielsen e Princípios do Design de Benyon.....	21
Tabela 2 - Heurísticas violadas pelas bibliotecas virtuais .....	27
Tabela 3 - Análise de heurísticas da interface .....	32

## SUMÁRIO

<b>AGRADECIMENTOS</b> .....	<b>2</b>
<b>RESUMO</b> .....	<b>2</b>
<b>ABSTRACT</b> .....	<b>3</b>
<b>1 INTRODUÇÃO</b> .....	<b>8</b>
1.1 OBJETIVO GERAL .....	9
1.2 OBJETIVOS ESPECÍFICOS .....	9
1.3 JUSTIFICATIVA .....	9
<b>2 FUNDAMENTAÇÃO TEÓRICA</b> .....	<b>11</b>
2.1 WEB SERVICE .....	11
2.1.1 REST .....	11
2.2 JAVASCRIPT OBJECT NOTATION.....	13
2.3 ANDROID .....	14
2.4 INTERFACE HUMANO COMPUTADOR (IHC) .....	15
2.4.1 <i>DESIGN</i> .....	16
2.4.2 AVALIAÇÃO DE SISTEMAS .....	18
2.4.3 Avaliação Heurística .....	19
<b>3 MATERIAL E MÉTODOS</b> .....	<b>23</b>
3.1 IDENTIFICAÇÃO DAS BIBLIOTECAS VIRTUAIS .....	23
3.2 PROCEDIMENTOS PARA DESENVOLVIMENTO DO APLICATIVO .....	28
<b>4 RESULTADOS E DISCUSSÃO</b> .....	<b>30</b>
4.1 PROPOSTA DE WEBSERVICE .....	30
4.2 DESENVOLVIMENTO DO PROTÓTIPO .....	31
<b>5 CONSIDERAÇÕES FINAIS</b> .....	<b>35</b>
5.1 CONCLUSÕES .....	35
5.2 TRABALHOS FUTUROS .....	36
<b>REFERÊNCIAS</b> .....	<b>37</b>

## 1 INTRODUÇÃO

Tecnologias como *tablets* e *smartphones* proporcionam ao usuário utiliza-las a todo o momento e são conhecidas como uma tecnologia móvel ou *mobile*. Esse segmento da tecnologia móvel, evoluiu rápido em virtude da possibilidade de acessar dados e informações em qualquer lugar e a qualquer momento. Essa mobilidade que já foi considerada uma facilidade para as tarefas cotidianas e profissionais, no entanto é uma necessidade para o usuário.

As bibliotecas virtuais são uma maneira de popularizar a informação sem restrições geográfica, utilizando a Internet como o meio entre usuário e a informação desejada. As informações por estarem em diferentes locais podem acarretar dificuldade em seu uso.

Devido às bibliotecas virtuais serem desenvolvidas para diferentes instituições, privadas ou públicas, ocorre que o acesso à informação torna-se complicada, pois cada uma é construída com um *design* e/ou filtros próprios, assim o usuário tem uma curva de aprendizado diferente para cada biblioteca virtual.

Desse modo, este trabalho tem como objetivo, realizar avaliações de interface com técnicas de Interface Humano Computador (IHC) e desenvolver para Android um protótipo de aplicativo chamado Repositório de Bibliotecas Virtuais, o qual possibilitará um fácil acesso e utilizando a mesma plataforma com uma interface amigável e de fácil acesso a informações em diferentes bibliotecas virtuais utilizando à mesma plataforma, descomplicando o acesso à informação.

## 1.1 OBJETIVO GERAL

Realizar análises de interface utilizando avaliação heurística e desenvolver um protótipo de aplicativo móvel que simule o acesso à algumas bibliotecas virtuais.

## 1.2 OBJETIVOS ESPECÍFICOS

O desenvolvimento do projeto será realizado conforme os seguintes objetivos específicos:

- a. Pesquisar e apresentar conceitos por meio de um referencial teórico sobre Interface Humano Computador (IHC), JSON e Webservice.
- b. Selecionar bibliotecas virtuais para realização do estudo.
- c. Avaliar as bibliotecas virtuais acessíveis em dispositivos móveis a partir de técnicas de IHC.
- d. Elaborar um protótipo de um repositório de bibliotecas virtuais para dispositivos móveis seguindo conceitos de IHC.

## 1.3 JUSTIFICATIVA

As Bibliotecas Virtuais são uma realidade e estão presentes há alguns anos na Internet e permitem um conceito de “tempo” e “lugar” diferente para bibliotecas. O conceito de "lugar" torna-se secundário, tanto para bibliotecários quanto para usuários, pois o acesso se torna pela Internet. O que é importante passa a ser o acesso e, com frequência, a "confiabilidade" da informação (LEVACOV, 1997). O “tempo” torna-se um conceito de instantaneidade, ou seja, enfatizando a velocidade o que permite rápido acesso e facilidade em publicações.

Com o aumento do consumo, ou seja a compra, de *smartphones* em relação aos Desktops (UOL, 2011), surge uma mudança de paradigma entre biblioteca física e virtual, computador e celular.

Este trabalho tem como propósito por meio de avaliações de interface apresentar um modelo para incorporar as bibliotecas virtuais em um único padrão para as tecnologias portáteis, especificamente o Android e com este modelo, é possível permitir mais uma maneira de oferecer à educação outro caminho ao público interessado.

## 2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção são apresentados conhecimentos básicos para a compreensão do assunto e conceitos utilizados.

### 2.1 WEB SERVICE

*Web service* é uma aplicação, cliente e/ou servidor, que se comunica por meio da *World Wide Web's* (WWW) e do *HyperText Transfer Protocol* (HTTP). Como descrito pelo *World Wide Web Consortium* (W3C), *web services* fornecem meios para interoperar com *software* de diversas plataformas e/ou *frameworks*. *Webs services* também são caracterizadas por sua grande interoperabilidade e extensibilidade, assim como seu processamento de dados, geralmente utilizando JSON ou XML (ORACLE, 2013).

*Web service* fornece recursos, no qual o serviço pode ser acessado na aplicação cliente, por meio da Internet. O consumidor e provedor de serviços utilizam mensagens para trocar informações de requisições e respostas na forma de documentos que fazem poucas suposições das capacidades tecnológicas do receptor (ORACLE, 2013).

#### 2.1.1 REST

REST é uma abreviação de *Representational State Transfer* (Transferência de Estado Representacional) e é um estilo de desenvolvimento de *web services* que teve origem na tese de doutorado de Roy Fielding. Este, por sua vez, é co-autor de um dos protocolos mais utilizados no mundo, o HTTP (*HyperText Transfer Protocol*).

Assim, é notável que o protocolo REST é guiado (dentre outros preceitos) pelo que seriam as boas práticas de uso de HTTP (SAUDATE, 2012).

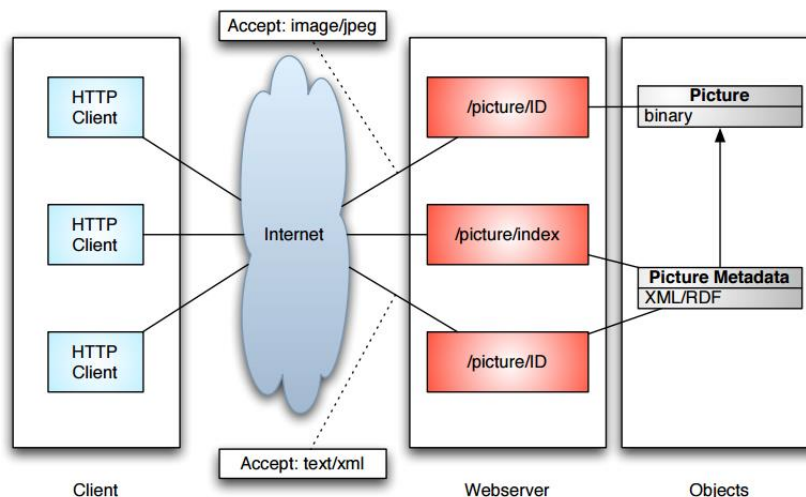
REST depende de dois “papéis”, cliente e servidor e é baseado no protocolo HTTP. O propósito é, utilizar simples chamadas HTTP para conectar máquinas. (LIMA, 2012).

REST é dividido em três classes de elementos: *data elements* (dados), *connecting elements* (conectores) e *processing elements* (componentes) (JAKL, 2005).

Um fundamento do REST é o estado dos dados, seus componentes se comunicam transferindo representações do estado do elemento desejado. Essas comunicações ocorrem por meio de *Resources* (recursos). Tudo o que pode ser nomeado pode ser um recurso, a arquitetura REST usa *Uniform Resource Identifier* (URI) (JAKL, 2005).

Segundo Saudate (2012), “URL significa *Universal Resource Locator* e URI, *Universal Resource Identifier*. Uma URI, como diz o próprio nome, pode ser utilizada para identificar qualquer recursos em um *Web Service* - dar um caminho para um determinado conteúdo, dar nome à este. Já uma URL pode ser utilizada apenas para fornecer caminhos - sendo que uma URL é, portanto, uma forma de um URI”.

Na Figura 1 é mostrado um exemplo de um *Web Service* baseado em REST, no qual um cliente por meio de uma requisição HTTP acessa um recurso (URI). A cada nova requisição deve conter toda a informação necessária para completá-la e não deve depender em interações prévias com o mesmo cliente, isto é, não depende de nenhum estado prévio determinando um comportamento *stateless*.



**Figura 1 - Exemplo de Web Service.**  
**Fonte: JAKL (2005).**

## 2.2 JAVASCRIPT OBJECT NOTATION

*JavaScript Object Notation* conhecido como JSON é uma linguagem de marcação criada por Douglas Crockford e serve como uma equivalência ao *eXtensible Markup Language* (XML). O próprio autor diz que a principal motivação é o tamanho reduzido se comparado ao XML (SAUDATE, 2012)

O modelo JSON atende aos seguintes critérios:

1. Um objeto contém zero ou mais membros;
2. Um membro contém zero ou mais pares e zero ou mais membros;
3. Um par contém uma chave e um valor, apresentado no Quadro 1;
4. Um membro também pode ser um lista, apresentado no Quadro 2.

```

{"nome do objeto" : {
  "nome do par" : "valor do par"
}
}

```

**Quadro 1 - Exemplo de objeto JSON.**  
**Fonte: (SAUDATE, 2012).**



```
{ "nome do objeto" : [  
  { "nome do elemento" : "valor" },  
  { "nome do elemento" : "valor" }  
 ]  
 }
```

Quadro 2 - Exemplo de lista de objetos JSON.  
Fonte: (SAUDATE, 2012).

O JSON é um formato para intercâmbio de dados, é um formato-texto utilizado para trocar dados entre plataforma, com o qual muitos sistemas concordaram em usar na comunicação de dados. JSON também é conhecido como um “formato para intercâmbio de dados” ou simplesmente como um “formato de dados”. Esse formato é importante para trocar dados entre sistemas diferentes (BASSETT, 2015).

Diferente do formato XML, JSON não usa uma chave de fechamento (*tag end*), é menor, mais rápido de ler e escrever e pode utilizar listas (*array*). Porém a maior diferença é que o XML tem que ser analisado com um *XML Parser*, um analisador externo a aplicação, enquanto o JSON pode ser analisado por uma função padrão do JavaScript (W3SCHOOLS, 2015).

## 2.3 ANDROID

Android é o nome de um sistema operacional mantido por uma empresa americana chamada Google. Android por ser um grande sistema operacional do mercado móvel, destaca-se em smartphones, *tablets* relógios, TV e até mesmo carros.

Desenvolver para Android é trabalhar com um conjunto de ferramentas de desenvolvimento de software para telefones celulares, criado pelo Google e pela *Open Handset Alliance*. Android está presente dentro de milhões de celulares e outros dispositivos, tornando o Android uma plataforma majoritária para desenvolvedores de aplicações. (BURNETTE, 2009)

Por ser código aberto, o Android possui diversas bibliotecas e *frameworks* que auxiliam no desenvolvimento. Programar para Android é programar em Java por meio das bibliotecas que o Google disponibiliza para o desenvolvedor.

O sistema operacional Android é baseado no Linux e teve seu começo no ano de 2003 pela empresa Android Inc, sendo comprada pelo Google no ano de 2005. (MONTEIRO, 2012)

## 2.4 INTERFACE HUMANO COMPUTADOR (IHC)

A IHC, tem por objetivo proporcionar análise de um projeto (*design*), avaliação e implementação de sistemas computacionais interativos para humanos.

Deste modo, analisar a natureza da interação é estudar o que ocorre quando um usuário utiliza um sistema interativo em suas atividades, com isso existe a possibilidade de descrever, explicar e prever fenômenos e consequências desse uso na vida das pessoas (BARBOSA; SILVA, 2010).

Segundo Prates e Barbosa (2006), para saber se um software é adequado aos usuários, tanto nas funcionalidades e no ambiente a ser utilizado, a avaliação da interface é importante para determinar a qualidade de uso do software, quanto maior a antecedência dos problemas de interface ou interação encontrados, menor o custo de tratá-los.

Alguns dos principais objetivos de realizar essa análise, de acordo com Prates e Barbosa (2006), são:

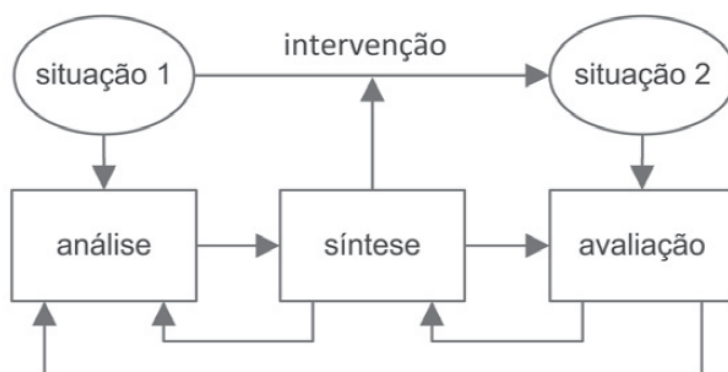
- a. Identificar as necessidades de usuários ou verificar o entendimento dos projetistas sobre estas necessidades;
- b. Identificar problemas de interação ou de interface;
- c. Investigar como uma interface afeta a forma de trabalhar dos usuários;
- d. Comparar alternativas de projeto de interface;
- e. Alcançar objetivos quantificáveis em métricas de usabilidade;
- f. Verificar conformidade com um padrão ou conjunto de heurísticas.

### 2.4.1 DESIGN

O produto resultante da inteligência e trabalho humano e que foi desenvolvido por um propósito específico, é chamado de artefato. O artefato não é originado na natureza, ele tem sua função, forma, estrutura, qualidade e é construído por alguém. Introduzir um artefato ao cotidiano de uma pessoa, traz consequências positivas e negativas. Junto ao artefato vem a análise da situação na qual pondera-se processos, artefatos, pessoas e demais elementos e suas relações na busca de problemas ou condições que possam ser melhoradas (BARBOSA; SILVA, 2010).

Segundo Barbosa e Silva (2010), pode-se definir o *design* da seguinte maneira (Figura 2):

- a. A análise da situação atual: estudar e interpretar a situação atual;
- b. A síntese de uma intervenção: planejar e executar uma intervenção na situação atual;
- c. A avaliação da nova situação: verificar o efeito da intervenção, comparando a situação analisada anteriormente com a nova situação, atingida após a intervenção.



**Figura 2 - Atividades do *Design*.**  
Fonte: (BARBOSA; SILVA, 2010).

A análise da situação atual que também nomeada de análise de problemas, nem sempre é relacionada a um problema, pois a situação atual pode ser satisfatória e que por conta de uma nova tecnologia, por exemplo, existe uma possibilidade de desenvolver uma situação mais desejável. Essa é uma situação chamada de “como melhorar a situação atual?”.

Quando se trata de sistemas computacionais, é normal analisar todos os elementos envolvidos durante o uso como, usuários e suas características, necessidades e preferências; as atividades e os objetivos em questão, considerando os artefatos e sistemas computacionais utilizados; e o contexto físico, social e cultural de uso ao longo do tempo (BARBOSA; SILVA, 2010).

Segundo Benyon (2011), o desenvolvimento de sistemas interativos preocupa-se com 12 princípios de *design*:

- 1 a 4 abordam acesso, facilidade de aprendizado e lembrança;
- 5 a 7 abordam facilidade de uso;
- 8 a 9 abordam segurança;
- 10 a 12 adaptabilidade às diferenças entre as pessoas e do respeito a essas diferenças.

A seguir é detalhado cada característica de um princípio (BENYON, 2011):

1. Visibilidade – permitir que o usuário possa ver qual função está disponível ou acontecendo no momento. Também é possível deixar ‘visível’ por meio do som ou toque;
2. Consistência – por ser um conceito indefinido, pois a consistência depende da situação, por exemplo um site de artes com intuito de chamar atenção descola obras de forma aleatória, situações onde ser inconsistente é bom, pois chama a atenção. De modo geral, consistência seria a coerência das ideias, lógica;
3. Familiaridade – utilizar linguagem e símbolos com os usuários estejam familiarizados, quando não for possível, utilizar metáforas para transmitir conhecimentos;
4. *Affordance* – crie o *design* das coisas de forma que fique explícita sua utilidade, ou seja, deixar claro as propriedades de um objeto e como é usada, por exemplo, um botão precisa parecer um botão e poder ser apertado;
5. Navegação – permitir que o usuário possa se movimentar pelo sistema;
6. Controle - comunique de forma evidente quem ou o que está no controle do sistema, permita que o usuário possa também assumi-lo;
7. Retorno (*Feedback*) – informação do resultado da ação do usuário, sempre retorne rapidamente e de forma consistente e constante;

8. Recuperação – torne possível recuperar de forma rápida e eficaz de ações, erros ou enganos;
9. Restrições – assegurar que há restrições que não permitam que o usuário realize ações inadequadas, assim impedindo de realizar erros graves;
10. Flexibilidade – permitir que haja diversas maneiras de realizar as ações, assim usuários com diferentes níveis de experiência possam ser atendidas. Permitir mudar a aparência e forma, deste modo o sistema torna-se personalizável;
11. Estilo – *design* deve ser elegante e atraente;
12. Sociabilidade – sistema deve possuir respostas educadas e ter diálogos amistosos com o usuário.

#### 2.4.2 AVALIAÇÃO DE SISTEMAS

Quando um sistema é avaliado, quer dizer que a ideia de *design* será revisada, experimentada ou testada para ver se o produto (*software*), atende aos critérios solicitados (BENYON, 2011).

Além disso, quando se avalia um sistema, também é levado em conta a sua usabilidade, que é refere-se a qualquer característica que afete o usuário no momento de realizar suas tarefas de forma eficiente e eficaz (VALIATI, 2008).

São apresentadas por Barbosa e Silva (2010), diversas maneiras de realizar uma avaliação IHC, sendo elas, inspeção (avalição heurística, percurso cognitivo e semiotica) e observação (teste de usabilidade, avaliação de comunicabilidade e prototipação em papel).

Quando se avalia o *design* de um sistema, também deve levar-se em conta como o usuário navega. Deste modo Benyon (2011), detalha algumas maneiras de navegação:

- Onisciência – usuário possui um conhecimento perfeito e não comete erros;

- Racionalidade – usuários raciocinam perfeitamente, porém só conhecem o que já viram;
- *Satisficing* – (do inglês *satisfy* + *suffice*), usuário fica satisfeito com os requisitos mínimos, deve-se tornar links e conteúdos importantes imediatamente disponíveis;
- Mapas mentais - usuários utilizam as pistas disponíveis para entender a estrutura do sistema;
- Memorização pelo hábito – usuários identificam um caminho que funciona e depois decoram e refazem este caminho;
- Busca de Informação – usuário procura o máximo de informação em único lugar;
- Custos de Informação – usuário possui conhecimento e raciocínio limitados, deve manter o custo mental minimizado, ou seja, lembrar, planejar e tomar decisões.

### 2.4.3 Avaliação Heurística

A avaliação heurística é um método de avaliação IHC realizado por especialistas em IHC e é usado para identificar problemas de usabilidade durante o processo de design.

A avaliação heurística tem como referência um conjunto de diretrizes de usabilidades que explica características desejáveis da interface e interação, conhecidas como heurísticas (BARBOSA; SILVA, 2010).

A análise do sistema avaliado com as heurísticas pré-determinadas, é a base da avaliação porém o avaliador tem liberdade para utilizar outros princípios relevantes a avaliação. (ROSA; VERAS, 2013).

Existem diversos grupos de heurísticas, como as dez heurísticas de Nielsen ou os princípios do *design*, citados anteriormente e que são usados neste trabalho. As heurísticas de Nielsen e os princípios do design são semelhantes pois abordam os mesmos critérios avaliados.

Deste modo, Rosa e Veras (2013) apresentam as 10 heurísticas de Nielsen e em seguida é apresentado uma relação entre os doze princípios do *design* descritos anteriormente, assim é evidenciado a relação e diferença entre as duas heurísticas.

1. **Visibilidade do estado do sistema:** deve-se manter os usuários informados do seu estado a cada momento, em tempo razoável, através de feedback apropriado;
2. **Correspondência entre o sistema e o mundo real:** deve-se falar a linguagem do usuário com convenções do mundo real, de maneira que a informação seja oferecida em uma ordem lógica e natural;
3. **Liberdade e controle por parte do usuário:** os usuários escolhem funções do sistema por engano, o qual lhes deve ser oferecida uma alternativa, sinalizada, para que possam sair do estado indesejado. Deve ser facilitada a opção de desfazer e refazer ações;
4. **Consistência e padrões:** Regras devem ser respeitadas, de modo a evitar que os usuários se perguntem a respeito de redundâncias acerca dos significados de palavras, ações ou situações, quando de fato esses significam a mesma coisa;
5. **Prevenção de erros:** O sistema deve eliminar as condições que levam a falhas e design que evite a ocorrência de problemas ou apresentar ao usuário opções de confirmação, antes dele executar determinadas ações
6. **Reconhecimento preferível à memorização:** As instruções de uso do sistema devem ser visíveis ou facilmente recuperáveis; minimizar a quantidade de informação que o usuário precisará lembrar para usá-lo.
7. **Flexibilidade e eficiência de uso:** Permitir que usuários mais experientes possuam maneiras mais rápidas e eficientes, invisível ao usuário inexperiente, de forma que o sistema possa atender aos dois perfis.
8. **Design estético e minimalista:** Cada unidade extra de informação, em um diálogo, concorre com as unidades de informação relevantes e diminui a sua visibilidade, assim não utilizar diálogos ou textos irrelevantes;
9. **Ajuda aos usuários para reconhecer, diagnosticar e se recuperar dos erros:** Mensagens de erro devem ser compreensível e sem

códigos, indicando o problema e sugerindo uma solução de forma construtiva;

10. **Suporte e documentação:** o sistema deve ser usado sem documentação, porém se necessário deve fornecer suporte e documentação ao usuário.

**Tabela 1 - Comparativo entre Heurísticas de Nielsen e Princípios do Design de Benyon**

Heurística de Nielsen	Princípios do Design	Relação
Visibilidade do estado do sistema	Visibilidade	O usuário sempre possa ver suas funções disponíveis no momento conciliando com heurística que estabelece que o sistema deve informar os usuários do estado a cada momento dentro de um tempo razoável.
Correspondência entre o sistema e o mundo real	Familiaridade affordance	O sistema deve comunicar-se com o usuário com a linguagem familiar do mesmo, como por exemplo palavras, frases e conceitos.
Liberdade e controle por parte do usuário	Recuperação	Ambos concordam que é necessário a possibilidade do usuário recuperar-se de erros, em caso de erro ou escolha de funções erradas por engano, deste modo oferecendo uma saída de emergência
Consistência e padrões:	Consistência	Ambas compactuam nas convenções de padrões para que o usuário siga uma lógica, para que o mesmo não venha se perguntar se palavras, situações ou ações significam a mesma coisa porém o princípio ainda afirma que há situações onde a inconsistência é desejada pois a chama atenção
Prevenção de erros	Restrição	Design que leve a evitar falhas ou mensagens de confirmação antes de concluir ações, sendo necessário também assegurar restrições, segundo o princípio.
Reconhecimento preferível memorização	Visibilidade, familiaridade, à <i>affordance</i> , navegação e retorno	Busca minimizar a quantidade de informação que o usuário precisa memorizar, sendo que objetos ações e opções devem ser visíveis e facilmente entendidas permitindo uma navegação fluida eficiente sempre com respostas das ações do usuário.
Flexibilidade e eficiência	Flexibilidade	Mecanismos de eficiência de uso, ações para



de uso			usuários iniciantes e experientes no sistemas, assim agilizando ou facilitando para cada nível de usuário
<i>Design</i> estético e minimalista		<i>Design</i>	Tratam da apresentação do sistema, sendo essa simples, elegante e atraente, sempre na busca de um <i>design</i> minimalista para que não concorra com informações relevantes.
Ajuda aos usuários para reconhecer, diagnosticar e se recuperar dos erros	Retorno e sociabilidade		Tornam o sistema mais compreensível indicando o problema e solução de forma construtiva
Suporte e documentação	Não há		Não há

**Fonte: Autoria própria**

O princípio do controle e a heurística do suporte e documentação são isolados pois são pontos específicos dos autores Benyon e Nielsen, respectivamente o princípio apresenta a importância da comunicação quem ou o que controla o sistema e a heurística diz que é importante a documentação e suporte porém é preferível que o usuário possa realizar as ações sem elas.

### 3 MATERIAL E MÉTODOS

Este capítulo apresenta a base para o entendimento acerca da construção do protótipo, *web service*, o qual foi realizado com base em linguagem de programação JAVA e Android, levando em consideração a utilização dos conceitos do Paradigma de Orientação a Objetos e análises IHC sobre as bibliotecas virtuais.

#### 3.1 IDENTIFICAÇÃO DAS BIBLIOTECAS VIRTUAIS

Por meio da ferramenta de busca Google foi identificada três bibliotecas virtuais, Arquivo Público do Estado de São Paulo (Figura 3), Bibliomania (Figura 4) e Domínio Público (Figura 5). As bibliotecas selecionadas (Figuras 3-5), estão representadas com o acesso a partir de um navegador de Internet para computadores *Desktops*. Essas foram escolhidas, pois além de ser públicas foram citadas em sites de recomendações de bibliotecas virtuais, tais como o site canal do ensino (ENSINO, 2015).

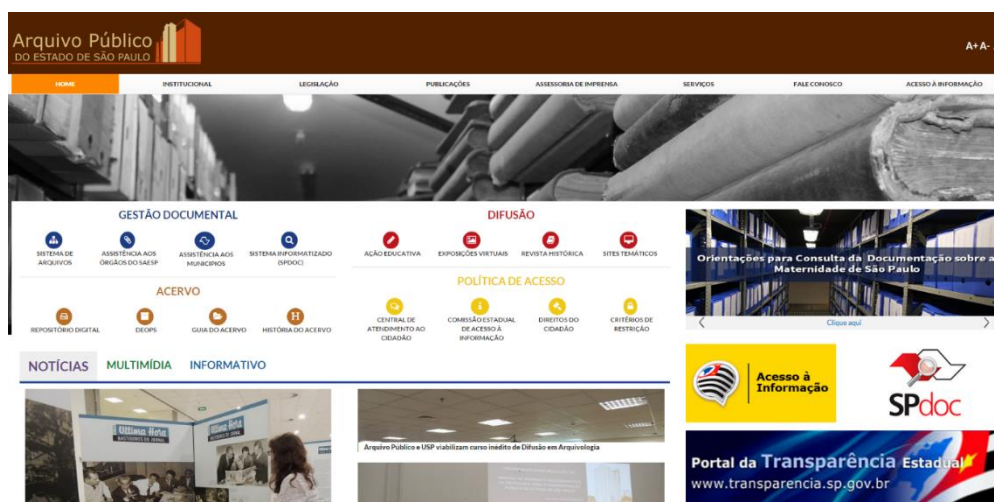


Figura 3 - Página principal da biblioteca virtual Arquivo Público do Estado de São Paulo<sup>1</sup>.  
Fonte: Autoria própria.

<sup>1</sup> Disponível em: <http://www.arquivoestado.sp.gov.br/site/>



Figura 4 – Página principal da biblioteca virtual Bibliomania<sup>2</sup>.  
Fonte: Autoria própria.

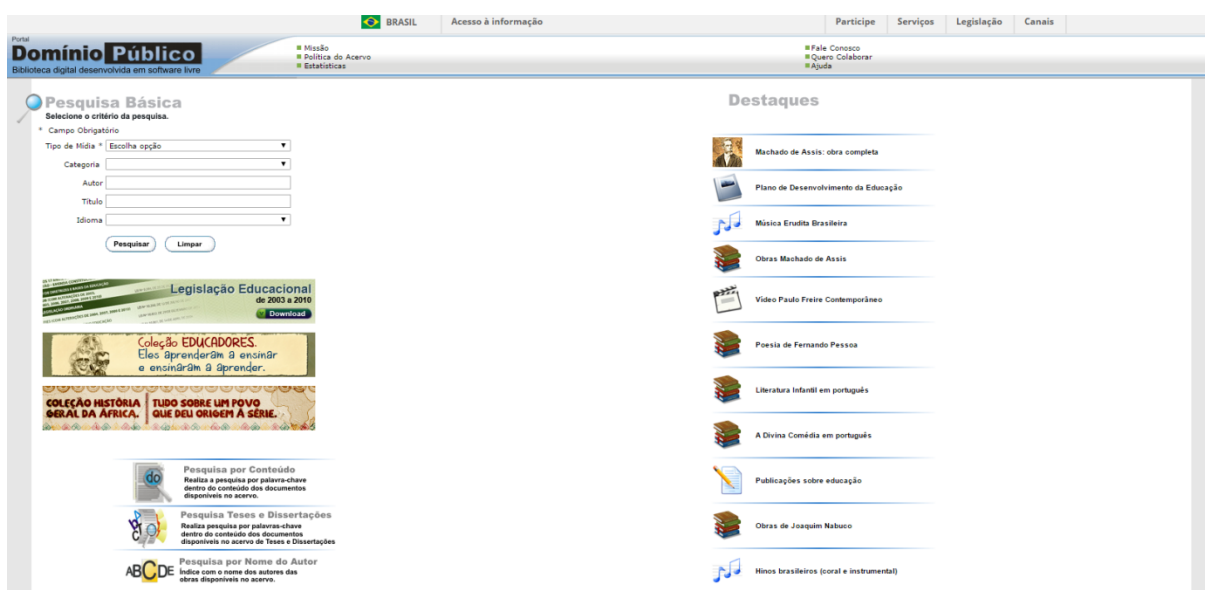
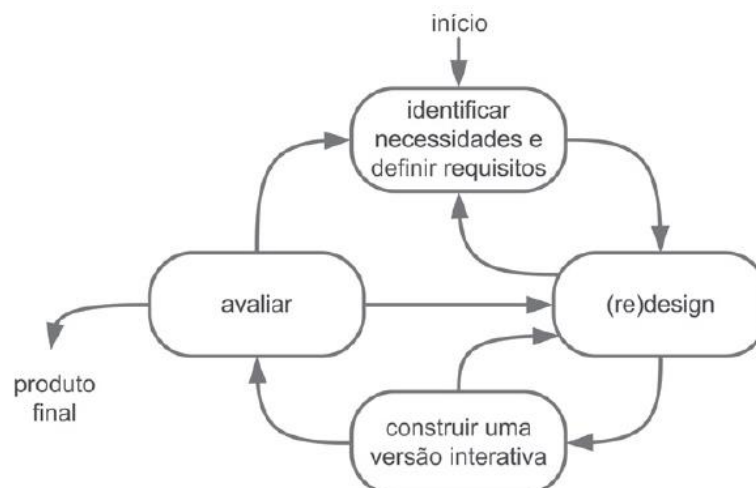


Figura 5 - Página principal da biblioteca virtual Domínio Público<sup>3</sup>.  
Fonte: Autoria própria.

O próximo passo foi a realização do processo de *design* apresentado na Figura 6. Análise da situação atual foi realizada sobre a versão *mobile* das bibliotecas virtuais, assim identificando as necessidades e requisitos a partir da seleção e uso dessas bibliotecas.

<sup>2</sup> Disponível em: <http://www.bibliomania.com/>

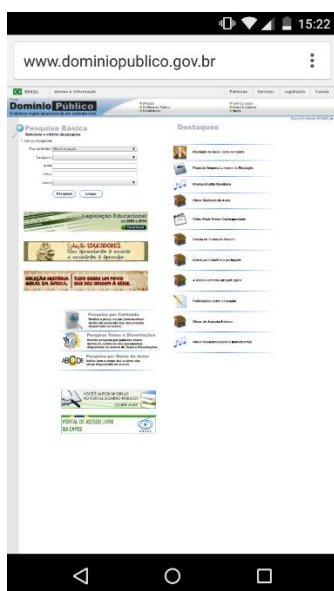
<sup>3</sup> Disponível em: <http://www.dominiopublico.gov.br/pesquisa/PesquisaObraForm.jsp>



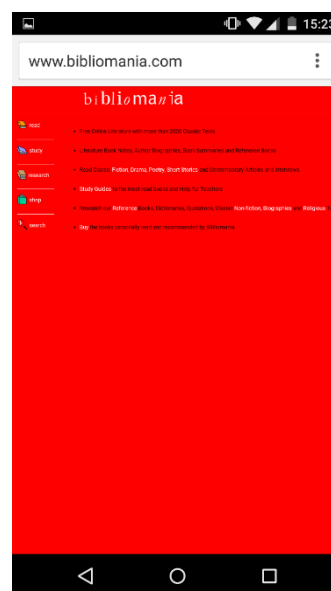
**Figura 6 - Processo de *design* de IHC.**  
**Fonte: (BARBOSA ; SILVA, 2010).**

Para a construção de um design interativo foi necessário um *Web Service* que abastece o aplicativo, assim é apresentado uma sugestão de como desenvolvê-lo.

A versão *mobile*, Domínio Público (Figura 7), Bibliomania (Figura 8) e Arquivo Público do Estado de São Paulo (Figura 9 e Figura 10). Sendo que a intervenção sobre os problemas serão aplicadas no protótipo de aplicativo para bibliotecas virtuais.



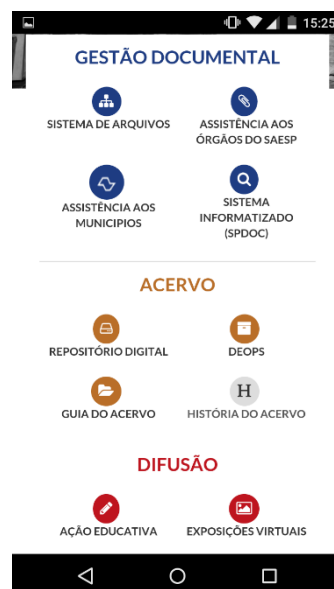
**Figura 7 - Página principal da biblioteca virtual Domínio público acessada a partir de um navegador de dispositivo móvel (smartphone).**  
**Fonte: Autoria própria.**



**Figura 8 - Página principal da biblioteca virtual Bibliomania acessada a partir de um navegador de dispositivo móvel (smartphone)**  
**Fonte: Autoria própria.**



**Figura 9 - Página principal da biblioteca virtual Arquivo Público do Estado de São Paulo acessada a partir de um navegador de dispositivo móvel (smartphone).  
Fonte: Autoria própria.**



**Figura 10 - Página principal da biblioteca virtual Arquivo Público do Estado de São Paulo acessada a partir de um navegador de dispositivo móvel (smartphone).  
Fonte: Autoria própria.**

Deste modo é avaliado as bibliotecas com base nos princípios do *design* e sua relação com a heurística de Nielsen, são utilizados três princípios base sendo Estilo, Visibilidade, Navegação.

A primeira biblioteca avaliada, Domínio Público (Figura 7), obteve os seguintes resultados:

1. Primeira violação Estilo, apesar de ser minimalista a biblioteca contém espaços em brancos onde não tem consistência com a página, deste modo tornando informações relevantes mal posicionadas.
2. Segunda violação Navegabilidade, por possuir um *design* falho, tornou as ações do site desfocadas e legibilidade reduzida assim dificultando a sua navegação.
3. Terceira violação, Visibilidade, por consequência das violações anteriores este princípio também é violado pois as falhas acabam gerando dificuldade em se identificar funções do sistemas e quais seus estados.

Desta três violações dos princípios do *Design* é possível verificar as seguintes violações com relação as heurísticas, Reconhecimento preferível à memorização, Flexibilidade e eficiência de uso, *design*.

A segunda biblioteca avaliada, Bibliomania (Figura 8), obteve os seguintes resultados nos princípios utilizados:

1. Primeira violação, Estilo, uso incorreto de cores para plano de fundo, sendo vermelho uma cor que representa emoções fortes, tornando o *design* da biblioteca malfeito assim tirando o foco de informações relevantes.
2. Segunda violação, Navegabilidade, o site apesar de ter um menu à esquerda trabalha com *links* nos textos, desta forma usuário não identifica que é possível navegar, sem qualquer tipo de resposta do sistema se deve ou não clicar no link.
3. Terceira violação, Visibilidade, por consequência da primeira violação a biblioteca possui baixa legibilidade deste modo dificultando a identificação de ações da biblioteca.

Essa análise infere que as seguintes heurísticas de Nielsen são violadas, Correspondência entre o sistema e mundo real, Visibilidade do estado do sistema e *Design* estético e minimalista.

A terceira biblioteca avaliada, Arquivo Público do Estado de São Paulo (Figuras 9 e 10), apresenta um caso de sucesso pois baseada nas análises dos princípios do *Design* utilizados ela não os viola pois, possui *design* elegante e minimalista, boa navegabilidade entre sistema e visibilidade das ações disponíveis.

Assim considerando as heurísticas de Nielsen com base nos três princípios do design, a biblioteca está em conformidade com os mesmos. A Tabela 1 apresenta um resumo dos princípios do Design violados.

**Tabela 2 - Heurísticas violadas pelas bibliotecas virtuais**

	Estilo	Visibilidade	Navegação
Bibliomania	X	X	X
Dom. Público	X	X	X
Arq. São Paulo			

**Fonte: Autoria própria.**

Assim é apresentado uma lista de requisitos necessária para desenvolvimento do protótipo

Por meio das análises realizadas baseado na Figura 6, o protótipo apresentado deve possuir os seguintes requisitos funcionais:

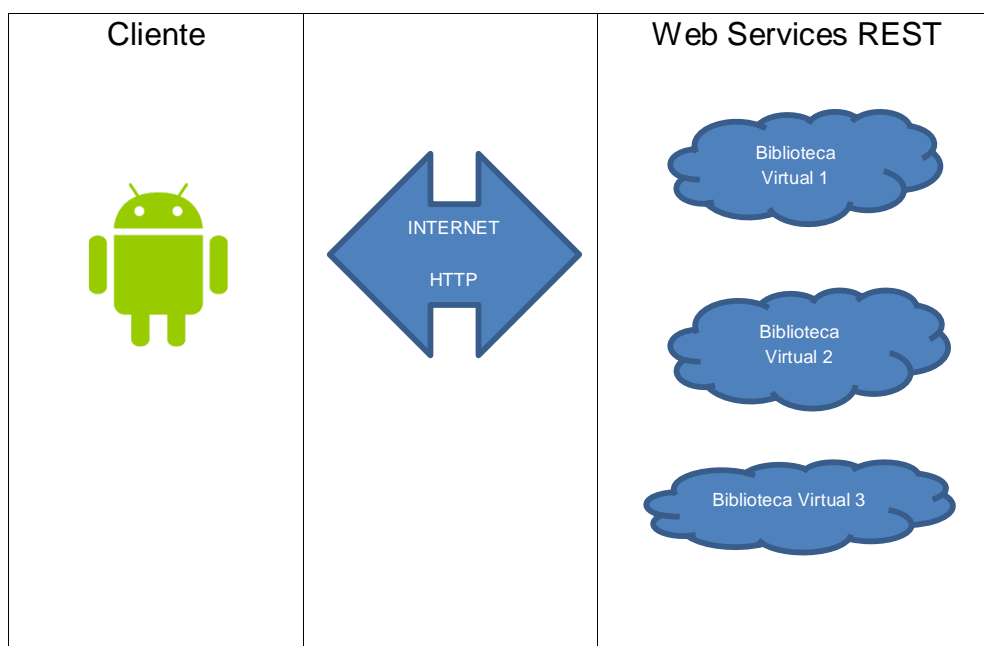
- Listagem das bibliotecas cadastradas;
- Listagem dos documentos cadastrados na biblioteca;
- Apresentar descrição do documento, conforme o JSON recebido;
- Disponibilizar buscas para serem realizadas na biblioteca por título, autor, assunto e palavras chaves;
- Identificar documento por uma imagem.

Simultaneamente os requisitos não funcionais, que se referem a características da aplicação, como usabilidade, *design* são descritos a seguir:

- Cores que estimulem o silêncio e tranquilidade utilizadas adequadas a uma biblioteca, como Azul (SILVA;LEME, 2013);
- Fonte legível.

### 3.2 PROCEDIMENTOS PARA DESENVOLVIMENTO DO APLICATIVO

Para a construção do aplicativo, inicialmente é proposto sua arquitetura, a Figura 11 - Arquitetura do Projeto



**Figura 11 - Arquitetura do Projeto**  
**Fonte: Autoria Própria**

A aplicação é dividida em três Camadas (etapas), como mostra à Figura 11:

- Cliente: é a própria aplicação Android, que transforma os dados consumidos em conteúdo legível ao usuário, ou seja, recebe os dados em formato JSON e os manipula para apresentação;
- Requisições HTTP da aplicação aos *Web Service*, sendo a própria aplicação Android que as realiza;
- *WebService*: são as bibliotecas com os recursos implementados

Para o desenvolvimento do protótipo foi utilizado à linguagem Java, que possui um conjunto de ferramentas para desenvolvimento *mobile* conhecido como Android. O Android também é mantido pela Google que realiza constantes atualizações. A escolha dessa linguagem é devido a ter grande número de usuários e possuir diversas bibliotecas *open-source* para utilizar.

O ambiente utilizado para o desenvolvimento do aplicativo é o *Android Studio*, sendo que essa *Integrated Development Environment* (IDE), é própria ao desenvolvimento Android possuindo diversas ferramentas de apoio ao desenvolvedor, deste modo aumentando a produtividade.

As bibliotecas necessárias para o desenvolvimento do protótipo foram:

- AQuery<sup>4</sup> é uma biblioteca que facilita a programação Android, porém especificamente nesse projeto foi utilizado para fazer requisições Ajax, assim fazendo solicitações HTTP (POST/GET) para envio e aquisição de string em formato JSON;
- GSON<sup>5</sup> é desenvolvido pela Google e tem como objetivo converter Objetos Java em sua representação JSON ou vice-versa. É simples e útil de se utilizar de modo rápido e de fácil manutenção, pois todo objeto é construído por meio da reflexão, sendo necessário construir apenas classes Java de Modelo para fazer essas conversões.

---

<sup>4</sup> Disponível em: <https://code.google.com/p/android-query/>

<sup>5</sup> Disponível em: <https://github.com/google/gson>



## 4 RESULTADOS E DISCUSSÃO

Este capítulo detalha todo o processo de modelagem e desenvolvimento de um protótipo de um repositório de bibliotecas virtuais para dispositivos móveis. O desenvolvimento baseia-se em informações sobre as funcionalidades básicas do sistema e análises das bibliotecas virtuais escolhidas. O objetivo foi a criação de um protótipo, ou seja, que possa ser utilizada para futuros trabalhos. O desenvolvimento teve como ponto de partida a modelagem das interfaces utilizadas no aplicativo. A seguir serão descritas as etapas para desenvolvimento do aplicativo, das análises de *design* das bibliotecas selecionadas e do modelo de *web service*.

### 4.1 PROPOSTA DE WEBSERVICE

As funcionalidades mais importantes para uma biblioteca virtual são suas buscas. Assim é proposta uma interface em Java (Quadro 3), exemplificando as buscas necessárias para este WebService.

```
public interface LibraryEndpoint{  
    public String buscaAutor(String autor);  
    public String buscaTituloDocumento(String titulo);  
    public String buscaPalavrasChaves(String[] palavras);  
}
```

**Quadro 3 - Interface inicial de Web Service.**  
**Fonte: Autoria própria.**

Essa interface é uma exemplificação em JAVA, sendo que a consistência entre as implementações deve ser o nome dos métodos, estes seriam replicados como *endpoint* da aplicação REST. A implementação das buscas não é relevante para o modelo, pois cada biblioteca pode definir como será feito, exemplo, quantos resultados retornará.

Enquanto o retorno *string* é importante, pois é a resposta do que o usuário solicitou, sendo essa resposta um JSON. No Quadro 4, é apresentado o modelo JSON de como é a resposta que é enviada ao usuário.

Essa resposta é convertida pelo aplicativo para apresentar de forma amigável na plataforma *mobile*.

```
{
  "autor": "Author",
  "titulo": "Titulo",
  "palavrasChaves": "Computação, Programação",
  "assunto": "Assunto",
  "Resumo": "Resumo",
  "linkDocumentos": [{"URL": "link"}, {"URL": "link"}],
  "linksReferencias": [{"URL": "link"}, {"URL": "link"}]}
}
```

**Quadro 4 - Modelo JSON.**  
**Fonte: Autoria própria.**

## 4.2 DESENVOLVIMENTO DO PROTÓTIPO

Por meio das análises foram elaborados os requisitos e que destes foi desenvolvido o protótipo que baseado nos princípios do Design avaliados, Design, Visibilidade e Navegabilidade.

Primeiramente para o desenvolvimento da interface foi utilizado a ferramenta *Android Studio* esta permite a criação de alta fidelidade da interface, além de uma pré-visualização também permite simular em um emulador Android a interface.

A primeira tela, Figura 14, possui a tela inicial, está tendo as duas funcionalidade básicas que são busca de documento e bibliotecas respectivamente pelos ícones de Lupa e Lista nos cantos superiores.

As telas das Figuras 12 e 13 (busca de bibliotecas), com um *design* minimalista apresenta somente informações relevantes, como últimas bibliotecas acessadas e uma ação para buscar bibliotecas.

A tela da Figura 15 (busca de documentos) apresenta somente informações relevantes para realizar a busca.

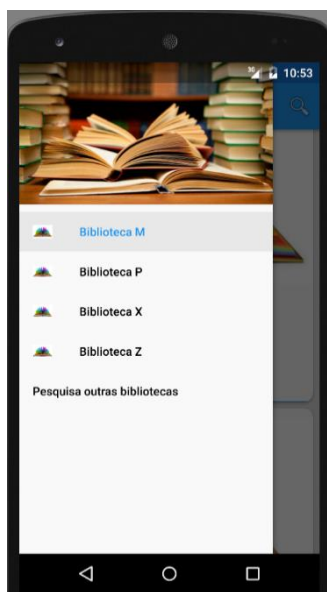
**Tabela 3 - Análise de heurísticas da interface**

Figura	Heurística	Justificativa
12	Design estético e minimalista	Apenas informações necessária, para buscar e acessar bibliotecas virtuais
13	Design estético e minimalista Correspondência entre o sistema e o mundo real	Apresenta no formato de lista, assim apresenta somente informações necessárias para selecionar bibliotecas virtuais.
14	Reconhecimento preferível a memorização, Correspondência entre o sistema e o mundo real, Consistência	Os ícones representam ações, busca de documentos e bibliotecas, que faz relação entre o mundo real e o sistema, tornando-se consistente.
15	Design estético e minimalista	Somente com Informações relevantes apresenta o diálogo com as informações necessárias

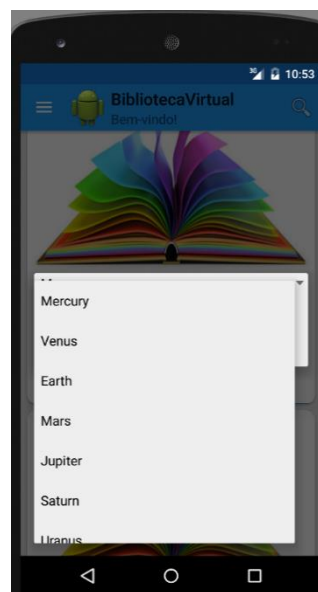
**Fonte: Autoria própria.**

Assim o aplicativo consegue obedecer as seguintes heurísticas de Nielsen: Reconhecimento preferível a memorização, Correspondência entre o sistema e o mundo real, Design estético e minimalista.

Isso é garantido pois por ter um *design* simples é fácil identificar suas ações pois os ícones possuem correspondência com as ações realizadas.



**Figura 12 - Lista de Bibliotecas.  
Fonte: Autoria própria.**



**Figura 13 - Lista de Bibliotecas.  
Fonte: Autoria própria.**



**Figura 14 - Lista de Documentos**  
Fonte: Autoria própria.



**Figura 15 - Busca na biblioteca**  
Fonte: Autoria própria.

Seguidamente, é construído um modelo JSON em uma classe JAVA para capturar e converter o objeto JSON por meio do GSON, a classe é apresentada no Quadro 5, semelhante ao Quadro 4, o GSON faz uma reflexão por meio do nome das variáveis, ou seja, o nome do objeto JAVA deve ser idêntico ao nome da chave JSON. Com isso o GSON converte uma *string* JSON em um Objeto JAVA.

```
public class JsonModel {
    private String autor;
    private String title;
    private String palavrasChaves;
    private String assunto;
    private String resumo;
    private List<String> linkDocumentos;
    private List<String> linkReferencias;

    public JsonModel() {...}
}
```

**Quadro 5 - Objeto Java pelo GSON.**  
Fonte: Autoria própria.

Assim utilizando a biblioteca AQuery é utilizada para realizar a comunicação com o *WebService REST*, implementado pelas bibliotecas, pois cada biblioteca, disponibilizará as URL de seus serviços, onde a aplicação fará uma requisição POST com os parâmetros necessários e terá como resultado a *string* JSON.

No Quadro 6 é exemplificado como é realizado esse procedimento utilizando a biblioteca AQuery.

```
public class HttpPost {
    AQuery aq;
    String url;
    JsonModel modelo;
    private JsonModel MetodoPost(){
        try {
            JSONObject input = new JSONObject();
            input.putOpt("parametro", "valor");
            aq.post(url, input, String.class, callback(url, result, status) → {
                if(result != null){
                    Gson gson = new Gson();
                    modelo = gson.fromJson(result, JsonModel.class);
                }
            });
        } catch (JSONException e) {
            e.printStackTrace();
        }
        if(modelo != null){
            return modelo;
        }else{
            return new JsonModel();
        }
    }
}
```

Quadro 6 - Processo de requisição ao *WebService*.  
Fonte: Autoria própria.

Assim, tendo uma consistência entre as URL foi possível ampliar à abrangência das bibliotecas apenas alterando parte da URL, por exemplo, <http://someurl/method>, assim somente a parte em negrito é alterada, permitindo que em futuras bibliotecas sejam adicionadas.

## 5 CONSIDERAÇÕES FINAIS

### 5.1 CONCLUSÕES

As avaliações mostraram que falta uma manutenção ou reformulação das IHC das bibliotecas virtuais, sendo realizada uma avaliação na situação atual, deste modo é apresentado uma síntese para o problema que é o protótipo que dispõe dos princípios do *design*.

O protótipo propõe uma maneira mais centralizada e padronizada de dados, utilizando *web service* e JSON, conseqüentemente permitindo a adesão de novas bibliotecas de um modo mais amigável e rápido, sendo o protótipo uma possível solução ao problema que as diferentes bibliotecas virtuais experimentam, como problemas de IHC e os usuários necessitarem aprenderem diversos ambientes para obterem o mesmo resultado.

A avaliação heurística unida aos princípios do *design* permitiu a construção de uma interface para o produto que não violasse os princípios do *design* e as heurísticas de Nielsen avaliadas. Essas heurísticas permitem com maior facilidade identificar problemas de IHC em *web sites* e aplicativos, por isso são importantes no desenvolvimento.

Para comunicação entre serviços de diferentes plataformas utilizou os dados em formato JSON devido sua facilidade de leitura unido as tecnologias como o GSON, desta forma resulta em um rápido desenvolvimento.

O trabalho e protótipo desenvolvidos são importantes pois mesmo que todos os sites fossem totalmente responsivos e válidos perante aos princípios do *design*, o usuário que precisasse realizar uma busca em diversas bibliotecas ainda teria o obstáculo de aprender o ambiente de cada biblioteca. O protótipo sugerido resolveria tanto os problemas de responsividade e a necessidade de aprendizado de diversas bibliotecas virtuais.

## 5.2 TRABALHOS FUTUROS

O trabalho fornece uma sugestão de interface para o desenvolvimento de um aplicativo funcional. Sendo assim pode haver a necessidade de uma avaliação sobre o próprio protótipo para o desenvolvimento deste novo produto, tendo melhorias tanto na parte de *Design* ou *Dados*.

O trabalho estabelece um modo de avaliação para transição de ambiente Web ao *mobile*, utilizando as heurísticas de Nielsen e os princípios do *design* como parâmetro para a construção da interface. Assim abrange mais do que as bibliotecas virtuais e sim áreas/produtos com interesse no desenvolvimento de novas interfaces.

## REFERÊNCIAS

BARBOSA, S. D. J.; SILVA, B. S. D. **Interação Humano-Computador**. [S.l.]: [s.n.], 2010.

BASSETT, L. **Introdução ao JSON**. [S.l.]: Novatec, v. I, 2015.

BENYON, D. **Interação Humano-Computador**. 2nd. ed. [S.l.]: [s.n.], 2011.

BROWNING, J. Libraries Without Walls for Books Without Pages, v. 1, p. 62-65, 1993.

ENSINO, C. D. Canal do Ensino. **Canal do Ensino**, 2015. Disponível em: <<http://canaldoensino.com.br/blog/24-bibliotecas-virtuais-que-voce-deveria-conhecer>>. Acesso em: 2015.

JAKL, M. **REST**. University of Technology Vienna. Vienna, p. 24. 2005.

LEVACOV, M. Scielo, 1997. Disponível em: <[http://www.scielo.br/scielo.php?pid=S0100-19651997000200003&script=sci\\_arttext](http://www.scielo.br/scielo.php?pid=S0100-19651997000200003&script=sci_arttext)>. Acesso em: 14 August 2015.

LIMA, J. C. R. **WEB SERVICES (SOAP X REST)**. FACULDADE DE TECNOLOGIA DE SÃO PAULO. SÃO PAULO, p. 41. 2012.

ORACLE. The Java EE 6 Tutorial, 2013. Disponível em: <<https://docs.oracle.com/javase/6/tutorial/doc/gijvh.html>>. Acesso em: 18 August 2015.

PINTO, J.; PORTILHO, G. Revista Escola. **O que são bibliotecas virtuais?**, 2014. Disponível em: <<http://revistaescola.abril.com.br/fundamental-2/sao-bibliotecas-virtuais-681243.shtml>>. Acesso em: 2015.

PRATES, R. O.; BARBOSA, S. D. J. **Avaliação de Interfaces de Usuário – Conceitos e Métodos**. [S.l.]. 2006.

ROSA, J. M.; VERAS, M. **Avaliação heurística de usabilidade em jornais online: estudo de caso em dois sites**. [S.l.]: [s.n.], 2013.

SAUDATE, A. **REST Construa API's inteligentes de maneira simples**. São Paulo: Casa do Código, 2012.

SILVA, A. M. D.; LEME, R. R. O USO DAS CORES NO DESENVOLVIMENTO DE WEBSITES COM FOCO EM USABILIDADE. **FGH Escola de Negócios**, 2013.

UOL. **Venda de smartphones deve superar a de computadores no Brasil em 2011**, 2011. Disponível em: <<http://tecnologia.uol.com.br/ultimas->



noticias/redacao/2011/02/17/venda-de-smartphones-deve-superar-a-de-computadores-no-brasil-em-2011-diz-idc.jhtm>. Acesso em: 2015.

VALIATI, E. R. D. A. **Avaliação de Usabilidade de Técnicas de Visualização de Informações Multidimensionais**. Universidade Federal do Rio Grande do Sul. [S.l.]. 2008.

W3SCHOOLS. W3Schools, 2015. Disponível em:  
<[http://www.w3schools.com/json/json\\_intro.asp](http://www.w3schools.com/json/json_intro.asp)>. Acesso em: 26 August 2015.