

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Danilo Carneiro Ikeda

**Bateria Avaliativa Informatizada com Ênfase  
em Funções Cognitivas na Terceira Idade**

**Uberlândia, Brasil**

**2016**

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Danilo Carneiro Ikeda

**Bateria Avaliativa Informatizada com Ênfase em Funções  
Cognitivas na Terceira Idade**

Trabalho de conclusão de curso apresentado  
à Faculdade de Computação da Universidade  
Federal de Uberlândia, Minas Gerais, como  
requisito exigido parcial à obtenção do grau  
de Bacharel em Ciência da Computação.

Orientador: Professor Fabiano Dorsa

Universidade Federal de Uberlândia – UFU

Faculdade de Ciência da Computação

Bacharelado em Ciência da Computação

Uberlândia, Brasil

2016

Danilo Carneiro Ikeda

## **Bateria Avaliativa Informatizada com Ênfase em Funções Cognitivas na Terceira Idade**

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Ciência da Computação.

Trabalho aprovado. Uberlândia, Brasil, 16 de dezembro de 2016:

---

**Professor Fabiano Dorsa**  
Orientador

---

**Professor**

---

**Professor**

---

**Professor**

Uberlândia, Brasil  
2016

*Dedico esse trabalho a todas as pessoas me apoiaram, incentivaram e acompanharam o processo de desenvolvimento desse trabalho*

# Agradecimentos

Primeiramente gostaria de agradecer a Deus, nada disso seria possível sem sua benção. Agradeço também a Ele por ter me guiado não apenas nesses anos facultativos mas também em todos os momentos de minha vida.

À Universidade Federal de Uberlândia por ter me acolhido e proporcionado uma estrutura educacional que foi e será fundamental do meu crescimento educacional e profissional.

À Universidade Federal do Triângulo Mineiro, junto com a professora e Vice-coordenadora do Programa de Pós-Graduação em Psicologia da UFTM, Sabrina Barroso. Sem seus protocolos e documentos criados, esse projeto não estaria sendo desenvolvido. Não tenho palavras para dizer o quanto estou agradecido.

Aos meus professores que além terem proporcionado uma enorme fonte de conhecimento, contribuíram diretamente na moldura do meu caráter e amadurecimento.

Ao meu orientador e professor, Fabiano Dorça. Talvez seja as características básicas da maioria dos orientadores serem pacientes, dedicados e conselheiros, porém o que o Fabiano me proporcionou através dos seus ensinamentos e de seu caráter, isso foi além de qualquer expectativa que eu tivesse. Estarei eternamente grato por tudo que me forneceu e acima de tudo por sua amizade.

Por fim, gostaria de deixar minha gratidão aos meus familiares e amigos que contribuíram para minha formação acadêmica.

# Resumo

Por fornecer maior rapidez, acessibilidade e organização, o uso de tecnologias tem afetado vários setores. Aplicações web são umas dessas tecnologias, estas permitem que usuários utilize o sistema desenvolvido através de navegadores, sem não necessariamente estar conectado a internet. Com o objetivo de virtualizar testes e avaliações feitos por psicólogos, a faculdade de psicologia da Universidade Federal do Triângulo Mineiro criou um documento composto por uma bateria informatizada de avaliação para idosos contendo atividades que estimulam funções cognitivas e em parceria com a Universidade Federal de Uberlândia foi desenvolvido uma aplicação para web baseado nessa bateria informatizada. Esse trabalho tem como objetivo mostrar as etapas, ferramentas e linguagens utilizadas, métodos e outros para a construção de um sistema web utilizado por psicólogos.

**Palavras-chave:** Aplicação web, Bateria Avaliativa para Idosos, Função Cognitiva **Palavras-chave:**

# Lista de ilustrações

Figura 1 – Exemplo HTML . . . . .	13
Figura 2 – Apresentação no navegador . . . . .	13
Figura 3 – Tag <audio> . . . . .	14
Figura 4 – Tamanho dos Armazenamento de Dados . . . . .	16
Figura 5 – Exemplo 1 - CSS . . . . .	17
Figura 6 – Exemplo 2 - CSS . . . . .	17
Figura 7 – Exemplo - XML . . . . .	18
Figura 8 – Exemplo - XML Schema . . . . .	19
Figura 9 – Exemplo - JSON . . . . .	19
Figura 10 – Fluxo do RuP . . . . .	20
Figura 11 – Fases do RUP . . . . .	21
Figura 12 – XML Schema - Tela2 . . . . .	25
Figura 13 – XML - Tela2 . . . . .	26
Figura 14 – Fluxograma . . . . .	27
Figura 15 – Página de Gerenciamento . . . . .	28
Figura 16 – HTML-Página de Gerenciamento . . . . .	29
Figura 17 – JavaScript-Página de Gerenciamento . . . . .	30
Figura 18 – CSS-Página de Gerenciamento . . . . .	31
Figura 19 – Tela 2 . . . . .	32
Figura 20 – JavaScript - Verificar Paciente . . . . .	32
Figura 21 – JavaScript - Carregar Dados . . . . .	33
Figura 22 – JavaScript - Salvar . . . . .	34

# Lista de abreviaturas e siglas

UFU	Universidade Federal de Uberlândia
UFTM	Universidade Federal do Triângulo Mineiro
XML	eXtensible Markup Language
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
JSON	JavaScript Object Notation
RUP	Rational Unified Process



# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>9</b>
<b>1.1</b>	<b>Considerações Iniciais</b>	<b>9</b>
<b>1.2</b>	<b>Justificativa</b>	<b>9</b>
<b>1.3</b>	<b>Objetivos</b>	<b>10</b>
1.3.1	Objetivos Gerais	10
1.3.2	Objetivos Específicos	10
1.3.2.1	Requisitos Funcionais	10
1.3.2.2	Requisitos Não Funcionais	11
<b>1.4</b>	<b>Metodologia</b>	<b>11</b>
<b>2</b>	<b>REVISÃO BIBLIOGRÁFICA</b>	<b>13</b>
<b>2.1</b>	<b>HTML</b>	<b>13</b>
<b>2.2</b>	<b>HTML5</b>	<b>14</b>
<b>2.3</b>	<b>LocalStorage</b>	<b>14</b>
<b>2.4</b>	<b>SessionStorage</b>	<b>15</b>
<b>2.5</b>	<b>JavaScript</b>	<b>16</b>
<b>2.6</b>	<b>CSS</b>	<b>16</b>
<b>2.7</b>	<b>XML</b>	<b>17</b>
<b>2.8</b>	<b>XML Schema</b>	<b>18</b>
<b>2.9</b>	<b>JSON</b>	<b>19</b>
<b>2.10</b>	<b>RuP (Rational Unified Process)</b>	<b>20</b>
<b>3</b>	<b>DESENVOLVIMENTO</b>	<b>21</b>
<b>3.1</b>	<b>Fase de Concepção / Iniciação</b>	<b>22</b>
3.1.1	Tela de Login	22
3.1.2	Gerenciamento de Pacientes	22
3.1.3	Telas da Avaliação Cognitiva	23
<b>3.2</b>	<b>Fase de Elaboração</b>	<b>23</b>
<b>3.3</b>	<b>Fase de Construção</b>	<b>24</b>
<b>3.4</b>	<b>Fase de Transição</b>	<b>34</b>
<b>4</b>	<b>CONCLUSÃO</b>	<b>36</b>
<b>4.1</b>	<b>Considerações Finais</b>	<b>36</b>
<b>4.2</b>	<b>Trabalhos Futuros</b>	<b>36</b>
	<b>REFERÊNCIAS</b>	<b>38</b>

# 1 Introdução

## 1.1 Considerações Iniciais

Os estudos sobre a aprendizagem vêm relacionando o processo de aprender às capacidades de planejamento e regulação da própria atividade em função de determinados objetivos (Corso, Sperb, Jou, Salles, 2013). Cognitivamente as capacidades de planejamento são coordenadas pelas funções executivas e influenciadas pela memória e a atenção (Yates et al., 2013). Os estudos sobre envelhecimento demonstram que ocorre um declínio significativo em funções como atenção, memória e funções executivas de planejamento mesmo em idosos não-acometidos por doenças (Yassuda Abreu, 2006; Irigaray, Gomes Filho, Schneider, 2012).

O envelhecimento saudável está diretamente ligada ao fato de conseguirmos alcançar a longevidade realizando funções cognitivas. Diversos fatores podem influenciar em atividades que necessitam habilidades cerebrais tais como memória, atenção, raciocínio e atenção, dois exemplos são: doenças cognitivas e envelhecimento. Estudos mostram que ambos podem ser evitados ou postergados. Embora a perda cognitiva seja inevitável, pesquisas sobre intervenções cognitivas demonstram que o treino cognitivo pode ocasionar aumento do desempenho ou permitir a manutenção de habilidades cognitivas em idosos saudáveis (Bavelier, Achtman, Föcker, 2012; Irigaray et al., 2012; Green Bavelier, 2008).

Com o auxílio tecnológico a área da psicologia relacionada a cognição ganhou um forte aliado, a Universidade Federal de Uberlândia (UFU) em parceria com a Universidade Federal do Triângulo Mineiro (UFTM) decidiram criar uma ferramenta virtual na qual, através de uma série de atividades e questionários, o idoso possa treinar suas funções cognitivas e além disso através dos dados gerados por essa ferramenta virtual, o psicólogo poderia dizer se haveria uma possibilidade do idoso saudável vir a desenvolver uma patologia neuropsiquiátrica ou se o paciente estiver com um declínio na qualidade das funções cognitivas causado pela velhice, acima da média. A aplicação web foi desenvolvida pela equipe de Ciência da Computação e Sistema da Informação da UFU, baseado em um documento avaliativo criado pela equipe de Psicologia da UFTM.

## 1.2 Justificativa

Por anos jogos como xadrez onde faz o jogador pensar de certas maneira, tem ajudado no processo de raciocínio e concentração. Talvez a migração dos jogos físicos para o virtual seja apenas uma característica de jogo e o resultado positivos que é o

mais importante, talvez seja o mesmo. Alguns estudos têm mostrado potencial para a utilização de interações eletrônicas com jogos e programas específicos de computador no tratamento de patologias como o Transtorno de Déficit de Atenção e o Autismo (Capovilla & Capovilla, 2007).

A digitalização de testes antes feitos e armazenados em papéis permitem que atividades e exames possam ser feitos com uma maior agilidade e em maior escala, sendo que as informações coletadas sejam guardadas e compartilhadas de maneira que os dados não sejam perdidos. Com o avanço tecnológico é possível que dados sejam apresentados em gráficos e utilizados como estatísticas, o que resulta em uma leitura das informações mais precisas e consequentemente os resultados também. Além disso, vale lembrar que com um número maior de compartilhamento de dados, ou seja, com um maior acesso às informações, permite o aumento do estudo nessa área.

## 1.3 Objetivos

### 1.3.1 Objetivos Gerais

O conceito desse projeto está relacionado à conversão de baterias avaliativas que antes eram feitas em papel e caneta para o mundo virtual.

### 1.3.2 Objetivos Específicos

O objetivo é desenvolver uma ferramenta web na qual auxilie os psicólogos a aplicarem os testes em seus pacientes, e onde um responsável (pesquisador) consiga visualizar o resultado dos questionários (respostas dos pacientes). Realizado a bateria avaliativa espera-se que o avaliador consiga detectar se o paciente está propenso a desenvolver alguma doença neuropsiquiátrica causando disfunção cognitiva ou se o paciente possui um declínio cognitivo acima da média.

Para que o projeto fosse concluído, foi utilizado a linguagem de marcação HTML (HyperText Markup Language) para a construção das páginas web, o CSS (Cascading Style Sheets ) para definir a aparência das páginas e JavaScript para o controle do HTML. A aplicação foi construída de modo offline, onde foi utilizado funções introduzidas no HTML5 como o LocalStorage e o SessionStorage. Quanto ao armazenamento e compartilhamento de dados, foi utilizado o JSON (JavaScript Object Notation) e o XML (Extensible Markup Language).

#### 1.3.2.1 Requisitos Funcionais

- Criação de paciente

- Seleção de paciente
- Remoção de paciente
- Tela de visualização da avaliação
- Avançar para a próxima página
- Voltar página
- Salvar progresso da avaliação
- Inserir nome do psicólogo

#### 1.3.2.2 Requisitos Não Funcionais

- É uma aplicação web
- A aplicação deve ser utilizada apenas em computadores
- Deverá ser aberto para os navegadores: Mozilla Firefox ou Google Chrome
- Os dados gerados serão armazenados no navegador utilizado
- Uma vez utilizado em um navegador, o usuário não poderá abrir o projeto em outro e usufruir das informações já geradas
- Assim que aberto, a aplicação deverá carregar os dados salvos
- Deve ser implementado utilizando as linguagens HTML5, JavaScript e CSS
- O sistema deve ser implementado para uso offline
- A aplicação estará sempre disponível ao usuário
- Para cada sistema, o aplicador deverá atender um paciente por vez, podendo mudá-lo sem necessariamente ter terminado a avaliação
- O Psicólogo não poderá passar para uma próxima atividade se não tiver terminado a atual

## 1.4 Metodologia

O desenvolvimento desta aplicação foi feito a partir de uma bateria avaliativa para idosos, feito pela UFTM. Essa bateria avaliativa foi feita a partir de 4 etapas, levantamento de informações sobre o uso de tecnologias por pessoas acima de 60 anos, identificação de atividades úteis na no treinamento cognitivo, desenvolvimento da bateria referente as

informações coletadas e por fim a parte de teste em idosos saudáveis. É interessante destacar que além das linguagens citadas anteriormente, foi utilizado metodologia de desenvolvimento de software Rational Unified Process (RuP)..

Métodos utilizados:

- Documentação referente à avaliação
- Foi implementado utilizando o modo de desenvolvimento iterativo e incremental, RuP.
- Foi realizado reuniões presenciais e via e-mail com o coordenador da Faculdade de computação da UFU, Fabiano Dorça
- Foi realizado trocas de informações via e-mail com a Vice-coordenadora do Programa de Pós-Graduação em Psicologia da UFTM, Sabrina Barroso
- Foi feito reuniões com os desenvolvedores do projeto afim de definir a estrutura do projeto
- Foi utilizado as linguagens computacionais: HTML, JavaScript, CSS, XML, XML Schema

## 2 Revisão Bibliográfica

### 2.1 HTML

HTML (Hypertext Markup Language) é uma linguagem de marcação utilizada na estruturação e construção de páginas na Web, onde o navegador é responsável de interpretar o arquivo html e apresentar ao usuário os elementos definidos no arquivo.

Para que um navegador possa ler um arquivo HTML, a linguagem faz o uso de tags, que são elementos na qual definem o tipo e o conteúdo das informações (Exemplo: botões, títulos, áudios e imagens). Segue abaixo o exemplo de uma estrutura de um arquivo html e sua apresentação em um navegador:

```
<!DOCTYPE html>
<html lang="pt-br">
  <head>
    <title>Projeto TCC</title>
  </head>
  <body>
    <h1> Bateria Avaliativa para Idoso </h1>
    <p> conteúdo avaliativo </p>
  </body>
</html>
```

Figura 1 – Exemplo HTML



Figura 2 – Apresentação no navegador

Como podemos ver a estrutura básica de um arquivo HTML é dada pelas seguintes tags: <head> e <body>. Segundo o website “www.w3school.com” na tag head é definido tudo o que não fará parte do conteúdo do documento, tal como:

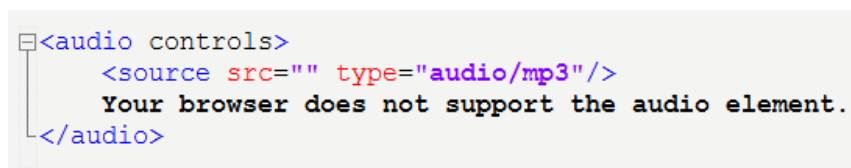
- <title>: Título da pagina

- <Style>: Definição do design da página
- <Link>: Definição para a importação de arquivos externos
- <Meta>: Essa tag contém metadados, ou seja informações que não serão exibidos. O principal objetivo dessa tag é o auxílio nos mecanismos de buscas
- Essa tag é utilizada para definir um script, substituindo o uso de arquivos externos

Já a tag body será composto de todos os componentes que serão exibidos no navegador. Alguns exemplos de tags utilizados na body são: <audio>, <p>, <br>, <button>, <input>, <h1>, assim como outros.

## 2.2 HTML5

O HTML5, apresentado em outubro de 2014, é a quinta versão da linguagem de estruturação e apresentação, este surgiu a partir de uma parceria entre a W3C e a WHATWG (Web Hypertext Application Technology Working Group). Segundo o website Devmedia.com, o html5 trouxe melhorias para a estruturação dos arquivos html com um maior número de componentes e funções. Um dos componentes implementado nessa nova versão do HTML é a tag <audio>, antes da implementação desse elemento áudios eram inseridos através da instalação de plugins.



```
<audio controls>
  <source src="" type="audio/mp3"/>
  Your browser does not support the audio element.
</audio>
```

Figura 3 – Tag <audio>

Porém o grande recurso que foi implementado no HTML5 e que foi o mais utilizado nessa aplicação foi o uso da API Storage. Essa novidade permite que navegadores armazenem informações no próprio navegador, com isso pode-se realizar armazenamento e troca de dados sem o uso de cookies, comumente utilizado nas versões anteriores ao HTML5, ou qualquer tipo de servidor. A API Storage do HTML5 é composto por dois objetos: localStorage e sessionStorage.

## 2.3 LocalStorage

O localStorage é responsável pelo armazenamento de dados no navegador, sendo que esses dados não podem ser transferidos de forma direta entre navegadores pois são

armazenados localmente na memória do navegador. A característica que define o `localStorage` é que ele permite que esses dados sejam armazenados sem data de validade, ou seja, caso o usuário feche o navegador, mude de página Web ou estes fiquem guardados por um longo período de tempo, estes continuarão armazenados. Os dados guardados possuem uma estrutura `<Key, Value>`, isto é, estes são identificados por uma chave única e seus dados são armazenados em formato de String.

Alguns dos métodos do objeto `localStorage` são:

- `localStorage.setItem(key, value)`: permite a inserção de uma chave e valor novos caso a chave não exista ou o valor será substituído caso a chave exista
- `localStorage.getItem(key)`: retornará o valor referente à chave
- `localStorage.key(value)`: retornará a chave referente ao valor
- `localStorage.removeItem(key)`: removerá apenas a chave e o conteúdo referente à esta
- `localStorage.clear()`: excluirá todas as chaves, assim como seus conteúdos

Embora o `localStorage` seja uma ótima ferramenta, esta possui uma limitação no armazenamento de dados. A quantidade de dados armazenados varia de acordo com navegador e plataforma, sendo que em computadores os navegadores podem guardar um tamanho de pelo menos 5mb. Navegadores como Google Chrome, Mozilla Firefox e Internet Explorer podem armazenar uma quantidade máxima de 10mb.

## 2.4 SessionStorage

O `sessionStorage` é bastante similar ao `localStorage`. O que os diferem é fato do `sessionStorage` ser armazenado para apenas uma sessão, isto é, caso o usuário feche o navegador ou mude de página, os dados destes serão removidos. Os métodos atribuídos ao `sessionStorage` são os mesmos pertencentes ao `localStorage`, porém a quantidade de dados armazenados são diferentes para alguns navegadores. O navegador Safari permite um número ilimitado de dados, enquanto o Google Chrome, Mozilla Firefox e Internet Explorer obedecem a mesma política do `localStorage`.



	Chrome	Firefox	Safari	Safari	IE	IE
	40	34	6, 7	8	9	10, 11
<b>Application Cache</b>	up to quota	500MB, Unlimited	Unlimited?	Unlimited?		100MB?
<b>FileSystem</b>	up to quota					
<b>IndexedDB</b>	up to quota	50MB, Unlimited		up to quota?		10MB, 250MB (~999MB)
<b>WebSQL</b>	up to quota		5MB, 10MB, 50MB, 100MB, 500MB, 600MB, 700MB...	5MB, 10MB, 50MB, 100MB, 500MB, 600MB, 700MB...		
<b>LocalStorage</b>	10MB	10MB	5MB	5MB	10MB	10MB
<b>SessionStorage</b>	10MB		Unlimited	Unlimited	10MB	10MB

Figura 4 – Tamanho dos Armazenamento de Dados

Fonte: <<https://www.html5rocks.com/en/tutorials/offline/quota-research/>>

## 2.5 JavaScript

O JavaScript é uma linguagem orientada a objetos, ou seja, ela é capaz de controlar os objetos (elementos definidos em um arquivo HTML) de uma página, sendo que sua utilização é feita localmente o que permite uma maior manipulação no controle e acesso de informações enviadas e recebidas. Essa linguagem tem como objetivo dar funcionalidades a uma página Web, podendo controlar o objetos como por exemplo o clique de um botão, ou manipular informações de determinadas tags.

## 2.6 CSS

O CSS (Cascading Style Sheets) é uma linguagem de folhas de estilos onde através de recursos como background-color, text-align e font-size pode-se definir como os elementos de uma página HTML serão apresentados aos usuários.

Com o CSS é possível definir cores, posição, transparência, formato, fonte e entre outras características. Abaixo podemos ver o exemplo de um botão onde foi definido sua altura, largura, cor do texto, cor de fundo e borda.

```
button
{
    height: 60px;
    width: 200px;
    color: f78f38;
    background-color: 37474F;
    border-radius: 25px;
```

```
}
```

Esse arquivo css aplicado a um html, tem-se o seguinte resultado:

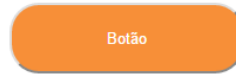


Figura 5 – Exemplo 1 - CSS

O CSS reconhece o componente por sua tag, no caso acima se não houver nenhuma restrição no arquivo HTML, todos os componentes do tipo button serão alterados de acordo com o componente button descrito no arquivo CSS. Tem-se a opção também de definir qualquer nome para a tag no CSS e assim define-se essa tag como uma class no html, assim como descrito abaixo.

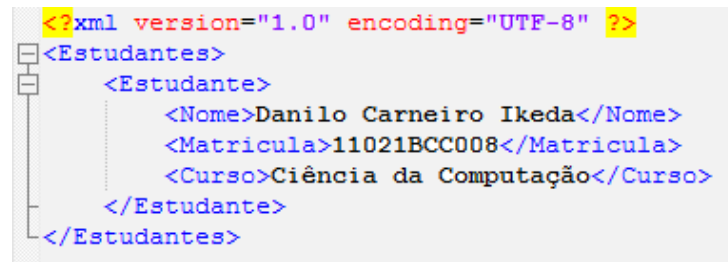
```
<!DOCTYPE html>
<html>
<head>
<style>
.mudarBotao
{
    height: 60px;
    width: 200px;
    color: #ffffff;
    background-color: #f78f38;
    border-radius: 25px;
}
</style>
</head>
<body>
<button class="mudarBotao">Botão</button>
</body>
</html>
```



Figura 6 – Exemplo 2 - CSS

## 2.7 XML

O eXtended Markup Language, ou XML é uma linguagem de marcação assim como o HTML, porém ao contrário do HTML que é utilizado para definir elementos, o XML é utilizado para definir dados, ou seja, são criadas tags nas quais essas não são pré-definidas e são inseridas informações a estas tags, com isso facilita o processo de compartilhamento e validação de dados já que o arquivo estaria organizado por marcações. Por exemplo:

The image shows a snippet of XML code with a corresponding tree diagram on the left. The XML code is: 

```
<?xml version="1.0" encoding="UTF-8" ?>
<Estudantes>
  <Estudante>
    <Nome>Danilo Carneiro Ikeda</Nome>
    <Matricula>11021BCC008</Matricula>
    <Curso>Ciência da Computação</Curso>
  </Estudante>
</Estudantes>
```

The tree diagram on the left illustrates the hierarchical structure: a root node 'Estudantes' contains one child node 'Estudante', which in turn contains three child nodes: 'Nome', 'Matricula', and 'Curso'.

Figura 7 – Exemplo - XML

“XML é uma tecnologia simples que tem ao seu redor outras tecnologias que a complementam e a fazem muito maior e com possibilidades muito mais amplas. XML representa uma maneira distinta de fazer as coisas, cuja principal novidade consiste em permitir compartilhar. XML também permite ao programador e aos suportes dedicar seus esforços às tarefas importantes quando trabalha com os dados, já que algumas tarefas trabalhosas como a validação destes ou o percorrido das estruturas corre a cargo da linguagem e está especificado pelo padrão.” ([www.significados.com.br/xml/](http://www.significados.com.br/xml/))

## 2.8 XML Schema

O XML Schema nada mais é que uma estrutura para um arquivo XML, sua utilização define quais elementos o documento XML terá, definindo também características dos atributos deste documento. Segundo o website a W3School.com, é possível definir:

- quais elementos e atributos aparecerão no arquivo XML
- a quantidade de filhos cada elemento terá
- os tipos dos atributos
- valores padrão ou fixo de cada atributo

Além de ser utilizado com um validador, o XML Schema é uma ótima ferramenta quando há mais que um desenvolvedor e um projeto, pois com ele define-se como um arquivo XML deverá ser gerado.

```
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Estudantes">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Estudante">
          <xs:complexType>
            <xs:sequence>
              <xs:attribute type="xs:string" name="Nome"/>
              <xs:attribute type="xs:string" name="Matricula"/>
              <xs:attribute type="xs:string" name="Curso"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Figura 8 – Exemplo - XML Schema

Um XML Schema é definido por elementos e atributos, os mais comumente utilizados e que são utilizados no exemplo acima são:

- `complexType`: como o próprio nome o define, é um tipo complexo. Ou seja, é um elemento na qual possui outros elementos e/ou atributos
- `sequence`: define que os elementos e/ou atributos de um arquivo XML devem aparecer em uma sequência
- `attribute`: é um elemento simples na qual não possui elementos ou atributos filhos

## 2.9 JSON

JavaScript Object Notation ou simplesmente JSON é um modelo de armazenamento e compartilhamento de dados em formato de texto. Arquivo JSON são similares aos arquivos XML, pois ambos possuem a característica de definirem dados, porém o uso do JSON é mais comumente utilizado no JavaScript pois este possui funções que auxiliam na leitura e manipulação do arquivo.

Comparado ao XML, o JSON é modelo mais leve portanto mais compacto, o que em algumas situações torna-se mais rápido a transferência de dados e mais vantajoso o armazenamento destes.

```
{
  "Estudantes": {
    "Estudante": {
      "Nome": "Danilo Carneiro Ikeda",
      "Matricula": "11021BCC008",
      "Curso": "Ciência da Computação"
    }
  }
}
```

Figura 9 – Exemplo - JSON

## 2.10 RuP (Rational Unified Process)

O Processo Unificado da Rational ou simplesmente RuP é um método desenvolvimento de software iterativo e incremental, geralmente o RuP é utilizado para software orientado à objetos.

De acordo com Larman (LARMAN,2007) o processo unificado possui quatro fases. A primeira etapa ou chamada fase Concepção será criado uma visão aproximada do escopo do projeto, tais como os requisitos e objetivos. A segunda etapa ou fase Elaboração, será definido o escopo e os requisitos do projeto. A fase de Construção é a etapa na qual será modelado o projeto, ou seja, é a parte na qual o projeto será desenvolvido utilizando as ferramentas descritas nas etapas anteriores. Por fim, a fase Transição, onde além de se realizar testes, também dá-se a entrega do projeto para o usuário final e a eles farão teste para o projeto beta.

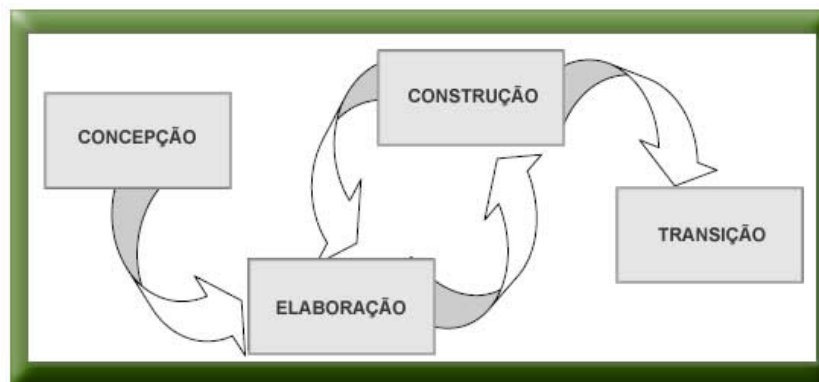


Figura 10 – Fluxo do RuP

Fonte: <<http://www.infoescola.com/engenharia-de-software/rup/>>

### 3 Desenvolvimento

Nesse tópico será exposto passos importante para o desenvolvimento do protocolo de avaliação cognitivo para idosos. Será apresentado a evolução do sistema, mostrando além as linguagens de programação pré-definidas na fase de análise do projeto, as alterações que o projeto sofreu e as tecnologias que foram utilizadas para que este pudesse ser concluído. O tópico será dividido em 4 etapas, de acordo com a metodologia de desenvolvimento RUP. “O RUP organiza o desenvolvimento de software em quatro fases, onde são tratadas questões sobre planejamento, levantamento de requisitos, análise, implementação, teste e implantação do software.” (<http://www.infoescola.com/engenharia-de-software/rup/>)

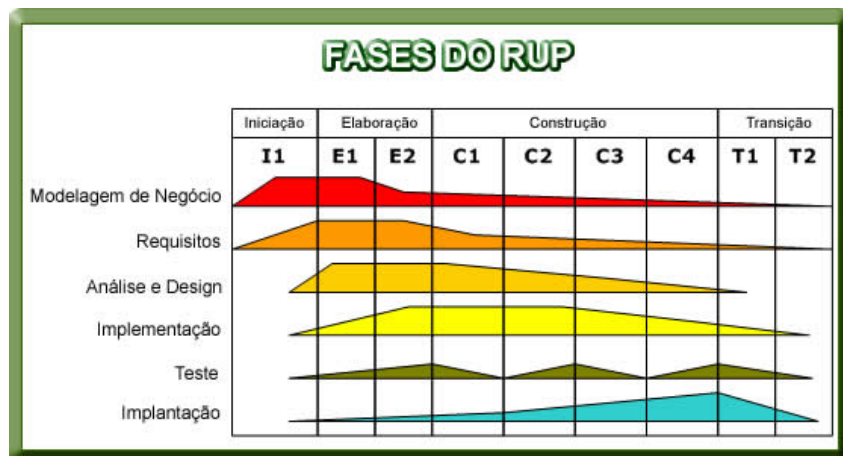


Figura 11 – Fases do RUP

Como já dito anteriormente, este projeto surgiu a partir da ideia de passar para o mundo virtual a bateria de exercícios elaborada pela equipe da UFMT. Pela extensão da aplicação e diferentes partes, o sistema foi dividido em 3 projetos, onde três alunos da UFU trabalharam nele.

- Projeto 1: tela de login, gerenciamento de pacientes e parte das telas dos testes
- Projeto 2: parte metade das telas, recolhimento das informações geradas e envio dos dados para o pessoal da UFTM
- Projeto 3: sistema de visualização por parte da equipe da UFTM

Este documento é referente ao projeto 1.

## 3.1 Fase de Concepção / Iniciação

Na Fase de Concepção da-se o processo de levantamento e planejamento de informações entre o cliente e o desenvolvedor. Uma parte da etapa foi feita pela equipe da UFTM onde se foi definido quem deverá interagir com o sistema e foi feito estudos pela equipe do curso de psicologia da UFTM a fim de elaborar a bateria avaliativa para idosos. A UFU ficou encarregada de atribuir o projeto criado para alunos relacionado a Computação, tendo como orientador o coordenador da faculdade de computação da UFU, Fabiano Dorça. Após decidido, foi levantado conceitos básicos que iriam compor o sistema.

### 3.1.1 Tela de Login

Inicialmente o usuário iria para uma tela de login, na qual cada usuário (psicólogo) teria uma chave de registro armazenada em um servidor de banco de dados, garantindo que apenas psicólogos autorizados pudessem utilizar o sistema (veremos na próxima fase que esta parte foi removida devido a uma alteração no projeto). Para que fosse cadastrado novos usuários era necessário que o psicólogo enviasse um e-mail para a equipe da UFTM requisitando um registro.

A tela de login seria composta pelos seguintes campos:

- Nome
- E-mail
- CRP
- Código de Acesso

O CRP (Conselho Regional de Psicologia) é um registro único do profissional da psicologia. Este número possui de 4 a 11 dígitos. O Código de Acesso seria uma chave de registro criada pela equipe de suporte ao sistema.

### 3.1.2 Gerenciamento de Pacientes

Feito o login, o usuário seria redirecionado para um menu, onde há uma lista de pacientes já criados pelo próprio usuário, tendo a opção de escolher, adicionar ou remover um dos pacientes. Caso o avaliador escolhesse um paciente já registrado, o sistema deve recuperar informações sobre qual tela o avaliando parou e deverá preencher os campos na qual já tivessem sido preenchidos anteriormente. Além disso será na página de Gerenciamento de Pacientes que serão apresentados os logos das universidades

participantes do projeto e também um botão contendo informações sobre os participantes do projeto.

### 3.1.3 Telas da Avaliação Cognitiva

Passado as telas iniciais, começa a parte da bateria de avaliações, essa parte é composta por 109 telas ao total e 55 telas para este projeto, onde cada tela teria um exercício diferente com as opções de avançar, voltar e salvar/sair da tela e o componente de cada tela foi definida por um documento enviado pela vice-coordenadora da faculdade de psicologia da UFTM, Sabrina Barroso.

A função “avançar” só pode ser executada se todos os exercícios referentes a tela atual foram preenchidos e que tenha um paciente selecionado. Após alcançados os requisitos, as informações que foram preenchidas deverão ser armazenadas e assim serão recuperadas quando a página for acessada novamente utilizando o respectivo paciente.

A função “voltar” irá redirecionar o usuário para tela anterior, podendo esta ser executada a qualquer momento. Porém não pode ser perdido o que já foi feito e o sistema deverá recuperar os dados da página na qual o usuário foi redirecionado.

A função “salvar/sair” irá salvar as informações geradas da tela onde o usuário se encontra e ir para a tela de gerenciamento de paciente.

## 3.2 Fase de Elaboração

Nessa fase é analisado de uma maneira mais detalhada os requisitos, riscos e as possíveis mudanças do projeto. Neste momento se é estudado e apresentado as ferramentas e métodos que serão utilizados para o desenvolvimento do sistema.

Inicialmente na etapa de elaboração do sistema de avaliação para idosos, decidimos que o projeto seria composto, além das linguagens JavaScript, HTML5 e CSS, por um servidor de banco de dados na qual seriam armazenados os dados dos pacientes e das atividades sendo que as informações seriam divididas por pacientes e este por psicólogos. Também cada psicólogo teria um registro, tendo assim uma restrição no acesso ao sistema. Porém houve uma mudança de planos, pois foi decidido que o sistema fosse uma aplicação simples e offline, ou seja, não usufrísse de um servidor e banco de dados. Portanto foram marcadas algumas reuniões onde foi discutido possibilidades para que isso fosse possível.

A primeira solução seria o uso de um arquivo XML contendo os dados dos pacientes. O documento ficaria armazenado localmente e todas as vezes que usuário abrisse a aplicação em um navegador, este teria que localizar a pasta de destino do arquivo XML e fazer o upload do arquivo na página correspondente. Como pode ser visto, essa é uma solução inviável uma vez que é trabalhosa para o usuário pois teria que identificar o local



do documento XML e fazer o upload a todo o momento, e também não seria viável pelo risco de adulteração ou exclusão dos dados.

Logo após foi levantado a possibilidade do uso dos componentes `localStorage` e `sessionStorage` para a manipulação e armazenamento dos dados. Como dito anteriormente, o `localStorage` e `sessionStorage` são objetos que armazenam dados no navegador utilizado, portanto seus dados ficariam vulneráveis para qualquer pessoa que utilizá-lo. Com isso o sistema de login tornou-se inutilizável pois os dados que permitiriam o acesso ao sistema, estariam expostos. Portanto foi decidido que a tela de login fosse excluída do projeto.

Decidimos também que pela extensão da criação do sistema de avaliação para idosos e o sistema de visualização e controle por parte da equipe da UFTM, estes fossem divididos em 3 projetos. Onde ao final de cada um, seria realizado uma reunião para que os projetos fossem integrados e testados.

### 3.3 Fase de Construção

“A fusão de vários artefatos de software ocorre neste momento, possibilitando que o sistema seja implementado quase que completamente. Tem-se uma visão geral de como o Baseline do projeto está sendo seguido. No final dessa fase, o sistema deve estar totalmente preparado para a transição ao usuário.” ([www.devmedia.com.br/artigo-engenharia-de-software-o-processo-unificado-integrado-ao-desenvolvimento-web](http://www.devmedia.com.br/artigo-engenharia-de-software-o-processo-unificado-integrado-ao-desenvolvimento-web))

Como os três projetos irão compartilhar dos mesmos dados, inicialmente definimos um XML Schema, onde o armazenamento e validação dos arquivos XML teriam que obedecer a estrutura deste XML Schema. Segue abaixo uma imagem de como é esse XML Schema, sendo mostrado apenas a Tela2 (primeira página da avaliação do idoso):

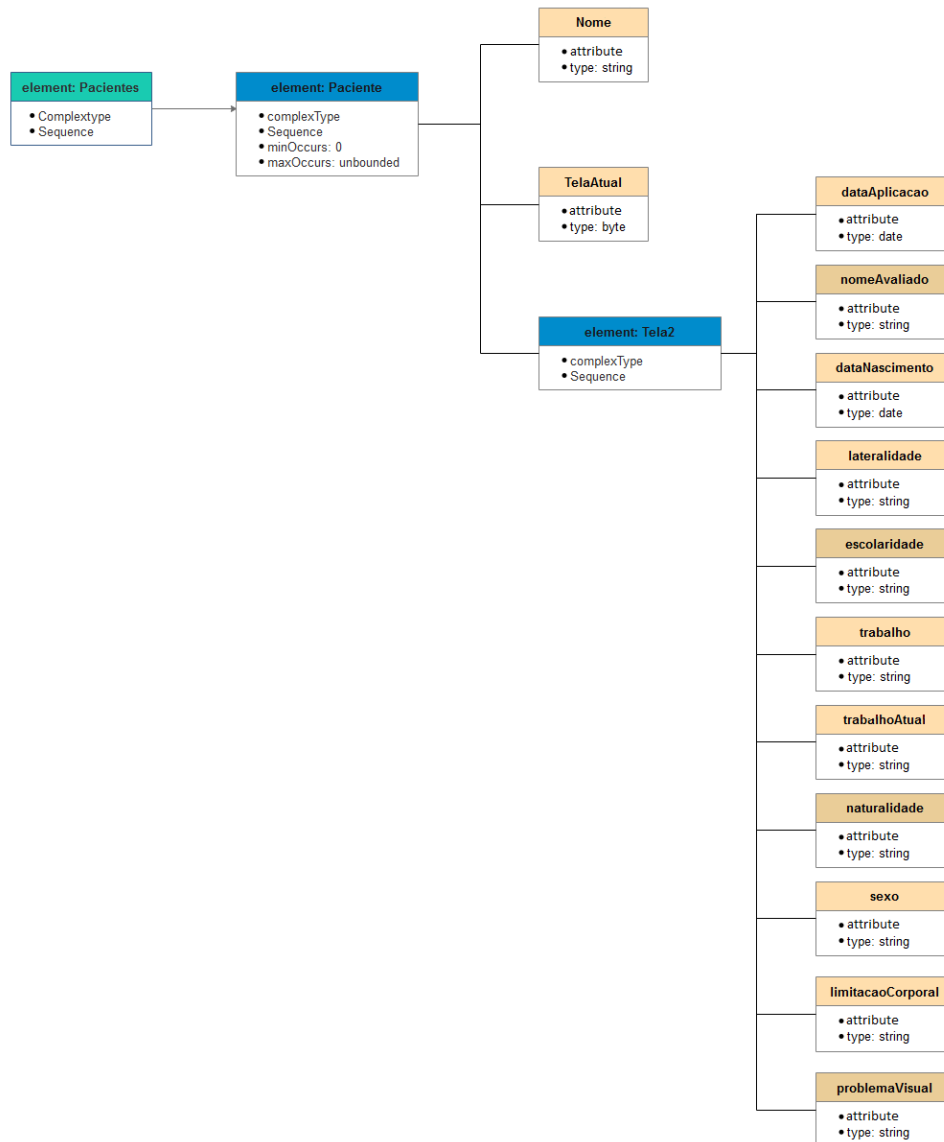


Figura 12 – XML Schema - Tela2

Como o sistema permite a criação de vários pacientes (ou seja vários elementos e atributos) então o elemento "Pacientes" é um complexType. Da mesma forma o elemento "Paciente", na qual é um elemento atribuído a apenas um paciente, será um complexType pois terá várias informações sobre o paciente. Na figura 12, as caixas em amarelo são os atributos, pois terão informações dadas a estes e também estas caixas não irão possuir elementos ou atributos filhos.

Com essa estrutura foi montado no decorrer do processo avaliativo da bateria, o arquivo XML na qual será armazenado e compartilhado. A partir do XML Schema descrito na imagem acima, foi criado o seguinte XML:

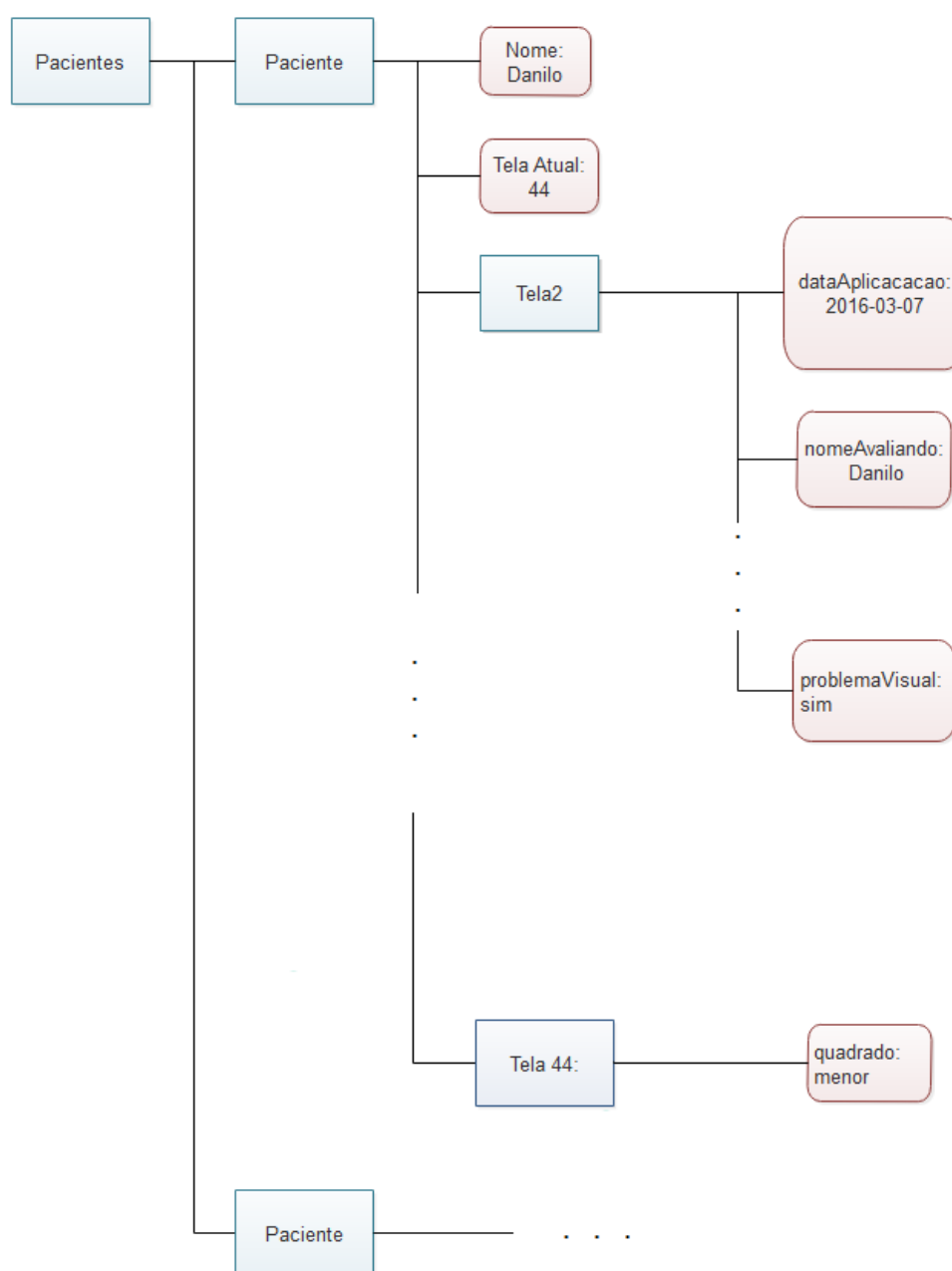


Figura 13 – XML - Tela2

Para que os dados fossem armazenados, inicialmente o usuário deveria fazer o upload do arquivo XML para que os dados de um respectivo paciente fossem recuperados, além desse modelo não ser uma maneira prática a utilização deste poderia ficar confuso para leigos no uso de tecnologias. Portanto optamos pelo uso do localStorage e sessionStorage. Para o armazenamento no localStorage, utilizamos a chave “Pacientes” com um valor no formato de JSON, sendo utilizado por ter uma estrutura mais leve comparado ao XML e como o localStorage possui uma quantidade de armazenamento máxima de dados, isso é de extrema importância. O sessionStorage foi utilizado para identificar qual paciente está sendo avaliado, sendo que no caso do usuário fechar o navegador e abrir o

projeto novamente, o sistema não reconhecerá o último paciente avaliado pois não terá nenhum dado no sessionStorage, fazendo com que o psicólogo seja redirecionado para a página gerenciamento de pacientes.

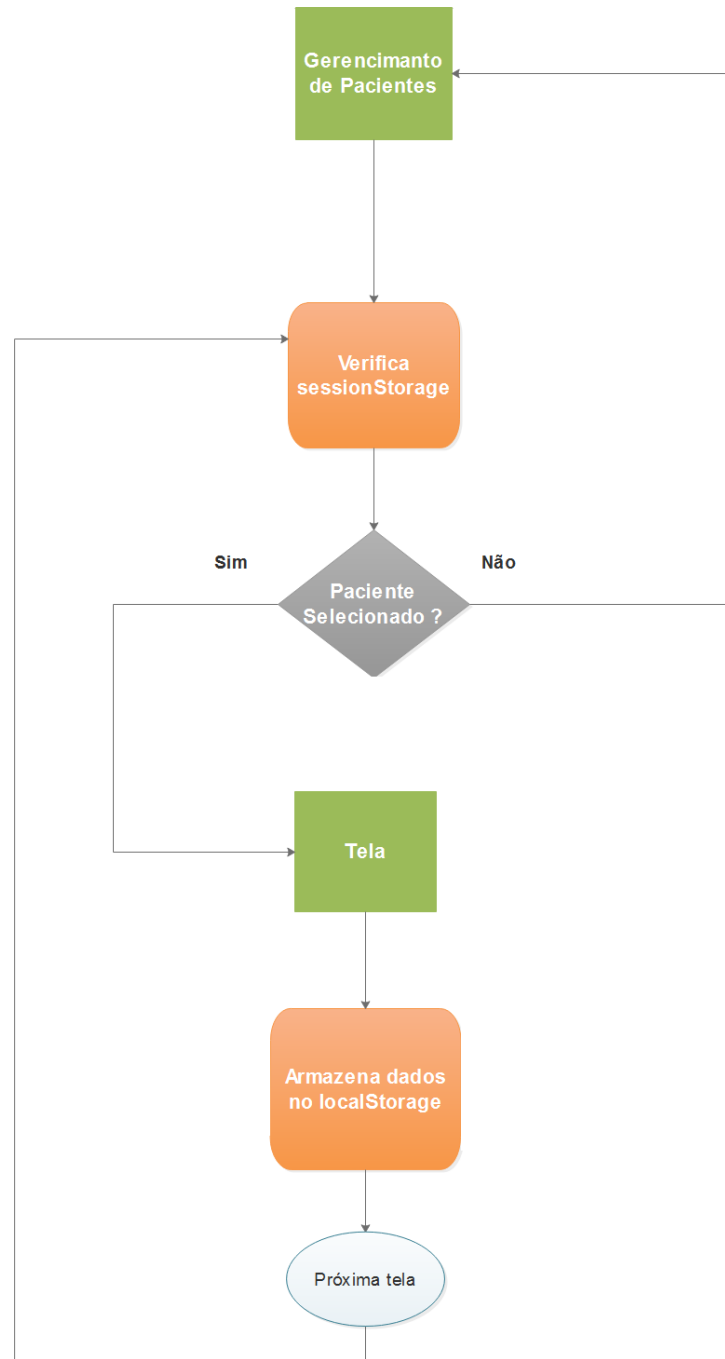


Figura 14 – Fluxograma

Após a definição do XML Schema começou-se o desenvolvimento das telas tanto na parte HTML quanto ao JavaScript pertencente a esta.

A primeira tela a ser implementada foi a tela de gerenciamento, como pode ser visto na figura 15, onde o usuário teria a opção de criar, selecionar e excluir um paciente.

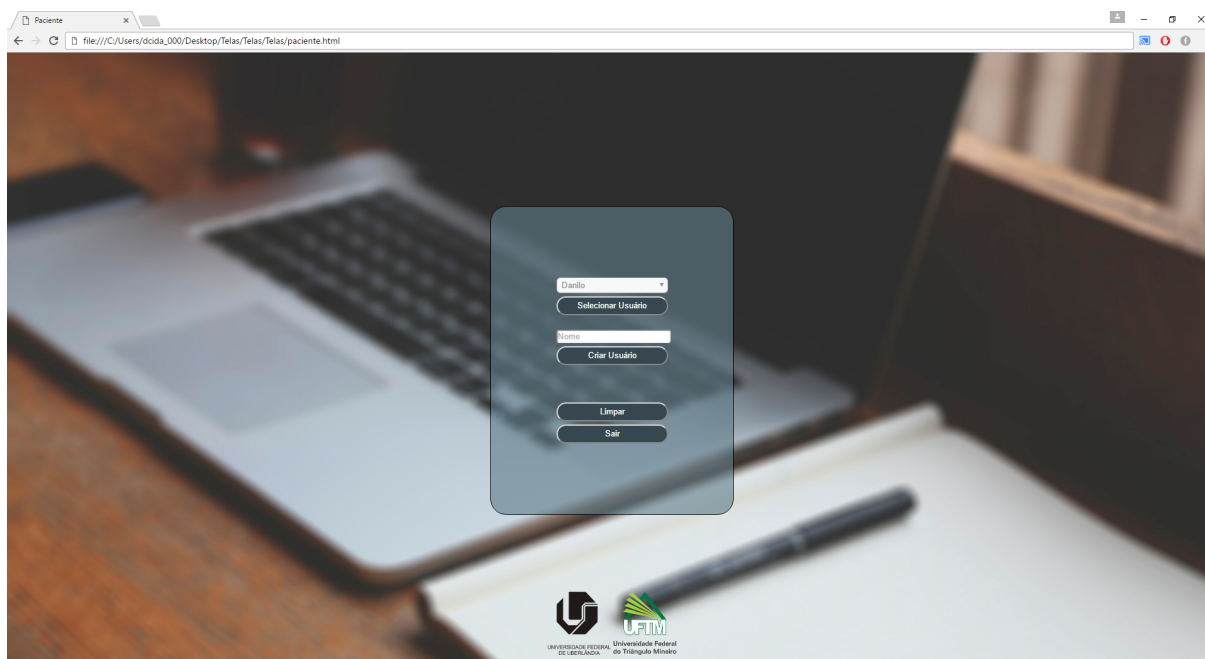


Figura 15 – Página de Gerenciamento

Para o HTML dessa tela foram utilizados componentes como botão, imagem e select.

```
<!DOCTYPE html>

<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta charset="utf-8" />
  <title>Paciente</title>
  <script type='text/javascript' src="paciente.js"></script>
  <link rel="stylesheet" type="text/css" href="css/paciente.css"/>
</head>
<body onload="carregar_dados()" class="background-image">
  <div class="background-image"></div>
  <div id = "paciente" class="center">
    <div>
      <select id="selectPaciente"></select>
      <br/>
      <button onclick="selecionar()">Selecionar Usuário</button>
      <br/><br/>
      <input type="text" placeholder="Nome" id="nome"> </input>
      <br/>
      <button onclick="criar_paciente()">Criar Usuário</button>
      <br/><br/><br/><br/>
      <button onclick="limpar()">Limpar</button>
      <br/>
      <button onclick="sair()">Sair</button>
    </div>
  </div>
  <div class="logo">
    
    
  </div>
</body>
</html>
```

Figura 16 – HTML-Página de Gerenciamento

Como podemos ver na figura 16 ao carregar a tag <body> é chamado a função carregarDados que está contido em um arquivo js (JavaScript) que será responsável por procurar pacientes já cadastrados e adicionar na tag <select>.

```
function carregar_dados() {  
    carregar_pacientes();  
}  
  
function carregar_pacientes() {  
    var select = document.getElementById("selectPaciente");  
    select.length = 0;  
  
    if (localStorage.getItem("Pacientes") !== null) {  
        var jsonObj = JSON.parse(localStorage.getItem("Pacientes"));  
        for (i = 0; i < jsonObj.Pacientes.length ; i++) {  
            var opcao = document.createElement("option");  
            opcao.text = jsonObj.Pacientes[i].Nome;  
            select.add(opcao);  
        }  
    }  
}
```

Figura 17 – JavaScript-Página de Gerenciamento

Além de funcionalidades como posicionamento, formato e tamanho dos componentes, o CSS foi utilizado para inserir uma imagem de fundo à tela de gerenciamento de clientes e prover um certo grau de transparência para a classe center.

```
.background-image {
    background: url("../images/background.jpg");
    -webkit-background-size: cover;
    -moz-background-size: cover;
    -o-background-size: cover;
    background-size: cover;
    background-repeat: no-repeat;
}

.center {
    position: absolute;
    background-color: rgba(96, 125, 139, 0.6);
    border-radius: 25px;
    text-align: center;
    display: table-cell;
    vertical-align: middle;
    border: 1px solid #000000;
    width: 20%;
    height: 50%;
    top: 0;
    bottom: 0;
    left: 0;
    right: 0;

    margin: auto;

    display: flex;
    align-items: center;
    justify-content: center
}
```

Figura 18 – CSS-Página de Gerenciamento

Logo após a implementação da tela de gerenciamento, começamos o desenvolvimento das telas referentes as páginas da bateria de avaliação. Abaixo é mostrado a primeira página da avaliação do paciente.



**BACI**

Porcentagem de itens respondidos

Tempo

Paciente: Augusto

Data de aplicação do teste: 12/11/2016

Nome do avaliado: Augusto

Data de nascimento: 01/07/1998

Idade: 24

Lateralidade: ☒ Destro ☐ Canhoto

Educacionalidade: ☒ Analfabeto/nunca frequentou escola ☐ 0 a 3 anos de escolarização ☐ 4 a 8 anos de escolarização ☐ mais de 8 anos de escolarização

Trabalho: Aposentado

Trabalho Atual: ☒ não, porque está aposentado ☐ não porque está desempregado/licenciado ☐ sim, está trabalhando

Natural de que estado/cidade: Uberlândia-MG

Sexo: ☐ Masculino ☒ Feminino

Possui limitação corporal visível? ☒ Sim ☐ Não

Possui problemas visual ou usa óculos? ☒ Sim ☐ Não

Salvar/Sair

Sair

SubmitPróximo

Figura 19 – Tela 2

As telas da avaliação possuem um padrão no carregamento e armazenamento de dados e verificação do paciente, mudando apenas dados pertencente a cada página.

Ao iniciar a tela será verificado se existe algum paciente relacionado com a página a atual, isso é feito para que o usuário não abra uma página sem antes ter selecionado um paciente. Como pode ser visto na figura 20, o código verifica se há algum paciente no localStorage, se caso não existir será mostrado uma mensagem de erro e o psicólogo será redirecionado para a tela de gerenciamento de pacientes, caso exista o paciente selecionado será transferido para a sessionStorage e removido da localStorage, garantindo que não haja nenhum paciente selecionado caso a página seja fechada.

```
function checar_paciente() {
    if(localStorage.getItem("PacienteSelecionado") == null)
    {
        alert("Selecione um paciente");
        window.location.href = "paciente.html";
        return false;
    }
    else{
        sessionStorage.setItem("PacienteSelecionado", localStorage.getItem('PacienteSelecionado'));
        localStorage.removeItem("PacienteSelecionado");
        paciente = sessionStorage.getItem("PacienteSelecionado");
        document.getElementById("nome").innerHTML = paciente;
        return true;
    }
}
```

Figura 20 – JavaScript - Verificar Paciente

Após verificado se há algum paciente indicado, será feito o carregamento dos dados caso este tenha dados armazenados. Com o uso do JSON, que está localizado no

localStorage, será carregado nos componentes HTML os dados que estão armazenados no arquivo JSON.

```
function carregar_paciente(json, i, nome) {  
    document.getElementById("dataAplicacao").value = json.Pacientes[i].Tela2.dataAplicacao;  
    document.getElementById("nomeAvaliado").value = json.Pacientes[i].Tela2.nomeAvaliado;  
    document.getElementById("dataNascimento").value = json.Pacientes[i].Tela2.dataNascimento;  
    document.getElementById("naturalidade").value = json.Pacientes[i].Tela2.naturalidade;  
    document.getElementById("trabalho").value = json.Pacientes[i].Tela2.trabalho;  
  
    if(json.Pacientes[i].Tela2.lateralidade != "")  
        document.getElementById(json.Pacientes[i].Tela2.lateralidade).checked = true;  
  
    if(json.Pacientes[i].Tela2.escolaridade != "")  
        document.getElementById(json.Pacientes[i].Tela2.escolaridade).checked = true;  
  
    if(json.Pacientes[i].Tela2.trabalhoAtual != "")  
        document.getElementById(json.Pacientes[i].Tela2.trabalhoAtual).checked = true;  
  
    if(json.Pacientes[i].Tela2.sexo != "")  
        document.getElementById(json.Pacientes[i].Tela2.sexo).checked = true;  
  
    if(json.Pacientes[i].Tela2.limitacaoCorporal != "")  
        document.getElementById(json.Pacientes[i].Tela2.limitacaoCorporal).checked = true;  
  
    if(json.Pacientes[i].Tela2.problemaVisual != "")  
        document.getElementById(json.Pacientes[i].Tela2.problemaVisual).checked = true;  
}
```

Figura 21 – JavaScript - Carregar Dados

Caso o cliente deseje trocar de paciente ou ir para a próxima página, o sistema deverá salvar as informações geradas. Na figura abaixo pode-se ver que será buscado o paciente selecionado no arquivo JSON para que as informações sejam salvas para o avaliando correto e então serão armazenados os dados coletados.

```

function salvar()
{
    json = JSON.parse(localStorage.getItem("Pacientes"));

    for(i = 0; i < json.Pacientes.length; i++)
    {
        if(json.Pacientes[i].Nome == sessionStorage.getItem("PacienteSelecionado"))
        {
            json.Pacientes[i].Tela2.dataAplicacao = document.getElementById("dataAplicacao").value;
            json.Pacientes[i].Tela2.nomeAvaliado = document.getElementById("nomeAvaliado").value;
            json.Pacientes[i].Tela2.dataNascimento = document.getElementById("dataNascimento").value;
            json.Pacientes[i].Tela2.trabalho = document.getElementById("trabalho").value;
            json.Pacientes[i].Tela2.naturalidade = document.getElementById("naturalidade").value;

            var allRadio = document.getElementsByTagName('input');
            for(j = 0; j < allRadio.length; j++)
            {
                if(allRadio[j].type == 'radio' && allRadio[j].checked == true)
                {
                    if(allRadio[j].name == 'lateralidade')
                        json.Pacientes[i].Tela2.lateralidade = allRadio[j].value;
                    else if(allRadio[j].name == 'escolaridade')
                        json.Pacientes[i].Tela2.escolaridade = allRadio[j].value;
                    else if(allRadio[j].name == 'trabalhoAtual')
                        json.Pacientes[i].Tela2.trabalhoAtual = allRadio[j].value;
                    else if(allRadio[j].name == 'sexo')
                        json.Pacientes[i].Tela2.sexo = allRadio[j].value;
                    else if(allRadio[j].name == 'limitacaoCorporal')
                        json.Pacientes[i].Tela2.limitacaoCorporal = allRadio[j].value;
                    else if(allRadio[j].name == 'problemaVisual')
                        json.Pacientes[i].Tela2.problemaVisual = allRadio[j].value;
                }
            }
        }
    }

    var ls = JSON.stringify(json);
    localStorage.setItem("Pacientes", ls);
}

```

Figura 22 – JavaScript - Salvar

## 3.4 Fase de Transição

A Fase de Transição é a parte na qual o sistema ou o software foi concluído e está disponível para o uso pelo usuário final. É claro que ao longo dessa fase, pode possuir alguns erros que precisam serem reajustados, então na etapa de transição também faz-se uso de testes, no caso feito pelas pessoas que utilizarão este.

Para este projeto, a fase de transição final será feita a partir do momento que houver a integração dos três projetos, entende-se por fase de transição final quando o usuário do sistema, no caso os psicólogos, forem utilizar o sistema avaliativo. Para esta parte do trabalho, foram realizados experimentos pelos desenvolvedores e também outros testes serão feitos pela equipe da UFTM.

Os testes realizados pelos desenvolvedores mostrou que o sistema é bem leve e ágil, não havendo uma percepção na rapidez de um navegador para outro, considerando que foram testados nos navegadores Mozilla Firefox e Google Chrome. Testes foram realizados utilizando o navegador Internet Explorer Versão 11, porém este restringe vários arquivos, entre eles o uso de scripts (ou seja o uso dos arquivos JavaScript), assim este se torna inviável o uso. Mesmo dando permissão para o uso de scripts foi detectado que

algumas funcionalidades ainda estão sendo bloqueadas e o navegador exibe uma mensagem de permissão constantemente. Também não houve falhas no projeto web, onde foram considerados falhas nos seguintes itens:

- Inconsistência nos dados
- Erros nos layouts
- Erros nas gravações de dados
- Erros nas funcionalidades dos componentes
- Erros nas leituras e recuperação das informações
- Erros por diferença de navegadores, com exceção do Internet Explorer

## 4 Conclusão

### 4.1 Considerações Finais

Um dos índices para que se tenha um envelhecimento com qualidade de vida e o bem-estar é os cuidados psicológicos. Um dos métodos utilizados por psicólogos é o treinamento cognitivo através de testes, jogos ou qualquer atividade que envolva o exercício adequado para o cérebro. Além do treinamento, outra medida importante para cuidados psicológicos é a prevenção de doenças neurais que atingem idosos. Dessa forma foi construído uma aplicação web, baseado em uma bateria avaliativa para idosos construída pela equipe da Universidade Federal do Triângulo Mineiro, utilizando ferramentas e linguagens computacionais a fim de facilitar o método utilizado pelos psicólogos e também o compartilhamento dos dados gerados a partir desses exercícios.

O projeto construído consiste em dois sistemas na qual foi desenvolvimento pela equipe da Universidade Federal de Uberlândia. Um utilizado pela UFTM, na qual consiste em um sistema gerenciador dos dados de cada paciente e psicólogo. E outro sistema utilizado pelos próprios psicólogos onde é composto por duas partes, a primeira é uma página de gerenciamento de pacientes e a segunda são várias páginas onde cada tela é dividido de acordo com as atividades envolvidas na bateria avaliativa.

Os sistemas foram construídos utilizando métodos e tecnologias a fim de facilitar o uso pelos avaliadores, considerando estes usuários leigos de sistemas. Como o projeto foi desenvolvido da maneira mais simples possível, não é preciso a instalação de qualquer software para o uso deste. Além disso, não é preciso que haja conexão com a internet, portanto é um sistema que sempre estará disponível para uso.

### 4.2 Trabalhos Futuros

Um projeto como este, onde o estudo e atividades cognitivo se encontra com a tecnologia, abre-se várias portas para possíveis melhorias em ambas as áreas. Uma área que tem crescido muito nos últimos anos, é a inteligência artificial na qual um sistema autônomo busca resolver problemas através de mecanismos e softwares a partir de um conjunto de dados. Seguindo essa linha de pensamento, um projeto futuro que seria interessante é a utilização da inteligência artificial no auxílio de detecção de doenças cognitivas.

Outros trabalhos possíveis seriam o uso de servidores de banco de dados, na qual facilitaria o compartilhamento de informações, podendo assim abranger o estudo nas áreas

relacionadas a cognição e também a criação de um número maior de atividades.

# Referências

Corso, H. V., Sperb, T. M., Jou, G. I., Salles, J. F. (2013). Metacognição e funções executivas: relações entre os conceitos e implicações para a aprendizagem. *Psicologia: Teoria e Pesquisa*, 29(1), 21 – 29.

Yates, D. B., Zibettia, M. R., Pawlowskib, J., Sallesa, J. F., Parentec, M. A. M. P., Argimond, I. L., Fonseca, R. P., Trentini, C. M. (2013). WCST and NEUPSILIN: relationships among executive functions, attention, memory and language. *Psicologia: Reflexão e Crítica*, 26(3), 506-515.

Yassuda, M. S., Abreu V. P. S. (2006). Avaliação cognitiva. In E. V. Freitas, L. Py, F. A. X. Cançado, J. Doll, M. L. Gorzoni (Eds.), *Tratado de geriatria e gerontologia* (pp.1252-1266). Rio de Janeiro, RJ: Koogan.

Irigaray, T. Q., Gomes Filho, I., Schneider, R. H. (2012). Efeitos de um treino de atenção, memória e funções executivas na cognição de idosos saudáveis. *Psicologia: Reflexão e Crítica*, 25(1), 188 - 202.

Bavelier, D., Achtman, R., Föcker, M. (2012). Neural bases of selective attention in action video game players. *Vision Res*, 61, 132 – 143.

Green, C. S., Bavelier, D. (2008). Exercising Your Brain: A Review of Human Brain Plasticity and Training-Induced Learning. *Psychology and Aging*, 23(4), 692 – 701.

Capovilla, A. G. S.; Capovilla, F. C. (2007). *Teoria e pesquisa em Avaliação neuropsicológica*. São Paulo: Memnom.

DEVMEDIA.O que é o html5. Disponível em: <<http://www.devmedia.com.br/o-que-e-o-html5/25820>>. Acesso em: 29 nov. 2016.

DEVMEDIA.Artigo engenharia de software - o processo unificado integrado ao desenvolvimento web. Disponível em: <<http://www.devmedia.com.br/artigo-engenharia-de-software-o-processo-unificado-integrado-ao-desenvolvimento-web/8032>>. Acesso em: 02 dez. 2016.

SIGNIFICADO.Significado de xml. Disponível em: <<https://www.significados.com.br/xml/>>. Acesso em: 30 nov. 2016.

W3SCHOOL.Xml schema. Disponível em: <[http://www.w3schools.com/xml/schema\\_intro.asp](http://www.w3schools.com/xml/schema_intro.asp)> .Acessoem : 30nov.2016.

LARMAN, Craig.Utilizando uml e padrões:Uma introdução à análise a ao projeto orientados a objeto e ao desenvolvimento iterativo.3 ed.[S.L.]:Bookman,2007.685 p.