**Documento de requisitos do sistema MatriculeMe**

**Versão 1.0**

**Histórico de Alterações**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| 21/09/2016 | 1.0 | Criação do documento | Nicolas Meinen |

1. **Requisitos funcionais (Casos de uso)**

|  |  |
| --- | --- |
| [RF01] | Identificação |

**Descrição:** O usuário deverá se identificar para que sejam possíveis as ações.

**Entrada:** Matrícula e senha.

**Saídas:** Com o usuário identificado, haverá a extração de dados deste e o armazenamento no banco de dados.

|  |  |
| --- | --- |
| [RF02] | Visualização de componente |

**Descrição:** O usuário terá a possibilidade de visualizar seus dados.

**Entrada:** O que o usuário deseja ver.

**Saídas:** Os dados de fato, como o IRA, opções sugeridas, etc. Aqui entra todo e qualquer tipo de dado: só será possível, portanto, requisitar algumas coisas depois do *software* realizar o necessário para colocar essas informações.

|  |  |
| --- | --- |
| [RF03] | Aprendizagem com base nos dados do usuário |

**Descrição:** Baseado no que fora fornecido ao sistema, este deverá aprender como é o comportamento do usuário em termos escolares para que seja possível gerar sugestões para o mesmo.

**Entrada:** Dados já presentes no sistema.

**Saídas:** -

|  |  |
| --- | --- |
| [RF04] | Determinação de dados usáveis |

**Descrição:** No geral, o *software* deve perguntar ao usuário quais dados poderá acessar dentro do sistema de matrícula. Isso poderá ser feito previamente, como no momento de apresentação dos Termos de Uso. O usuário deverá ter a possibilidade de alterar isso *a qualquer momento*.

**Entrada:** O que o usuário permite que o sistema use.

**Saídas:** O software manipula o banco de dados (armazena, deleta, altera) de acordo com o que lhe for fornecido.

|  |  |
| --- | --- |
| [RF05] | Montagem e apresentação de possíveis grades |

**Descrição:** Tornar possível o usuário selecionar diversas disciplinas e montar diversas grades.

**Entrada:** Pode ou não haver a montagem manual desses horários por parte do usuário.

**Saídas:** Apresentação dessas grades.

|  |  |
| --- | --- |
| [RF06] | Exportação de grades |

**Descrição:** Exportar foto das grades para o telefone do usuário.

**Entrada:** -**.**

**Saídas:** Fotos dessas grades (de fácil visualização) armazenadas no telefone do usuário.

|  |  |
| --- | --- |
| [RF07] | Extração de dados do Matricula Web |

**Descrição:** O *software* deverá extrair os dados do matricula web de cada usuário.

**Entrada:** -.

**Saídas:** Armazenamento desses dados num banco de dados próprio do sistema.

|  |  |
| --- | --- |
| [RF08] | Segurança |

**Descrição:** Dado que o usuário realizará *login* no matricula web, o software deverá garantir a criptografia desses dados.

**Entrada:** Dados.

**Saídas:** Dados encriptografados.

1. **Requisitos não funcionais**

|  |  |
| --- | --- |
| [RNF01] | Facilidade de uso |

É fato que nem todo usuário está acostumado a utilizar aplicativos mirabolantes e que requerem concentração na hora de uso. Além disso, o usuário não vai sentir a vontade de baixar o aplicativo se tiver de levar horas lendo documentações sobre como usar. Portanto, fica claro aqui que o *software*, apesar de sofisticado em camadas mais internas, deverá ser totalmente acessível a qualquer tipo de usuário, do mais leigo ao mais profissional. Isso consiste em:

* Botões auto explicativos – quando o usuário apertar em *sugestões*, que sejam apresentadas sugestões.
* Pouca quantidade de entrada de dados – quanto mais rápido e prático for para que o usuário possa se matricular, melhor. Não seria bom haver de clicar ali, ditar que tipo de sugestão quer, escrever qual a forma de apresentar, etc.

|  |  |
| --- | --- |
| [RNF02] | Desempenho |

Tendo em vista que a ideia do aplicativo surgiu a partir da demora e dificuldade da matrícula, o sistema em si deverá apresentar um desempenho considerável, com uma resposta num tempo menor do que 2 segundos para as sugestões.

|  |  |
| --- | --- |
| [RNF03] | Compatibilidade e linguagem de alto nível |

Será necessária a escolha de uma linguagem de alto nível com alta portabilidade, fácil reuso e flexível. Tendo isso em mente, a sugestão mais viável seria o desenvolvimento da aplicação em Java. Em termos de *Hardware*, qualquer telefone que utilize como SO o Android poderá utilizar-se da aplicação. O único detalhe é que, em celulares menos rápidos, a aplicação também não terá todo o rendimento desejado.