Análise Comparativa de Requisitos

Nome: Lukas Lorenz de Andrade

1. Requisitos atuais:
   * Impressão 3d;
   * Circuito de adaptação da fonte de alimentação dos LEDs;
   * 1 Fita de LED para iluminação interna;
   * 1 Raspberry pi 3b;
   * 4 Pés de borracha aparafusados;
   * 1 Suporte para os CARDS;
   * CARDS;
   * 1 Tela LCD;
   * Câmera 5Mp 720p foco automático;
2. Especificações da tela LCD:
   * Touch do tipo resistivo (sensor de baixa resolução e qualidade);
   * Resolução 480x320 (320p);
   * Polegadas 3.5in (8.5x5.5cm);
   * SO: Raspbean, debian, kali – todas distribuições Linux;
3. Requisitos desejáveis:
   * Resolução 768p (1366x768)
   * Dimensão 9 ou 10 in (20x10cm ou 20x15cm)
   * Touch capacitivo
   * SO linux ou windows;
4. Quadro comparativo :

|  |  |  |
| --- | --- | --- |
| Especificações | TELA | Tablet |
| Resolução | Geralmente abaixo da desejada, algumas com capacidade de aumentar via software | Naturalmente abaixo ou acima da desejada, sem ajuste de configuração |
| Dimensão | 7 ou 9in | 7 a 9in |
| SO | Linux ou windows | Android |
| Touch | Pode ser Capacitivo (não se tem informações sobre precisão) | Capacitivo (não se tem informações sobre precisão) |

1. Considerações finais:
   * Tela: Mais fácil de integrar, sistema operacional já compatível com o backend e frontend, necessidade do Raspberry pi, quantidade de elementos dos requisitos iniciais permanece o mesmo, programação em qualquer linguagem sem necessidade de adequar os códigos até então já escritos (backend: Python; frontend: html, css, js) ao SO, mais barato em média (200 a 400).
   * Tablet: SO android o que necessita que os códigos sejam adequados para java (rescrever em java e C++ ou traduzir para java – retrabalho do backend), quantidade de elementos dos requisitos iniciais reduzida ao tablet e a fita de LED, melhor adequação do frontend devido a resoluções mais próximas da desejável, mais caro em média (350 a 500), geralmente megapixels menores que a câmera usada (precisa analisar a resolução da câmera).

OBS: Com o Tablet é possível fazer o double check dos cards, verificação em duas etapas das amostras, devido ao fato de possuir uma câmera frontal e a câmera principal. **Teria de ser feito um aplicativo android** para interface e integração com o IoTest. Recompilação dos frameworks, como a opencv (visão computacional), para android.

OBS: Com o Tablet, o processamento de visão computacional poderia ser feito em nuvem, de forma a diminuir o esforço de processamento sobre o tablet e sem necessidade de adequação do backend para java. Isso implicaria em um custo mensal de aluguel serviço em nuvem. As máquinas alugadas para o sistema lacpoint poderiam ser aproveitadas para o processamento do IoTest contanto que seja adequado. Ou seja, uma vez alugadas as máquinas do serviço em nuvem para o sistema LacPoint, teria de ser feito um upgrade pra máquinas melhores para comportar processamentos simultâneos do sistema do IoTest, sendo um pouco mais caro que o custo inicialmente planejado. Outra opção para não depender de aluguel é comprar um computador gamer, por exemplo, e usá-lo como servidor na Macofren, hospedando o serviço LacPoint e IoTest.

Sugestão de compra do tablet: <https://www.arantxamagazine.com.br/item/Tablet-Samsung-Galaxy-Tab-A-T280-8GB-Wi%252dFi-Tela-7%22.html>