

# Imaginando Milton

Hugo Cristo Sant'Anna - hugo.santanna@ufes.br

## Sumário

<b>1</b>	<b>Contextualização</b>	<b>1</b>
<b>2</b>	<b>Ok, estou motivado</b>	<b>3</b>
<b>3</b>	<b>Milton imaginado</b>	<b>6</b>
3.1	Para estudantes . . . . .	6
3.2	Para professores . . . . .	7
3.3	Para familiares e responsáveis . . . . .	8
3.4	Para instituições de ensino . . . . .	8
3.5	Para investidores e apoiadores . . . . .	8
3.6	Para desenvolvedores . . . . .	8
<b>4</b>	<b>Detalhamento de hardware e software</b>	<b>9</b>
4.1	Componentes . . . . .	9
4.2	Ambiente de desenvolvimento . . . . .	10
<b>5</b>	<b>Documentação do processo</b>	<b>10</b>
<b>6</b>	<b>Por que desenvolver este projeto?</b>	<b>10</b>
<b>7</b>	<b>Atualizações</b>	<b>11</b>
<b>8</b>	<b>Licença</b>	<b>12</b>

## 1 Contextualização

No dia 01/07/2020, a notícia sobre o computador de R\$ 22 criado por Ciswal Santos se espalhou pelas redes. Em tempos menos difíceis, o invento interessaria basicamente a entusiastas da eletrônica e computação.

Entretanto, a suspensão das aulas presenciais na rede pública em função da pandemia de Covid-19 muda radicalmente a relevância da iniciativa: muitos estudantes de baixa renda não dispõem de computadores pessoais (PCs) para continuarem seus estudos na modalidade de ensino remoto.

Os celulares são muito populares, porém ergonomicamente problemáticos para uso contínuo em atividades que seriam ordinárias em PCs — produção de textos, edição de apresentações e imagens etc. Os cenários de uso dos aplicativos do

tipo *office* para dispositivos móveis são mais relacionados à leitura, acesso rápido ou correções simples de documentos editados previamente em PCs.

Nesse sentido, a ideia de Ciswal (“HYTEC One”) é muito potente:

- Custo de R\$ 22;
- Tamanho de uma caixa de fósforo;
- Pode ser conectado a qualquer tela (monitor ou celular);
- Sistema operacional próprio com pacote de aplicações do tipo *office*.

As notícias sobre o HYTEC não incluem detalhes das tecnologias utilizadas. A foto do protótipo divulgada exhibe uma mini *protoboard* com os tais “componentes simples”, teclado e mouse USB conectados, e a tela de celular com a interface do Windows 98.



Figure 1: HYTEC One (foto do arquivo de Ciswal Santos)

A tentativa de ofertar de computadores para contextos educacionais não é nova. Tivemos o *time-sharing* de DartMouth nos anos 1960 com a linguagem BASIC utilizada por estudantes de *todas* as áreas. Seymour Papert e colaboradores exploraram possibilidades desde o final dos anos 1960, envolvendo a linguagem LOGO. Quando o mercado dos computadores pessoais domésticos finalmente engrenou (1980+), surgiram plataformas maravilhosas como BBC Micro, explicitamente educacional, e outras como o MSX, que estavam na área cinza jogos-educação.

Nenhuma dessas alternativas era barata. Havia máquinas muito acessíveis, como o Sinclair ZX Spectrum inglês (£125 em 1982, ~R\$ 2mil hoje), embora chegassem a países como o Brasil com certa dificuldade. Clones locais ajudaram

na popularização, apesar dos preços altos para a maior parte da população.

A busca por *baixo custo* entrou no horizonte *tardamente*. Provavelmente, se projetos como Arduino e Raspberry Pi não existissem, o cenário atual para Ciswal seria outro. Há amplo mercado para microcontroladores e demais componentes de baixo custo, utilizados por estudantes, profissionais e curiosos. Sites com o Mercado Livre, DF Robot e Ali Express são parques de diversões para quem quer gastar uns trocados e montar em robôs caseiros.

O mérito da iniciativa de Ciswal Santos é inegável e o momento é perfeito: o Brasil enfrentará níveis alarmantes de evasão em todos os níveis de ensino se a inclusão digital não for *política de Estado*. Empurrar os estudantes para a frente do PC, por pior que seja, pressupõe a existência do equipamento adequado, da conexão à internet com velocidade suficiente, dos programas necessários para o processo de ensino-aprendizagem remota para cada disciplina e assim por diante (leia mais aqui).

Acredito que muitos pesquisadores das áreas de Design, Computação e Educação ficaram entusiasmados com a proposta do HYTEC. Eu espero que seja viável, que chegue ao mercado o quanto antes e que motive mais gente a experimentar este tipo de solução de baixo custo.

## 2 Ok, estou motivado

Não tenho a formação necessária para projetar PCs como o HYTEC integralmente. Sou programador autodidata, iniciado no MSX ainda nos anos 1980, e sigo com interesses permanentes nas interfaces entre Computação, Design e Psicologia. Tenho minha coleção de Arduinos, Raspberry Pi e outros microcontroladores menos populares, e desenvolvi projetos simples por diversão.

Arduinos são plataformas de prototipagem baratas (placa original US\$ 23, clones por ~R\$ 30 no Brasil) simples, fáceis de aprender e de usar, porém com recursos distantes de PCs contemporâneos — processador 8 de bits funcionando a 16MHz, 32KB de memória Flash, 2KB de RAM. Não obstante, há projetos incríveis construídos utilizando as plaquinhas:

- ArduBoy é um console retrô para jogos 8bit construído sobre o Arduino Leonardo;
- Arduino BASIC PC utiliza dois Arduinos trabalhando em conjunto: um rodando o interpretador TinyBasic e controlando teclado, outro gerando saída de vídeo VGA em cores (120x60 pixels).

É possível aumentar os recursos das plaquinhas empilhando *shields* de rede, leitor de cartões SD, câmeras e muitos outros. As bibliotecas desenvolvidas pela comunidade facilitam a comunicação de dados com e sem fio, geração de som e imagem. A programação dos Arduinos é feita principalmente em C, no ambiente de desenvolvimento integrado (IDE) mais popular do setor. Outras placas utilizam o ambiente do Arduino em função da familiaridade dos usuários.

O Raspberry Pi é outro mundo: computador completo em placa única, incluindo rede, USB, HDMI e cartão de memória, rodando Linux. O modelo atual (Pi 4) tem processador de 64 bits rodando a 1.5GHz, *duas* saídas de vídeo HDMI (meu

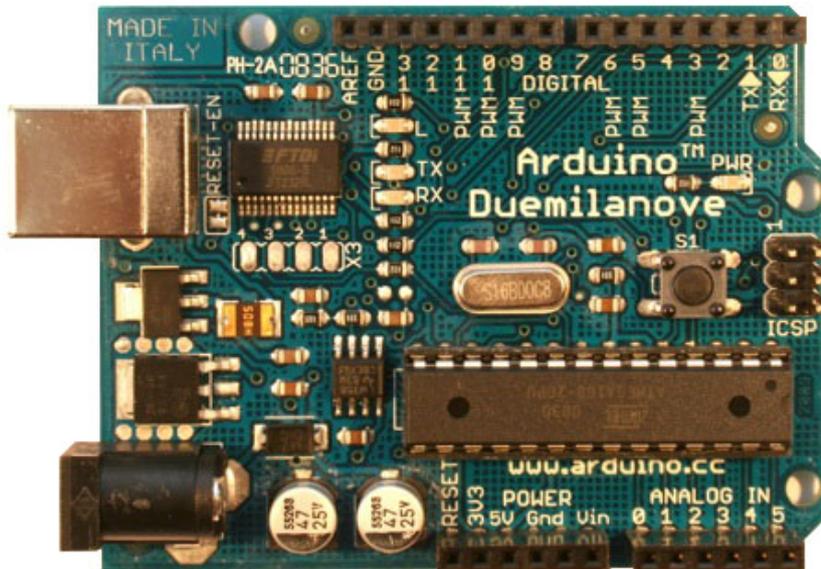


Figure 2: Arduino Duemilanove, meu primeiro!

PC de trabalho só tem uma) e pode chegar a 8GB de memória RAM. Tudo isso mantendo os pinos para acoplar sensores e *shields*, como os Arduinos.

A programação do Raspberry pode ser feita em muitas linguagens utilizadas em PCs (C, Python, Javascript) e em projetos educacionais (Scratch ou Snap!). Existem versões de praticamente tudo disponível no Linux para o Pi e há pessoas utilizando a placa como computador principal.

No Brasil, o Pi 4 é comercializado por *no mínimo* R\$ 600. Com mais R\$ 400-500, é possível comprar *notebooks quadcore* de marcas genéricas com 2GB de RAM e HD de 32GB. Apesar de não serem máquinas potentes, acompanham teclado, tela, *touchpad*, rede sem fio e o básico para a maioria dos estudantes.

Há uma terceira via, chamada NodeMCU. É compatível com Arduino e direcionado a projetos da chamada “Internet das Coisas” (IoT), automação residencial e monitoramentos diversos. O principal atrativo é a conexão Wi-Fi integrada e custo baixo (~R\$ 30-40 no Brasil).

A plataforma utiliza o sistema-em-um-chip ESP8266, com processador de 32 bits, 128KB de RAM e ao menos 4MB de memória Flash. Pode ser programada em C, Lua e Javascript. Eu não conhecia a placa e resolvi comprar para experimentar.

Eu estou desenvolvendo uma linguagem de montagem e máquina virtual em C99, chamadas respectivamente **Adele Assembler** e **Adele VM**. Meu objetivo original foi construir o conjunto compilador + interpretador portátil, capaz de ser compilado e rodar em qualquer sistema operacional atual — Windows, macOS, \*nix — como em PCs antigos rodando MS-DOS e Arduinos. A máquina virtual tem API em C e é bastante adaptável, permitindo reduzir ou ampliar

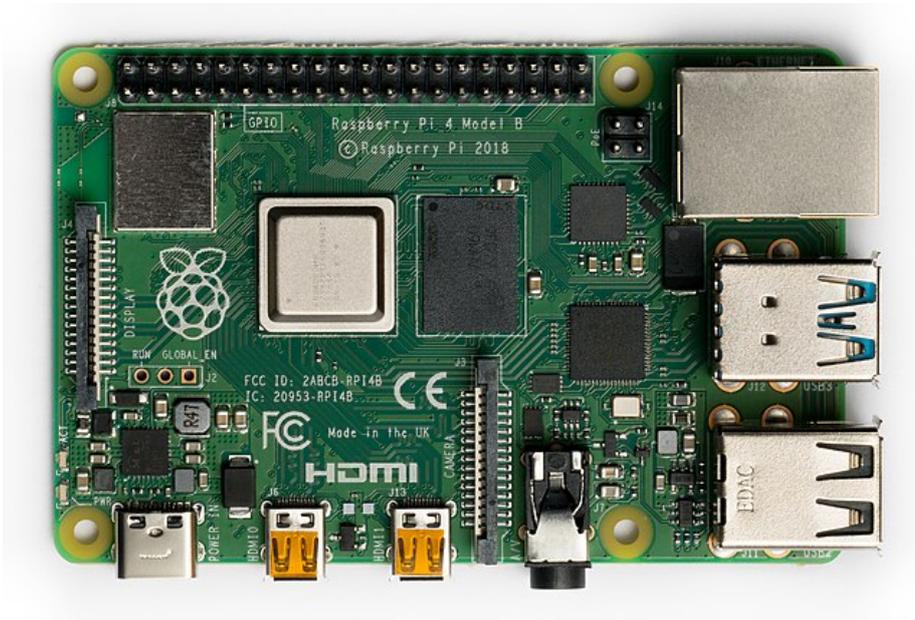


Figure 3: Raspberry Pi 4

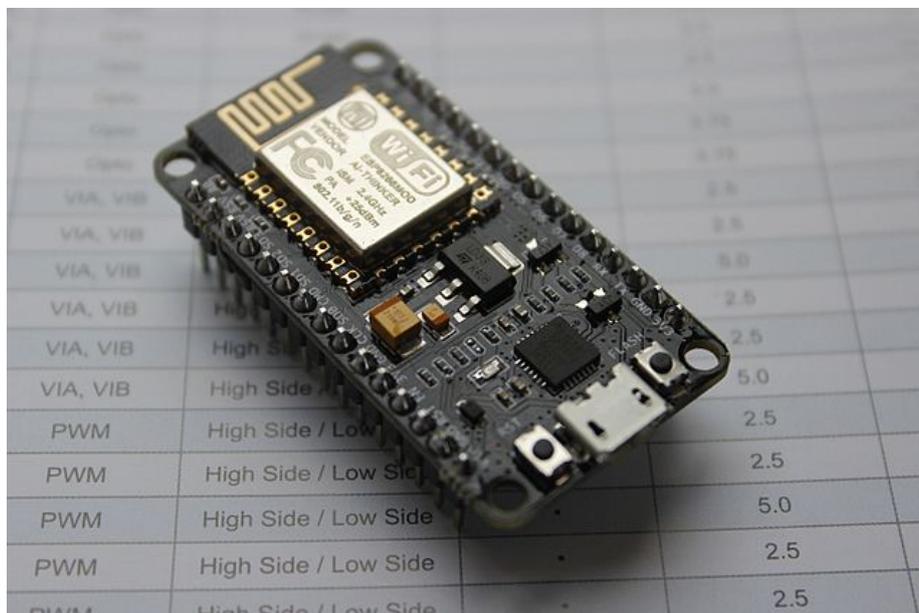


Figure 4: NodeMCU

bastante as funcionalidades e uso de recursos conforme a plataforma de destino.

A descoberta do NodeMCU resolveria algumas limitações que tenho com a implementação de Adele no Arduino (leia-se memória) e reduziria o número de *shields* para habilitar conexão de rede. Além disso, o processador é mais robusto, ajudando na geração de vídeo e áudio, e há mais memória Flash disponível. *Eis que surge a proposta.*

### 3 Milton imaginado

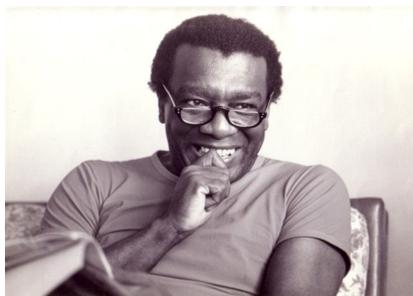


Figure 5: Milton Santos

Milton Santos (1926-2001) foi um dos mais importantes pensadores negros brasileiros. Bacharel em direito e doutor em Geografia, estudou os problemas sociais do Brasil a partir da investigação crítica do espaço, dos meios sócio-técnicos e relações com a urbanização e globalização.

Homenagear Milton Santos me pareceu adequado: a educação é parte indissociável do processo de redução das desigualdades no país e meios sócio-técnicos precisam ser considerados de forma crítica na formulação de alternativas.

O projeto **Milton** visa oferecer *mais* do que um computador pessoal de baixo custo. É um sistema computacional em rede com diversos elementos articulados para apoiar o processo de ensino-aprendizagem remoto.

Inicialmente, o foco de Milton é o ensino de disciplinas de **Português** e **Matemática**, com extras para o ensino de pensamento computacional.

#### 3.1 Para estudantes

Computadores compartilhados são a realidade de muitos domicílios brasileiros. Milton consiste no ponto de acesso ao conteúdo e de submissão de atividades.

Estudantes podem usar Milton como dispositivo principal para acessar e submeter informações, ou podem utilizá-lo como servidor daquelas informações para outros dispositivos — celulares, tablets e computadores.

Milton é flexível:

- Onde houver rede com ou sem fio, Milton pode ser utilizado como meio de acesso principal a conteúdos didáticos;

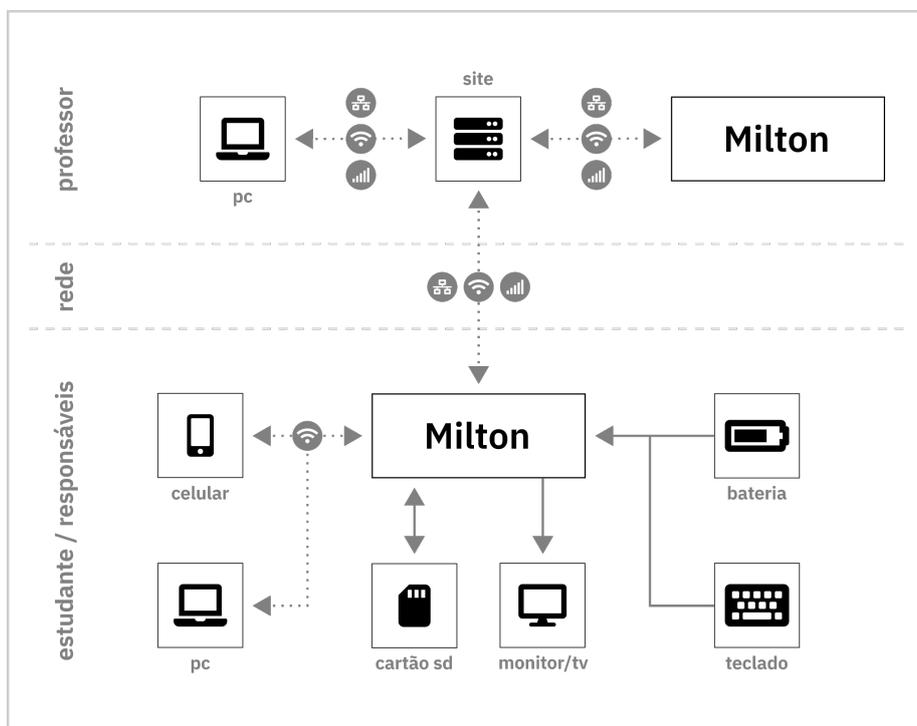


Figure 6: Arquitetura Milton (v0.1)

- Onde houver rede de dados móveis (3G+), Milton pode acessar conteúdos didáticos conectado por meio de qualquer celular capaz de rotear a conexão;
- Onde não houver conexão, Milton pode ser transportado até o ponto de Internet mais próximo para fazer a recuperação dos conteúdos didáticos e submissão de atividades feitas, alimentado apenas por baterias — a *sincronização*;
- Como dispositivo principal, Milton pode ser conectado a monitores e televisores (porta VGA) ou utilizar o visor embutido para leitura dos conteúdos e realização das atividades. A interação e navegação com os materiais pode ser feita por meio da interface física (botões e visor) ou com auxílio de teclados padrão PS/2 (~R\$ 30).
- Como servidor de conteúdos e atividades, cada unidade Milton pode ser acessada por até cinco (05) dispositivos simultaneamente, conectados via TCP/IP;
- A alimentação de Milton pode ser feita por baterias ou adaptadores (9V).

### 3.2 Para professores

Milton consiste na plataforma de gestão do ensino, facilitando a publicação de conteúdos e avaliação de atividades de maneira integrada e agnóstica em relação às formas de acesso dos estudantes.

Milton é flexível:

- O professor com acesso permanente à Internet e computador próprio pode publicar conteúdos e atividades didáticas no site do projeto. Todos os dispositivos Milton inscritos na referida disciplina receberão os materiais sob demanda, assim que conectados à Internet;
- O professor sem acesso permanente à Internet e computador próprio pode se conectar à sua unidade Milton e editar conteúdos e atividades localmente. Em seguida, pode transportar Milton até um ponto de acesso, alimentado apenas por baterias, para fazer a publicação na rede e receber atividades concluídas pelos estudantes para correção — a *sincronização*;
- O professor sem acesso permanente à Internet e sem computador próprio pode conectar Milton a monitores e televisores (porta VGA) ou visualizar os conteúdos no visor embutido do dispositivo. Para a edição de conteúdos extensos, o uso dos botões da interface física é possível, apesar de trabalhoso. Recomenda-se a utilização de teclado PS/2 (~R\$ 30) ou edição por meio de tablets e celulares conectados a Milton como servidor.

### 3.3 Para familiares e responsáveis

Milton permite o acompanhamento sistemático dos estudantes. Todos os conteúdos didáticos acessados e atividades entregues, bem como suas avaliações, podem ser visualizadas diretamente em Milton ou acessadas por meio de celular, tablet ou computador conectado ao dispositivo.

### 3.4 Para instituições de ensino

Milton centraliza a disponibilização e acesso a conteúdos didáticos a uma única plataforma de operação assíncrona por cada perfil de usuário. Todos os relatórios por estudante, turma e professor estão disponíveis no site do projeto de forma segura:

- Rendimento individual e por turma;
- Taxa de acesso por conteúdo, disciplina, aluno e turma;
- Atividades enviadas, concluídas, não realizadas, corrigidas e faltando correção.

### 3.5 Para investidores e apoiadores

Estou trabalhando com o horizonte de **R\$ 50** de custo, incluindo a placa NodeMCU, visor LCD embutido, botões, carenagem e conectores.

Buscarei apoiadores interessados em viabilizar unidades Milton para grupos de estudantes e professores de escolas públicas.

### 3.6 Para desenvolvedores

Aplicações para Milton poderão ser desenvolvidas utilizando o SDK do NodeMCU ou em linguagens que compilem para Adele Assembler. Pretendo disponibilizar as seguintes alternativas:

- HTML/CSS/JS para aplicações em que Milton é servidor;
- C, Lua e Adele Assembler para aplicações em que Milton é o dispositivo principal;

- Ambiente de desenvolvimento para PCs, com pré-visualização das aplicações e recursos de publicação em plataforma de aplicativos Milton.

Naturalmente será possível produzir jogos e aplicações com finalidades paradiáticas para Milton.

## 4 Detalhamento de hardware e software

A lista a seguir é *preliminar* e exhibe os componentes e programas desenvolvidos até o estágio atual do projeto.

### 4.1 Componentes

Os custos *unitários* correspondem ao valor de venda ao consumidor final (varejo) no Brasil (sem importação) em **reais**. Os custos por *atacado* foram estimados para a importação de 20 (vinte) ou mais unidades do componente (sem impostos e sem frete, em R\$). Componentes menores (resistores e capacitores) foram contabilizados, porém os custos unitários são desprezíveis.

Componente	Função	Unidade	Atacado
NodeMCU v3	Controlador principal	30,00	10,00
Visor OLED 1.4pol 240x240	Configurações iniciais de Milton (Wi-fi), senhas, saída de vídeo	29,00	9,00
Soquete PS/2	Conexão do teclado	9,00	0,90
Arduino Nano V3	Controle de E/S	30,00	10,00
Módulo Bluetooth HC-06	Conexão de periféricos sem fio (teclado, mouse etc.)	20,00	2,30
Leitor de cartão MicroSD (até 128gb)	Memória Flash para dados do usuário	15,00	2,60
2x memória EPROM Atmel AT24C64AN (8kb)	Armazenamento elementos básicos da interface com o usuário (16kb)	5,00	0,40
Buzzer	Emissão de tons	2,00	0,30
Conector Jack J2 estéreo	Saída para fone ouvido ou caixas acústicas	1,60	0,40
Jack RCA fêmea duplo	Saída TV (A/V)	2,50	1,50
Conector DB15 fêmea VGA	Circuito VGA	2,50	0,40
2x Resistores 220 Ohms	Circuito VGA	-	-
1x Resistor 470 Ohms	Circuito VGA	-	-
Jumpers			
Solda			

## 4.2 Ambiente de desenvolvimento

Estou programando o NodeMCU em duas plataformas simultaneamente, para comparar prós e contras de cada uma: Arduino IDE (1.8, em C) e NodeMCU SDK (v3.0, em Lua). Algumas partes do sistema utilizam bibliotecas geradas por terceiros, outras foram adaptadas ou integralmente desenvolvidas por mim.

Biblioteca ou programa	Função	Autor(es)
Arduino IDE 1.8	Ambiente de desenvolvimento em C e bibliotecas de base	Arduino
NodeMCU SDK 3.0	Ambiente de desenvolvimento em Lua e bibliotecas de base	NodeMCU
espgax2	Geração de sinal VGA 4bits	Sandro Maffiodo
Lua.vm.js	Implementação Javascript da máquina virtual da linguagem Lua para criação e edição de imagens em <code>milton:p</code> e <code>milton:u</code>	Lua: PUC-Rio Lua.vm.js: Daurnimator
<code>milton:p</code>	Editor de imagens 4 bits para utilização com a biblioteca <code>espgax2</code> (HTML5/CSS/JS/Lua)	Hugo Cristo (demo)
<code>milton:u</code>	Editor de interfaces utilizando imagens 4 bits geradas em <code>milton:p</code> (HTML5/CSS/JS/Lua)	Hugo Cristo (demo)
<code>milton:d</code>	Sistema do dispositivo, responsável pelo acesso a arquivos e integração de todas as aplicações (C e Lua)	Hugo Cristo
<code>milton:f</code>	Editor de fontes bitmap apropriadas às múltiplas saídas e resoluções.	Hugo Cristo

## 5 Documentação do processo

Em 08 de agosto de 2020 criei o blog do projeto para detalhar as dificuldades de progressos. Este documento permanece como síntese final das arquiteturas, custos e conceito geral de Milton.

Acesse: [hugocristo.com.br/projetos/milton/blog](http://hugocristo.com.br/projetos/milton/blog)

## 6 Por que desenvolver este projeto?

Estou envolvido com educação tecnológica de 2000. Sou professor de disciplinas que ensinam programação no Curso de Design da Ufes desde 2009. Entre 2012 e 2015 realizei oficinas de introdução à linguagem Scratch para professores do

ensino básico e médio. Minha tese de doutorado em Psicologia (2014) investigou formas de ensinar princípios da computação para estudantes de Design. No meu estágio de pós-doutorado (2018), pesquisei sistemas personalizados de instrução.

Milton é uma tentativa de desenvolver um sistema de ensino-aprendizagem completo (*online+offline*), considerando as necessidades de estudantes, familiares, professores e instituições durante e após a pandemia, mantendo os requisitos de custo baixo, acessibilidade e facilidade de uso.

## 7 Atualizações

### 08/08/2020

- Decisão de subdividir a arquitetura em três (03) implementações distintas: A (com Arduino), E (com ESP8266) e E32 (com ESP32).
- Criação do blog do processo.

### 06/08/2020

- Adoção temporária de Arduino Nano V3 como controlador de entradas e saídas de teclado e bluetooth do ESP8622.
- Retorno da conexão PS/2 :)

### 04/08/2020

- Substituição de PS/2 por Bluetooth (excesso de instabilidades no ESP8266)

### 31/07/2020

- Implementação do acesso aos dados armazenados na EEPROM
- Gravação de dados da interface na EEPROM (programador CH431A)

### 30/07/2020

- Servidor das aplicações via Wi-fi funcional
- Primeiras versões funcionais das aplicações milton:p e milton:u
- Saída TV (PAL) funcional (baixa resolução, adaptação de arduino-tvout)

### 27/07/2020

- Inclusão da lista de hardware e software
- Teclado PS/2 em funcionamento
- Sinal VGA monocromático (1 bit), dicromático (2 bit) e 16 cores (4 bits)
- Conexão wi-fi e servidor web funcionais
- Emissão de som
- Configurações básicas por visor embutido OLED
- Memória adicional EPROM para componentes da interface
- Leitura e gravação de cartão Micro SD

### 11/07/2020

- Primeira versão do documento e pesquisa de links de apoio.
- Versão em PDF disponível.

## 8 Licença

Copyright (c) 2020 Hugo Cristo Sant'Anna

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”, to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

*Última atualização: 08/08/2020 - hugo.santanna@ufes.br*