

Tarefas Realizadas – Projeto Miracle Reader – janeiro/2018		
Iniciado pesquisa com componentes e bibliotecas		
	Sigla	Significado
Realizados testes com o tesseract no windows	RPI	Raspiberry PI
Realizado pesquisas com TTS e servidores que provendicam serviço de TTS	TTS	Text-to-Speech
Realizado pesquisa com raspberry PI e sistema linux	S.O.	Sistema operacional
Obtido do raspberry PI e da camera própria para ele	OCR	Optical Character Recognition
Obtido componentes secundários de uso vital para o projeto como a fonte do raspberry PI, cartãoSD, case e LCD de 3.2 polegadas	GTTS	Google TTS
Instalado o NOOBS no cartão SD pelo Windows e plugado na RPI para a instalação online do S.O. raspibian	GPIO	Pinos de I/O da RPI
Realizados estudos com o uso a S.O. do raspibian	GUI	Graphics User Interface
Instalado e adaptado o IP fixo no RPI (para uso facilitado do putty) e instalado e configurado tela de controle por acesso remoto utilizando o tightVnc		
Dado inicio aos estudos ao python		
Testes realizados usando o modelo de OCR tess4py e TTS utilizando o GTTS descartando o uso do windows e de algum servidor de TTS privado, agora sendo possível todo o processo sendo feito unicamente na RPI		
Iniciado a produção dos arquivos de programação 'definitivos' do projeto onde é criado o main.py que possuem operações distintas para interpretar a imagem para texto (ocr.py) e converter este texto para audio (tts.py) e reproduzir o audio (audio.py)		
Realizado estudo para tratar a imagem e reduzir a margem de erros na interpretação do OCR, a biblioteca utilizada é o openCV, criado a classe image.py onde transforma a imagem original em escala cinza para poder encaminhar para o ocr.py interpretar		
Desenvolvido o algoritmo de processamento geral com o arquivo de nome process.py, onde tem a função de juntar sequencialmente todas as outras operações e realizar todo o processo do sistema em um único comando		
Dado inicio aos estudos para o desenvolvimento de uma versão grafica do projeto, também chamado de GUI sendo inicialmente chamado apenas pelo modo bash (no caso pelo main.py), a ferramenta em que o projeto é desenvolvido é o PyQt		
Desenvolvido as telas guiHome.py, guiCamera.py (não funcional até o momento), guiImage.py, guiOcr.py, guiTts.py, e guiAudio.py, que continuam a utilizar as funcoes do bash porém sem a intervenção do main.py após ser iniciado pelo gui.py		
Feito o estudo e a instalação da tela LCD de 3.2 polegadas no RPI, até o momento é feito a conexão direta junto aos pinos GPIO da RPI,		
Configurado uma conexão direta a tela 0 do linux (a tela original), pelo x11vnc e conectado pelo tightvnc podendo agora mexer na tela de 3.2" diretamente pelo computador facilitando a execução de scripts de configuração e testes do sistema		
Readaptado o sistema de GUI na tela LCD fazendo com que o tamanho e a resolução fiquem ideiais para o uso touchscreen		
Iniciando estudos com a camera do raspiberry PI, criando o cam.py responsavel por capturar a imagem disponivel apenas para a versão bash, o cam.py também foi acoplado a lista de tarefas do process.py		
Realizado a implementação do modulo de captura de camera para o modo grafico, sendo este utilizando o capturecam.py uma nova biblioteca responsavel exclusivamente de capturar a imagem pelo modo grafico		
Desenvolvimento do modo de acessibilidade, onde é possível apenas controlar por botões físicos e guiado por voz		
Projeção de uma base para o apoio da camera para maior facilidade na captura das fotos, dois modelos em planejamento		
Verificar disponibilidade de adicionar um led de flash para facilitar na captura da foto		
Melhorar o tratamento da imagem, deixando além da escala cinza (grayscale), aumentar o constraste para que fique apenas em preto e branco, podendo distinguir melhor as letras		
Modo de leitura em livro por blocos		