

UFRN - PPG EMECA

FLUXO E METODOLOGIAS DE PROJETO DE
SISTEMAS EMBARCADOS

Sistemas de controle

Com ESP32

Deivison Luan

Introdução

O presente trabalho busca auxiliar o ensino de sistemas de controle

- Solução de baixo custo, fácil aplicação e acessível de qualquer lugar.
- Desenvolvida baseada em um microcontrolador de simples acesso.
- Demonstra na pratica conceitos importantes sobre a matéria, como amortecimento, overshoot, respostas para primeira e segunda ordem e diferenças entre malhas abertas ou fechadas

Problema de negócio

Existe uma lacuna no mercado de kits educacionais para o ensino de disciplinas de engenharia. Alguns kits são muito caros, alguns tem funcionalidades limitadas e alguns tem pouca mobilidade ou dificultam o acesso do aluno.

Focando na disciplina de sistemas de controle, e tentando sanar essas deficiências, o produto desenvolvido aqui é uma plataforma de baixo custo, acessível de qualquer lugar e com funcionalidades que ensinem os alunos sobre conceitos de sistemas de controle tanto no software quanto em hardware.

Espera-se que o sistema seja adotado por instituições de ensino de engenharia que tenham limitações orçamentarias e precisem de soluções completas.

Requisitos Funcionais

Descreve o que precisa ser feito no projeto.

RF1.

Receber e decodificar comandos (IRL e mensagens) vindos de um cliente web.

RF2.

Aplicar níveis de tensão aos circuitos usando as portas com DAC e avaliar suas respostas com ADC.

RF3.

Enviar informações para o cliente onde essas informações vão atualizar um gráfico e cards.

Requisitos Não-Funcionais

Descreve como deve ser feito o projeto.

RNF1.

Utilização de Websockets como protocolo de comunicação e FreeRTOS para funções de tempo real.

RNF2.

Tempo inicialização máximo: 8 s.

RNF3.

Alimentação 5 V e 2 A.

Regra de negócio

Descreve restrições para a execução do sistema.

RN1.

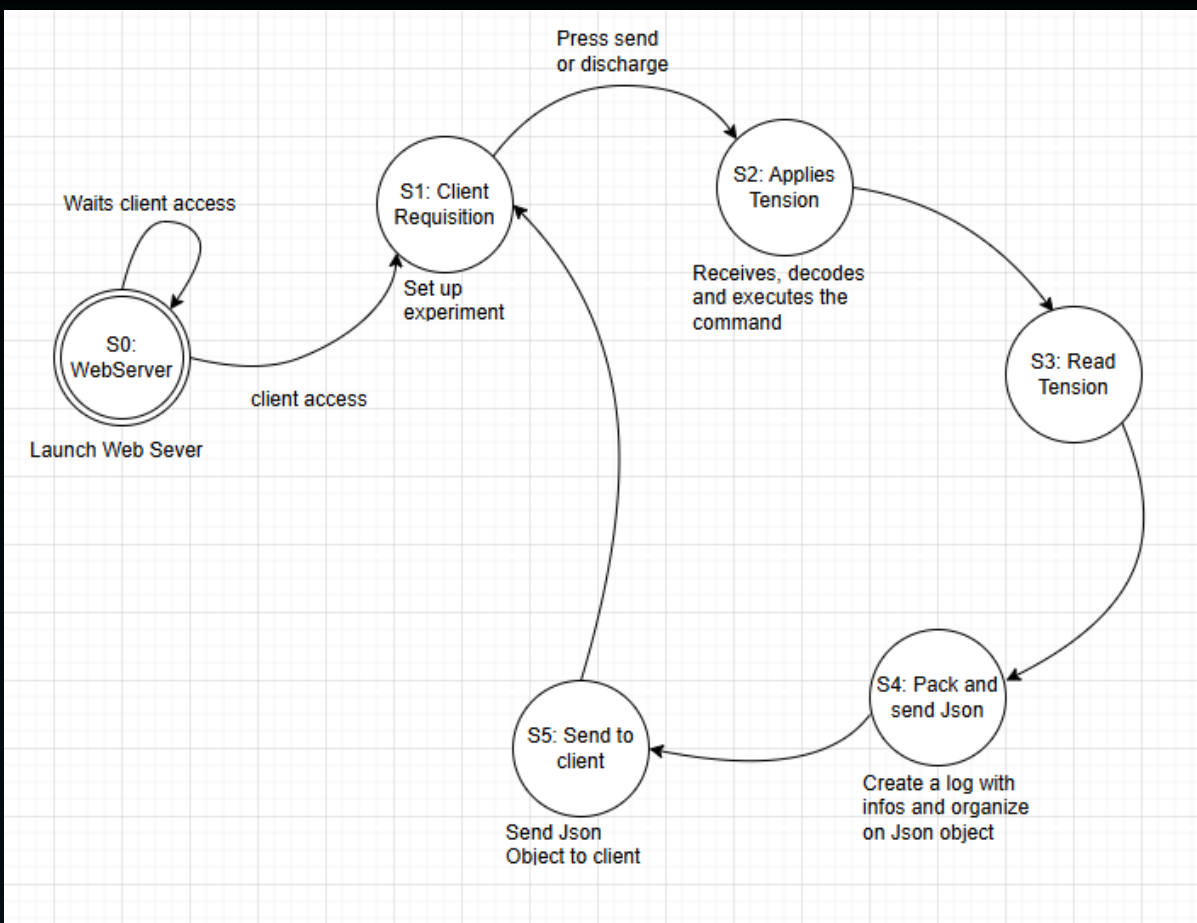
Na tela de interface do cliente o botão “descarregar capacitor” serve como um reset no experimento.

RN2.

O botão “Aplicar parâmetros” inicia o experimento com as configurações escolhidas acima.

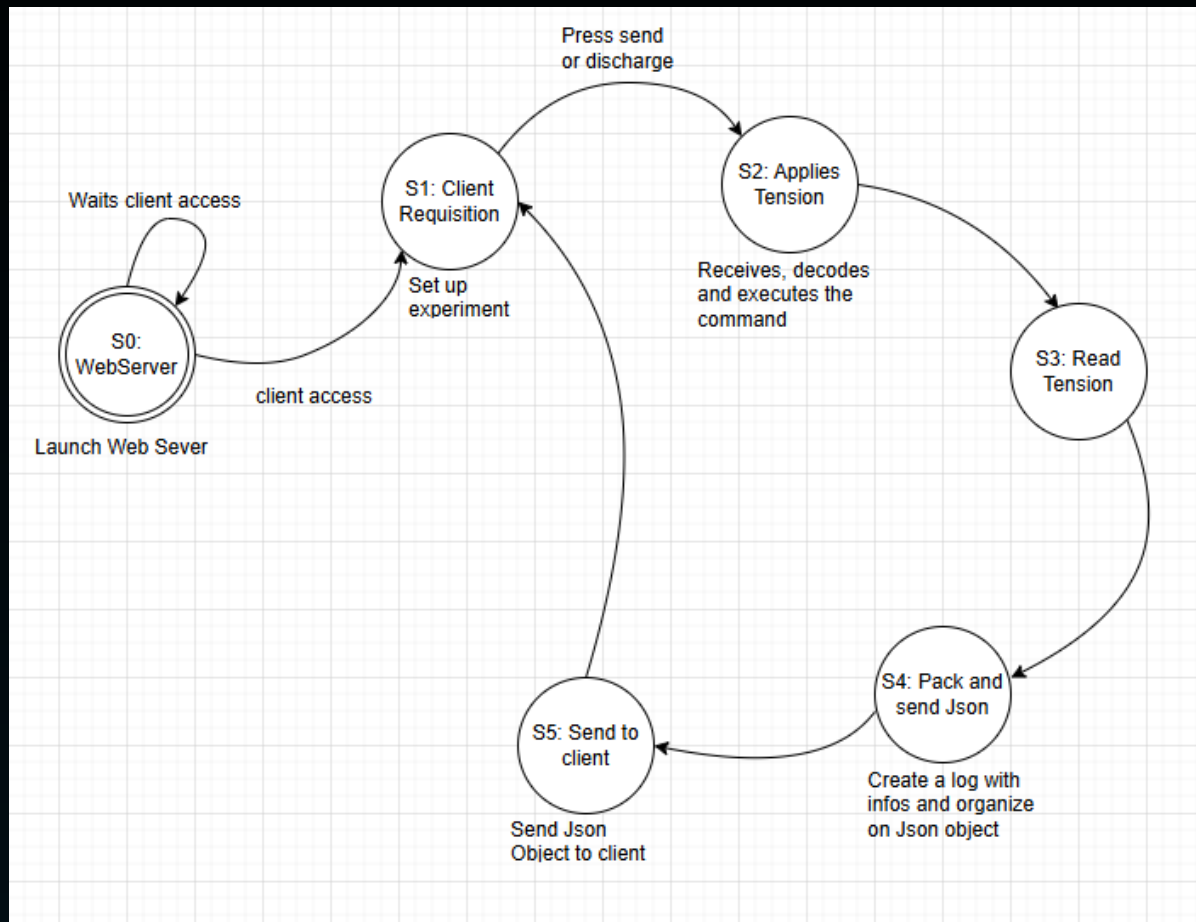
Atenção: Um indicador ao seu lado mostra quando o capacitor está sendo descarregado (vermelho) e quando está em uso normal (verde).

Casos de Uso (CSU0x)



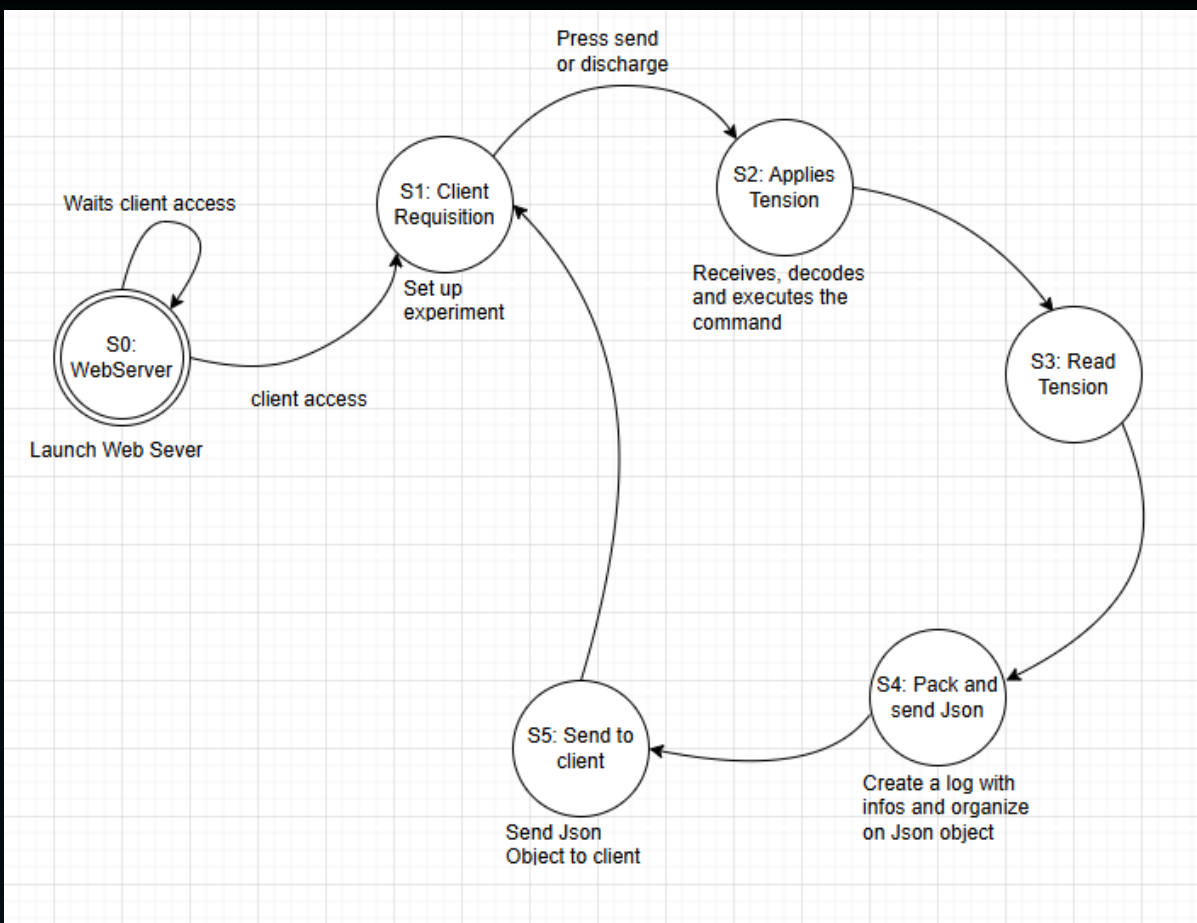
Esp32 como servidor – O esp32 está hospedando sendo utilizado como servidor web e como controlador do experimento. Ao ser energizado, o Esp32 está pronto para receber o acesso.

Casos de Uso (CSU0x)



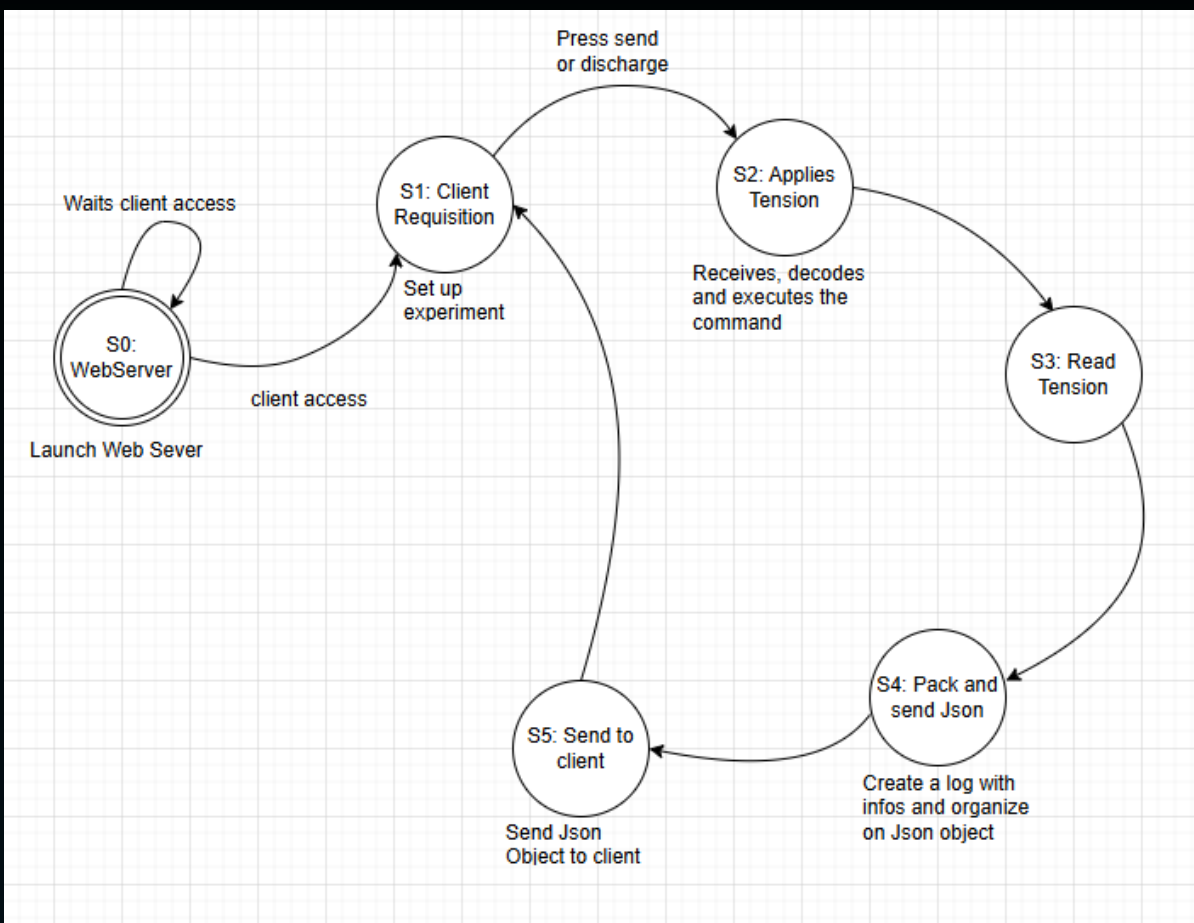
Configuração do cliente – Na página web, o usuário pode selecionar como o sistema vai se comportar, selecionando opções como qual planta será utilizada, qual ordem, se o sistema está em malha aberta ou fechada e os parâmetros no caso de uso de PID.

Casos de Uso (CSU0x)



Envio de parâmetros (Comunicação cliente-servidor) –
O servidor recebe as mensagens enviadas pelo cliente e executa o que for preciso, aplicando tensões ou descarregando o capacitor para reiniciar o experimento.

Casos de Uso (CSU0x)

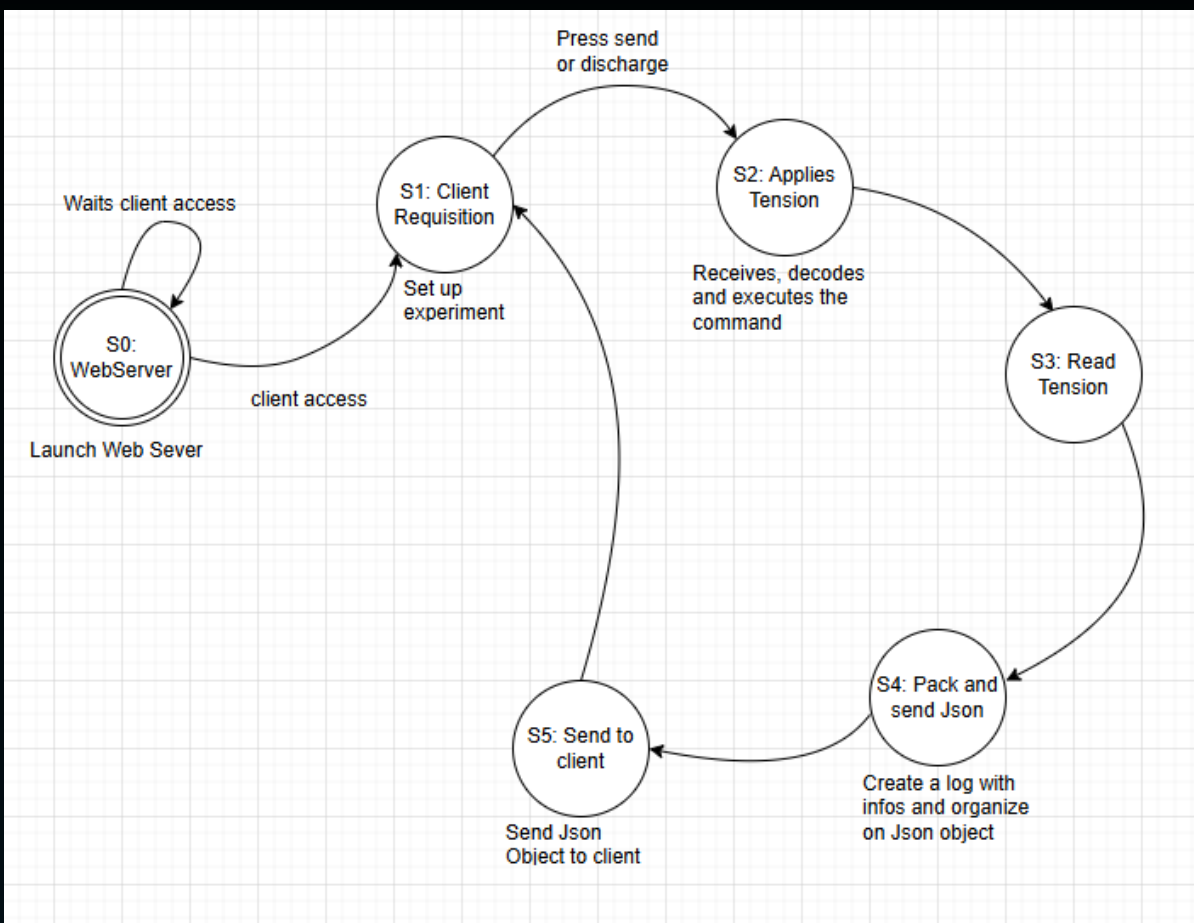


Aplicação nos circuitos – O Esp32 aplica a tensão indicada pelo Set Point (SP) e mede a tensão no capacitor como sua resposta.

Sistema de primeira ordem – Circuito RC com capacitor eletrolítico fixo em 100 uF e 4 resistores (1 kΩ, 10 kΩ, 22 kΩ e 56 kΩ) selecionáveis por um demux alterando o tempo de resposta do sistema.

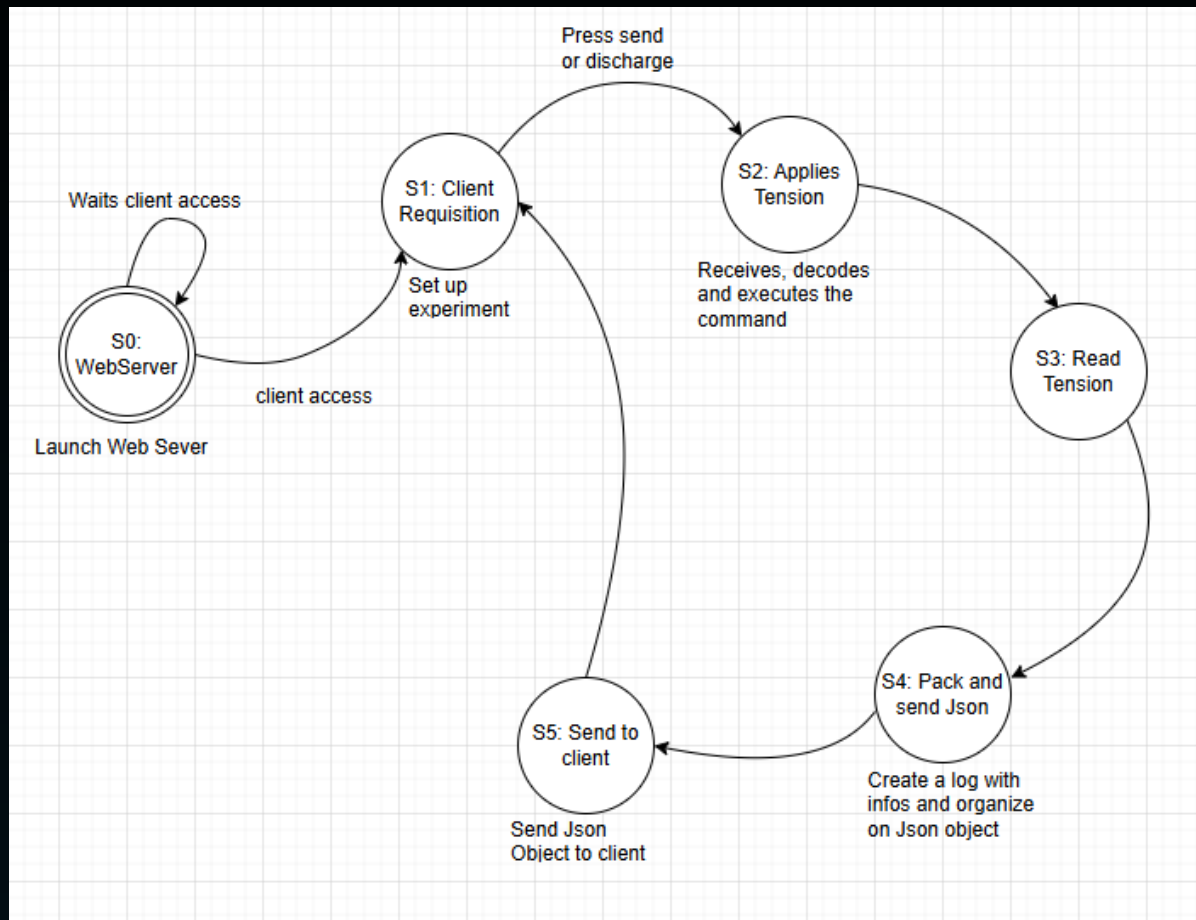
Sistema de segunda ordem – Circuito de um filtro passa-baixa ativo com topologia Sallen-Key (figura 1) com capacitores eletrolítico (4.7 uF e 10 uF) selecionáveis por um demux alterando a forma de resposta do sistema.

Casos de Uso (CSU0x)



Avaliação – Medir a tensão de saída dos circuitos (tensão nos capacitores utilizados) e organiza em um objeto JSON que vai ser enviado para o cliente web.

Casos de Uso (CSU0x)

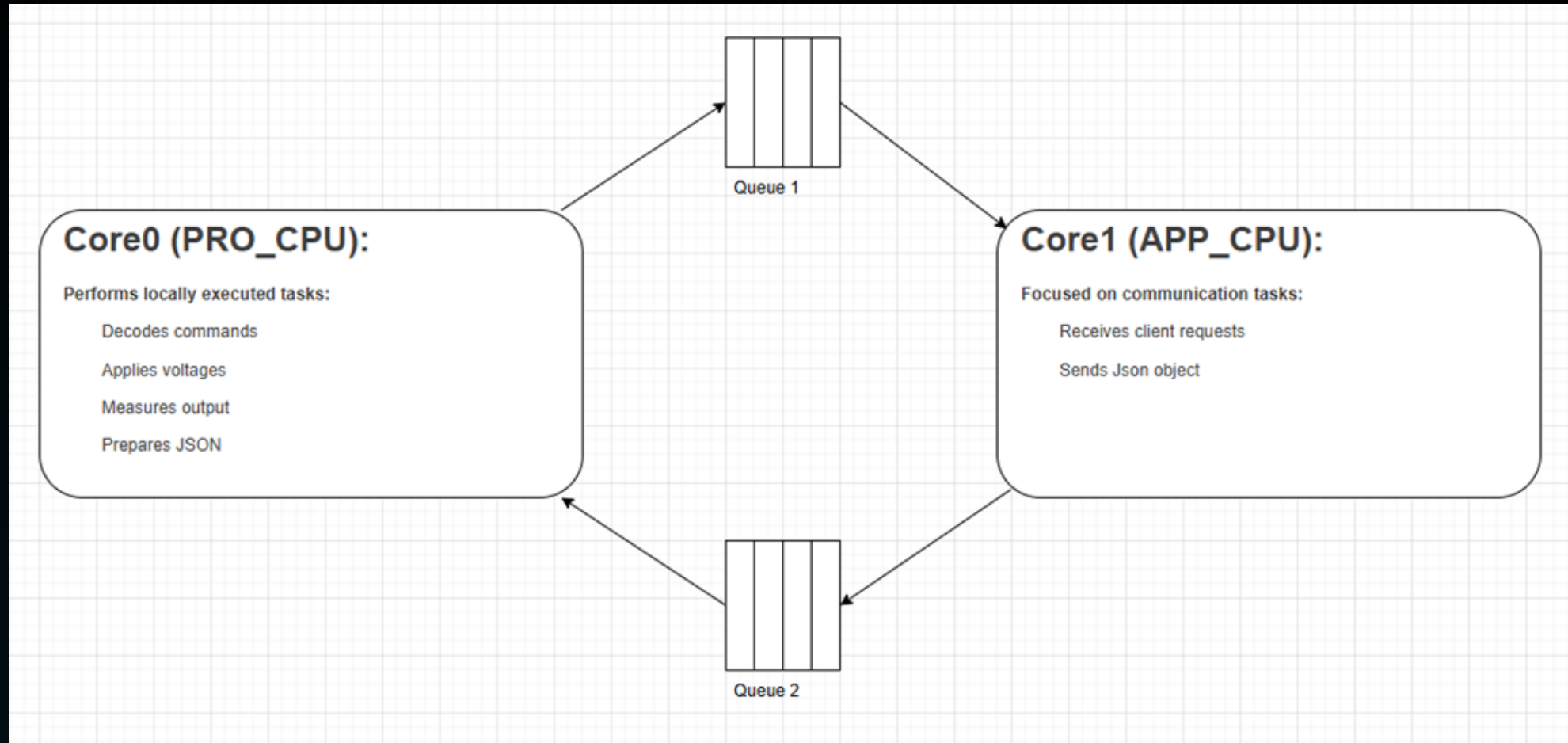


Exibir os resultados – Na interface do cliente, os resultados são mostrados em flashcards e em um gráfico.

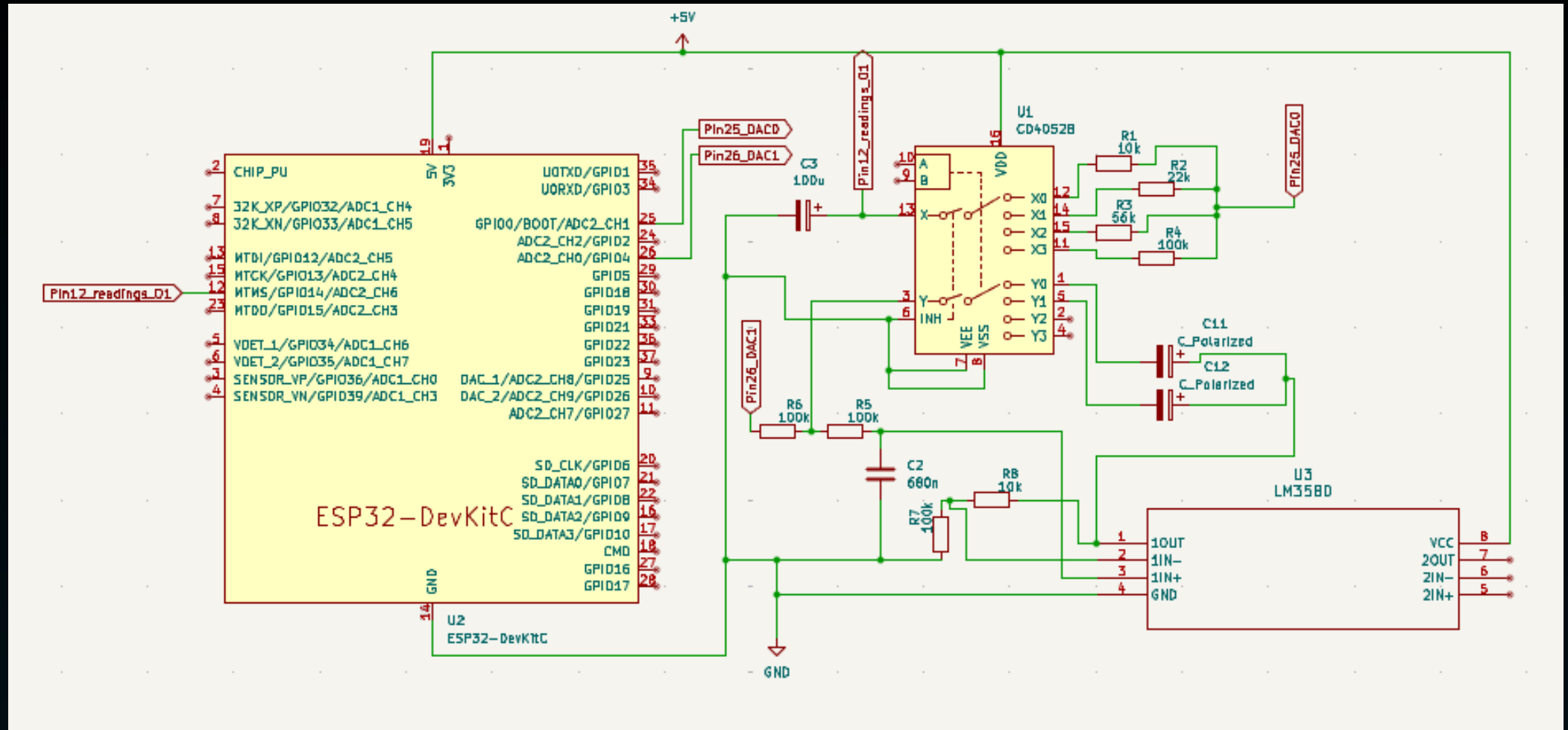
Interface Cliente web



Estrutura interna



Esquemático da PCB



Lista de materiais e orçamento

Categoria	Item	Quant.	Preço unitário	Total
Microcontrolador	ESP32 DEV-KIT	1	R\$ 61,00	R\$ 61,00
Componentes	Capacitor eletrolítico 4.7 uF	4	R\$ 0,30	R\$ 1,20
	Capacitor eletrolítico 10 uF	1	R\$ 0,15	R\$ 0,15
	Capacitor eletrolítico 100 uF	1	R\$ 0,20	R\$ 0,20
	Capacitor ceramico 680 nF	1	R\$ 0,85	R\$ 0,85
	Resistores de 10 kΩ	2	R\$ 0,07	R\$ 0,14
	Resistores de 22 kΩ	1	R\$ 0,45	R\$ 0,45
	Resistores de 56 kΩ	1	R\$ 0,07	R\$ 0,07
	Resistores de 100 kΩ	4	R\$ 0,50	R\$ 2,00
	Ampop LM358	1	R\$ 1,40	R\$ 1,40
	Mux/Demux 4052 BE	1	R\$ 1,30	R\$ 1,30
Energia	Fonte de alimentação 220 v - 5 v	1	R\$ 35,00	R\$ 35,00
	Bornes de duas conexões	2	R\$ 0,80	R\$ 1,60
	Ventoinha	1	R\$ 20,00	R\$ 20,00
Estrutura e Acabamento	Carcaça impressa em 3D (ou acrílico/ABS)	1		R\$ -
	Parafusos M2 ou M3	4		R\$ -
Ferramentas	Ferro de solda + Estanho 60/40	1		R\$ -
	Multímetro	1		R\$ -
	Impressora 3D / Serviço de corte a laser	1		R\$ -
			Total	R\$ 125,36

Repositório e bibliotecas

https://github.com/DeivisonLuan/Esp32_Control_Systems#

- **Arduino_JSON**: Processa JSON em projetos com microcontroladores Arduino e derivados.
- **WebSockets**
- Protocolo WebSockets para Arduino (Server + Client).

Author:	Arduino
Website:	https://github.com/arduino-libraries/Arduino_JSON
Category:	Other
License:	LGPL 2.1
Library Type:	Official
Architectures:	Any

Author:	Markus Sattler
Website:	https://github.com/Links2004/arduinoWebSockets
Category:	Communication
License:	LGPL 2.1
Library Type:	Contributed
Architectures:	Any



Obrigado

Deivison Luan

Deivison.luan@live.com