

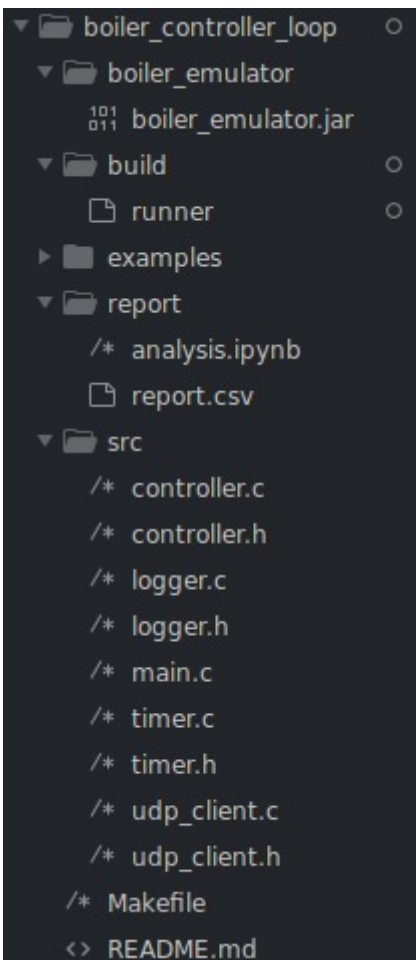
T1 - Trabalho do controlador de sistemas contínuos sequencial e tempo real

- Andre Martins Pio de Mattos (16103374)
- Lukas Alberto Belck (18150497)

```
import pandas as pd
import plotly.graph_objects as go
```

Programa

- Setup (exemplo: makefile)
- Abstrações (exemplo: controller, timer, logger, udp_client)
- Confiabilidade (exemplo: socket com timeout, reset socket com erro)
- Previsibilidade (exemplo: temporização com nanosleep, margem entre execuções)

[+ Code](#)[+ Text](#)

```

void main(void) {

    task_init();

    /* Main loop (cyclic executive) */
    while(1) {

        /* 30ms, each cycle: Periodically run the controllers */
        task_controller();

        /* 1s, each TICKS_TO_1_SECOND cycles: Periodically prints user information */
        if(!(cycle%TICKS_TO_1_SECOND)) {
            task_user_info();
        }

        /* 30s, after buffer size achieved: Periodically save the logger session */
        if((cycle%FILE_BUFFER_SIZE+1) >= FILE_BUFFER_SIZE) {
            task_logger_file();
        }

        /* Increment cycle counter variable */
        cycle++;

        /* Delay until next system loop cycle (system tick: 30ms) */
        timer_delay(SYS_TICK);
    }
}

```

▼ Resultados

```

NANOSEC_PER_MICRO = 1000
response_times = pd.read_csv('report.csv')
response_times.columns = ['response_times_ns']
rtmis = response_times.response_times_ns / NANOSEC_PER_MICRO
response_times.head()

```

	response_times_ns
0	1257696.0
1	2006405.0
2	3043844.0
3	3015044.0
4	2818172.0

```

fig = go.Figure(data=[go.Histogram(x=rtmis)])
fig.update_layout(
    title="Distribuição dos tempos de resposta para controle da caldeira",
    xaxis_title="Tempo de resposta [μs]",
    yaxis_title="Contagem",
)
fig.show()

```

Distribuição dos tempos de resposta para controle da caldeira



```
print(f'Numero de amostras: {response_times.shape[0]}')
print(f'Media: {rtmis.mean()} μs')
print(f'Mediana: {rtmis.median()} μs')
print(f'Desvio padrao: {rtmis.std()} μs')
print(f'Max: {rtmis.max()} μs')
print(f'Min: {rtmis.min()} μs')
```

```
Numero de amostras: 22999
Media: 904.5328950389144 μs
Mediana: 812.804 μs
Desvio padrao: 747.7501062713792 μs
Max: 75798.16 μs
Min: 315.818 μs
```

Podemos notar pelo histograma e pelas estatísticas que na média possuímos uma margem de mais de 90%. Portanto, seria possível reduzir o período de cada ciclo de execução.

```
fig = go.Figure(data=[go.Histogram(x=rtmis)])
fig.update_layout(
    title="Distribuição dos tempos de resposta para controle da caldeira com zoor",
    xaxis_title="Tempo de resposta [μs]",
    yaxis_title="Contagem",
)
fig.show()
```

Distribuição dos tempos de resposta para controle da caldeira com zoom r

