

**INSTITUTO
FEDERAL**
Santa Catarina

Câmpus
São José

Relatório técnico do eLab

Interface de Gerência

Aluno
João Pedro

28 de fevereiro de 2024

Sumário

Resumo	2
Sistema de Gerência	3
Diagrama de blocos	3
Diagrama do sistema em si	3
Propósito	4
Dependências	4
Instalação	4
NodeJS	4
MySQL	4
Servidor MQTT	4
O sistema	4
Execução	4
Acesso	4
Descrição	4
Servidor	4
API	5
Placas controladoras	5
Registro	5
Controle	5
Gestão de usuários	5
Placas FPGA	5
Conclusão	6

Resumo

Este documento visa explicar o desenvolvimento do sistema de gerência para o projeto “FPGA eLab: Plataforma de Desenvolvimento Avançado para Ensino e Pesquisa”. Para que a implementação possa ser expandida de forma facilitada, foi criada uma API, escrita na linguagem de programação para a *Web* JavaScript e foi utilizado o servidor NodeJS para disponibilizar a API. O sistema comunica-se com uma placa controladora, que por sua vez controla uma placa FPGA. Para tal comunicação, foi desenvolvido um protocolo simples que associa ao pino da placa controladora com a ação a ser realizada naquele pino. Além disso, o sistema permite o cadastro de múltiplas placas controladoras, de forma que, por consequência, é possível controlar múltiplas placas FPGA. Para esse cadastro, foi criado um pequeno mecanismo que garante que a placa foi devidamente registrada no sistema, utilizando o sistema *pare-espere*.

Palavras-chave: NodeJS. MQTT. RESTFull. API.

Sistema de Gerência

Componentes do sistema que integra a placa controladora, monitoramento por vídeo, programação da FPGA e ambiente virtual de desenvolvimento.

Diagrama de blocos

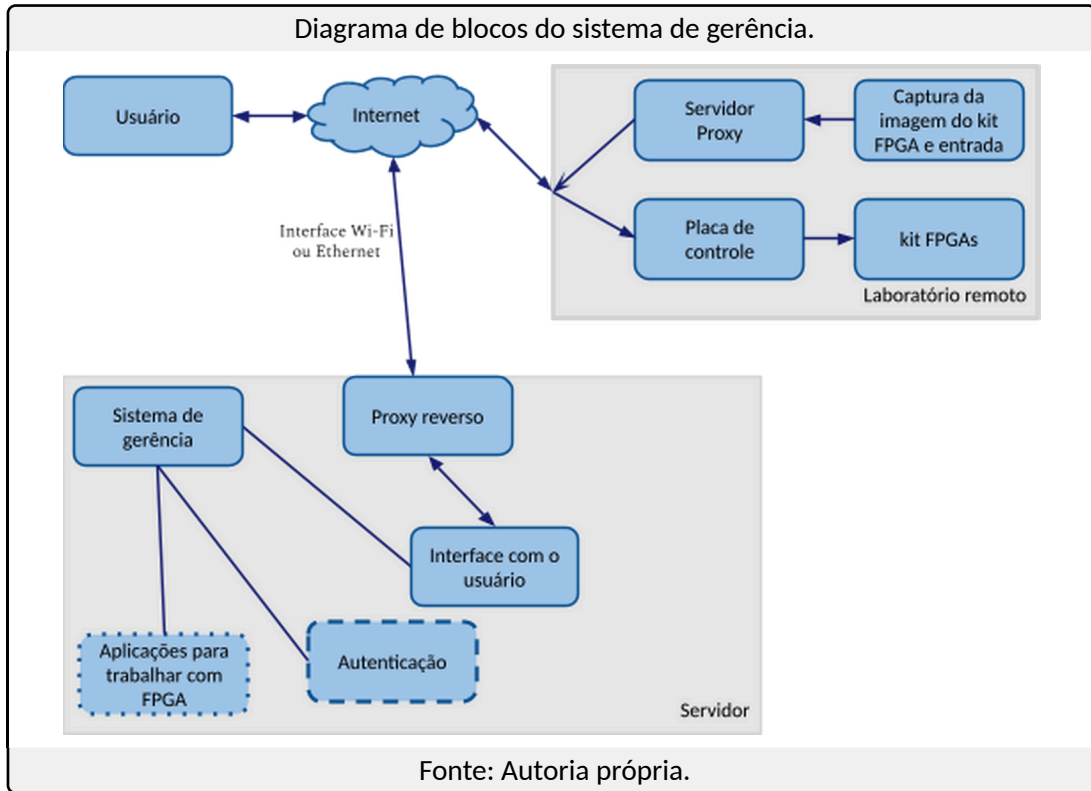
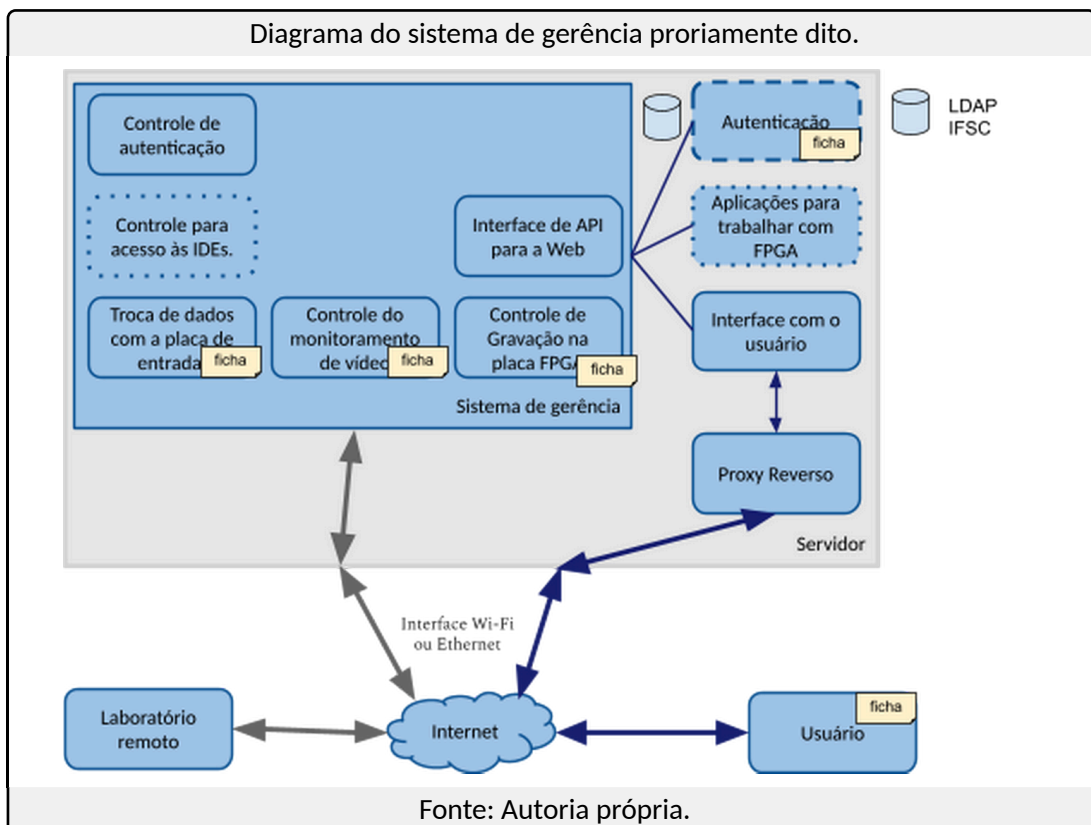


Diagrama do sistema em si



Propósito

O sistema de gerência permite ao usuário utilizar os componentes do ambiente de desenvolvimento, programação e testes de forma que os resultados possam ser monitorados em tempo real.

Dependências

- NodeJs (18 ou 20)
- NPM
- MySQL
- Servidor MQTT

Instalação

NodeJS

- Instalação no OpenSUSE Tumbleweed:
 - NodeJS 20:

```
sudo zypper install nodejs
```
 - NodeJS 18 (via OPI):

```
opi nodejs18
```
 - Escolha a opção 1:

```
1. nodejs18
```
 - Escolha novamente a opção 1:

```
1. devel:languages:nodejs
```
- Instalação no Debian 12 (NodeJS 18):

```
sudo apt install nodejs npm
```

MySQL

A instanciação do servidor MySQL pode ser realizada utilizando as instruções do repositório “[banco-de-dados](#)”.

Servidor MQTT

A instanciação do servidor MQTT pode ser realizada utilizando as instruções do repositório “[mqtt](#)” do projeto.

O sistema

Execução

Para executar o sistema, dentro do diretório da aplicação, execute o comando:

```
npm run dev
```

É preciso que os servidores MQTT e MySQL estejam em execução.

O serviço é recarregado em tempo real caso haja alguma modificação em alguns dos seus arquivos.

Acesso

O sistema pode ser acessado através do endereço: <http://localhost:3000>. Uma página web simples é disponibilizada para testes com os botões.

Descrição

Servidor

O sistema utiliza o NodeJs para compor o servidor e disponibilizar a API para acesso das funcionalidades. A API utiliza o padrão RESTFull e é implementada através do módulo Express.

API

No atual estágio de implementação, é possível utilizar a API para interagir com as placas controladoras registradas no servidor. No arquivo "[index.html](#)" é possível ter uma demonstração de uso de diferentes tipos de botões e selecionar diferentes placas que estejam cadastradas.

Placas controladoras

Registro

As placas controladoras são cadastradas por mensagem MQTT, através do tópico "eLab/register". Quando a placa é ligada, esta envia um mensagem em formato JSON com as informações de identificação da placa e o nome da mesma, conforme mostrado abaixo:

```
1 {  
2   "id": "c9efcd6e",  
3   "name": "ESP-12E"  
4 }
```

No repositório da placa controladora é mostrado como esse identificador "id" é gerado. O nome atribuído à placa também é definido pelo usuário no momento da compilação.

Assim que o servidor recebe a mensagem, os dados informados no JSON são processados e armazenados em uma tabela do tipo chave/valor, cuja chave é o identificador da placa e o valor é o nome da mesma. Essas informações também são utilizadas no menu em cascata da página de testes.

O servidor responde com uma mensagem no tópico contendo o identificador da placa, nesse exemplo, o tópico seria "eLab/c9efcd6e". A mensagem de resposta é literalmente a palavra "ACK". No lado da placa controladora, a mensagem é aguardada por dez tentativas e, caso a mensagem de confirmação não seja recebida, a placa reenvia a mesma.

Controle

O controle também é realizado via mensagem MQTT. A mensagem é enviada no tópico contendo o identificador da placa e possui o formato "Numero_do_Pino:Comando". Os comandos disponíveis são:

- 0: Pino em nível baixo;
- 1: Pino em nível alto;
- 2: Pino em alta impedância.

O sistema não espera mensagem de confirmação. O valor do pino é um número natural e sequencial que é abstraído pela placa controladora. Dessa forma, o sistema não possui a identificação do pino físico em si, mas de sua representação. Dessa forma, abaixo é exemplificada a mensagem de controle dos pinos:

0:1 -> Pino mapeado em 0 é posto em nível alto;
0:0 -> Pino mapeado em 0 e posto em nível baixo;
0:2 -> Pino mapeado em 0 é configurado como alta impedância.

O número dos pinos é mapeado internamente pela Placa Controladora e, portanto, são abstraídos para o sistema. Isso permite o controle mais genérico, já que cada tipo de placa possui sua própria pinagem.

Gestão de usuários

Na atual implementação, não há controle de usuários funcional.

Placas FPGA

É possível listar as placas FPGAs disponíveis. Isso dá-se pelo comando `jtagconfig`, que deve estar configurado para que os servidores estejam cadastrados no mesmo. A ferramenta é limitada em apenas aceitar conexões do localhost, sendo necessário que a porta do servidor remoto seja mapeada localmente. Uma forma de atingir esse objetivo é utilizando o comando `ssh`:

```
ssh -L 1309:localhost:60001 usuario@servidor
```

O comando acima mapeia a porta 1309 do servidor remoto para a porta 60001 do endereço local. O `jtagconfig` permite configurar servidores JTAG no módulo `jtagd`.

Conclusão

Embora a implementação careça de certos recursos, como um controle de usuários, a prova de conceito realizada permite o controle de uma placa FPGA remotamente, de modo que seja possível expandir o sistema a fim de possuir os recursos faltantes.

Desenvolver esse projeto exigiu que a comunicação entre o servidor e a placa de controle fosse confiável e que os comandos enviados fossem bem implementados para que não houvesse erros.