

# FPGA eLab: Plataforma de Desenvolvimento Avançado para Ensino e Pesquisa

RESUMO ESTENDIDO - Disciplina de TCC029009

**João Pedro Menegali Salvan Bitencourt**

Estudante do Curso de Engenharia de Telecomunicações

**Marcos Moecke**

Professor orientador

Semestre 2024.1

**Resumo-** Este trabalho tem como objetivo propor um sistema que provê uma plataforma de desenvolvimento FPGA, permitindo o acesso à diversos kits de placas de forma remota. A proposta surgiu em decorrência do alto custo desses equipamentos, além da complexidade da instalação e manutenção dos ambientes integrados de desenvolvimento (IDE) utilizados na criação dos projetos e na programação dos mesmos. Adicionalmente, permitirá ao usuário monitorar e controlar a placa programada em tempo real. A gestão de usuários permitirá diferentes níveis de acesso, sendo que o administrador poderá visualizar e interagir com os experimentos dos utilizadores menos privilegiados. Em ambiente acadêmico, estudantes terão acesso a um amplo conjunto de ferramentas e kits sem investimento pessoal e sem estar restrito ao horário de aula. Já o professor, como administrador, poderá monitorar e auxiliar nas atividades de forma facilitada, além de poder propor projetos que outrora seriam inviabilizados pela falta de equipamentos.

**Palavras-chave:** FPGA. Plataforma remota. Sistema de gerência.

## 1 Introdução

O uso de Dispositivos Lógicos Programáveis (DLP) está presente em diversas aplicações da tecnologia atual, como sistemas embarcados, processamento de imagens, computação em nuvem, entre outros. Além disso, os DLP são normalmente integrados à sensores e atuadores, que permitem a coleta de uma grande quantidade de dados em tempo real, estando presentes em sistemas automotivos e médicos e dispositivos inteligentes, por exemplo. Além disso, esses componentes podem ser utilizados no meio acadêmico. Segundo Batchelder e Meliksetian (1993), as mesmas vantagens proporcionadas para engenheiros que atuam na indústria podem ser utilizadas no ensino de estudantes de graduação da área, cuja maioria dos projetos envolvem circuitos complexos montados em matrizes de contato, que são difíceis de serem

depurados e testados. Além disso, a utilização de ferramentas para projetos utilizando DLP permite que o estudante crie a maior parte do circuito em um único chip e verifique o funcionamento utilizando um simulador antes de ir de fato para a implementação física.

Entretanto, kits de desenvolvimento para *Field Programmable Gate Array* (FPGA) possuem alto custo, sendo necessário investimento por parte do estudante ou da instituição de ensino. Além disso, a instalação e manutenção dos ambientes integrados de desenvolvimento (IDE) utilizados na criação dos projetos e na programação dos mesmos são complexas e podem ser um obstáculo para estudantes iniciantes. Além disso, pode haver problemas de compatibilidade de *hardware* e *software*, o que pode dificultar ainda mais o processo de instalação. A programação desses DLP é realizada através de *softwares* como Quartus Prime da Intel (Intel, 2024), Vivado Design Suite da AMD (AMD, 2024), além de outras opções mais genéricas como o GHDL (GHDL, 2017), Icarus Verilog (Stephen Williams, 2019), Matlab/Simulink (MathWorks) quando associado ao toolbox HDL Coder (MathWorks, 2024) e LabVIEW (National Instruments) (National Instruments, 2024).

Uma forma de facilitar o acesso dos estudantes tanto ao *hardware* quando aos ambientes integrados de desenvolvimento, é disponibilizar esse recurso através de uma plataforma em nuvem, que permitirá o acesso a partir de qualquer dispositivo com acesso à Internet e um navegador. Além de que, com o controle de usuário capaz de definir diferentes níveis de acesso, o professor poderá monitorar e auxiliar nas atividades de forma facilitada, além de poder propor projetos que outrora seriam inviabilizados pela falta de equipamentos ou *software* especializado. Ademais, o estudante não ficará limitado à disponibilidade presencial dos recursos, com a possibilidade de exercer seus estudos no horário que mais lhe convier.

O sistema de gerência será desenvolvido de forma a permitir a concentração de kits de desenvolvimento FPGA oriundos de diferentes localidade físicas. Isso permitirá que instituições compartilhem equipamentos entre si diminuindo o custo de ensino para disciplinas que podem beneficiar-se dos mesmos. Além disso, possuirá uma interface Web para controlar botões e chaves deslizantes das placas, de forma a replicar a forma de interação que ocorreria fisicamente.

Como objetivo geral, será desenvolvido um sistema de gerência que integrará uma interface Web com diferentes laboratórios físicos com as placas de desenvolvimento. Já como objetivos específicos, serão desenvolvidos módulos para a gestão de usuários, monitoramento e controle remoto das placas de desenvolvimento, disponibilização de ambientes integrados de desenvolvimento para a criação dos projetos que serão gravados nas placas.

## **2 Metodologia**

Para a execução do trabalho proposto, foram estabelecidas as seguintes etapas:

### **2.1 Definição de objetivos**

Foram definidos os objetivos geral e específicos do trabalho, que foram apresentados na seção anterior, sendo o enfoque na construção de um sistema base e modular que possa ser expandido para atender os quesitos restantes em possíveis trabalhos futuros.

## 2.2 Organização do sistema

Toda a plataforma está organizada em servidores com propósitos diferentes de abstração:

- Servidor central: responsável por gerenciar os servidores de FPGAs e de mídia, o controle de usuários, além de prover uma API para comunicação com a interface Web;
- Servidor de FPGAs: responsável por abstrair e gerenciar o acesso às placas de desenvolvimento FPGA, permitindo a programação e controle remoto das mesmas;
- Servidor de mídia: responsável pelo roteamento da imagem em tempo real de acordo com a localização do usuário, de tal forma que o fluxo de vídeo não precise passar pelo servidor central. Isso reduz o tráfego gerado e permite usuários que estejam na mesma rede em que o laboratório está situado, possam experimentar respostas visuais com menor latência.

## 2.3 Tecnologias utilizadas

Para a criação da API (*Application Programming Interface*) será utilizado o NodeJS, sobre o qual serão concebidos módulos para interagir com o banco de dados, e com os servidores de FPGAs e de mídia. Estes também conterão APIs que serão desenvolvidas utilizando a mesma tecnologia.

No servidor de FPGAs, haverá um módulo para comunicação utilizando o protocolo MQTT (*Message Queuing Telemetry Transport*) que será o responsável enviar comandos à uma placa controladora, que, por sua vez, estará conectada fisicamente a um *kit* de desenvolvimento FPGA, através dos pinos GPIO (*General Purpose Input/Output*).

No quesito infraestrutura, pretende-se utilizar contêineres Docker ou Podman com o objetivo facilitar a distribuição dos componentes da plataforma, já que toda a aplicação será encapsulada com todas as dependências necessárias para sua execução, que permitirá com que se haja a padronização do ambiente de execução.

Por fim, para a criação dos arquivos necessários para a programação dos *chips* de FPGA, serão utilizados alguns componentes da ferramenta Quartus Lite, para a verificação de erros e compilação do código das linguagens VHDL e Verilog. Essa etapa está contida no servidor de FPGAs.

## 3 Conclusão

A plataforma de ensino e desenvolvimento proposta permitirá o acesso à recursos de *hardware* outrora escassos para estudantes e professores. Além disso, as instituições de ensino poderão cooperar entre si compartilhando recursos, como os *kits* de FPGA. Isso possibilitará a realização de projetos que seriam inviabilizados pela falta de equipamentos ou *software* especializado. A plataforma será desenvolvida de forma modular, permitindo a expansão e a adição de novos recursos sem a necessidade de refatoração das implementações já realizadas.

Além disso, a estruturação proposta visa não atribuir demasiadas responsabilidades a um único servidor, facilitando a manutenção de todo o sistema e detecção de erros, já que é possível validar cada servidor de forma separada.

## Referências

AMD. *Vivado Design Suite*. 2024. Online. Disponível em: <<https://www.xilinx.com/products/design-tools/vivado.html#overview>>. Acesso em: 2 de abr. 2024.

BATCHELDER, M.; MELIKSETIAN, D. Programmable logic devices in undergraduate digital design courses. In: . Tuscaloosa, AL, United states: [s.n.], 1993. p. 416 – 417. Design practice;Design-automation tools;Digital design course;Logical operations;Programmable logic device;Senior Design project;Single chips;. Disponível em: <<http://dx.doi.org/10.1109/SSST.1993.522814>>.

GHDL. *GHDL*. 2017. Online. Disponível em: <<http://ghdl.free.fr>>. Acesso em: 2 de abr. 2024.

Intel. *Intel® FPGA Academic Program Teaching Materials*. 2024. Online. Disponível em: <<https://www.intel.com/content/www/us/en/developer/topic-technology/fpga-academic/tools.html>>. Acesso em: 4 de abr. 2024.

MathWorks. *HDL Coder*. 2024. Online. Disponível em: <<https://www.mathworks.com/products/hdl-coder.html>>. Acesso em: 4 de abr. 2024.

National Instruments. *What is NI LabVIEW*. 2024. Online. Disponível em: <<https://www.ni.com/en/shop/labview.html>>. Acesso em: 4 de abr. 2024.

Stephen Williams. *ICARUS Verilog*. 2019. Online. Disponível em: <<https://steveicarus.github.io/iverilog/>>. Acesso em: 3 de abr. 2024.