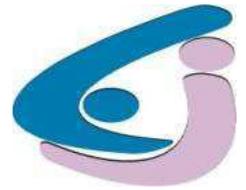




UNIVERSIDADE FEDERAL DE CAMPINA GRANDE - PARAÍBA  
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA  
UNIDADE ACADÊMICA DE ENGENHARIA ELÉTRICA  
PROGRAMA DE PÓS-GRADUAÇÃO DE ENGENHARIA ELÉTRICA



Dissertação de Mestrado

**METODOLOGIA PARA INTEGRAÇÃO DE PROJETOS DIDÁTICOS  
IMPLEMENTADOS EM FPGAs**

PEDRO PAULO MARQUES

Orientadora: Maria de Fátima Queiroz Vieira, PhD.

CAMPINA GRANDE  
2011

PEDRO PAULO MARQUES

**METODOLOGIA PARA INTEGRAÇÃO DE PROJETOS DIDÁTICOS  
IMPLEMENTADOS EM FPGAs**

Dissertação de mestrado submetida à Coordenação dos Cursos de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande – Campus I, como parte dos requisitos necessários para obtenção do grau de Mestre em Ciências no domínio da Engenharia Elétrica. Área de Concentração: Engenharia da Computação.

Orientadora: Maria de Fátima Queiroz Vieira, PhD

CAMPINA GRANDE  
2011

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL DA UFCG

M357m Marques , Pedro Paulo.

Metodologia para integração de projetos didáticos implementados em FPGAs / Pedro Paulo Marques. – Campina Grande, 2011.

104 f.: il.

Dissertação (Mestrado em Engenharia Elétrica) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática.

Orientadora: Profa. Dra. Maria de Fátima Queiroz Vieira.

Referências.

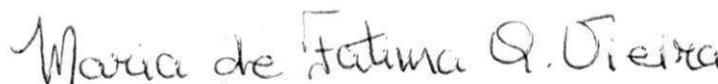
1. Experimentações Metodológicas.
2. Ensino de HDL/FPGAs.
3. Sócio Interacionismo. I. Título.

CDU 004.5 (043)

**METODOLOGIA PARA INTEGRAÇÃO DE PROJETOS DIDÁTICOS  
IMPLEMENTADOS EM FPGAS**

**PEDRO PAULO MARQUES**

Dissertação Aprovada em 30.06.2011

  
**MARIA DE FÁTIMA QUEIROZ VIEIRA, Ph.D., UFCG**  
Orientadora

  
**ELMAR UWE KURT MELCHER, Dr., UFCG**  
Componente da Banca

  
**ALEXANDRE CUNHA OLIVEIRA, D.Sc., UFCG**  
Componente da Banca

  
**JOSÉ EWERTON POMBO DE FARIAS, Dr., UFCG**  
Componente da Banca

CAMPINA GRANDE - PB  
JUNHO -2011

*Dedico este estudo primeiramente à minha mãe por acreditar na ação libertadora através da construção do conhecimento.*

*À minha Avó por me fazer compreender e encontrar um caminho para sair da curiosidade ingênua para uma curiosidade epistemológica.*

*À minha irmã "Nina" pelos constantes apoio e incentivos, reforçando o pensamento da nossa mãe.*

*Às minhas irmãs: Maria por ter dedicado parte de sua vida para cuidar de mim. À Cícera e a Rosa pelos constantes estímulos.*

*À Tânia Maria e Aline, esposa e filha, pelo apoio e incentivos, além da compreensão durante minha ausência.*

*Ao pássaro Curruíra, que da copa das árvores, tantas vezes me acompanhou nos meus estudos pelas madrugadas afora, com o seu canto, lembrando-me que o dia amanhecera e era preciso descansar, e às vezes me alertando que chegara a hora de acordar...*

## **AGRADECIMENTOS**

Aos professores do Departamento de Física da UFPE, Ernesto Raposo e Erivaldo Montarroyos pela confiança e recomendação.

À professora orientadora, Maria de Fátima Queiroz Vieira, pela sabedoria e comprometimento no processo de ensino e aprendizagem.

Aos companheiros do Laboratório de Interface Homem-Máquina LIHM, pelos ensinamentos e contribuições, em especial, a Raffael Carvalho, Gerson Focking, Ademar Virgolino Netto, Flávio Torres, Daniel Cavalcante, Diego Lopes, e Daniel Sherer.

A Meuse Nogueira, pela iniciativa de elaboração e coordenação do projeto Minter/IFPE, e o apoio contínuo.

A Carlos Gomes, analista de sistema da reitoria do IFPE, pelas orientações UML.

Aos amigos do IFPE, Frederico Braga, Zilcio de Lavor, e Paulo Sérgio pelas orientações e ensinamentos.

Aos professores Jorge de Carvalho, Cristina Ramos, e Rogério Arruda, pelo apóio.

Ao amigo e irmão Gilmar Brito, pelo apóio e constantes incentivos.

Ao amigo Moacir Martins Machado, pelo incentivo e reflexões sobre o andamento dos trabalhos.

À Roseane Nascimento, técnica administrativa do nosso departamento no IFPE, *campus* Recife, pelo apoio e atenção.

À Tânia Santos, colaboradora de serviços gerais da coordenação de eletrônica do IFPE, *campus* Recife, pela sua alegria e presteza.

Aos amigos Abner Barros e Wellington Messias pelas orientações técnicas.

Ao colega de curso e trabalho, Luciano Lindoso por compartilhar seus conhecimentos nos momentos de estudos.

Aos companheiros de Curso, José Neves, Marcílio Acioly, Cláudio Delgado, e Sérgio Ferraz pelos constantes incentivos, e momentos de reflexão sobre o andamento dos trabalhos.

Ao companheiro Évio Rocha por compartilhar seus conhecimentos em momentos comuns de estudos, além de disponibilizar seu apoio durante essa jornada.

Aos "ministros dos transportes" Osglay Izídio, no traslado Campina Grande - Recife, e José Neves, no deslocamento entre Recife-Campina Grande.

Ao amigo Osglay Izídio, pelo apoio e incentivo.

Muito obrigado a todos.

*Eu, enquanto homem, não existo somente como criatura individual, mas me descubro membro de uma grande comunidade humana. Ela me dirige corpo e alma, desde o nascimento até a morte... Sou realmente um homem quando meus sentimentos, pensamentos e atos têm uma única finalidade: a comunidade e seu progresso...*

*Infelizmente, a nossa tecnologia excedeu a nossa humanidade.*

Albert Einstein, Físico e humanista.



## RESUMO

O estudo e desenvolvimento de projetos didáticos implementados com FPGAs tornaram-se uma prática muito comum em cursos como Engenharia Elétrica, Ciência da Computação, além dos cursos Técnicos e Tecnólogos. O problema é a falta de uma metodologia que contemple tecnologias de integração associado a um suporte didático-pedagógico. O trabalho objetiva aplicar conceitos oriundos da área de educação ao ensino de tecnologia. A Metodologia aqui proposta para integração de projetos baseia-se em atividades colaborativas fundamentadas na abordagem de ensino sócio-interacionista. O desafio consiste em viabilizar a realização de trabalhos em equipe que permitam a construção do conhecimento de forma socializada, respeitando as restrições de prazo impostas às disciplinas nos ambientes de ensino. A metodologia proposta aborda quatro etapas: (1) preparação didático-pedagógica dos módulos propostos; (2) organização do ambiente de trabalho; (3) desenvolvimento dos módulos e (4) integração dos módulos em um Projeto Global Integrado. Neste trabalho foram utilizados a placa DE2 e o software de desenvolvimento Quartus II v. 8.1, ambos da Altera. A linguagem de descrição de hardware adotada foi Verilog. A validação da metodologia consistiu em aplicá-la no processo de integração de dois módulos construídos em uma situação real de ensino da disciplina Arquitetura de sistemas digitais ministrada no DEE da UFCG. Como resultado preliminar concluiu-se que as etapas e artefatos propostos para a metodologia de integração foram adequados e suficientes para promover as atividades colaborativas, com enfoque sócio-interacionista, entre os participantes do experimento. Para apoiar a aplicação da metodologia propõe-se a construção de uma ferramenta cuja especificação faz parte deste texto de dissertação. Com este trabalho espera-se contribuir para dinâmica e a eficácia do processo de ensino-aprendizagem de projetos didáticos com FPGA.

**Palavras-chave:** Experimentações metodológicas. Ensino de HDL/FPGAs. Sócio interacionismo. Metodologia para integração de módulos de projetos didáticos.

## ABSTRACT

Developing educational projects implemented with FPGAs has become a common practice in courses such as Electrical Engineering, and Computer Science, besides Technical degrees. Nonetheless, a problem still remains regarding the lack of a methodology to address the integration of modular projects from the didactical and pedagogical points of view. This dissertation aims to apply concepts from the area of education to technology education. The methodology proposed here focus on the integration of project modules based on collaborative activities such as those proposed by social interaction teaching. The challenge is to make possible the teamwork that allows knowledge building in a socialized environment, respecting the time constraints imposed at the classroom. The proposed methodology addresses four steps: (1) didactic-pedagogical preparation of the project modules, (2) organization of the work environment, (3) project modules development and (4) the integration of the modules in Global Project. During this work it was used the DE2 board and development software Quartus II v. 8.1, both supplied by the company Altera. The hardware description language adopted was Verilog. The validation of the methodology consisted in applying it in the integration process of two project modules that had been previously built during the teaching of the discipline "digital systems architecture" at the Electrical Engineering Department at UFCG. From this preliminary validation exercise it was concluded that the integration methodology was adequate to promote collaborative activities, such as the social interaction among the participants during the experiment. This text also presents the specification of a supporting tool to simplify the methodology application. This work is expected to contribute to dynamic and effective teaching-learning instructional designs with FPGA.

**Key-words:** Methodological experimentations. Teaching HDL/FPGA. Social-interactionism. Methodology for the integration of didactical projects modules.

## SUMÁRIO

<b>INTRODUÇÃO</b> .....	<b>11</b>
OBJETIVOS .....	12
MOTIVAÇÃO E RELEVÂNCIA .....	12
ESTRUTURA DA DISSERTAÇÃO .....	13
<b>1 ABORDAGENS DE ENSINO</b> .....	<b>14</b>
1.1 ABORDAGEM TRADICIONAL .....	14
1.2 ABORDAGEM TECNICISTA .....	14
1.3 ABORDAGEM CONSTRUTIVISTA-INTERACIONISTA .....	17
<b>2 PROJETOS DIDÁTICOS COM FPGA</b> .....	<b>22</b>
2.1 UNIVERSIDADE FEDERAL DE CAMPINA GRANDE .....	22
2.2 IFPE, CAMPUS RECIFE .....	23
2.3 UNIVERSIDADE FEDERAL DE SÃO CARLOS .....	24
2.4 ALTERA UNIVERSITY PROGRAM .....	24
2.5 FACULTY OF INFORMATION SCIENCES. HIROSHIMA, JAPAN. ....	26
2.6 INTERAÇÃO COM PROFISSIONAIS QUE ATUAM NA ÁREA .....	27
2.7 SÍNTESE DA ANÁLISE DA LITERATURA AVALIADA .....	28
<b>3 METODOLOGIA DE INTEGRAÇÃO DE PROJETOS DIDÁTICOS COM FPGA</b> .....	<b>30</b>
3.1 INTRODUÇÃO .....	30
3.2 SÍNTESE DO PROCESSO DE INTEGRAÇÃO DOS SUBPROJETOS .....	34
3.3 ETAPA 1 – PREPARAÇÃO DIDÁTICO-PEDAGÓGICA .....	35
3.4 ETAPA 2 – ORGANIZAÇÃO DO VIRTUAL AMBIENTE DE TRABALHO .....	40
3.5 ETAPA 3 – DESENVOLVIMENTO DOS SUBPROJETOS .....	42
<b>3.5.1 Subprojeto Controlador</b> .....	<b>44</b>
3.5.1.1 Código de descrição do módulo controlador .....	44
3.5.1.2 Código de instanciação do módulo controlador .....	46
3.5.1.3 Diagrama em blocos – módulo controlador .....	48
3.5.1.4 Quadro de teste de funcionalidade .....	48
3.5.1.5 Quadro das interfaces de integração do subprojeto controlador .....	51
<b>3.5.2 Subprojeto Processador</b> .....	<b>54</b>
3.5.2.1 Código de descrição do módulo processador .....	54
3.5.2.2 Código de instanciação do módulo processador .....	55
3.5.2.3 Diagrama em blocos – Módulo Processador .....	56
3.5.2.4 Quadro de teste de funcionalidade .....	56
3.5.2.5 Quadro das interfaces de integração do subprojeto Processador .....	59
3.6 ETAPA 4 – PROJETO INTEGRADO .....	60
<b>3.6.1 Código de instanciamento do módulo processador</b> .....	<b>64</b>
<b>3.6.2 Quadro de teste de funcionalidade do projeto integrado</b> .....	<b>64</b>
<b>4 A METODOLOGIA NUMA PERSPECTIVA SÓCIO-INTERACIONISTA</b> .....	<b>68</b>
<b>5 FERRAMENTA DE APOIO À INTEGRAÇÃO – MODELAGEM UML</b> .....	<b>73</b>
5.1 INTRODUÇÃO .....	73
5.2 LINGUAGEM UML .....	73
5.3 DESCRIÇÃO GERAL DO SISTEMA .....	74
5.4 DIAGRAMAS UML DA FERRAMENTA DE APOIO À INTEGRAÇÃO .....	76
<b>5.4.1 Diagrama de caso de uso</b> .....	<b>76</b>

<b>5.4.2 Diagramas de sequência .....</b>	<b>85</b>
<b>5.4.3 Diagrama de classes.....</b>	<b>86</b>
<b>6 AVALIAÇÃO DA METODOLOGIA – UMA APLICAÇÃO .....</b>	<b>87</b>
6.1 PREPARAÇÃO .....	87
6.2 CRITÉRIOS DE AVALIAÇÃO .....	88
6.3 O PROJETO DE TESTE.....	88
6.4 REALIZAÇÃO DO EXPERIMENTO .....	89
<b>6.4.1 Apresentação das tarefas.....</b>	<b>89</b>
<b>6.4.2 Relato das observações.....</b>	<b>90</b>
6.5 ANÁLISE DOS RESULTADOS .....	90
<b>CONSIDERAÇÕES FINAIS.....</b>	<b>92</b>
<b>BIBLIOGRAFIA .....</b>	<b>96</b>
<b>APENDICE A – Validação da Metodologia : Quadro das Etapas e Tarefas .....</b>	<b>101</b>
<b>APÊNDICE B – Formulário de observação e entrevista.....</b>	<b>103</b>

## INTRODUÇÃO

O avanço tecnológico na área da Microeletrônica e da Engenharia da Computação possibilitou aos projetistas de Sistemas Digitais a utilizar cada vez mais computação reconfigurável, a partir de *Programmable Logic Devices - PLDs* e *Programmable Gate Array - FPGAs*, no desenvolvimento de seus trabalhos. Para Cunha e Vieira [1], isso se deve a complexidade crescente dos sistemas digitais, uma vez que projetos no nível de portas lógicas com CIs TTL/CMOS específicos, montados em matriz de contatos, tornaram-se impraticáveis devido entre outros fatores, ao atraso global de resposta, o tamanho do layout físico, e a necessidade de construir protótipos para testes.

Edson Ferlin e Valfredo Pilla [2] afirmam que na sua instituição de ensino, criou-se uma disciplina específica de Computação Reconfigurável, a partir de outras disciplinas já existentes, onde se faziam uso das tecnologias de PLDs e FPGAs.

Nos Cursos de Engenharia Elétrica e Ciência da Computação, além dos Cursos Técnicos e de Tecnólogos, nessa área e em áreas afins, a tendência é a oferta de disciplinas para que os alunos possam conhecer e desenvolver projetos de circuitos digitais com dispositivos lógicos programáveis.

Dentre inúmeras Instituições de Ensino que oferecem aulas práticas e teóricas com essa tecnologia, podemos citar: o Instituto Indiano de Tecnologia, na cidade de Kapur [3], A Universidade de York, na Inglaterra [4], Faculdade da Ciência da Informação, em Hiroshima – Japão [5], a Universidade Federal de São Carlos – UFSCAR[6], a Universidade de Pernambuco – UPE [7], o Instituto Federal de Educação, Ciência e Tecnologia de Pernambuco – IFPE, Campus Recife [8], a Universidade Federal de Pernambuco - UFPE [9], a Universidade Federal de Campina Grande – UFCG [1][10], e a Universidade Federal do Espírito Santo UFES [11].

Entre os estudos já realizados sobre Implementação de Sistemas Digitais em FPGA, destacam-se: Arquitetura de Sistemas Digitais: Projetos em HDL[1], Projetos Didáticos de Sistemas e Arquitetura de Computadores Baseados em Placas FPGA [5], Microcontrolador SAP1 [6], Microcontrolador Básico com Processador core Nios II [12][13][21][22], e Projetos e Aplicações de Sistemas Digitais em FPGA [14].

O eixo central de investigação, objeto de estudo, é a Integração de Projetos Didáticos de Sistemas Digitais implementados em FPGA.

Esse objeto de estudo surgiu com professora Fátima Vieira, a partir de sua vivência em sala de aula, na necessidade de interligar projetos digitais didáticos, criados por seus alunos, em diferentes equipes de trabalho. A professora Fátima Vieira é Docente no Departamento de Engenharia Elétrica da UFCG [1].

## OBJETIVOS

O objetivo principal deste estudo foi a elaboração de uma metodologia de integração de projetos didáticos implementados em FPGA, com foco em uma abordagem de ensino sócio-interacionista.

Especificamente, para cumprir o objetivo principal foram realizadas as seguintes tarefas: pesquisa bibliográfica; identificação análise e uso das ferramentas disponíveis no Quartus II v.8.1 [15][16][17][18]; análise e uso das ferramentas disponíveis na placa DE2 da Altera - Hardware (Kit de desenvolvimento) Cyclone II FPGA[19][20]; criação de modelos de projetos didáticos com FPGA [21] para realizar ensaios em Verilog [11][22]; descrição UML de uma metodologia específica para apoiar a referida metodologia de Integração de projetos didáticos; realização de testes de validação da metodologia, e a análise dos dados coletados na observação e na entrevista, sobre os experimentos de validação.

## MOTIVAÇÃO E RELEVÂNCIA

O motivo da escolha por essa área de pesquisa, processamento da informação, emerge a partir do assunto "Modularidade e Integração de Projetos Didáticos com FPGAs" apresentado como um dos temas no processo de seleção de mestrado, pela Universidade Federal de Campina Grande, no projeto Interinstitucional – Minter, do IFPE, Campus Recife, e a referida Universidade. O tema possibilitou conciliar Ciência, Tecnologia e Educação, áreas de atuação do mestrando, enquanto docente nos cursos de Eletrônica e Telecomunicações do mencionado IFET.

Ao abordar este tema, percebeu-se que os fabricantes de ferramentas de desenvolvimento de projetos didáticos com FPGA oferecem recursos que possibilitam a integração de projetos, mas não apresentam metodologia que sistematizem a

integração de projetos didáticos, voltada para uma didática especial, como é o caso da metodologia desenvolvida neste estudo. Portanto, trata-se de um objeto de estudo ainda pouco explorado nesse contexto, o que amplia a contribuição que irá trazer para todos que atuam na área de Educação, envolvendo ensino, ciência e tecnologia.

## ESTRUTURA DA DISSERTAÇÃO

Nessa dissertação, no Capítulo 1 é feito um estudo sobre as principais formas de abordagens de ensino, com ênfase na abordagem tecnicista e na abordagem interacionista. Apresenta também um quadro comparativo entre as referidas abordagens.

A seguir, no Capítulo 2, são apresentados trabalhos de projetos digitais, desenvolvidos em diferentes instituições de ensino. No final desse capítulo é apresentado um quadro comparando os referidos trabalhos, relacionado com as ferramentas de desenvolvimento, a linguagem de descrição adotada, e a forma de integração.

No Capítulo 3 são apresentadas as etapas e os materiais necessários para uso da metodologia desenvolvida nesse estudo. Além disso, é apresentado um exemplo de aplicação para a referida metodologia, onde é feita uma análise passo a passo dos procedimentos de execução.

No capítulo 4, é feita uma explanação para reflexão sobre a formação e a prática de um profissional que atua em sala de aula, em relação ao professor educador e sua práxis. Ressaltam-se também as questões sobre mudanças de paradigmas relacionadas às abordagens de ensino.

É apresentada no Capítulo 5 a modelagem UML de uma ferramenta de apoio à metodologia de integração de projetos didáticos digitais, além da sugestão de uma linguagem de programação para o desenvolvimento da referida ferramenta.

Para validação do trabalho realizado, no Capítulo 6, é descrito o teste de validação da metodologia realizado com usuários, e logo em seguida é feita a análise dos dados coletados.

## 1 ABORDAGENS DE ENSINO

*“O principal objetivo da educação é criar indivíduos capazes de fazer coisas novas e não simplesmente repetir o que as outras gerações fizeram.” Jean Piaget.*

O tema desta pesquisa, Integração de Projetos Didáticos com FPGAs, nos remete a um estudo complementar de fundamentação pedagógica, a partir da prática tradicional de ensino.

A teoria sobre abordagem de ensino é usada como referência para direcionar o modo como as pessoas aprendem e como devemos ensinar [23]. Entre as diferentes formas de abordagem de ensino será dada ênfase ao tecnicismo e o interacionismo.

### 1.1 ABORDAGEM TRADICIONAL

A abordagem tradicionalista não é uma teoria. É na verdade uma prática comum de Ensino, onde o professor é a fonte de conhecimento e o aluno um sujeito passivo. No pensamento de Paulo Freire [24], trata-se de uma “educação bancária<sup>1</sup>”. Neste tipo de abordagem a aprendizagem ocorre através da memorização. Não há espaço para o diálogo, criatividade, nem reflexão. O professor age com autoritarismo sobre o comportamento dos alunos, e determina o que podem e o que não podem, dentro e fora do âmbito escolar.

### 1.2 ABORDAGEM TECNICISTA

A abordagem tecnicista apresenta em seus estruturantes elementos do ensino tradicional, porém fundamentada na Teoria Behaviorista. Nessa abordagem, o importante são as técnicas de ensino, os instrumentos de controle e avaliação, além dos instrumentos de estímulos audiovisuais.

---

<sup>1</sup> De modo semelhante, como se deposita dinheiro em um banco; o professor, fonte de conhecimentos, deposita conteúdos na cabeça vazia do aluno.

O centro do ensino não é mais o professor, nem mais o aluno, mas as técnicas. Daí o nome desta Pedagogia: tecnicismo ou escola tecnicista. Partindo dela, reorganiza-se o processo educativo no sentido de torná-lo objetivo e operacional. As escolas passam a burocratizar-se. Exige-se dos professores a operacionalização dos objetivos, como instrumento para medir comportamentos observáveis, válidos porque são mensuráveis e controláveis. Dissemina-se o uso da instrução programada (auto-ensino), das máquinas de ensinar, testes de múltipla-escolha, do tele-ensino e múltiplos recursos audiovisuais [25].

Em um plano de aula tecnicista, os tópicos como a introdução, o desenvolvimento, a conclusão e avaliação são organizados sistematicamente para serem apresentados um a um, em tempos rígidos e bem definidos. Não há espaço para o dialógico, porque os recursos didáticos audiovisuais, em conjunto com técnicas de ensino apropriadas para condicionar os alunos, são vistos como suficientes para garantir o sucesso do ensino-aprendizagem.

O Tecnicismo é uma concepção baseada nas idéias behavioristas de Skinner.

Para o psicólogo behaviorista norte-americano Burrhus Frederic Skinner (1904-1990), **a educação deve ser planejada passo a passo, de modo a obter os resultados desejados na "modelagem" do aluno.** (...) No behaviorismo, o fundamento é centrado ao comportamento (behavior, em inglês), tomado como um conjunto de reações dos organismos aos estímulos externos. Seu princípio é que só é possível teorizar e agir sobre o que é cientificamente observável. (...) A Ciência é exaltada em sua plenitude. Somente através dela, pode-se chegar à verdade. [26].

No Brasil, o tecnicismo foi implantado em plena ditadura militar, num cenário onde se almejava crescimento industrial.

A escola deveria ser produtiva, racional e organizada e **formar indivíduos capazes de se engajar rápida e eficientemente no mercado de trabalho.** (...) Para tanto, à imagem da empresa, a escola deveria apresentar uma produtividade eficiente e eficaz. (...) É claro que os treinamentos de educadores nos anos 70 refletiram, e muito, esta tendência que **valorizava fundamentalmente os meios, as tecnologias e os procedimentos de ensino** - apresentados sempre como "neutros", "eficientes" e "eficazes". E **isto teve conseqüências negativas na educação escolar brasileira que perduram até o presente momento.** [27].

Na abordagem de ensino tecnicista, o aluno é um mero receptor de conteúdos, criados a partir de interesses simplesmente mercadológicos, transmitidos de forma sequencial e unilateral, pelo professor. Na verdade, no ponto de vista político, o

próprio professor também é um sujeito passivo, pois simplesmente transmite o que já está preconizado, deliberado por aqueles que estão no poder.

No Behaviorismo o relacionamento humano fica comprometido. A relação professor-aluno não é relevante, assim também como as relações grupais, uma vez que a ênfase se encontra na programação, onde a aprendizagem ocorre mecanicamente (...) Lima [28].

Nessa visão, o professor não é valorizado, mas sim os instrumentos, e a tecnologia. O professor passa a ser apenas um especialista, um elo entre a verdade científica e o aluno.

Na avaliação em sala de aula, valoriza-se a memorização. Nas ciências exatas, a solução do problema está na lembrança de fórmulas: é só substituir os valores. Na área das ciências humanas, a ordem é decorar para passar: perguntas prontas, para respostas prontas, sem espaço para reflexão, nem para a crítica. Nas atividades de laboratório, o objetivo é verificar se o que foi mostrado na teoria funciona na prática. Nesse contexto, é comum ver um aluno parado, aguardando a presença do mestre na sua bancada, quando sua prática não funciona, conforme proposto no manual ou roteiro de atividades práticas.

Instrumentos como teste de múltiplas escolhas, preenchimento de lacunas, perguntas com respostas previsíveis, incluindo SIM ou NÃO, são usadas no processo de avaliação. Nas correções, desde que as respostas estejam de acordo com o que foi ensinado, aparecem palavras de incentivo ou reforço: "muito bem", "continue assim"... Valorizam-se os resultados quantitativos. Não há espaço para o potencial cognitivo, nem para a subjetividade do aluno.

Num processo de ensino-aprendizagem, no qual se pratica uma abordagem predominantemente tecnicista, onde a média necessária para aprovação é sete, a seguinte situação é comum:

Aluno A: N1 = 3,0; N2 = 10,0; M=6,5. Está reprovado.

Aluno B: N1 = 10,0; N2 = 4,00; M = 7,0. Está aprovado.

No tecnicismo, a aprendizagem é fruto de memorizações, de reconstrução passo-a-passo daquilo que já foi pensado e construído. Valoriza-se a habilidade de produção manual, como processo eficaz de aprender fazendo. Trata-se de uma abordagem de ensino com finalidade de se modelar o indivíduo com comportamentos desejados para o mercado de trabalho [29].

Na Abordagem tecnicista, não é previsto o diálogo entre aluno-professor, nem tão pouco, entre aluno-aluno. É um ensino de transmissão unilateral de informações, que deverão ser posteriormente reproduzidas pelos alunos, da forma como lhe foi enviada.

### 1.3 ABORDAGEM CONSTRUTIVISTA-INTERACIONISTA

No Construtivismo/Interacionismo, a essência está no fato do educando poder interagir com o objeto do conhecimento, e acima de tudo, poder processar as informações, através de processos mentais, de reflexões<sup>2</sup>. Ele passa a ser um sujeito ativo dentro do processo de ensino-aprendizagem. O desenvolvimento intelectual é resultado das relações recíprocas do homem com o meio. Daí a denominação de Abordagem Interacionista.

Nessa linha destacam-se Jean Piaget e Vigotsky. Para Piaget, precursor do construtivismo cognitivo<sup>3</sup>, o aluno aprende sozinho a partir de sua própria interpretação de mundo, independente de relações interpessoais. No pensamento de Vigotsky, sócio-construtivista, além das estruturas cognitivas e a capacidade de leitura e interpretação de mundo, a interação social entre os sujeitos é essencial para haver aprendizagem e desenvolvimento. A abordagem de ensino sócio-construtivista é também denominada de humanística ou sócio-cultural.

Em Piaget, a aprendizagem ocorre em etapas ligadas diretamente ao desenvolvimento mental e individual de cada aluno, que deverá construir seu próprio conhecimento, sem levar em conta o contexto histórico-social do ambiente no qual se encontra o objeto do conhecimento. Para ele, os diferentes níveis de aprendizagem ocorrem em fases bem definidas ao longo do desenvolvimento humano, referente ao crescimento e à idade. Trata-se de um processo universal, puramente biológico e individual. As estruturas universais, propostas por Jean Piaget estão divididas em quatro etapas ou estágios [29]:

---

<sup>2</sup> O conjunto de esquemas mentais, relacionados com a consciência e o pensamento sobre si mesmo, para examinar o seu próprio conteúdo por meio do entendimento, da razão [30].

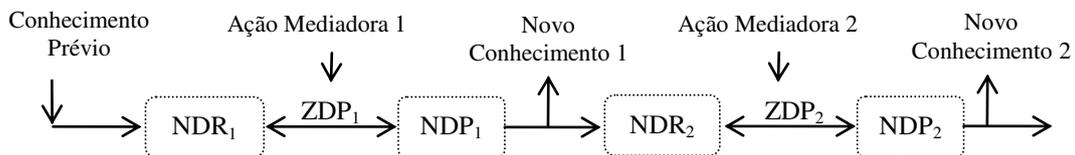
<sup>3</sup> Segundo Aurélio [30], o conjunto dos processos mentais usados no pensamento, na percepção, na classificação, e reconhecimento etc. Relativo ao pensamento.

- a) estágio sensório-motor, de 0 aos 2 anos: o indivíduo relaciona-se com o mundo através de suas estruturas físicas, a partir de ações mecânicas tais como deixar cair, tatear, olhar, sugar, pegar e cheirar.
- b) estágio pré-operacional, de 2 aos 7 anos: o indivíduo é capaz de lidar com imagens concretas;
- c) estágio das operações concretas, de 7 aos 11 anos: capacidade de operações mentais, inclusive operações lógicas, a partir de objetos concretos;
- d) estágio das operações formais, dos 11 anos à fase adulta: realiza atividades mentais, não apenas com objetos concretos, mas também com símbolos. Capacidade de pensar em termos de hipóteses e possibilidades. É o início do raciocínio científico, em sua forma sistemática.

A implantação da abordagem construtivista de Piaget nas Escolas deu-se na década de 80. Nela o papel do professor é de favorecer a descoberta individual do aluno.

Para Vygotsky [31], atingida uma fase orgânica/biológica mínima/elementar, a aprendizagem ocorre a partir de interações sociais. Para ele a aprendizagem ocorre na região ZDP<sup>4</sup>, situada entre os níveis NDR e NDP:

- a) Nível de Desenvolvimento Real – NDR: representa as coisas que o indivíduo é capaz de fazer sozinho, com o conhecimento já adquirido;
- b) Nível de Desenvolvimento Potencial – NDP: representa as coisas que o indivíduo é capaz de fazer com a ajuda de pessoas mais experientes do grupo, num processo de socialização da construção do conhecimento.



**Figura 1** - Esquema explicativo dos níveis de desenvolvimento de Vygotsky.

A ZDP não é a mesma para todos os indivíduos. Por isso cada aluno apresenta seu próprio tempo pedagógico de aprender. A função do professor é interferir diretamente no processo, atuando na ZDP do aluno com situações-problema<sup>5</sup>, que provoquem desequilíbrio cognitivo e estimulem sua curiosidade epistemológica, no sentido de promover avanços que não ocorrem espontaneamente. O processo de desenvolvimento mental é intrapessoal e ocorre imediatamente, logo após a

<sup>4</sup>A Zona de Desenvolvimento Proximal - ZDP é a região de transição entre os níveis NDR e o NDP, própria para ação mediadora do professor que ajudará na aprendizagem do aluno [31].

<sup>5</sup>Situações-problema são questões elaboradas de forma contextualizada.

aprendizagem, através de relações interpessoais. Para Vygotsky, a aprendizagem e o desenvolvimento não ocorrem simultaneamente, como propõe Jean Piaget.

Nessa abordagem de ensino, a educação escolar é essencial para o desenvolvimento sistemático e sócio-cultural do aluno, na medida em que a aprendizagem ocorre através de relações interpessoais, com grupos inseridos num determinado contexto.

A aplicação da abordagem sócio-interativa de Vygotsky na prática educacional requer que o professor reconheça a idéia de ZDP e estimule o aluno com trabalhos colaborativos, de modo a potencializar o desenvolvimento do aluno, a partir de interações com outros alunos. Isso significa que o aluno aprende junto ao outro o que o grupo social produz, além do conhecimento em si, a linguagem, valores e atitudes [29].

A aprendizagem é construída mediante um processo de interação do indivíduo com seu ambiente sociocultural e com o suporte de outros indivíduos mais experientes.

Independente da abordagem, o ensino pode ser auxiliado por computador. Numa abordagem sócio-interacionista o uso de aplicativos de software possibilitam ao aluno uma interação com os objetos virtuais de tais ambientes, a partir de atividades colaborativas, de situações problematizadoras.

Nessa situação, “o papel de observador/experimentador passa a ser do professor, com a função objetiva e clara de orientar o aprendizado e problematizar<sup>6</sup> as situações vivenciadas [32][33]”.

Enfim, na sua práxis, o professor pode optar por qualquer tipo de abordagem de ensino, principalmente quando essa escolha depende de sua intencionalidade política. Mesmo assim, dentro e fora de sala de aula, nos diferentes níveis de ensino, ainda é visível com certa frequência, a predominância de uso da abordagem tradicional e/ou da abordagem tecnicista, aplicadas no processo de ensino-

---

<sup>6</sup> No entendimento de Paulo Freire [24], ao contrário da concepção “bancária” e antidialógica, uma educação problematizadora organiza-se em torno da visão do mundo dos educandos. A aprendizagem não ocorre através do envio de conteúdos em forma de dados que os alunos devem memorizar, mas como atividade deliberada, que busca soluções para problemas contextualizados e relevantes à vida dos educandos.

Problematizar é contextualizar um assunto. É introduzir um tema e criar problemas para serem resolvidos com atividades através das quais se resgatam os conhecimentos prévios, e o cotidiano social e/ou profissional do aluno, motivando-o ao estudo, à reflexão, e a enfrentar novos desafios para construir explicações satisfatórias ao problema proposto.

aprendizagem. Isso é fruto da forte influência que esse tipo de abordagem de ensino exerceu, e ainda exerce, sobre a formação dos professores que tiveram mestres que também aprenderam e foram formados nesse mesmo contexto.

O problema se agrava mais ainda, quando esse professor acredita ser a fonte do saber, apenas pelo fato de deter algum tipo de conhecimento, e, além disso, ignora de alguma forma a educação enquanto ciência, e todos os conhecimentos criados por ela, principalmente ao que se refere aos fundamentos didáticos-pedagógicos.

No Quadro 1 é apresentada uma análise comparativa entre as abordagens Tecnicista e Interacionista. O termo interacionista refere-se ao fato da aprendizagem ocorrer a partir de interações entre o sujeito e o objeto do conhecimento. Construtivismo é relativo à construção do conhecimento, em oposição à transmissão passiva de informação, praticada no tecnicismo.

Ao que se refere ao ensino, a Metodologia desenvolvida neste estudo para a integração de projetos didáticos com FPGA, usa como suporte didático-pedagógico abordagens interacionistas, com ênfase em atividades colaborativas, tendo como ferramenta de apoio o computador.

**Quadro 1** - Comparativo entre as abordagens de ensino analisadas

	<b>TECNICISMO</b>	<b>INTERACIONISMO COGNITIVO</b>	<b>SÓCIO-INTERACIONISMO</b>
Organização	O Ensino se resume na transmissão unilateral de conteúdos, destinados a modelar os alunos para o mercado de trabalho. (FUSARI, 2010).	O Ensino é organizado visando o desenvolvimento cognitivo do aluno.	O Ensino é centrado na socialização do conhecimento. Baseia-se, sobretudo no diálogo.
Objetivos	Os objetivos propostos são comportamentais. Valorizam ações do "fazer para aprender fazendo". Não são valorizadas as relações interpessoais.	Os objetivos são estruturados para serem alcançados através de atividades que valorizem as ações individualizadas do aluno, diante da sua interação c/o objeto do conhecimento.	Os objetivos são organizados com a finalidade de elaborar atividades voltadas para ações colaborativas entre alunos, e também entre alunos e professor.
Foco	O foco do ensino não é mais o professor, nem o aluno (DONATO, 2010), mas as técnicas instrumentais de ensino; daí o nome: tecnicismo. Os recursos didáticos audiovisuais, em conjunto com as técnicas de ensino e avaliação empregadas para condicionar os alunos, são considerados suficientes para assegurar os objetivos desejados.	O foco é o aluno. O professor não é mais o centro do saber. É um facilitador da aprendizagem, que auxilia o aluno, propondo-lhe situações-problema, que estimulam seu desenvolvimento individual e espontâneo.	O foco é o aluno, e suas relações interpessoais na construção do conhecimento. O professor é o mediador da aprendizagem, que atua diretamente na ZDP (Zona de Desenvolvimento Proximal) do aluno, para auxiliá-lo na aquisição socializada de novos conhecimentos.
Papel do aluno	O aluno é um receptor passivo, destinado a reproduzir aquilo que já foi pensado e criado. Quando avaliado são valorizadas sua capacidade de memorização e sua destreza manual.	O aluno é um autodidata. Ele aprende a partir de suas próprias descobertas e reflexões, ao interagir com o objeto do conhecimento.	O aluno aprende a partir de interações sociais, em torno do objeto do conhecimento. A mediação é do professor ou de outro aluno mais experiente.
Aprendizagem	No pensamento tecnicista a aprendizagem é focada no condicionamento dado ao indivíduo, num processo de estímulo-resposta. A reflexão não é valorizada. Assim, consiste em um processo predominantemente alienador.	Segundo Piaget, a aprendizagem e o desenvolvimento ocorrem simultaneamente. A aprendizagem depende da capacidade cognitiva do aluno, o qual aprende a partir de suas próprias descobertas. Toda aprendizagem depende apenas do desenvolvimento biológico e da idade cronológica do aluno.	Para Vygotsky, primeiro ocorre a aprendizagem, a partir de relações interpessoais. A aprendizagem é mediada pelo professor na ZDP de cada aluno. Depois, através de complexas estruturas mentais, ocorre o desenvolvimento, de forma intrapessoal.

## 2 PROJETOS DIDÁTICOS COM FPGA

O desenvolvimento de projetos didáticos no ensino de HDL tornou-se atividade em cursos técnicos e superiores, em diferentes áreas: Ciência da Computação, Sistemas da Informação, Engenharia Elétrica e Eletrônica, Engenharia da Computação, entre outras. Neles são oferecidas disciplinas sobre dispositivos programáveis e reconfiguráveis, a exemplo da tecnologia FPGA.

Com o avanço da tecnologia surgiram os FPGAs. Esta tecnologia inovadora está viabilizando a construção e prototipação de circuitos digitais complexos sem a necessidade de muitos recursos computacionais e financeiros<sup>7</sup>. A possibilidade de implementar um circuito digital em um ambiente simplificado e de baixo custo está popularizando cada vez mais esta tecnologia [14].

No desenvolvimento de projetos didáticos com FPGA nem sempre são previstas atividades colaborativas para a realização dos respectivos projetos.

O foco dessa pesquisa é o estudo sobre a integração de projetos didáticos implementados com FPGAs, desenvolvidos por diferentes equipes de trabalho. Nesse contexto, deverão ser observados os seguintes aspectos: interação e mediação no processo de ensino-aprendizagem; e a integração entre projetos escritos em HDL.

A seguir são apresentados e avaliados referenciais teóricos com temas relacionados com o nosso objeto de estudo, a partir dos quais, será proposta uma metodologia para a integração de projetos didáticos implementados com FPGA.

### 2.1 UNIVERSIDADE FEDERAL DE CAMPINA GRANDE

Fátima Vieira e Alexandre Cunha [1] propõem projetos para serem desenvolvidos em Verilog, relacionados com microprocessadores e microcontroladores. As diversas propostas de projetos são divididas entre os alunos para formar grupos de trabalho. Cada grupo deve criar e validar seu projeto. Os alunos devem: desenhar o diagrama em blocos; ilustrar as entradas e saídas dos sinais envolvidos; construir a tabela verdade do comportamento dos circuitos, programar o dispositivo FPGA na placa de

---

<sup>7</sup> A construção de circuitos digitais complexos era uma tecnologia dominada apenas por grandes empresas ou universidades de renome internacional [14].

desenvolvimento; elaborar sequência de testes, usando chaves e leds para validar o comportamento expresso na referida tabela verdade. No final, devem apresentar ao professor o projeto em funcionamento, explicando todo processo de construção. É exigido também, o relatório sobre o experimento, segundo as normas explícitas no roteiro de projetos, entregue pelo professor.

O diagrama de cada estágio deve ser apresentado, com suas respectivas entradas/saídas e seus sinais de controle.

Na metodologia usada por Vieira e Cunha não há proposta de atividades colaborativas para diferentes equipes de trabalho. Cada equipe recebe uma placa FPGA para validação individual do seu projeto. Na proposta de trabalho é solicitado, para todos os projetos, que cada equipe apresente o diagrama em blocos do sistema, acompanhado de sua respectiva análise e síntese. No caso de estudos com comunicação serial e controlador de interrupção fez-se a interligação entre duas placas, porém por uma mesma equipe. A integração foi feita com cabo *flat*, através da saída GPIO\_0, da placa DE2 da Altera. As equipes eram formadas por no máximo quatro, e no mínimo dois alunos. Predomina uma abordagem de ensino interacionista cognitiva.

## 2.2 IFPE, CAMPUS RECIFE

Remy Eskinazi Sant'anna [8] orientou trabalho de iniciação científica para implementação de um microcontrolador em FPGA, a partir do livro de Edward Ordonez [14]. O projeto foi desenvolvido em VHDL, de forma modular, simulado com *waveform* do Quartus II, e posteriormente gravado na placa FPGA DE1 da Altera, onde foram feitos os ensaios de validação do hardware de cada módulo. Depois, os módulos foram integrados num mesmo projeto para formar o sistema. Em seguida, compilou-se e realizaram-se os testes de simulação. No final, o sistema foi gravado na placa FPGA para os ensaios de validação do hardware.

O foco principal foi validar o projeto apresentado no livro de Ordonez, fazendo modificações na CPU. O código VHDL do sistema original encontra-se disponibilizado no livro. Criaram-se os módulos e as integrações no mesmo arquivo do projeto integrador. Apresentou-se o diagrama em blocos do sistema, mas sem a

preocupação com padrões para as interfaces. Segundo o professor Remy Eskinazi, o referido projeto foi resultado de um trabalho de orientação de iniciação científica. A abordagem predominante foi a Interacionista cognitiva.

### 2.3 UNIVERSIDADE FEDERAL DE SÃO CARLOS

O professor Edilson Kato[6], da Universidade Federal de São Carlos, propôs para seus alunos, a construção do microprocessador SAP1 de Malvino [36] na plataforma Quartus II, usando *block Diagram/Schematic File*. O objetivo foi desenvolver o projeto do SAP1 usando a plataforma do Quartus II, e após compilação, realizar ensaios na placa FPGA, Flex 10K, da Altera. O sistema foi reproduzido na íntegra, conforme apresentado por Malvino, a partir do esquema elétrico apresentado em seu livro texto.

Não houve integração de projetos/módulos de forma colaborativa<sup>8</sup>. O projeto foi desenvolvido por equipes isoladas, sem interação com outros grupos. A orientação foi feita pelo professor com base no livro texto de Malvino. Há tendência interacionista cognitiva na forma de conduzir o ensino. A escolha pela descrição em forma de diagrama em blocos acarretou na necessidade da análise e síntese de funcionamento de todo sistema, conforme apresentado no livro texto. No entanto, não foram enfatizadas a importância e a metodologia de uso do diagrama em blocos como ferramenta principal na construção do sistema.

### 2.4 ALTERA UNIVERSITY PROGRAM

É proposto como exercício, a criação de um microcomputador básico com o processador core<sup>9</sup> NIOS II [12][13][37][38], a partir do uso do software *SOPC Builder* [39] para integrar os módulos que irão compor o sistema. Cada componente é selecionado pelo projetista, e consiste de um conjunto estruturado de arquivos dentro de um diretório, criados automaticamente pelo *SOPC Builder*. Os referidos

---

<sup>8</sup> Nesta proposta, trabalho em equipe não é sinônimo de trabalho colaborativo. Aqui, o sentido de colaborativo está relacionado com atividades onde existem interações entre membros de diferentes equipes de trabalho.

<sup>9</sup> O core NIOS II é um núcleo de processador desenvolvido pela Altera, disponível em hardware reconfigurável, num chip FPGA.

arquivos são criados automaticamente com a linguagem HDL escolhidos no início da criação do projeto, no Quartus II<sup>10</sup>. O objetivo é a implementação de um computador básico a partir do core Nios II e fazer uso dos recursos da placa DE2.

Não há proposta específica de trabalhos interativos entre equipes, na montagem e *start* de funcionamento do microcomputador. Isso ocorre porque o programa fonte e a ligação entre os respectivos módulos que compõem o microcontrolador são feitos automaticamente via software *SOPC Builder*<sup>11</sup>. Depois de configurado, o programa que irá rodar no micro criado deve ser criado, e compilado. Somente assim, o micro funcionará em sua plenitude. Nesse caso, a compilação é feita com o Software *Debug Client* da Altera.

Em 2006, foi proposto no mesmo programa da Altera, também como exercício de atividades, a criação de um sistema simples de computador usando o Quartus II e o *SOPC Builder*. O processador foi criado a partir do core NIOS II. A diferença, em relação ao projeto proposto em 2007, é o que o referido processador possui apenas A CPU e uma Memória *on-chip*, caracterizando um uP de arquitetura mínima, ideal para aulas introdutórias sobre o tema.

Em ambos os casos foram apresentados roteiros de atividades. Conforme cada roteiro trata-se de um ensino predominantemente tecnicista, onde o objetivo é reproduzir as atividades passo a passo, de acordo com os procedimentos pré-estabelecidos pela Altera. O roteiro é parte do material de cursos de capacitação oferecido pela Altera, com o objetivo de ensinar como usar os recursos do Quartus II, associado ao *SOPC Builder*.

---

<sup>10</sup> Quartus II é uma ferramenta de software criada pela Altera para análise e síntese de projetos de HDL, o que permite ao desenvolvedor: compilar seus projetos, realizar análise de timing, examinar os diagramas RTL, realizar simulações, gravar projeto em placas FPGA, entre outros atributos.

<sup>11</sup> *SOPC Builder* é um software Altera que automatiza conexões entre componentes de hardware para criar um sistema de computador completo, a partir do core microprocessador NIOS II, que funciona em qualquer um dos seus vários chips FPGA. *SOPC Builder* incorpora uma biblioteca de componentes pré-fabricados, incluindo o Nios II, controladores de memória, interfaces e periféricos. As interligações são feitas a partir de um barramento lógico, denominado de *Avalon*.

## 2.5 FACULTY OF INFORMATION SCIENCES. HIROSHIMA, JAPAN.

Hiroyuki Ochi [8] apresentou um trabalho sobre Ensino de Sistemas Embarcados em FPGAs. O objetivo de Hiroyuko foi apresentar um plano de ensino que permitisse estudantes de graduação entender e desenvolver um microprocessador RISC básico, de 16 bits, com *pipeline*, em cursos sobre introdução ao computador e hardware do sistema. De acordo com a proposta de Hiroyuki, para que seja possível a realização de cursos introdutórios sobre arquiteturas de microprocessadores, o ensino deve ser feito em quatro etapas: implementação; análise de funcionamento; correção dos possíveis erros, e reavaliação; apresentação dos resultados. Segundo Hiroyuki, embora se utilize um processador simples, o aluno pode testar sua funcionalidade, inclusive executar programas de aplicação a partir de um código objeto. O referido modelo possui atributos suficientes para evitar riscos de desmotivação do aluno, devido a sensação de incapacidade, afirma Hiroyuki.

Os projetos foram descritos em Verilog. Toda descrição foi elaborada no *Cadence Synergy 2.2*, e implementado no FPGA XC4013-6 da Xilinx.

A abordagem proposta, segundo o autor, é mais efetiva no processo de ensino e aprendizagem de microprocessadores, porque o aluno não ganha apenas o conhecimento sobre processador RISC, mas também a sua própria forma de compreensão do hardware, e sua própria metodologia sobre o desenvolvimento de algo novo. Em sua explanação, Hiroyuki apresenta as diretrizes de um plano de ensino de um curso básico de microprocessadores. O foco é de uma abordagem de ensino, com predominância interacionista.

A proposta de Hiroyuki Ochi é valiosa, na medida em que integra tecnologia eletrônica e ciência da computação, com a Educação, dentro do processo de ensino e aprendizagem em laboratório de desenvolvimento de projetos didáticos, implementados com FPGA. Apresentou roteiro de abordagem de Ensino, dedicado à compreensão e desenvolvimento de um processador didático com FPGA, em laboratório, para estudantes de graduação. Não apresentou sugestões para uma metodologia de trabalho com atividades colaborativas, entre diferentes equipes.

## 2.6 INTERAÇÃO COM PROFISSIONAIS QUE ATUAM NA ÁREA

Zilcio Sales de Lavor [40], professor titular da Universidade de Pernambuco (UPE), afirma que a forma correta de desenvolvimento de projetos de sistemas digitais é usar a técnica de integração de módulos, mesmo quando todo sistema é descrito por um só aluno ou grupo de alunos. A linguagem de descrição usada pelo referido professor é VHDL. Suas turmas são compostas por 17 alunos, e a placa de desenvolvimento é a *UP2 Education Board*, da Altera. A Disciplina é “automação de projetos de circuitos integrados”, no turno da noite, e a maioria dos alunos trabalha. Atualmente suas aulas são ministradas com predominância interacionista cognitiva, onde cada equipe é desafiada a resolver uma situação-problema, na forma de projeto. Também são ministradas aulas expositivas, teóricas, com exercícios resolvidos e propostos, além de atividades práticas onde se valorizam a capacidade de aprender fazendo, com o objetivo de o aluno conhecer e usar os principais recursos da ferramenta de desenvolvimento de projetos. A partir daí são propostas atividades que exigem produção intelectual.

Segundo Wellington Messias [41][42], coordenador de equipes, e integrador de projetos da Procenge tecnologia, dependendo da extensão e complexidade do projeto, torna-se necessário dividi-lo em módulos, entre os membros que formam sua equipe de trabalho. Posteriormente, a integração do sistema é feita por um único profissional, denominado de Integrador de projetos. Nesse contexto, o método de trabalho adotado por Messias é o *SCRAM* [43]<sup>12</sup>.

Paulo Sérgio Brandão [44], professor titular do Instituto Federal de Educação, Ciência e Tecnologia – IFPE, Campus Recife, afirma que sua intencionalidade de ensino é interacionista, a partir de atividades colaborativas<sup>13</sup>. No entanto, na maioria das vezes só consegue apenas resultados com características tecnicistas: “os alunos apresentam comportamento uniforme, apenas no desejo de obter soluções sem pensar. Nesse sentido são todos iguais. É uma geração de conflitos e de resistência à

---

<sup>12</sup> Na Engenharia de Software existem metodologias de projetos bem definidas. Entre elas destacam-se a metodologia *Extreme Programming (XP)*, e a *Scrum*. Estes são processos que demandam interações frequentes entre os membros das equipes de desenvolvimento.

<sup>13</sup> O referido professor é titular da disciplina técnicas digitais. As práticas são realizadas na plataforma de desenvolvimento da Altera: placa DE1 + Quartus II.

construção e a socialização do conhecimento, diz Paulo Sérgio". As atividades práticas servem apenas para verificar o que foi ensinado na teoria. A prática é a mesma para todos. São trinta alunos, com no máximo três alunos por bancada. A plataforma de desenvolvimento é da Altera: placa DE1 + Quartus II. As práticas consistem na descrição do circuito em diagrama elétrico/blocos, simulação, e gravação na placa FPGA.

## 2.7 SÍNTESE DA ANÁLISE DA LITERATURA AVALIADA

Em todos os trabalhos apresentados o tema principal de estudo foi o desenvolvimento de projetos implementados em FPGA, no domínio de sistemas digitais. Destacam-se entre eles, o trabalhos de Vieira e Cunha[1], e de Hiroyuki SCHIO [8], nos quais são apresentados roteiros de atividades sobre o tema, sendo que em seu estudo SCHIO apresenta sugestões para elaboração e execução de uma plano de ensino. Outra contribuição relevante é encontrada no trabalho do professor Zilcio de Lavor, que mostra sua preocupação em desenvolver um ensino fundamentado em uma abordagem interacionista. Esta também é uma preocupação da professora Fátima Vieira, da UFCG.

Em Fátima Vieira observa-se o uso de diagrama em blocos como ferramenta de análise e síntese de funcionamento de sistemas. No entanto, não é usado como elemento de referência para amostra e definição das interfaces.

Em Messias, observa-se a aplicação de trabalhos colaborativos em atividades profissionais, fora do âmbito acadêmico.

No quadro 2 é apresentada uma síntese dos trabalhos avaliados, onde são apontados os principais estruturantes, relacionados à plataforma de desenvolvimento, a linguagem de descrição usada em cada projeto apresentado, e o indicativo das características correlacionadas com esta proposta de pesquisa.

**Quadro 2** - Comparação entre trabalhos, com base nas ferramentas de desenvolvimento e a forma de integração.

Identificação		Plataforma			Descrição do Projeto					Modos de Integração dos módulos	
Autores	Instituições, ano	Altera	Xilinx	Outra	Verilog	VHDL	System Verilog	Blocos Estruturais	Sopc Builder	Colocam-se os códigos de Integração e de descrição dos módulos individuais em um mesmo arquivo. A tarefa é feita por um único projetista, ou por equipes que trabalham de modo independente.	Os códigos de Integração e de descrição dos módulos individuais são criados em arquivos diferentes, e em seguida integrados. A tarefa é desenvolvida de forma colaborativa, por diferentes equipes de trabalho.
Fátima Vieira e Alexandre Cunha	UFCG-PB, 2009	X			X					X	
Remy Eskinazi Sant'anna	IFPE, 2009	X				X				X	
Edilson Kato	UFSCAR, 2008	X						X		X	
Altera Corporation	Altera Corporation, 2007	X							X	X	
Altera Corporation	Altera Corporation, 2006	X							X	X	
Hirouyk Ochi	Faculdade da C. da Informação, 1997		X		X					X	
<b>Metodologia Desenvolvida</b>	<b>UFCG / IFPE</b>	<b>X</b>			<b>X</b>					<b>X</b>	<b>X</b>

### **3 METODOLOGIA DE INTEGRAÇÃO DE PROJETOS DIDÁTICOS COM FPGA**

#### **3.1 INTRODUÇÃO**

Quando projetos didáticos digitais implementados com FPGAs são desenvolvidos, de forma modular, executados por diferentes equipes de trabalho, no final torna-se necessário integrar os subprojetos para se obter o circuito global funcionando em sua plenitude.

A metodologia de integração de projetos didáticos com HDL e FPGA foi criada visando sistematizar e dinamizar as atividades de desenvolvimentos de tais projetos, envolvendo professores e alunos na execução das tarefas.

No que se refere à Educação, fundamenta-se no uso de abordagens de ensino com enfoque sócio-interacionista, com o objetivo de construir o conhecimento de forma socializada, a partir de propostas de situações-problema, cujas soluções estejam na sua maioria atrelada em atividades colaborativas.

Nesse contexto a metodologia se torna relevante porque os fabricantes disponibilizam apenas os recursos, mas não a forma específica de como aplicá-la, principalmente quando se deseja fazer uso de uma didática especial, usada como suporte didático-pedagógico.

Segundo Piletti [45], "a Didática estuda a técnica de ensino em todos os seus aspectos práticos e operacionais". Divide-se em Geral e Especial. Piletti acrescenta:

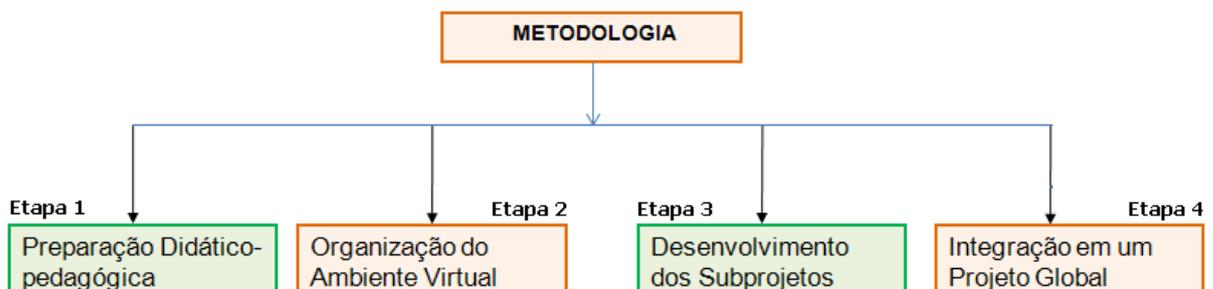
A Didática Geral estuda os princípios, as normas e as técnicas que devem regular qualquer tipo de ensino, para qualquer tipo de aluno. A Didática Geral nos dá uma visão geral da atividade docente.

A Didática Especial estuda aspectos científicos de uma determinada disciplina ou faixa de escolaridade. Analisa os problemas e as dificuldades que o ensino de cada disciplina apresenta e organiza os meios e as sugestões para resolvê-los.

Do ponto de vista tecnológico, neste trabalho foi adotado o ferramental de apoio do fabricante Altera, o qual fornece a placa DE2 para desenvolvimento de projetos com FPGA; e o software de desenvolvimento de projetos digitais Quartus II. Também é fornecido pelo fabricante um arquivo genérico (DE2\_TOP.v) contendo a configuração de todos os pinos do componente que foram associados à placa DE2,

os quais estão relacionados às interfaces de E/S, tais como chaves, leds, displays e outras interfaces para uso no projeto. Este fabricante também disponibiliza suporte didático aos professores que utilizarem este ferramental, no entanto não propõe qualquer método de apoio ao desenvolvimento de projetos como se propõe neste trabalho.

A metodologia sistematiza o uso da ferramenta de software e hardware de modo a se obter uma integração de projetos desenvolvidos de forma modular (subprojeto), por diferentes equipes de trabalho. Fundamenta-se em princípios da engenharia de software, contendo etapas, durante as quais são realizados processos, os quais têm artefatos como entradas (artefatos recebidos) e saídas (artefatos produzidos). Os processos por sua vez são realizados pelos atores (professores e alunos) com o auxílio de ferramentas computacionais. Aborda quatro etapas: (1) preparação didático-pedagógica dos módulos propostos; (2) organização do ambiente virtual de trabalho; (3) desenvolvimento dos subprojetos (módulos), e (4) integração dos subprojetos (módulos) em um projeto global integrado. A Fig. 2 ilustra as etapas que constituem a metodologia.



**Figura 2** - Etapas da Metodologia proposta

Para validar a metodologia proposta adotou-se como circuito modelo de teste a “Unidade de Controle com Interrupção”, usado em sala de aula pela professora Fátima Vieira [1]. No projeto original, tanto o código de descrição como o código de instanciação dos módulos foram criados dentro do módulo genérico DE2\_TOP.v, como mostra a fig. 3.

```

DE2_TOP.v

/*-- início do projeto genérico DE2_TOP--*/
module DE2_TOP

//< código do projeto genérico DE2_TOP>

/*****INÍCIO DO CÓDIGO DO PROJETO PROPOSTO*****/

//campo de edição do código de descrição e instanciamento dos módulos do projeto proposto

/*****FIM DO CÓDIGO DO PROJETO PROPOSTO*****/

endmodule //fim do projeto genérico DE2_TOP + Projeto Proposto

```

**Figura 3** – Estrutura unificada para criação dos códigos de descrição/instanciação

A estrutura da fig. 3 não é adequada para aplicações da metodologia proposta neste trabalho, que pressupõe criar o código de descrição dos módulos independentes, por diferentes equipes de trabalho. Por isso, optou-se por uma estrutura onde o código de instanciação e o código de descrição dos módulos são criados em arquivos independentes. Nessa organização, cada equipe deve desenvolver seu subprojeto de acordo com os seguintes procedimentos básicos (acompanhe o fluxo na fig. 4, a partir da numeração abaixo):

1. criar uma pasta de trabalho com o nome do subprojeto;
2. consultar o diagrama de blocos do circuito do subprojeto<sup>14</sup>;
3. abrir projeto genérico DE2\_TOP.qpf e gerar arquivos .v para criar código de descrição do(s) módulo(s) dos subprojetos;
4. criar dentro do arquivo genérico DE2\_TOP.v o código de instanciamento;
5. criar procedimentos de teste de funcionalidade do subprojeto;
6. abrir projeto genérico DE2\_TOP.qpf para análise de sintaxe, compilação, e gravação na placa FPGA;
7. realizar os testes de funcionalidade na placa FPGA;
8. preencher quadro com código de instanciamento do subprojeto.

<sup>14</sup> O diagrama de blocos apresenta o fluxo e a conexão das interfaces internas (entre blocos), e externas (entre blocos e a placa FPGA).

Para se obter o projeto integrado é necessário:

9. criar uma pasta com o nome do projeto integrado;
10. consultar o diagrama de blocos do projeto global;
11. importar todos os arquivos do projeto genérico DE2\_TOP para a pasta do projeto integrado. Estão apresentados apenas os arquivos .v;
12. preencher, analisar e ajustar o quadro de instanciamento global;
13. copiar o código de instanciamento global dentro do arquivo genérico DE2\_TOP.v;
14. importar de cada equipe os arquivos .v (descrição dos módulos) p/ a pasta do projeto integrado;
15. criar quadro de teste de funcionalidade global;
16. abrir projeto genérico DE2\_TOP.qpf para análise de sintaxe, compilação, e gravação na placa FPGA.
17. realizar os testes de funcionalidade.

No final, a pasta do referido projeto deve conter os seguintes arquivos de descrição dos módulos (extensão .v), arquivo genérico (DE2\_TOP.v) com o código de instanciamento global, e o arquivo de projeto DE2\_TOP.qpf, além dos demais arquivos gerados pelo software de desenvolvimento, após a compilação do projeto.

A seguir serão detalhadas as etapas de utilização da metodologia proposta, a partir de um quadro explicativo do fluxo de execução. Os procedimentos e o projeto são os mesmos utilizados nos testes de validação.

### 3.2 SÍNTESE DO PROCESSO DE INTEGRAÇÃO DOS SUBPROJETOS

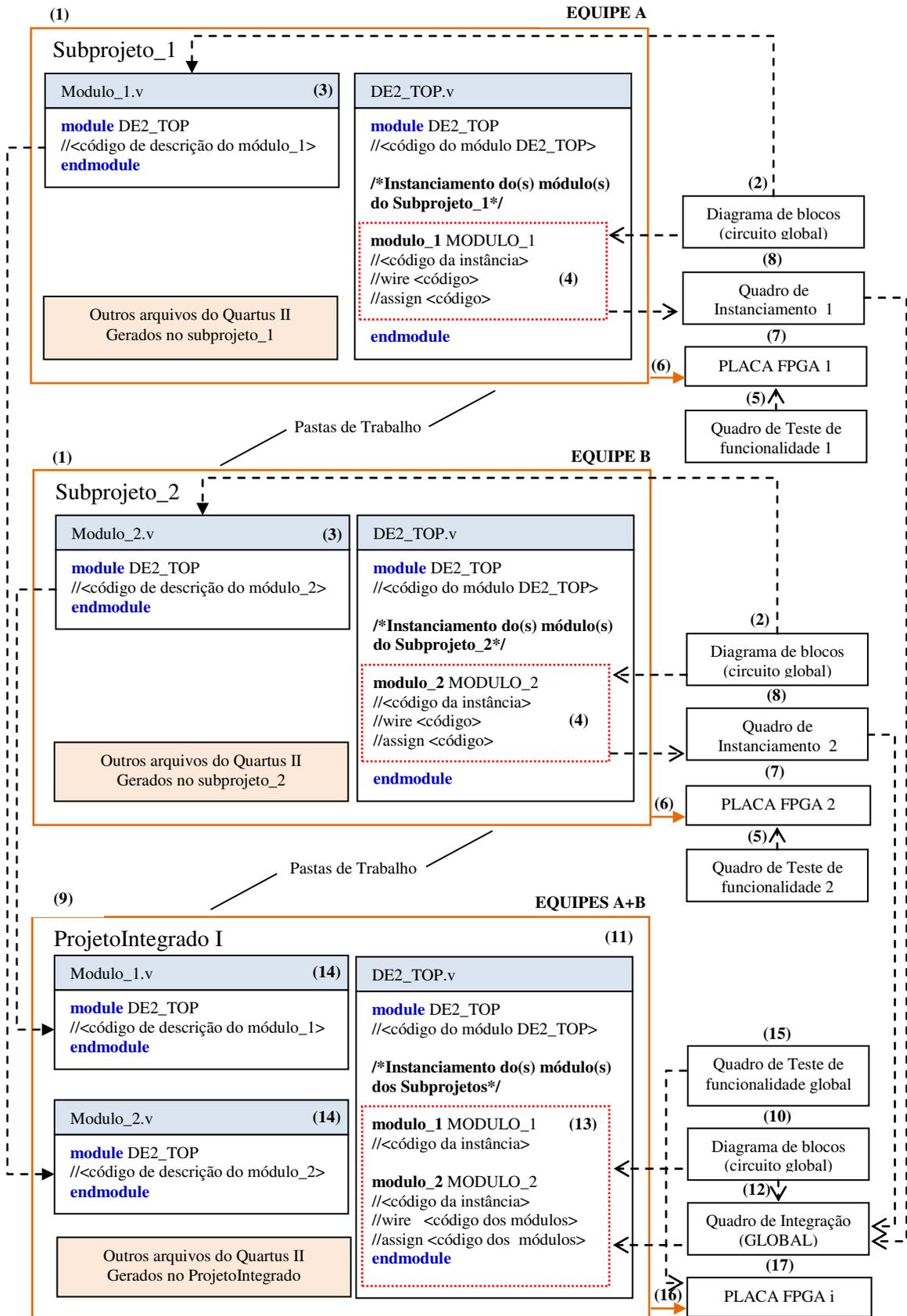


Figura 4 – Criação de um projeto Integrado a partir de dois Subprojetos

### 3.3 ETAPA 1 – PREPARAÇÃO DIDÁTICO-PEDAGÓGICA

A etapa de preparação didático-pedagógica consiste na concepção do projeto global e sua subdivisão em módulos que serão alocados às equipes de projeto para desenvolvimento e posterior integração. No quadro 3 temos os detalhes de execução da etapa 1.

**Quadro 3** - Parte didático-pedagógica

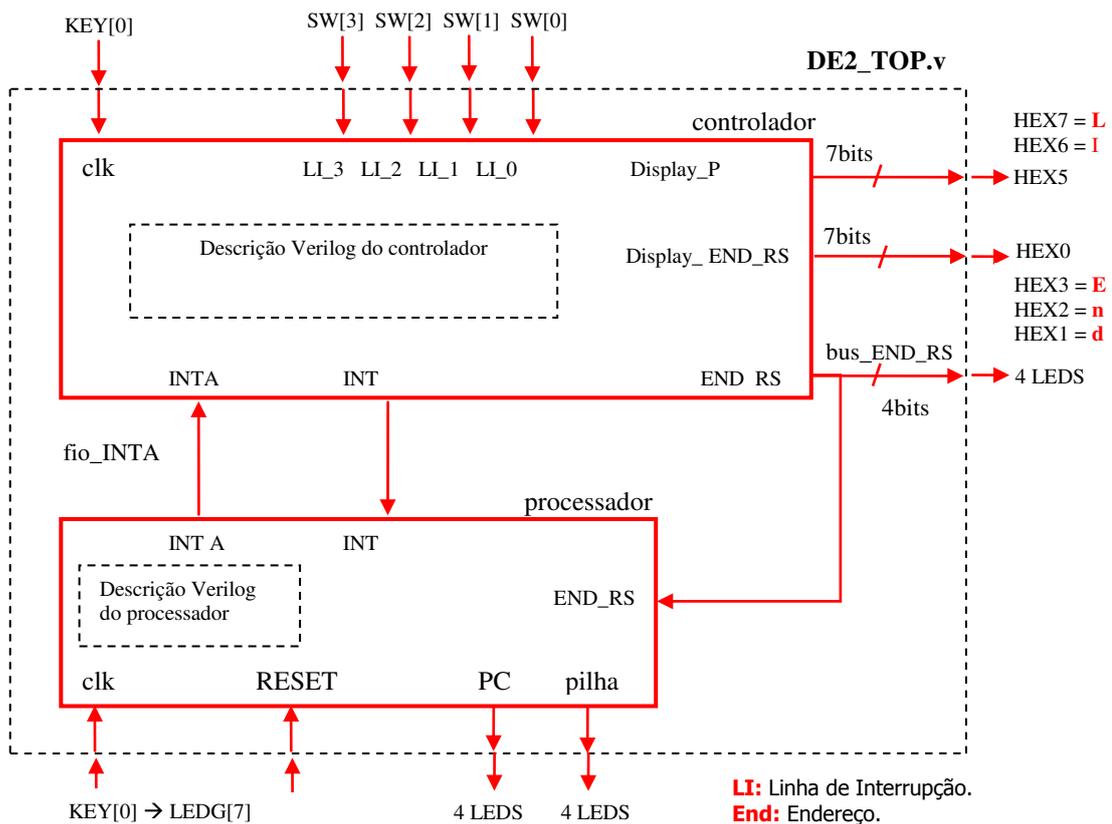
SUBETAPAS	PROCESSOS	TAREFAS	ATOR
1 Planejamento	1.1 Elaboração do plano de aula 1.2 Criação do guia e roteiro de atividades	<b>T1</b> - Elaborar plano de aula para um processo de ensino-aprendizagem, com ênfase na abordagem sócio-interacionista. <b>T2</b> – Criar guia e roteiro de atividades para o aluno	Professor
2 Diretrizes de trabalho	2.1 Definição das equipes 2.2 Elaboração do contrato Didático <sup>15</sup>	<b>T3</b> – Definir as equipes de trabalho: papéis e responsabilidades <b>T4</b> – Estabelecer as diretrizes do contrato didático.	Professor e Alunos
3 Requisitos do projeto	3.1 Definição dos requisitos do projeto	<b>T5</b> – Estabelecer os requisitos do projeto. A análise dos requisitos deve garantir uma relação de compatibilidade entre a proposta inicial e o produto final, com o mínimo de distorção. Algumas questões que podem ser levantadas: Qual a finalidade do projeto? Quem serão os usuários? Quais são as variáveis de entrada, e que resposta pretende-se obter a partir dessas variáveis?  <b>T6</b> – Dividir o circuito em partes para distribuir entre as equipes. Cada parte do circuito constituirá subprojetos que deverão ser desenvolvidos pelas equipes. Dependendo da quantidade de módulos e da complexidade do projeto uma equipe poderá receber uma ou mais partes para desenvolvimento do projeto. Essa é uma análise e decisão do professor.  <b>T7</b> – Definir as principais interfaces dos módulos que deverão compor o projeto. Essas interfaces devem ser indicadas no diagrama em blocos do circuito global.	Professor e Alunos

<sup>15</sup> Contrato didático: é uma discussão e reflexão sobre as diretrizes que nortearão o andamento do curso em sala de aula: tipo de abordagem do ensino, horário, formas de avaliação, normas de uso do Laboratório, entre outras diretrizes.

A Etapa de preparação didático-pedagógica (Quadro 3) consiste na ação do professor. Nesse estudo não será apresentado nenhum modelo de plano de aula, nem detalhes sobre o contrato didático, porque poderão ser utilizados os modelos das instituições em que os mesmos se encontram vinculados.

### Projeto Unidade de Controle com Interrupções

Adotou-se como circuito modelo de teste a “Unidade de Controle com Interrupção”, usado em sala de aula pela professora Fátima Vieira [1], na disciplina Arquitetura de Sistemas Digitais, no Curso Engenharia Elétrica da UFCG. Conforme a fig. 5, o circuito completo é composto por um processador onde se encontra a unidade de controle, e um controlador de interrupções.



**Figura 5** - Diagrama em blocos do circuito global

No projeto original precisou-se de duas placas de desenvolvimento, interligadas via *flatcable* – *GPIO\_0*, com o objetivo de aumentar os recursos de chaves e leds. Nesse estudo, reduziu-se o tamanho das palavras digitais originais de 1 byte para 1 nibble<sup>16</sup>, de tal forma que será necessário apenas uma placa de desenvolvimento.

## Descrição geral do projeto

O projeto foi dividido em duas partes. Cada parte constituirá um subprojeto que será desenvolvido por duas equipes de trabalho: Equipe A - subprojeto do módulo controlador; Equipe B – Subprojeto do módulo Processador.

### *O Controlador de Interrupção*

Dentre os tipos de controle de interrupções apresentados no projeto original, optou-se pelo controle com pesquisa por varredura. Na pesquisa por varredura, o controlador de interrupção testa em uma seqüência pré-definida, cada linha de interrupção - LI, que vai de LI\_0 a LI\_3 até localizar o dispositivo externo, representados pelas chaves SW0 a SW3, que solicitou a interrupção. Além disso, cada dispositivo está associado a um registrador que armazena o endereço da rotina de serviço de determinada interrupção, enviada através de sua respectiva linha LI, como mostra a tabela 1.

**Tabela 1** - Elementos associados ao pedido de interrupção.

Dispositivos Externos	Chaves representativas dos dispositivos	Linhas de Interrupção	Registrador RL associado A cada dispositivo	Endereços das Rotinas de Serviço END_RS, armazenados nos respectivos RL. São apresentados Valores em binário e em hexadecimal
3	SW[3]	LI_3	RL3	4'b1010 = AH
2	SW[2]	LI_2	RL2	4'b1011 = BH
1	SW[1]	LI_1	RL1	4'b1100 = CH
0	SW[0]	LI_0	RL0	4'b1101 = DH

<sup>16</sup> O nibble é uma palavra de quatro bits.

A prioridade dos dispositivos é dada pela seqüência de teste das linhas. Sendo assim, o controlador inicia a varredura a partir do primeiro dispositivo, e segue na seqüência. Utilizou-se o registrador P para armazenar o endereço do próximo dispositivo a ser verificado. A seqüência e cada dispositivo correspondente são mostrados na tabela 2.

**Tabela 2** - Reg. de endereço do próximo dispositivo de entrada de pedido de INT

Registrador P		Dispositivos → chaves
bit1	bit0	x
0	0	0 → SW[0]
0	1	1 → SW[1]
1	0	2 → SW[2]
1	1	3 → SW[3]

Quando houver pedidos originados de um ou mais dispositivos, o controlador reconhecerá os pedidos, mas só repassará ao processador **o pedido da próxima linha**, na seqüência de varredura. Após atender ao pedido de um dispositivo, o controlador armazena no registrador reg\_P o endereço do próximo dispositivo que será verificado na seqüência de varredura.

### *O Processador*

O pedido de interrupção é reconhecido pelo processador (unidade de controle UC), através do sinal *interrupt request (INT)*. Quando o processador habilita o pedido, envia ao controlador um sinal *interrupt acknowledge (INTA)*, no nível alto. O controlador de interrupção responde enviando o endereço da rotina de serviço END\_RS associada ao dispositivo cuja interrupção está sendo atendida.

Para que a execução do programa não seja descontinuada, o processador coloca na PILHA o conteúdo atual do contador de programa (PC) e desvia para a rotina de serviço, cujo endereço foi repassado pelo controlador.

Ao término do atendimento à interrupção, o processador envia ao controlador o sinal *INTA*, agora no nível baixo, sinalizando fim do referido atendimento, e que está disponível para a próxima transferência. Um *clock* após isto, o conteúdo da pilha = [conteúdo do PC, antes da interrupção] é restaurado para que o programa volte do ponto onde havia parado. Ao receber este sinal o controlador volta a realizar a

varredura dos dispositivos a partir do canal cujo endereço está armazenado no registrador `reg_P`.

### *Os indicadores*

Os endereços da Rotina de Serviço de Interrupção – `END_RS` devem ser apresentados simultaneamente em hexadecimal, e em binário, através de displays e LEDs, respectivamente. Antes do display indicador de `END_RS` configurar os displays `HEX3`, `HEX2`, e `HEX1` para formar a palavra fixa "End" seguido do `HEX0` onde serão mostrados os respectivos endereços.

Os conteúdos do PC e da pilha devem ser mostrados em binário através de seus respectivos LEDs.

A sequência de amostragem das linhas de interrupção – `LI` deve ser apresentada no display `HEX5`. A palavra fixa "LI" deve ser configurada com os displays `HEX7` e `HEX6`. Lembrar que a sequência de `LI` é dada pelo gerador de varredura de linha P.

O relógio *clk* deve ser sinalizado pelo led `LEDG[7]` toda vez que houver uma transição em `KEY[0]`.

### 3.4 ETAPA 2 – ORGANIZAÇÃO DO VIRTUAL AMBIENTE DE TRABALHO

A etapa de organização do ambiente virtual refere-se à criação e estruturação de diretórios e pastas de trabalho onde serão colocados os arquivos produzidos e utilizados ao longo do desenvolvimento do projeto. No quadro 4 temos o detalhe de execução da referida etapa.

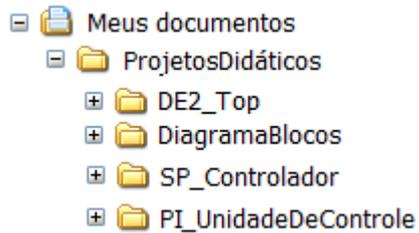
**Quadro 4 - Organização do ambiente virtual de trabalho**

SUBETAPAS	PROCESSOS	TAREFAS	ATOR
4 Preparação	<p>4.1 Verificação da versão do software de desenvolvimento</p> <p>4.2 Criação das pastas de trabalho</p>	<p><b>T8</b> – Verificar a versão do Software de desenvolvimento<sup>17</sup> para avaliar seus recursos e limitações.</p> <p><b>T9</b> - Criar pasta de trabalho Criar no computador de trabalho, em "Meus documentos", uma pasta denominada "Projetos Didáticos". Essa pasta deve ser usada para colocar todos os materiais necessários ao desenvolvimento das atividades.</p> <p><b>T10</b>- Localizar a pasta DE2_Top e copiá-la na pasta Projeto Didáticos. A pasta DE2_Top contém o projeto DE2_TOP.qpf e o arquivo DE2_TOP.v com a descrição de todos os recursos da placa DE2, além da configuração com os pinos do chip FPGA da referida placa. Ela será usada como base para o desenvolvimento de todos os projetos e subprojetos.</p> <p><b>T11</b> - Copiar e renomear a pasta DE2_Top com o nome do Subprojeto SP que será desenvolvido: SP_&lt;nome&gt;.</p> <p><b>T12</b> - Copiar e renomear a pasta DE2_Top com o nome do projeto PI que será integrado a partir dos subprojetos: PI_&lt;nome&gt;.</p> <p><b>T13</b> - Criar uma pasta denominada de "Diagrama Blocos" para guardar os arquivos com os esquemas dos circuitos projetados e em desenvolvimento.</p>	Alunos

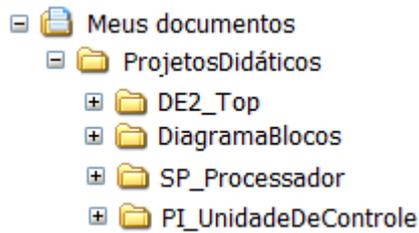
<sup>17</sup> Neste estudo usou-se o software Quartus II - versão 8.1 e a placa FPGA de desenvolvimento DE2, ambos do fabricante Altera.

### *Procedimentos*

Execute as tarefas **T8** a **T13** do Quadro 4, sem considerar o uso da ferramenta de apoio comentada em nota, na descrição da tarefa T11. O resultado é mostrado nas figuras 6 e 7.



**Figura 6** – Organização das pastas de trabalho da equipe A.



**Figura 7** – Organização das pastas de trabalho da equipe B.

### 3.5 ETAPA 3 – DESENVOLVIMENTO DOS SUBPROJETOS

A etapa de desenvolvimento dos subprojetos funda-se no trabalho individual das equipes voltado para a construção e teste dos respectivos subprojetos, porém visando a posterior integração.

Subprojeto é o projeto de uma das partes de um projeto de um circuito global, que deverá ser desenvolvido e testado por diferentes equipes. Depois dessa etapa, os referidos subprojetos são integrados para forma o circuito original, com sua funcionalidade preservada. No quadro 5 é apresentado os detalhes de execução da etapa de desenvolvimento de um subprojeto. No item 3.5.1 é apresentado o desenvolvimento do subprojeto controlador, utilizado na validação da metodologia proposta, com base no quadro 5.

**Quadro 5 - Desenvolvimento dos subprojetos**

SUBETAPAS	PROCESSOS	TAREFAS	ATOR
5 Desenvolvimento de Subprojetos	5.1 Criação do código de descrição do(s) módulo(s) do subprojeto.  Nota: consultar diagrama em blocos do circuito global.	<p><b>T14</b> - Criar a partir do Quartus II, um arquivo .v para cada módulo do subprojeto. O arquivo.v é o local onde o código de descrição do(s) respectivo(s) módulo(s) será criado. Um caminho para criar o referido arquivo no Quartus II: <i>File→New→Verilog HDL</i> <i>File→OK→File→Save As→&lt;nome do módulo&gt;→&lt;pasta do subprojeto&gt;→Salvar.</i></p> <p><b>T15</b> – Criar o código de descrição do(s) módulo(s) nos respectivos Arquivos .v. <b>T16</b> – Fazer análise de sintaxe<sup>18</sup> para cada módulo. Um caminho para análise de sintaxe no Quartus II: <i>Processing→Analyze current File</i></p>	Alunos
	5.2 Criação do código de Instanciação dos módulos do Sub(projeto)  Nota: consultar diagrama em blocos do circuito global.	<p><b>T17</b> – Abrir o software de desenvolvimento Quartus II <b>T18</b> – Abrir através do Quartus II, o projeto genérico DE2_TOP.qpf que se encontra na pasta do sub(projeto) que está sendo desenvolvido. <b>T19</b> – Abrir o editor de texto de DE2_TOP que se encontra no Quartus II, e localizar a linha de <i>endmodule</i>. <b>T20</b> – Definir a área de edição do</p>	

<sup>18</sup> Sintaxe: forma de escrita do código. Lembrar que a linguagem Verilog diferencia maiúsculas de minúsculas.

SUBETAPAS	PROCESSOS	TAREFAS	ATOR
		<p>código de Instanciação do(s) subprojeto(s).</p> <p>- Iniciar antes da linha <i>endmodule</i>, dentro de DE2_TOP.v, da seguinte forma:</p> <pre> /****INÍCIO DO CÓDIGO****/ /*Área de criação do código de Instanciação dos módulos do(s) subprojeto(s)*/ /*****FIM DO CÓDIGO*****/ <b>endmodule</b> </pre> <p><b>T21</b> – Criar o código de Instanciação do sub(projeto) com auxílio do diagrama em blocos do circuito.</p> <p>- Nesse diagrama encontram-se as interfaces básicas de integração definidas pelo professor e pelos alunos durante a análise dos requisitos do projeto.</p>	
	<p>5.3 Análise de Sintaxe</p> <p>Nota: consultar diagrama blocos do circuito global.</p>	<p><b>T22</b> – Fazer análise de sintaxe.</p> <p>- Um caminho para análise de sintaxe no Quartus II: <i>Processing</i> → <i>Analyze current File</i></p>	Alunos
	<p>5.4 Compilação do sub(projeto)</p> <p>Nota: consultar diagrama blocos do circuito global.</p>	<p><b>T23</b> – Compilar o código do sub(projeto)</p> <p>- Um caminho para compilação no Quartus II: <i>Processing</i> → <i>File</i> → <i>Start Compilation</i></p> <p>- Lembrar que Erros de lógica não são detectados na compilação.</p>	Alunos
	<p>5.5 Criação do Quadro de síntese de funcionamento</p>	<p><b>T24</b> - Criar um quadro de síntese de funcionamento do circuito projetado no sub(projeto).</p> <p>- Usar as chaves da Placa DE2 como elemento de excitação de entrada. Os Leds e Displays serão utilizados como elementos de saída - respostas às entradas.</p>	
	<p>5.6 Gravação do código do sub(projeto) na Placa FPGA</p>	<p><b>T25</b> – Gravar o sub(projeto) compilado no chip FPGA da Placa de Desenvolvimento.</p> <p>- Lembrar que o erro de lógica não causam problemas de gravação. No entanto, comprometerá o funcionamento do circuito.</p>	Alunos
	<p>5.7 Verificação da funcionalidade do circuito projetado.</p>	<p><b>T26</b> – Fazer o teste de funcionalidade do circuito projetado no sub(projeto)</p> <p>- Consultar o quadro de Testes de funcionalidade do circuito (T24). Entradas → são representadas por chaves SW e/ou KEY Saídas: LED´s e/ou Displays.</p>	Alunos
	<p>5.8 Preenchimento do Quadro de integração do subprojeto.</p> <p>Nota: consultar diagrama blocos do circuito global.</p>	<p><b>T27</b> - Preencher o quadro com o código de instanciação dos módulos do subprojeto</p>	Alunos

### 3.5.1 Subprojeto Controlador

Executor: Equipe A

Módulo(s): Controlador de Interrupção

Função: Recebe o pedido de interrupção e envia o sinal de INT para o processador.

Armazena no reg END\_RS o endereço da rotina de serviço de interrupção (END\_RS)

#### 3.5.1.1 Código de descrição do módulo controlador

O código de descrição de um módulo deve ser criado à parte, num editor de texto, como arquivo de extensão .v. Ele deve ser construído a partir dos requisitos do projeto. Nesse contexto, o diagrama em blocos do circuito global (vê fig. 7) torna-se o artefato essencial de apoio ao projetista pelos seguintes motivos:

- ✓ apresenta interfaces definidas na análise de requisitos;
- ✓ dar uma visão sistêmica do projeto em desenvolvimento;
- ✓ auxilia na criação e manutenção dos códigos;
- ✓ auxilia na elaboração do diagrama em blocos do subprojeto.

#### *Procedimentos para criação do código de descrição do Subprojeto Controlador*

Execute as tarefas **T14** a **T16** do Quadro 5. O resultado é o código de descrição abaixo:

```
module controlador
(
    LI_0, LI_1, LI_2, LI_3,
    RL0, RL1, RL2, RL3,
    clk,
    INT, controle,INTA,
    P,END_RS,
    display_P,
    display_END_RS,
);
//Definicao das portas
input
    clk,
    LI_0, LI_1, LI_2, LI_3,
    INTA
;
output INT, controle;
output [1:0]P;
output [3:0]RL0;
output [3:0]RL1;
output [3:0]RL2;
output [4:0]RL3;
output [4:0]END_RS;
output [6:0]display_END_RS;
```

```

output [6:0]display_P;
//Elementos Internos: definição dos fios

wire clk;
wire LI_0, LI_1, LI_2, LI_3,INTA;

//Elementos Internos: definição dos registradores

reg [3:0] RL0=4'b1010; //Registrador com o endereço da interrupção do dispositivo RL0
reg [3:0] RL1=4'b1011; //Registrador com o endereço da interrupção do dispositivo RL1
reg [3:0] RL2=4'b1100; //Registrador com o endereço da interrupção do dispositivo RL2
reg [3:0] RL3=4'b1101; //Registrador com o endereço da interrupção do dispositivo RL3
reg [1:0] P=2'b00; //Contador do controlador
reg [3:0] END_RS; //Reg que irá armazenar o end da rotina de interrupção
reg INT; //Indicador do pedido de interrupção
reg controle; //Indicador de tratamento de interrupção

reg [6:0]display_END_RS;
reg [6:0]display_P;

//Descrição do módulo

always@(posedge clk)

begin
  if(!INT) //se não houver interrupções, o P é adicionado de um a cada pulso de clock
    P=P+2'b01;

    case(P)
      2'b00: if(LI_0)// Se P=00 e RL0 gera uma interrupção entra na condição
        begin
          END_RS=RL0; //Faz o registrador saída igual ao RL0
          INT = 1;//Ativa o pedido de interrupção

        end
      2'b01: if(LI_1) // Se P=01 e RL1 gera uma interrupção entra na condição
        begin
          END_RS=RL1; //Faz o registrador saída igual ao RL1
          INT =1;//Ativa o pedido de interrupção

        end
      2'b10: if(LI_2) // Se P=10 e RL2 gera uma interrupção entra na condição
        begin
          END_RS=RL2; //Faz o registrador saída igual ao RL2
          INT =1;//Ativa o pedido de interrupção

        end
      2'b11: if(LI_3) // Se P=11 e RL3 gera uma interrupção entra na condição
        begin
          END_RS=RL3; //Faz o registrador saída igual ao RL3
          INT =1;//Ativa o pedido de interrupção

        end
    end
  endcase
end

```

```

        case(END_RS)
        4'b0000:display_END_RS=7'b1000000;
        4'b1010:display_END_RS=7'b0001000;
        4'b1011:display_END_RS=7'b0000011;
        4'b1100:display_END_RS=7'b1000110;
        4'b1101:display_END_RS=7'b0100001;
        endcase

        case(P)
        2'b00:display_P=7'b1000000;//0
        2'b01:display_P=7'b1111001;//1
        2'b10:display_P=7'b0100100;//2
        2'b11:display_P=7'b0110000;//3
        endcase

if(!LI_3 && !LI_2 && !LI_1 && !LI_0)
begin
INT=0;
display_END_RS=7'bzzzzzzz;
display_P=1'bz;
P=1'bz;
END_RS=4'bzzzz;
end

if(INTA) //Se INTA em nível alto, ativa o indicador de tratamento de interrupção
    controle=1;

if(!INTA)
begin
if(controle) // Se o indicador de tratamento de interrupção quando INTA em nível baixo
begin
INT=0;// Desativa o INTA
controle=0; //Desativa o indicador de tratamento de interrupção
end
end
end
endmodule

/*----- fim da função -----*/

```

### 3.5.1.2 Código de instanciação do módulo controlador

O código de instanciação dos módulos de um subprojeto deve ser criado no mesmo arquivo onde se encontra o código genérico DE2\_TOP do subprojeto em desenvolvimento. Ele deve ser construído a partir dos requisitos do projeto, sob consulta do diagrama em blocos do circuito global.

*Procedimentos para criação do código de instanciação do subprojeto controlador*

Execute as tarefas **T17** a **T21** do Quadro 5. O resultado é o código de instanciação dos módulos do subprojeto criado, e compilado na sua totalidade, integrado ao respectivo módulo. Vê código abaixo.

```

/*****INÍCIO DO CÓDIGO DE INTEGRAÇÃO*****/
398
399 //INSTÂNCIAS
400
401 Controlador CONTROLADOR //nome do módulo + nome da instância do módulo
402 ( //início da instância
403 .LI_3(SW[3]), //usa chave SW[3] p/gerar pedido interrupção linha L3
404 .LI_2(SW[2]),
405 .LI_1(SW[1]),
406 .LI_0(SW[0]),
407 .Display_END_RS(HEX0), //Usa display 0 p/mostrar end de rotina de serviço INT
408 .END_RS(LED[R[10:7]]), //Usa 4 leds vermelhos para mostrar nibble do END_RS
409 .Display_P(HEX5), //Usa display 5 p/mostrar a varredura do reg END_RS
410 .INT(LED[R[15]]), //Usa led vermelho 15 p/sinalizar pedido de interrupção
411 .INTA(SW[17]), //Usa chave SW[17] p/ comando manual do sinal INTA
412 .clk(KEY[0]), //Usa chave KEY[0] como gerador de clock do sistema
413 ); //fim da instância do módulo CONTROLADOR
414
415 //wire <não foi usado neste subprojeto>
416 assign LED[R[17]]=SW[17]; //Usa LED verde 17 p/sinalizar o estado da chave SW17
417
418 /*****FIM DO CÓDIGO DE INTEGRAÇÃO *****/
419
420 endmodule //fim do código DE2_TOP + Instância do módulo CONTROLADOR

```

Observe que o código de DE2\_TOP.v foi omitido porque se encontra descrito entre as linhas 1 e 398.

**Nota:** *para evitar conflito de multiplicidade de uso de componentes da placa é necessário comentar no código do projeto genérico o(s) componente(s) descrito(s) simultaneamente no projeto genérico e no projeto em desenvolvimento.*

3.5.1.3 Diagrama em blocos – módulo controlador

O diagrama em blocos do módulo controlador, desenvolvido pela equipe A, é mostrado na fig. 8. A descrição de funcionamento do mesmo é feita no item 3.5.1.4, logo abaixo.

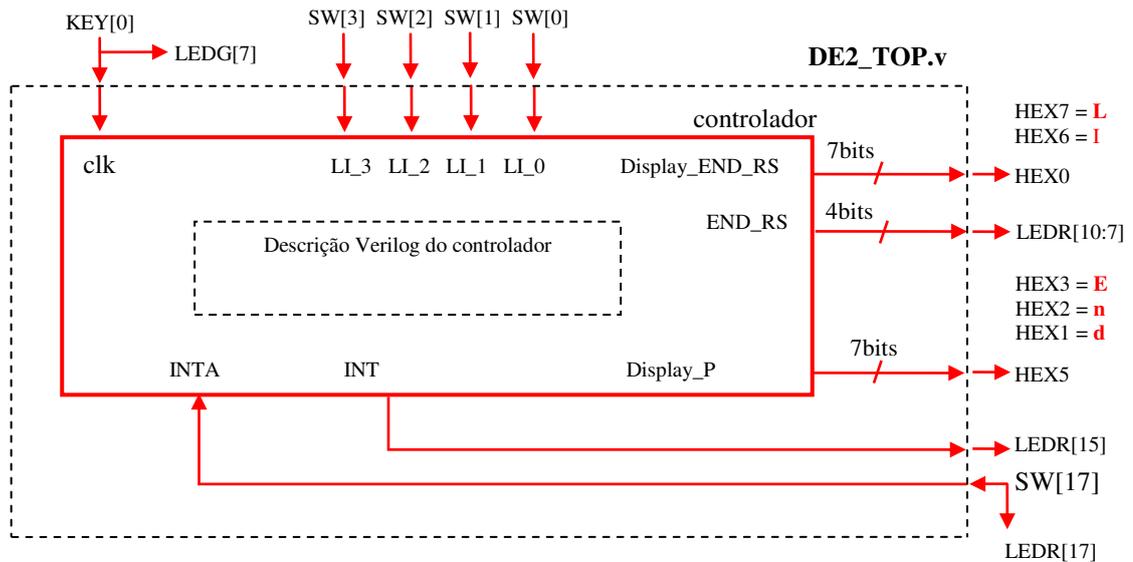


Figura 8 - Módulo controlador e suas interfaces: diagrama em blocos.

3.5.1.4 Quadro de teste de funcionalidade

No quadro 6 é apresentado o estado lógico inicial das interfaces de entrada e saída do módulo controlador.

Quadro 6 – Estado lógico das interfaces para teste do módulo controlador

Clk	LI = SW[3:0]	LI = P	INT	INTA	END_RS	PC	Pilha	LI = P
KEY[0] e LEDG[7]	SW[3:0]	HEX5	LEDR[15]	SW[17] e LEDR[17]	LEDR[10:7] e HEX0	LEDR[3:0]	LEDG[7:0]	HEX5
0 e 0	0000	7'bzzzzzzz	0	0 e 0	4'bzzzzz e 7'bzzzzzzz	0000	0000	7'bzzzzzzz

Legenda do quadro:

chave KEY: 0 → não houve transição; 1 → houve transição.

chave SW: 0 → off = posição para baixo; 1 → on = posição para cima.

LEDs: 0 → apagado; 1 → aceso. Exceto LEDG[7], que é indicador de transição clk  
z = terceiro estado → alta impedância

Procedimentos

Execute as tarefas **T22** a **T26** do Quadro 5. Serão analisadas duas situações. O passo a passo é apresentado a seguir.

## *SITUAÇÃO 1 - ANÁLISE PARA UM ÚNICO PEDIDO DE INTERRUPÇÃO*

### **Primeiro passo**

- i. Antes de iniciar certifique-se de que todas as chaves estão na posição inferior  $SW[17:0]=0$ .
- ii. No momento em que o código é gravado na placa DE2 observe se os estados das principais interfaces comportam-se como mostrado abaixo, no quadro de funcionamento do circuito:
- iii. Enquanto não houver pedido de interrupção  $INT=0$  e  $INTA=0$ , a partir do primeiro pulso de  $clk$ :  $PC=PC+1$ . Os demais permanecem inalterados. Verifique isso atuando na chave  $KEY[0]$ .

### **Segundo passo**

- i. Simule um pedido de interrupção fazendo  $SW[3]=1$ . Logo em seguida gere uma transição de  $clk$  atuando em  $KEY[0]$ . Observe no display HEX5, que nesse momento o indicador de linha inicia a varredura em  $LI=0$ .
- ii. Continue atuando em  $KEY[0]$  até a varredura encontrar a linha  $LI=3$ , que solicitou a interrupção. No momento em que a referida linha for encontrada pela varredura ocorrerá o seguinte: será mostrado o endereço da rotina de serviço dessa interrupção:  $END\_RS=DH$  no display HEX5 e, ao mesmo tempo, 1101 nos LEDs  $LEDR[10:7]$ .

Lembrar que os endereços das respectivas rotinas de serviços de interrupção  $END\_RS$  foram estabelecidos pelo projetista e podem ser visualizadas simultaneamente em hexadecimal, no display HEX0, ou em binário, nos LEDs  $LEDR[10:7]$ :

**$END\_RS[0]$**  = AH em hexadecimal, ou **1010** em binário.  
 **$END\_RS[1]$**  = BH em hexadecimal, ou **1011** em binário.  
 **$END\_RS[2]$**  = CH em hexadecimal, ou **1100** em binário.  
 **$END\_RS[3]$**  = DH em hexadecimal, ou **1101** em binário.

- iii. Enquanto  $INTA=0$ , o  $PC=PC+1$ , mas  $LI = 3$ , até o pedido de interrupção  $LI[3]$  ser completamente atendido. Verifique isso atuando em  $KEY[0]$ .

### **Terceiro passo**

- i. Faça  $INTA=1$  atuando em  $SW[17]$ . Nesse momento o led  $LEDR[17]$  acende para sinalizar que a ação foi realizada com sucesso. Após o próximo pulso de  $clk$  ocorrerá o seguinte: o endereço 1101 da rotina de serviço correspondente a linha de interrupção  $LI[3]$  será transferido para o PC. A partir daí o sistema irá "travar" por completo, mesmo que você continue a gerar os pulsos de  $clk$ , porque a continuidade dessa operação é feita pelo processador. O desenvolvimento do subprojeto processador é o objeto de estudo da equipe B.
- ii. Faça  $INTA=0$ , e  $SW[3:0]=0$ . Logo em seguida dê um pulso de  $clk$  via  $KEY[0]$ , e caso deseje repita o processo a partir do segundo passo, senão passe para a análise seguinte.

### *SITUAÇÃO 2 - ANÁLISE PARA MAIS DE UM PEDIDO SIMULTÂNEO*

- i. Após reiniciar o processo no item anterior 3.ii com  $INTA=0$  e  $SW[3:0]=0$ , ative duas ou mais linhas de interrupção com auxílio das chaves  $SW[3:0]$ .
- ii. Dê pulsos de  $clk$  com a chave  $KEY[0]$  e observe a mudança do gerador de varredura de  $LI$ . Quando a primeira linha for encontrada na ordem de varredura, o contador irá parar até que a referida linha seja atendida com  $INTA=1$ .
- iii. Com  $INTA=1$ , após transição de  $clk$ , o endereço da rotina de serviço da linha selecionada será enviado para o PC. Como já foi dito no item 3.i, o sistema irá travar.
- iv. Desabilite a linha que já foi atendida. Faça  $INTA=0$ , e logo em seguida, continue a gerar os pulsos de  $clk$ . Com isso, na devida sequência, a próxima linha deverá ser atendida do seguinte modo: faça  $INTA=1$  atuando em  $SW[17]=1$ . Depois, dê um pulso de  $clk$ ... E assim por diante, até o último atendimento.



## COMO PREENCHER UM QUADRO DE INSTANCIAMENTO DE SUBPROJETOS?

**Primeiro passo:** a partir do código de descrição dos módulos e do diagrama em blocos do circuito do projeto inicie a relação das interfaces internas que serão instanciadas com as interfaces externas, através de ligações diretas ou via código *wire*:

conexão direta: vê linhas  $n = 1$  a  $n = 10$ .

conexão através de *wire*: não utilizada

conexão através de *assign*: vê linhas  $n = 11$  a  $n = 15$ .

**Segundo passo:** fazer a relação dos comandos do tipo *wire*. Não se utilizou nesse exemplo.

A escolha do nome de um *wire* é arbitrária. Fica por conta da lógica do projetista. No entanto, recomenda-se que se use um nome que lembre a que se destina o referido barramento *bus*. Deve-se usar o mesmo procedimento para se criar um *wire* do tipo fio<sup>19</sup>.

Um *wire* é necessário todas as vezes que se deseja interligar interfaces internas, seja de uma interface interna para outra interface interna, ou de uma interface interna para uma interface externa<sup>20</sup>.

**Terceiro passo:** fazer a relação dos comandos do tipo *assign*. Vê item  $n = 11$  a 16, do Quadro 8.

## INFORMAÇÕES COMPLEMENTARES SOBRE *assign*

Um *assign* é necessário todas as vezes que:

a) cria-se *wire* e se deseja mostrar seus dados através de leds:

`assign LEDR[3:0]=bus_saida_1.`

<sup>19</sup> fio: acopla sinais de um bit. Barramento ou bus: acopla sinais de  $n$  bits. Exemplo: `wire [3:0] bus_saida_1` representa um barramento de quatro bits.

<sup>20</sup>As Interfaces internas são aquelas criadas pelo projetista. São interfaces específicas do projeto em desenvolvimento.

As interfaces externas são as interfaces disponíveis na placa FPGA, tais como LEDs (`LEDR[i]`, `LEDG[i]`), Displays(HEXi) e chaves(`KEY[i]`, `SW[i]`).



### *Procedimento para preenchimento do mapa (vê quadro 9)*

A partir do quadro de integração, ou do diagrama em blocos de um subprojeto marcar na cor amarela a quadrícula que indica a interface usada. As interfaces que forem usadas mais de uma vez devem ser marcadas na cor vermelha para posteriormente ser analisado se houve multiplicidade indevida de uso. Lembrar que em nenhuma hipótese pode haver multiplicidade de uso de LEDs e displays. Porém, uma mesma chave pode ser usada mais de uma vez, em instâncias de módulos diferentes, desde que possuam a mesma função dentro do projeto.

Exemplo – no mapa do quadro 9 existem disponíveis na placa FPGA 1 displays HEX4,  $i=4$ ; 8 leds vermelhos LEDR[ $i$ ],  $i = 4$  a 6, 11 a 14 e 16; 7 leds verdes LEDG[ $i$ ],  $i = 0$  a 6; 13 chaves SW[ $i$ ],  $i = 4$  a 16; 3 chaves KEY[ $i$ ],  $i = 1, 2$  e 3.

### **3.5.2 Subprojeto Processador**

Executor: Equipe B

Módulo(s): Processador

Função do módulo Processador: controlar o pedido de interrupção.

Usam-se os mesmos procedimentos de desenvolvimento do Subprojeto Controlador. A seguir, além do diagrama em blocos, é apresentada uma versão completa do código de descrição do módulo processador, o código de instanciação e o quadro de integração.

#### *3.5.2.1 Código de descrição do módulo processador*

```

module processador
(
    INT, clk, END_RS, INTA,
    PC, pilha, controle_x, RESET,
);

//Declaração das portas
input INT;
input clk;
input END_RS; //END_RI--> Endereço de Rotina de Interrupção
input INTA;
input RESET;

output PC;
output pilha;
output controle_x;

```

```

//Elementos internos
wire INT, clk, INTA;
wire [3:0] END_RS;
wire RESET;

reg [3:0] PC = 4'b0000; //Contador do processador
reg [3:0] pilha; //pilha
reg controle_x; //Registrador para controle

//Descrição do módulo
always@(posedge clk)

begin

    PC = PC + 4'b0001; //Faz com que o PC conte a cada pulso de clock

    if(RESET)
        begin
            PC=0;
            pilha=0;
        end

    if(INTA && INT && !controle_x)//Se o interrupt acknowledge for ativado entra na condição
        begin
            pilha=PC-1; //Salva o conteúdo do PC na pilha
            PC = END_RS; //representa dados de saída dos registradores
            controle_x=1; //Ativa o flag de controle
        end

    if(!INTA&INT&controle_x)
        begin
            controle_x=0; //Desativa o flag de controle
            PC=pilha; //PC recupera o conteúdo que possuía antes da interrupção
        end

end
endmodule

```

### 3.5.2.2 Código de instanciação do módulo processador

```

/*-----INÍCIO DO INSTANCIMENTO-----*/

//INSTÂNCIA - CONEXÕES
processador PROCESSADOR
(
    .INT(SW[17]),
    .INTA(fio_INTA),,
    .END_RS(SW[10:7]), // gera endereços das rotinas de interrupção para TESTES
    .clk(KEY[0]),
    .PC(LED[3:0]), // sinalização dos dados do PC.
    .pilha(LEDG[3:0]), // sinalização dos dados da PILHA temp.
    .RESET(SW[14]),
);

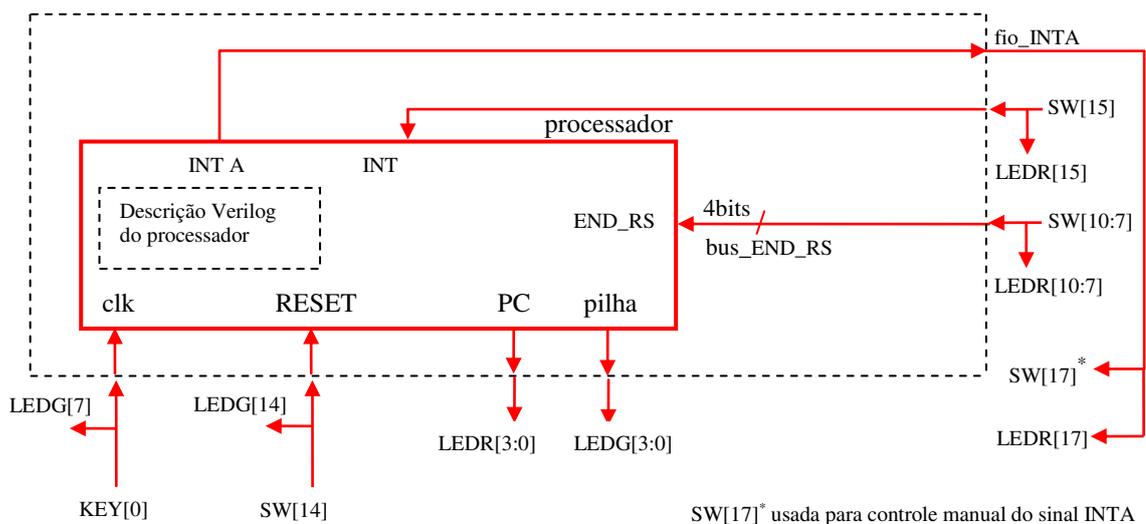
```

```
//DEFINIÇÃO DOS FIOS/BARRAMENTOS wire E ATRIBUIÇÕES assign
wire fio_INTA;
assign LEDR[17]=fio_INTA;
assign fio_INTA=SW[17];
assign LEDR[15]=fio_INT;
assign LEDG[7]=KEY[0];
assign LEDR[14]=SW[14];

/*-----FIM DO INSTANCIMENTO -----*/
```

### 3.5.2.3 Diagrama em blocos – Módulo Processador

O diagrama em blocos do módulo processador, desenvolvido pela equipe B, é mostrado na fig. 9. A descrição de funcionamento do mesmo é feita no item 3.5.2.4, logo abaixo.



**Figura 9.** Diagrama em blocos do módulo controlador e suas interfaces

### 3.5.2.4 Quadro de teste de funcionalidade

No quadro 10 é apresentado o estado lógico inicial das interfaces de entrada e saída do módulo processador.

**Quadro 10** – Estado lógico das interfaces para teste do módulo controlador

Clk	INT	INTA	END_RS	PC	pilha	RESET
KEY[0] e LEDG[7]	SW[15] e LEDR[15]	SW[17] e LEDR[17]	SW[10:7] e LEDR[10:7] e HEX0	LEDR[3:0]	LEDG[3:0]	SW[14] e LEDR[14]
0 e 0	0	0 e 0	0000 e 4'bzzzz e 7'bzzzzzz	0000	0000	0 e 0

Legenda do quadro:

chave KEY: 0 → não houve transição; 1 → houve transição.

chave SW: 0 → off = posição para baixo; 1 → on = posição para cima.

LEDs: 0 → apagado; 1 → aceso. Exceto LEDR[14] que sinaliza transição de clk

Execute as tarefas **T22 a T26** do Quadro 5. O passo a passo é apresentado a seguir.

### Primeiro passo

- i. Antes de iniciar verifique se todas as chaves SW estão na posição para baixo: SW[17:0]=0.
- ii. No momento em que o código é gravado na placa DE2 observe se os estados das principais interfaces comportam-se como mostrado no Quadro 10.
- iii. Enquanto não houver pedido de interrupção INT= 0 e INTA = 0, a partir do primeiro pulso de clk: PC = PC+1. Os demais permanecem inalterados. Verifique isso atuando em KEY[0].
- iv. Continue a sequência de clk observando as mudanças ocorridas em PC. Lembrar que o PC é o contador que aponta sempre para a próxima instrução do programa. Pare com o PC em 1111 = LEDR[3:0].

### Segundo passo

- i. Atue nas chaves SW[10:7] para configurar um endereço de rotina de serviço igual a 1010: SW[10]=1; SW[9]=0; SW[8]=1; e SW[7]=0. Observe que todos os leds LEDR[10:7] correspondentes às chaves SW[10:7] acendem.
- ii. Atue em SW[15] para gerar o sinal de INT. O LEDR[15] ira acender para sinalizar que esse pedido foi feito.
- iii. Observe os valores de END\_RS=LEDR[10:7] =1010, e do PC=LEDR[3:0] = 1111. Logo em seguida atue na chave SW[17] para fazer INTA=1(aceitação do pedido de INT), indicado pelo led LEDR[17].

- iv. Logo em seguida dê um pulso de clk através de KEY[0] e observe atentamente que: o conteúdo do PC=1111 irá ser armazenado na pilha, e seu será indicado por LEDG[3:0]; e, ao mesmo tempo, o conteúdo de END\_RS = 1010 será transferido para o PC: Vê LEDR[3:0].
- v. Dê seis pulsos de clk e observe as mudanças no PC através de LEDR[3:0]. Essas mudanças estão relacionadas à execução da rotina de serviço de interrupção, a partir do endereço de END\_RS = 1010.
- vi. Para retornar ao ponto do endereço onde o PC estava apontando faça INTA=0, e logo em seguida, dê um pulso de clk através de KEY[0]. Nesse momento observe que o conteúdo da pilha – vê LEDG[3:0], que é o endereço apontado pelo PC antes da interrupção, será transferido para o PC: vê LEDR[3:0].
- vii. Para verificar as mudanças no PC apontando para o endereço da próxima instrução ser executada pelo processador gere uma sequência de pulsos de clk através de KEY[0].

**Caso deseje repetir a última situação proceda do seguinte modo:**

- ✓ mantenha as chaves SW[10:7] nas mesmas posições. Logo em seguida atue em SW[17] para fazer INTA=0: observe o led LEDR[17] apagar.
- ✓ reset o processador através de SW[14]. O LEDR[14] irá acender para indicar que o processador está "resetado".
- ✓ dê apenas um pulso de clk através de KEY[0]. Isso fará PC=0, e pilha = 0. Na sequência gere pulsos de clk e observe que não haverá mais mudanças porque o sistema está "resetado".
- ✓ desative o RESET com SW[14]=0. Nesse momento LEDR[14] deve apagar. A partir daí, se desejar, você pode repetir o processo a partir do item ii, do 2º passo.

*Procedimentos para iniciar teste com novo endereço de rotina de interrupção*

Após o item vii, mantenha INTA=0. Mude o endereço da rotina de serviço de interrupção para um valor desejado através das chaves SW[10:7]. Continue o processo a partir do item ii.

*3.5.2.5 Quadro das interfaces de integração do subprojeto Processador*

O quadro 11 apresenta o código de instanciamento do Subprojeto Processador.

**Quadro 11 – Instanciamento do subprojeto processador**

N	Instâncias – assign – wire SUBPROJETO_CONTROLADOR	Interface da Placa	Aplicação no projeto
1	.INT(SW[15]),	SW[15]	Gera sinal INT: pedido de Interrupção
2	.INTA(fio_INTA),	Não	Vê wire fio_INTA
3	.END_RS(SW[10:7]),	SW[10:7]	Gera END_RS
4	.clk(KEY[0]),	KEY[0]	Pushbutton gerador do clock do sistema
5	.PC(LED3:0),	LED3:0	Sinalização do conteúdo do PC
6	.pilha(LEDG3:0),	LEDG3:0	Sinalização do conteúdo da pilha
7	.RESET(SW[14]),	SW[14]	Gera sinal de RESET
8	wire fio_INTA;	Não	Acopla sinal INTA para: SW[17]; LEDR[17]
9	assign LEDR[15]=SW[15];	LEDR[15]	Sinaliza INT
10	assign fio_INTA=SW[17];	SW[17]	Gera INTA
11	assign LEDR[17]=fio_INTA;	LEDR[17]	Sinaliza INTA
12	assign LEDR[10:7]=SW[10:7];	LEDR[10:7]	Mostra END_RS bit a bit
13	assign LEDG[7]=KEY[0];	LEDG[7]	Sinaliza clk
14	assign LEDR[14]=SW[14];	LEDR[14]	Sinaliza sinal de RESET

MAPA DE USO DE INTERFACES - PLACA DE DESENVOLVIMENTO FPGA: LEDES, DISPLAYS, E CHAVES														QD	QD = Quantidade Disponível p/ uso							QD							
HEXi	7	6	5	4	3	2	1	0							8								x						
LEDR	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	8	LEDG	7	6	5	4	3	2	1	0	3
SW	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	7	KEY					3	2	1	0	3

### 3.6 ETAPA 4 – PROJETO INTEGRADO

Na etapa final, as equipes se juntam para integrar as instancias e o módulo de seus subprojetos com o objetivo de obter o projeto proposto. No quadro 12 temos os detalhes de execução.

**Quadro 12** - Projeto integrado - integração global dos subprojetos

SUBETAPAS	PROCESSOS	TAREFAS	ATOR
6 Desenvolvimento do Projeto Integrado	6.1 Reunião das equipes de trabalho	<p><b>T28</b> – Formar grupos de trabalhos com as equipes cujos subprojetos formam o circuito global.</p> <p><b>T29</b> – Escolher um dos alunos do grande grupo para atuar como Integrador de Projetos.</p> <p>A partir daí o aluno Integrador irá executar as tarefas restantes, mediado pelo grupo.</p>	Alunos
	6.2 Importação de arquivos hdl dos subprojetos	<p><b>T30</b> - Colocar na pasta "ProjetoIntegradoNome" uma cópia de cada arquivo .v dos subprojetos criados pelas diferentes equipes.</p> <p><b>T31</b> - Colocar na pasta "InstanciaçãoSubprojeto" uma cópia de cada arquivo tipo txt – bloco de notas – dos códigos de instanciação dos subprojetos criados pelas diferentes equipes.</p> <p>Observação: a tarefa T32 é realizada somente no processo semi-automático de integração. Encontra-se disponível apenas o modelo UML.</p>	Alunos
	6.3 Preenchimento do Quadro de integração global	<p><b>T32</b> – Preencher o quadro de integração global</p> <p>Durante a integração global:</p> <ul style="list-style-type: none"> <li>-usar os quadros de integração dos subprojetos para formar o quadro global;</li> <li>- eliminar os códigos das funções e/ou módulos de testes usados nos subprojetos;</li> <li>- unificar as variáveis com nomes diferentes que possuem as mesmas funções.</li> </ul>	Alunos
	6.4 Criação do código de Integração global.	<p><b>T33</b> – Criar o código de Integração global.</p> <p>MODO MANUAL</p> <p>Repetir as tarefas <b>T17 a T26</b> a partir do quadro de integração global. Lembrar que é necessário criar um quadro de teste de funcionalidade para o circuito global.</p>	Aluno Integrador

Essa é a etapa final do processo onde o aluno integrador reúne os subprojetos desenvolvidos pelas diferentes equipes A e B para formar um projeto integrado. A seguir é mostrado o passo a passo do processo de integração de projetos a partir do modelo usado para validar a metodologia proposta.

### *Procedimentos*

Inicialmente execute as tarefas **T28** a **T31** do quadro 12. Logo em seguida execute as tarefas **T32** e **T33**. Os códigos de instanciamento de todos os subprojetos desenvolvidos pelas diferentes equipes são colocados em um único quadro (Quadro 13) para posterior análise e ajuste (Quadro 14), com o objetivo de se obter o instanciamento global para o projeto integrado (Quadro 15).

**Quadro 13** – Quadro com o código de instanciamento dos subprojetos

n	Instância SubprojetoControlador	Interface da Placa	Aplicação no projeto
1	.LI_0(SW[0]),	SW[0]	Chave Geradora de Interrupção
2	.LI_1(SW[1]),	SW[1]	Chave Geradora de Interrupção
3	.LI_2(SW[2]),	SW[2]	Chave Geradora de Interrupção
4	.LI_3(SW[3]),	SW[3]	Chave Geradora de Interrupção
5	.clk(KEY[0]),	KEY[0]	Pushbutton gerador do clock do sistema
6	.INT(LED[15]),	LED[15]	Sinaliza o pedido de Interrupção INT
7	.INTA(SW[17]),	SW[17]	Acopla sinal INT: pedido de Interrupção
8	.END_RS(LED[10:7]),	LED[10:7]	Apresenta o END da Rotina de Interrupção bit a bit
9	.display_P(HEX5)	HEX5	Apresentação da Linha de Interrupção LI
10	.display_END_RS(HEX0)	HEX0	Apresentação HEXADECIMAL do END da Rotina de Int
<b>11</b>	<b>Instância SubprojetoProcessador</b>	<b>Interface da Placa</b>	<b>Aplicação no projeto</b>
12	.INT(SW[15]),	SW[15]	Gera sinal INT: pedido de Interrupção
13	.INTA(fio_INTA),	Não	Vê wire fio_INTA
14	.END_RS(SW[10:7]),	SW[10:7]	Gera END_RS
15	.clk(KEY[0]),	KEY[0]	Pushbutton gerador do clock do sistema
16	.PC(LED[3:0]),	LED[3:0]	Sinalização do conteúdo do PC
17	.pilha(LEDG[3:0]),	LEDG[3:0]	Sinalização do conteúdo da pilha
18	.RESET(SW[14]),	SW[14]	Gera sinal de RESET
<b>19</b>	<b>WIRE E ASSIGN</b>		
20	wire fio_INTA;	Não	Acopla sinal INTA para: SW[17]; LED[17]
21	assign LEDG[7]=KEY[0];	LEDG[7]	Sinaliza transição de clk
22	assign HEX7=7'b1000111;	HEX7	Letra L da palavra Linha de Instrução - LI
23	assign HEX6=7'b1001000;	HEX6	Letra I da palavra Linha de Instrução - LI
24	assign HEX3=7'b0000110;	HEX3	Letra E da palavra Endereço - End
25	assign HEX2=7'b0101011;	HEX2	Letra n da palavra Endereço - End
26	assign HEX1=7'b0100001;	HEX1	Letra d da palavra Endereço - End
27	assign LEDR[15]=SW[15];	LEDR[15]	Sinaliza INT
28	assign fio_INTA=SW[17];	SW[17]	Gera INTA
29	assign LEDR[17]=fio_INTA;	LEDR[17]	Sinaliza INTA



n	Instância SubprojetoControlador	Comentários
24	assign HEX3=7'b0000110;	
25	assign HEX2=7'b0101011;	
26	assign HEX1=7'b0100001;	
27	assign LEDR[15]=SW[15];	Vê Linhas n = 6 e 12.
28	assign fio_INTA]=SW[17];	Vê linha n = 13.
29	assign LEDR[17]=fio_INTA;	Vê linha n=13.
30	assign LEDR[10:7]=SW[10:7];	Vê linha n = 8.
31	assign LEDR[14]=SW[14];	Vê linha n = 18.

### Quadro 15 – Código de instanciamento global ajustado para o projeto integrado

n	Instância SubprojetoControlador	Interface da Placa	Aplicação no projeto
1	.LI_0(SW[0]),	SW[0]	Chave Geradora de Interrupção
2	.LI_1(SW[1]),	SW[1]	Chave Geradora de Interrupção
3	.LI_2(SW[2]),	SW[2]	Chave Geradora de Interrupção
4	.LI_3(SW[3]),	SW[3]	Chave Geradora de Interrupção
5	.clk(KEY[0]),	KEY[0]	Pushbutton gerador do clock do sistema
6	.INT(fio_INT),	Não	vê wire fio_INT
7	.INTA(fio_INTA),	SW[17]	vê wire fio_INTA
8	.END_RS(bus_END_RS),	Não	Barramento de 4bits do END da Rotina de Interrupção
9	.display_P(HEX5)	HEX5	Apresentação da Linha de Interrupção LI
10	.display_END_RS(HEX0)	HEX0	Apresentação HEXADECIMAL do END da Rotina de Interrupção
11	.clk(KEY[0]),	KEY[0]	Pushbutton gerador do clock do sistema
12	.INT(fio_INT),	Não	vê wire fio_INT
13	.INTA(fio_INTA)),	Não	vê wire fio_INTA
14	.END_RS(bus_END_RS)),	Não	vê wire bus_END_RS
15	.PC(LED[3:0]),	LEDR[3:0]	Sinalização do conteúdo do PC
16	.pilha(LEDG[3:0]),	LEDG[3:0]	Sinalização do conteúdo da pilha
17	.RESET(SW[14]),	SW[14]	Gera sinal de RESET
18	<b>WIRE E ASSIGN</b>		
19	wire fio_INT;	Não	Acopla sinal INT para: LEDR[15]; Módulo processador.
20	wire fio_INTA;	Não	Acopla sinal INTA para: LEDR[17]; Módulo Controlador.
21	wire [3:0] bus_END_RS;	Não	Acopla End da Rotina Serviço de Inter para: LEDR[10:7]; Módulo Processador.
22	assign LEDR[15]=fio_INT;	LEDR[15]	Sinaliza INT
23	assign fio_INTA]=SW[17];	SW[17]	Gera INTA
24	assign LEDR[17]=fio_INTA;	LEDR[17]	Sinaliza INTA
25	assign LEDR[10:7]=bus_END_RS;	LEDR[10:7]	Sinaliza Endereço Rotina de Serviço de Interrupção
26	assign HEX7=7'b1000111;	HEX7	Letra L da palavra Linha de Instrução - LI
27	assign HEX6=7'b1001000;	HEX6	Letra I da palavra Linha de Instrução - LI
28	assign HEX3=7'b0000110;	HEX3	Letra E da palavra Endereço - End
29	assign HEX2=7'b0101011;	HEX2	Letra n da palavra Endereço - End
30	assign HEX1=7'b0100001;	HEX1	Letra d da palavra Endereço - End
31	assign LEDR[14]=SW[14];	LEDR[14]	Sinaliza sinal de RESET

MAPA DE USO DE INTERFACES - PLACA DE DESENVOLVIMENTO FPGA: LEDS, DISPLAYS, E CHAVES													QD	QD = Quantidade Disponível p/ uso							QD								
HEXi	7	6	5	4	3	2	1	0					1									x							
LEDR	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	6	LEDG	7	6	5	4	3	2	1	0	4
SW	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	11	KEY					3	2	1	0	3

### 3.6.1 Código de instanciamento do módulo processador

```

/*-----INÍCIO DO CÓDIGO GLOBAL DE INTEGRAÇÃO -----*/

//INSTÂNCIAS - CONEXÕES
processador PROCESSADOR
(
    .INT(fio_INT),
    .INTA(fio_INTA),
    .END_RS(bus_END_RS),
    .clk(KEY[0]),
    .PC(LED[3:0]),           // sinalização dos dados do PC.
    .pilha(LEDG[3:0]),      // sinalização dos dados da PILHA temp.
    .RESET(SW[14]),
);
controlador CONTROLADOR
(
    .LI_0(SW[0]),
    .LI_1(SW[1]),
    .LI_2(SW[2]),
    .LI_3(SW[3]),
    .clk(KEY[0]),
    .INT(fio_INT),
    .INTA(fio_INTA),
    .END_RS(bus_END_RS),
    .display_P(HEX5),
    .display_END_RS(HEX0),
);

//DEFINIÇÃO DOS FIOS/BARRAMENTOS wire E ATRIBUIÇÕES assign
wire fio_INT;
wire fio_INTA;
wire [3:0] bus_END_RS;

assign LEDR[10:7] = bus_END_RS;
assign LEDR[15] = fio_INT; //Indica no led vermelho quando é gerado uma interrupção
assign LEDG[17] = KEY[0];
assign HEX7 = 7'b1000111; //Letra L de Linha
assign HEX6 = 7'b1001111; //Letra I de Interrupção
assign HEX3 = 7'b0000110; //Letra E da palavra End de Endereço
assign HEX2 = 7'b0101011; //Letra n da palavra Edd de Endereço
assign HEX1 = 7'b0100001; //Letra d da palavra End de Endereço
assign fio_INTA=SW[17];
assign LEDR[17]=fio_INTA; // Sinalizador do sinal INTA
assign LEDR[14]=SW[14];

/*-----FIM DO CÓDIGO GLOBAL DE INTEGRAÇÃO -----*/

```

### 3.6.2 Quadro de teste de funcionalidade do projeto integrado

No quadro 16 é apresentado o estado lógico inicial das interfaces de entrada e saída do módulo processador.

### Quadro 16 – Estado lógico das interfaces para teste do módulo controlador

clk	INT	INTA	END_RS	PC	pilha	RESET	LI[3:0]	LI = P
KEY[0] e LEDG[7]	SW[15] e LEDR[15]	SW[17] e LEDR[17]	SW[10:7] e LEDR[10:7] e HEX0	LEDR[3:0]	LEDG[3:0]	SW[14] e LEDR[14]	SW[3:0]	HEX5
0 e 0	0	0 e 0	0000 e 4'bzzzz e 7'bzzzzzzz	0000	0000	0 e 0	0000	7'bzzzzzzz

Legenda do quadro:

chave KEY: 0 → não houve transição; 1 → houve transição.

chave SW: 0 → off = posição para baixo; 1 → on = posição para cima.

LEDs: 0 → apagado; 1 → aceso. Exceto LEDR[14] que sinaliza transição de clk

Execute as tarefas **T24** a **T27** do quadro 5. O passo a passo é apresentado a seguir.

#### Primeiro passo

- i. Antes de iniciar certifique-se se todas as chaves estão na posição inferior SW[17:0]=0.
- ii. No momento em que o código é gravado na placa DE2 observe se os estados das principais interfaces comportam-se como mostrado no quadro 16.
- iii. Enquanto não houver pedido de interrupção INT= 0 e INTA = 0, a partir do primeiro pulso de clk: PC = PC+1. Os demais permanecem inalterados. Verifique isso atuando na chave KEY[0].

#### Segundo passo

- i. Simule um pedido de interrupção fazendo SW[3]=1. Logo em seguida gere uma transição de clk atuando em KEY[0], e observe que nesse momento o indicador de linha inicia a varredura em LI=0.
- ii. Continue atuando em KEY[0] até a varredura encontrar a linha LI=3, que solicitou a interrupção. Nesse instante ocorrerá o seguinte: será mostrado o endereço da rotina de serviço dessa interrupção END\_RS = DH no display HEX0, e 1101 nos LEDs LEDR[10:7].

Lembrar que os endereços das respectivas rotinas de serviços de interrupção END\_RS foram estabelecidos pelo projetista e podem ser visualizadas

simultaneamente em HEXADECIMAL, no display HEX0, ou em binário, nos LEDs LEDR[10:7]:

**END\_RS[0] = AH** em hexadecimal, ou **1010** em binário.

**END\_RS[1] = BH** em hexadecimal, ou **1011** em binário.

**END\_RS[2] = CH** em hexadecimal, ou **1100** em binário.

**END\_RS[3] = DH** em hexadecimal, ou **1101** em binário.

iii. Enquanto INTA=0, o PC=PC+1, mas LI=3, até o pedido de interrupção LI[3] ser completamente atendido. Verifique isso atuando em KEY[0].

### Terceiro passo

- i. Faça INTA=1 atuando em SW[17]: o led LEDR[17] acenderá para sinalizar o sinal INTA. Após o próximo pulso de clk ocorrerá o seguinte: o endereço 1101 da rotina de serviço correspondente a LI[3] será transferido para o PC; e, ao mesmo tempo, o conteúdo do PC será transferido para a pilha.
- ii. Logo em seguida gere pulsos de clk através de KEY[0] e perceba o PC mudar a partir de 1101. Essas mudanças representam a sequência de apontamento do endereço da próxima instrução para execução da referida rotina de serviço.

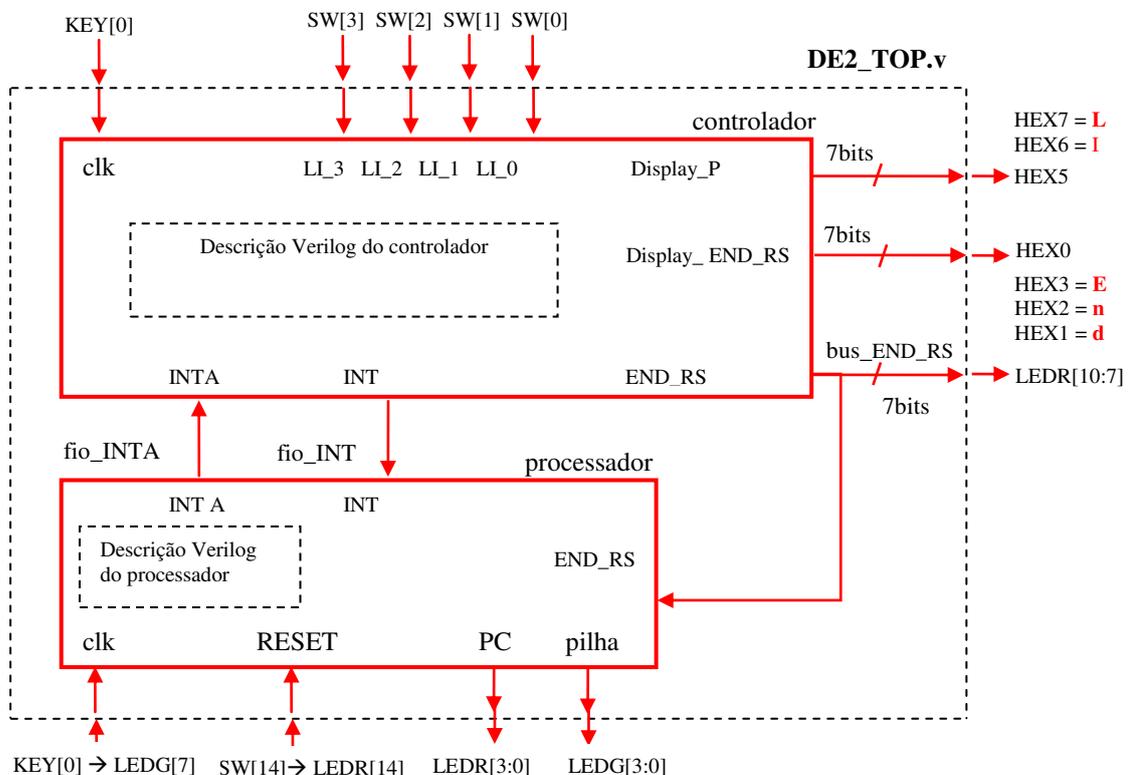
### Quarto passo

- i. Faça INTA=0 atuando em SW[17]: o led LEDR[17] apagará para indicar que o sinal INTA foi desativado. Após o próximo pulso de clk ocorrerá o seguinte: o conteúdo da pilha será devolvido para o PC. Isso significa que o PC recuperou o endereço da próxima instrução do programa principal, que havia armazenado na pilha. Para simular a atuação do PC gere pulsos de clk através de KEY[0] e perceba o mesmo mudar a partir do endereço recuperado.
- ii. Se desejar reiniciar o processo faça INTA=0, através de SW[17], e SW[3:0]=0. Repita o processo individualmente para cada linha de interrupção e observe, ao mesmo tempo, os respectivos endereços de rotina de serviço de interrupção END\_RS: na forma hexadecimal - em HEX0: AH, BH, CH e DH; em binário - nos leds LEDR[10:7]: 1010, 1011, 1100 e 1101, respectivamente.

## ANÁLISE PARA MAIS DE UM PEDIDO SIMULTÂNEO

- i. Após reiniciar o processo fazendo  $INTA=0$  e  $SW[3:0]=0$ , ative duas ou mais linhas de interrupção com auxílio das chaves  $SW[3:0]$ .
- ii. Dê pulsos de  $clk$  com a chave  $KEY[0]$  e observe a mudança do gerador de varredura  $LI$  mudar. Quando a primeira linha for encontrada na ordem de varredura, o contador irá parar até a referida linha seja atendida com  $INTA=1$ .
- iii. Com  $INTA=1$ , após transição de  $clk$ , o endereço da rotina de serviço da linha selecionada será enviado para o PC: vê  $LEDR[3:0]$ , e o conteúdo do PC será transferido para a pilha: vê  $LEDG[3:0]$ .
- iv. Em seguida, desabilite a linha que já foi atendida e, depois faça  $INTA=0$ . Continue a gerar os pulsos de  $clk$ . Com isso, na devida sequência, a próxima linha deverá ser atendida da seguinte forma: faça  $INTA=1$  atuando em  $SW[17]$ , e logo depois dê um pulso de  $clk$ ... E assim por diante, até o último atendimento.

DIAGRAMA EM BLOCOS DO CIRCUITO PROJETADO – Vê fig. 10



**Figura 10** – Diagrama em blocos da unidade de controle (UC)

#### 4 A METODOLOGIA NUMA PERSPECTIVA SÓCIO-INTERACIONISTA

*A aprendizagem pela atividade grupal possibilita ao aluno um exercício para a prática profissional, pois ele aprender que opiniões, crenças e experiências são condições singulares construídas nas relações sociais, e que, para lidar com o outro, é preciso flexibilidade e respeito, saber escutar e dialogar, o que, para ser desenvolvido, necessita de contínuo aprendizado reflexivo. (...) No entanto, limites do próprio estudante e do professor como mediador precisam ser considerados. De qualquer modo, a estratégia grupal é mais uma possibilidade que permite transformar a prática pedagógica, trazendo contribuições ao processo ensino-aprendizagem, já que promove mudanças na relação do estudante com o conhecimento, com o professor, com o colega e consigo mesmo. Nesse contexto a atuação do professor continua sendo fundamental, todavia, envolvendo um papel mais dinâmico, ativo, sintonizado com o processo coletivo e a realidade concreta dos estudantes.*

*(...) Para trabalhar em grupo o professor precisa ser flexível a mudanças, saber aceitar erros, ouvir, falar, sentir, pensar, agir, encontrar soluções com o grupo. No entanto, para alguns professores, o processo de mudança é difícil... Ainda existem alguns educadores que estão centrados em apenas transmitir conhecimento, sem conseguir perceber o funcionamento coletivo e suas implicações para a aprendizagem [50].*

O processo de ensino pode ocorrer de forma sistemática ou assistemática<sup>21</sup>. Quando se trata de ensino sistemático, o professor necessita de um ponto de referência para conduzir sua prática docente. Essa escolha depende de vários fatores, tais como o perfil e as expectativas dos alunos, e a infraestrutura física e administrativa da instituição a qual o professor está vinculado, ou pretende se vincular, e acima de tudo da sua intencionalidade política.

Segundo Paulo Freire [34], “todo ato de educar é um ato político”, portanto, “devemos educar, não para domesticar, e sim, para libertar”.

Dentre várias inquietações que permeiam o pensamento do professor destaca-se: como conduzir uma turma com abordagem interacionista cognitiva e/ou sócio-

---

<sup>21</sup> A diferença entre ambas é o fato de que na forma assistemática, não se tem um planejamento, nem cronograma. Ela se dá de modo natural, ou à medida que a necessidade exige como é o caso da transferência de cultura.

interacionista tendo que acompanhar e respeitar o tempo pedagógico de cada aluno se existe uma carga horária pré-estabelecida, uma quantidade excessiva de alunos por turma, e um conjunto de conteúdo para ser cumprido? Para responder tais questionamentos vale salientar que nenhuma teoria de ensino é auto-suficiente por si mesma, capaz de resolver os problemas reais, em torno do processo de ensino-aprendizagem.

No ponto de vista do processo do ensino-aprendizagem, a metodologia desenvolvida nesse estudo está fundamentada em uma abordagem de ensino sócio-interacionista, mesmo sabendo, que não é raro encontrar nas escolas, e com mais frequência nas universidades, professores cujo ensino reflete os paradigmas das práticas tradicionais, com abordagens de tendências puramente tecnicistas.

Além disso, há o paradoxo dos professores que não suportam ou não acreditam nas contribuições da educação através de ações didático-pedagógicas, que atuam em sala de aula na crença de que basta apenas saber para ensinar, sem a necessidade de nenhum apoio didático-pedagógico que a educação, enquanto ciência, pode proporcionar para melhoria do ensino e da aprendizagem.

Às vezes é necessário que o professor faça uso de artefatos da escola tradicional/tecnicista, seja porque é mais apropriado para determinadas situações, ou porque não é possível usar outras técnicas por não haver como conciliar o tempo escolar, com o tempo pedagógico de cada aluno. Contudo, se a intencionalidade construtivista do professor for mantida, prevalecerá sempre um ensino com tendências onde o aprendiz possa criar seu próprio conhecimento, e não simplesmente reproduzir, de forma alienada, aquilo que já foi pensado.

No dizer de Paulo Freire [35] é necessária uma prática de ensino onde seja possível estimular o aluno a sair de sua "curiosidade ingênua para uma curiosidade epistemológica<sup>22</sup>".

Independente da área que se pretende formar o aprendiz, a sala de aula está diretamente ligada às questões didático-pedagógicas relativas ao ensino e a aprendizagem, da mesma forma que uma bancada de desenvolvimento de Projetos Eletrônicos está para as questões da Engenharia.

---

<sup>22</sup> Segundo Aurélio [30], Epistemologia é o Estudo crítico dos princípios, hipóteses e resultados das ciências já constituídas; teoria da ciência.

Portanto, é no mínimo razoável para quem se propõem entrar em uma sala de aula para mediar a construção de determinado conhecimento, conhecer e se apropriar dos fundamentos didático-pedagógicos para conduzir sua aula, principalmente quando se trata de um processo de ensino-aprendizagem sistemático, e profissional, pois um professor educador é um mediador desse processo, e não simplesmente uma fonte que transfere conhecimento para quem deseja aprender.

Um dos maiores equívocos praticado por quem deseja fazer uso de teorias educacionais em sala de aula é ignorar o contexto de sua aplicação, além de relutar às mudanças de velhos paradigmas. É necessário lembrar que a prática educacional docente está imbricada aos contextos econômicos, sociais, políticos, culturais e históricos dos atores envolvidos no processo de ensino-aprendizagem.

Sua eficácia, entre outros aspectos depende de como são elaboradas e contextualizadas as situações-problema, antes de serem propostas aos alunos. É imperativo que resulte em conhecimentos valiosos à vida do aprendiz.

Cada instituição de ensino possui estrutura administrativa e física próprias que interferem principalmente na organização curricular, e na infraestrutura de laboratórios, além do andamento pedagógico em sala de aula, onde o professor tem que cumprir incondicionalmente a carga horária, e ministrar conteúdos.

Outro fator desabonador é a quantidade excessiva de alunos por turma. Também, é comum encontrar alunos desmotivados em sala de aulas, com idéias compartilhadas de querer tudo pronto, apenas para reproduzir o que já foi pensado.

Outra questão preocupante é a dificuldade que esses alunos apresentam para trabalhar em equipe.

Diante do exposto, porém considerando a existência de uma infraestrutura mínima de hardware e software necessários ao desenvolvimento de projetos didáticos com FPPGA, levantou-se a seguinte questão norteadora: como aplicar uma abordagem de ensino na qual seja possível o desenvolvimento de atividades colaborativas<sup>23</sup>, que propicie um ambiente motivador para os alunos, onde o conhecimento possa ser construído de forma socializada, a partir de propostas para soluções de situações-problema?

---

<sup>23</sup> Vale lembrar que atividades colaborativas e atividades em grupo nem sempre são sinônimos. Nos trabalhos colaborativos as atividades são realizadas a partir de relações interpessoais, e o conhecimento é construído de forma socializada.

A proposta desta pesquisa foi criar uma metodologia de trabalho constituída de uma estrutura que permitisse o uso de uma abordagem de ensino sócio-interacionista, no desenvolvimento das tarefas propostas, associada ao uso de ferramentas de software e hardware para o desenvolvimento dos referidos projetos.

Esta metodologia valoriza as relações interpessoais, e a interação do aprendiz com o objeto de estudo. É refratária ao processo de ensino que leva à subordinação, e à alienação. Ao contrário, propõe estimular o sujeito a refletir, e a construir seu próprio conhecimento, de tal forma que resulte em mudanças valiosas para sua vida.

O uso desta Metodologia não é uma panacéia<sup>24</sup> para os problemas relacionados ao ensino, nem tão pouco àqueles relacionados à aprendizagem, mas apresenta técnicas que propõem dinamizar as aulas – com atividades colaborativas, e apresenta também métodos de análises de diagramas em blocos de circuitos, que proporcionam aos alunos, e ao professor, uma visão sistêmica de todo processo: funcionamento global do circuito, e o andamento do projeto.

Ajuda no processo de avaliação do projeto e subprojetos, pois devido a sua estrutura de organização, facilita a localização dos respectivos módulos, e a manutenção do respectivo código de descrição e/ou integração dos módulos ou subprojetos. Lembrar que o módulo de integração encontra-se separado do módulo de descrição de funcionalidade do circuito.

Proporciona mais autonomia para o estudante desenvolver suas atividades, e amplia o campo de mediação do professor. Além disso, apresenta alternativas para o professor planejar suas aulas numa perspectiva de obter um manejo de classe dinâmico e motivador, devido às estruturas voltadas para uma abordagem sócio-interacionista, que possibilitam atividades colaborativas.

Seu uso depende fortemente da intencionalidade política do professor, que deve ter determinação e comprometimento para experimentar novos paradigmas, principalmente quando se trata de propostas que preconizam mudanças para melhoria do ensino e da aprendizagem.

---

<sup>24</sup> Segundo Dicionário Eletrônico Aurélio, panacéia é pretensamente um remédio eficaz para todos os males de um problema.

Enfim, a ênfase principal da metodologia é o desenvolvimento de trabalhos colaborativos, com enfoque sócio-interacionista. Está dividida em duas partes básicas: a parte relacionada à Educação, e a parte relacionada ao uso da tecnologia usada na ferramenta de desenvolvimento dos projetos didáticos, que se subdivide em duas etapas: a etapa do desenvolvimento dos subprojetos, e a etapa do projeto integrado.

## 5 FERRAMENTA DE APOIO À INTEGRAÇÃO – MODELAGEM UML

### 5.1 INTRODUÇÃO

Na metodologia desenvolvida nesse estudo, um dos métodos utilizados no processo é o preenchimento de quadros com as interfaces de integração. Essa tarefa é realizada em duas etapas: a primeira é feita de forma individual, por cada equipe. Na segunda etapa, as equipes se juntam com seus respectivos quadros para construir um quadro de integração global. Nesses quadros, além das conexões entre as interfaces do projeto, encontram-se informações sobre as aplicações e a quantidade de interfaces disponíveis na placa FPGA. Esses artefatos auxiliam o projetista durante a criação dos códigos de integração, com informações sobre quais das interfaces que ainda se encontram disponíveis para uso na placa FPGA de desenvolvimento.

O cenário apresentado acima possibilita o desenvolvimento de atividades colaborativas, em uma perspectiva sócio-interacionista. Além disso, a execução das referidas tarefas podem ser auxiliadas por uma ferramenta de apoio à integração, mediante automação de parte do processo. Nesse estudo foi feita uma representação UML dessa ferramenta.

### 5.2 LINGUAGEM UML

A *Unified Modeling Language – UML* ou Linguagem de Modelagem Unificada é utilizada para modelar sistemas computacionais por meio do paradigma de Orientação a Objetos. Ela não é uma linguagem de programação porque não possui uma semântica, nem tão pouco é compilável.

Para Guedes[46]:

O objetivo da UML é auxiliar na definição das características de um software, tais como seus requisitos, seu comportamento, sua estrutura lógica, a dinâmica de seus processos, e até mesmo suas necessidades físicas em relação ao equipamento sobre o qual o sistema deverá ser implementado. Todas essas características são definidas por meio da UML antes de o software começar a ser realmente desenvolvido.

Isso significa que seu uso independe da linguagem futura que o desenvolvedor do software irá usar.

A modelagem de software através de UML é feita a partir de diagramas, dentre os quais se destacam o Diagrama de classes, o diagrama de casos de uso, e o diagrama de sequência.

Segundo Silva [47], UML possui treze modelos gráficos, ou diagramas, classificados como diagramas estruturais, e diagramas de comportamento. Os primeiros representam o conjunto de elementos que compõem um software orientado a objetos, e seus relacionamentos. Trata-se de uma representação dos elementos estáticos. O diagrama de classes é o mais importante entre eles, além de ser comumente usado para especificações de software em diversas modelagens de software orientado a objetos.

Os diagramas dinâmicos representam o conjunto de funcionalidades do software e o seu detalhamento, explicitando como os objetos, que são as instâncias das classes, interagem durante a execução para efetuar cada funcionalidade.

Segundo Silva [47], abaixo são apresentados os diagramas em suas respectivas classificação:

- i. Diagramas estruturais: diagrama de classes, diagrama de objetos, diagrama de pacotes, diagrama de componentes, diagrama de estrutura composta, e o diagrama de utilização.
- ii. Diagramas comportamentais: diagrama de casos de uso, diagrama de máquina de estados, diagrama de atividades. Como subcategoria, têm os diagramas de interação: diagrama de sequência, diagrama de comunicação, diagrama de visão geral de interação, e o diagrama de temporização.

Silva [47] afirma que “as metodologias de análise e projeto que adotam UML não prevêm a possibilidade de uso de todos os seus diagramas.”

Na descrição da ferramenta de apoio à integração proposta neste estudo usaram-se os seguintes diagramas: diagrama de classes, diagrama de casos de uso, e o diagrama de sequência.

### 5.3 DESCRIÇÃO GERAL DO SISTEMA

A ferramenta de apoio foi proposta para auxiliar o processo de integração de projetos didáticos desenvolvidos com HDL/FPGA.

O sistema terá como atores principais o professor e o aluno integrador de projetos. O papel fundamental do aluno integrador é acessar o sistema para juntar arquivos com o código de descrição dos módulos dos subprojetos, e os arquivos com os códigos de instanciamento dos referidos módulos, desenvolvidos por diferentes equipes de trabalho, com o objetivo de obter um Projeto Integrado.

O sistema proposto deverá possibilitar aos atores:

- solicitar cadastro;
- realizar *login*;
- alterar senha;
- importar arquivos;
- Instanciar projeto;
- ajustar código de instanciamento global;
- verificar e imprimir estatística de utilização das interfaces da placa FPGA;
- exportar projetos.

Além das ações listadas acima, apenas o ator professor poderá:

- cadastrar usuário;
- importar e excluir diagramas de blocos dos projetos integrados;
- importar e excluir *backup*<sup>25</sup>.

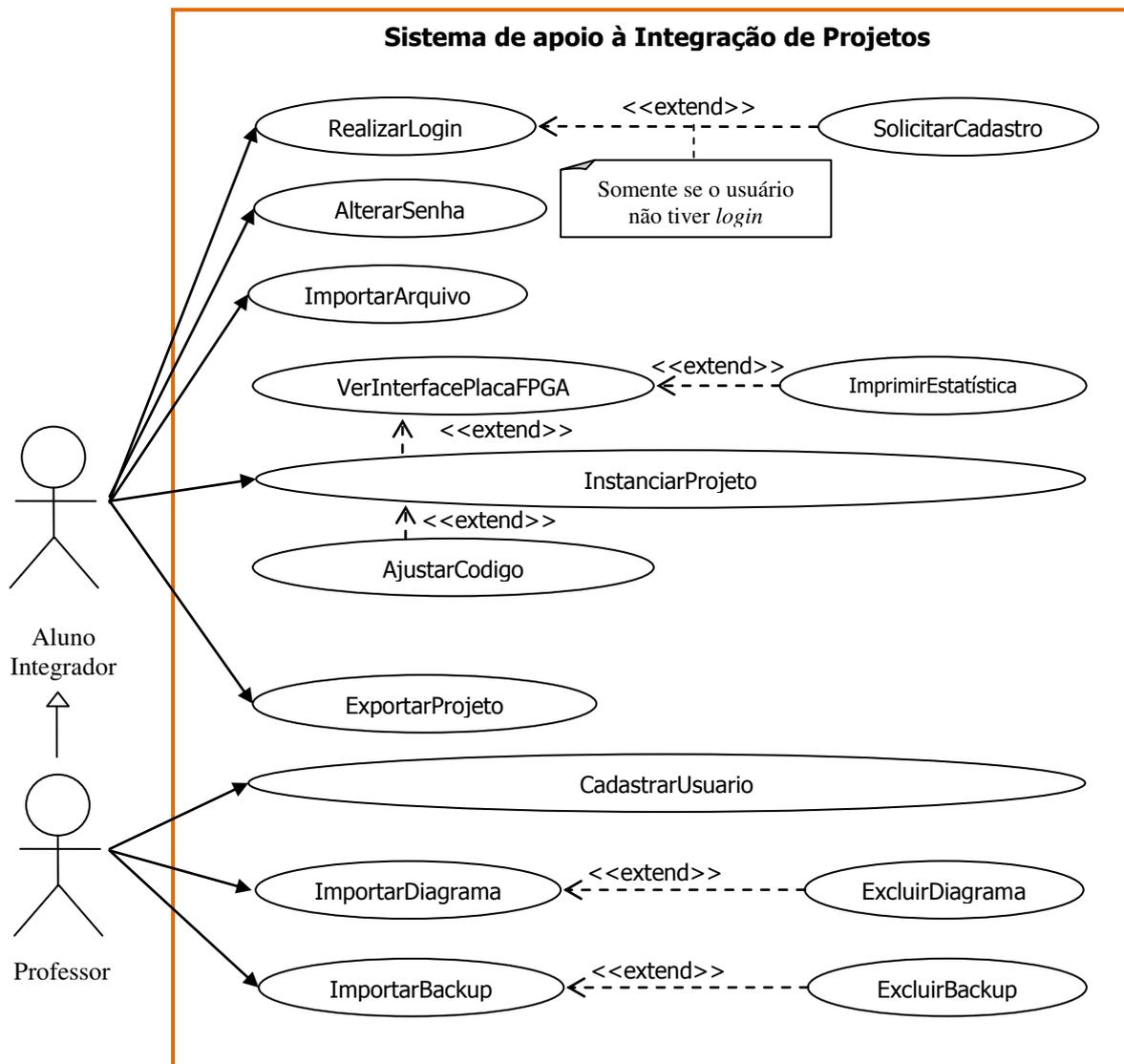
---

<sup>25</sup> *Backup* : unidade sobressalente de arquivos que contêm os códigos de descrição ou instanciamento de módulos.

## 5.4 DIAGRAMAS UML DA FERRAMENTA DE APOIO À INTEGRAÇÃO

Serão apresentados três diagramas UML: o diagrama de caso de uso – fig. 11; o diagrama de sequência – fig. 12, e o diagrama de classes – fig. 13. A documentação dos casos de uso é apresentada nos quadros 17 a 30.

### 5.4.1 Diagrama de caso de uso



**Figura 11** – Diagrama de caso de uso

### 5.4.1.1 – Documentação dos casos de uso

**Quadro 17** – Caso de uso 1: realizar *login*

<b>Caso de uso</b>	<b>Resumo</b>	<b>Pré-condição</b>	<b>Pós-condição</b>
Realizar <i>Login</i> Ator: Professor, aluno. Tipo: geral  Quem inicia o sistema: Professor ou aluno.	Descreve as etapas percorridas por um ator para acessar o sistema	O ator precisa estar cadastrado.	O ator tem acesso ao sistema com a devida permissão de acesso.
Caso de uso geral:			
<b>Ações principais do ator</b>		<b>Ações do sistema</b>	
1 Clicar sobre o ícone do sistema		2 Apresentar janela com campo para o ator digitar usuário e senha (quatro dígitos no mínimo).	
3 Digitar usuário e senha		4 Validar o usuário e senha. 5 Apresenta tela com <i>menu</i> de acesso aos recursos do sistema.	
<b>Fluxo(s) Alternativo(s)</b>			
1 Havendo falha na primeira tentativa de acesso, apresentar mensagem "usuário ou senha incorretos, tente novamente". 2 A partir da segunda tentativa, se a falha persistir, apresentar mensagem "solicitar cadastro ao administrador via e-mail", caso o usuário não esteja cadastrado.			

**Quadro 18** – Caso de uso 1.1: solicitar cadastro

<b>Caso de uso</b>	<b>Resumo</b>	<b>Pré-condição</b>	<b>Pós-condição</b>
Solicitar Cadastro Ator: aluno. Tipo: <i>extend</i>  Quem inicia o sistema: aluno.	Descreve as etapas percorridas por um ator para solicitar cadastro.	Não possuir cadastro	O ator pode fazer <i>login</i> com a devida permissão.
Caso de uso geral: Realizar <i>Login</i>			
<b>Ações principais do ator</b>		<b>Ações do sistema</b>	
1 Clicar sobre o ícone do sistema → Realizar login → Solicitar Cadastro.		2 Apresentar janela com e-mail do administrador para que o ator possa solicitar cadastro através de seu provedor de internet.	
<b>Fluxo Alternativo</b>			
Não.			

**Quadro 19** – Caso de uso 2: alterar senha

<b>Caso de uso</b>	<b>Resumo</b>	<b>Pré-condição</b>	<b>Pós-condição</b>
Alterar Senha Ator: aluno, professor. Tipo: geral  Quem inicia o sistema: Professor ou aluno.	Descreve as etapas percorridas por um ator para alterar senha.	Estar "logado".	Senha alterada.
Caso de uso geral:			
<b>Ações principais do ator</b>		<b>Ações do sistema</b>	
1 Clicar sobre o ícone do sistema → Alterar Senha.		2 Apresentar janela com campo para alteração de senha.	
3 Digitar senha anterior		4 Validar senha	
5 Digitar nova senha – mínimo quatro dígitos		6 Armazenar senha digitada	
7 Confirmar senha atual		6 Validar nova senha 7 Apresentar tela principal com menu de acesso aos recursos do sistema.	
<b>Fluxo Alternativo</b>			
1 Havendo falha na primeira tentativa de acesso, apresentar mensagem "usuário ou senha incorretos, tente novamente".			
2 Se o sistema não validar a nova senha voltar para o item 5.			

**Quadro 20** – Caso de uso 3: importar arquivo

<b>Caso de uso</b>	<b>Resumo</b>	<b>Pré-condição</b>	<b>Pós-condição</b>
Importar Arquivo Ator: Professor, aluno. Tipo: geral  Quem inicia o sistema: Professor ou aluno.	Descreve as etapas percorridas por um ator para Importar arquivos:  <nome>.v <nome>.txt	O usuário tem que estar "logado".	O sistema deverá armazenar os arquivos importados em uma pasta denominada "<Nome do Projeto>".
Caso de uso geral:			
<b>Ações principais do ator</b>		<b>Ações do sistema</b>	
1 Atuar na opção "Importar Arquivos".		2 Apresentar janela com a opção para digitar o nome do projeto.	
3 Digitar nome do projeto e enviar para o sistema.		4 Criar pasta com o nome do projeto digitado pelo usuário. 5 Apresentar janela com caminhos para usuário indicar os arquivos de importação.	
6 Selecionar Arquivo(s) para importação.		7 Destacar arquivos selecionados	
8 Atuar na opção Enviar Arquivo(s)		9 Importar arquivos selecionados para pasta <nomeDoProjeto> 10 Apresentar mensagem: "arquivo(s) importados com sucesso".	
11 Atuar na opção "Tela Principal" ou "Sair".		12 Voltar para tela principal com menu de acesso aos recursos do sistema ou fechar aplicativo.	

<b>Fluxo(s) Alternativo(s)</b>
<p>1 Caso o sistema não consiga importar os arquivos, exibir mensagem "Importação Inválida", e voltar para o passo 1.</p> <p>2 Caso alguma equipe não possua arquivo de determinado módulo ou instância, utilizar <i>backup</i> disponível no sistema, se existir. Neste caso, o sistema deverá registrar que tal arquivo foi utilizado pela referida equipe.</p> <p>3 Se a pasta do projeto já existir apresentar mensagem: "A pasta do projeto já existe. Deseja continuar? SIM ou NÃO? Em caso positivo ir para o item 6. Caso contrário, ir para o item 2.</p>

### Quadro 21 – Caso de uso 4: integrar projeto

<b>Caso de uso</b>	<b>Resumo</b>	<b>Pré-condição</b>	<b>Pós-condição</b>
<p>Instanciar Projeto Ator: Professor, aluno. Tipo: geral</p> <p>Quem inicia o sistema: Professor ou aluno.</p>	<p>Descreve as etapas percorridas por um ator para instanciar o projeto global.</p>	<p>O usuário tem que estar "logado".</p> <p>Os arquivos com extensão txt devidamente importados.</p>	<p>Quadro de instanciamento para criação do projeto integrado preenchido com o código de instanciação de todos os subprojetos</p>
Caso de uso geral:			
<b>Ações principais do ator</b>		<b>Ações do sistema</b>	
1 Atuar na opção "Integrar projeto".		2 Apresentar janela com a opção para digitar o nome do projeto.	
3 Digitar nome do projeto e "Enviar" para o sistema		4 Criar pasta com o nome do projeto digitado pelo usuário. 5 Criar um arquivo global denominado <NomeDoProjeto>.txt e salvar o referido arquivo na pasta anterior.	
6 Atuar na opção "Iniciar Integração".		7 Copiar o código de todos os arquivos <nome>.txt no arquivo global <NomeDoProjeto>.txt . O sistema deverá unificar os referidos arquivos, respeitando a formatação dos arquivos originais, um abaixo do outro. 8 Apresentar editor de texto com o código de instanciação de todos os módulos dos subprojetos para que o usuário possa fazer análise e ajustes no mesmo. 9 Apresentar quadro para consulta, com a estatística da quantidade de interfaces da placa FPGA utilizadas e disponíveis. Deve indicar também se alguma equipe fez uso de <i>backup</i> . O referido quadro deve possuir opção de minimizar e restaurar tamanho	
10 Atuar na opção "Tela Principal" ou "Sair".		11 Voltar para tela principal com menu de acesso aos recursos do sistema ou fechar aplicativo.	
<b>Fluxo(s) Alternativo(s)</b>			
<p>1 Se nenhuma instância de subprojeto for encontrada o sistema deverá apresentar uma mensagem "importar código de instanciação".</p> <p>3 Se a pasta do projeto já existir apresentar mensagem: "A pasta do projeto já existe. Deseja continuar? SIM ou NÃO? Em caso positivo ir para o item 8. Caso contrário, ir para o item 2.</p>			

**Quadro 22** – Caso de uso 4.1: ajustar código de instanciação global

<b>Caso de uso</b>	<b>Resumo</b>	<b>Pré-condição</b>	<b>Pós-condição</b>
Ajustar código de instanciação global. Ator: Professor, aluno. Tipo: <i>extend</i>  Quem inicia o sistema: Professor ou aluno.	Descreve as etapas percorridas por um ator para ajustar o código global de instanciação do projeto.	O usuário tem que estar "logado".  O projeto deve estar previamente instanciado.	Quadro com o código de instanciação do projeto integrado modificado, com a opção de salvar ou manter o código anterior.
Caso de uso geral: Instanciar Projeto			
<b>Ações principais do ator</b>		<b>Ações do sistema</b>	
1 Atuar na opção "Escolher Projeto".		2 Apresentar janela com a opção para escolher um projeto.	
3 Atuar na opção "Ajustar Código".		4 Apresentar Diagrama de blocos do circuito do projeto. 5 Apresenta o código da instância global, e habilita o campo de edição de texto permitindo que seja feito os devidos ajustes.	
6 Ajusta código		7 Apresenta código modificado	
8 Fechar editor de texto		9 Caso as alterações feitas ainda não tenham sido salvas pelo ator, o sistema deverá emitir a seguinte mensagem: Salvar Alterações? SIM ou NÃO?	
10 Escolher a opção desejada SIM ou NÃO		11 Executa função desejada e fecha o editor de texto. 12 Voltar para tela principal com menu de acesso aos recursos do sistema.	
<b>Fluxo(s) Alternativo(s)</b>			
1 Se nenhuma instância de subprojeto for encontrada armazenada o sistema deverá apresentar uma mensagem "importar código de instanciação".			

**Quadro 23** – Caso de uso 4.2: ver interfaces da placa FPGA

<b>Caso de uso</b>	<b>Resumo</b>	<b>Pré-condição</b>	<b>Pós-condição</b>
Verificar Chaves e LEDs. Ator: Professor, aluno. Tipo: <i>extend</i>  Quem inicia o sistema: Professor ou aluno.	Descreve as etapas percorridas por um ator para verificar a quantidade de chaves, LEDs e Displays, disponíveis e utilizadas na placa FPGA de desenvolvimento.	O usuário tem que estar "logado". O instanciamento global deve ter sido feito.	É apresentado um quadro com a relação e quantidade de todos os LEDs, Displays e Chaves usadas no projeto e disponíveis na placa FPGA. O quadro Indica também quando há multiplicidade de uso dos LEDs, chaves e displays.
Caso de uso geral: Instanciar Projeto.			
<b>Ações principais do ator</b>		<b>Ações do sistema</b>	
1 Atuar na opção "Escolher Projeto".		2 Apresentar janela com a opção para	

	escolher um projeto.
3 Selecionar "Ver Interface Placa FPGA".	4 Apresenta quadro com a relação dos LEDs, chaves e Displays utilizados e disponíveis da placa FPGA, além de indicar multiplicidade de uso dos mesmos, quando existir.
5 Atuar na opção "Tela Principal" ou "Sair".	6 Voltar para tela principal com menu de acesso aos recursos do sistema ou fechar aplicativo.
<b>Fluxo(s) Alternativo(s)</b>	
1 Se nenhum quadro for encontrado armazenado, o sistema deverá apresentar uma mensagem "Quadro de Chaves e LEDs não encontrado".	

### Quadro 24 – Caso de uso 4.3: imprimir estatística

Caso de uso	Resumo	Pré-condição	Pós-condição
Imprimir Estatística. Ator: Professor, aluno. Tipo: <i>extend</i>  Quem inicia o sistema: Professor ou aluno.	Descreve as etapas percorridas por um ator para imprimir a estatística sobre a quantidade de chaves, LEDs e Displays, disponíveis e utilizadas na placa FPGA de desenvolvimento.	O usuário tem que estar "logado".  O instanciamento global deve ter sido feito.	Relação impressa com a quantidade de todos os LEDs, Displays e Chaves usadas no projeto e disponíveis na placa FPGA.  A relação impressa Indica também os casos de multiplicidade de uso dos LEDs, chaves e displays.
Caso de uso geral: Instanciar Projeto.			
<b>Ações principais do ator</b>		<b>Ações do sistema</b>	
7 Atuar na opção "Imprimir Estatística".		8 Apresentar janela com a opção para escolher um projeto.	
9 Selecionar "Projeto".		10 Apresenta quadro com a relação dos LEDs, chaves e Displays utilizados e disponíveis da placa FPGA, além de indicar multiplicidade de uso dos mesmos, quando existir.	
11 Atuar na opção "Imprimir"		12 Apresenta janela para configurar impressão.	
13 Configurar impressão.		14 Imprime estatística.	
15 Atuar na opção "Tela Principal" ou "Sair".		16 Voltar para tela principal com menu de acesso aos recursos do sistema ou fechar aplicativo.	
<b>Fluxo(s) Alternativo(s)</b>			
1 Se nenhum quadro for encontrado armazenado, o sistema deverá apresentar uma mensagem "Quadro de Chaves e LEDs não encontrado".			

**Quadro 25 – Caso de uso 5: exporta Projeto**

<b>Caso de uso</b>	<b>Resumo</b>	<b>Pré-condição</b>	<b>Pós-condição</b>
Exportar Projeto. Ator: Professor, aluno. Tipo: geral  Quem inicia o sistema: Professor ou aluno.	Descreve as etapas percorridas por um ator para exportar projeto integrado.	O usuário tem que estar "logado".	Pasta contendo projeto integrado <nome do projeto> transferida para diretório de escolha do usuário.
Caso de uso geral:			
<b>Ações principais do ator</b>		<b>Ações do sistema</b>	
1 "Exportar Projeto".		2 Apresentar janela com a relação dos projetos disponíveis para exportação para outro diretório	
3 Selecionar projeto para exportar.		4 Mostra diretório do projeto selecionado para exportação.	
5 Indicar local para exportação		6 Transfere cópia do diretório selecionado para o local indicado.	
7 Atuar na opção "Tela Principal" ou "Sair".		8 Voltar para tela principal com menu de acesso aos recursos do sistema ou fechar aplicativo.	
<b>Fluxo(s) Alternativo(s)</b>			
1 Se o arquivo não puder ser salvo apresentar mensagem "Arquivo não pode ser salvo".			

**Quadro 26 – Caso de uso 6: cadastrar usuário**

<b>Caso de uso</b>	<b>Resumo</b>	<b>Pré-condição</b>	<b>Pós-condição</b>
Cadastrar Usuário. Ator: Professor. Tipo: geral  Quem inicia o sistema: Professor.	Descreve as etapas percorridas pelo ator para cadastrar usuário.	O usuário tem que ser o professor.	Usuário com acesso aos recursos do sistema.
Caso de uso geral:			
<b>Ações principais do ator</b>		<b>Ações do sistema</b>	
1 Cadastrar Usuário.		2 Apresentar tela com campos para ser preenchido com os dados do usuário, e o tipo (professor ou aluno).	
3 Confirmar cadastro do usuário.		4 Validar dados cadastrados.	
5 Atuar na opção "Tela Principal" ou "Sair".		6 Voltar para tela principal com menu de acesso aos recursos do sistema ou fechar aplicativo.	
<b>Fluxo(s) Alternativo(s)</b>			
1 Se o usuário já estiver cadastrado o sistema deverá apresentar a mensagem "Usuário já cadastrado".			
2 Se dados não validados voltar ao passo 2.			

**Quadro 27** - Caso de uso 7: importar diagrama

<b>Caso de uso</b>	<b>Resumo</b>	<b>Pré-condição</b>	<b>Pós-condição</b>
Importar Diagrama. Ator: Professor. Tipo: geral  Quem inicia o sistema: Professor.	Descreve as etapas percorridas pelo ator para importar diagrama.	"Logado" no sistema como professor.	Arquivo com o diagrama de blocos armazenado na pasta do sistema denominada "DIAGRAMA_BLOCOS".
Caso de uso geral:			
<b>Ações principais do ator</b>		<b>Ações do sistema</b>	
1 "Importar Diagrama".		2 Apresentar janela para o ator indicar o local onde se encontra o diagrama que será importado para o diretório do sistema.	
3 Selecionar <nome do arquivo> para importar.		4 Mostra arquivo selecionado.	
5 Indicar local para importação		6 Copiar o arquivo selecionado na pasta do sistema denominada "DIAGRAMA_BLOCOS".	
7 Iniciar importação		8 Importa arquivo para local.	
9 Atuar na opção "Tela Principal" ou "Sair".		10 Voltar para tela principal com menu de acesso aos recursos do sistema ou fechar aplicativo.	
<b>Fluxo(s) Alternativo(s)</b>			
1 Em caso de não importar apresentar mensagem: "Erro de Importação".			

**Quadro 28** - Caso de uso 7.1: excluir diagrama

<b>Caso de uso</b>	<b>Resumo</b>	<b>Pré-condição</b>	<b>Pós-condição</b>
Excluir Diagrama. Ator: Professor. Tipo: <i>extend</i>  Quem inicia o sistema: Professor.	Descreve as etapas percorridas pelo ator para excluir diagrama.	"Logado" no sistema como professor.	Diagrama excluído do sistema.
Caso de uso geral: Inserir Diagrama			
<b>Ações principais do ator</b>		<b>Ações do sistema</b>	
1 "Excluir Diagrama".		2 Apresentar janela com relação dos diagramas armazenados no sistema.	
3 Selecionar <nome do arquivo> para Excluir.		4 Mostra arquivo selecionado.	
5 Excluir		6 Exclui diagrama selecionado na pasta "DIAGRAMA_BLOCOS" do sistema.	
7 Atuar na opção "Tela Principal" ou "Sair".		8 Voltar para tela principal com menu de acesso aos recursos do sistema ou fechar aplicativo.	
<b>Fluxo(s) Alternativo(s)</b>			
1 Se não tiver permissão para excluir arquivo exibir mensagem: "Não foi possível excluir arquivo".			

**Quadro 29** - Caso de uso 8: importar *backup*

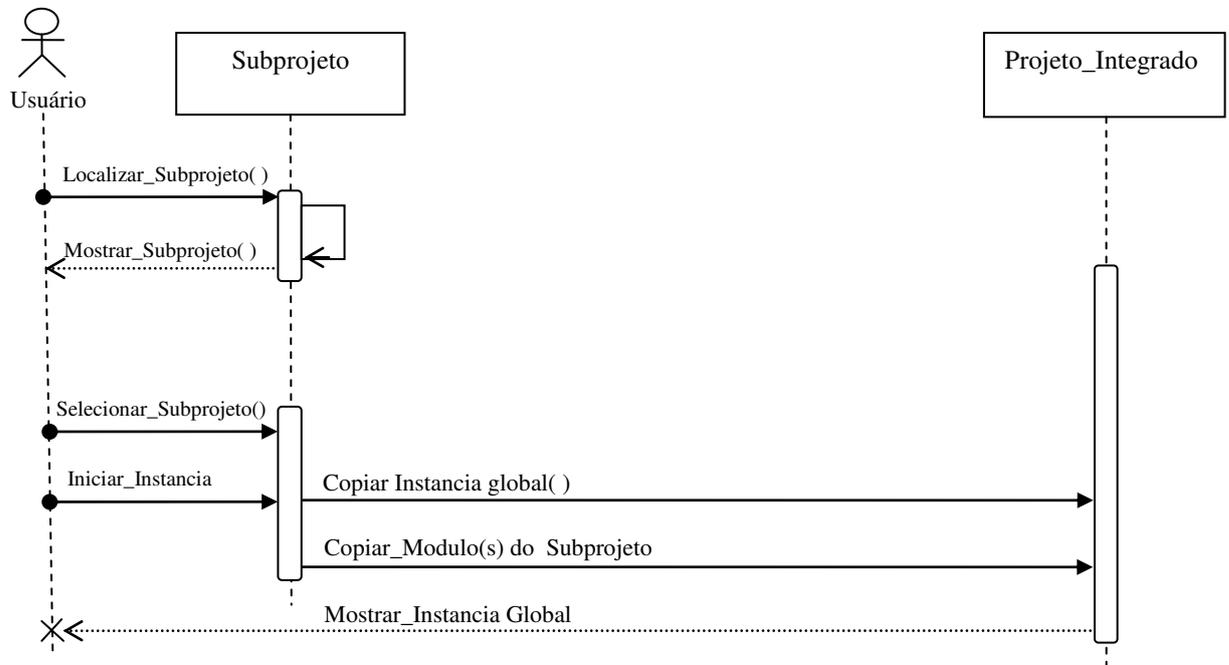
<b>Caso de uso</b>	<b>Resumo</b>	<b>Pré-condição</b>	<b>Pós-condição</b>
Importar <i>Backup</i> . Ator: Professor. Tipo: geral  Quem inicia o sistema: Professor.	Descreve as etapas percorridas pelo ator para Importar Backup.	"Logado" no sistema como professor.	Arquivo .txt ou .v importados para as pastas específica do sistema:  - BACKUP_INSTANCIA: armazena os arquivos txt contendo o código das instâncias dos subprojetos;  - BACKUP_MODULO: armazena os arquivos .v contendo o código de descrição dos módulos.
Caso de uso geral:			
<b>Ações principais do ator</b>		<b>Ações do sistema</b>	
1 "Importar <i>Backup</i> ".		2 Apresentar janela para localizar o local onde se encontra os arquivos .v e .txt dos subprojetos que compõem o projeto integrado.	
3 Selecionar <nome do arquivo> para importar.		4 Mostra arquivo selecionado.	
5 Indicar local para importação		6 Transfere cópia do arquivo selecionado para as devidas pastas do sistema: "BACKUP_MODULOS" ou "BACKUP_INSTANCIAS", dependendo da extensão do arquivo, .v para os módulos e .txt para as instâncias.	
7 Iniciar importação		8 Importa arquivo para local indicado.	
9 Atuar na opção "Tela Principal" ou "Sair".		10 Voltar para tela principal com menu de acesso aos recursos do sistema ou fechar aplicativo.	
<b>Fluxo(s) Alternativo(s)</b>			
1 Se não tiver permissão para importar arquivo exibir mensagem: "Não foi possível importar arquivo".			

**Quadro 30** - Caso de uso 8.1: excluir *backup*

<b>Caso de uso</b>	<b>Resumo</b>	<b>Pré-condição</b>	<b>Pós-condição</b>
Excluir Backup. Ator: Professor. Tipo: <i>extend</i>  Quem inicia o sistema: Professor.	Descreve as etapas percorridas pelo ator para excluir backup.	"Logado" no sistema como professor.	<i>Backup</i> excluído do sistema.
Caso de uso geral: Inserir Diagrama			
<b>Ações principais do ator</b>		<b>Ações do sistema</b>	
1 "Excluir <i>Backup</i> ".		2 Apresentar janela com relação dos backup armazenados no sistema.	
3 Selecionar <nome do arquivo> para Excluir.		4 Mostra arquivo selecionado.	
5 Excluir		6 Exclui backup selecionado do diretório do sistema.	
7 Atuar na opção "Tela Principal" ou "Sair".		8 Voltar para tela principal com menu de acesso aos recursos do sistema ou fechar aplicativo.	
<b>Fluxo(s) Alternativo(s)</b>			
1 Se não tiver permissão para excluir arquivo exibir mensagem: "Não foi possível excluir arquivo".			

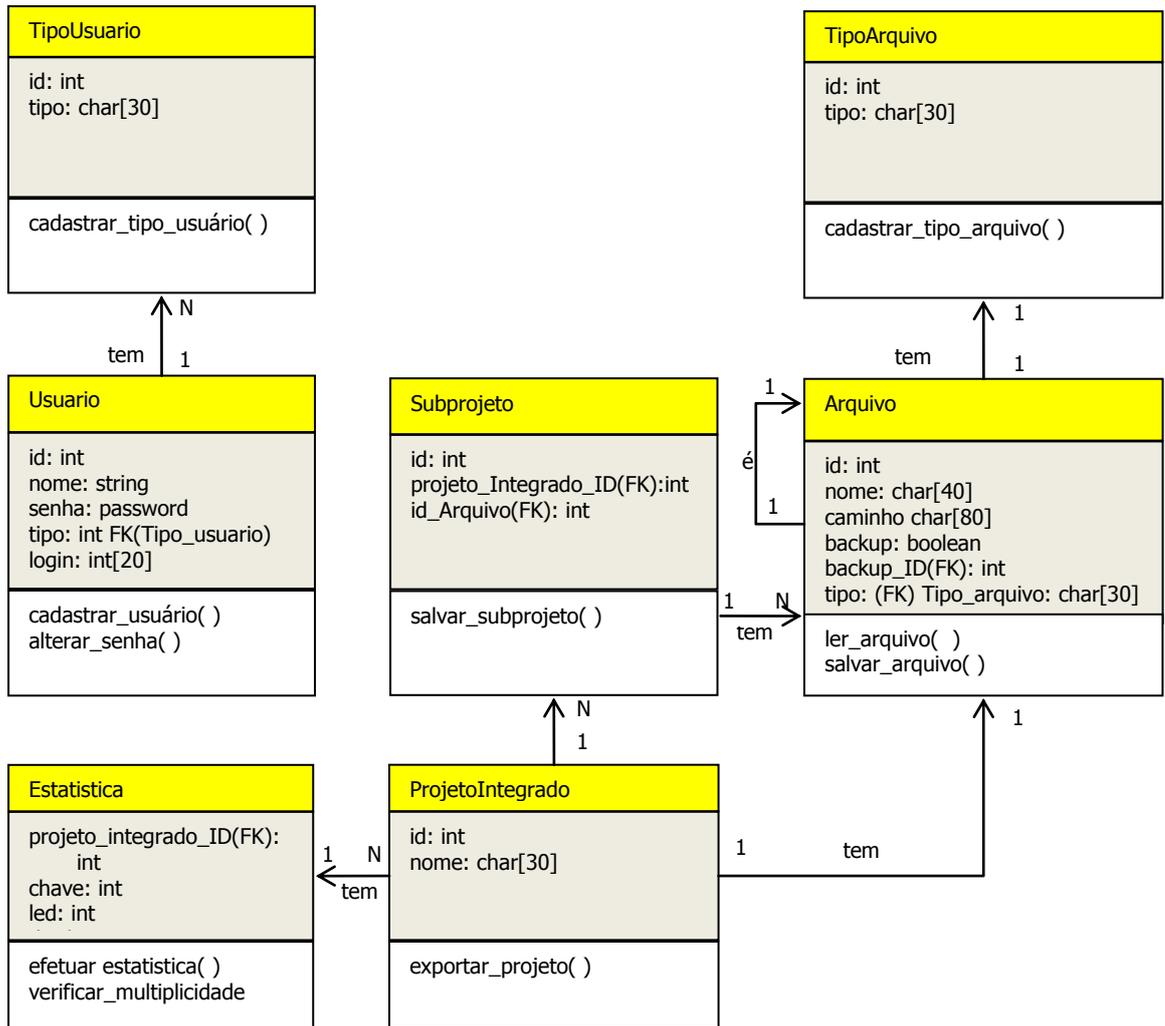
## 5.4.2 Diagrama de sequência

### 5.4.2.1 Instanciar Projeto



**Figura 12** – Diagrama de sequência do caso de uso Instanciar Projeto

### 5.4.3 Diagrama de classes



**Figura 13** – Diagrama de classes do sistema de integração de projetos

## **6 AVALIAÇÃO DA METODOLOGIA – UMA APLICAÇÃO**

O propósito foi avaliar se a metodologia é de fácil compreensão e utilização; e se é eficaz no processo de integração de projetos didáticos. Durante o experimento foi avaliado se as etapas e artefatos propostos são adequados e suficientes para permitir a realização de atividades colaborativas durante o processo de integração de projetos didáticos, a partir da socialização do conhecimento.

Nesse capítulo é mostrado o estudo de caso onde se usou a metodologia desenvolvida para integração de projetos didáticos digitais, com enfoque sócio-interacionista.

### **6.1 PREPARAÇÃO**

Inicialmente criaram-se os artefatos de validação da metodologia: guia do usuário, roteiro de atividades do aluno, quadros de integração, situação-problema do projeto didático, diagrama em blocos do circuito do projeto, formulário de entrevista, e o guia do professor.

Em seguida convidaram-se três voluntários do departamento de engenharia da UFCG, integrantes do laboratório de interfaces homem-máquina (LIHM), para realizar o teste piloto: dois doutorandos X e Y, e um mestre Z. Todos foram ex-alunos da referida universidade, e cursaram a disciplina onde se desenvolviam projetos didáticos digitais implementados com FPGA. Além disso, conheciam e sabiam usar: a linguagem Verilog e a ferramenta de software/hardware de desenvolvimento usado na pesquisa. Com os resultados obtidos foram feitos ajustes nos artefatos de teste de validação da metodologia.

Para realizar os testes de validação da metodologia desenvolvida convidaram-se três voluntários para fazer os papéis dos alunos projetistas A e B, e do professor C para fazer a mediação do processo de ensino-aprendizagem. Os alunos A e B representaram duas equipes diferentes.

Os voluntários A e B eram alunos da graduação do curso de engenharia da UFCG, que cursaram a disciplina onde eram desenvolvidos projetos digitais didáticos com FPGA. O papel de professor C foi desempenhado por um mestre, também ex-

aluno do departamento de engenharia da UFCG, e que também cursou a referida disciplina, além de ter sido monitor da mesma. Na ocasião que cursaram a citada disciplina, os referidos alunos não fizeram uso de nenhuma metodologia de integração de projetos didáticos em uma abordagem de ensino com enfoque sócio-interacionista.

Os voluntários A, B e C possuíam conhecimento sobre uso da ferramenta Quartus II, e a placa DE2 de desenvolvimento, além de conhecer e aplicar a linguagem Verilog para descrição de hardware.

No cenário de teste planejado, o aluno no papel de professor disponibilizou os artefatos, e preparou o ambiente para realização do experimento.

Inicialmente foi ministrado um mini curso de quarenta e cinco minutos sobre a metodologia desenvolvida. Em seguida, os alunos A e B fizeram uma leitura de vinte e cinco minutos sobre o projeto de validação e seus artefatos, com a mediação professor C.

Acompanharam-se as ações dos alunos A e B, e do professor C.

## 6.2 CRITÉRIOS DE AVALIAÇÃO

Para atingir os objetivos do experimento foram adotados alguns critérios de avaliação, e métodos de coleta de dados.

Critérios de avaliação	Métodos utilizados
Usabilidade: facilidade de compreensão e uso Interatividade: ação que se exerce entre duas ou mais pessoas Eficácia: resultado esperado	Observação + entrevista

## 6.3 O PROJETO DE TESTE

O projeto de teste encontra-se detalhado no capítulo 3 deste documento.

## 6.4 REALIZAÇÃO DO EXPERIMENTO

Os dados foram coletados a partir das observações e anotações feitas, e da entrevista realizada com os usuários colaboradores A, B e C.

### 6.4.1 Apresentação das tarefas

Todas as tarefas foram realizadas num tempo total de duas horas.

Aluno projetista A:

- fez leitura silenciosa de vinte minutos sobre a metodologia e seus artefatos.
- preencheu o quadro de integração do subprojeto controlador.

Aluno projetista B:

- fez leitura silenciosa de vinte minutos sobre a metodologia e seus artefatos.
- preencheu o quadro de integração do subprojeto processador.

Alunos A e B juntos:

- Assistiram ao mini curso de quarenta e cinco minutos sobre a metodologia e seus artefatos.
- Integraram os subprojetos.
- Realizaram testes de funcionalidade do circuito projetado – projeto integrado

Professor C

- atuou como mediador no processo de integração.
- participou como professor adjunto no mini curso sobre a metodologia e seus artefatos.

### 6.4.2 Relato das observações

Para gerar o Projeto Integrado, os alunos A e B se juntaram para integrar os subprojetos que desenvolveram individualmente. Cada aluno, com seu quadro de integração nas mãos, compartilharam informações sobre as interfaces geradas, para filtrar as mesmas, e definir àquelas que iriam fazer parte do quadro geral de integração. Consultaram o diagrama em blocos do circuito global. Uma vez preenchido o quadro geral de integração, os alunos A e B iniciaram o processo de criação do projeto integrado. Os detalhes da integração encontra-se descrito no capítulo 3 deste documento.

Em síntese, após o preenchimento do quadro global de integração, os alunos A/B:

- i. Copiaram os arquivos.v de seus respectivos subprojetos na pasta Projeto\_UC\_Interupção.
- ii. Compilaram o projeto integrado.
- iii. Gravaram o código para ser executado na placa FPGA.
- iv. Fizeram o teste de funcionalidade do projeto integrado.

## 6.5 ANÁLISE DOS RESULTADOS

Foram disponibilizados os códigos dos módulos, o guia de integração, a descrição da metodologia, o quadro de funcionamento global do circuito, o diagrama em blocos do circuito do projeto com algumas interfaces de integração já definidas, e o quadro de integração para ser preenchido, com o objetivo de se fazer o teste de validação da metodologia proposta.

No final foram produzidos: diagrama em blocos com todas as interfaces de integração definidas, o código global de integração, e finalmente o circuito completo na placa de desenvolvimento FPGA, de acordo com os requisitos estabelecidos na proposta inicial.

Os usuários interagiram entre si, e realizaram com êxito todas as principais tarefas relativas à metodologia, sem intervenções do avaliador. Cometeram apenas dois erros de sintaxe durante o preenchimento do quadro global de integração. Todos os usuários eram conhecedores da linguagem Verilog de descrição de

Hardware, e, além disso, já possuíam conhecimento sobre o uso do software Quartus II, e da placa de desenvolvimento DE2. No entanto, nunca haviam desenvolvido projetos de forma modular, entre diferentes equipes de trabalho, com integração feita a partir de atividades colaborativas com ênfase em uma abordagem sócio-interacionista. Esses fatos corroboram na definição de facilidade de aplicação da metodologia.

Vale salientar que ponto de vista pedagógico, a metodologia também pode ser aplicada em atividades individuais, com uma abordagem interacionista, onde o aluno interage de forma cognitiva com o objeto de estudo para construir seu próprio conhecimento. O professor atua propondo novos desafios, causando desequilíbrio cognitivo no aluno, levando-o às novas reflexões, que contribuem para sua aprendizagem, aumentando seu cabedal<sup>26</sup> de conhecimentos.

Nesse experimento foram avaliada a eficácia e usabilidade da metodologia de integração de projetos didáticos com FPGA, a partir de atividades colaborativas com enfoque sócio-interacionista. Os seguintes pontos foram observados nas sessões de teste e, depois confirmados através da entrevista realizada com os usuários A e B, após os testes de validação:

- i. Os textos foram escritos com objetividade e clareza.
- ii. A metodologia é de fácil compreensão e utilização.
- iii. A metodologia é eficaz<sup>27</sup>.
- iv. A metodologia proporciona a execução de atividades numa abordagem sócio-interacionista.
- v. O uso do diagrama em blocos é um artefato essencial para o projetista, pois, além de auxiliar na criação e manutenção dos códigos, proporciona uma visão sistêmica do projeto.
- vi. A metodologia é aplicável mesmo sem o uso da ferramenta de apoio à integração.

---

<sup>26</sup>Conforme Aurélio [30], cabedal é o conjunto das Qualidades intelectuais de alguém.

<sup>27</sup> Segundo Dicionário Aurélio [30]: eficaz é aquilo que produz o efeito desejado.

## CONSIDERAÇÕES FINAIS

O tema abordado nesta pesquisa foi a integração de projetos didáticos digitais desenvolvidos de forma modular. Foi desenvolvida uma metodologia de integração de projetos didáticos com FPGA, com ênfase em uma abordagem sócio-interacionista.

O problema consistiu em responder a seguinte questão: como desenvolver e integrar projetos digitais didáticos, desenvolvidos em HDL, a partir de atividades colaborativas, realizadas por diferentes equipes de trabalho, de modo a proporcionar a construção socializada do conhecimento?

A metodologia proposta para auxiliar o desenvolvimento de projetos didáticos de sistemas digitais com HDL e FPGA se fundamenta em princípios da engenharia de software, contendo quatro etapas: (1) preparação didático-pedagógica dos módulos propostos; (2) organização do ambiente de trabalho; (3) desenvolvimento dos módulos e (4) integração dos módulos em um Projeto Global Integrado. Durante as respectivas etapas, professores e alunos realizam processos com o auxílio de ferramentas computacionais.

A etapa de preparação didático-pedagógica consiste na concepção do projeto global e sua subdivisão em módulos que serão alocados às equipes de projeto para desenvolvimento e posterior integração. A etapa de organização do ambiente virtual consiste na escolha e estruturação dos recursos que serão utilizados pelas equipes, tais como placa de desenvolvimento, ferramentas de software. Ainda nessa etapa, diretórios e pastas são sistematicamente organizados para alocação dos arquivos gerados ao longo do desenvolvimento do projeto. A etapa de desenvolvimento dos subprojetos consiste no trabalho individual das equipes voltado para a construção e teste dos respectivos subprojetos, porém visando a posterior integração. Finalmente, na etapa de integração as equipes integram os subprojetos. Nessa etapa são negociadas especificações de conexão entre interfaces, a partir da análise do diagrama de blocos do circuito global, previamente estabelecidas na análise de requisitos. A síntese de execução dessa etapa é mostrada no quadro 31.

**Quadro 31** - Detalhamento da Etapa “Integração em um projeto global”

INTEGRAÇÃO EM UM PROJETO GLOBAL	
Processo	Integração de um módulo ao código do projeto global (constituído dos vários subprojetos), e teste do projeto integrado resultante
Atores	Professor e alunos participantes das equipes.
Subetapas (tarefas)	<ol style="list-style-type: none"> <li>1 Copiar o código de instanciação de cada subprojeto no quadro global de integração. Nessa etapa, a descrição de funções e interfaces auxiliares de testes, utilizadas para simular as entradas provenientes dos demais subprojetos não deve ser copiada.</li> <li>2 Ajustar/corriger o código para a condição de instanciação global.</li> <li>3 Calcular e a quantidade de leds, displays e chaves utilizadas da placa FPGA e apresentar resultado numa tabela, identificando a referencia e a função de cada componente.</li> <li>4 Copiar os arquivos .v de cada subprojeto na pasta Projeto&lt;nome&gt;Integrado.</li> <li>5 Copiar e organizar o código de instanciação global no arquivo genérico DE2_TOP.v, contido na pasta Projeto&lt;nome&gt;Integrado. Não esquecer de salvar.</li> <li>6 Abrir no Quartus II o projeto DE2_TOP.qpf, que se encontra na pasta Projeto&lt;nome&gt;Integrado.</li> <li>7 Compilar Projeto.</li> <li>8 Corrigir Projeto, se necessário.</li> <li>9 Conectar e ligar placa FPGA.</li> <li>10 Gravar projeto na placa FPGA.</li> <li>11 Testar o módulo integrado.</li> </ol>
Artefatos de Entrada	<p>Diagrama de blocos do projeto global.</p> <p>Arquivo com o código de descrição do(s) módulo(s) criado por cada equipe.</p> <p>Quadro com o código de instanciação e a estatística de utilização das interfaces (LEDs, chaves e displays) da placa FPGA DE2, do subprojeto de cada equipe.</p>
Artefatos Produzidos	<p>Quadro global de Integração.</p> <p>Quadro de funcionalidade do projeto integrado para apoio aos testes de integração.</p>

Nesse estudo, o processo de interação entre equipes ocorreu durante o preenchimento do quadro de instanciação do projeto global, e no ajuste/correção do código e nos testes de funcionalidade do circuito.

A metodologia proposta permite abordagens de ensino sócio-interacionista, onde é possível aos alunos a construção socializada do conhecimento, quando fazem parte de equipes de desenvolvimento de projetos didáticos com FPGA, sob a mediação de um professor.

Nessa metodologia, na etapa de desenvolvimento dos subprojetos também é necessário se fazer a integração (instanciação) dos módulos que compõem o referido subsistema. Isso significa dizer que cada equipe trabalha de modo semelhante ao modo da etapa de integração global, onde o trabalho maior é ajustar o projeto para

se obter o que foi definido na análise de requisitos do mesmo. Isso agrega valor ao aluno, que passa a ter a necessidade de desenvolver uma visão sistêmica de todo processo produtivo do conhecimento. Um artefato criado para ajudar o processo de compreensão sistêmica é o diagrama de blocos do circuito global, utilizado para definição e identificação de interfaces, durante a fase de análise de requisitos e na criação e manutenção dos códigos de descrição e instanciação dos módulos. Proporciona um exercício de análise (compreensão das partes) e síntese de sistemas, durante todas as fases do projeto.

Na solução proposta, utilizaram-se dois projetos A e B aplicados em sala de aula, na Universidade Federal de Campina Grande (UFCG). Mesmo se tratando de projetos complementares, a proposta original era desenvolver tais projetos de forma isolada. O curso não foi planejado para ser desenvolvido na abordagem de ensino-aprendizagem proposta nessa pesquisa. Para criar um cenário de validação para esse estudo, os referidos projetos foram re-estruturados para serem desenvolvidos numa perspectiva sócio-interacionista.

A usabilidade e artefatos da metodologia foram avaliados através da integração de dois subprojetos A e B, com dois alunos da graduação e um aluno da pós-graduação. O terceiro aluno exerceu o papel de professor, e todos eles foram ex-alunos da disciplina onde foram desenvolvidos os projetos A e B, citados no parágrafo anterior. Os referidos alunos tinham conhecimento da linguagem Verilog, usada na descrição do circuito do projeto usado na validação. Os testes foram realizados no laboratório de Interface Homem Máquina (LIHM) da UFCG.

O cenário de teste envolvia um computador com o Quartus II – versão 8.1, placa de desenvolvimento FEPGA DE2, quadro de instanciação do projeto integrado para ser preenchido pelas equipes com os códigos dos subprojetos A e B, quadro de funcionalidade do projeto integrado, arquivos com os códigos de descrição dos módulos dos respectivos subprojetos. O quadro sete apresenta a síntese das tarefas necessárias para execução dessa fase de desenvolvimento.

Constatou-se em revisão bibliográfica e em entrevista realizadas com professores que ensinam projetos didáticos digitais implementados com FPGA, que não há orientações didático-pedagógicas como mostrado nesse estudo. Fabricantes

de ferramentas de desenvolvimento de projetos com FPGA e seus usuários focam apenas os recursos tecnológicos do software e hardware disponíveis.

Por outro lado, questionou-se sobre a eficácia da metodologia quando uma das equipes não apresenta parte ou até mesmo todo seu subprojeto, no momento da integração global. Nesse caso, o professor deve disponibilizar um *backup* do referido subprojeto ou módulo, criado previamente em função do planejamento didático e/ou emprestado de por outro grupo. Lembrar que um mesmo projeto pode ser proposto para mais de um grupo, tantas vezes quanto for necessário, dependendo da sua complexidade e interesse pedagógico do professor, além da quantidade de alunos, em sala de aula.

Após a validação percebeu-se que os artefatos criados para experimentação dessa metodologia foram suficientes para se desenvolver projetos didáticos digitais a partir de atividades colaborativas. Portanto, como resultado deste trabalho, concluiu-se que a metodologia proposta pode ser usada como apoio didático-pedagógico na integração de projetos com FPGA.

O processo de validação da especificação UML da ferramenta de apoio à metodologia consistiu na análise da documentação de especificação por um desenvolvedor de software. A partir das sugestões desse profissional os diagramas foram revisados e completados. No entanto esta especificação ainda necessita ser melhorada com a inclusão de mais recursos didático-pedagógicos para o professor, e para contemplar a utilização de outras interfaces da placa FPGA utilizada.

Propõe-se como continuidade desta pesquisa: (1) validar a metodologia com um número maior de participantes, e projetos em um ambiente de ensino real, ao longo de todas as etapas do projeto; (2) averiguar os limites da metodologia proposta experimentando com outras ferramentas e linguagens adotadas no ensino de HDL e FPGA. (3) implementar a ferramenta de apoio à aplicação da metodologia, de modo a automatizar parte do processo, sem prejudicar o processo de produção intelectual do aluno.

## BIBLIOGRAFIA

- [1] VIEIRA, Fátima; CUNHA, Alexandre. **Arquitetura de Sistemas Digitais: Projetos HDL – Período 2009.1.** Roteiro de projetos em Verilog HDL para fins didáticos. Disponível em: <http://www.dee.ufcg.edu.br/~fatima/ASD/asd.htm>. Acessado em: 10/05/2010.
- [2] FERLIN, Edson Pedro; PILLA Jr., Valfredo. O Ensino da Computação Reconfigurável no Curso de Engenharia da Computação. In: XXXIV COBENGE, 2006, Passo Fundo. **Anais.** Passo Fundo: Universidade de Passo Fundo. Disponível em: [http://www.dee.ufma.br/~fsouza/anais/arquivos/6\\_138\\_788.pdf](http://www.dee.ufma.br/~fsouza/anais/arquivos/6_138_788.pdf). Acesso em: 15/05/2010.
- [3] BHATELE, Abhinav; MEHTA, Shashank K. **An Implementation of Picoblaze 8-bit Microcontroller in Verilog.** Department of Computer Science and Engineering Indian Institute of Technology. Kanpur, India. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.120.5231&rep=rep1&type=pdf>. Acesso em: 13/05/2010.
- [4] DUTTA, Abhishek; SHRESTHA, Bibek. **Design a 8-bit microprocessor using Verilog and verify it's operations:** Use SAP-1 (Simple As Possible) architecture as your reference. The University of York - Department of Computer Science. Computer Architecture & Design Lab Assignment. Management report. Disponível em: <http://www-users.cs.york.ac.uk/~abhishek/docs/be/sap1/CAD-SAPreport.pdf>. Acesso em: 03/05/2010.
- [5] OCHI, Hiroyuki; KAMIDOI, Yoko. KAWABATA, Hideyuki. **ASAver.1 : An FPGA-Based Education Board for Computer Architecture/System Design.** Department of Computer Engineering Faculty of Information Sciences. Hiroshima, Japan. Disponível em: [http://www.cs.york.ac.uk/rts/docs/SIGDA-Compendium-1994-2004/papers/1997/aspdac97/pdf/files/03a\\_3.pdf](http://www.cs.york.ac.uk/rts/docs/SIGDA-Compendium-1994-2004/papers/1997/aspdac97/pdf/files/03a_3.pdf). Acesso em 13/05/2010.
- [6] KATO, Edilson Reis Rodrigues et al. **Microprocessador SAP1.** 2008. 50f. Relatório (LAB de arquitetura e organização de computadores) – Universidade Federal de São Carlos. Centro de Ciências Exatas e de Tecnologia. Departamento da Computação. São Carlos, 2008. Disponível em: <http://www2.comp.ufscar.br/~cesarsouza/archive/UFSCar/Lab.%20Arquitetura%20de%20Computadores/Relatorio%2002%20-%20Sap-1.pdf> . Acesso em: 16/05/2010.
- [7] SALES, Zilcio de Lavor. **VHDL.** Disponível em [www.zilcio.com.br](http://www.zilcio.com.br). Acesso em 10/01/10.
- [8] SANT'ANNA, Remy Skinazi; LIMA, Vinícius Augusto Tenório. **Implementação de Controlador em FPGA.** Relatório de Pesquisa (PIBIC-Técnico) – IFPE. Departamento Acadêmico de Sistemas, Processos e Controles Eletro-Eletrônicos. 2009.

[9] SANT'ANNA, Remy Skinazi et al.. **Aquarius:** Uma Plataforma Computacional Dinamicamente Reconfigurável. CIENTEC · Revista de Ciência, Tecnologia e Humanidades do IFPE - Ano I, Nº 1 · Abril/2009.

[10] FECHINE, Joseana Macêdo. **Linguagem de descrição de Hardware:** parte I. Notas de aulas. Disponível em: <http://www.dsc.ufcg.edu.br/~joseana/OAC1-20082.html>. Acesso: 10 set 2008.

[11] UFES – ENGENHARIA ELÉTRICA. **Verilog.** Disponível em: <http://www.ele.ufes.br/~hans/antigo/verilog/index.html>. Acesso em: 09 set 08.

[12] ALTERA CORPORATION. **Basic Computer System for Altera DE2 Board.** Disponível em: <http://www.altera.com/education/univ/support/examples/unv-example-systems.html>. Acesso em: 13/05/2010. Publicado em 2007.

[13] ALTERA CORPORATION. **A Simple Computer System.** Disponível em : [ftp://ftp.altera.com/up/pub/Altera\\_Material/QII\\_9.0/Computer\\_Organization/DE2/Laboratory\\_Exercises/Set1/lab1.pdf](ftp://ftp.altera.com/up/pub/Altera_Material/QII_9.0/Computer_Organization/DE2/Laboratory_Exercises/Set1/lab1.pdf) . Acesso em: 13/05/2010. Publicado em 2006.

[14] ORDONEZ, Edward David Moreno; PEREIRA, Fábio Dacêncio; PENTEADO, Cesar Giacomini; PERICINI, Rodrigo de Almeida. **Projeto, Desempenho e Aplicações de Sistemas Digitais em Circuitos Programáveis (FPGAs).**São Paulo: Bless, 2003. 240p.

[15] ALTERA CORPORATION. **Quartus II Development Software Literature:** versão 8.1. **Disponível em:** [http://www.altera.com/literature/rn/rn\\_qts\\_81.pdf](http://www.altera.com/literature/rn/rn_qts_81.pdf). Acesso em: 15/05/2010.

[16] ALTERA CORPORATION. **Quartus II Development Software Literature:** versão 8.1. **Disponível em:** [http://www.altera.co.jp/literature/rn/rn\\_qts\\_81\\_dev\\_support.pdf](http://www.altera.co.jp/literature/rn/rn_qts_81_dev_support.pdf). Acesso em: 15/05/2010.

[17] ALTERA CORPORATION. **Quartus II Development Software Literature:** versão 9.1. Disponível em: <http://www.altera.com/literature/lit-qts.jsp>. Acesso em: 15/05/2010.

[18] ALTERA CORPORATION. **DE2 Development and Education Board – User Manual.** Disponível em: [http://www.terasic.com/literature/DE2\\_UserManual\\_1.4\\_final.pdf](http://www.terasic.com/literature/DE2_UserManual_1.4_final.pdf)>. Acesso: julho de 2009.

[19] ALTERA CORPORATION. **Cyclone II FPGA Starter Development Board.** Reference Manual. Disponível em: [http://www.altera.com/literature/manual/mnl\\_cii\\_starter\\_board\\_rm.pdf](http://www.altera.com/literature/manual/mnl_cii_starter_board_rm.pdf) Acesso: 10 set 2008.

[20] ALTERA CORPORATION. **Cyclone II FPGA Starter Development Kit**. Disponível em: <http://www.altera.com/products/devkits/altera/kit-cyc2-2C20N.html>. Acesso: 10/09/2008.

[21] COSTA, Cesar da. **Projetos de Circuitos Digitais com FPGA**. São Paulo: Érica, 2009.

[22] PALNITIKAR, Samir. **Verilog HDL: A Guide to Digital Design and Synthesis**. California – EUA: SunSoft Press, 1996.

[23] SANTOS, Roberto Vatar dos. **Abordagens do Processo de Ensino e aprendizagem**. Revista Integração, jan. fev. mar de 2005. Ano XI, nº40, p.19-31. Disponível em: [http://www.usjt.br/prppg/revista/integracao\\_40.php](http://www.usjt.br/prppg/revista/integracao_40.php). Acesso em: 08/07/2010.

[24] FREIRE, Paulo. **Pedagogia do Oprimido**. Rio de Janeiro: Paz e Terra, 1974.

[25] DONATO, Ausônia. **Em torno de algumas questões Educacionais**. Disponível em: [http://www.obore.com/acontece/textos\\_especiais\\_em\\_torno\\_de\\_algunas.asp](http://www.obore.com/acontece/textos_especiais_em_torno_de_algunas.asp). Acesso em: 30/05/2010.

[26] REVISTA ESCOLA. **Skinner**. Edição especial, julho de 2008. Disponível em: <http://revistaescola.abril.com.br/historia/pratica-pedagogica/skinner-428143.shtml>. Acesso Em: 01/06/10.

[27] FUSARI, José Cerchi. **Tendências Históricas do Treinamento em Educação**. Disponível em: [http://www.crmariocovas.sp.gov.br/pdf/ideias\\_03\\_p013-027\\_c.pdf](http://www.crmariocovas.sp.gov.br/pdf/ideias_03_p013-027_c.pdf) . Acesso em: 05/06/2010.

[28] LIMA, Deyse Áurea Queiroz. **Psicopedagogia institucional**. Recife: [s.n], 2005. No prelo.

[29] BARROS, Simone; CAVALCANTE, Patrícia Smith. **Os Recursos Computacionais e suas Possibilidades no Ensino Segundo as Abordagens de Ensino-aprendizagem**. Disponível em: <http://homes.dcc.ufba.br/~frieda/mat061/as.htm>. Acesso em: 25/05/2010.

[30] FERREIRA, Aurélio Buarque de Holanda. **Novo dicionário da língua portuguesa**. Rio de Janeiro: Nova Fronteira, 1986.

[31] ANTUNES, Celso. **Vygotsky, quem diria?! Em minha sala de aula**: fascículo 12. Petrópolis, RJ: Vozes, 2002

[32] VALENTE, José Armando. **Computadores e Conhecimento: Repensando a Educação.** Campinas, SP: Gráfica Central da UNICAMP, 1993.

[33] VALENTE, José Armando. **Informática na Educação: Instrucionismo x Construcionismo.**

[34] FREIRE, Paulo. **Política e Educação.** Rio de Janeiro: Paz e Terra, 1996

[35] FREIRE, Paulo. **Pedagogia da Autonomia: saberes necessários à prática educativa.** São Paulo: Paz e Terra, 2006

[36] MALVINO, Albert Paul. **Microcomputadores e Microprocessadores.** São Paulo: McGraw-Hill, 1985. 578p.

[37] ALTERA CORPORATION. **Nios II Software Developer's Handbook.** Disponível em: <[http://www.altera.com/literature/hb/qts/n2sw\\_nii5v2.pdf](http://www.altera.com/literature/hb/qts/n2sw_nii5v2.pdf)>. Acesso em: 15/07/2009.

[38] ALTERA CORPORATION. **Nios II Processor Reference Handbook.** Disponível em: <[http://www.altera.com/literature/hb/qts/n2cpu\\_nii5v1.pdf](http://www.altera.com/literature/hb/qts/n2cpu_nii5v1.pdf)>. Acesso em: julho de 2009.

[39] ALTERA CORPORATION. **Introduction to the Altera SOPC Builder.** Disponível em: [http://www.altera.com/literature/hb/qts/tut\\_sopc\\_introduction.pdf](http://www.altera.com/literature/hb/qts/tut_sopc_introduction.pdf). Acesso em: julho de 2009.

[40] SALES, Zilcio de Lavor. **Entrevista concedida a Pedro Paulo Marques.** Recife, 30 nov. 2010.

[41] MESSIAS, Wellington Ariano. **Entrevista concedida a Pedro Paulo Marques.** Recife, 06 nov. 2010.

[42] PROCENGE. Disponível em: <http://www.procenge.com.br/index.php> . Acesso em: 15/09/2010.

[43] AZAMBUJA, Rogério Xavier. **Revisão de métodos ágeis empregados no desenvolvimento de softwares.** Disponível em: <http://revista.grupointegrado.br/revista/index.php/campodigital/article/viewFile/312/146> . Acesso em: 25/10/2010. Jornada de Pesquisa e Extensão 2008. ULBRA, Santa Maria – RS.

[44] BRANDÃO, Paulo Sérgio. **Entrevista concedida a Pedro Paulo Marques.** Recife, 06 dez. 2010.

[45] PILETTI, Claudino. **Didática Geral.** Série educação. 20ª ed. São Paulo: Ática, 1997.

[46] GUEDES, Gilleanes Thorwald Araújo. **UML2**: guia prático. São Paulo: Novatec Editora, 2007.

[47] SILVA, Ricardo Pereira e. **UML2**: Modelagem Orientada a Objetos. Florianópolis: Visual Book, 2007.

[48] SCHIAVONI, Flávio Luiz. **Razões por que usar Java**. Disponível em: <http://flavioschiavoni.blogspot.com/2008/09/10-razes-por-que-usar-java.html>. Acesso em: 25/05/2011.

[49] HORSTMANN, Cay. **BIG JAVA**. Trad. Edson Furmankiewicz. Porto Alegre: Bookman, 2004.

[50] CORRÊA, Adriana Katia; SOUZA, Maria Conceição B. M; BARBATO, Roberta Genaro. SILVA, M. Aprender em grupo: experiência de estudantes de Enfermagem e implicações para a formação profissional. Esc. Anna Nery vol.14 no.1 Rio de Janeiro jan./mar. 2010. Disponível em: [http://www.eean.ufrj.br/revista\\_enf/revistas.htm](http://www.eean.ufrj.br/revista_enf/revistas.htm). Acesso em: 20/06/2011.

## APENDICE A – Validação da Metodologia : Quadro das Etapas e Tarefas

### Quadro 32 – Etapas de validação da metodologia proposta

ETAPAS	PROCESSOS	TAREFAS	RECURSOS NECESSÁRIOS	ATOR
1 Planejamento	<p>1.1 Elaboração do cronograma de atividades</p> <p>1.2 Elaboração do roteiro de observação da aplicação da metodologia</p> <p>1.3 Agendamento do Ambiente da Validação</p> <p>1.4 Formação da equipe de trabalho</p>	<p>T1 - Listar as atividades em ordem crescente e cronológica</p> <p>T2 - Elaborar roteiro de observação da aplicação da metodologia</p> <p>T3 – Reservar local da validação e agendar o dia do teste de validação</p> <p>T4 - Convidar colaboradores e definir papéis e responsabilidades para formar a equipe de trabalho</p>	Calendário	Pesquisador
2 Capacitação da Equipe de Trabalho e teste piloto	<p>2.1 Preparação do ambiente</p> <p>2.2 Execução de curso de como usar a metodologia de integração de projetos didáticos com FPGA, na visão do tutor, e do aluno.</p> <p>2.3 Realização do teste piloto.</p>	<p>T5 – Organizar ambiente para curso de capacitação.</p> <p>T6 – Ministrando curso de capacitação de uso da metodologia de integração de projetos didáticos, numa perspectiva sócio-interacionista.</p> <p>T7 – Organizar ambiente para o teste de validação.</p> <p>T8 – Realizar teste piloto de uso da metodologia</p>	<p><b>Recurso material:</b></p> <ul style="list-style-type: none"> <li>- Guia do Professor</li> <li>- Roteiro de atividades do aluno</li> <li>- PC com Quartus II</li> <li>- Placa DE2 Altera</li> <li>- cronômetro digital</li> <li>- Formulário para anotações das observações e entrevista.</li> </ul> <p><b>Recurso humano:</b></p> <p>Alunos da pós-graduação com experiência em sala de aula (monitor da disciplina de projetos digitais) e prática no uso do Quartus II em desenvolvimento de projetos didáticos com FPGA, usando linguagem Verilog.</p>	Pesquisador e alunos
4 Teste de validação e observação	<p>4.1 Preparação do material e ambiente de teste</p> <p>4.2 Realização dos testes de validação</p> <p>4.3 Observação e anotação</p> <p>4.4 Aplicação de questionário e entrevista</p>	<p>T9 – Preparar ambiente de teste</p> <p>T10 – Entregar os roteiros de testes para os alunos A, B e C, e o guia do Professor para o aluno tutor C.</p> <p>T11 – Observar e anotar:</p> <p>i. a interação entre os alunos na realização das atividades.</p>	<p><b>Recurso material:</b></p> <ul style="list-style-type: none"> <li>- Guia do Professor</li> <li>- Roteiro de atividades do aluno</li> <li>- PC com Quartus II</li> <li>- Placa DE2 Altera</li> <li>- cronômetro digital</li> <li>- Formulário para anotações das observações e entrevista.</li> </ul> <p><b>Recurso humano:</b></p> <p>Dois Alunos A e B da</p>	Pesquisador e alunos

ETAPAS	PROCESSOS	TAREFAS	RECURSOS NECESSÁRIOS	ATOR
		<p>ii. O tempo de execução das atividades propostas.</p> <p>T12 Aplicar questionário e realizar entrevista com os alunos A e B após aplicação da metodologia com o objetivo avaliar a usabilidade da metodologia no ponto de vista do aluno.</p>	<p>graduação e um aluno C, da pós-graduação que tenham cursado a disciplina de projetos digitais, e tenham conhecimento no uso do Quartus II em desenvolvimento de projetos didáticos com FPGA, usando linguagem Verilog.</p> <p>Os Alunos da pós-graduação devem ter também experiência em sala de aula como monitor da disciplina de projetos digitais.</p>	
5 Resultados	<p>5.1 Tabulação dos dados</p> <p>5.2 Análise dos resultados</p>	<p>T13 Organizar os dados coletados em tabelas e quadros</p> <p>T14 Analisar os dados para representar resultados e conclusões</p>		Pesquisador

## **APÊNDICE B** – Formulário de observação e entrevista

### OBJETIVOS

Avaliar, através da observação e de uma entrevista, o uso da metodologia de Integração de Projetos didáticos Digitais, a partir de atividades colaborativas, com enfoque sócio-interacionista.

### PARTICIPANTES

Alunos projetistas A e B.  
Aluno professor C.

### PROCEDIMENTOS

#### Antes da Experimento

- Entregar o material dos testes ao professor C para entregar aos alunos A e B, no dia do experimento.
- Agendar o experimento.

#### Tarefas dos Alunos A e B no dia do Experimento

- Realizar leitura silenciosa sobre a metodologia e seus artefatos
- Realizar testes de integração e validação de projetos

#### Tarefas do professor C no dia do Experimento

- Ministrando curso com duração de 40 minutos sobre a metodologia e seus artefatos
- Mediar o processo de validação da metodologia

#### Tarefas do avaliador no dia do Experimento – tempo máximo de 2h

- Apresentar os objetivos e as diretrizes para execução do experimento.
- Observar os alunos A e B, e o professor C.
- Anotar o tempo total de execução das tarefas:  
Aluno A:  
Aluno B:
- Realizar entrevista com os colaboradores após os testes de validação.

## QUESTÕES SOBRE O EXPERIMENTO

<b>1 A proposta do projeto e a Situação-problema</b>	<b>SIM</b>	<b>NÃO</b>	<b>Anotações complementares</b>
A situação-Problema despertou interesse para o desenvolvimento do projeto e busca da solução?			
<b>2 Execução</b>	<b>SIM</b>	<b>NÃO</b>	<b>Anotações complementares</b>
Houve Interatividade entre os participantes?			
Os alunos apresentaram dificuldades para compreender os textos de descrição e procedimentos de execução das tarefas?			
O diagrama em blocos facilitou o processo de definição das Interfaces entre os módulos do sistema, e a compreensão global do projeto?			
O diagrama em blocos ajudou na construção dos quadros de integração, e na criação do código de integração global?			
<b>3 Resultados</b>	<b>SIM</b>	<b>NÃO</b>	<b>Anotações complementares</b>
Alguma tarefa deixou de ser realizada? Em caso positivo identifique a tarefa, e explique o motivo?			
As tarefas propostas conduziram aos resultados esperados?			
A metodologia se adéqua ao grau de complexidade dos projetos didáticos desenvolvidos na disciplina de sistemas digitais?			

### Considerações finais: