



THE IMAGINATION UNIVERSITY PROGRAMME

RVfpga-SoC

Guia de Instalação

Índice

Introdução	3
Instalação para o Lab 1	4
Instalação para o Lab 2	6
Instalação para o Lab 3	10
Instalação para o Lab 4	11
Apêndice A: Instalação de drivers no Windows para usar o PlatformIO	13
Apêndice B: Instalação do Verilator e do GTKWave no Windows	15



1. Introdução

Este guia mostra como instalar as ferramentas e o hardware necessários para o RVfpga-SoC no Sistema Operativo (SO) Ubuntu 18.04. **As instruções abaixo são para um Sistema Operativo Ubuntu 18.04**, mas outros sistemas operativos Linux e Windows seguem etapas semelhantes (se não exatamente as mesmas). Nalguns casos, inserimos caixas com instruções específicas para o sistema operativo Windows. Se estiver usando o Ubuntu, ignore essas caixas.

Esse processo pode levar várias horas (ou mais, dependendo da velocidade do download), mas a maior parte do tempo é gasta na espera enquanto os programas são descarregados e instalados.

A Tabela 1 lista o software e o hardware necessários para o RVfpga-SoC.

Tabela 1. Software e Hardware Necessários para o RVfpga-SoC

Software		
Nome	Website	Custo
Vivado 2019.2 WebPACK	https://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/vivado-design-tools/2019-2.html	gratuito
VSCode	https://code.visualstudio.com/Download	gratuito
PlatformIO	https://platformio.org/ Instalado no VSCode	gratuito
Verilator (simulador HDL) e GTKWave	https://github.com/verilator/verilator http://gtkwave.sourceforge.net/	gratuito
FuseSoC	https://github.com/olofk/fusesoc	gratuito
Toolchain RISC-V e OpenOCD	https://github.com/riscv/riscv-gnu-toolchain https://github.com/riscv/riscv-openocd Instalado no PlatformIO	gratuito
Zephyr Project	https://github.com/zephyrproject-rtos/zephyr	gratuito
Hardware*		
Nome	Site	Custo
Placa FPGA Nexys A7*	https://store.digilentinc.com/nexys-a7-fpga-trainer-board-recommended-for-ece-curriculum/	\$341 (preço acadêmico: US\$ 261,75)
Núcleo RISC-V e sistema em chip (SoC)**		
Nome	Site	Custo
A Western Digital Núcleo SweRV EH1	https://github.com/chipsalliance/Cores-SweRV	gratuito
SweRVolf	https://github.com/chipsalliance/Cores-SweRVolf	gratuito

* O hardware é opcional.

** O núcleo SweRV EH1 e o SweRVolf são fornecidos como parte do pacote RVfpga-SoC.

2. Instalação Para o Lab 1

Esta seção mostrará como instalar o software necessário para realizar o Lab 1 do curso RVfpga-SoC.

1. Instalar o Vivado:

O Vivado é uma ferramenta da Xilinx para visualizar, modificar e sintetizar o código Verilog. Será usado amplamente em laboratórios posteriores. As instruções de instalação estão disponíveis em <https://reference.digilentinc.com/vivado/installing-vivado/start> e são resumidas a seguir.

Windows: a página da Web mencionada acima (<https://reference.digilentinc.com/vivado/installing-vivado/start>) também inclui instruções detalhadas para a instalação do Vivado no Windows. Abaixo, inserimos caixas quando instruções específicas são necessárias para o Windows.

Etapas 1. Navegue até <https://reference.digilentinc.com/vivado/installing-vivado/start>

Etapas 2. Será direcionado para a página de download da Xilinx:
<https://www.xilinx.com/support/download.html>

Etapas 3. Recomenda-se que instale o "Self Extracting Web Installer". No momento em que este documento foi escrito, ele estava neste link na página de download: [Xilinx Unified Installer 2019.2: Linux Self Extracting Web Installer](#).

WINDOWS: No momento da redação deste documento, o "Self Extracting Web Installer" para Windows está neste link na página de download: [Xilinx Unified Installer 2019.2: Windows Self Extracting Web Installer](#)

Etapas 4. Será solicitado que faça login na sua conta Xilinx antes de fazer o download do instalador. Se ainda não tiver uma conta, precisará de criar uma.

Etapas 5. Execute o ficheiro binário. Abra um terminal e torne-o root (escreva "sudo su"). Em seguida, arraste o ficheiro binário (Xilinx_Unified_2019.2_1106_2127_Lin64.bin) para o terminal. Se ele solicitar que torne o ficheiro executável e o execute, selecione OK.

Solução de problemas: Se o terminal indicar permissão negada, escreva o seguinte no terminal (no mesmo diretório que o ficheiro binário):

- `sudo chmod +x ./Xilinx_Unified_2019.2_1106_2127_Lin64.bin`
- `sudo ./Xilinx_Unified_2019.2_1106_2127_Lin64.bin`

WINDOWS: No Windows, basta executar o ficheiro .exe que descarregou nas etapas 3 e 4, clicando duas vezes nele.

Etapas 6. O instalador do Vivado guiará o processo de instalação. Notas importantes:

- Selecione **Vivado** (*não* Vitis) como o produto a ser instalado.
- Selecione Vivado HL **Webpack** (*não* Vivado HL System Edition); o Webpack é gratuito.
- Caso indicado o contrário, os valores por omissão (*default*) devem ser selecionados.

Dica: Se alterar o diretório de instalação do Vivado, será necessário modificar o caminho adequadamente nas etapas a seguir.

WINDOWS: As etapas 7 e 8 não são necessárias para o Windows. pode simplesmente ignorar essas duas etapas e ir diretamente para a etapa 9.

Etapa 7. Depois do Vivado tiver sido instalado, precisará de configurar o ambiente. Abra um terminal e escreva:

```
> source /tools/Xilinx/Vivado/2019.2/settings64.sh
```

Adicione a linha (`source /tools/Xilinx/Vivado/2019.2/settings64.sh`) ao ficheiro `~/.bashrc` para que seja executado sempre que abra um terminal.

Etapa 8. Teste o Vivado escrevendo o seguinte num terminal:

```
> vivado
```

Solução de problemas:

- Se o sistema não conseguir encontrar esse executável, será necessário adicionar o seguinte ao caminho:

```
/tools/Xilinx/DocNav  
/tools/Xilinx/Vivado/2019.2/bin
```

- Se receber um erro como "falha na inicialização específica do aplicativo...", escreva o seguinte num terminal:

```
> sudo ln -s /lib/x86_64-linux-gnu/libtinfo.so.6  
/lib/x86_64-linux-gnu/libtinfo.so.5
```

2. Instale os Drivers da Placa:

Etapa 9. Precisar de instalar manualmente os drivers da placa Nexys A7 FPGA.

Escreva o seguinte numa janela de terminal:

```
> cd  
/tools/Xilinx/Vivado/2019.2/data/xicom/cable_drivers/lin64/install_script/install_drivers/
```

```
> sudo ./install_drivers
```

WINDOWS: A instalação do Vivado no Windows instala automaticamente os drivers para a placa Nexys A7, que não são compatíveis com o PlatformIO. Portanto, se usar o Windows, **deverá atualizar os drivers conforme explicado no Apêndice A do Guia de Instalação.**

3. Instale os Board Files da Digilent:

Também precisará instalar manualmente os Board Files da Digilent.

Etapa 10. Faça download do [ficheiro](#) do vivado-boards do repositório do Github e extraia-o.

Etapa 11. Abra a pasta extraída do ficheiro e navegue até o diretório `new/board_files`. Selecione todas as pastas dentro desse diretório e copie-as.

Etapa 12. Abra a pasta na qual o Vivado foi instalado (`/tools/Xilinx/Vivado` por omissão). Nessa pasta, navegue até diretório `<version>/data/boards/board_files` e, em seguida, cole os ficheiros da placa nesse diretório.

Etapa 13. Também pode usar o terminal, entrando no diretório `new/board_files` e digitando:

```
> sudo cp -r *  
/tools/Xilinx/Vivado/2019.2/data/boards/board_files
```

WINDOWS: copie&cole as pastas descarregadas conforme explicado na Etapa 10. No Windows, pode encontrar a pasta `board_files` do Vivado em:
`C:\Xilinx\Vivado\2019.2\data\boards\board_files`

3. Instalação Para o Lab 2

Esta seção mostrará como instalar o software necessário para realizar o Lab 2 do curso RVfpga-SoC. Ela inclui instruções para a instalação de softwares como o Visual Studio Code (VSCode), PlatformIO, Verilator e GTKWave.

1. Instale o VSCode:

Siga estas etapas para instalar o VSCode:

Etapa 1. Faça o download do ficheiro .deb no seguinte link:

<https://code.visualstudio.com/Download>

Etapa 2. Abra um terminal, instale e execute o VSCode digitando o seguinte no terminal:

```
> cd ~/Downloads  
> sudo dpkg -i <nome do ficheiro>.deb (Altere o nome do  
ficheiro)  
> code
```

Windows: Os pacotes VSCode também estão disponíveis para Windows (ficheiro .exe) em <https://code.visualstudio.com/Download>. Siga as etapas comuns usadas para instalar e executar uma aplicação nesse sistema operativo.

2. Instale o PlatformIO sobre o VSCode:

O PlatformIO é um ambiente de desenvolvimento integrado (IDE) para sistemas embebidos, desenvolvido com base no Visual Studio (VS) Code da Microsoft. Ele permite que programe o processador RISC-V (localizado na FPGA) usando C ou Assembly. O PlatformIO é multiplataforma e inclui um depurador integrado.

Siga estas etapas para instalar o PlatformIO:

Etapa 3. Instale os utilitários python3 digitando o seguinte em um terminal:

```
> sudo apt install -y python3-distutils python3-venv
```

Windows: esta etapa 3 não é necessária no Windows.

Etapa 4. Se ainda não estiver aberto, inicie o VSCode selecionando o botão Iniciar e escreva "VSCode" no menu de pesquisa e, em seguida, selecione VSCode ou escrevendo `code` num terminal.


Etapa 5. No VSCode, clique no ícone Extensions (Extensões)  localizado na barra lateral esquerda do VSCode (ver Figura 1).



Figura 1. Ícone de extensões do VSCode

Etapla 6. Escreva *PlatformIO* na caixa de pesquisa e instale o *IDE PlatformIO* clicando no botão de instalação ao lado dele (ver Figura 2).

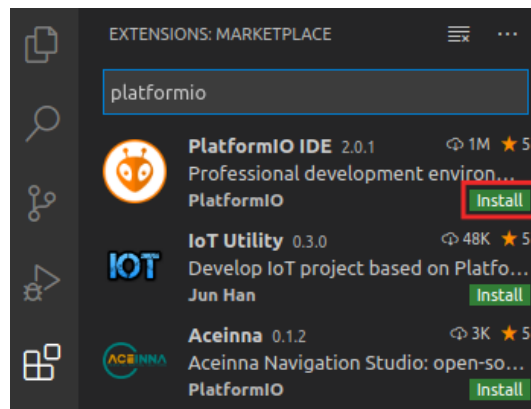


Figura 2. Extensão do IDE PlatformIO

Etapla 7. A janela OUTPUT na parte inferior o informará sobre o processo de instalação. Quando terminar, clique em "Reload Now" (Recarregar agora) na janela inferior direita, e o PlatformIO será instalado dentro do VSCode (veja a Figura 3).

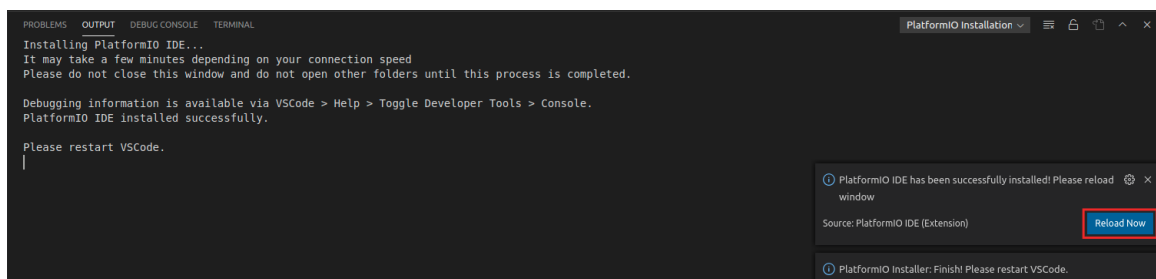


Figura 3. Reiniciar após a instalação do PlatformIO

Na primeira vez em que um exemplo é aberto no PlatformIO, a plataforma Chips Alliance é instalada automaticamente (pode visualizá-la dentro do PIO Home, conforme mostrado na Figura 4). Essa plataforma inclui várias ferramentas que usará mais tarde, como uma toolchain de ferramentas RISC-V pré-construída, OpenOCD para RISC-V, um ficheiro bitstream RVfpgaNexys e RVfpgaSim, scripts JavaScript e Python e vários exemplos.

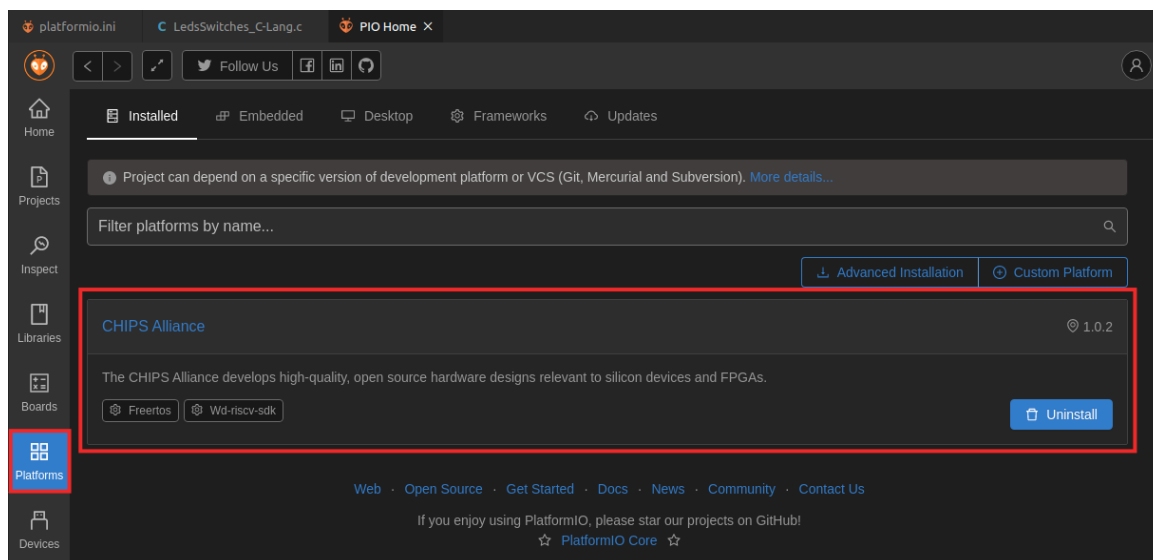






Figura 4. Plataforma Chips Alliance instalada no PlatformIO

Se, por algum motivo, a plataforma Chips Alliance não foi instalada automaticamente quando abriu um exemplo no Lab 2, pode instalá-la manualmente seguindo as próximas etapas:

- Visualize o menu de acesso rápido clicando no botão  , localizado na barra lateral esquerda (consulte a Figura 5). Em seguida, no PIO Home, clique no botão  e, depois, na guia  (Figura 5). Procure **Chipsalliance** (a plataforma que usamos no RVfpga) e abra-a clicando no botão  (Figura 5).

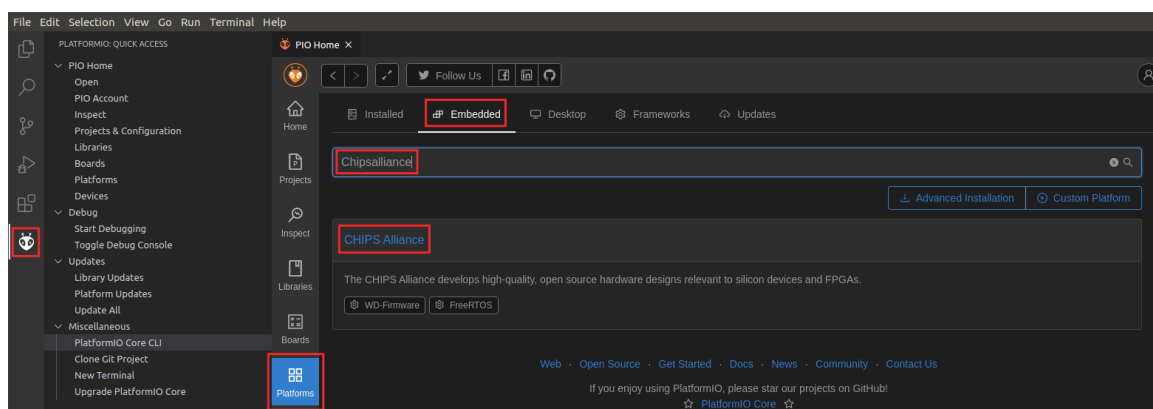




Figura 5. Seleção da plataforma CHIPS Alliance

- Depois de clicar no botão  , verá os detalhes da plataforma Chips Alliance (como na Figura 6). Instale-a clicando no botão  (Figura 6).

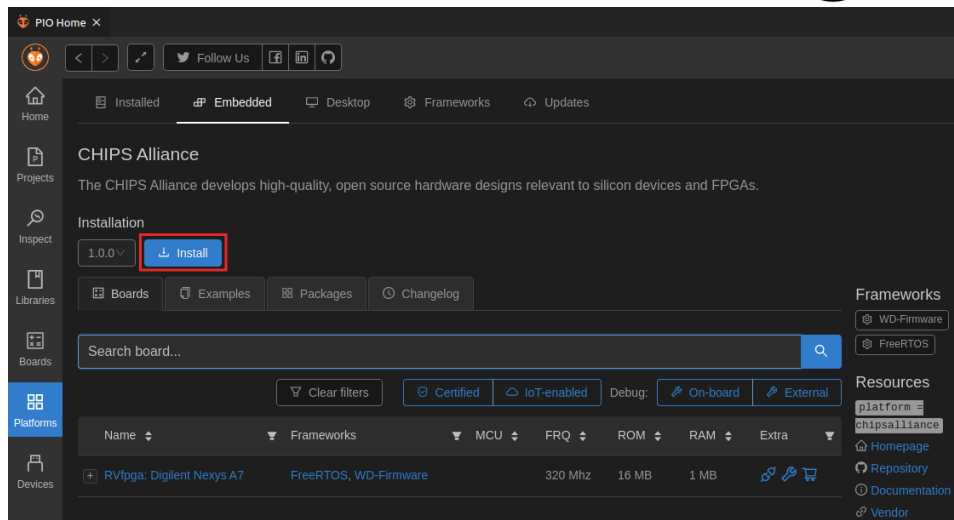


Figura 6. Instalação da plataforma CHIPS Alliance

- Após a conclusão da instalação, é exibido um resumo das ferramentas que foram instaladas, como na Figura 7. Clique em **OK** para fechar essa janela.

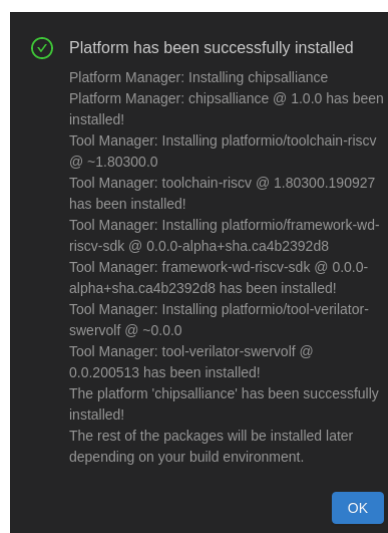


Figura 7. Instalação bem-sucedida da CHIPS Alliance Platform

3. Instale o GTKWave:

Siga as próximas etapas para instalar o GTKWave no sistema Linux Ubuntu 18.04. Abra o terminal do Ubuntu e escreva os seguintes comandos:

- `sudo apt-get install git make autoconf g++ flex bison libfl2 libfl-dev`
- `sudo apt-get install -y gtkwave`

Windows: Consulte o Apêndice B para obter informações sobre a instalação do GTKWave para Windows.

4. Instale o Verilator:

Siga as próximas etapas para instalar o Verilator (as instruções estão disponíveis em <https://www.veripool.org/projects/verilator/wiki/Installing>, mas também estão resumidas abaixo)

- `git clone https://github.com/verilator/verilator`
- `cd verilator`
- `git pull`
- `git checkout v4.106`
- `autoconf`
- `./configure`
- `make` (como alternativa, pode usar `make -j$(nproc)` para acelerar o processo)
- `sudo make install`
- `export PATH=$PATH:/usr/local/bin` (altera o PATH no sistema)

Para adicionar `/usr/local/bin` permanentemente ao PATH, adicione a última linha ao ficheiro `~/.bashrc`.

Observação: A instalação pode demorar um pouco, dependendo do computador.

Windows: Consulte o Apêndice B para obter informações sobre a instalação do Verilator para Windows.

4. Instalação para o Lab 3

As instruções de instalação do Lab 3 em diante são específicas para o sistema operativo Ubuntu 18.04. Os utilizadores do Windows 10 podem executar as partes de simulação dos laboratórios usando o [Windows Subsystem for Linux](#). As instruções funcionarão da mesma forma para qualquer outra versão mais recente do Ubuntu. É altamente recomendado que execute todos esses laboratórios a seguir no sistema operativo Ubuntu.

Esta seção mostrará como instalar o software necessário para realizar o Lab 3 do curso RVfpga-SoC.

1. Instalar o pip:

O "pip" é necessário para a instalação do "FuseSoC" e do "west". Abra um terminal do Ubuntu e escreva o seguinte comando:

- `sudo apt install python3-pip`

2. Instale o pyelftools:

O pyelftools é necessário para a compilação com west. Podemos instalar o pyelftools de duas formas:

- `pip3 install pyelftools`
ou
- `sudo apt-get install -y python3-pyelftools python-pyelftools`

3. Instale o FuseSoC:

Para instalar a versão estável atual do FuseSoC, abra uma janela de terminal e execute o seguinte comando. Se uma versão mais antiga do FuseSoC for encontrada no sistema, isso atualizará a versão para a versão estável mais recente.

```
> sudo pip3 install --upgrade fusesoc
```

Observação: a execução do "apt upgrade" tem 261 pacotes. Espere um tempo de processamento maior. ⌚

4. Instale o OpenOCD:

O OpenOCD é um depurador aberto no chip que permite que os utilizadores programar e depurar dispositivos embebidos. Siga as próximas etapas para instalar o RISC-V OpenOCD no computador:

Etapla 1. Use o "apt-get" para instalar as dependências necessárias:

```
> sudo apt-get install libusb-1.*
> sudo apt-get install pkg-config
> sudo apt-get install libtool
```

Etapla 2. Clone o repositório riscv-openocd do github:

```
> git clone https://github.com/riscv/riscv-openocd.git
> cd riscv-openocd
> ./bootstrap
```

Etapla 3. Faça o download e instale os ficheiros de desenvolvimento da biblioteca de programação USB (user space).

```
> sudo apt-get install libusb-1.0-0-dev
```

Etapla 4. Configure o servidor JTAG ao qual o OpenOCD se pode ligar

```
> ./configure --enable-jtag_vpi --enable-ftdi
> make
> sudo make install
```

5. Instalação para o Lab 4

Esta seção mostrará como instalar o software necessário para realizar o Lab 4 do curso RVfpga-SoC.

Abra o terminal do Ubuntu e escreva os seguintes comandos:

1. Pré-requisitos e dependências:

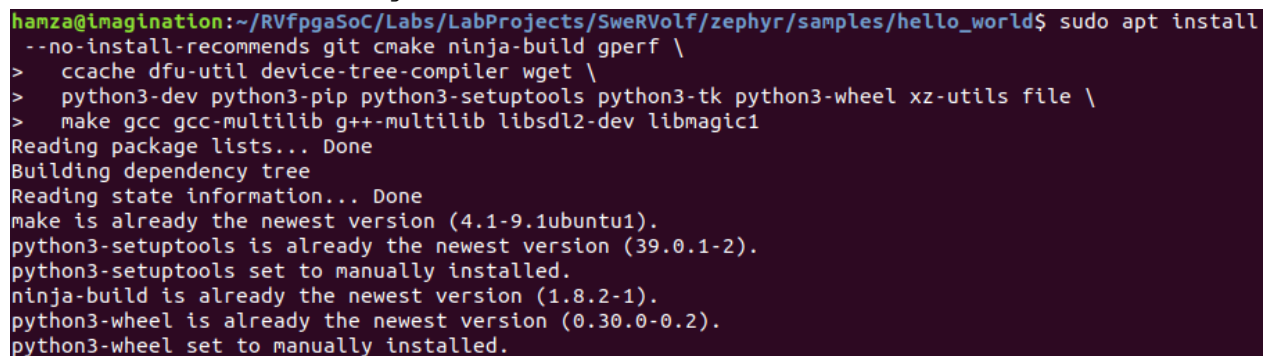
Etapla 1. Atualize os repositórios usando:

```
> sudo apt update
> sudo apt upgrade
```

Observação: a execução do "apt upgrade" no Ubuntu 18.04 atualiza mais de 261 pacotes. Espere um tempo de processamento maior.

Etapla 2. Use o "apt" para instalar as dependências necessárias:

- `sudo apt install --no-install-recommends git cmake
ninja-build gperf \`
 - `ccache dfu-util device-tree-compiler wget \`
 - `python3-dev python3-pip python3-setuptools
python3-tk python3-wheel xz-utils file \`
 - `make gcc gcc-multilib g++-multilib libsdl2-dev
libmagic1`



```
hamza@imagination:~/RVfpgaSoC/Labs/LabProjects/SweRVolf/zephyr/samples/hello_world$ sudo apt install
--no-install-recommends git cmake ninja-build gperf \
> ccache dfu-util device-tree-compiler wget \
> python3-dev python3-pip python3-setuptools python3-tk python3-wheel xz-utils file \
> make gcc gcc-multilib g++-multilib libsdl2-dev libmagic1
Reading package lists... Done
Building dependency tree
Reading state information... Done
make is already the newest version (4.1-9.1ubuntu1).
python3-setuptools is already the newest version (39.0.1-2).
python3-setuptools set to manually installed.
ninja-build is already the newest version (1.8.2-1).
python3-wheel is already the newest version (0.30.0-0.2).
python3-wheel set to manually installed.
```

Figura 8. Instalação de Dependências

2. Atualize a versão do *cmake* para 3.13.1 ou superior:

Pode seguir as instruções para adicionar o [repositório apt de terceiros do kitware](https://apt.kitware.com/) para obter uma versão atualizada do cmake usando o apt.

Etapla 3. Escreva os seguintes comandos para fazer download e instalar/atualizar o cmake:

- `wget -O - https://apt.kitware.com/keys/kitware-
archive-latest.asc 2>/dev/null |`
- `sudo apt-key add -`
- `sudo apt-add-repository 'deb
https://apt.kitware.com/ubuntu/ bionic main'`
- `sudo apt update`
- `sudo apt install cmake`

Observação: Se não conseguir atualizar o "cmake" usando as instruções acima? Não se preocupe! Acesse o link abaixo para obter instruções alternativas sobre como atualizar o "cmake" e garantir que tem a versão mais recente. Basta seguir as etapas fornecidas para obter um processo de atualização perfeito.

Link: <https://askubuntu.com/questions/355565/how-do-i-install-the-latest-version-of-cmake-from-the-command-line>

3. Instalar o West:

Etapa 4. Instale o West e certifique-se de que ~/.local/bin esteja em sua variável de ambiente PATH.

- pip3 install --user -U west
- echo 'export PATH=~/.local/bin:"\$PATH"' >> ~/.bashrc
- source ~/.bashrc

4. Instalar o Zephyr SDK :

O Zephyr Software Development Kit (SDK) contém ferramentas para cada uma das arquiteturas suportadas pelo Zephyr. Ele também inclui ferramentas adicionais, como binários QEMU personalizados e um compilador. Instalaremos o Zephyr SDK versão 0.12.4

Escreva os seguintes comandos no diretório **pessoal**:

Etapa 5. Faça o download do [instalador do SDK da versão 0.12.4](https://github.com/zephyrproject-rtos/sdk-ng/releases/download/v0.12.4/zephyr-sdk-0.12.4-x86_64-linux-setup.run):

- wget https://github.com/zephyrproject-rtos/sdk-ng/releases/download/v0.12.4/zephyr-sdk-0.12.4-x86_64-linux-setup.run

Etapa 6. Execute o instalador, instalando o SDK em ~/zephyr-sdk-0.12.4:

- chmod +x zephyr-sdk-0.12.4-x86_64-linux-setup.run
- ./zephyr-sdk-0.12.4-x86_64-linux-setup.run -- -d ~/zephyr-sdk-0.12.4

Se estiver instalando o Zephyr SDK fora de qualquer um desses locais, leia: [Instalar o Kit de Desenvolvimento de Software Zephyr \(SDK\)](#). Não pode mover o diretório do SDK depois de tê-lo instalado.

5. Instalar o PuTTY :

Etapa 7. Escreva o seguinte comando para instalar o PuTTY :

- sudo apt-get install -y putty

6. Apêndice A: Instalação de Drivers no Windows Para Usar o PlatformIO

Para fazer o download do executável do Zadig, navegue até o seguinte site (Figura 9):

<https://zadig.akeo.ie/>

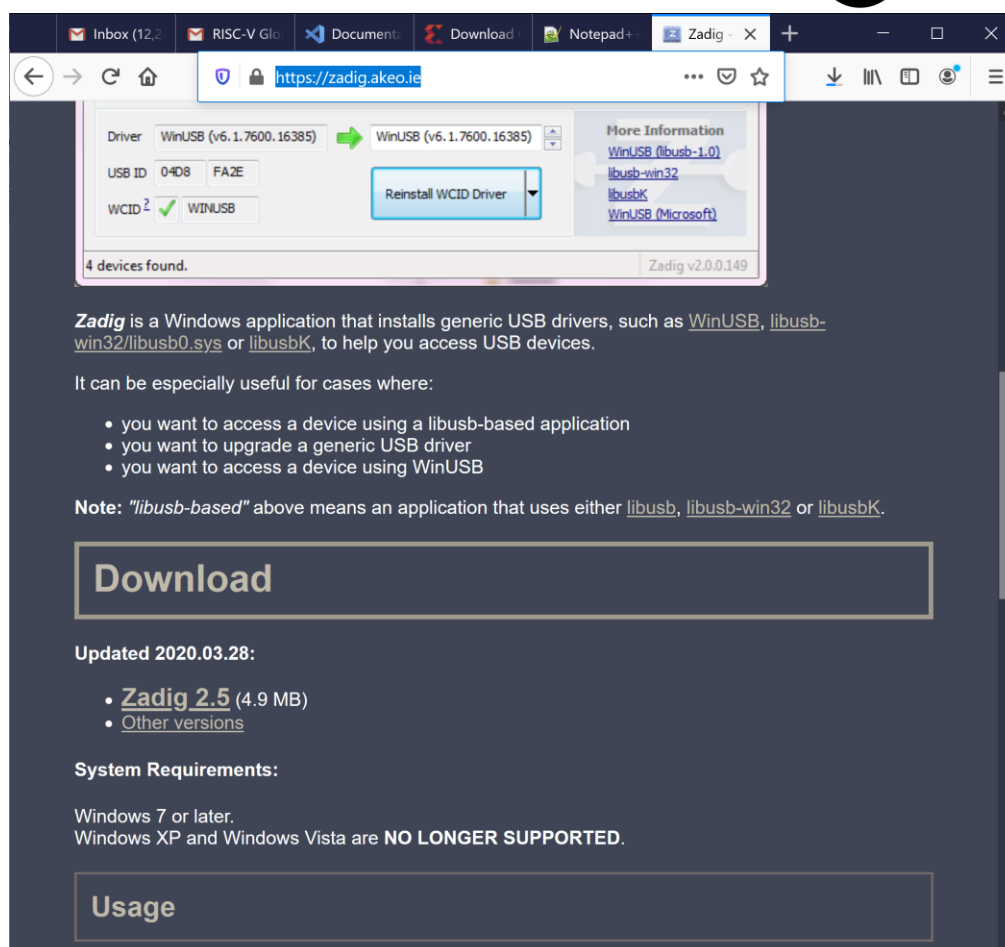


Figura 9. Instalação do driver da placa Nexys A7 usado pelo PlatformIO

Clique em Zadig 2.5 e salve o executável. Em seguida, execute-o (zadig-2.5.exe), que está localizado onde fez o download. Também pode escrever zadig no menu Iniciar para encontrá-lo. Provavelmente será perguntado se deseja permitir que o Zadig faça alterações no computador e se deseja permitir que ele verifique se há atualizações. Clique em Sim nas duas vezes.

Conecte a placa Nexys A7 ao computador e ligue-a. No Zadig, clique em Options → List All Devices (Opções → Listar todos os dispositivos) (ver Figura 10).

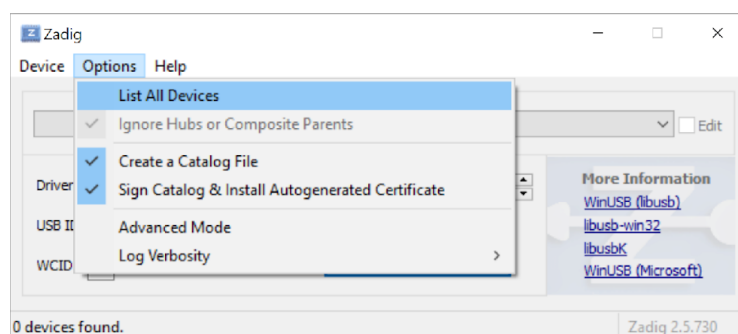


Figura 10. Lista de todos os dispositivos no Zadig

Se clicar no menu *drop-down*, verá listados o Dispositivo USB da Digilent (Interface 0) e o Dispositivo USB da Digilent (Interface 1). Instalará novos drivers apenas para o Digilent USB Device (Interface 0) (Figura 11).

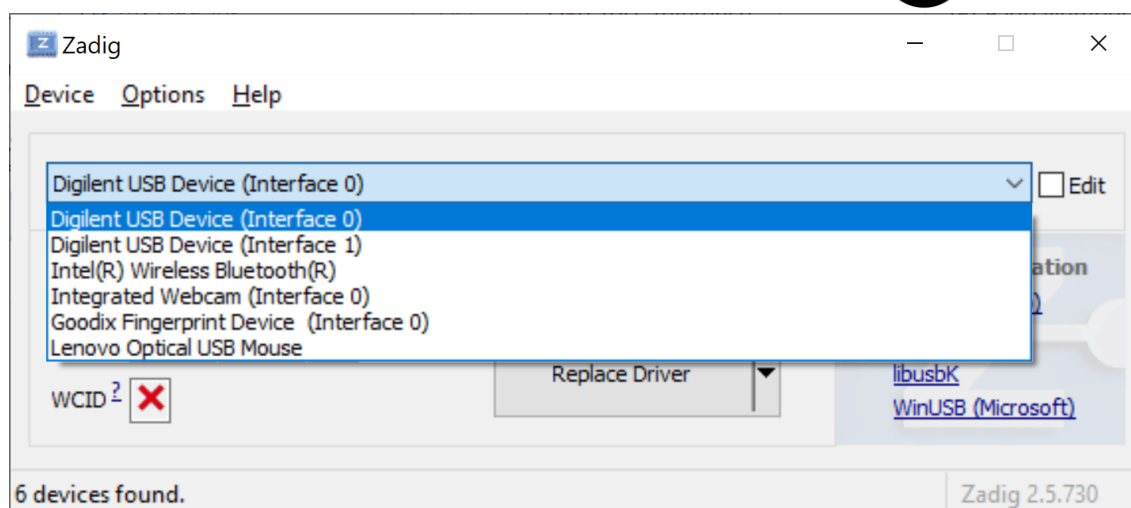


Figura 11. Instalação do driver WinUSB para o dispositivo USB da Digilent (Interface 0)

Agora substituirá o driver FTDI pelo driver WinUSB, conforme mostrado na Figura 12. Clique em Replace Driver (ou Install Driver) para Digilent USB Device (Interface 0). Estará a instalar o driver para a placa Nexys A7 ou, se tiver instalado o Vivado anteriormente, está substituindo o driver FTDI usado pelo Vivado pelo driver WinUSB usado pelo PlatformIO.

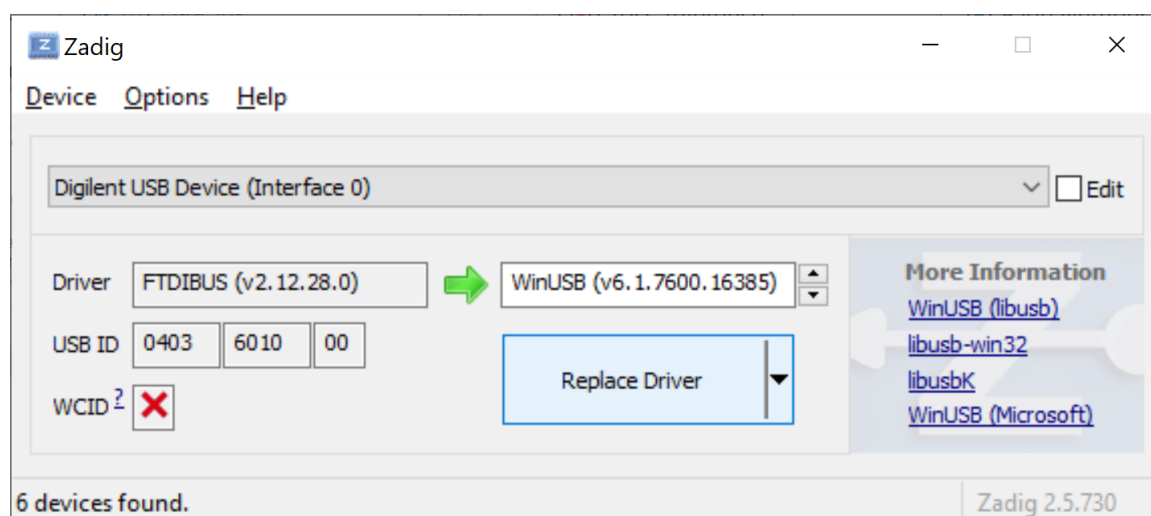


Figura 12. Substituir o driver da placa Nexys A7

Após algum tempo, normalmente vários minutos, o Zadig indicará que o driver foi instalado corretamente. Clique em Close e, em seguida, feche a janela do Zadig.

Na próxima vez que usar o PlatformIO, não será necessário reinstalar o driver. Entretanto, observe que **esse driver não é compatível com o Vivado no Windows**.

7. Apêndice B: Instalação do Verilator e do GTKWave no Windows

Nesta seção, explicaremos como instalar o Verilator e o GTKWave no Windows 10. No Windows, deve usar o Cygwin para instalar o Verilator, portanto, primeiro explicaremos como instalar esse ambiente de programação/runtime.

1. Instalação do Cygwin:

Conforme descrito em sua página da Web (<https://www.cygwin.com>), o Cygwin consiste em ferramentas GNU e de código aberto que fornecem funcionalidade no Windows semelhante à de uma distribuição Linux. Siga as próximas etapas para instalar o Cygwin no Windows 10.

1. Navegue até a página da Web de instalação (<https://cygwin.com/install.html>) e faça o download do ficheiro de instalação, chamado `setup-x86_64.exe` (Figura 13).

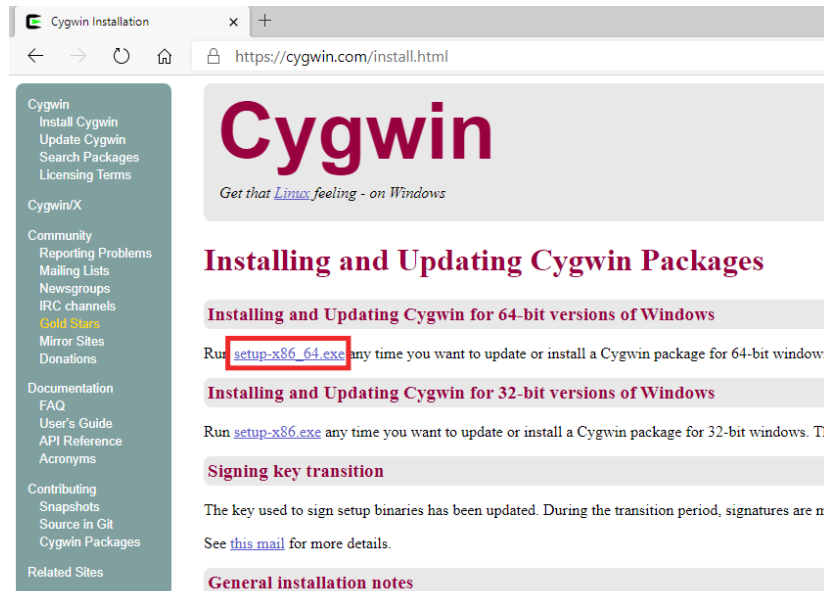


Figura 13. Página da Web de instalação do Cygwin

2. Execute o ficheiro de configuração em sua máquina clicando duas vezes nele (Figura 14). Clique em **Next** várias vezes, mantendo as opções padrão. O instalador solicitará que escolha **um site de download** (Figura 15); pode escolher qualquer um deles.

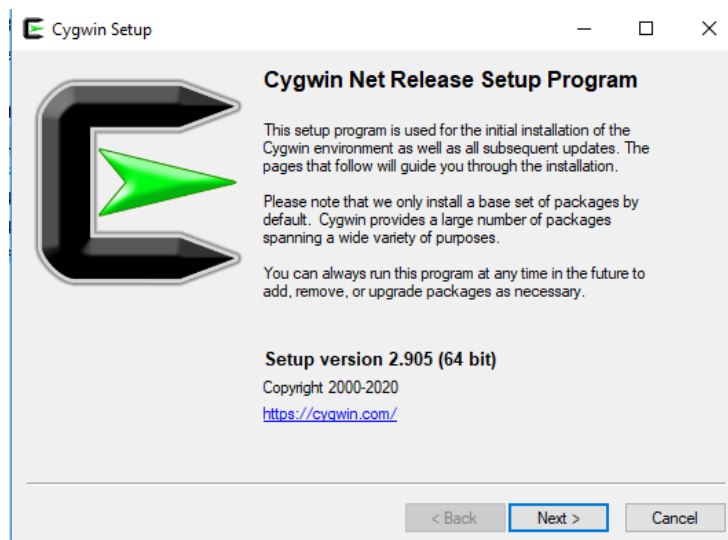


Figura 14. Janela de instalação do Cygwin

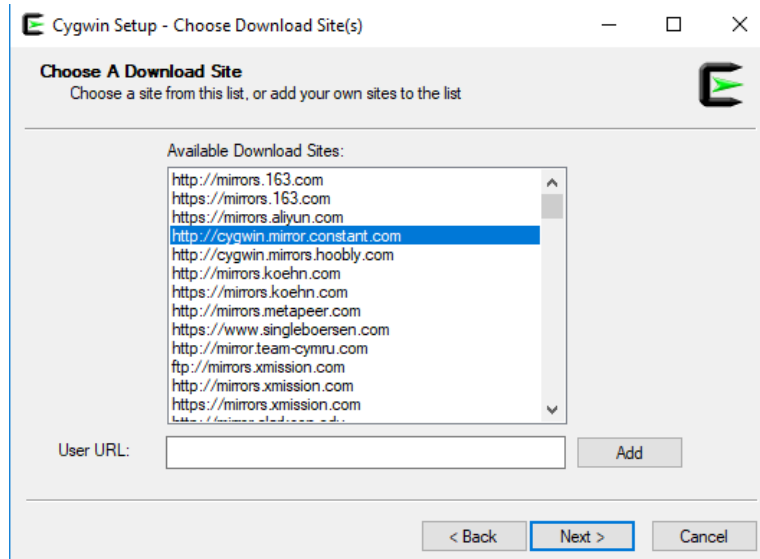


Figura 15. Selecionar site de download

3. Após várias etapas, chegará à janela **Select Packages (Selecionar pacotes)** (Figura 16). Selecione a visualização **Completa**, conforme mostrado na Figura 16.

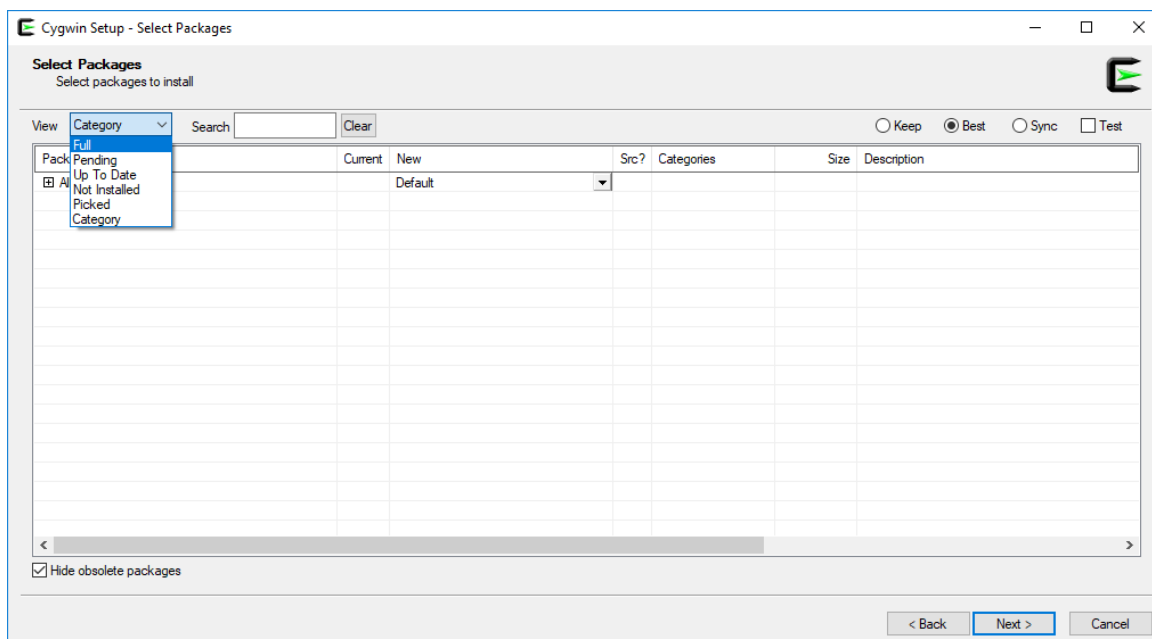


Figura 16. Janela Selecionar Pacotes

4. A lista completa de pacotes que pode instalar será exibida (Figura 17). Na caixa **Pesquisar**, selecione os pacotes específicos que deseja instalar.

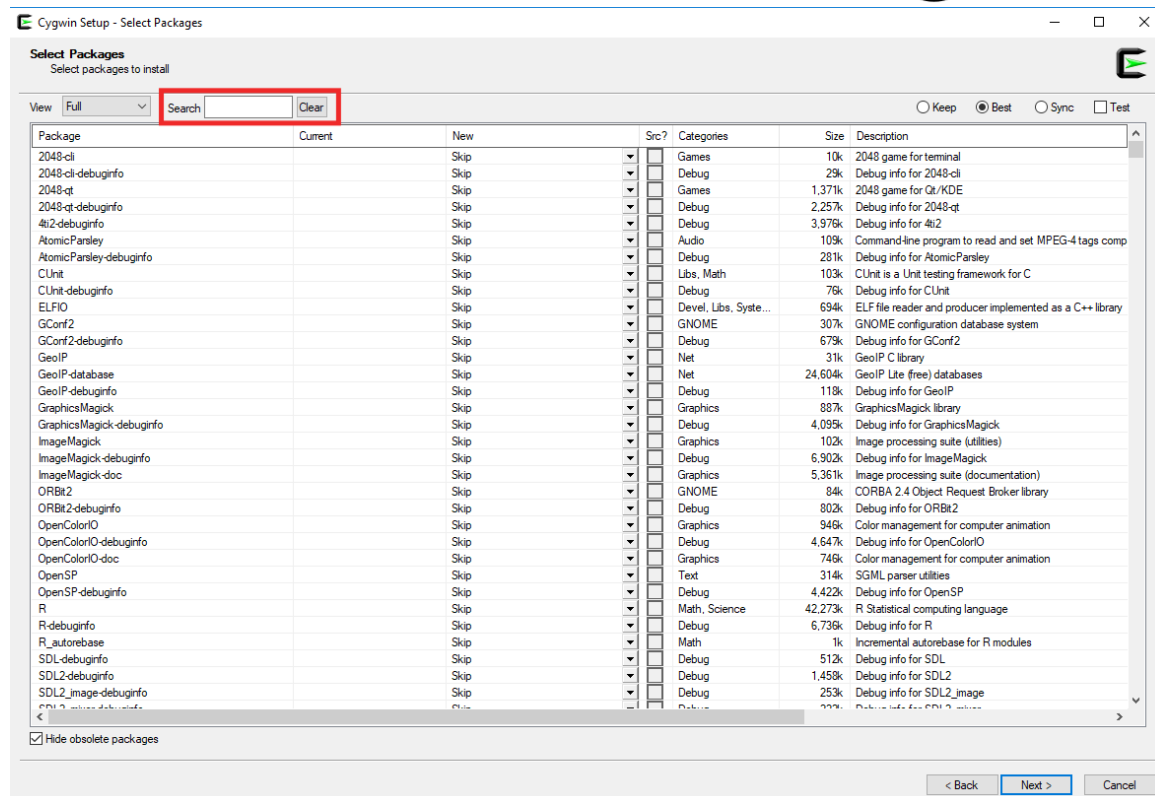


Figura 17. Janela Selecionar pacotes - Visualização completa

Para poder compilar o Verilator e gerar um novo binário do simulador, precisa de instalar os seguintes pacotes:

- git
- make
- autoconf
- gcc-core
- gcc-g++
- flex
- bison
- perl
- libargp-devel

Inclua pelo menos esses pacotes na instalação do Cygwin. Selecione-os individualmente seguindo as etapas abaixo (mostramos apenas as etapas detalhadas para o primeiro pacote da lista, `git`; o processo é o mesmo para os outros pacotes):

- Procure o pacote `git` na caixa **Search** (Figura 18).

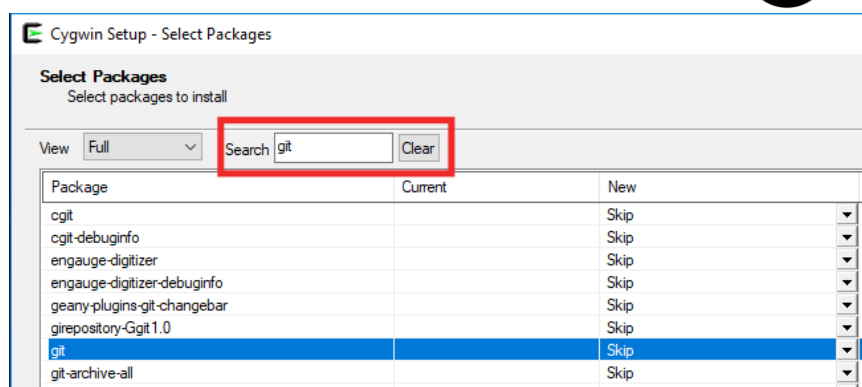


Figura 18. Procure o pacote git

- Selecione a versão mais atualizada no menu suspenso e marque a caixa (Figura 19).



Figura 19. Selecione a versão mais atualizada e marque a caixa

- Faça o mesmo com os pacotes restantes da lista acima.

Mas para os pacotes "gcc-g++" e "gcc-core", instale a versão "10.2.0", pois essa versão é compatível com a versão do Verilator usado nestes laboratórios.

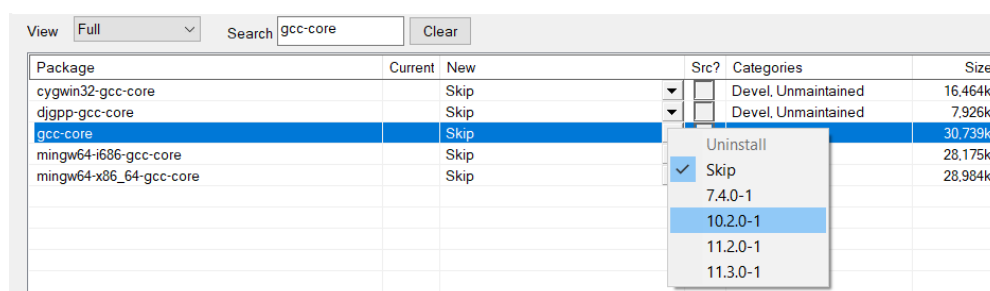


Figura 20. gcc-core versão 10.2.0

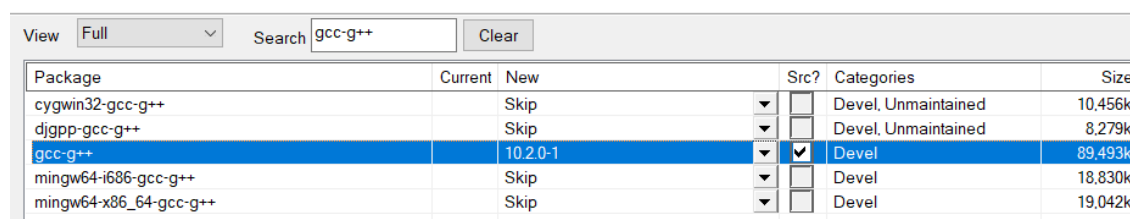


Figura 21. gcc-g++ versão 10.2.0

- Depois de selecionar os nove pacotes, clique em **Next** nas janelas subsequentes para incluir esses pacotes na instalação do Cygwin (o processo de instalação pode levar vários minutos, veja a Figura 22) e finalize a instalação clicando em Finish (Figura 23).

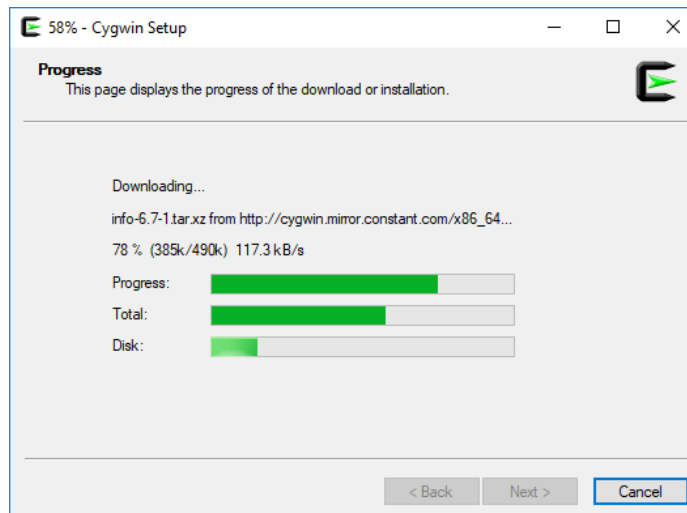


Figura 22. Configuração do Cygwin

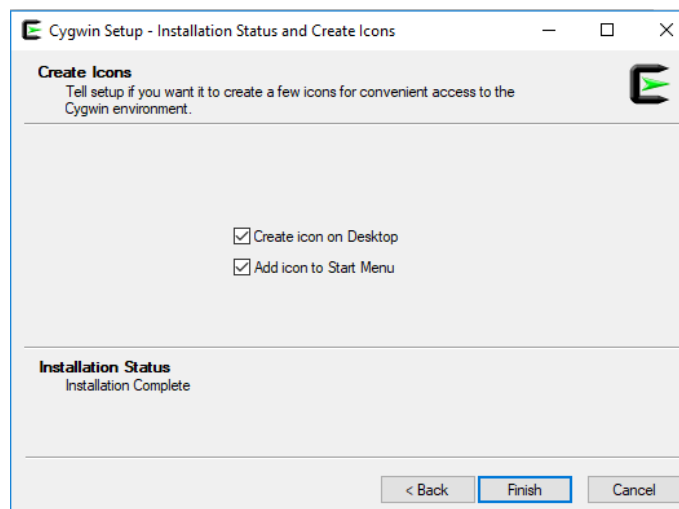


Figura 23. Concluir a instalação

6. Se precisar adicionar um pacote à sua instalação do Cygwin, repita as etapas 2 a 5 para esse pacote.

2. Instalação do Verilator no Windows:

Siga as próximas etapas para instalar o Verilator no Windows 10.

1. Abra o terminal Cygwin (Figura 24), disponível na área de trabalho do Windows ou no menu Iniciar.



Figura 24. Terminal Cygwin

2. Crie e instale o Verilator seguindo estas etapas. Isso pode levar algum tempo (até mesmo horas), dependendo da velocidade do computador:

```
➤ git clone https://github.com/verilator/verilator
➤ cd verilator
➤ git pull
➤ git checkout v4.106
➤ autoconf
➤ ./configure
➤ make
➤ make install
```

3. Instalação do GTKWave no Windows:

O GTKWave pode ser descarregado como um pacote pré-compilado em <https://sourceforge.net/projects/gtkwave/files/>. Procure o pacote mais recente para Windows (no momento em que este documento foi escrito, ele chamava-se **gtkwave-3.3.100-bin-win64**), faça o download e descompacte-o. Pode encontrar um ficheiro executável chamado *gtkwave* dentro da pasta *bin*, que pode ser executado e usado num computador com Windows.