

Haskell

Ambientes

Local - GHCup

- <https://www.haskell.org/downloads/>
- <https://www.haskell.org/ghcup/#>

Local - GHCup

To install on Linux, macOS, FreeBSD or WSL2

run the following in a terminal (as a non-root user):

```
curl --proto '=https' --tlsv1.2 -sSf https://get-ghcup.haskell.org | sh
```

To install on Windows

run the following in a PowerShell session (as a non-admin user):

```
Set-ExecutionPolicy Bypass -Scope Process  
-Force; [System.Net.ServicePointManager]::SecurityProtocol =  
[System.Net.ServicePointManager]::SecurityProtocol -bor 3072; try {  
Invoke-Command -ScriptBlock ([ScriptBlock]::Create((Invoke-WebRequest  
https://www.haskell.org/ghcup/sh/bootstrap-haskell.ps1  
-UseBasicParsing))) -ArgumentList $true } catch { Write-Error $_ }
```

Local - GHCup

Vamos utilizar GHCup pra instalar:

- GHC: Glasgow Haskell Compiler. Na prática vamos usar Cabal ou Stack
- cabal-install: Ferramenta de instalação do Cabal. Vamos usar Cabal para estruturar os projetos, definir dependências, executar, buildar...
- Stack: Alternativa do Cabal
- haskell-language-server: Para integrar haskell com editores/IDE

```
➔ ~ curl --proto '=https' --tlsv1.2 -sSf https://get-ghcup.haskell.org | sh
```

Welcome to Haskell!

This script can download and install the following binaries:

- * ghcup - The Haskell toolchain installer
- * ghc - The Glasgow Haskell Compiler
- * cabal - The Cabal build tool for managing Haskell software
- * stack - A cross-platform program for developing Haskell projects (similar to cabal)
- * hls - (optional) A language server for developers to integrate with their editor/IDE

ghcup installs only into the following directory,
which can be removed anytime:
/Users/larissa/.ghcup

Press ENTER to proceed or ctrl-c to abort.

Note that this script can be re-run at any given time.



- O Script perguntará se deseja instalar algumas bibliotecas. Recomendo instalar tudo
- Aqui estou usando zsh shell, mas ele deve pedir pra incluir o PATH no bash shell também

```
Detected zsh shell on your system...
```

```
Do you want ghcup to automatically add the required PATH variable to "/Users/lar  
issa/.zshrc"?
```

```
[P] Yes, prepend [A] Yes, append [N] No [?] Help (default is "P").
```

- Se quiserem integrar o Haskell com alguma IDE, por exemplo VSCode, recomendo pressionar [Y] (yes) para o haskell-language-server (HLS)
- Percebam que aqui o default está [N] (no).

```
-----  
Do you want to install haskell-language-server (HLS)?  
HLS is a language-server that provides IDE-like functionality  
and can integrate with different editors, such as Vim, Emacs, VS Code, Atom, ...  
Also see https://haskell-language-server.readthedocs.io/en/stable/  
  
[Y] Yes  [N] No  [?] Help (default is "N").
```

- Recomendo pressionar sim aqui também, assim vocês poderão utilizar as versões do GHC gerenciadas pelo GHCup.

```
-----  
Do you want to enable better integration of stack with GHCup?  
This means that stack won't install its own GHC versions, but uses GHCup's.  
For more information see:  
  https://docs.haskellstack.org/en/stable/yaml_configuration/#ghc-installation-c  
ustomisation-experimental  
If you want to keep stacks vanilla behavior, answer 'No'.  
  
[Y] Yes  [N] No  [?] Help (default is "Y").
```


| % Total | % Received | % Xferd | Average Speed | Time | Time | Time | Current |
|-----------|------------|---------|---------------|---------|---------|----------|---------|
| | | | Dload Upload | Total | Spent | Left | Speed |
| 100 27.5M | 100 27.5M | 0 0 | 430k 0 | 0:01:05 | 0:01:05 | --:--:-- | 517k |

[Info] downloading: <https://raw.githubusercontent.com/haskell/ghcup-metadata/master/ghcup-0.0.7.yaml> as file /Users/larissa/.ghcup/cache/ghcup-0.0.7.yaml

| % Total | % Received | % Xferd | Average Speed | Time | Time | Time | Current |
|---------|------------|---------|---------------|----------|---------|----------|---------|
| | | | Dload Upload | Total | Spent | Left | Speed |
| 0 0 | 0 0 | 0 0 | 0 0 | --:--:-- | 0:00:02 | --:--:-- | 0 |

[Info] Upgrading GHCup...

[Warn] No GHCup update available

System requirements

Note: On OS X, in the course of running ghcup you will be given a dialog box to install the command line tools. Accept and the requirements will be installed for you. You will then need to run the command again.

On Darwin M1 you might also need a working llvm installed (e.g. via brew) and have the toolchain exposed in PATH.

Press ENTER to proceed or ctrl-c to abort.

Installation may take a while.

```
=====
OK! /Users/larissa/.zshrc has been modified. Restart your terminal for the changes to take effect,
or type "source /Users/larissa/.ghcup/env" to apply them in your current terminal session.
=====
```

All done!

To start a simple repl, run:
ghci

To start a new haskell project in the current directory, run:
cabal init --interactive

To install other GHC versions and tools, run:
ghcup tui

If you are new to Haskell, check out <https://www.haskell.org/ghcup/steps/>
→ ~

Local - GHCup

- Depois de instalar, só reiniciar o terminal.
- O ghcup servirá agora como um gerenciador de versões, vocês podem ter múltiplas versões do GHC instaladas e usar em diferentes versões em diferentes projetos
- Para gerenciar rodem o comando:

```
ghcup tui
```

GHCup

| | Tool | Version | Tags | Notes |
|----|-------|----------|-----------------------|-------------|
| x | Stack | 2.11.1 | latest | |
| ✓✓ | Stack | 2.9.3 | recommended | |
| x | Stack | 2.9.1 | | |
| x | Stack | 2.7.5 | | |
| ✓✓ | HLS | 2.2.0.0 | latest, recommended | |
| x | HLS | 2.1.0.0 | | |
| ✓ | HLS | 2.0.0.1 | | |
| ✓ | HLS | 2.0.0.0 | | |
| ✓ | HLS | 1.10.0.0 | | |
| x | HLS | 1.9.1.0 | | |
| x | cabal | 3.10.1.0 | latest | |
| x | cabal | 3.8.1.0 | | |
| ✓✓ | cabal | 3.6.2.0 | recommended | |
| x | GHC | 9.6.2 | latest, base-4.18.0.0 | hls-powered |

q:Quit i:Install u:Uninstall s:Set c:ChangeLog a:Show all versions t:Show all tools ↑:Up ↓:Down

Local - GHC

Vocês podem verificar se a instalação do ghc funcionou de forma correta:

```
ghc --version
```

Ambiente Interativo:

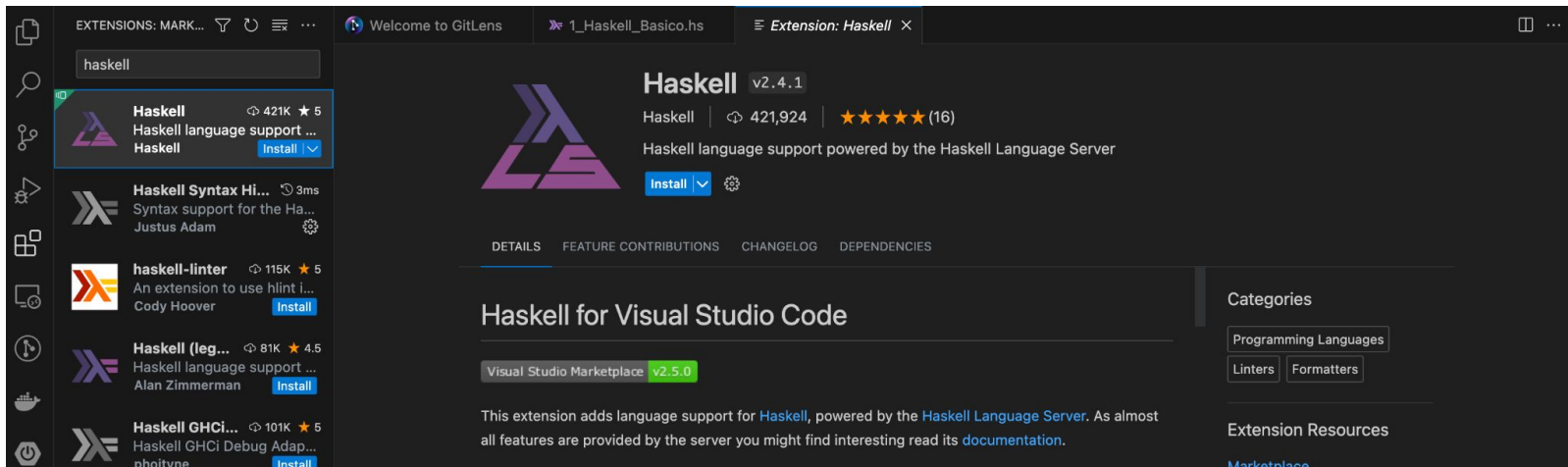
```
ghci
```

```
haskell ghc
GHCi, version 9.2.8: https://www.haskell.org/ghc/  :? for help
[ghci> :load 1_Haskell_Basico.hs
[1 of 1] Compiling Main                  ( 1_Haskell_Basico.hs, interpreted )
Ok, one module loaded.
[ghci> square 4
16
[ghci> allEqual 1 1 1
True
```

Local - VSCode

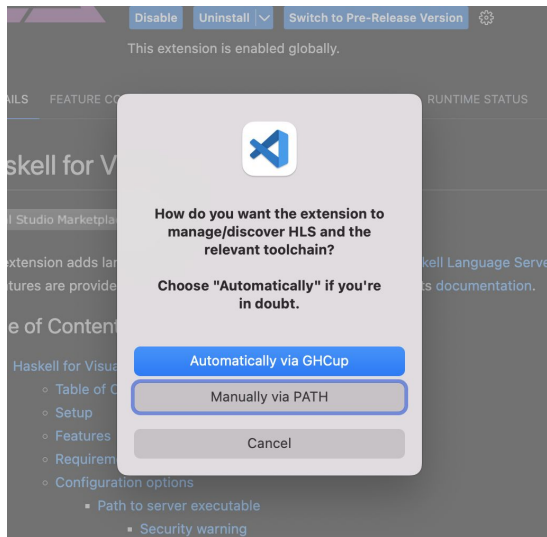
Vocês podem utilizar qualquer editor de texto para editar arquivos haskell, mas aqui mostraremos a integração do Haskell com o **VSCode**

- Instalem a extensão do Haskell no VSCode



Local - VSCode

- Depois de instalar, aparecerá se você deseja que a extensão gerencie o HLS (que instalamos no passo anterior via GHCup)
- Pressionem "Automatically via GHCup"



Local - VSCode

- Só abrir qualquer arquivo .hs (haskell) com a IDE. Aqui abri o 1_Haskell_Basico.hs
- Observem que já aparecerá inferência de tipos, highlights, correções e outras features

```
Users > larissa > workspace > mestrado > haskell > 1_Haskell_Basico.hs > casoTesteAllEqual_1
1 square :: Int -> Int
2 square x = x * x
3
4
5 allEqual :: Int -> Int -> Int -> Bool
6 allEqual m n p = (m == n) && (n == p)
7
8 casoTesteAllEqual_1 :: Bool
9 casoTesteAllEqual_1 = allEqual 2 2 2 == True
10
11 resultadoCasosTesteAllEqual :: Bool
12 resultadoCasosTesteAllEqual = casoTesteAllEqual_1
13
14 maxi :: Int -> Int -> Int
15 maxi n m
16 | n >= m = n
17 | otherwise = m
18
19 casoTesteMaxi_1 :: Bool
20 casoTesteMaxi_1 = maxi 2 3 == 3
21
22 casoTesteMaxi_2 :: Bool
23 casoTesteMaxi_2 = maxi 60 30 == 60
24
25 resultadoCasosTesteMaxi :: Bool
26 resultadoCasosTesteMaxi = foldl (&&) True [casoTesteMaxi_1, casoTesteMaxi_2]
27
28 fat :: Int -> Int
29 fat n
30 | n == 0 = 1
31 | n > 0 = n * fat (n-1)
32
33 casoTesteFat_1 :: Bool
34 casoTesteFat_1 = fat 5 == 120
35
36 casoTesteFat_2 :: Bool
37 casoTesteFat_2 = fat 9 == 362880
```

Redundant ==

Found:

allEqual 2 2 2 == True

Why not:

allEqual 2 2 2

hlint(refact:Redundant ==)

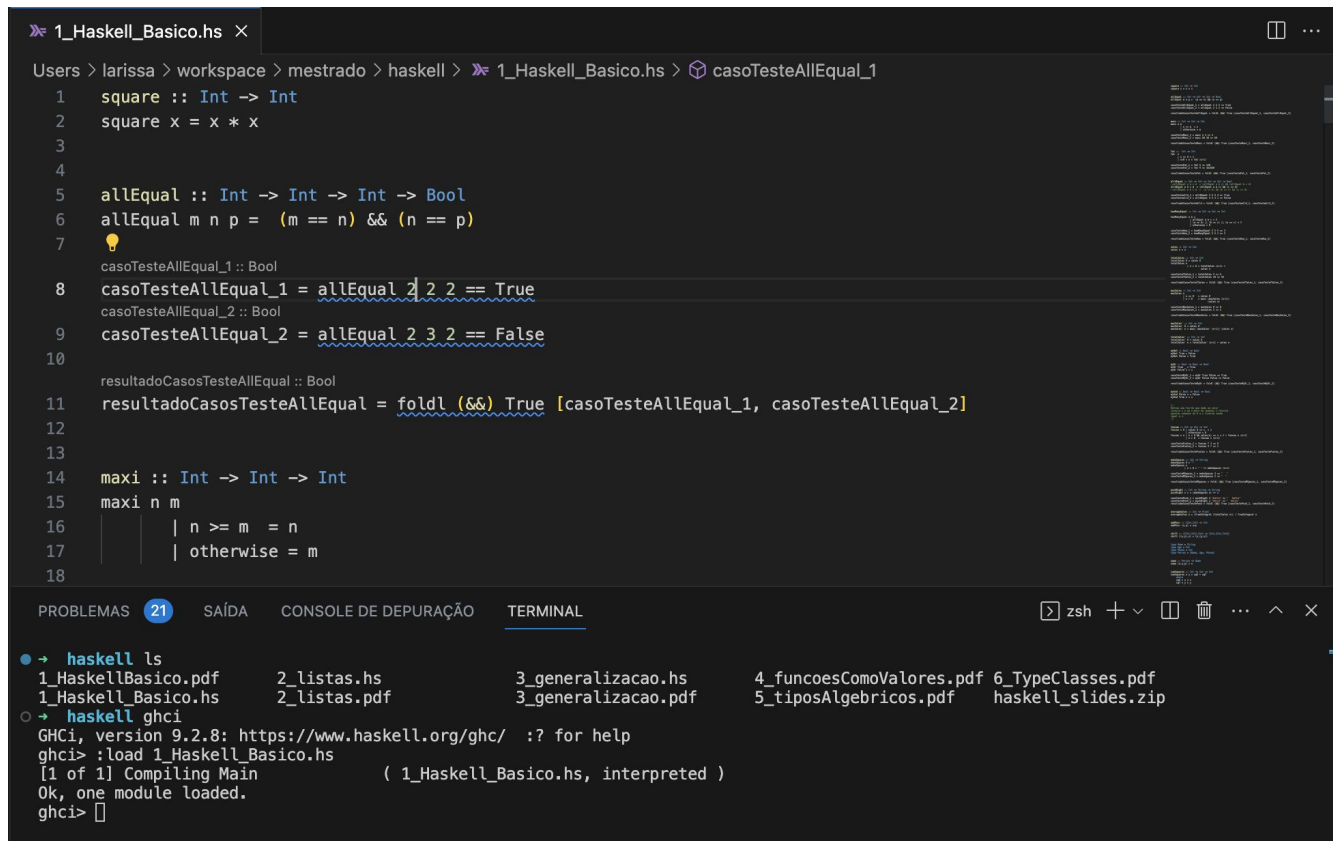
allEqual :: Int -> Int -> Int -> Bool

Defined at /Users/larissa/workspace/mestrado/haskell/1_Haskell_Basico.hs:6:1

Exibir o Problema (⌘F8) Correção Rápida... (⌘.)

Local - VSCode

- Vocês podem usar o próprio terminal do vscode para rodar o ghci, cabal ou stack



The screenshot shows the VS Code interface with a Haskell file named `1_Haskell_Basico.hs` open. The file contains several Haskell functions and test cases. The terminal at the bottom shows the output of running `ghci` and loading the file.

```
1 square :: Int -> Int
2 square x = x * x
3
4
5 allEqual :: Int -> Int -> Int -> Bool
6 allEqual m n p = (m == n) && (n == p)
7
8 casoTesteAllEqual_1 :: Bool
9 casoTesteAllEqual_1 = allEqual 2 2 2 == True
10 casoTesteAllEqual_2 :: Bool
11 casoTesteAllEqual_2 = allEqual 2 3 2 == False
12
13 resultadoCasosTesteAllEqual :: Bool
14 resultadoCasosTesteAllEqual = foldl (&&) True [casoTesteAllEqual_1, casoTesteAllEqual_2]
15
16 maxi :: Int -> Int -> Int
17 maxi n m
18   | n >= m = n
19   | otherwise = m
```

The terminal output shows the following commands and results:

```
haskell ls
1_HaskellBasico.pdf 2_listas.hs 3_generalizacao.hs 4_funcoesComoValores.pdf 6_TypeClasses.pdf
1_Haskell_Basico.hs 2_listas.pdf 3_generalizacao.pdf 5_tiposAlgebricos.pdf 7_haskell_slides.zip
haskell ghci
GHCi, version 9.2.8: https://www.haskell.org/ghc/ :? for help
ghci> :load 1_Haskell_Basico.hs
[1 of 1] Compiling Main (1_Haskell_Basico.hs, interpreted)
Ok, one module loaded.
ghci>
```

Utilizando Cabal

```
$ cabal init myfirstapp
```

```
$ cd myfirstapp
```

```
$ cabal run myfirstapp
```

```
$ cabal build myfirstapp
```

```
$ tree
```

```
├── myfirstapp
│   ├── app
│   │   └── Main.hs
│   ├── CHANGELOG.md
│   └── myfirstapp.cabal
```

```
$ cabal run myfirstapp
```

```
...
```

```
Hello, Haskell!
```

Utilizando Cabal

- Vocês podem escrever novo código dentro da pasta app
- As dependências do projeto (biblioteca) ficam no arquivo .cabal (myfirstapp.cabal no nosso caso)
- `cabal install` instala as dependências presentes no .cabal
- `cabal install specific-library` instala uma dependência publicada no cabal

Online - Replit


- <https://replit.com/new/haskell>

Create a Repl

Import from GitHub


Template

Haskell



Haskell ✓

Haskell is a general-purpose, purely functional programming language with type inference.

 replit

18 + 101.8K

Title

SaneCrimsonCharactercode

Public

Anyone can view and fork this Repl.

Upgrade to make private

Create Repl

Online - Replit

- projeto teste (podem copiar, mas não mexam):
<https://replit.com/join/lprthayutw-fischertayna>
- O Replit precisa de um main se não dá erro:

```
GHCi, version 9.0.2: https://www.haskell.org/ghc/  :? for help
Loaded GHCi configuration from /home/runner/SaneCrimsonCharacter/code/.ghci
[1 of 1] Compiling Main                                ( Main.hs, interpreted )
Ok, one module loaded.
>
<interactive>:1:1: error:
  • Variable not in scope: main
  • Perhaps you meant 'min' (imported from Prelude)
>
```



SaneCrimsonCharactercode



▶ Run

Search

▼ Files



HaskellBasico.hs

Main.hs

Main.hs

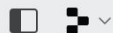


HaskellBasico.hs



Main.hs > Main > main

```
1 module Main where
2
3 import HaskellBasico
4
5 main = do
6     print (square 2)
7     print (allEqual 1 2 3)
8     print (allEqual 2 2 2)
9     print (maxi 2 64)
10    print (maxi 64 2)
```



SaneCrimsonCharactercode



Run

Search

Files



HaskellBasico.hs

Main.hs

Tools



Ghostwriter



Deployments



Chat



Main.hs HaskellBasico.hs

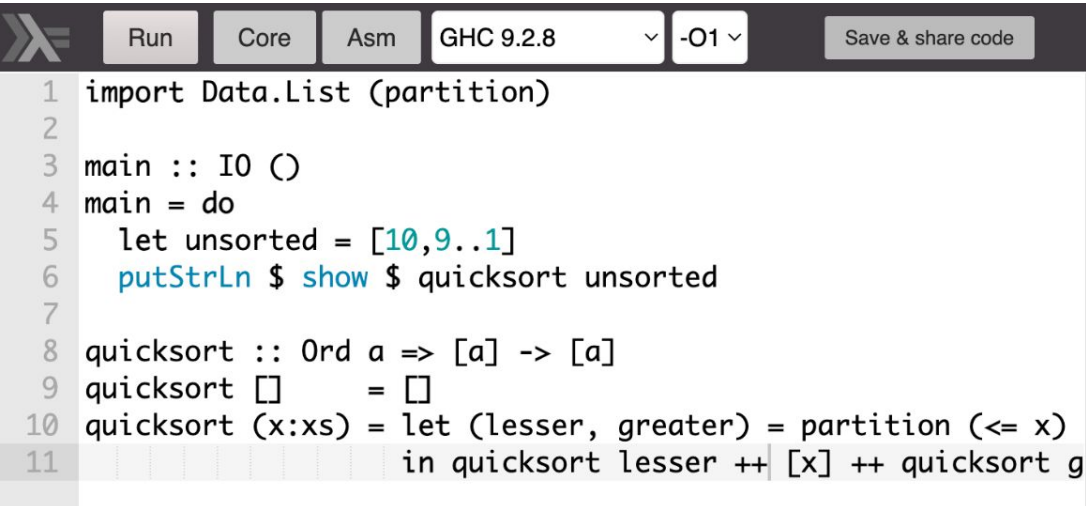


HaskellBasico.hs > HaskellBasico > ...

```
1 module HaskellBasico where
2
3 square :: Int -> Int
4 square x = x * x
5
6
7 allEqual :: Int -> Int -> Int -> Bool
8 allEqual m n p = (m == n) && (n == p)
9
10 casoTesteAllEqual_1 = allEqual 2 2 2 == True
11 casoTesteAllEqual_2 = allEqual 2 3 2 == False
12
13 resultadoCasosTesteAllEqual = foldl (&&) True [casoTesteAllEqual_1,
14 casoTesteAllEqual_2]
15
16 maxi :: Int -> Int -> Int
17 maxi n m
18     | n >= m = n
19     | otherwise = m
20
```

Online - Haskell playground

- O próprio haskell oferece um playground
- Indico apenas para testes curtos
- <https://play.haskell.org/>



The image shows a screenshot of the Haskell Playground web interface. At the top, there is a dark grey header bar with a logo on the left and several buttons: 'Run', 'Core', 'Asm', 'GHC 9.2.8' (with a dropdown arrow), '-O1' (with a dropdown arrow), and 'Save & share code'. Below the header, the main area is split into two panels. The left panel is a code editor with line numbers 1 through 11 on the left margin. It contains Haskell code for a quicksort function. The right panel is the 'OUTPUT' window, which displays the result of the code execution: '[1,2,3,4,5,6,7,8,9,10]'. The code in the editor is as follows:

```
1 import Data.List (partition)
2
3 main :: IO ()
4 main = do
5     let unsorted = [10,9..1]
6     putStrLn $ show $ quicksort unsorted
7
8 quicksort :: Ord a => [a] -> [a]
9 quicksort []      = []
10 quicksort (x:xs) = let (lesser, greater) = partition (<= x)
11                   in quicksort lesser ++ [x] ++ quicksort g
```

OUTPUT

```
[1,2,3,4,5,6,7,8,9,10]
```


Links para estudo

- Documentação haskell: <https://www.haskell.org/documentation/>
- Learn you haskell: <http://learnyouahaskell.com/chapters>