Enferrujando a Web

Tchelinux Porto Alegre

Patrocínio

Conheçam mais no site

poa.tchelinux.org

Michel Martinez

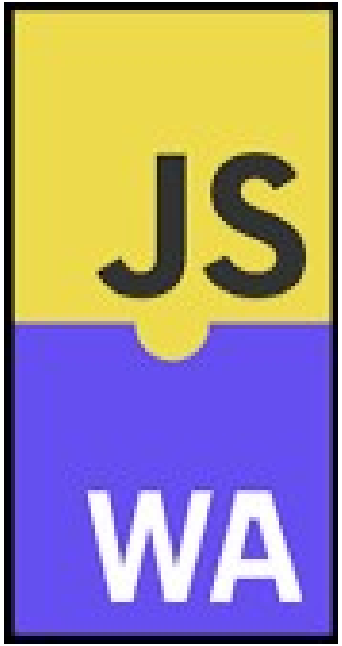
Analista de Sistemas na Getnet

Casado com a Fernanda

Ex-estudante de engenharia e atualmente em ADS


WA
WEBASSEMBLY

O que NÃO é WebAssembly



Um novo JS

Uma nova linguagem que você tem que aprender!

O que é WebAssembly então?

WebAssembly ou a abreviação Wasm é um formato binário para rodar em uma máquina virtual

0000000001100001 0111001101101101 0000000100000000 0000000000000000
0000000000001100 0000011001100100 0111100101101100 0110100101101110
0110101110010000 1000000011000000 0000001000000000 0000000110001000
1000000010000000 1000000000000000 0000001001100000 0000000000000001
0111111101100000 0000000000000000 0000001011000001 1000000010000000
1000000000000000 0000010000000011 0110010101101110 0111011000001010
0110110101100101 0110110101101111 0110110101101111 0100001001100001
0111001101100101 0000001101111111 0000000000000011 0110010101101110
0111011000000110 0110110101100101 0110110101101111 0111001001111001
0000001000000000 1000000000000010 0000001101100101 0110111001111011
0011101100000110 0110110101100101 0110110101101111 0111001001111001
0000001000000000 1000000000000010 0000001101100101 0110111001110110
0000010101110100 0110000101100010 0110110001100101 0000000101110000
0000000000000000 0000001101100101 0110111001110110 0000100101110100
0110000101100010 0110110001100101 0100001001100001 0111001101100101
0000001101111111 0000000000000011 1000010010000000 1000000010000000
0000000000000011 0000000000000001 0000000100000110 1001000010000000
1000000010000000 0000000000000011 0111111100000001 0100000100000000

0100000100000000 0000101100000111 1011100010000000 1000000010000000
0000000000000100 0000011001011111 0110001101101111 0111010101101110
0111010000000000 0000000000010010 0101111101011111 0111000001101111
0111001101110100 0101111101101001 0110111001110011 0111010001100001
0110111001110100 0110100101100001 0111010001100101 0000000000000010
0000101101110010 0111010101101110 0101000001101111 0111001101110 10
0101001101100101 0111010001110011 0000000000000001 0000100001011111
0110001101101111 0111010101101110 0111010001100101 0111001000000011
0000010000001001 1000000110000000 1000000010000000 0000000000000000
0000101011000011 1000000010000000 1000000000000000 0000001110011000
1000000010000000 1000000000000000 0000000100000001 0111111100000010
0111111100100011 0000000000100011 0000000000101000 0000001000000000
0100000100000001 0110101000100010 0000000000110110 0000001000000000
0010000000000000 0000101100001011 1000001110000000 1000000010000000
0000000000000000 0000000100001011 1001100010000000 1000000010000000
0000000000000000 0000001001000000 0010001100000000 0100000100010000
0110101000100100 0000001000100011 0000001001000001 1000000010000000
1100000000000010 0110101000100100 0000001100010000 0000000100001011
0000101100001011 1000011110000000 1000000010000000 0000000000000001
0000000000100011 0000000000001011 0000000101100100

Possui um formato texto (.Wat)

```
(module
  (type $t0 (func (result i32)))
  (type $t1 (func))
  (import "env" "memoryBase" (global $g0 i32))
  (import "env" "memory" (memory $M0 256))
  (import "env" "table" (table $T0 0 anyfunc))
  (import "env" "tableBase" (global $g1 i32))
  (func $f0 (export "_count") (type $t0) (result i32)
    (local $l0 i32)
    (block $B0 (result i32)
      (i32.store
        (get_global $g0)
        (tee_local $l0
          (i32.add
            (i32.load
              (get_global $g0))
            (i32.const 1))))
      (get_local $l0)))
  (func $f1 (export "runPostSets") (type $t1)
    (nop))
  (func $f2 (export "__post_instantiate") (type $t1)
    (block $B0
      (set_global $g2
        (i32.add
          (get_global $g0)
          (i32.const 16)))
      (set_global $g3
        (i32.add
          (get_global $g2)
          (i32.const 5242880)))
      (call $f1)))
  (global $g2 (mut i32) (i32.const 0))
  (global $g3 (mut i32) (i32.const 0))
  (global $g4 (export "_counter") i32 (i32.const 0))
  (data (get_global 0) "d"))
```

Porém é feito para ser um target de compilação para outras linguagens


```
cargo build --target wasm32-unknown-unknown
```

Roda em uma máquina virtual

WA

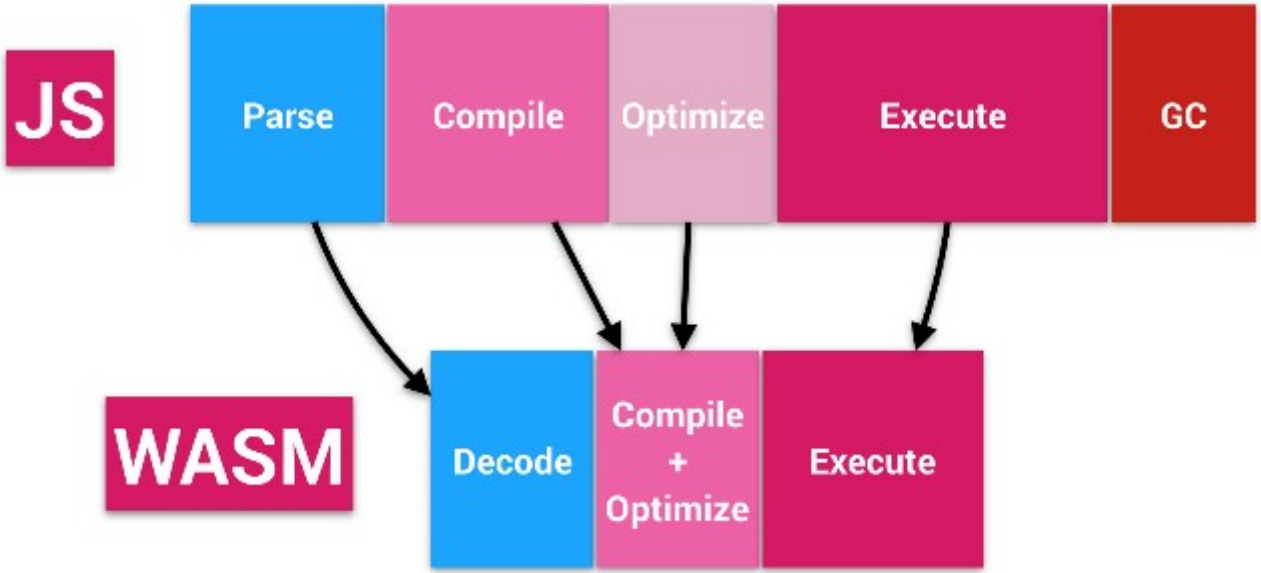
WEBASSEMBLY

Overview | Getting Started | Docs | Spec | Community | Roadmap | FAQ

WebAssembly 1.0 has shipped in 4 major browser engines. Learn more

WebAssembly (abbreviated *Wasm*) is a binary instruction format for a stack-based virtual machine. Wasm is designed as a portable target for compilation of high-level languages like C/C++/Rust, enabling deployment on the web for client and server applications.

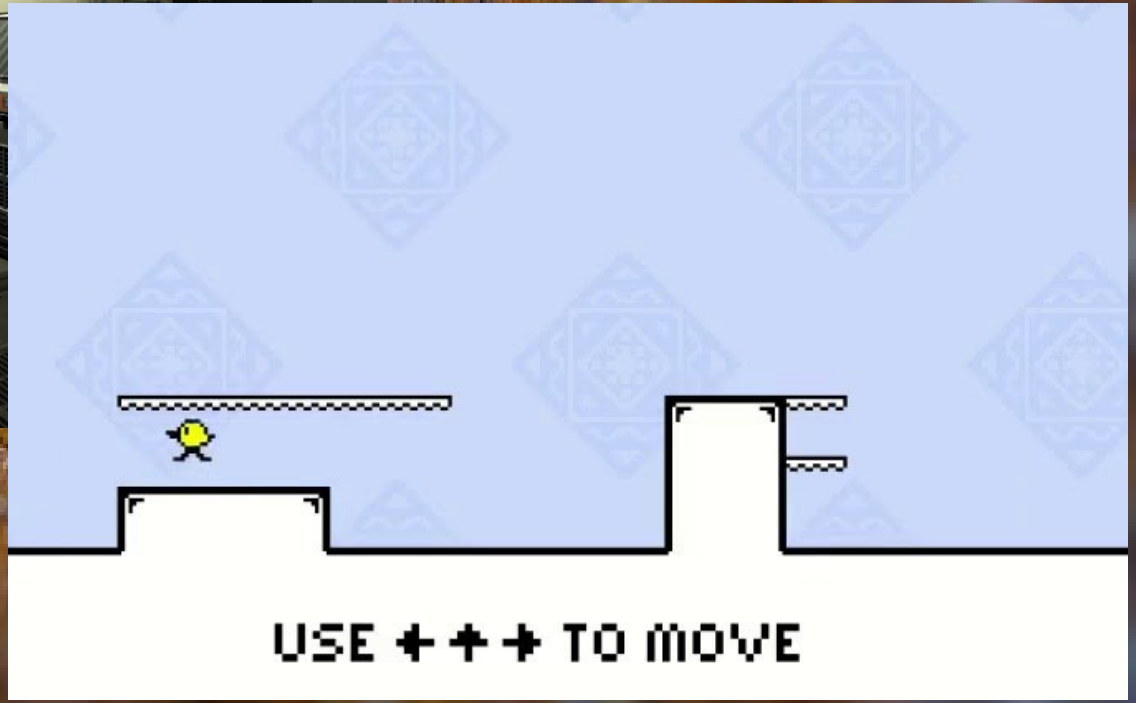
Pensado para ser leve e rápido



Tá e na prática???


25fps
INFORMATION


fps: 57
path: 14


USE ✚✚✚ TO MOVE

Qt

Qt Examples for WebAssembly

Download Qt

Blazor App

Home

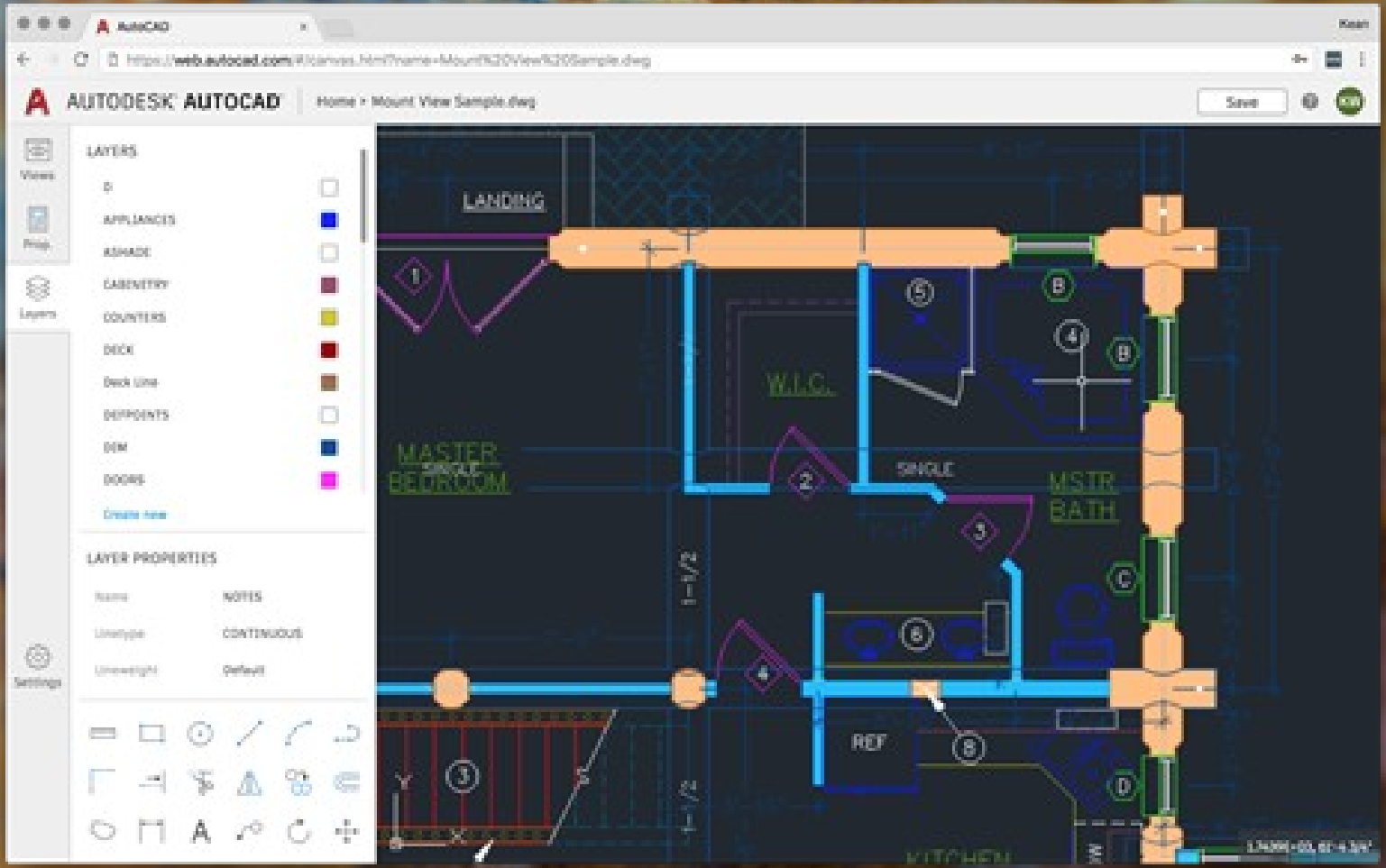
Counter

Fetch data

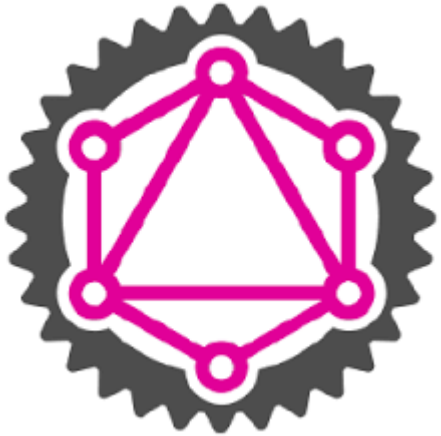
Welcome to your new app.

Blazor

Do you want to *learn more* about Blazor? Yes!


AutoCAD
AUTODESK AUTOCAD
Home › Mount View Sample.dwg
Save
Views
Prop.
Layers
Settings
LAYERS
0
APPLIANCES
ASHADE
CABINETRY
COUNTERS
DECK
Deck Line
DEFPOINTS
DEM
DOORS
Create new
LAYER PROPERTIES
Name
NOTES
Linetype
CONTINUOUS
Linewidth
Default
LANDING
W.I.C.
MASTER BEDROOM
SINGLE
MSTR BATH
REF
KITCHEN

graphql_client



build failing | docs 0.8.0 | crates.io v0.8.0 | gitter join chat

A typed GraphQL client library for Rust.

Features

- Precise types for query variables and responses.
- Supports GraphQL fragments, objects, unions, inputs, enums, custom scalars and input objects.
- Works in the browser (WebAssembly).
- Subscriptions support (serialization-deserialization only at the moment).
- Copies documentation from the GraphQL schema to the generated Rust code.
- Arbitrary derives on the generated responses.
- Arbitrary custom scalars.
- Supports multiple operations per query document.
- Supports setting GraphQL fields as deprecated and having the Rust compiler check their use.
- web client for boilerplate-free API calls from browsers.

E por onde começo?

webassembly.studio

BETA WebAssembly Studio

Fork | Create Gist | Download | Share | Build | Run | Build & Run

Preview README.md | main.c

- README.md
- build.ts
- package.json
- src
  - main.c
  - main.html
  - main.js
- out
  - main.wasm

```c
#include <stdio.h>
#include <sys/uio.h>

#define WASM_EXPORT __attribute__((visibility("default")))

WASM_EXPORT
int main(void) {
  printf("Hello World\n");
}

/* External function that is implemented in JavaScript. */
extern void putc_js(char c);

/* Basic implementation of the writev sys call. */
WASM_EXPORT
size_t writev_c(int fd, const struct iovec *iov, int iovcnt) {
  size_t cnt = 0;
  for (int i = 0; i < iovcnt; i++) {
    for (int j = 0; j < iov[i].iov_len; j++) {
      putc_js(((char *)iov[i].iov_base)[j]);
    }
```

Output (4) | Problems (0)

```
[info]: Task build is running...
[info]: Task build is completed
Hello World
```

BETA WebAssembly Studio

Fork | Create Gist | Download | Share | Build | Run | Build & Run | Help & Privacy | GitHub Issues

Preview README.md | lib.rs | Save

- README.md
- build.ts
- package.json
- src
  - lib.rs
  - main.html
  - main.js
- out
  - main_bg.wasm
  - main.js

```rust
// Current prelude for using `wasm_bindgen`, and this'll get smaller over time!
#![feature(proc_macro, wasm_custom_section, wasm_import_module)]
extern crate wasm_bindgen;
use wasm_bindgen::prelude::*;

// Here we're importing the `                                using
// `#[wasm_bindgen]` to gener
#[wasm_bindgen]
extern {
    fn alert(s: &str);
}

// Here we're exporting a fun                                isplay a greeting
// for `name` through a dialog.
#[wasm_bindgen]
pub fn greet(name: &str) {
    alert(&format!("Hello, {}! \ndireto do Rust!", name));
}
```

Hello, The Linux!
direto do Rust!

OK

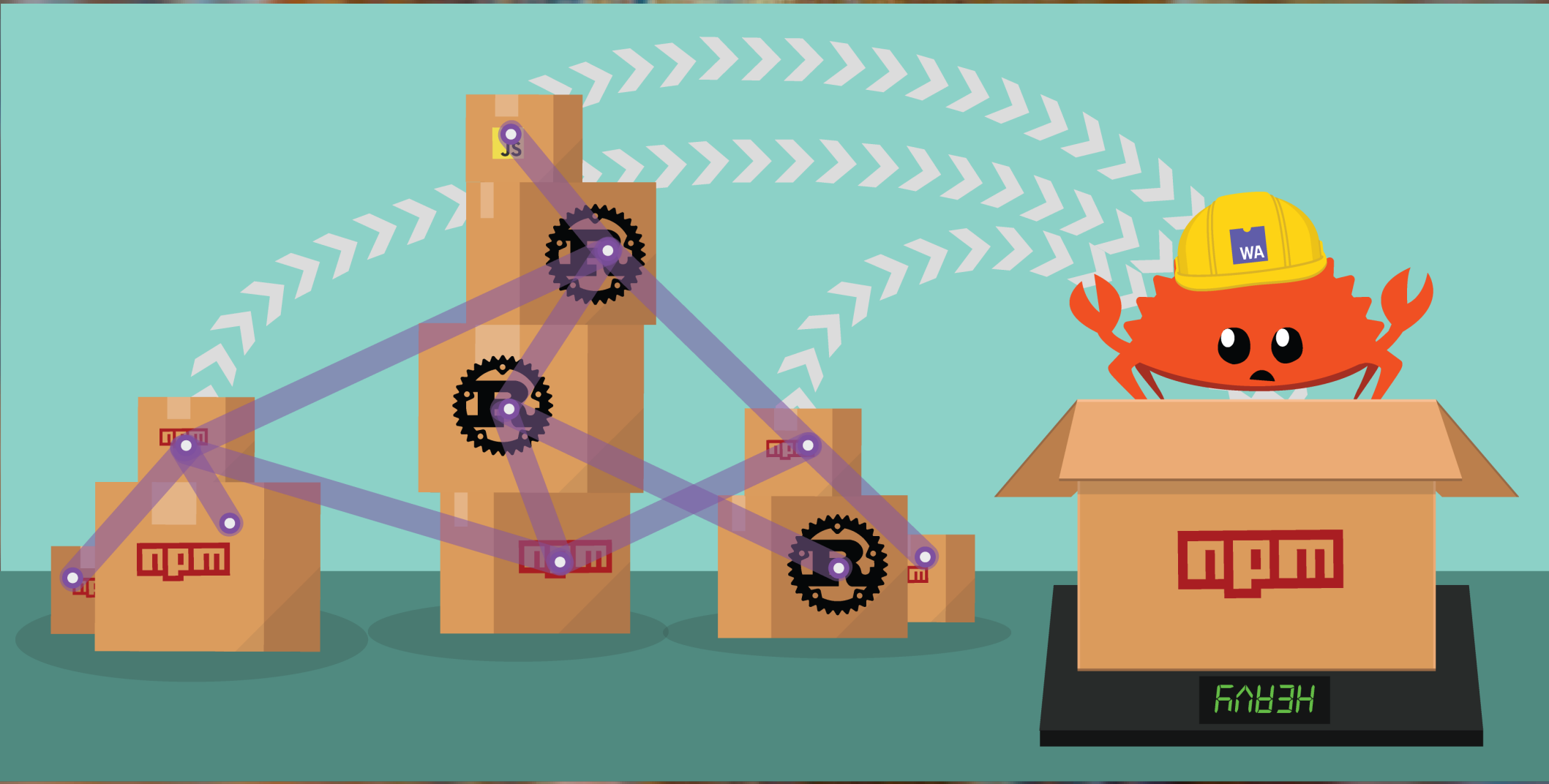
Output (10) | Problems (0)

```
[warn]: Changes in Project/src/lib.rs were ignored, save your changes.
[info]: Task build is completed
[info]: Task build is running...
[info]: Task build is completed
[info]: Task build is running...
[info]: Task build is completed
```

Como compilar C:

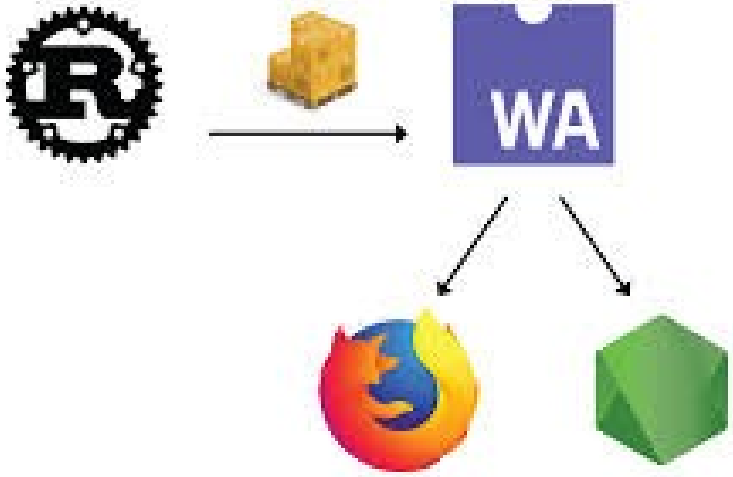
https://developer.mozilla.org/en-US/docs/WebAssembly/C_to_wasm

Como gero um projeto?


JS
npm
WA
npm
HEAVY


WA


WA


WA

Setup

Instalar Rust e Node

Todo o processo e mais pode ser visto em:

https://rustwasm.github.io/book/

Instalar o wasm-pack

https://rustwasm.github.io/wasm-pack/installer/

Install `wasm-pack`

```
curl https://rustwasm.github.io/wasm-pack/installer/init.sh -sSf | sh
```

Instalar o cargo-generate

cargo install cargo-generate

Tilix: Padrão 13 de dez 23:42

1 / 1 Tilix: Padrão

1: martinez@localhost:~/git

```
[martinez@localhost git]$ cargo generate --git https://github.com/rustwasm/wasm-pack-template
 Project Name: wasm-thelinux
 Creating project called `wasm-thelinux`...
 Done! New project created /home/martinez/git/wasm-thelinux
[martinez@localhost git]$
```

Tilix: Padrão 13 de dez 23:44

1 / 1 Tilix: Padrão

1: martinez@localhost:~/git/wasm-thelinux

```
[martinez@localhost git]$ cd wasm-thelinux/
[martinez@localhost wasm-thelinux]$ tree
.
├── Cargo.toml
├── LICENSE_APACHE
├── LICENSE_MIT
├── README.md
├── src
│   ├── lib.rs
│   └── utils.rs
└── tests
    └── web.rs

2 directories, 7 files
[martinez@localhost wasm-thelinux]$
```

Tilix: Padrão 13 de dez 23:45

1 / 1 Tilix: Padrão

1: martinez@localhost:~/git/wasm-thelinux

```rust
mod utils;

use wasm_bindgen::prelude::*;

// When the `wee_alloc` feature is enabled, use `wee_alloc` as the global
// allocator.
#[cfg(feature = "wee_alloc")]
#[global_allocator]
static ALLOC: wee_alloc::WeeAlloc = wee_alloc::WeeAlloc::INIT;

#[wasm_bindgen]
extern {
    fn alert(s: &str);
}

#[wasm_bindgen]
pub fn greet() {
    alert("Hello, wasm-thelinux!");
}
~
"src/lib.rs" 19L, 367C                                    19,1          Tudo
```

Tilix: Padrão 14 de dez 00:04

1 / 1 Tilix: Padrão

1: martinez@localhost:~/git/wasm-thelinux

```
[martinez@localhost wasm-thelinux]$ npm init wasm-app www
npx: instalou 1 em 9.108s
🦀 Rust + 🕸Wasm = ❤
[martinez@localhost wasm-thelinux]$
```

```rust
use wasm_bindgen::prelude::*;

#[wasm_bindgen]
extern {
    fn alert(s: &str);

    #[wasm_bindgen(js_namespace = console, js_name = log)]
    fn log(s: &str);
}

macro_rules! console_js {
    ($($t:tt)*) => (log(&format_args!($($t)*).to_string()))
}

#[wasm_bindgen]
pub fn greet() {
    alert("Hello, wasm-thelinux!");
}

#[wasm_bindgen]
pub fn logar(n: u32) {
    console_js!("Testando o {:?} {}", "console", n);
}

#[wasm_bindgen]
pub fn tnks(t: &str) {
    alert(&format!("{:?}, até!", t));
}
```

```
import * as wasm from "wasm-thelinux";

wasm.greet();
wasm.logar(1);
wasm.logar(2);
wasm.logar(3);
wasm.tnks("obrigado");
```

Tilix: Padrão 14 de dez 00:18

1 / 1 Tilix: Padrão

1: martinez@localhost:~/git/wasm-thelinux

```
[martinez@localhost wasm-thelinux]$ wasm-pack build
[INFO]: Checking for the Wasm target...
[INFO]: Compiling to Wasm...
   Compiling proc-macro2 v1.0.6
   Compiling unicode-xid v0.2.0
   Compiling syn v1.0.11
   Compiling log v0.4.8
   Compiling wasm-bindgen-shared v0.2.55
   Compiling cfg-if v0.1.10
   Compiling lazy_static v1.4.0
   Compiling bumpalo v2.6.0
   Compiling wasm-bindgen v0.2.55
   Compiling quote v1.0.2
   Compiling wasm-bindgen-backend v0.2.55
   Compiling wasm-bindgen-macro-support v0.2.55
   Compiling wasm-bindgen-macro v0.2.55
   Compiling console_error_panic_hook v0.1.6
   Compiling wasm-thelinux v0.1.0 (/home/martinez/git/wasm-thelinux)
    Finished release [optimized] target(s) in 1m 24s
[INFO]: Installing wasm-bindgen...
[INFO]: Optional fields missing from Cargo.toml: 'description', 'repository', and 'license'. These are not necess
ary, but recommended
[INFO]: :-) Done in 1m 24s
[INFO]: :-) Your wasm pkg is ready to publish at ./pkg.
[martinez@localhost wasm-thelinux]$
```

Tilix: Padrão 14 de dez 00:43

1 / 1 Tilix: Padrão

1: martinez@localhost:~/git/wasm-thelinux/www

```
[martinez@localhost wasm-thelinux]$ cd www/
[martinez@localhost www]$ npm i --save ../pkg/
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.9 (node_modules/fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.9: wanted {"os":"darwin","arc
h":"any"} (current: {"os":"linux","arch":"x64"})

+ wasm-thelinux@0.1.0
added 591 packages from 361 contributors and audited 9103 packages in 19.194s
found 1 moderate severity vulnerability
  run `npm audit fix` to fix them, or `npm audit` for details
[martinez@localhost www]$
```

Tilix: Padrão 14 de dez 00:50

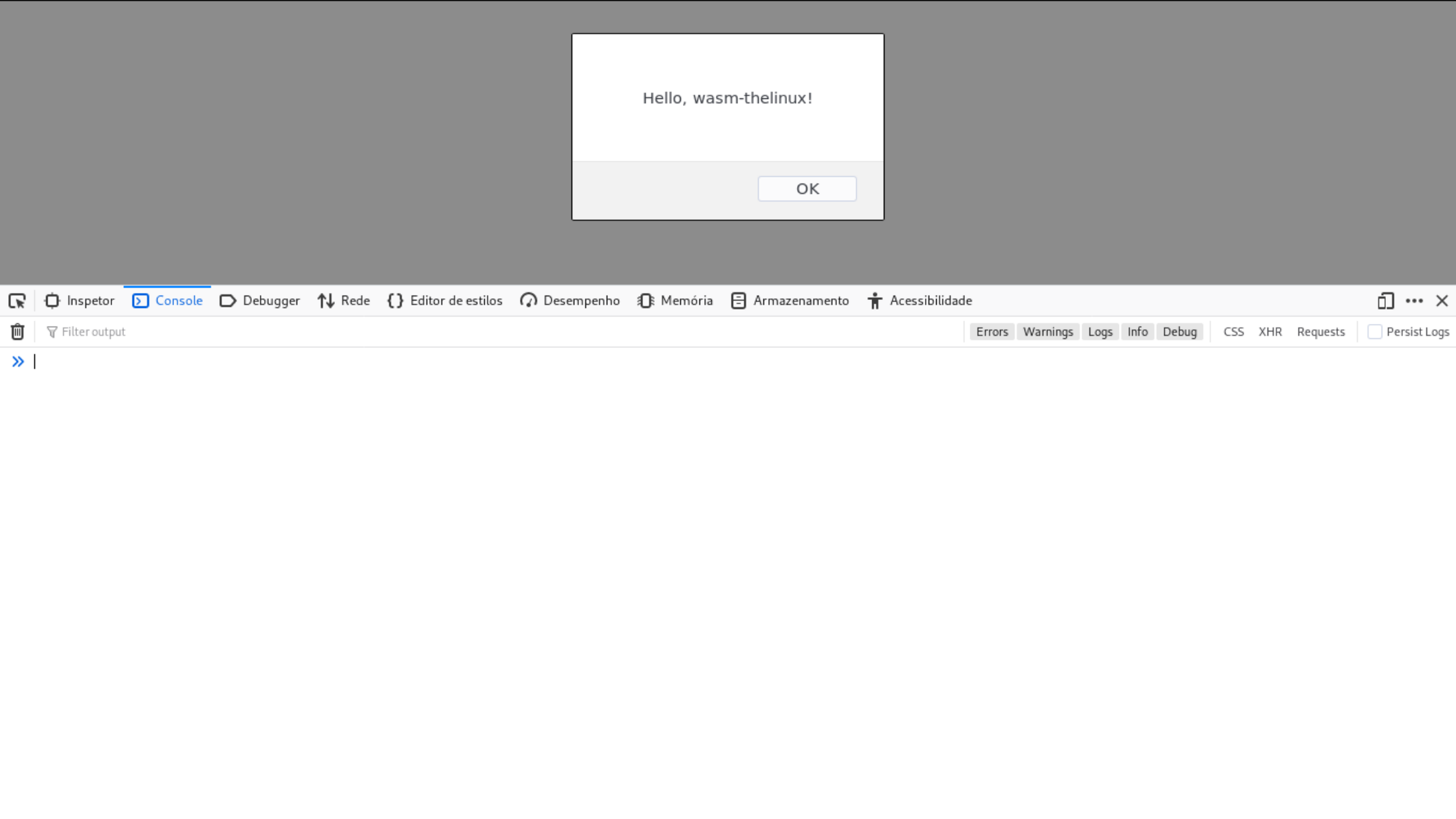
1/1 Tilix: Padrão

1: martinez@localhost:~/git/wasm-thelinux/www

```
[martinez@localhost www]$ npm start

> create-wasm-app@0.1.0 start /home/martinez/git/wasm-thelinux/www
> webpack-dev-server

i ｢wds｣: Project is running at http://localhost:8080/
i ｢wds｣: webpack output is served from /
i ｢wds｣: Content not from webpack is served from /home/martinez/git/wasm-thelinux/www
i ｢wdm｣: Hash: 543b8136ed82002d40ca
Version: webpack 4.41.2
Time: 1608ms
Built at: 2019-12-14 1:45:23
                         Asset       Size  Chunks                         Chunk Names
                 0.bootstrap.js   4.44 KiB       0  [emitted]
197ad94b13e834d2568c.module.wasm   31.9 KiB       0  [emitted] [immutable]
                    bootstrap.js    367 KiB    main  [emitted]              main
                      index.html  297 bytes          [emitted]
Entrypoint main = bootstrap.js
[0] multi (webpack)-dev-server/client?http://localhost:8080 ./bootstrap.js 40 bytes {main} [built]
[../pkg/wasm_thelinux.js] 1.06 KiB {0} [built]
[./bootstrap.js] 279 bytes {main} [built]
[./index.js] 176 bytes {0} [built]
[./node_modules/ansi-html/index.js] 4.16 KiB {main} [built]
[./node_modules/ansi-regex/index.js] 135 bytes {main} [built]
[./node_modules/strip-ansi/index.js] 161 bytes {main} [built]
[./node_modules/webpack-dev-server/client/index.js?http://localhost:8080] (webpack)-dev-server/client?http://localhost:8080 4.29 KiB {main} [built]
[./node_modules/webpack-dev-server/client/overlay.js] (webpack)-dev-server/client/overlay.js 3.51 KiB {main} [built]
[./node_modules/webpack-dev-server/client/socket.js] (webpack)-dev-server/client/socket.js 1.53 KiB {main} [built]
[./node_modules/webpack-dev-server/client/utils/createSocketUrl.js] (webpack)-dev-server/client/utils/createSocketUrl.js 2.89 KiB {main} [built]
[./node_modules/webpack-dev-server/client/utils/log.js] (webpack)-dev-server/client/utils/log.js 964 bytes {main} [built]
[./node_modules/webpack-dev-server/client/utils/reloadApp.js] (webpack)-dev-server/client/utils/reloadApp.js 1.59 KiB {main} [built]
[./node_modules/webpack-dev-server/client/utils/sendMessage.js] (webpack)-dev-server/client/utils/sendMessage.js 402 bytes {main} [built]
[./node_modules/webpack/hot sync ^\.\/log$] (webpack)/hot sync nonrecursive ^\.\/log$ 170 bytes {main} [built]
    + 21 hidden modules
```


Hello, wasm-thelinux!
OK
Inspetor
Console
Debugger
Rede
Editor de estilos
Desempenho
Memória
Armazenamento
Acessibilidade
Filter output
Errors
Warnings
Logs
Info
Debug
CSS
XHR
Requests
Persist Logs

"obrigado", até!

Bloquear janelas de confirmação desta página?

OK

Inspetor | Console | Debugger | Rede | Editor de estilos | Desempenho | Memória | Armazenamento | Acessibilidade

Filter output | Errors | Warnings | Logs | Info | Debug | CSS | XHR | Requests | Persist Logs

```
Testando o "console" 1                wasm_thelinux.js:97:13
Testando o "console" 2                wasm_thelinux.js:97:13
Testando o "console" 3                wasm_thelinux.js:97:13
```

Publicar npm

Tilix: Padrão 14 de dez 01:15

1 / 1 Tilix: Padrão

1: martinez@localhost:~/git/wasm-thelinux

```
[martinez@localhost wasm-thelinux]$ wasm-pack login
Username: michelsmartinez
Password:
Email: (this IS public) michelsm1990@gmail.com
Logged in as michelsmartinez on https://registry.npmjs.org/.
[INFO]: 👋  logged you in!
[martinez@localhost wasm-thelinux]$
```

Tilix: Padrão 14 de dez 01:17

1 / 1 Tilix: Padrão

1: martinez@localhost:~/git/wasm-thelinux

```
[martinez@localhost wasm-thelinux]$ wasm-pack publish
npm notice
npm notice 📦  wasm-thelinux@0.1.0
npm notice === Tarball Contents ===
npm notice 2.1kB  wasm_thelinux.js
npm notice 311B   package.json
npm notice 2.2kB  README.md
npm notice 194B   wasm_thelinux.d.ts
npm notice 35.4kB wasm_thelinux_bg.wasm
npm notice === Tarball Details ===
npm notice name:          wasm-thelinux
npm notice version:       0.1.0
npm notice package size:  17.7 kB
npm notice unpacked size: 40.1 kB
npm notice shasum:        06145481bb3a489651e9eee185e40cb792757783
npm notice integrity:     sha512-DXwWigQPR2+T4[...]42ZDl5sbwCvjQ==
npm notice total files:   5
npm notice
+ wasm-thelinux@0.1.0
[INFO]: 💥  published your package!
[martinez@localhost wasm-thelinux]$
```

wasm-thelinux

0.1.0 • Public • Published 3 minutes ago

Readme | Explore BETA | 0 Dependencies | 0 Dependents | 1 Versions

wasm-pack-template

A template for kick starting a Rust and WebAssembly project using wasm-pack.

build error

Tutorial | Chat

```
<a href="https://rustwasm.github.io/docs/wasm-pack/tutorials/npm-browse
<span> | </span>
<a href="https://discordapp.com/channels/442252698964721669/443151097398
```

Built with by The Rust and WebAssembly Working Group

About

Read this template tutorial!

This template is designed for compiling Rust libraries into WebAssembly and publishing the

Install

```
> npm i wasm-thelinux
```

Version	License
0.1.0	none

Unpacked Size	Total Files
40.1 kB	5

Last publish

3 minutes ago

Collaborators

Try on RunKit

Successfully published wasm-thelinux@0.1.0

Caixa de entrada ×

npm Inc notifications@npmjs.com por amazonses.com

para eu

Hi michelsmartinez!

A new version of the package wasm-thelinux (0.1.0) was published at 2019-12-14T04:16:59.060Z from 177.10.10.77. The shasum of this package was 06145481bb3a489651e9eee185e40cb792757783.

If you have questions or security concerns, you can reply to this message or email support@npmjs.com.

npm loves you.

Links uteis / referência:

https://webassembly.org/

https://www.rust-lang.org/pt-BR/what/wasm

https://medium.com/trainingcenter/webassembly-a-jornada-o-que-%C3%A9-wasm-75e3f0f03124

https://webassembly.studio/

https://rustwasm.github.io/book/introduction.html

https://github.com/rustwasm/wasm-pack

https://rustwasm.github.io/wasm-bindgen/introduction.html

https://rustwasm.github.io/wasm-pack/installer/

https://rustwasm.github.io/book/introduction.html

https://github.com/rustwasm/awesome-rust-and-webassembly

@rustlangbr

@rustinpoa

@MichelMartinez

Dúvidas ou comentários?

Obrigado!