

# Package: trama.launcher (via r-universe)

October 21, 2026

**Title** Instalador e Launcher do trama

**Version** 0.2.4

**Description** Motor de instalação e atualização do trama: lê o manifesto de release, calcula o status da instalação local, instala releases numa biblioteca isolada (só pacotes binários) e serve a tela de início em Shiny, de onde o editor é aberto.

**License** MIT + file LICENSE

**URL** <https://github.com/uaipetro/trama>

**BugReports** <https://github.com/uaipetro/trama/issues>

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**Imports** jsonlite, later, shiny, stats, utils, tools

**Suggests** testthat (>= 3.0.0), withr

**Config/testthat/edition** 3

**Config/roxygen2/version** 8.0.0

**Config/pak/sysreqs** cmake make libuv1-dev zlib1g-dev

**Repository** <https://uaipetro.r-universe.dev>

**Date/Publication** 2026-10-20 21:24:06 UTC

**RemoteUrl** <https://github.com/uaipetro/trama>

**RemoteRef** HEAD

**RemoteSha** a0a356f56db4d8598f58098a83208b81b51ce5ae

**RemoteSubdir** tools/installer/trama.launcher

## Contents

|                                |   |
|--------------------------------|---|
| abrir . . . . .                | 2 |
| tl_collection_add . . . . .    | 3 |
| tl_collection_remove . . . . . | 3 |
| tl_install_release . . . . .   | 4 |

|                            |          |
|----------------------------|----------|
| tl_manifest_read . . . . . | 4        |
| tl_rollback . . . . .      | 5        |
| tl_status . . . . .        | 5        |
| <b>Index</b>               | <b>6</b> |

---

|       |                                 |
|-------|---------------------------------|
| abrir | <i>Abre o launcher do trama</i> |
|-------|---------------------------------|

---

**Description**

Ponto de entrada chamado pelo atalho instalado (Trama-Setup.exe no Windows, install.sh no Linux): coloca a biblioteca da release atual na frente do .libPaths() (se já houver uma) e sobe a tela de início do launcher. Se ainda não houver nenhuma release instalada, é a tela de início quem dispara a primeira instalação (tl\_server(), em R/app.R) — *depois* dela estar no ar, não antes: instalar antes do runApp() rodava tl\_manifest\_read() sem UI nenhuma para mostrar o progresso ou um erro (e lançava erro de verdade se estivesse offline, derrubando abrir() inteiro antes do pesquisador ver qualquer coisa).

**Usage**

```
abrir(  
  porta = 8725L,  
  ja_rodando = .tl_launcher_rodando,  
  abrir_janela = tl_open_window  
)
```

**Arguments**

- porta            Porta onde a tela de início sobe; se ocupada, usa a primeira livre a partir daqui.
- ja\_rodando      Função que confere se já há um launcher respondendo em porta (padrão .tl\_launcher\_rodando()) — parâmetro só para os testes poderem mockar sem abrir um socket de verdade.
- abrir\_janela    Como abrir a janela quando o launcher já está de pé. Por padrão tl\_open\_window().

**Value**

Não retorna em uso normal — shiny::runApp() bloqueia a sessão enquanto a tela de início está aberta (a menos que já houvesse um launcher de pé, caso em que só abre a janela e retorna).

---

|                   |                                             |
|-------------------|---------------------------------------------|
| tl_collection_add | <i>Instala uma coleção na release atual</i> |
|-------------------|---------------------------------------------|

---

**Description**

Instala uma coleção na lib da release atual e a acrescenta ao estado. Recusa qualquer coleção que não conste no manifesto — o manifesto é a única fonte de verdade sobre o que é instalável nesta release.

**Usage**

```
tl_collection_add(m, nome, progresso = message)
```

**Arguments**

|           |                                               |
|-----------|-----------------------------------------------|
| m         | Manifesto de release.                         |
| nome      | Nome da coleção a instalar.                   |
| progresso | Função chamada com uma mensagem de progresso. |

**Value**

(Invisível) o novo estado gravado.

---

|                      |                                            |
|----------------------|--------------------------------------------|
| tl_collection_remove | <i>Remove uma coleção da release atual</i> |
|----------------------|--------------------------------------------|

---

**Description**

Remove uma coleção da lib da release atual e a tira do estado.

**Usage**

```
tl_collection_remove(nome, progresso = message)
```

**Arguments**

|           |                                               |
|-----------|-----------------------------------------------|
| nome      | Nome da coleção a remover.                    |
| progresso | Função chamada com uma mensagem de progresso. |

**Value**

(Invisível) o novo estado gravado.

---

|                    |                                                    |
|--------------------|----------------------------------------------------|
| tl_install_release | <i>Instala uma release numa biblioteca isolada</i> |
|--------------------|----------------------------------------------------|

---

### Description

Instala a release descrita em `m` (mais as colecoes pedidas) numa biblioteca isolada `tl_lib_dir(m$trama)`, só com pacotes binários.

### Usage

```
tl_install_release(m, colecoes = tl_state_read()$colecoes, progresso = message)
```

### Arguments

|                        |                                                                                     |
|------------------------|-------------------------------------------------------------------------------------|
| <code>m</code>         | Manifesto de release ( <code>tl_manifest_read()/tl_manifest_fetch()</code> ).       |
| <code>colecoes</code>  | Nomes de coleções a instalar junto com o núcleo.                                    |
| <code>progresso</code> | Função chamada com uma mensagem a cada pacote ( <code>message()</code> por padrão). |

### Details

Só grava o estado se TODOS os pacotes pedidos existirem na lib ao final; se algum faltar, apaga a lib parcial e lança erro sem tocar no estado — quem já tinha uma release funcionando continua com ela.

### Value

(Invisível) o novo estado gravado.

---

|                  |                                           |
|------------------|-------------------------------------------|
| tl_manifest_read | <i>Lê e valida o manifesto de release</i> |
|------------------|-------------------------------------------|

---

### Description

Lê e valida o manifesto de `src` (URL ou caminho local). Lança erro com classe `tl_error_manifesto` e mensagem em português se faltar alguma chave obrigatória, se `cran_snapshot` não for uma data ISO (AAAA-MM-DD) ou se `r` não estiver no formato `x.y.z`.

### Usage

```
tl_manifest_read(src = tl_manifest_url())
```

### Arguments

|                  |                                                                  |
|------------------|------------------------------------------------------------------|
| <code>src</code> | URL ou caminho local do manifesto ( <code>release.json</code> ). |
|------------------|------------------------------------------------------------------|

### Value

Lista com o manifesto validado.

---

|             |                                      |
|-------------|--------------------------------------|
| tl_rollback | <i>Volta para a release anterior</i> |
|-------------|--------------------------------------|

---

**Description**

Volta a release atual para a anterior mais recente (a lib da release atual continua no disco, só não é mais a apontada pelo estado — assim dá para avançar de novo sem reinstalar).

**Usage**

tl\_rollback()

**Value**

(Invisível) o novo estado gravado.

---

|           |                                   |
|-----------|-----------------------------------|
| tl_status | <i>Status da instalação local</i> |
|-----------|-----------------------------------|

---

**Description**

O que a tela de início lê para desenhar versões, botão de atualizar, coleções e o aviso de troca de linha do R. Nunca lança erro — mesmo sem instalação e sem rede, devolve uma resposta coerente (a página tem que conseguir se desenhar nesse estado).

**Usage**

tl\_status(m = tl\_manifest\_fetch(), s = tl\_state\_read())

**Arguments**

- m Manifesto (de tl\_manifest\_fetch()) ou NULL se offline.
- s Estado local (de tl\_state\_read()).

**Value**

Lista com release\_instalada, release\_disponivel, atualizar, r\_instalado, r\_exigido, troca\_de\_r, pacotes (data.frame nome/instalada/disponível) e colecoes (data.frame nome/titulo/instalada/disponivel/requi

# Index

abrir, [2](#)

tl\_collection\_add, [3](#)

tl\_collection\_remove, [3](#)

tl\_install\_release, [4](#)

tl\_manifest\_read, [4](#)

tl\_rollback, [5](#)

tl\_status, [5](#)