El gestor de paquetes Alire para Ada y SPARK

https://alire.ada.dev/

Alejandro R. Mosteo

27 de abril de 2022

CUD

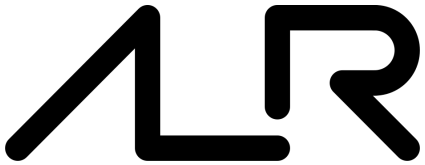
Centro Universitario de la Defensa Zaragoza

CONTENIDOS

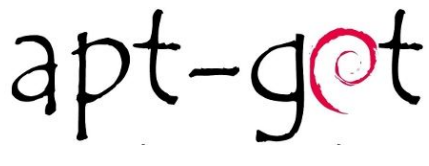
- Introducción
- Casos de uso
- Ecosistema

GESTORES DE PAQUETES

Ada world



AURA



LÍNEA TEMPORAL

- ➔ **Feb 2016: primer repositorio & debates**
- ➔ **Feb 2018: artículo en Ada-Europe 2018**
- ➔ Apr 2019: interés por parte de AdaCore
- ➔ **Aug 2019: Sitio web publicado**
- ➔ Nov 2019: beta interna
- ➔ **Feb 2021: versión 1.0**
- ➔ Sep 2021: 1.1
- ➔ May 2022: 1.2

TIPOS DE GESTORES

SISTEMA vs **USUARIO**

PLATAFORMA vs **LENGUAJE**

BINARIOS vs **FUENTES**

OFICIAL vs **COMUNIDAD**

PERSPECTIVA

`https://github.com/alire-project`

Alire

- Proyecto en su conjunto
- Índice comunitario
  - Paquetes disponibles
  - Paquetes = "*crates*"

`alr`

- Línea de órdenes
  - Solucionador de dependencias
  - Descarga de paquetes
  - Compilación

Ada **Li**brary **Re**pository

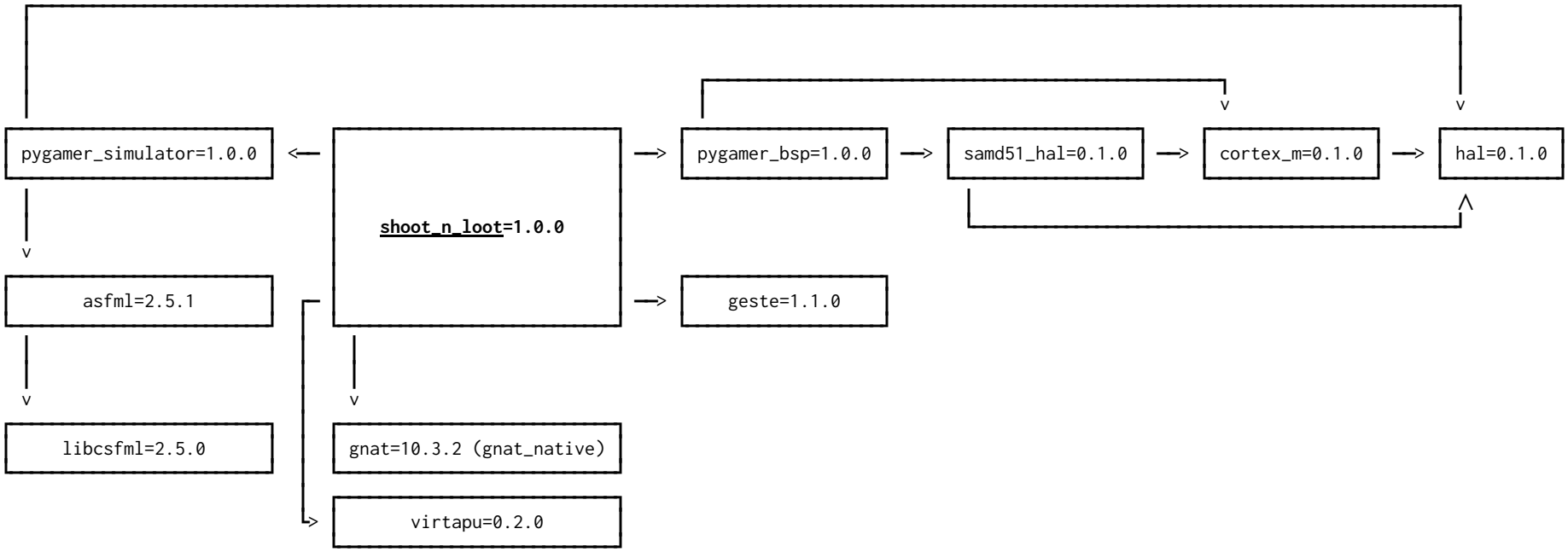
CASO DE USO: OBTENER UN PROGRAMA

```
$ alr get hangman

$ ls
hangman_1.0.0_a5790492

$ cd hangman_1.0.0_a5790492
$ alr run
***** W E L C O M E  T O  H A N G M A N  *****
 By: Jon Hollan, Mark Hoffman, & Brandon Ball

$ alr run --list
Project hangman builds these executables:
   hangmain (found at ./bin/hangmain)
```

CASO DE USO: COMENZAR UN PROYECTO

```
$ alr init --bin mi_proyecto
$ cd mi_proyecto

$ alr build
$ alr run
$ alr edit
```

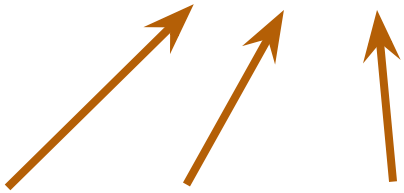
CASO DE USO: REUSO DE LIBRERÍAS / DEPENDENCIAS



- Importación de dependencias
  - Simplemente reusar código
    - Encontrando una combinación válida
- Actualizaciones

SEMANTIC VERSIONING

version 1.2.3-prerelease+anything



- **mayor . menor . parche**
  - Cambios mayores que rompen compatibilidad
  - Cambios menores que añaden funcionalidad
  - Parches que arreglan bugs sin cambiar la API
- Actualizaciones menores/parches son “seguras”

- Operadores especiales:
  - ^1.0 (>=1.0 & <2.0)
  - ~2.3 (>=2.3 & <2.4)

CASO DE USO: DEPENDENCIAS, BÚSQUEDA

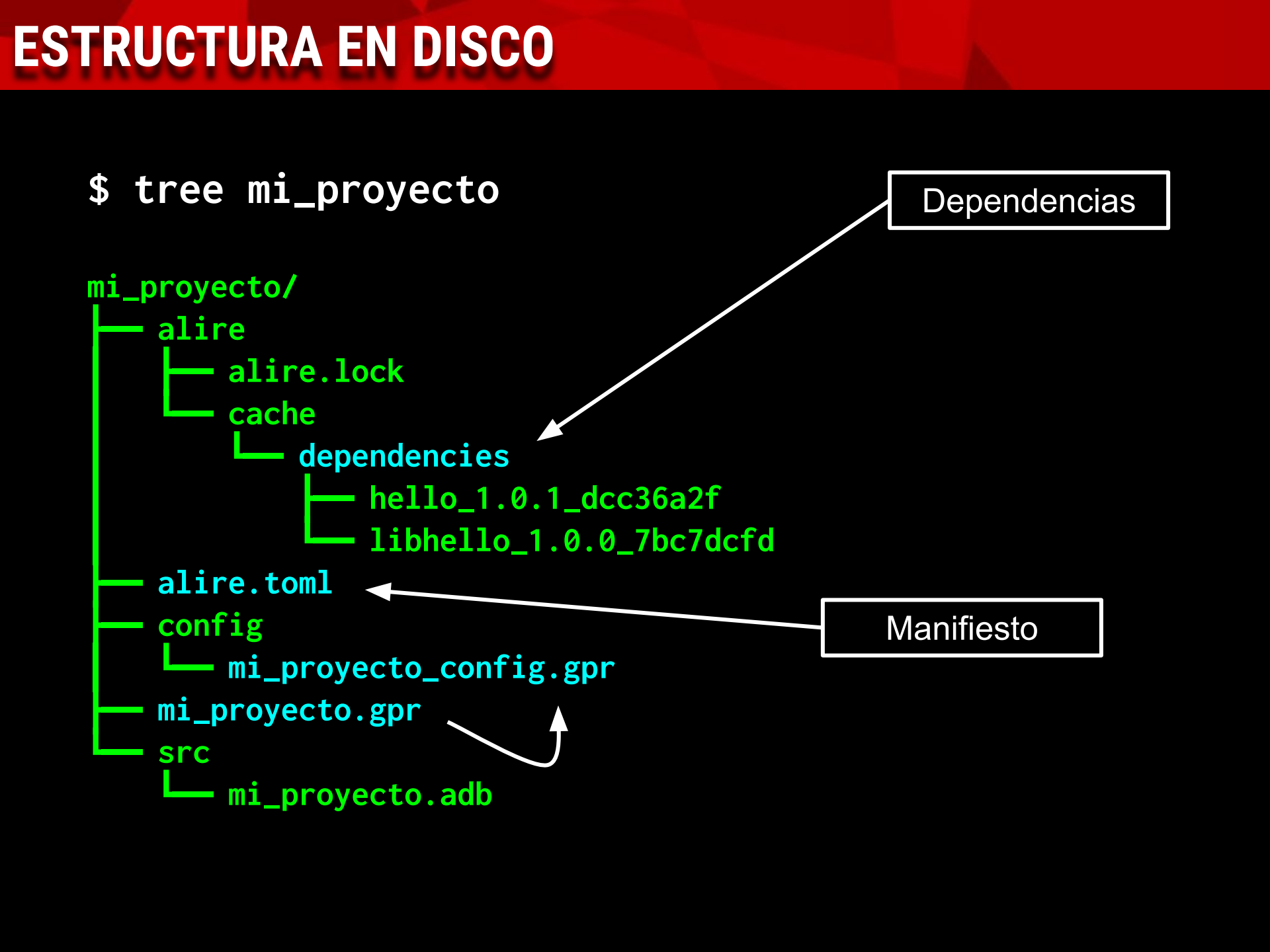
```
$ alr search --crates
ada_lua           An Ada binding for Lua
adacurses         Wrapper on different packagings of NcursesAda
adayaml           Experimental YAML 1.3 implementation in Ada
adayaml_server    Experimental YAML 1.3 server component
agpl              Ada General Purpose Library with a robotics flavor
ajunitgen         Generator of JUnit-compatible XML reports
alire             Alire project catalog and support files
alr               Command-line tool from the Alire project
apq               APQ Ada95 Database Library (core)
aunit             Ada unit test framework

$ alr search toml
NAME          STATUS  VERSION  DESCRIPTION
ada_toml              0.3.0    TOML parser for Ada
toml_slicer           0.1.0    Edit TOML files directly without parsing
```

CASO DE USO: DEPENDENCIAS, USO

```
$ alr with hello
Changes to dependency solution:
   + hello    1.0.1 (new)
   + libhello 1.0.0 (new,indirect)

$ alr with --tree
mi_proyecto=0.1.0-dev
└── hello=1.0.1 (^1.0.1)
    └── libhello=1.0.0 (^1.0)

$ alr with --graph
```

ESTRUCTURA EN DISCO

```
$ tree mi_proyecto

mi_proyecto/
├── alire
│   ├── alire.lock
│   └── cache
│       └── dependencies
│           ├── hello_1.0.1_dcc36a2f
│           └── libhello_1.0.0_7bc7dcfd
├── alire.toml
├── config
│   └── mi_proyecto_config.gpr
├── mi_proyecto.gpr
└── src
    └── mi_proyecto.adb
```



MANIFIESTO alire.toml

```
$ alr init --bin mi_proyecto && cd mi_proyecto && cat alire.toml

name = "mi_proyecto"
description = "Shiny new project"
version = "0.1.0-dev"

authors = ["Alejandro R. Mosteo"]
maintainers = ["Alejandro R. Mosteo <alejandro@mosteo.com>"]
maintainers-logins = ["mosteo"]

project-files = [“mi_proyecto.gpr”]
executables = ["mi_proyecto"]

[[depends-on]]
hello = "^1.0.1"
```

CASO DE USO: PUBLICACIÓN DE PROYECTOS

Asistente de publicación: `alr publish`

- Validación local
  - Metadatos
  - Compilación
- Generación de manifiesto
  - Abrir un *pull-request* sobre el índice comunitario
  - Validación remota basada en *GitHub Actions*
- `alr index --update-all`
  - Se pueden usar índices privados

EJEMPLO DE PUBLICACIÓN

```
$ alr publish
  ✓ Local repository is clean
  ✓ Revision exists in local repository (4550aa3)
  Publishing assistant: step 1 of 6: Verify origin URL
  ✓ Origin is of supported kind: GIT
  ✓ Origin is hosted on trusted site: github.com
  Publishing assistant: step 2 of 6: Verify GitHub infrastructure
  ✓ User has a GitHub account: mosteo
  ✓ User has forked the community repository
  ✓ User's fork contains base branch: stable-1.1
  Publishing assistant: step 3 of 6: Deploy sources
  Publishing assistant: step 4 of 6: Build release
  ✓ Sources built successfull
  Publishing assistant: step 5 of 6: User review
       minirest=0.2.0-dev: Minimalist Ada REST client library
       Origin: commit 4550aa3 from https://github.com/mosteo/minirest.git
       Properties:
          Description: Minimalist Ada REST client library
          License: MIT
          Maintainer: alejandro@mosteo.com
          Name: minirest
          Version: 0.2-dev
  Publishing assistant: step 6 of 6: Generate index manifest
  ✓ Manifest generated at ./alire/releases/minirest-0.2.0-dev.toml
  ⓘ Please upload this file to
https://github.com/mosteo/alire-index/upload/stable-1.1/index/mi/minirest
```

OBTENCIÓN DE HERRAMIENTAS GNAT (TOOLCHAINS)

- Alire indexa GNAT FSF para:
  - Windows
  - Linux
  - macOS
- Incluyendo compiladores cruzados para:
  - ARM Cortex-M
  - AVR
  - RISC-V
- Precompilados mediante *scripts* públicos ejecutados en GitHub:
  - https://github.com/alire-project/GNAT-FSF-builds

CASO DE USO: INSTALACIÓN DE HERRAMIENTAS

```
$ alr toolchain --select gnat gprbuild
$ alr toolchain --select
Please select the gnat version for use with this configuration
  1. gnat_native=11.2.4
  2. None
  3. gnat_external=9.4.0 [Detected at /usr/bin/gnat]
  4. gnat_arm_elf=11.2.4
  5. gnat_avr_elf=11.2.4
  6. gnat_riscv64_elf=11.2.4
  a. (See more choices...)
$ alr toolchain
CRATE         VERSION  STATUS    NOTES
gprbuild      22.0.1   Default
gprbuild      2019.0.0 Available Detected at /usr/bin/gprbuild
gnat_arm_elf  11.2.3   Available
gnat_native   10.3.2   Available
gnat_native   11.2.4   Default
gnat_external 9.4.0    Available Detected at /usr/bin/gnat
```

CASO DE USO: CONFIGURACIONES ESTÁTICAS

- Ni Ada ni GPRbuild tienen un pre-procesador
  - Complica compilaciones no triviales
- Alire tiene un paso previo a la compilación
  - En particular para generar ficheros con vistas a compilaciones estáticas
- Sección de configuraciones del manifiesto

CONFIGURACIONES

```
# En el proyecto “mi_proyecto”
[configuration.variables]
Device_Name = {type = "String",  default = "no device name"}
Print_Debug = {type = "Boolean", default = false}
Debug_Level = {type = "Enum", values = ["Debug", "Warn", "Error"], default = "Warn"}
Buffer_Size = {type = "Integer", first = 0, last = 1024, default = 256}
Max_Power   = {type = "Real", first = 0.0, last = 100.0, default = 50.0}
```

```
# En otro proyecto que depende de “mi_proyecto”
[configuration.values]
mi_proyecto.Device_Name = "Custom device name"
mi_proyecto.Print_Debug = true
mi_proyecto.Debug_Level = "Error"
mi_proyecto.Buffer_Size = 42
mi_proyecto.Max_Power   = 42.0
```

```
-- Generado por Alire (*.ads, *.h, *.gpr)
package my_crate_Config is
   Buffer_Size_First : constant := 0; Buffer_Size_Last : constant := 1024;
   Buffer_Size : constant := 42;

   type Debug_Level_Kind is (Debug, Warn, Error);
   Debug_Level : constant Debug_Level_Kind := Error;

   Print_Debug : constant Boolean := True;

   Max_Power_First : constant := 0.0; Max_Power_Last : constant := 100.0;
   Max_Power : constant := 42.0;

   Device_Name : constant String := "Custom device name";
end my_crate_Config;
```

CASO DE USO: desarrollos en paralelo

```
[[pins]]
foo = { version = "1.3.2+bugfix" }
# Uso de una versión específica

bar = { path = "../my/bar" }
# Uso de una carpeta local para proveer una dependencia

baz = { url = "https://github.com/baz.git" }
# Usar la rama predeterminada, actualizable con `alr update`

wip = { url = "https://gitrepo.com/wip.git" branch="feature" }
# Usar una rama concreta, actualizable con `alr update`

gru = { url = "https://gitlab.com/gru.git" commit="..." }
# Usar un commit remoto, nunca se actualizará
```

Los “pins” satisfacen de forma universal su dependencia correspondiente

CASOS DE USO: TESTS, DEMOSTRACIONES

Para enviar al índice comunitario

```
mi_proyecto
├── alire.toml
│
├── examples
│   └── alire.toml
│
├── src
│
└── tests
    └── alire.toml
```

```
executables = [“demo1”, “demo2”]
# Binarios de demostración que no se
# quiere compilar normalmente

[[pins]]
my_crate = { path=”..” }
```

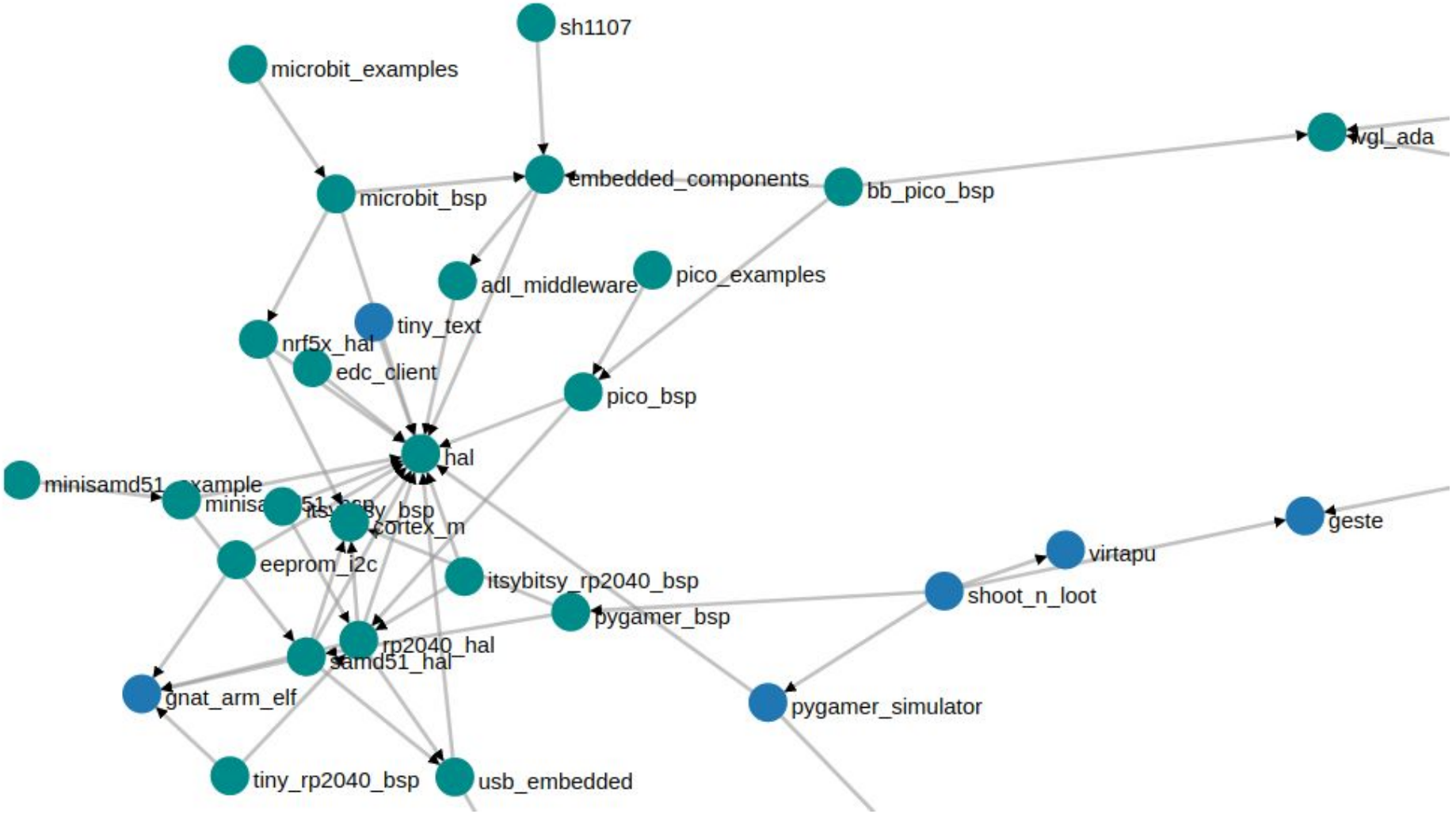
```
[[pins]]
my_crate = { path=”..” }

[[depends-on]]
# Dependencias extra, solo para testeo
aunit = “*”
```

ECOSISTEMA

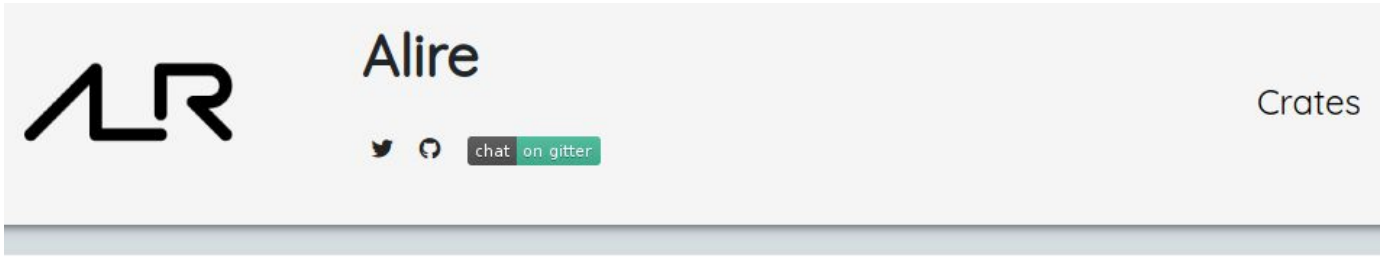
- 243 Crates
  - 35 Embedded
  - 14 SPARK
- Etiquetas *Top* 10
  - embedded(32) nostd(26) gnatcoll(14) web(12) spark(10) database(10) rp2040(9) bindings(9) sql(8) game(8)
- 22500+ descargas de “`alr`”
  - https://hanadigital.github.io/grev/?user=alire-project&repo=alire

https://alire.ada.dev/network

EMBEDDED


sh1107
microbit_examples
wgl_ada
embedded_components
microbit_bsp
bb_pico_bsp
adl_middleware
pico_examples
tiny_text
nrf5x_hal
edc_client
pico_bsp
hal
minisamd51_example
cortex_m
geste
virtapu
eeprom_i2c
itsybitsy_rp2040_bsp
shoot_n_loot
pygamer_bsp
rp2040_hal
samd51_hal
gnat_arm_elf
pygamer_simulator
tiny_rp2040_bsp
usb_embedded

WEBSITE



https://alire.ada.dev

Install Alire

Getting Started

ALIRE: Ada LIbrary REpository

A catalog of ready-to-use Ada/SPARK libraries plus a command-line tool (`alr`) to obtain, build, and incorporate them into your own projects. It aims to fulfill a similar role to Rust's `cargo` or OCaml's `opam` .

Design principles

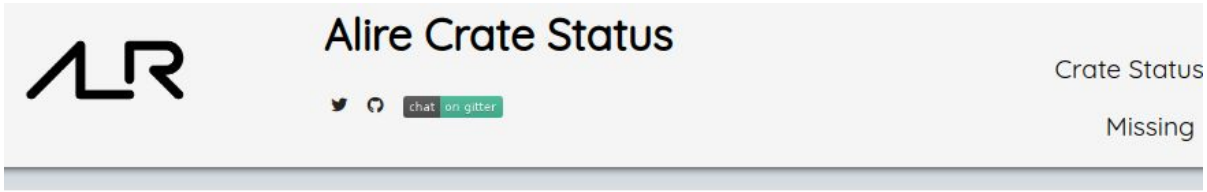
`alr` is tailored to userspace, in a similar way to Python's virtualenv. A project or workspace will contain all its dependencies.

Some projects require binary packages from the distribution (Debian/Ubuntu's apt, msys2's pacman on Windows). In this case the user will be asked to authorize an installation through the distribution package manager.

Properties and dependencies of projects are managed through a TOML file. This file exists locally for working copies of projects, and the Alire community index stores the files corresponding to its projects.

The complete build environment is set up by setting the `GPR_PROJECT_PATH` environment variable before running `gprbuild` , thus freeing the user from concerns about installation paths. The user simply adds the used projects to its own project GPR file with their simple name.

ESTADO DE LOS PAQUETES



https://alire-crates-ci.mosteo.com

aaa

CI aaa 14/14

arch | gnat 11.1.0 success | centos 8 | gnat 2020.0.0 success | debian 11 | gnat 10.2.1 success | fedora 33 | gnat 10.2.1 success

ubuntu 20.04 | gnat 10.2.1 success | ubuntu 20.04 | gnat 11.1.0 success | ubuntu 20.04 | gnat 2020.0.0 success | ubuntu 20.04 | gnat 9.3.0 success

v0.2.4 compiled with GNAT 11.1.0 on linux arch
- Status: SUCCESS
- Duration: 7.88
- Attempted: 2022-04-25 01:12:40 +0000
- Log

v0.2.4 compiled with GNAT 2020.0.0 on linux centos-8
- Status: SUCCESS
- Duration: 9.83
- Attempted: 2022-04-24 10:52:35 +0000
- Log

v0.2.4 compiled with GNAT 10.2.1 on linux debian-11
- Status: SUCCESS
- Duration: 15.94
- Attempted: 2022-04-24 11:07:46 +0000
- Log

v0.2.4 compiled with GNAT 10.2.1 on linux fedora-33
- Status: SUCCESS
- Duration: 8.91
- Attempted: 2022-04-24 22:04:11 +0000
- Log

v0.2.4 compiled with GNAT 10.2.1 on linux ubuntu-20.04
- Status: SUCCESS
- Duration: 8.74
- Attempted: 2022-04-24 21:59:06 +0000
- Log

v0.2.4 compiled with GNAT 10.3.2 on linux ubuntu-20.04
- Status: SUCCESS
- Duration: 6.58
- Attempted: 2022-04-23 13:35:37 +0000
- Log

CONCLUSIONES

- Gestor de paquetes para Ada/SPARK
- Solucionador de dependencias completo
  - Capaz de encontrar una solución si la hay
- Soporte a varios casos de uso:
  - Instalación de *toolchains*
  - Usuario interesado en binarios finales
  - Desarrollador interesado en dependencias
    - Con apoyo a múltiples desarrollos simultáneos
  - Desarrollador para *targets* cruzados/empotrados
    - Configuraciones estáticas
  - Contribuyente al código abierto
    - Asistente de publicación
- Ecosistema rico y en expansión
  - Desarrollo en GitHub
  - Chat en Gitter

GRACIAS POR LA ATENCIÓN

🔗 https://github.com/alire-project
https://alire.ada.dev
https://gitter.im/ada-lang/Alire
https://www.reddit.com/r/ada/

✉ amosteo@unizar.es

🐦 @mosteobotic (Alejandro R. Mosteo)
@DesChips (Fabien Chouteau)

Centro Universitario de la Defensa Zaragoza
cud.unizar.es

Academia General Militar · Ctra. Huesca s/n · 50090 Zaragoza · 976 739 500