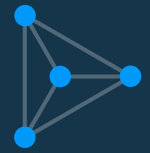


Appsilon

Estructura de un proyecto Shiny

Agustin Perez Santangelo

agustin@appsilon.com

Appsilon

Hoja de Ruta

- Motivación
- Estructura mínima
- Herramientas
    - **{renv}** para controlar dependencias (paquetes)
    - **{config}** para manejar perfiles de configuración/constantes
    - **{testthat}** para unit-testing
    - **{covr}** para trackear testing
    - **{lintr}** para mantener código limpio y consistente
    - **{styler}** para aplicar estilo consistente

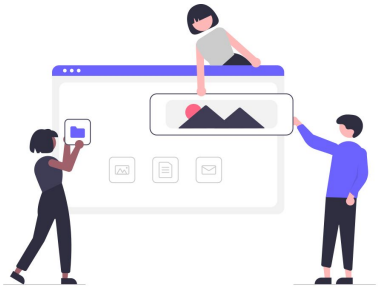
Appsilon

Motivacion

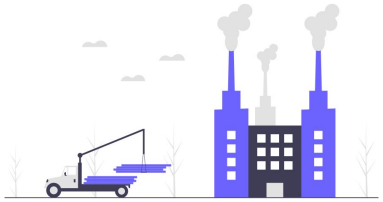
Buenas prácticas, para que?

Apps más complejas

Desarrollo en equipo



Comerciales/producción



Appsilon

Motivacion

Proyecto

- Requerimientos
- Herramientas

Estructura

Mejor organización

Desarrollo más rápido

Cooperación más sencilla

Appsilon

Estructura básica

- Cada proyecto tiene su idiosincrasia
- Punto de partida

  Un **proyecto** “típico” que incluya
  - Datos
  - Scripts de carga y procesamiento de datos
  - App
    - UI / view + Server
    - www (aka assets/static)
      - Imágenes
      - Audio
      - [CSS]
      - [JavaScript]
  - [Tests]

```
mi-proyecto-tipico/
├── app/
│   ├── data/
│   │   └── data.csv
│   ├── app.R
│   ├── helpers.R
│   └── www/
│       ├── mi_css.css
│       ├── mi_js.js
│       ├── imagen.png
│       └── sonido.wav
├── scripts/
│   ├── raw_data/
│   │   └── raw.csv
│   └── preprocess.R
└── tests
```

Appsilon

Propuesta

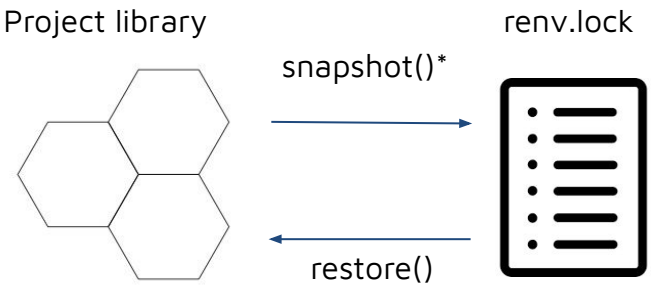
Estructura básica

- **0.** Proyecto como **.Rproj**
  - Ideal para compartimentalizar trabajo
  - Inicia nueva sesión
  - Lee .Renviron (env variables)
  - source("**.Rprofile**")
  - Setea directorio

Appsilon

Estructura básica

- **1.** Controlar entorno de desarrollo con **renv**
  - Versión de R y dependencias (paquetes)
  - Reproducibilidad
    - mismo código, mismos resultados
    - colaboración más sencilla
  - Facilita deployment
  - Cache global de paquetes
    - Carga más rápida
    - Ahorra espacio en disco




```
# Inicializar renv en el proyecto
renv::init()

# Salvar library a `renv.lock`
renv::snapshot()

# Cargar library desde `renv.lock`
renv::restore()
```

.Rprofile
renv.lock
renv/activate.R
renv/library

Appsilon

1_renv_init

Appsilon

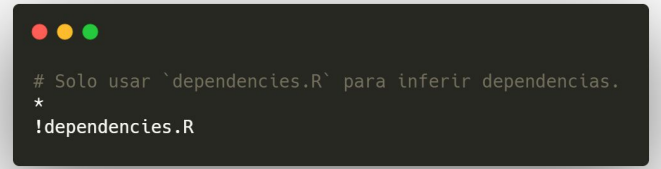
Estructura básica

- **1.** Controlar entorno de desarrollo con **renv**
  - Tipos de snapshot (registro en lock file)
    - *explicit*: dependencias declaradas en DESCRIPTION
    - *implicit*: la intersección entre library y código (default con init())
    - *all*: toda la library

Explicit es ideal para tener mayor control, pero trae problemas para deploy con rsconnect (shinyapps.io, RStudio Connect).

Solución:

- Implicit snapshots
- **.renvignore** (sigue misma lógica que .gitignore)
- dependencies.R
  - listamos pkgs llamando a library()



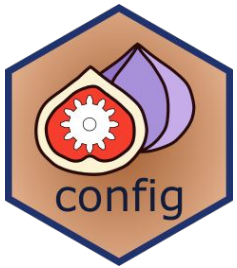
Appsilon

1_renv_init2

Appsilon

Estructura básica

- **2.** Extraer constantes con **config**
  - Permite crear perfiles de constantes
    - test
    - dev
    - production
  - Ahorra tiempo al momento de cambiar valores de constantes.
  - Menos números mágicos y hard-coding en el código de nuestra app.



Appsilon

2_config

Estructura básica

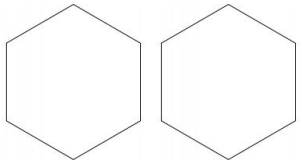
- **3.** Unit testing con **testthat** + tracking con **covr**
  - Chequear funcionalidad
  - Ciclo virtuoso con implementación
  - **covr**
    - trackear testing
    - generar reportes

Appsilon

3_testthat-covr

Estructura básica

- **4.** Código limpio y consistente con **lintr** y **styler**
  - Seguir guia de estilo **tidyverse** (convención adoptada en la comunidad)
  - Minimizar chances de bugs/errores
  - Código más legible y entendible para un humano



Appsilon

4_lintr-styler

Appsilon

Todo esto...y más!

- Docs
- 🎓 Tutorial
- Workshop

Appsilon

[5_rhino]

Appsilon

```
mi-proyecto-tipico/
├── app/
│   ├── data/
│   │   └── data.csv
│   ├── app.R
│   ├── helpers.R
│   └── www/
│       ├── mi css.css
│       ├── mi js.js
│       ├── imagen.png
│       └── sonido.wav
├── scripts/
│   ├── raw data/
│   │   └── raw.csv
│   └── preprocess.R
└── tests
```

```
mi-proyecto-tipico/
├── app
│   ├── js
│   │   └── index.js
│   ├── logic
│   │   └── __init__.R
│   ├── static
│   │   └── favicon.ico
│   ├── styles
│   │   └── main.scss
│   ├── view
│   │   └── __init__.R
│   └── main.R
├── tests
│   ├── cypress
│   │   └── integration
│   │       └── app.spec.js
│   ├── testthat
│   │   └── test-main.R
│   └── cypress.json
├── app.R
├── mi_proyecto_tipico.Rproj
├── dependencies.R
├── renv.lock
├── rhino.yml
├── config.yml
└── .lintr
```

3

0

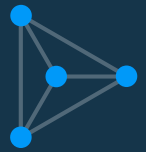
1

2

4

Appsilon

¿Preguntas?



Appsilon

¡Gracias!

Appsilon

Referencias

- .Rproj projects
- renv
- config
- testthat
- covr
- lintr
- styler
- rhino