

The background of the slide is a vibrant cosmic scene. It features a deep purple and blue space filled with numerous small white stars. Several bright, colorful nebulae in shades of pink, magenta, and cyan are scattered across the upper half. In the top right corner, a small blue planet with white clouds is visible. On the right side, a large, partially cut-off red planet with a textured surface is shown. At the bottom, a large blue planet with white clouds and a bright horizon line is visible.

Proyecto HCAI.

CR-Analyze and various Micro-projects

A personal package for counter-rotation
analysis and fun sides.
-By Marcos Bugueño

2 repositorios, uno de mayor prioridad.

CR-Analyzer



portafolio_hcai2023



2 repositorios, uno de mayor prioridad.

CR-Analyzer

Reorganización de las funciones, obtener modularidad, creación de catálogos personales. Posibilidad de generalización a otras simulaciones.



portafolio_hcai2023

Ejemplos, codigos, notas de clase entre otros.



portafolio_hcai2023



c_scripts

Códigos de ejemplo!
Armar un make basico!

python_scripts

Códigos de ejemplo!
Sandboxing!
Uso de los conceptos.

written_notes

>Markdowns, que he
realizado como notas de
clase.
>Escritura muy resumida
de las ideas que he
notado.

Consiste de notas varias, códigos de
ejemplo, para ser utilizados como
índices a futuro.

portafolio_hcai2023



```
graph TD; A[portafolio_hcai2023] --> B[c_scripts]; A --> C[python_scripts]; A --> D[written_notes]; B --> B1[>usando_make]; B --> B2[-arrays.c]; B --> B3[-basic_intro.c]; B --> B4[-condicionales.c]; B --> B5[-libreria_marcorito.h]; B --> B6[-pointers.c]; B --> B7[-prime_detector.c]; B --> B8[-probando_punto_h.c]; B --> B9[-probando_reemplazos.c]; C --> C1[>documentation_test]; C --> C2[>notebook]; C --> C3[>supermercado]; C --> C4[-clases.py]; C --> C5[-herencia.py]; C --> C6[-iterators.py]; C --> C7[-main.py]; C --> C8[-snake.py]; D --> D1[Class5.md]; D --> D2[Class6.md]; D --> D3[Class7.md]; D --> D4[.]; D --> D5[Class1302.md];
```

c_scripts

- >usando_make
- arrays.c
- basic_intro.c
- condicionales.c
- libreria_marcorito.h
- pointers.c
- prime_detector.c
- probando_punto_h.c
- probando_reemplazos.c

python_scripts

- >documentation_test
- >notebook
- >supermercado
- clases.py
- herencia.py
- iterators.py
- main.py
- snake.py

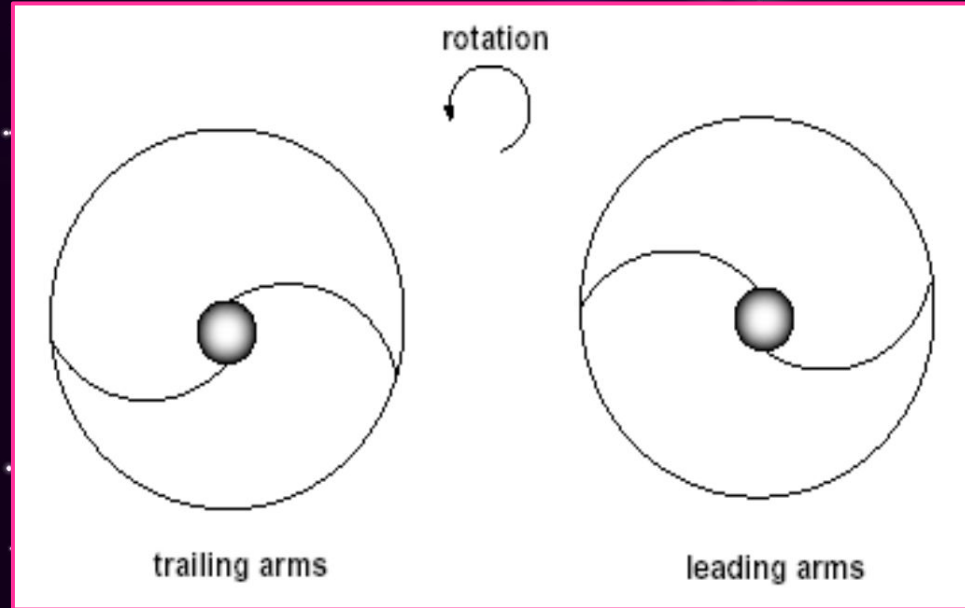
written_notes

- Class5.md
- Class6.md
- Class7.md
- .
- Class1302.md

Consiste de notas varias, códigos de ejemplo, para ser utilizados como índices a futuro.

Estructura y concepto de Contra-rotacion

Lo intuitivo es esperar que una galaxias gire o rote con una dada direccion, especialmente en espirales, en donde tenemos: Trailing arms y leading arms.

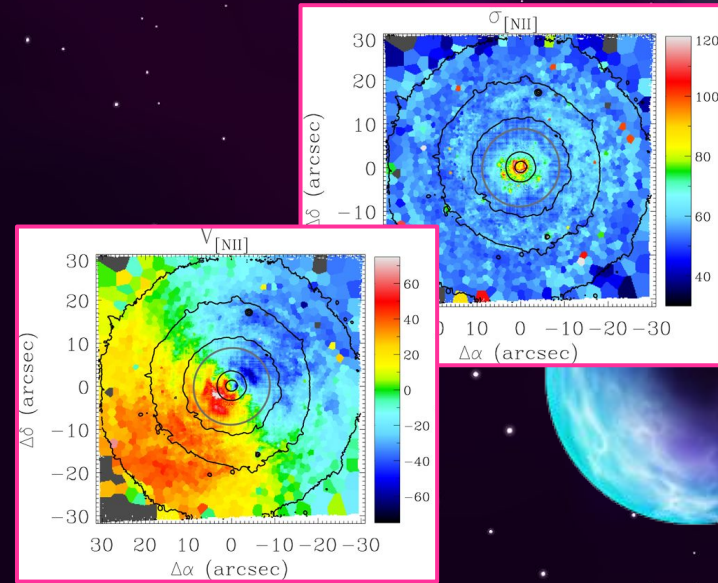


Estructura y concepto de Contra-rotacion

Lo intuitivo es esperar que una galaxias gire o rote con una dada direccion, especialmente en espirales, en donde tenemos: Trailing arms y leading arms.

Sin embargo, es extraño esperar que toda galaxia espiral carezca de una componente que no siga la regla.

>¿El fenómeno entrega información sobre el pasado de una galaxia?



CR-Analyze circularity.py

spherical_coords_from_vector(vector)

```
vector = [x,y,z]
-----
returns(r, theta, phi)
```

inertia_tensor(Masas, Coordenadas,
CentralPos)

```
masas = [m1, m2, ..., mN]
Coordenadas = [[x1, y1, z1], ..., [xN, yN, zN]]
CentralPos = [x,y,z]
-----
returns(I)
```

star_particles_rotated_once_euler_
method

```
subhaloID = int
snapNum = int
basepath = str
radius_limit = float
minmetal = float
maxmetal = float
snap_header = {}
-----
returns(stars=DataFrame, M=3x3array)
```

circularities_euler_method

```
subhaloID = int
snapNum = int
basepath = str
radius_limit = float
minmetal = float
maxmetal = float
radius_limit_rotation = float
minmetal_rotation = float
maxmetal_rotation = float
-----
returns(stars=DataFrame)
```

matrix_from_spherical(r, theta, phi)

```
r = float
theta = float(rad)
phi = float(rad)
-----
returns(M)
```

diagonalization_of_inertia(I)

```
I = 3x3 Array
-----
returns(rotation_matrix)
```

star_particles_rotated_once

```
subhaloID = int
snapNum = int
basepath = str
radius_limit = float
minmetal = float
maxmetal = float
snap_header = {}
-----
returns(stars=DataFrame, S,S1=3x3array)
```

circularities_diagmethod

```
subhaloID = int
snapNum = int
basepath = str
radius_limit = float
minmetal = float
maxmetal = float
radius_limit_rotation = float
minmetal_rotation = float
maxmetal_rotation = float
-----
returns(stars=DataFrame)
```


CR-Analyze particle_follower.py

load_counterrotating_particles

```
subhaloid = int
-----
returns(stars_CR=DataFrame)
```

filter_particles_by_ID

```
subhalo_df = DataFrame
star_data = {}
-----
returns(filtered_star_data_df)
```

generate_time_tables

```
subhaloid = int
start_snap = int
end_snap = int
-----
returns(0)
guarda en csv
```

generate_time_tables_full

```
subhaloid = int
start_snap = int
end_snap = int
-----
returns(0)
guarda en csv
```

load_stellar_particles

```
snapshot = int
-----
returns(star_data={})
```

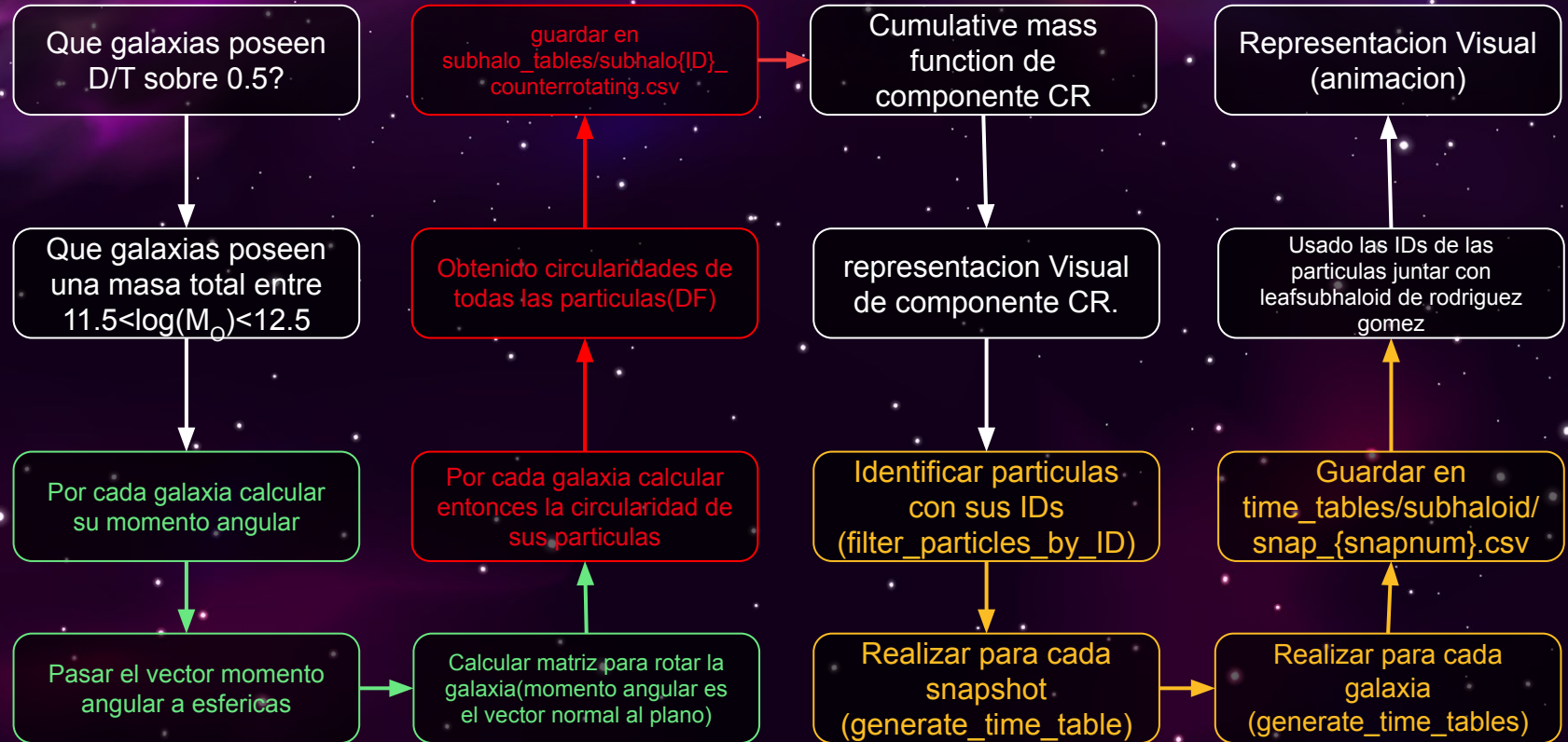
generate_time_table

```
subhalo_df = DataFrame
subhalopos_arr =
[[x0, y0, z0], ..., [x99, y99, z99]]
snapshot = int
-----
returns(star_data_df)
```

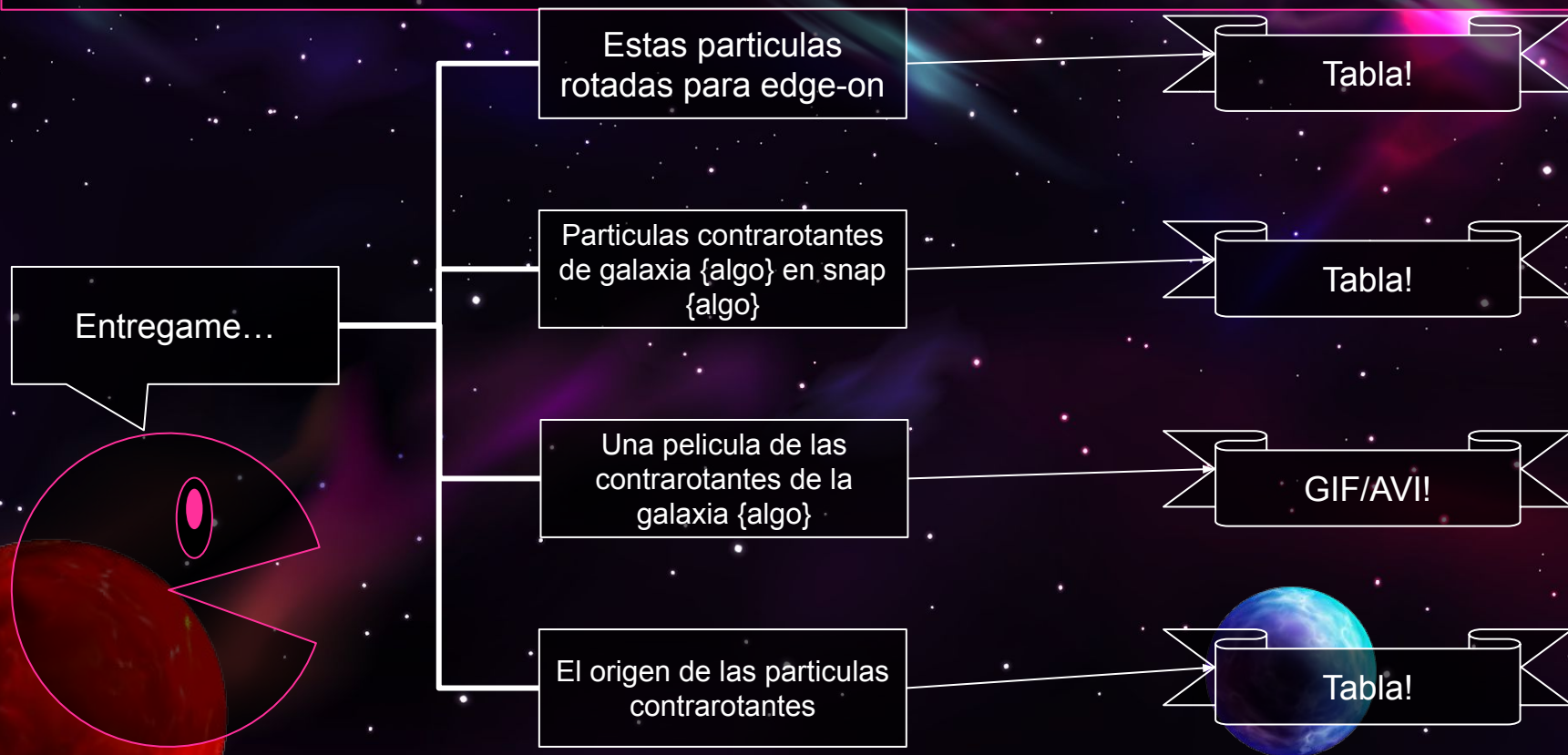
generate_time_table_full

```
subhalo_df = DataFrame
subhalopos_arr =
[[x0, y0, z0], ..., [x99, y99, z99]]
snapshot = int
-----
returns(subhalo_df)
```

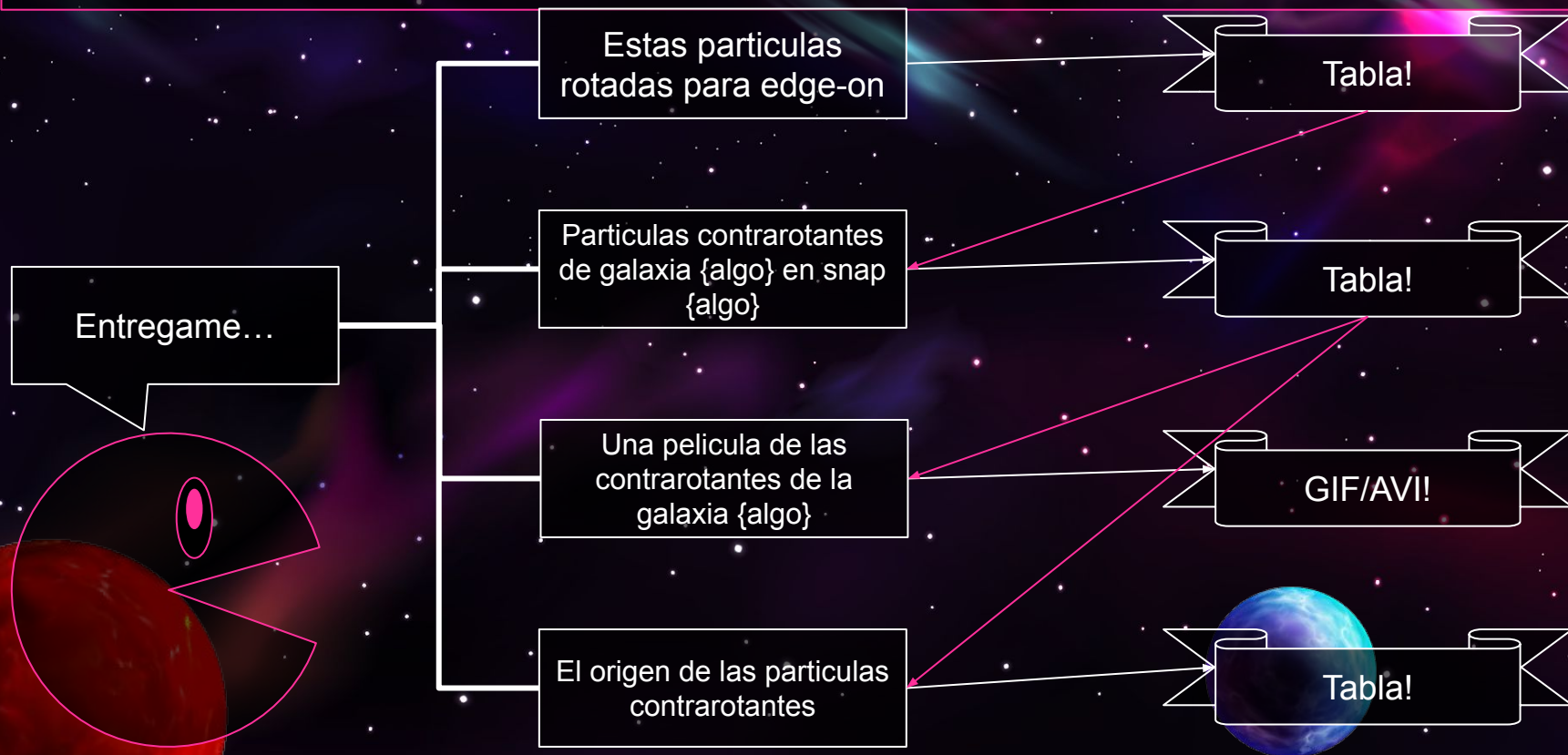
Método actual.



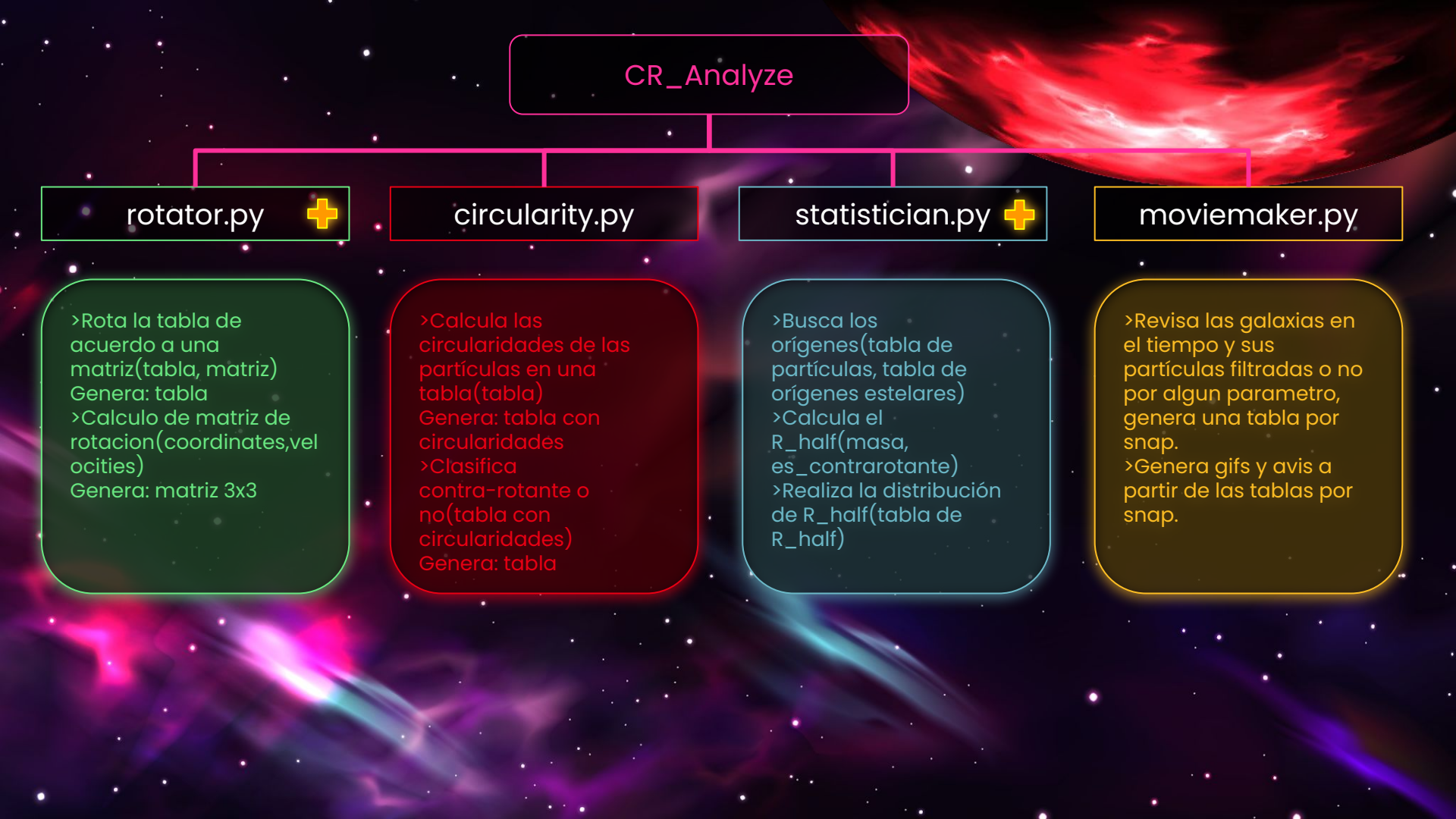
Counterrotation-Analyzer(Metodo ideal) SOBRESIMPLIFICACION



Counterrotation-Analyzer(Metodo ideal) SOBRESIMPLIFICACION



CR_Analyze



```
graph TD; CR_Analyze[CR_Analyze] --> rotator.py[rotator.py]; CR_Analyze --> circularity.py[circularity.py]; CR_Analyze --> statistician.py[statistician.py]; CR_Analyze --> moviemaker.py[moviemaker.py];
```

rotator.py



>Rota la tabla de acuerdo a una matriz(tabla, matriz)
Genera: tabla
>Calculo de matriz de rotacion(coordinates,velocities)
Genera: matriz 3x3

circularity.py

>Calcula las circularidades de las partículas en una tabla(tabla)
Genera: tabla con circularidades
>Clasifica contra-rotante o no(tabla con circularidades)
Genera: tabla

statistician.py



>Busca los orígenes(tabla de partículas, tabla de orígenes estelares)
>Calcula el R_half(masa, es_contrarotante)
>Realiza la distribución de R_half(tabla de R_half)

moviemaker.py

>Revisa las galaxias en el tiempo y sus partículas filtradas o no por algun parametro, genera una tabla por snap.
>Genera gifs y avis a partir de las tablas por snap.

CR_Analyze

rotator.py



spherical_coords_from_vector(vector)

matrix_from_spherical(r, theta, phi)

table_rotate(tabla, matriz)



star_particles_rotated_once_euler_method(tabla)

star_particles_rotated_n_euler_method(tabla)



> tabla rotada
> matriz de rotacion

circularity.py

circularities_euler_method

generate_cr_table



> tabla, con circularidades y CR true{1} o false{0}

statistician.py



calculate_r_half(tables)



distribution_of_r_half(table)



distribution_of_metallicity(table)



detect_origin(particle_table)



> tabla de distribuciones

moviemaker.py

generate_time_table_full

generate_time_tables_full

generate_time_table



generate_time_tables



record



> tablas por snap
> gif/avi de la galaxia

CR_Analyze

rotator.py



spherical_coords_from_vector(vector)

matrix_from_spherical(r, theta, phi)

table_rotate(tabla, matriz)



star_particles_rotated_once_euler_method(tabla)

star_particles_rotated_n_euler_method(tabla)



>tabla rotada
>matriz de rotacion

circularity.py

circularities_euler_method

generate_cr_table



>tabla, con circularidades y
CR true{1} o false{0}

statistician.py



calculate_r_half(tables)



distribution_of_r_half(table)



distribution_of_metallicity(table)



detect_origin(particle_table)



>tabla de distribuciones

moviemaker.py

generate_time_table_full

generate_time_tables_full

generate_time_table



generate_time_tables



record



>tablas por snap
>gif/avi de la galaxia

CR_Analyze

IMPORTANTE QUE LAS FUNCIONES HAGAN PREGUNTAS!
TENGO ESTA TABLA?
TENGO ESTE CATALOGO?
NO? GENERALO!

rotator.py

spherical_coords_from_vector(vector)

matrix_from_spherical(r, theta, phi)

table_rotate(tabla, matriz)

star_particles_rotated_once_euler_method(tabla)

star_particles_rotated_n_euler_method(tabla)

>tabla rotada
>matriz de rotacion

circularity.py

circularity(tabla)

generate_circularity(tabla)

>tabla, con circularidades y CR true{1} o false{0}

statisician.py

calculate_r_half(tables)

distribution_of_r_half(tables)

distribution_of_mass(tables)

detect_origin(particle_table)

>tabla de distribuciones

moviemaker.py

generate_time_table_full

generate_time_tables_full

generate_movie_tables

generate_time_tables

record

>tablas por snap
>gif/avi de la galaxia

Template by [Slidesgo](#) "Realistic Galaxy
Consulting Toolkit Infographics"

Gracias!

CR-Analyzer



portafolio_hcai2023



IDEAS ALEATORIAS

Peliculas en
3d

Angulo de
caida de
progenitores

Evolucion de
la masa de la
componente
CR

Calculo
alternativo del
birthprogenitor
?

Importa el
ambiente

Mas alla de
las galaxias
centrales?

R200 para
galaxias
satelite?

