viu Universidad Internacional de Valencia

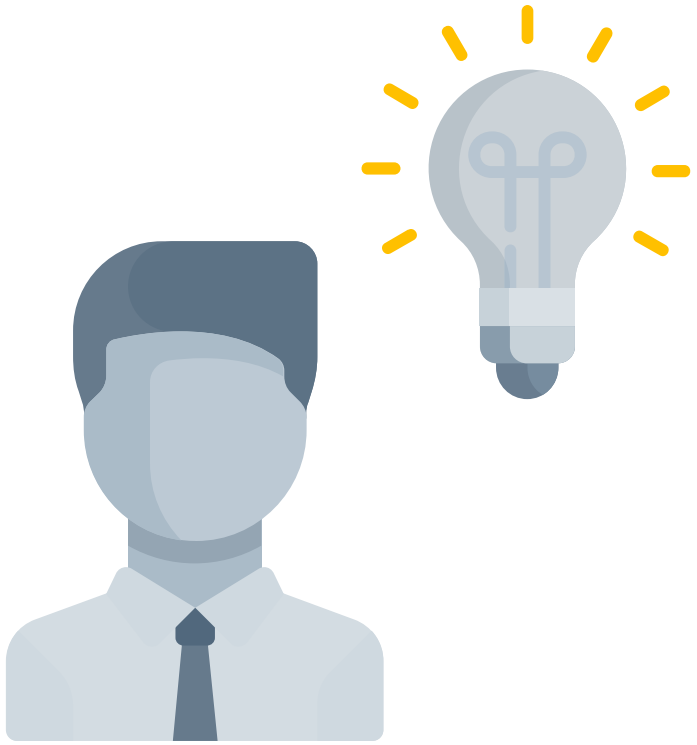
Máster en Inteligencia Artificial

Causal discovery Unit testing

Adrien Felipe

Director Gherardo Varando

Causal discovery Unit testing

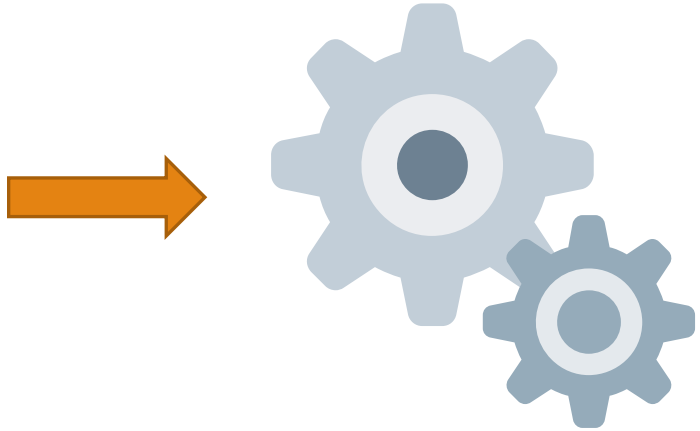


Causal discovery Unit testing

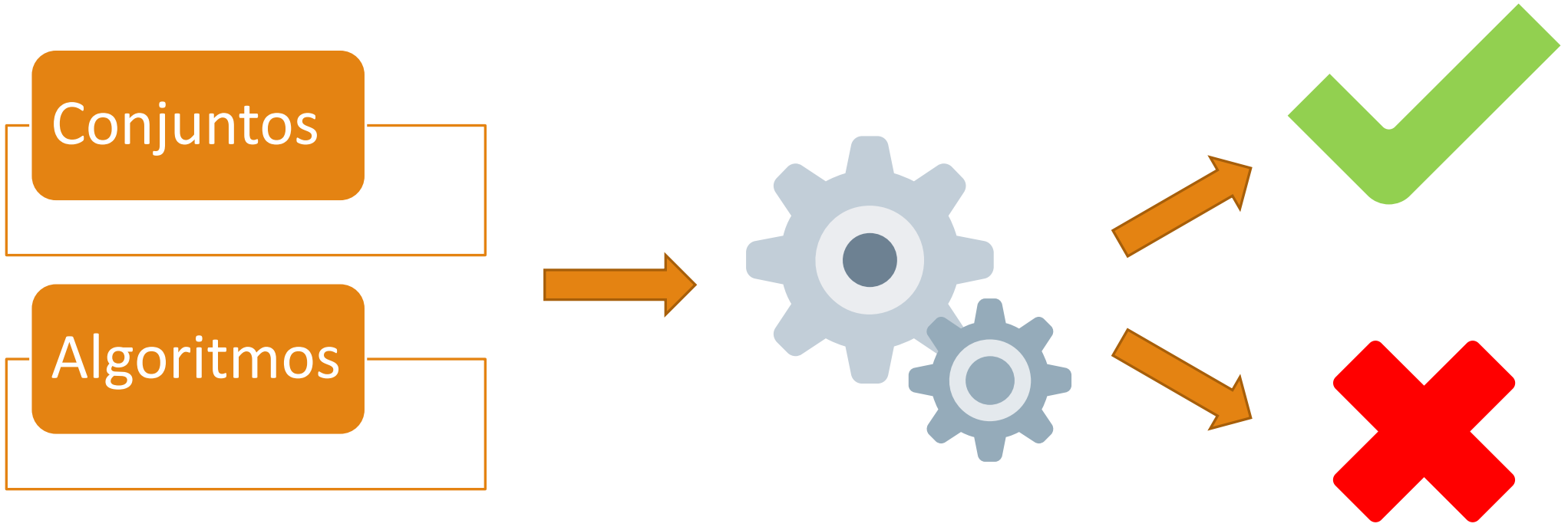
Causal discovery Unit testing

Conjuntos

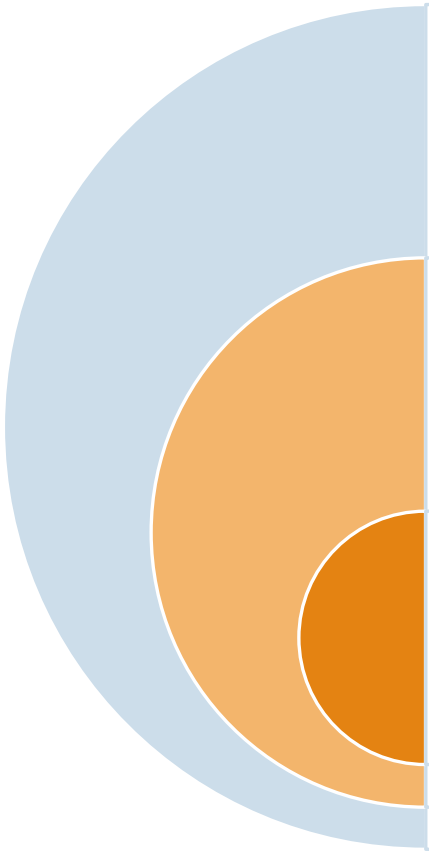
Algoritmos



Causal discovery Unit testing



Desglose



1. Introducción
   - Estado del arte
   - Objetivos
2. Framework
   - Funcionalidades
   - Live demo
3. Conclusión
   - Evaluación algoritmos
   - Desarrollos futuros

Causal discovery

Observaciones

Relaciones causales

Técnicas de exploración causal

3 métodos principales

Técnicas de exploración causal

3 métodos principales

Restricciones

Técnicas de exploración causal

3 métodos principales

Restricciones

Puntuación

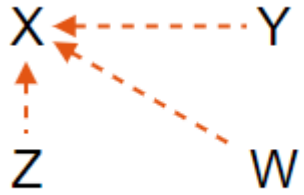
Técnicas de exploración causal

3 métodos principales

Restricciones

Puntuación

Asimetrías



Ecuaciones estructurales

Ecuaciones estructurales

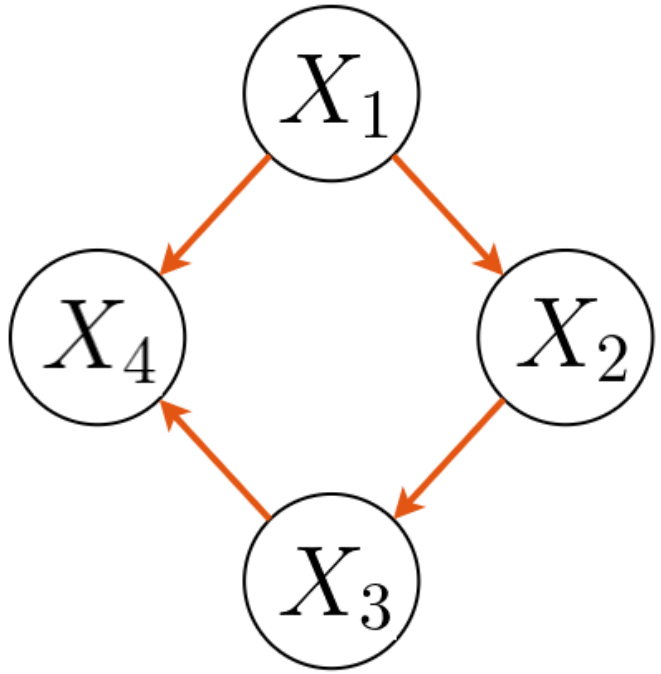
$$
\begin{aligned}
X_1 &:= f_1(N_1) \\
X_2 &:= f_2(X_1, N_2) \\
X_3 &:= f_3(X_2, N_3) \\
X_4 &:= f_4(X_3, X_1, N_4)
\end{aligned}
$$

Asignación de variables

Ecuaciones estructurales

$$
\begin{aligned}
X_1 &:= f_1(N_1) \\
X_2 &:= f_2(X_1, N_2) \\
X_3 &:= f_3(X_2, N_3) \\
X_4 &:= f_4(X_3, X_1, N_4)
\end{aligned}
$$

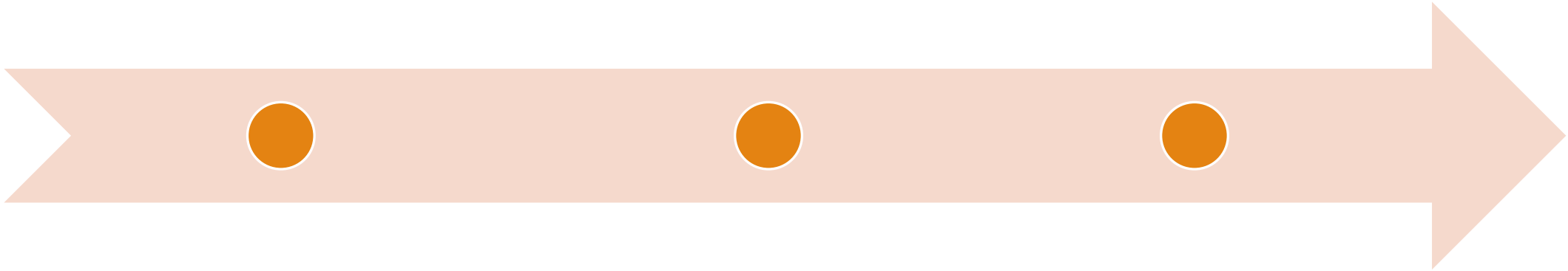
Asignación de variables



Grafo estructural

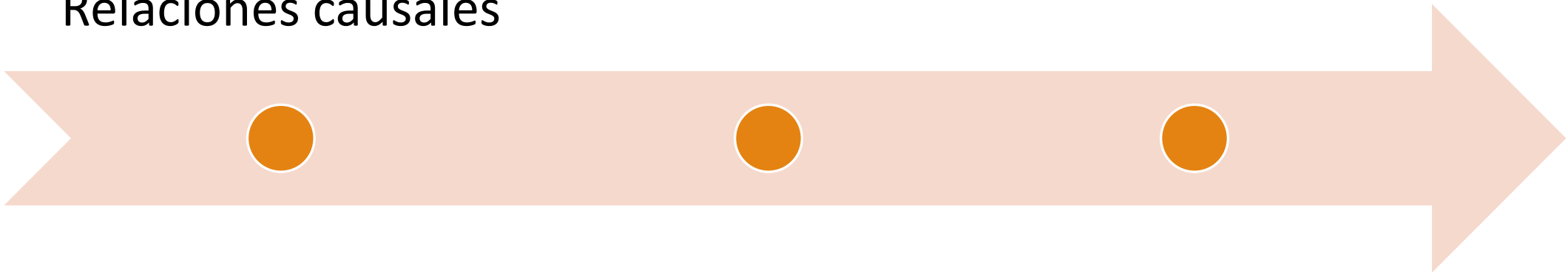
Introducción

Objetivos



Objetivos

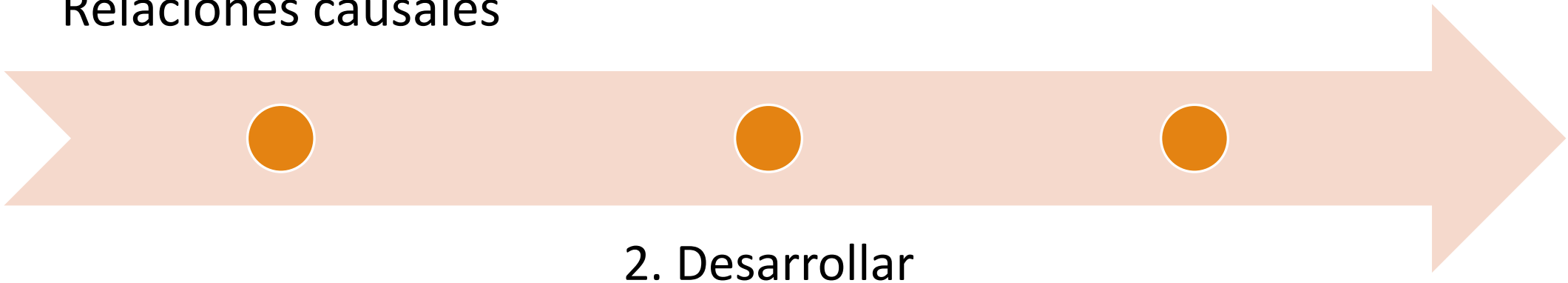
1. Determinar las Relaciones causales



Introducción

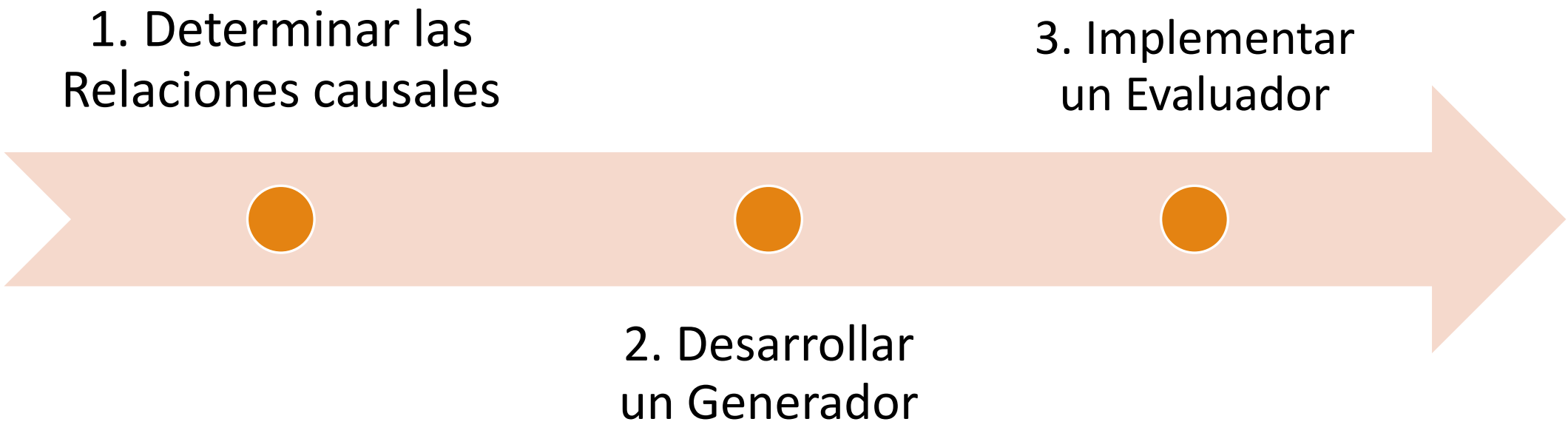
Objetivos

1. Determinar las Relaciones causales

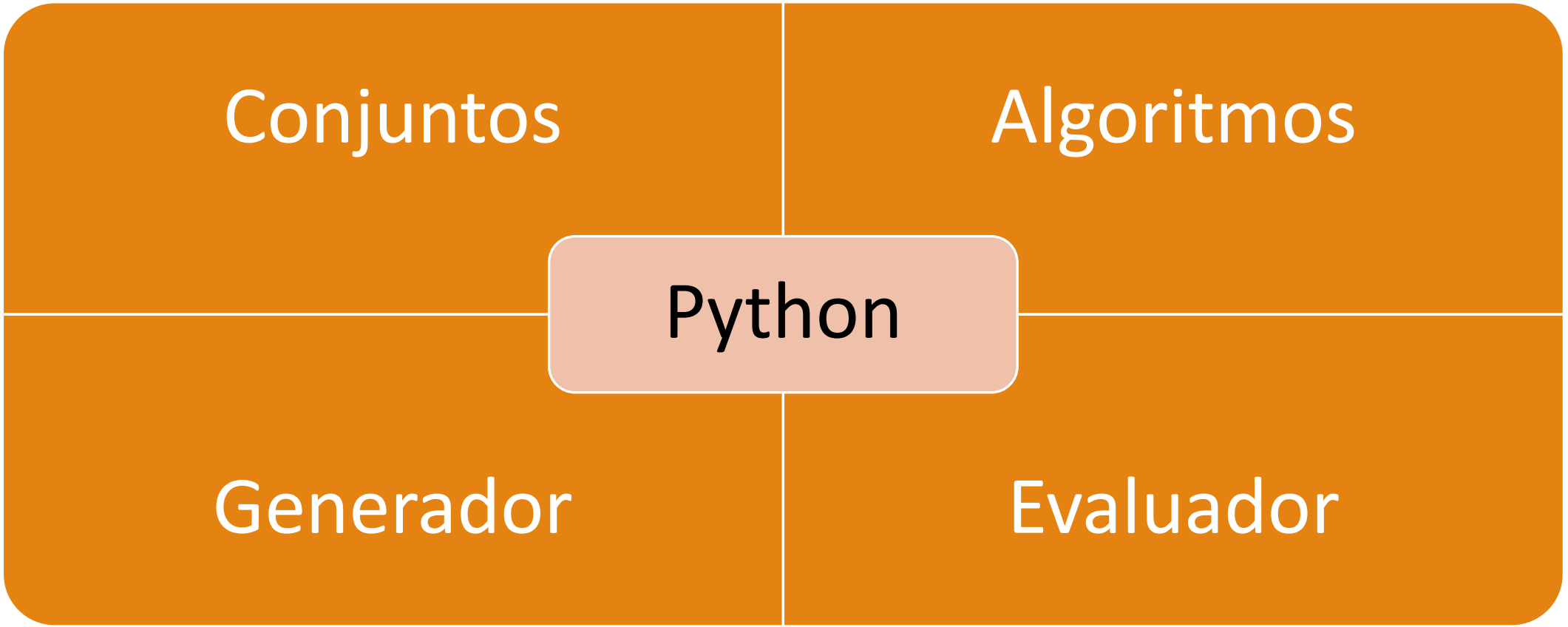


2. Desarrollar un Generador

Objetivos


1. Determinar las Relaciones causales
2. Desarrollar un Generador
3. Implementar un Evaluador

Framework



Framework

Conjuntos causales

Conjuntos causales

Elementales

- Directa
- Múltiple
- En cadena
- Relacional
- …

Framework

Conjuntos causales

Elementales

- Directa
- Múltiple
- En cadena
- Relacional
- …

Funcionales

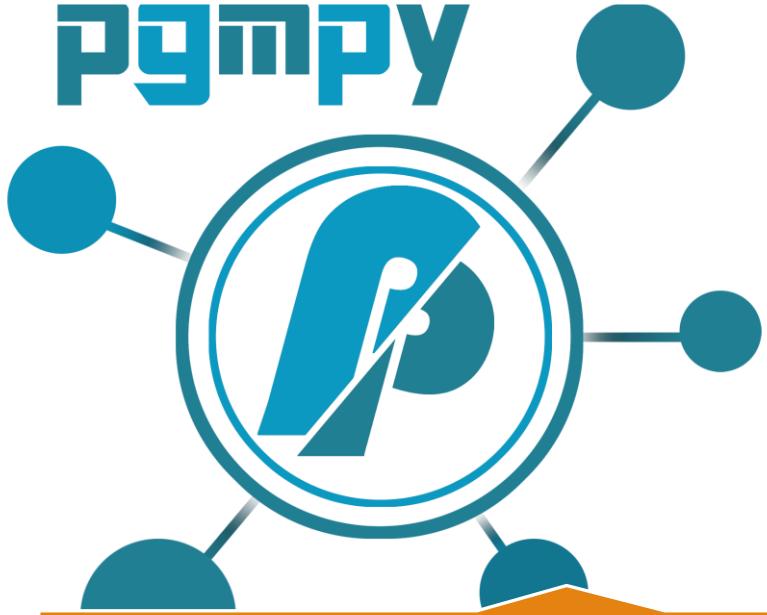
- Marketing
- Genes
- Logs
- Sensores
- …

Conjuntos causales

Elementales	Funcionales	Unitarios
• Directa • Múltiple • En cadena • Relacional • …	• Marketing • Genes • Logs • Sensores • …	• Instantaneo • Latente • Desfase

Framework

Algoritmos de exploración

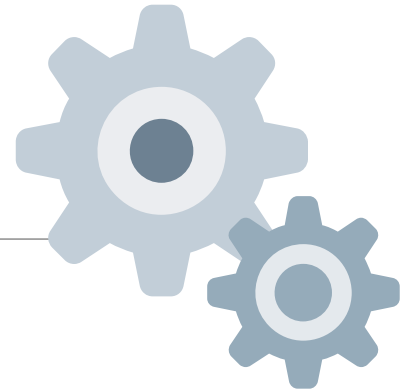


Pgmpy

PyAgrum

Framework

Generador – Propósito

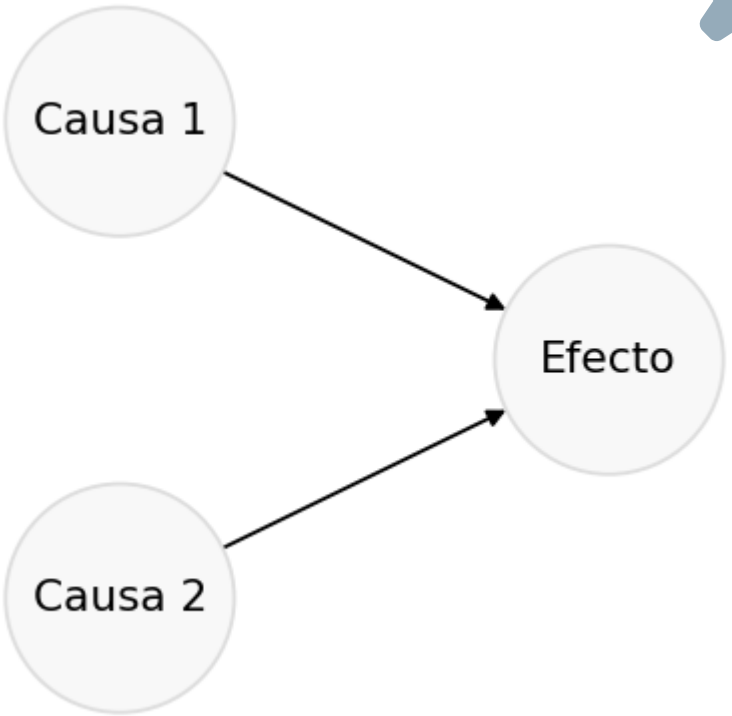


Causa 1	Causa 2	X 3	X 4	Efecto
8	13	2	0	21
7	3	3	23	10
6	8	2	12	14
4	10	4	28	14
9	6	0	7	15
6	15	3	9	16

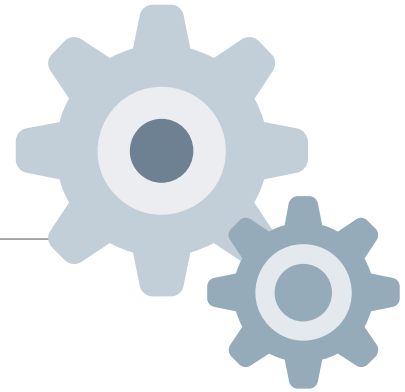
Framework

Generador – Propósito

Causa 1	Causa 2	X 3	X 4	Efecto
8	13	2	0	21
7	3	3	23	10
6	8	2	12	14
4	10	4	28	14
9	6	0	7	15
6	15	3	9	16

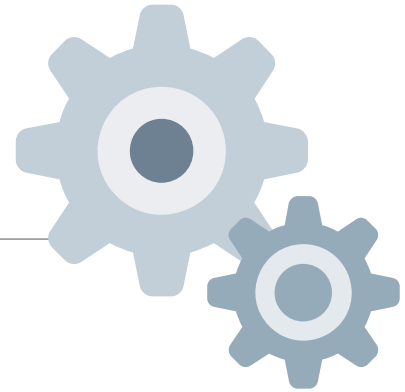


Generador – Ejemplo simple



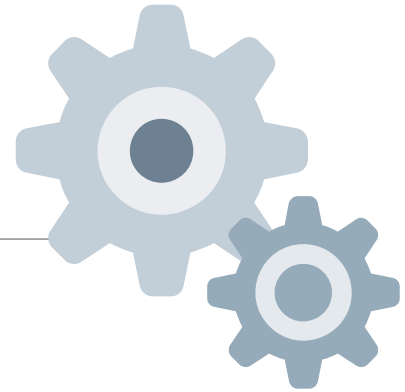
```
generator = Generator() \
```

Generador – Ejemplo simple



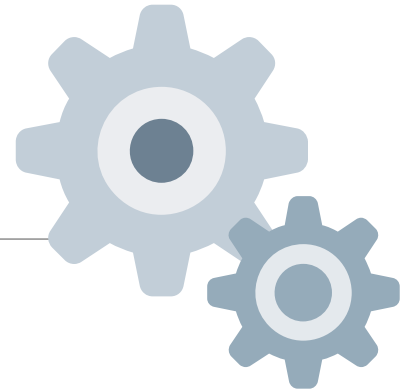
```
generator = Generator() \
    .add_discrete(10, label='Causa 1') \
```

Generador – Ejemplo simple



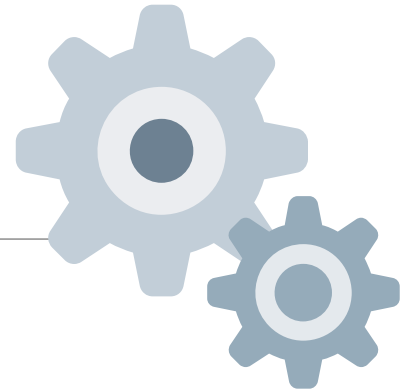
```
generator = Generator() \
    .add_discrete(10, label='Causa 1') \
    .add_discrete(20, label='Causa 2') \
```

Generador – Ejemplo simple



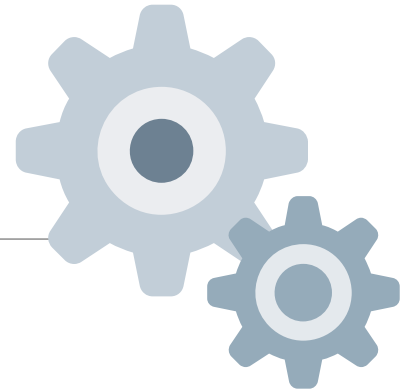
```
generator = Generator() \
    .add_discrete(10, label='Causa 1') \
    .add_discrete(20, label='Causa 2') \
    .add_discrete(5) \
```

Generador – Ejemplo simple



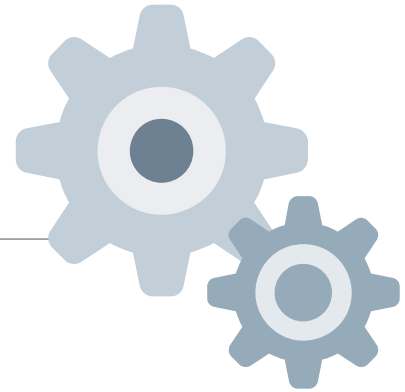
```
generator = Generator() \
    .add_discrete(10, label='Causa 1') \
    .add_discrete(20, label='Causa 2') \
    .add_discrete(5) \
    .add_discrete(30) \
```

Generador – Ejemplo simple



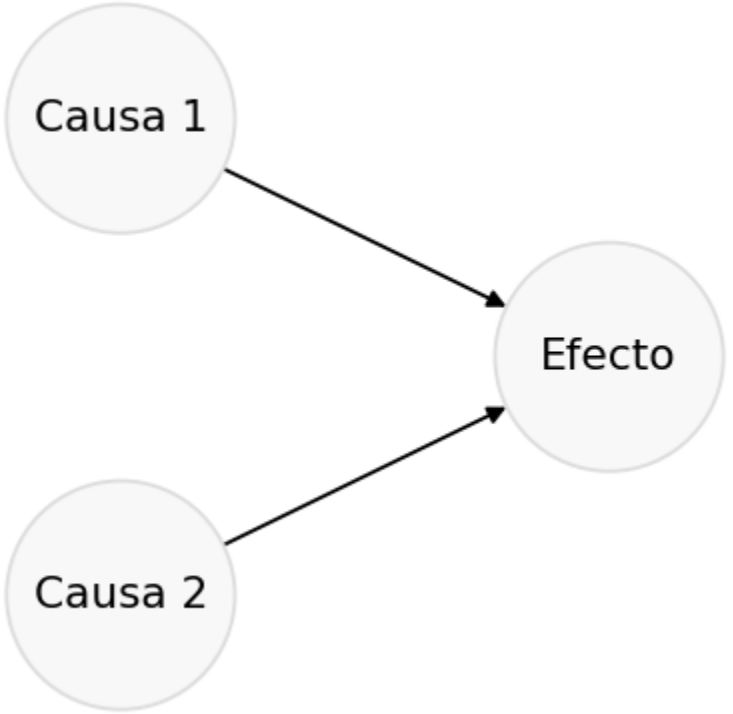
```
generator = Generator() \
    .add_discrete(10, label='Causa 1') \
    .add_discrete(20, label='Causa 2') \
    .add_discrete(5) \
    .add_discrete(30) \
    .add_function(lambda h: h.e(1) + h.e(2), label='Efecto')
```

Generador – Ejemplo simple

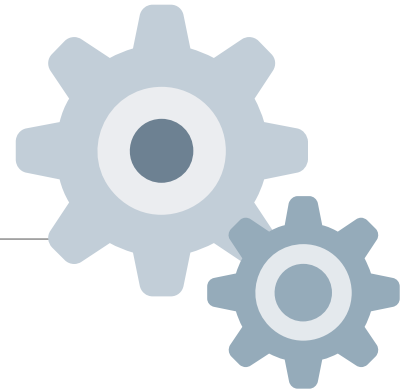


```
generator = Generator() \
    .add_discrete(10, label='Causa 1') \
    .add_discrete(20, label='Causa 2') \
    .add_discrete(5) \
    .add_discrete(30) \
    .add_function(lambda h: h.e(1) + h.e(2), label='Efecto')


generator.plot_relations()
```

Generador – Ejemplo simple

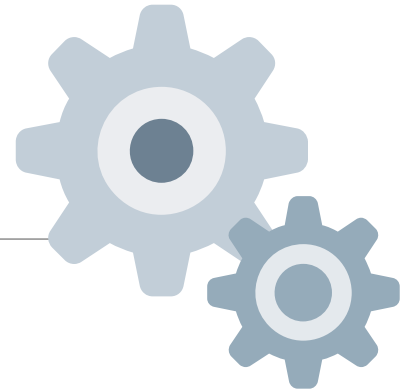


Generador – Ejemplo simple



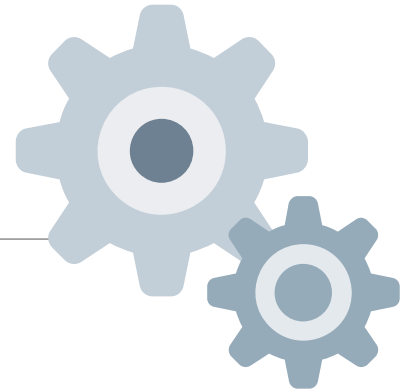
```
generator = Generator() \
    .add_discrete(10, label='Causa 1') \
    .add_discrete(20, label='Causa 2') \
    .add_discrete(5) \
    .add_discrete(30) \
    .add_function(lambda h: h.e(1) + h.e(2), label='Efecto')


generator.plot_relations()
```

Generador – Ejemplo simple



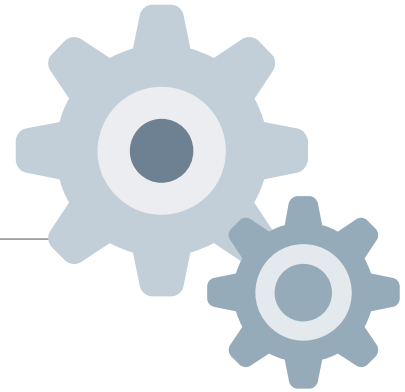
```
generator = Generator() \
    .add_discrete(10, label='Causa 1') \
    .add_discrete(20, label='Causa 2') \
    .add_discrete(5) \
    .add_discrete(30) \
    .add_function(lambda h: h.e(1) + h.e(2), label='Efecto')


generator.plot_relations()
df = generator.generate(1000)
```

Generador – Ejemplo simple



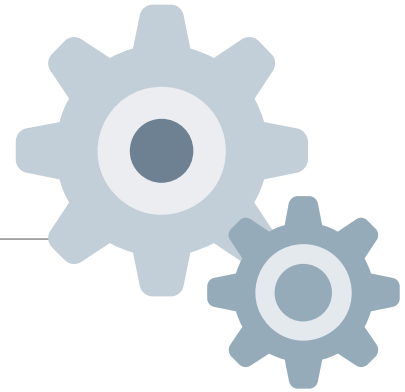
Causa 1	Causa 2	X 3	X 4	Efecto
8	13	2	0	21
7	3	3	23	10
6	8	2	12	14
4	10	4	28	14
9	6	0	7	15
6	15	3	9	16

Generador – Eventos disponibles



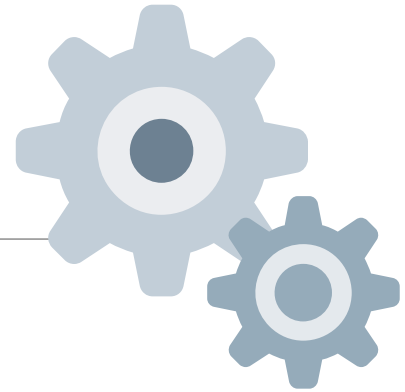
```
generator = Generator() \
```

Generador – Eventos disponibles



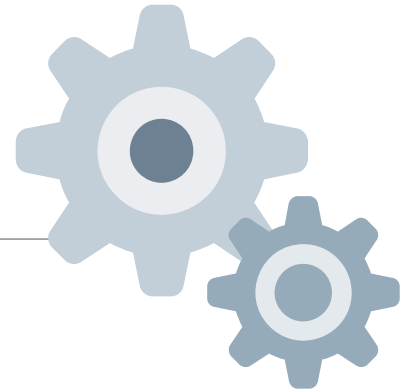
```
generator = Generator() \
    .add_discrete(3, weights=(0.2, 0.3, 0.5)) \
```

Generador – Eventos disponibles



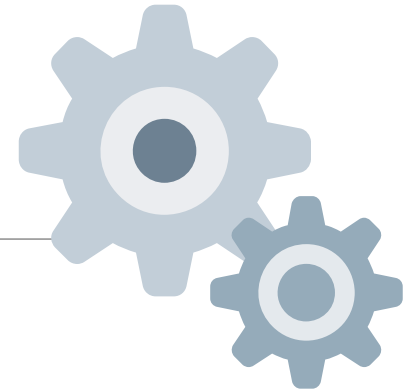
```
generator = Generator() \
    .add_discrete(3, weights=(0.2, 0.3, 0.5)) \
    .add_categorical(['Pequeño', 'Grande'], weights=(0.4, 0.6)) \
```

Generador – Eventos disponibles



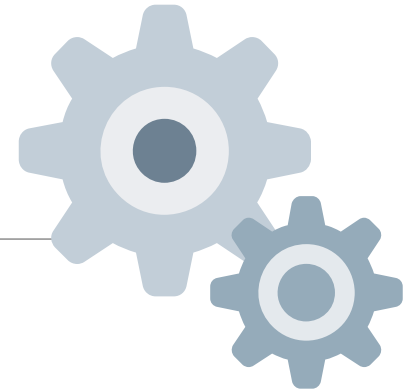
```
generator = Generator() \
    .add_discrete(3, weights=(0.2, 0.3, 0.5)) \
    .add_categorical(['Pequeño', 'Grande'], weights=(0.4, 0.6)) \
    .add_constant(48) \
```

Generador – Eventos disponibles



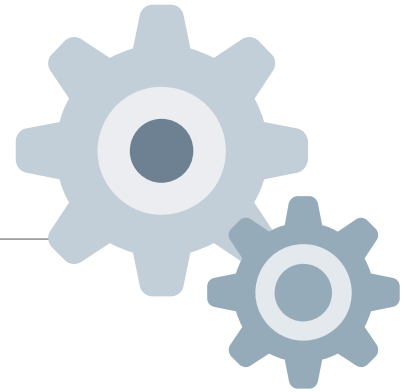
```
generator = Generator() \
    .add_discrete(3, weights=(0.2, 0.3, 0.5)) \
    .add_categorical(['Pequeño', 'Grande'], weights=(0.4, 0.6)) \
    .add_constant(48) \
    .add_uniform(min=0, max=1) \
```

Generador – Eventos disponibles



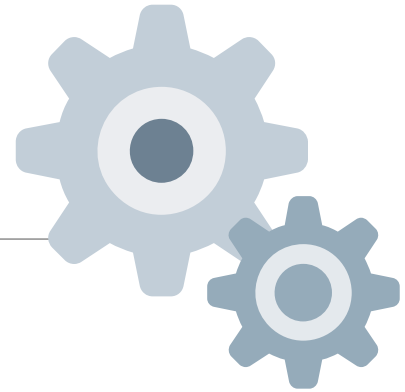
```
generator = Generator() \
    .add_discrete(3, weights=(0.2, 0.3, 0.5)) \
    .add_categorical(['Pequeño', 'Grande'], weights=(0.4, 0.6)) \
    .add_constant(48) \
    .add_uniform(min=0, max=1) \
    .add_gaussian(mu=0, sigma=1) \
```

Generador – Eventos disponibles



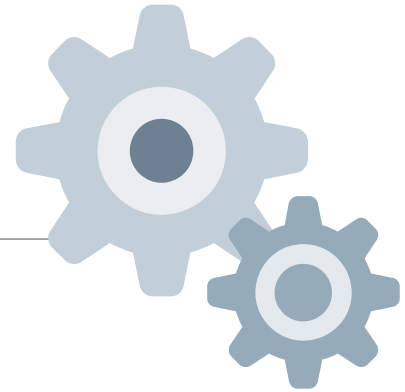
```
generator = Generator() \
    .add_discrete(3, weights=(0.2, 0.3, 0.5)) \
    .add_categorical(['Pequeño', 'Grande'], weights=(0.4, 0.6)) \
    .add_constant(48) \
    .add_uniform(min=0, max=1) \
    .add_gaussian(mu=0, sigma=1) \
    .add_linear(start=0, step=1) \
```

Generador – Eventos disponibles



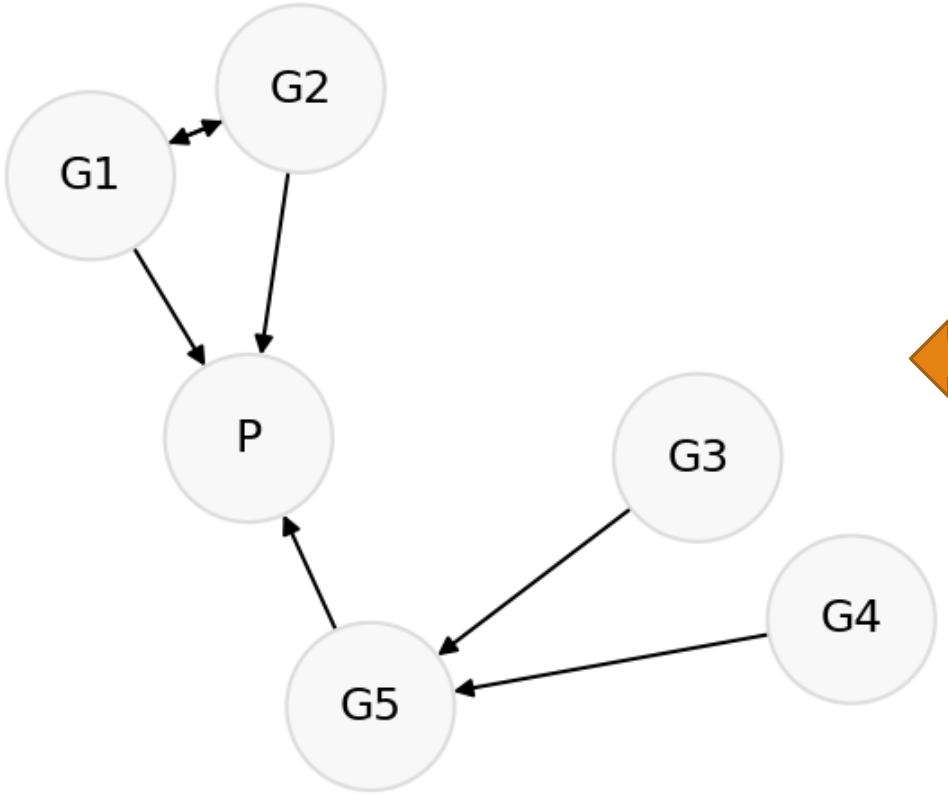
```
generator = Generator() \
    .add_discrete(3, weights=(0.2, 0.3, 0.5)) \
    .add_categorical(['Pequeño', 'Grande'], weights=(0.4, 0.6)) \
    .add_constant(48) \
    .add_uniform(min=0, max=1) \
    .add_gaussian(mu=0, sigma=1) \
    .add_linear(start=0, step=1) \
    .add_function(custom_function)
```

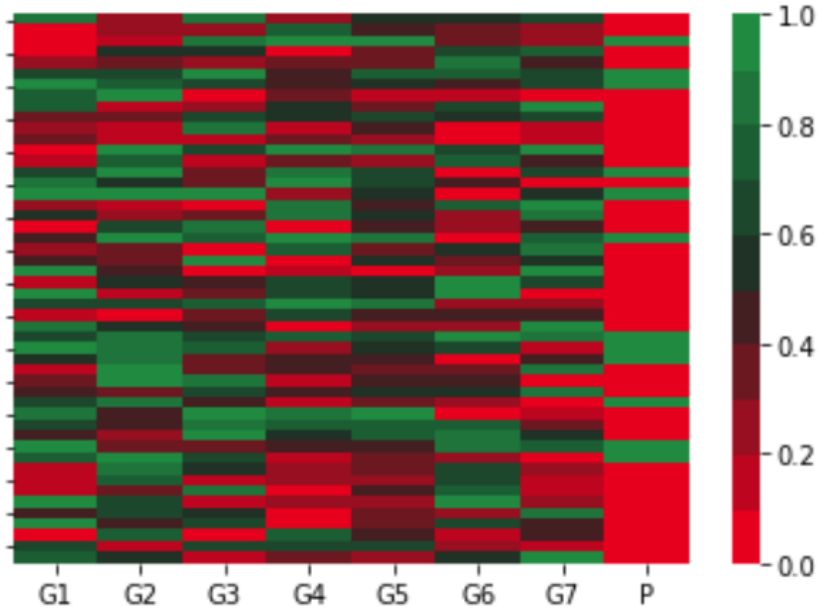
Generador – Evento función



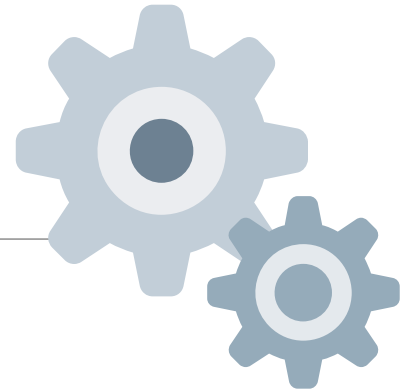
```python
def custom_function(history: History) -> bool:
    gene1 = history.get_event(1) > 0.5 + Gaussian(0, 0.1)
    gene2 = history.get_event(2) > 0.6 + Gaussian(0, 0.2)
    gene5 = history.get_event(5) > 0.9 + Gaussian(0, 0.1)

    return gene1 and gene2 or gene5
```

Generador – Expresión de genes



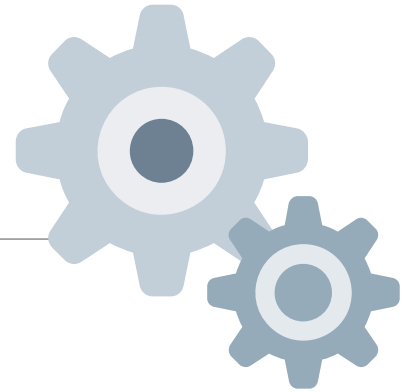


Generador – Control sobre variables



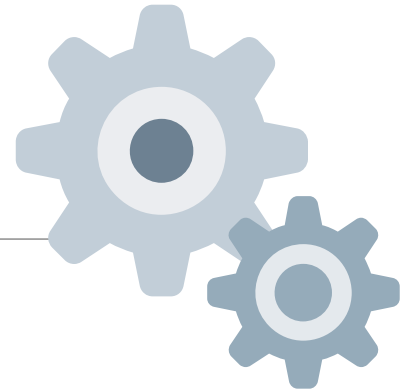
```
generator = Generator() \
```

Generador – Control sobre variables



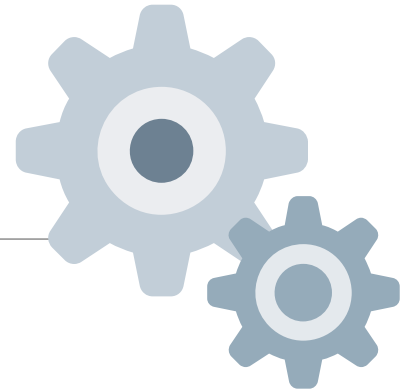
```
generator = Generator() \
    .add_gaussian(round=3) \
```

Generador – Control sobre variables



```
generator = Generator() \
    .add_gaussian(round=3) \
    .add_gaussian(dtype=int) \
```

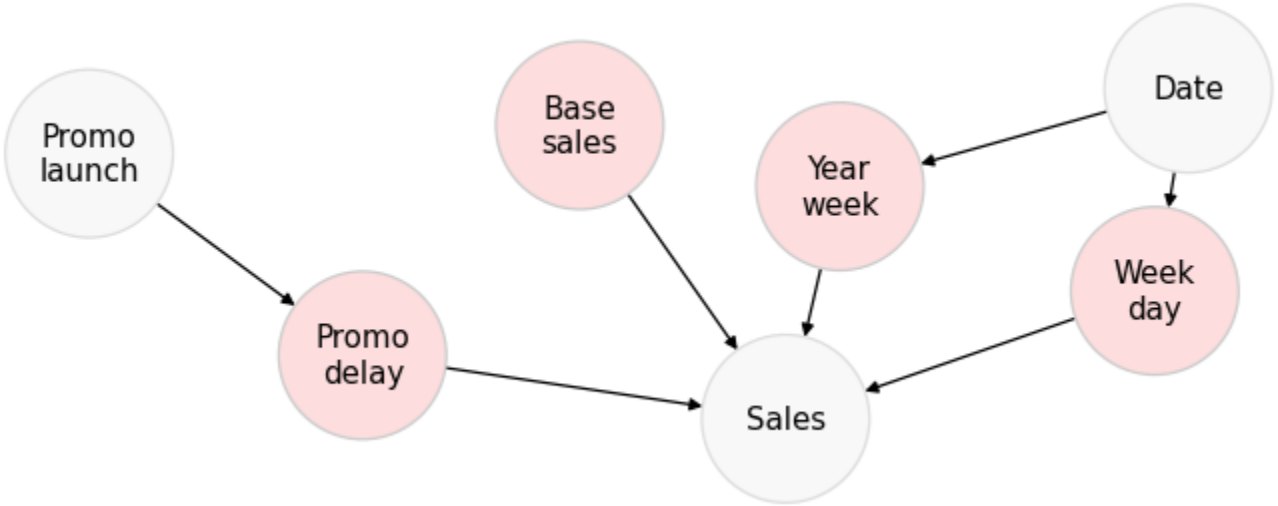
Generador – Control sobre variables

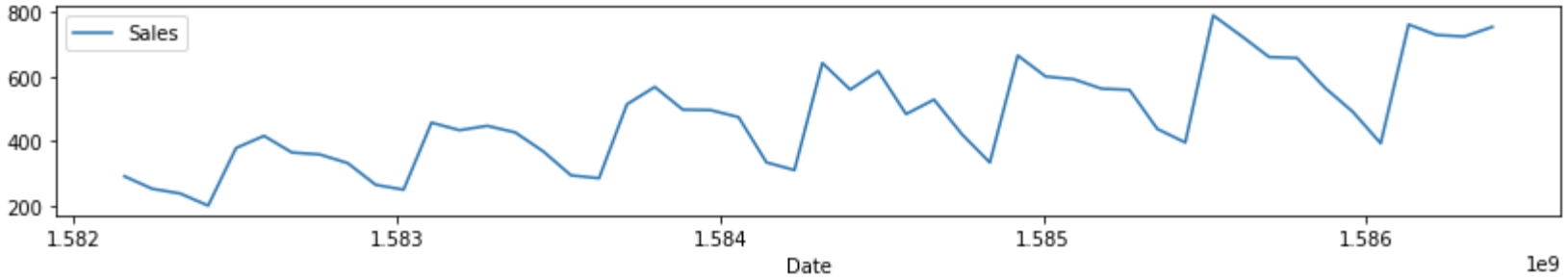


```
generator = Generator() \
    .add_gaussian(round=3) \
    .add_gaussian(dtype=int) \
    .add_gaussian(shadow=True)
```

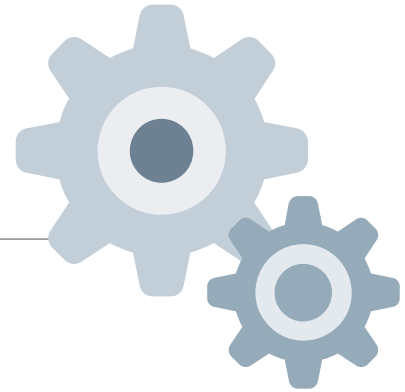
Framework

Generador – Histórico de ventas



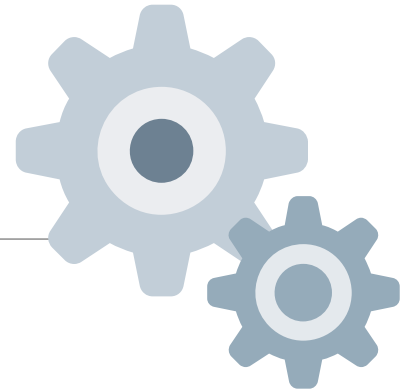


Generador – Variable de tiempo



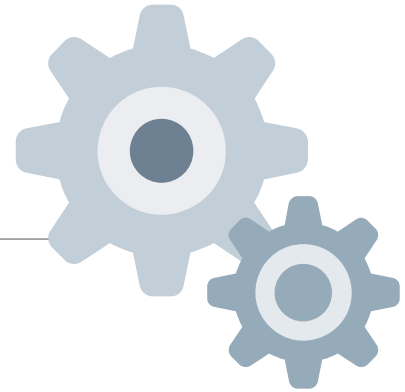
```
generator.set_time()
```

Generador – Variable de tiempo



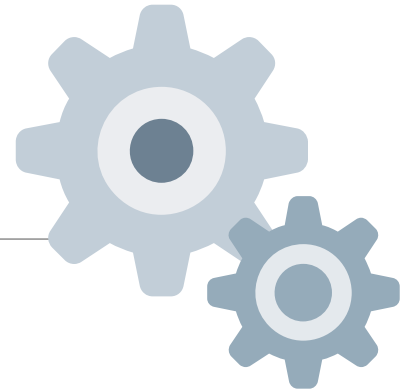
```
generator.set_time()

generator.set_time('2020-02-20 12:45')
```

Generador – Variable de tiempo



```
generator.set_time()

generator.set_time('2020-02-20 12:45')

generator.set_time('2020-02-20', step='1d')
```

Generador – Variable de tiempo



```
generator.set_time()

generator.set_time('2020-02-20 12:45')

generator.set_time('2020-02-20', step='1d')

generator.set_time('2020-02-20', step='1h', precision='10m')
```

Framework

Evaluador

```
from ui.config
```

Framework

Evaluador

```
from ui.config

scripts = [
    PgmpyScript.PC(),
    PyAgrumScript.TS(),
]
```

Framework

Evaluador

```
from ui.config

scripts = [
    PgmpyScript.PC(),
    PyAgrumScript.TS(),
]

datasets = [
    DirectCausalityDataset.linear(1000),
    ChainedCausalityDataset.linear(100),
]
```

Framework

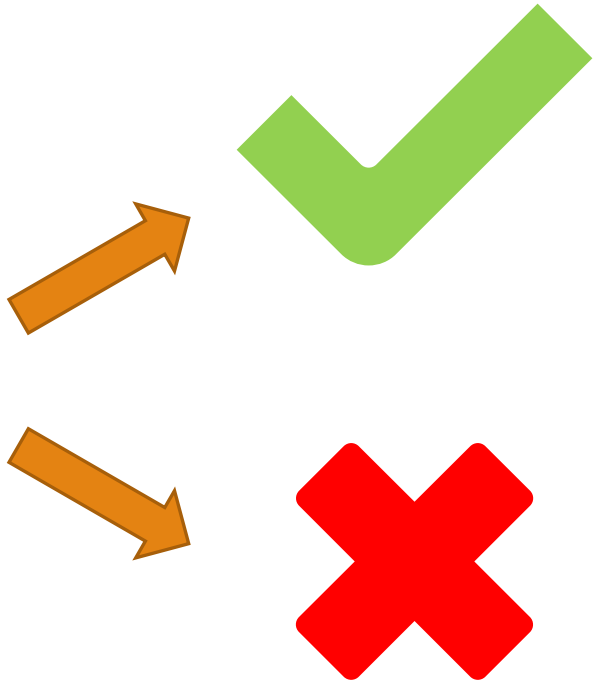
Evaluador

```
from ui.config

scripts = [
    PgmpyScript.PC(),
    PyAgrumScript.TS(),
]

datasets = [
    DirectCausalityDataset.linear(1000),
    ChainedCausalityDataset.linear(100),
]

threshold = 90
```

Framework

Evaluador

```
from ui.config

scripts = [
    PgmpyScript.PC(),
    PyAgrumScript.TS(),
]

datasets = [
    DirectCausalityDataset.linear(1000),
    ChainedCausalityDataset.linear(100),
]

threshold = 90
```



Framework

¡Demo!

Framework

Evaluador – Formato tabla

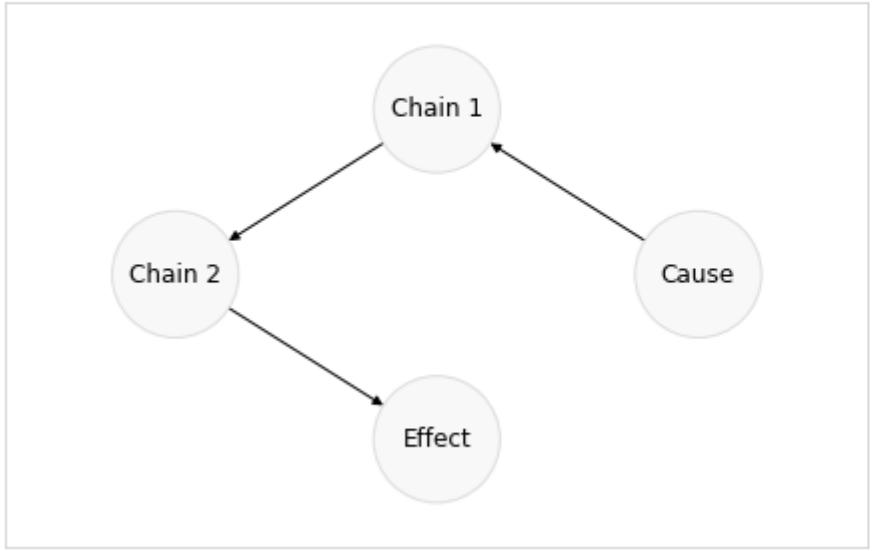
dataset	samples	library	algorithm	found	erroneous	inverted	missing	time
Multiple Causes - Discrete	1000	Pgmpy	PC	100%	0	0	0	64ms
Multiple Causes - Discrete	1000	Pgmpy	GES	0%	0	2	0	220ms
Multiple Causes - Discrete	1000	pyAgrum	GES	0%	0	2	0	11ms
Multiple Causes - Discrete	1000	pyAgrum	TS	0%	0	2	0	14ms
Direct Causality - Linear	1000	Pgmpy	PC	0%	0	1	0	37ms
Direct Causality - Linear	1000	Pgmpy	GES	100%	0	0	0	200ms
Direct Causality - Linear	1000	pyAgrum	GES	100%	0	0	0	8ms
Direct Causality - Linear	1000	pyAgrum	TS	100%	0	0	0	14ms
Chained Causality - Linear	100	Pgmpy	PC	0%	0	0	3	29ms
Chained Causality - Linear	100	Pgmpy	GES	19%	2	0	2	458ms
Chained Causality - Linear	100	pyAgrum	GES	50%	1	0	1	212ms
Chained Causality - Linear	100	pyAgrum	TS	25%	1	1	1	271ms

Framework

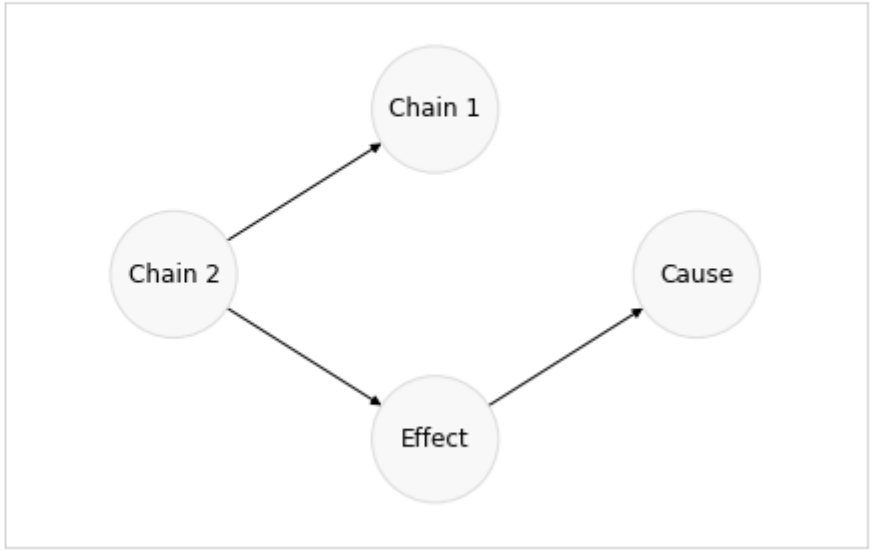
Evaluador – Formato grafo

```
Chained Causality - Linear → TS (pyAgrum): 25%
```

Original graph



Learned graph



Conclusión

Algoritmos

Conclusión

Algoritmos

- **Específico** al tipo de relaciones
-
-
-

Conclusión

Algoritmos

- **Específico** al tipo de relaciones
- **Repetibilidad** no garantizada
-
-

Conclusión

Algoritmos

- **Específico** al tipo de relaciones
- **Repetibilidad** no garantizada
- **Tiempo de ejecución** muy variable

Conclusión

Algoritmos

- **Específico** al tipo de relaciones
- **Repetibilidad** no garantizada
- **Tiempo de ejecución** muy variable
- **Recursos materiales** rápidamente limitantes

Conclusión

Desarrollos futuros

Desarrollos futuros

Media de ejecuciones

Conclusión

Desarrollos futuros

Media de ejecuciones

Visualizar desfases temporales

¡Gracias!