



**Tu universidad  
de postgrado**  
Your university  
for graduate  
studies



**Tu universidad  
para una formación  
permanente**  
Your lifelong  
learning university



**Tu universidad  
para una enseñanza  
innovadora**  
Your innovative  
education university



**La universidad  
para tu futuro**  
The university  
for your future

UNIVERSIDAD  
INTERNACIONAL  
DE ANDALUCÍA



**BioSiP**

Biomedical Signal Processing, Computational  
Intelligence and Communications Security



UNIVERSIDAD  
DE GRANADA



**DaSCI**

Instituto Andaluz Interuniversitario en  
Data Science and Computational Intelligence

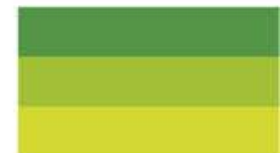
un  
i Universidad  
Internacional  
de Andalucía  
A

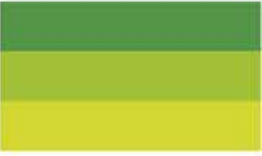


UNIVERSIDAD  
DE MÁLAGA

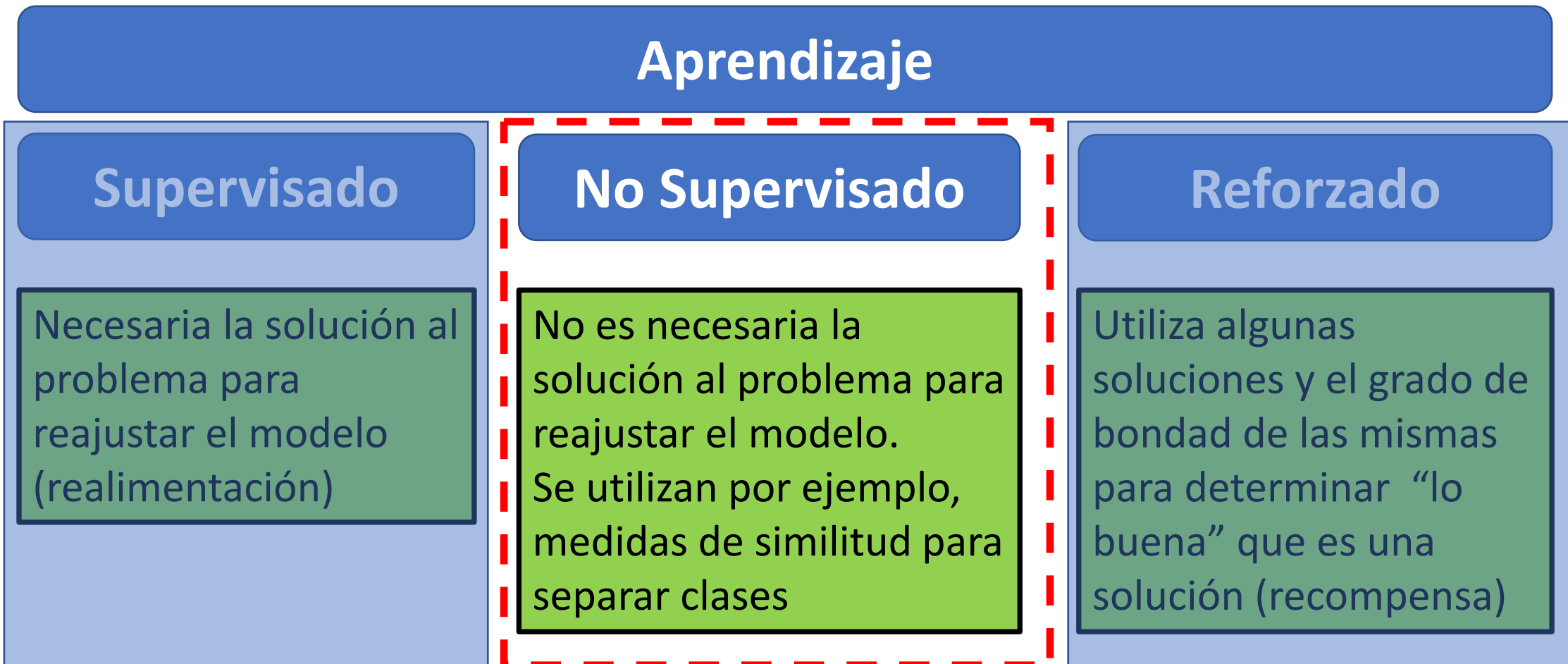
# Introducción práctica a la Inteligencia Artificial y al Deep Learning

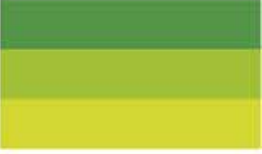
Aprendizaje no supervisado





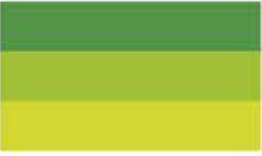
# Paradigmas de aprendizaje automático





# Aprendizaje no supervisado

- Algoritmos de Inteligencia Artificial para identificar patrones en conjuntos de datos no etiquetados y sin conocimiento previo.
- Son útiles para encontrar características que pueden ser útiles para la categorización (agrupamiento)
- **Es más fácil obtener datos no etiquetados, dado que el etiquetado de las muestras suele ser un proceso manual (muestra a muestra).**



## Principales aplicaciones

- **Agrupamiento** de datos (clustering) de acuerdo a su similitud (medida de similitud)
- **Detección de anomalías**: desviaciones con respecto a un comportamiento definido como normal
- **Compresión**: todos los puntos en un cluster pasan a ser representados por el centróide de dicho cluster
- **Modelos de variables latentes**: compresión, reducción de la dimensionalidad, eliminación de ruido

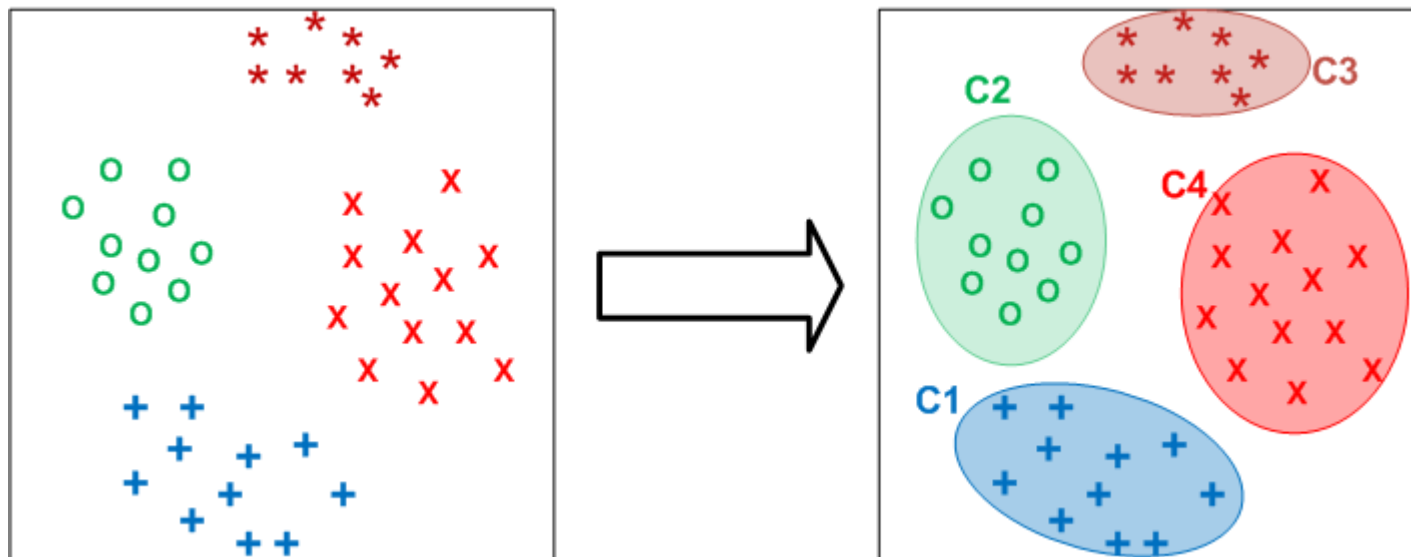
## Principal Inconveniente

- No se puede tener certeza acerca de la precisión, dado que no disponemos de etiquetas
- Requiere de una interpretación a posteriori para identificar los grupos

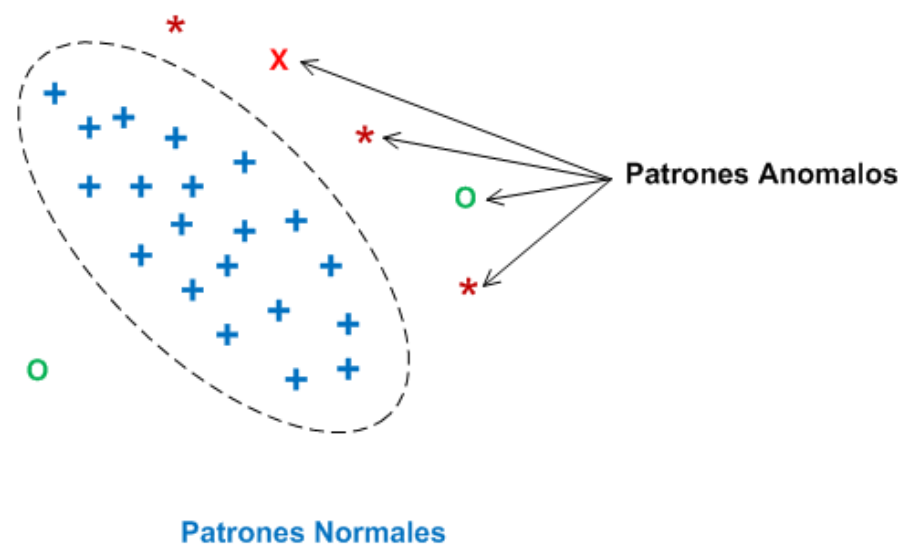


# Algoritmos de agrupamiento (clustering)

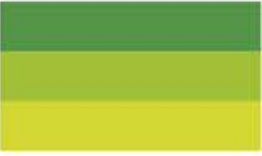
## Ejemplo de agrupamiento (clustering)



## Detección de anomalías



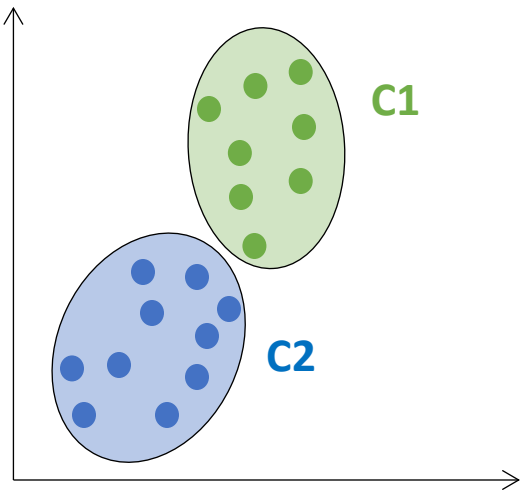




# Algoritmos de agrupamiento (clustering)

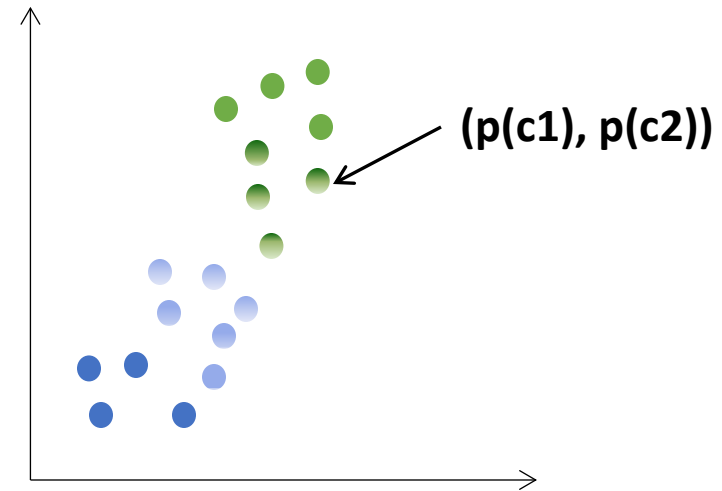
## Tipos de clustering

### Hard-clustering

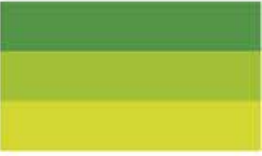


Cada muestra se asigna a un cluster

### Soft-clustering

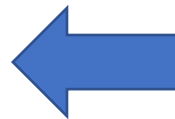


A cada muestra se le asigna una probabilidad de pertenencia a un cluster



## Diagrama de Voronoi

El diagrama de Voronoi de un conjunto de puntos en el plano, es la división de dicho plano en regiones, de tal forma que cada región contenga aquellos puntos que son más parecidos entre sí.



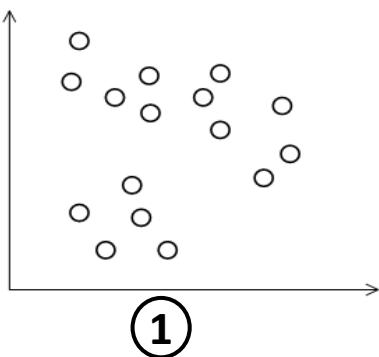
Dicha división depende de la medida de distancia utilizada.

Es usual utilizar la distancia Euclídea, pero podría utilizarse cualquier métrica

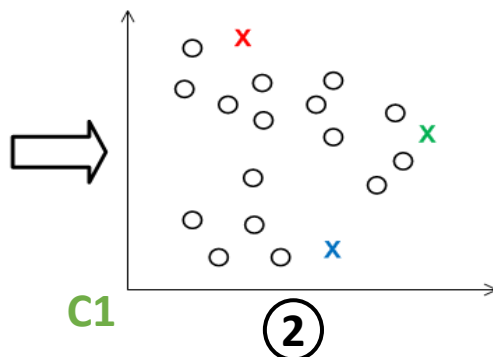


# Hard-Clustering. Algoritmo K-medias

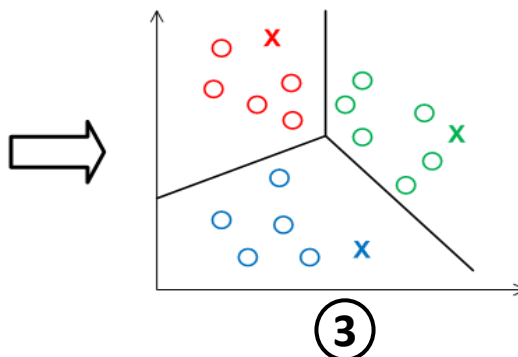
Datos iniciales  
Sin etiquetar



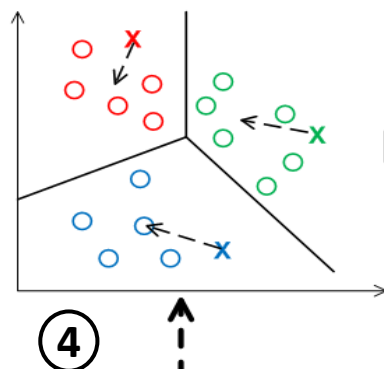
Asignación aleatoria de los  
centroides ( $k=3$ )



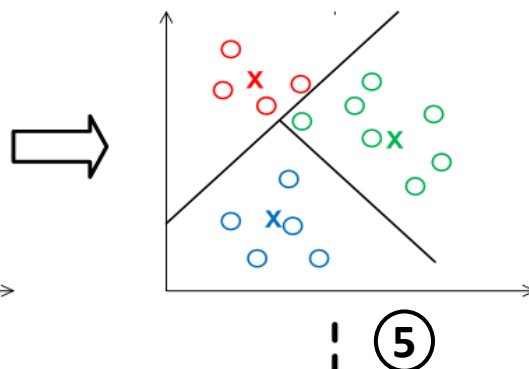
Asignación de cada punto al  
centroide más cercano



Actualización de los centroides



Reasignación de las muestras al  
centroide más cercano

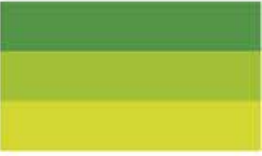


Repetir hasta alcanzar el  
criterio de parada

## Criterios de parada

1. Los centroides dejan de cambiar
2. Los puntos dejan de cambiar de *cluster*
3. Se alcanza el límite impuesto en el número de iteraciones

Estamos  
(re)calculando  
el diagrama de  
Voronoi !

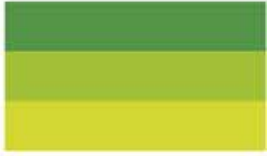


# ¿ Cómo podemos saber el número óptimo de clusters (k) ?

## Método Elbow

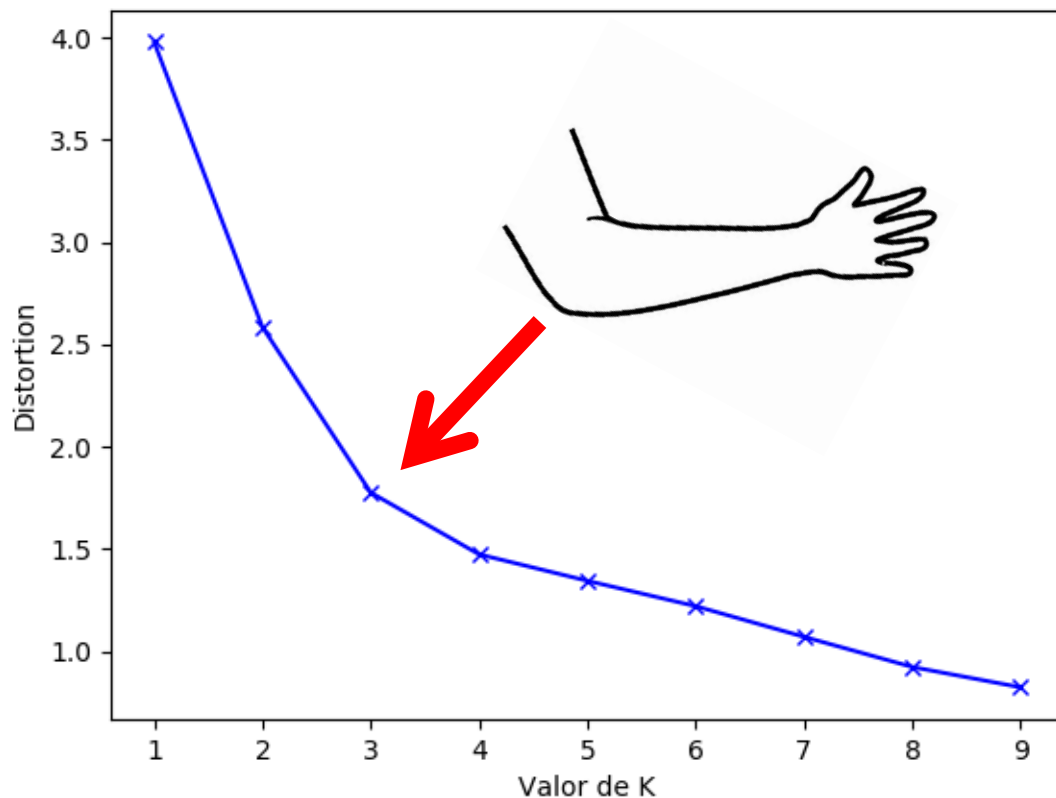
Se puede calcular a partir de dos medidas

1. **Distorsión:** media de las distancias al cuadrado entre los centros de los diferentes clusters
2. **Inercia:** suma de las distancias al cuadrado entre cada muestra y el centro del cluster al que corresponden



# ¿ Cómo podemos saber el número óptimo de clusters (k) ?

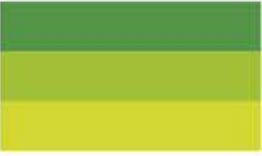
## Método Elbow



Según el método Elbow, el número óptimo de clusters en este ejemplo es de 3 ( $k=3$ )

Por tanto, para saber cual es el numero óptimo de clusters, tenemos que agrupar primero los datos para diferentes valores de k, y elegir el mejor



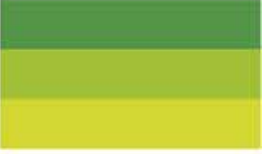


## Soft-Clustering. Algoritmo Fuzzy C-medias

- Se basa en la asignación de un grado de pertenencia de cada muestra a un cluster
- Dicho grado de pertenencia se calcula a partir de la distancia de una muestra al centroide de cada cluster y a la media de las muestras de cada cluster

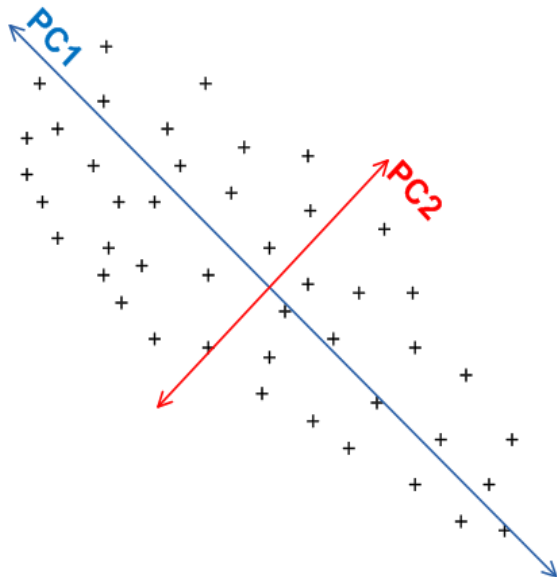
## Principales ventajas

- Se comporta mejor que k-medias para conjuntos de datos superpuestos
- Asigna la pertenencia a cada centro de clúster, por lo que los puntos de datos pueden pertenecer a más de un centro de clúster.

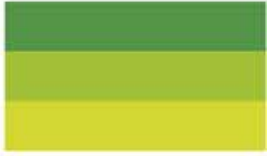


## Proyección lineal. Análisis de Componentes Principales (PCA)

- PCA es una técnica estadística que permite describir un conjunto de datos en términos de nuevas variables no correlacionadas.
- Estas nuevas variables, llamadas componentes principales, explican la varianza de las variables originales, de forma que aquellas componentes que expliquen mayor varianza indicarían las direcciones de máxima variación.

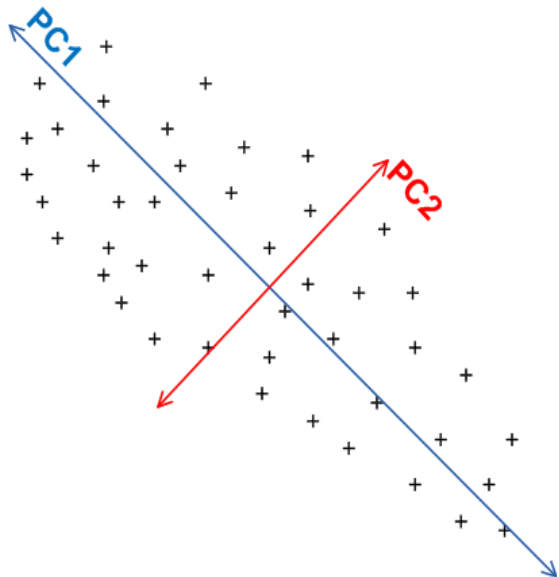


Las componentes principales se ordenan de acuerdo a la varianza que explican



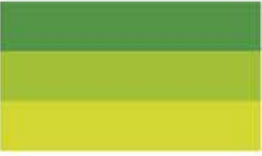
## Proyección lineal. Análisis de Componentes Principales (PCA)

- PCA es una técnica estadística que permite describir un conjunto de datos en términos de nuevas variables no correlacionadas.
- Estas nuevas variables, llamadas componentes principales, explican la varianza de las variables originales, de forma que aquellas componentes que expliquen mayor varianza indicarían las direcciones de máxima variación.



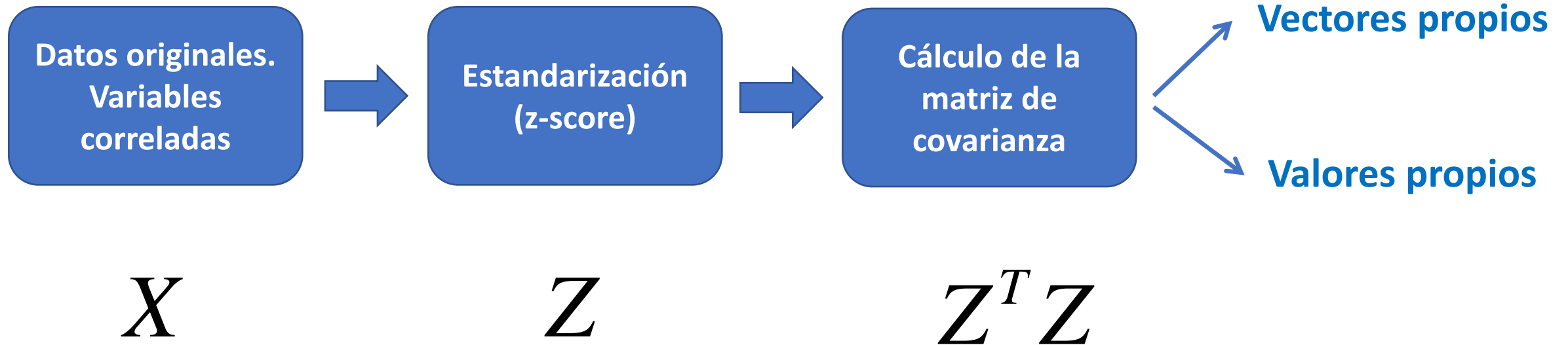
### Proyectando sobre las PCs:

- Eliminamos ruido
- Podemos reducir la dimensionalidad de los datos

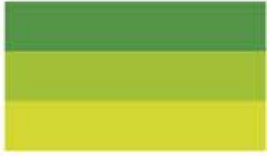


# Proyección lineal. Análisis de Componentes Principales (PCA)

## Algoritmo



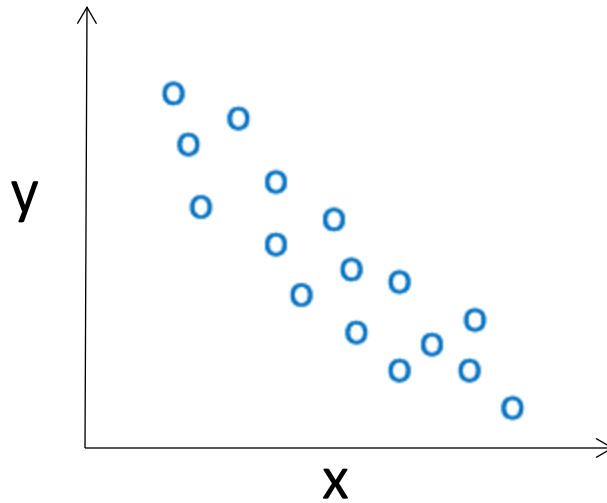




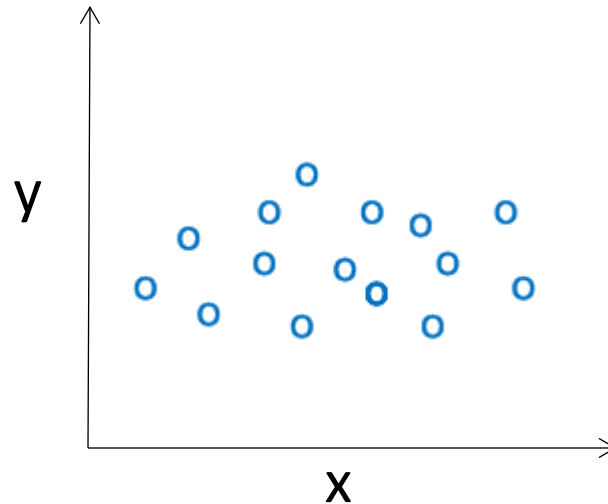
# Proyección lineal. Análisis de Componentes Principales (PCA)

Un poco de álgebra lineal (sólo un poco 😊)

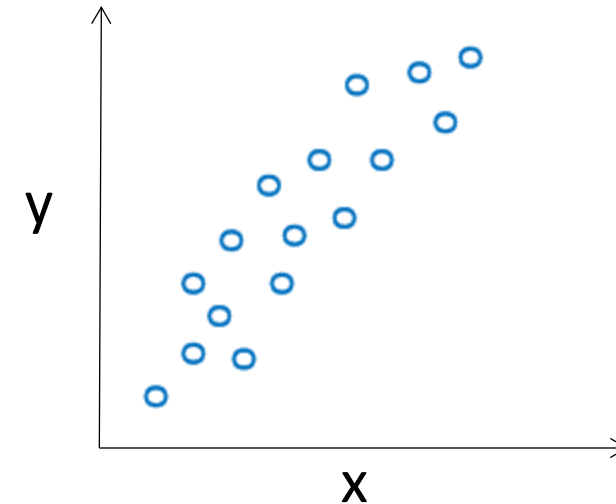
Correlación entre dos variables (x,y)



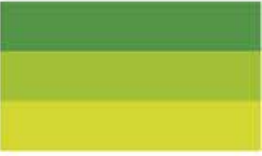
**Negativa**



**~ Nula**



**Positiva**



# Proyección lineal. Análisis de Componentes Principales (PCA)

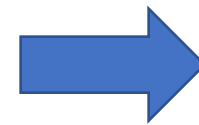
Un poco de álgebra lineal (sólo un poco 😊)

Sea un conjunto de datos

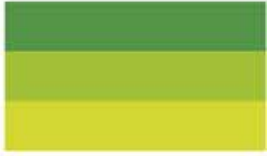
x	y	
1	5	→ Muestra 1
7	9	
.	.	
.	.	
10	4	→ Muestra n

Matriz de covarianza

$$\Sigma = \begin{bmatrix} \text{var}(x) & \text{cov}(x,y) \\ \text{cov}(y,x) & \text{var}(y) \end{bmatrix}$$



**$\text{cov}(x,y) = \text{cov}(y,x)$**   
**La matriz es simétrica!**

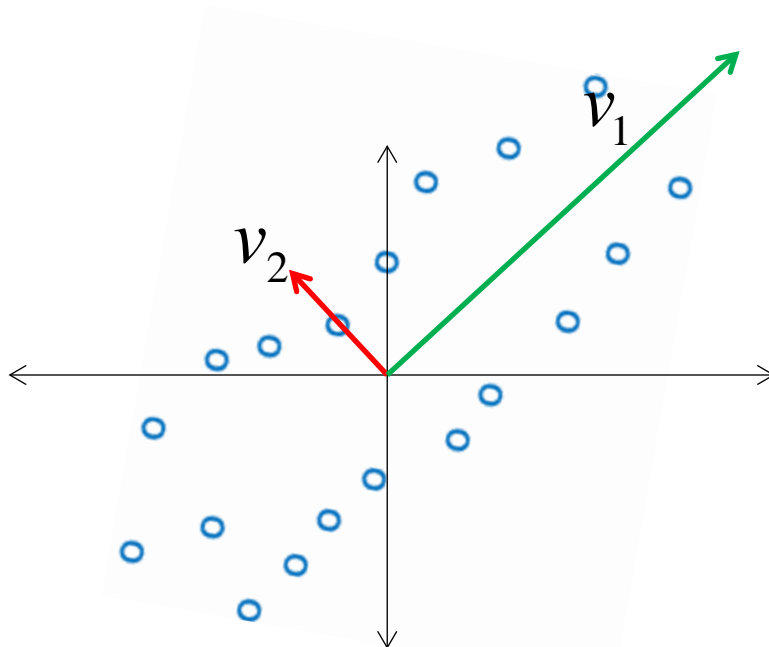


# Proyección lineal. Análisis de Componentes Principales (PCA)

Un poco de álgebra lineal (sólo un poco 😊)

Vectores propios y valores propios

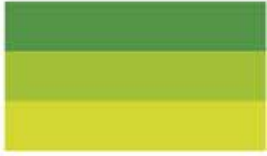
Dimension = 2 → 2 vectores propios



$$\Sigma v_1 = \lambda_1 v_1$$

$$\Sigma v_2 = \lambda_2 v_2$$

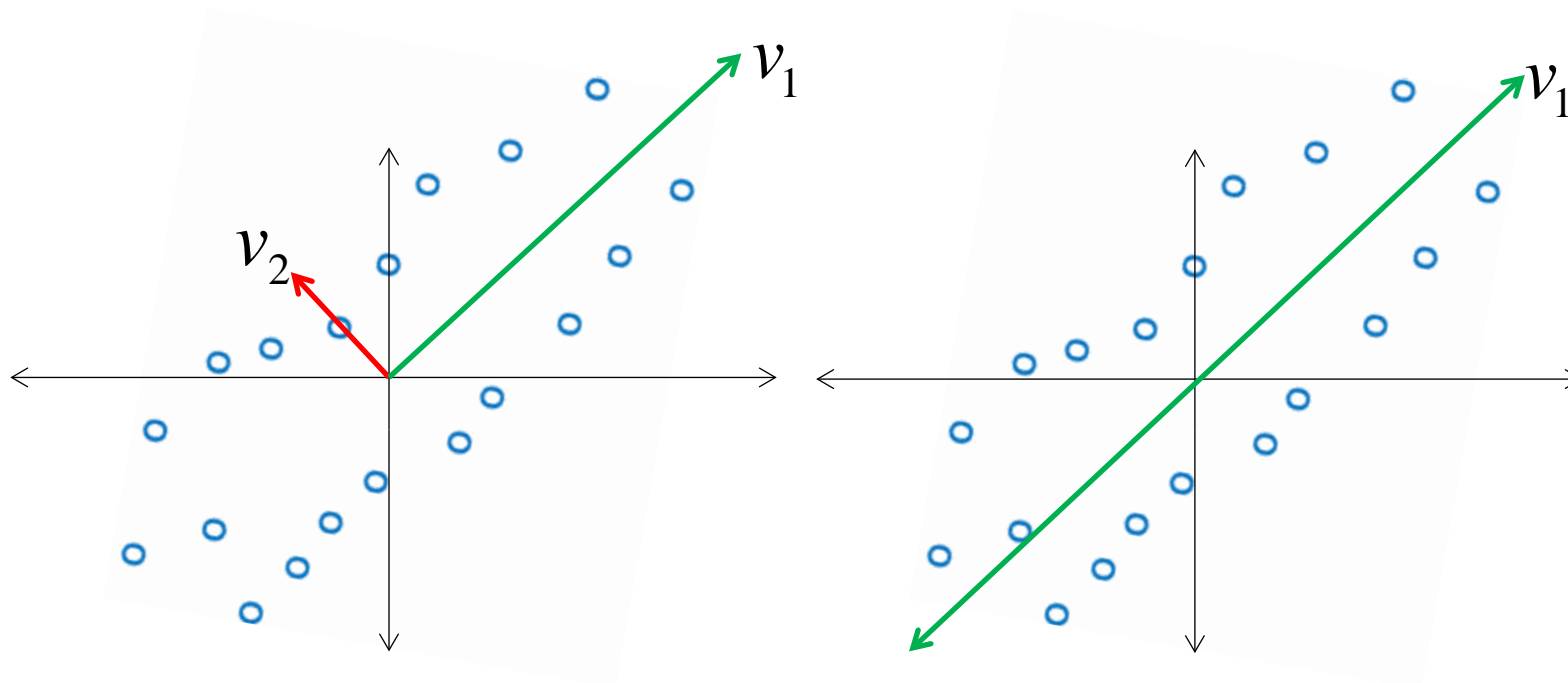
Dirección      Magnitud



# Proyección lineal. Análisis de Componentes Principales (PCA)

Un poco de álgebra lineal (sólo un poco 😊)

$V_1$  es el vector propio con un autovalor mayor ( $\lambda_1$ ) → La variación de los datos es mayor en esa dirección

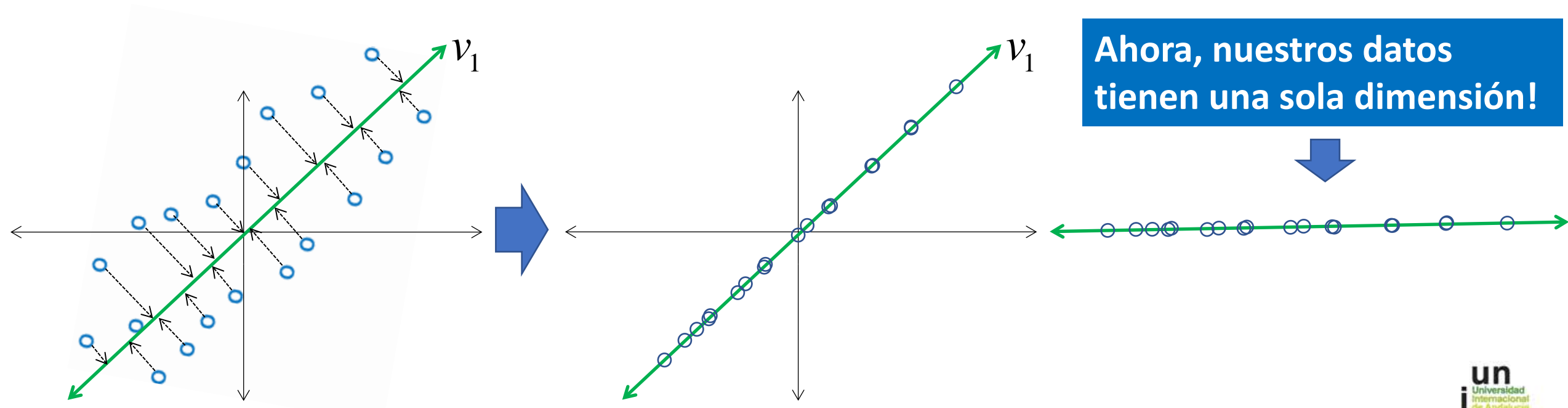


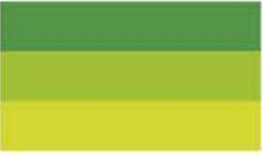


# Proyección lineal. Análisis de Componentes Principales (PCA)

Un poco de álgebra lineal (sólo un poco 😊)

Si proyectamos todos los datos sobre la dirección de máxima variación



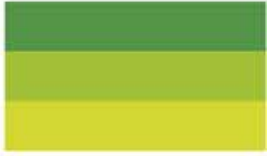


# Proyección lineal. Análisis de Componentes Principales (PCA)

Un poco de álgebra lineal (sólo un poco 😊 )

Vectores propios y valores propios

- Los vectores propios de una matriz simétrica son ortogonales entre sí  
→ Indican la dirección de variación
  - Los valores propios indican la longitud de los vectores propios → medida de la dispersión (varianza) de los datos en la dirección del vector propio correspondiente
- 
- La proyección de los datos sobre los vectores propios son las componentes principales (ortogonales → decorreladas)



# Proyección lineal. Análisis de Componentes Principales (PCA)

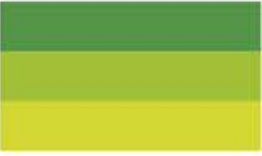
## Aplicaciones principales

- Extracción de características
- Selección de características
- Compresión de datos
- Análisis exploratorio → Contribución de cada variable a las componentes principales (loadings)

$$\text{loadings} = \text{vectores\_propios} * \sqrt{\text{valores\_propios}}$$

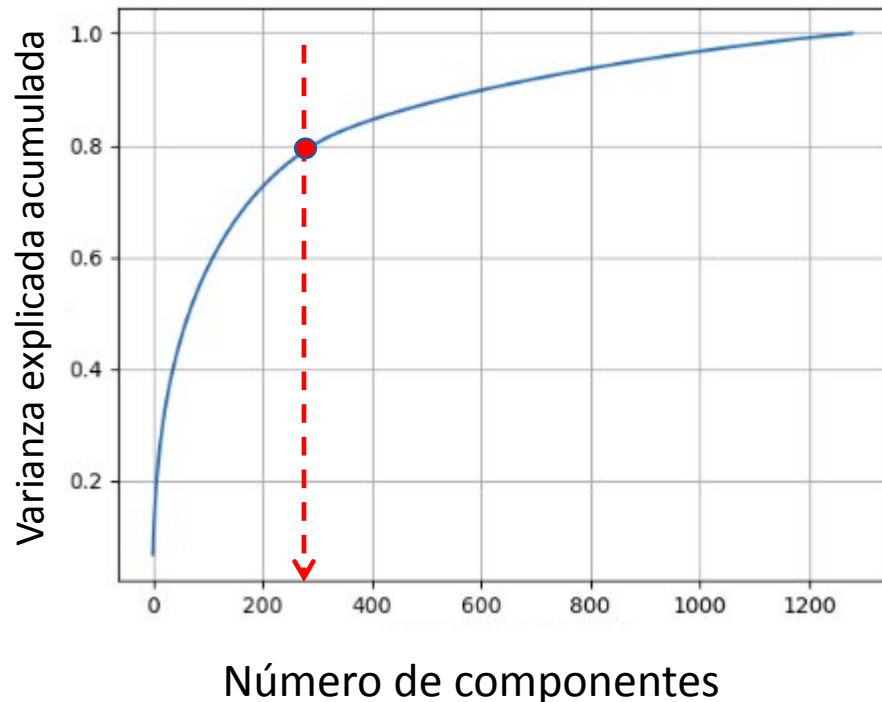
Las cargas (loadings) indican la correlación entre cada variable original y las componentes





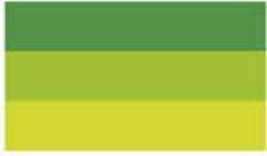
# Proyección lineal. Análisis de Componentes Principales (PCA)

## Selección del número de componentes óptimo



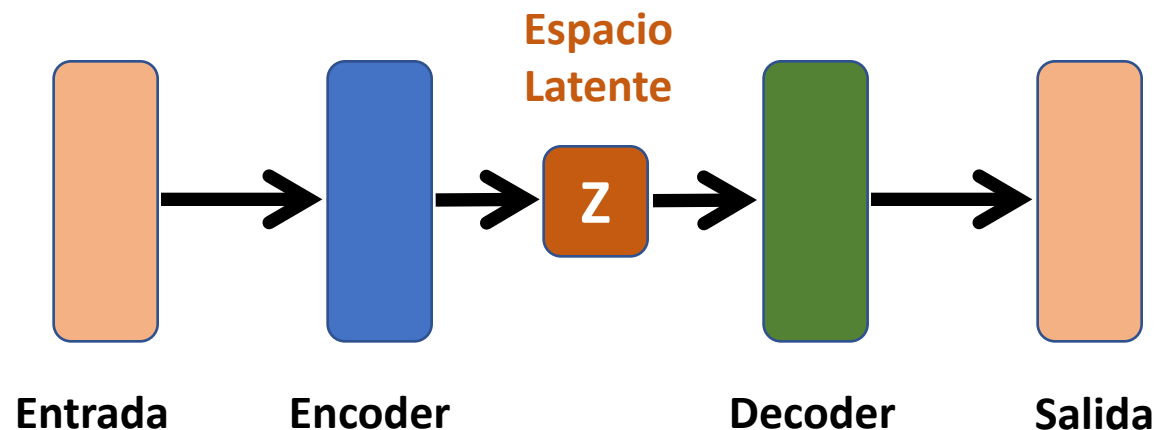
Es habitual seleccionar el número de componentes que expliquen alrededor del 80% de la varianza

$$\% \text{ var}(\text{PC}_i) = \frac{\lambda_i}{\lambda_1 + \lambda_2 + \dots + \lambda_n}$$



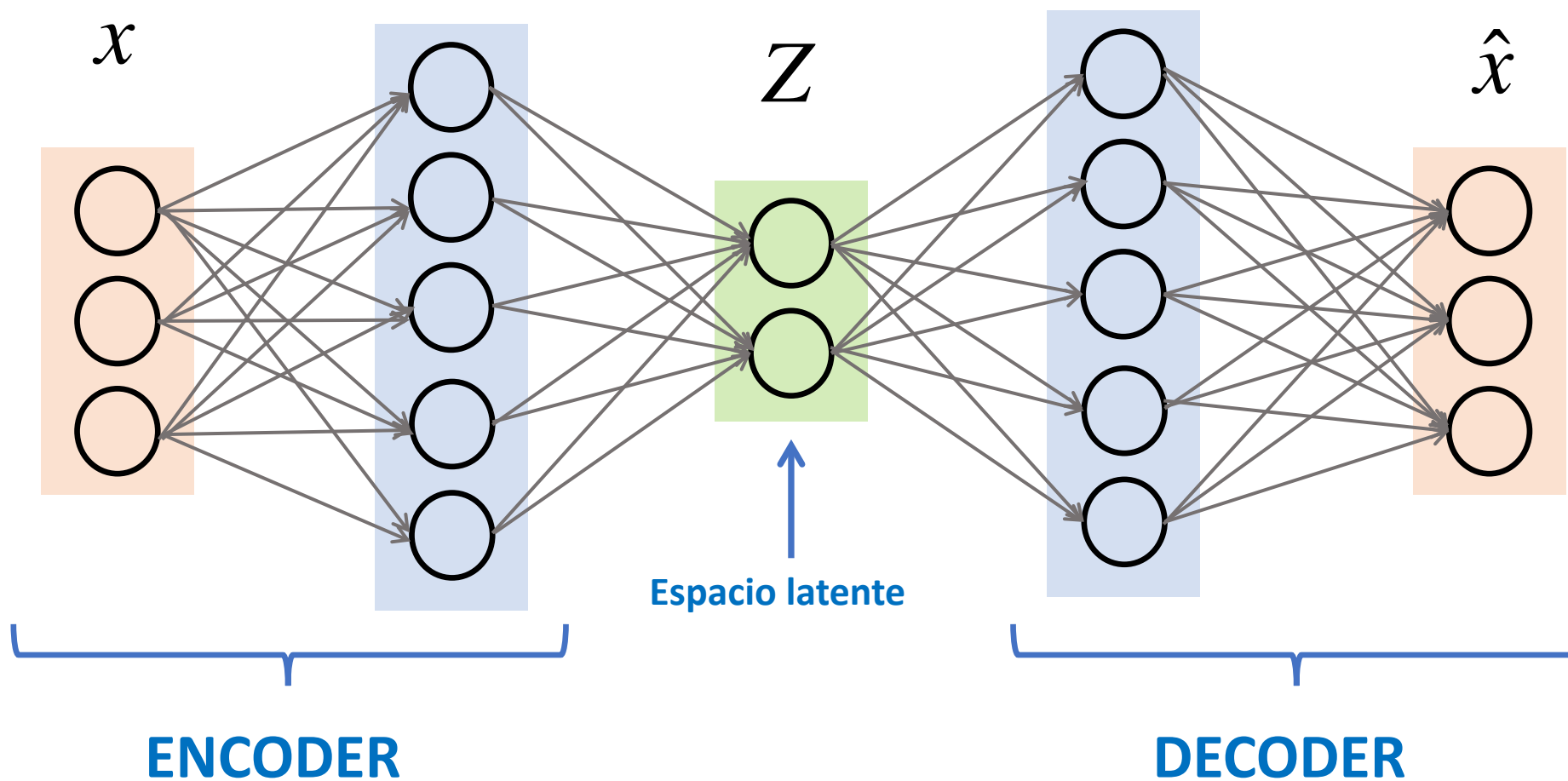
## Proyección no lineal. Autoencoders

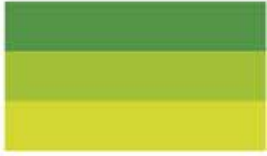
- Un autoencoder es una red neuronal cuya salida es del mismo tamaño que la entrada y que aprende a reconstruir las muestras de entrada. Por tanto, NO hace falta etiquetas para entrenarla → Aprendizaje NO SUPERVISADO
- Consta de dos partes: encoder y decoder, con un cuello de botella entre ellas
- El conjunto de muestras que se generan en este cuello de botella forman el espacio latente → Representación compacta (menor dimensión) de las muestras de entrada



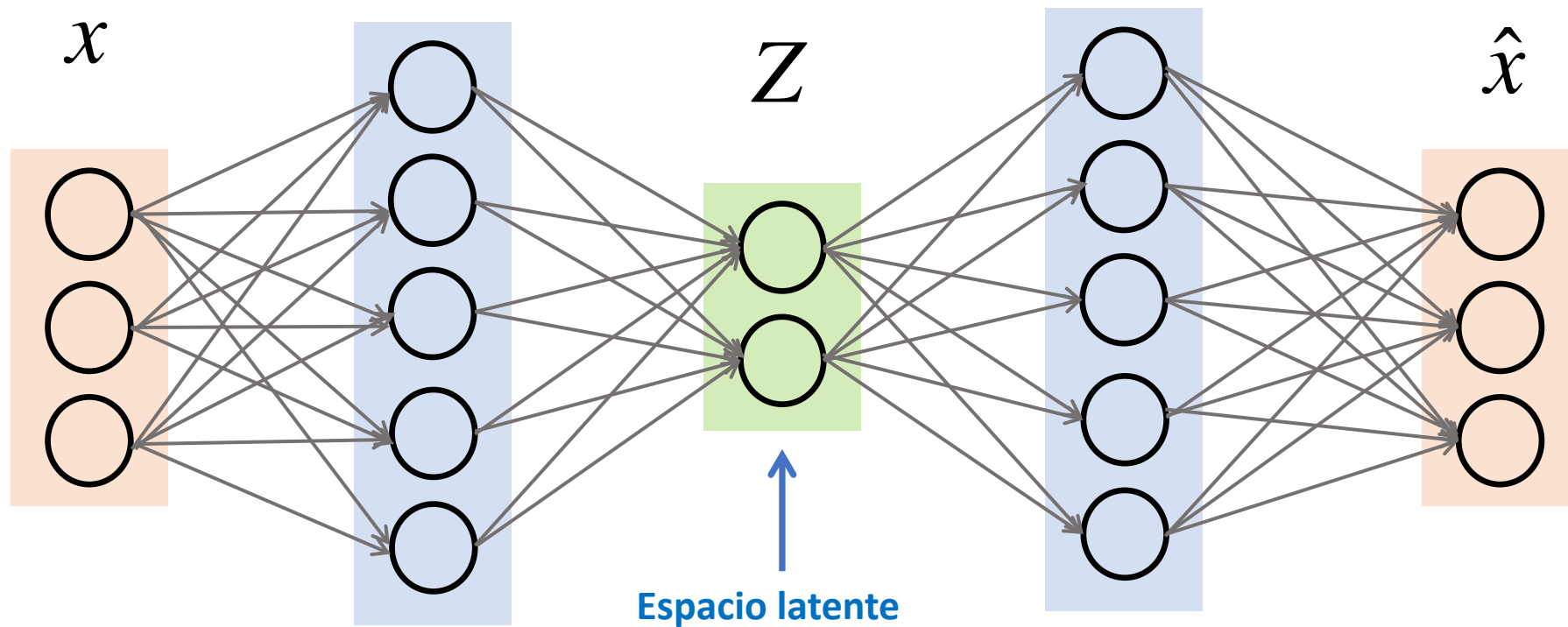


## Proyección no lineal. Autoencoders



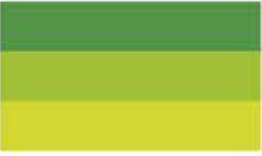


## Proyección no lineal. Autoencoders



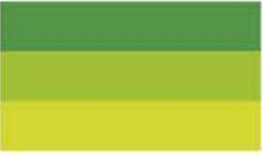
$$loss = MSE(x, \hat{x}) \rightarrow$$

Minimización del error de  
reconstrucción



## Proyección no lineal. Autoencoders

- Una vez entrenado, reconstruye las muestras del conjunto de entrenamiento
- La salida de la capa Z es una representación compacta (de baja dimensión) de una determinada muestra a partir de la cual, el autoencoder es capaz de reconstruir
- Un autoencoder con una sola capa en el encoder y decoder y que usa activaciones lineales, es equivalente a PCA.



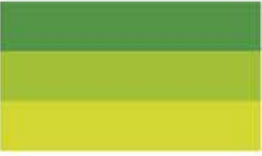
## Proyección no lineal. Autoencoders

Un autoencoder se entrena de forma no supervisada. Pero...

- ¿Qué ocurre si utilizo una entrada que no está en el conjunto de entrenamiento? → ¿Generalización ?
- Un autoencoder SOLO reconstruye las muestras con las que ha sido entrenado. El espacio latente generado es discontinuo
- No obstante, para muestras lo suficientemente parecidas a las de entrenamiento, la reconstrucción puede ser razonable

¿Solución a estos problemas?

**\*\* MODELOS GENERATIVOS \*\***



# Prácticas: Aprendizaje no supervisado

PROGRAMMING

