



Elementos de Machine Learning

Diplomado de Análisis Computacional Estadístico de Datos con Python

Impartido por: Ronaldo Canizales

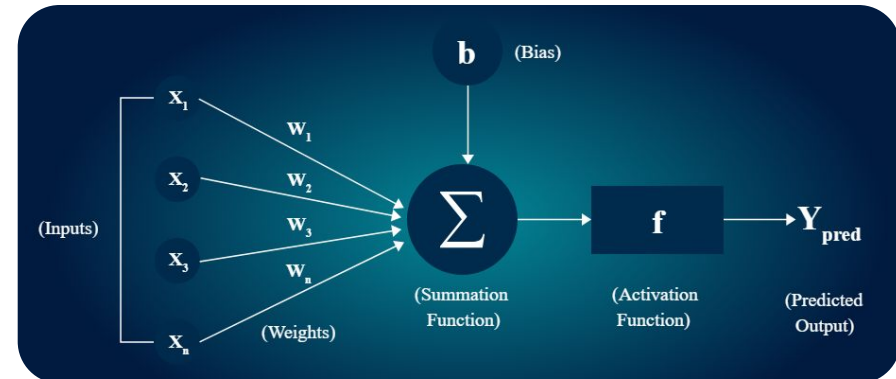
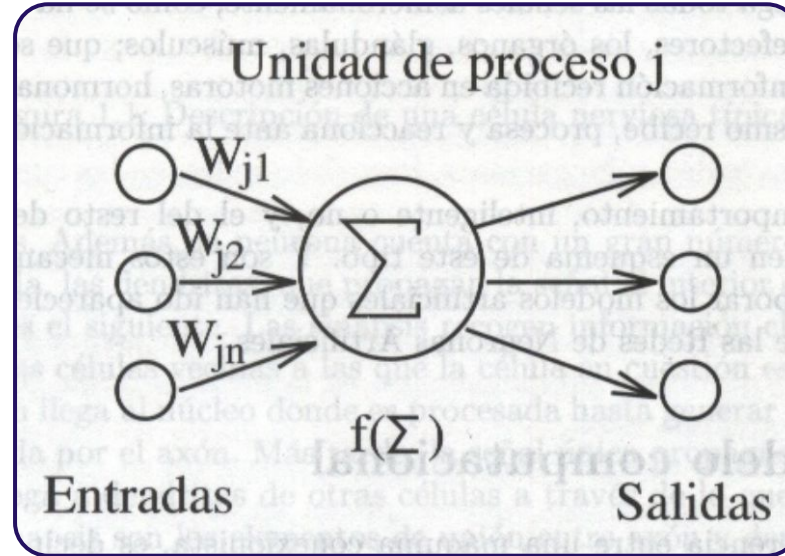
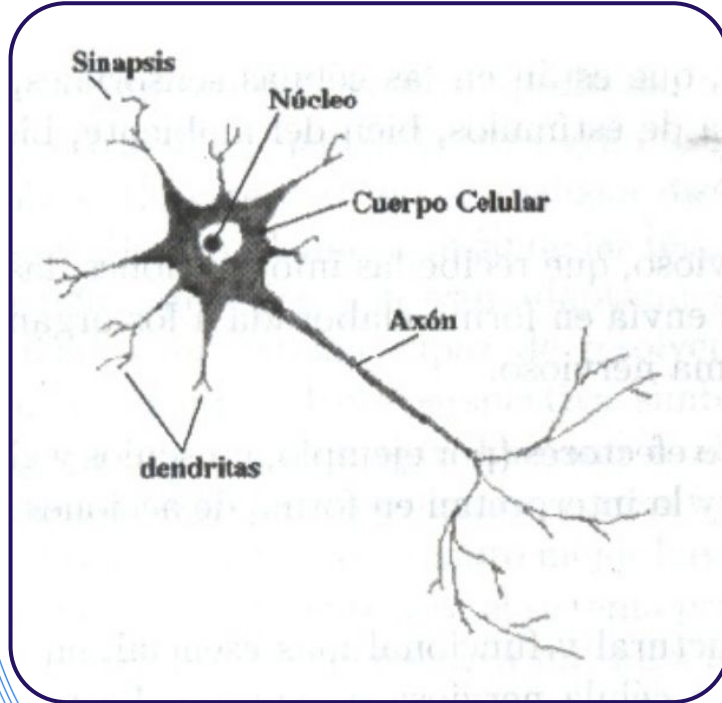


Bloque A

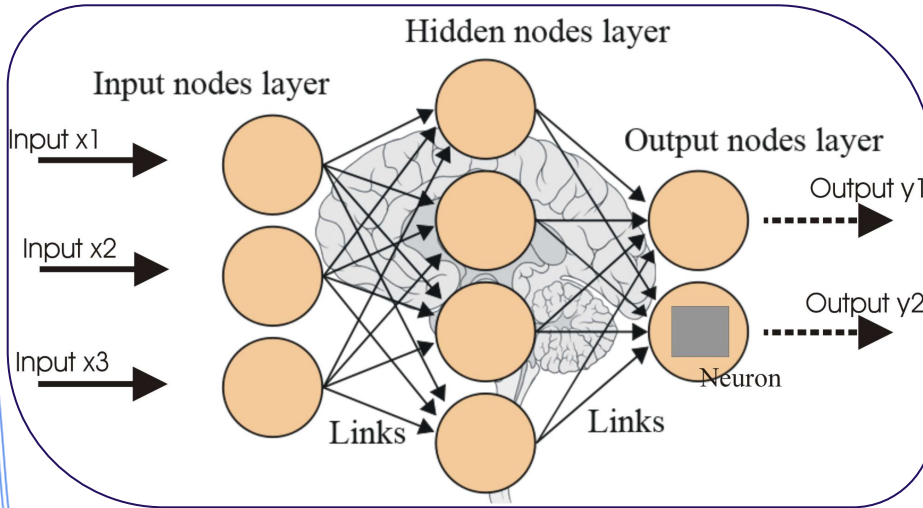
Desmitificando el
Deep Learning:
mi primer red neuronal

Esquema neurona artificial

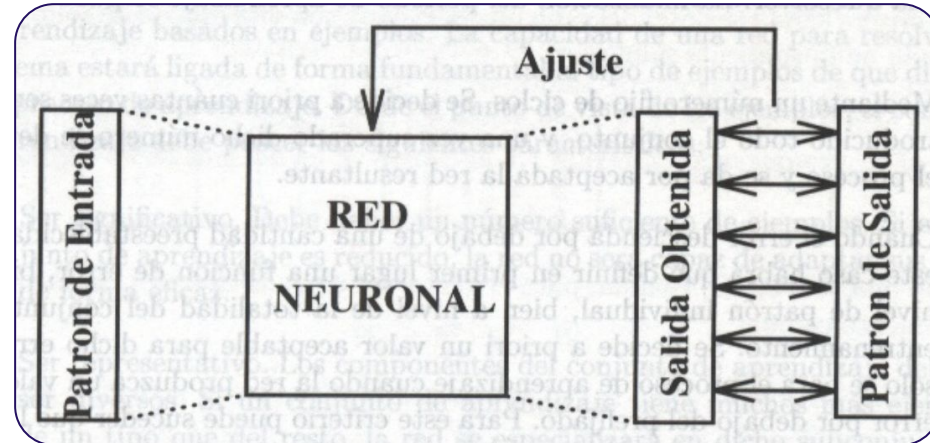
Neurona biológica típica



Esquema de una red de 3 capas totalmente interconectadas



Aprendizaje supervisado



A mostly complete chart of

Neural Networks

©2016 Fjodor van Veen - asimovinstitute.org

Backfed Input Cell

Input Cell

Noisy Input Cell

Hidden Cell

Probabilistic Hidden Cell

Spiking Hidden Cell

Output Cell

Match Input Output Cell

Recurrent Cell

Memory Cell

Different Memory Cell

Kernel

Convolution or Pool

Perceptron (P)



Feed Forward (FF)



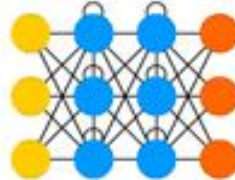
Radial Basis Network (RBF)



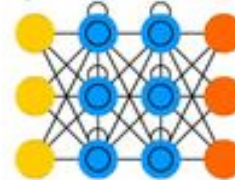
Deep Feed Forward (DFF)



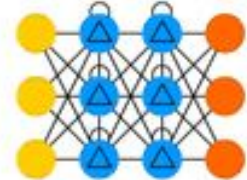
Recurrent Neural Network (RNN)



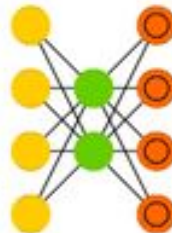
Long / Short Term Memory (LSTM)



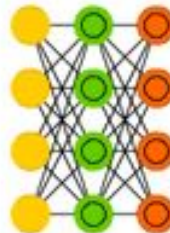
Gated Recurrent Unit (GRU)



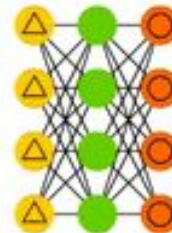
Auto Encoder (AE)



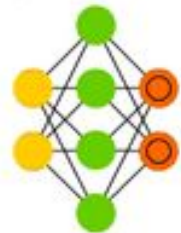
Variational AE (VAE)



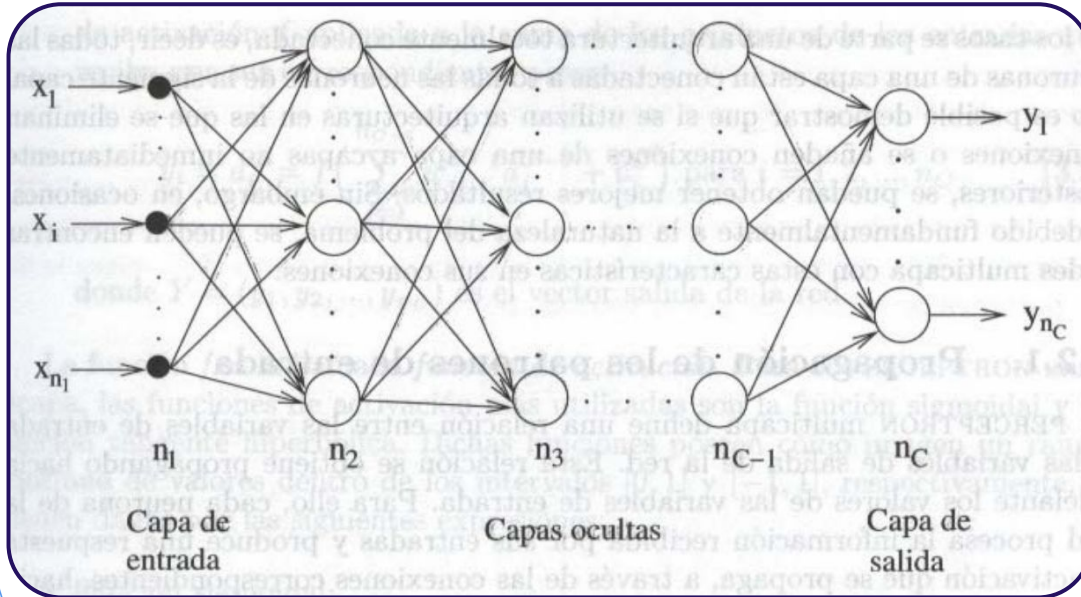
Denoising AE (DAE)



Sparse AE (SAE)



Arquitectura de un perceptrón multicapa

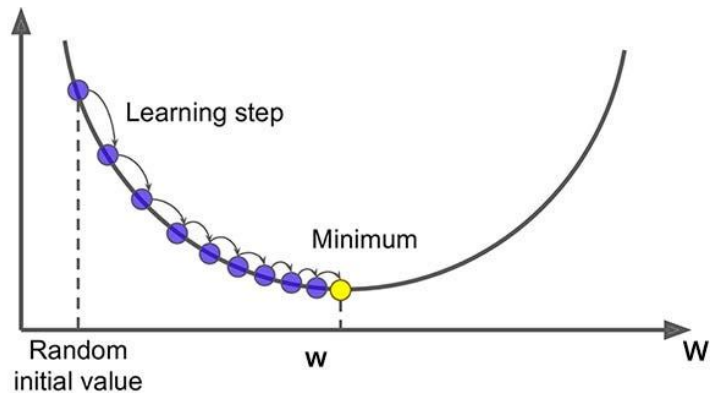


- **Capa de normalización.**
- **Capa de entrada.**
- **Capas ocultas (o de aprendizaje).**
 - **Cantidad de neuronas.**
 - **Función de activación.**
- **Capa de salida.**
- **Capa de desnormalización.**
- **Capa delimitadora.**

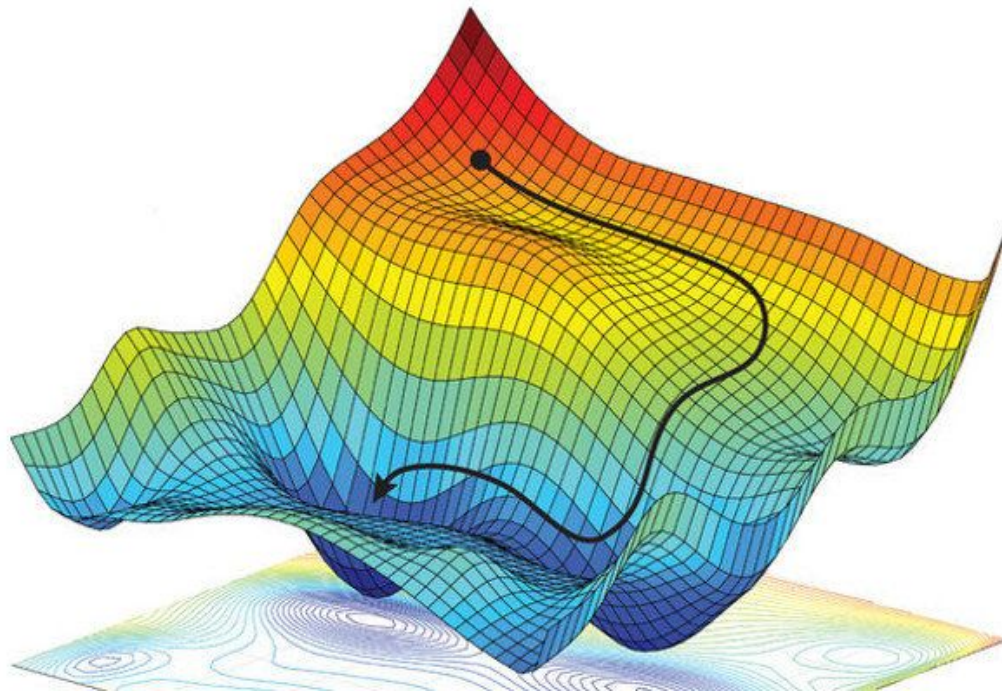
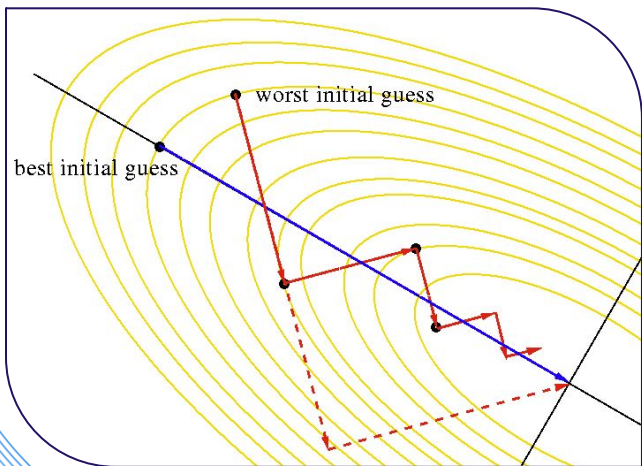
1. **Inicializar** pesos y umbrales **con valores aleatorios**.
2. Introducir **una muestra “n”** del conjunto de entrenamiento, obteniendo la respuesta de la red.
3. **Obtener el error** producido al predecir dicha muestra.
4. Aplicar el Algoritmo de **RetroPropagación** para **modificar los pesos y umbrales** de la red.
5. **Repetir (2), (3) y (4) para todas las muestras** del conjunto de entrenamiento (**iteración / ciclo de aprendizaje**).
6. **Criterios de paro:** error total E de la red ≈ 0 , parámetros de la red estables (norma del gradiente final ≈ 0), cantidad máxima de iteraciones o tiempo máximo transcurrido.
7. Acabar el proceso de aprendizaje y dar como salida la red obtenida.

Proceso de aprendizaje

Cost



Aprendizaje mediante Descenso del Gradiente



Una idea intuitiva del proceso de aprendizaje del Perceptrón Multicapa puede resumirse de la siguiente forma:

Partiendo de un punto aleatorio $W(0)$ del espacio \mathbf{R}^{n_w} , donde n_w es el número de parámetros de la red -pesos más umbrales-, el proceso de aprendizaje desplaza el vector de parámetros $W(n-1)$ en el espacio \mathbf{R}^{n_w} siguiendo la dirección negativa del gradiente del error en dicho punto, alcanzando así un nuevo punto en dicho espacio, $W(n)$, que estará más próximo al mínimo de la función error que el anterior. El proceso continúa hasta que se encuentre un mínimo de la función error E , lo cual sucede cuando $\frac{\partial E}{\partial w} \approx 0$. En este momento, y de acuerdo con la Ecuación (3.10), los parámetros dejan de sufrir cambios significativos de una iteración a otra y el proceso de aprendizaje finaliza.



$$w(n) = w(n-1) - \alpha \frac{\partial e(n)}{\partial w}$$

Activaciones de un Perceptrón Multicapa con dos neuronas de entrada, dos ocultas y una salida

Activaciones de las neuronas de entrada:

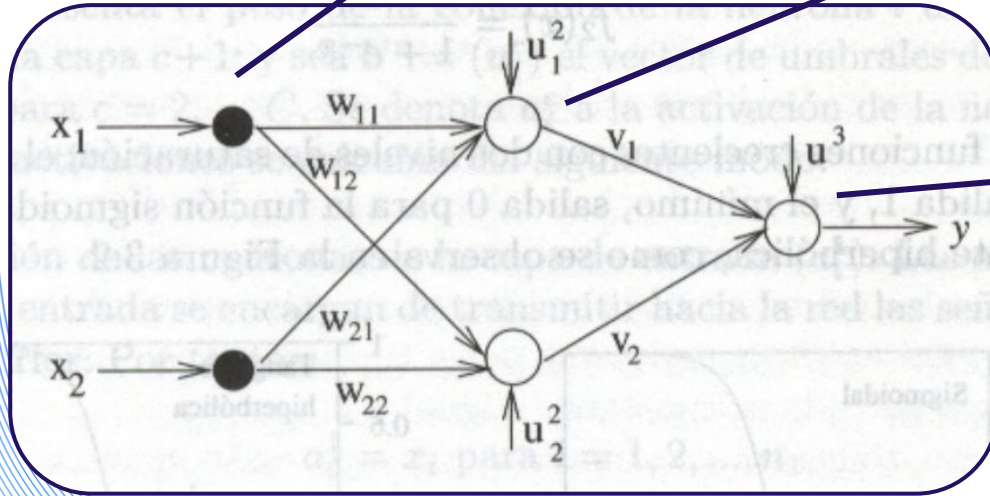
$$a_1^1 = x_1 \text{ y } a_2^1 = x_2$$

Activaciones de las neuronas ocultas:

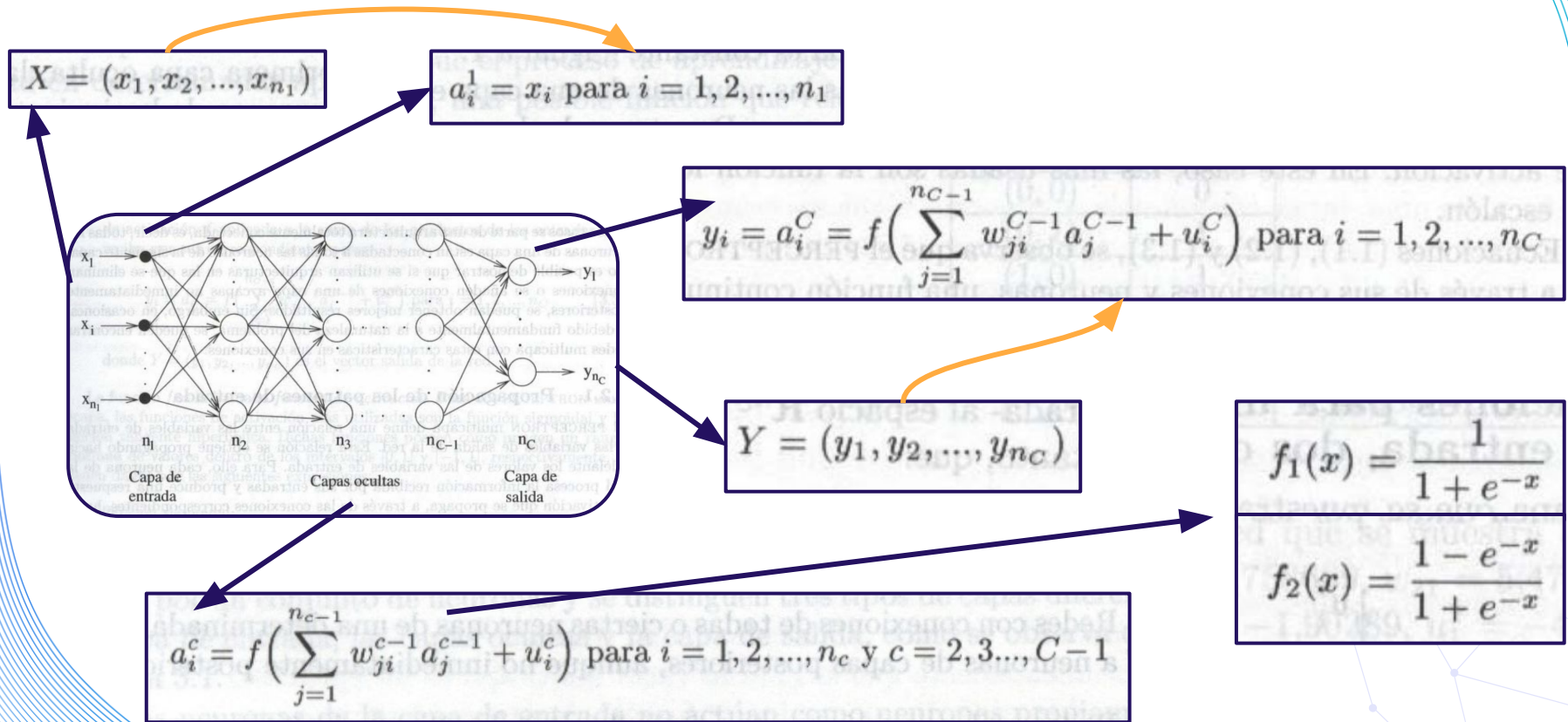
$$a_1^2 = f(w_{11}a_1^1 + w_{21}a_2^1 + u_1^2) \text{ y } a_2^2 = f(w_{12}a_1^1 + w_{22}a_2^1 + u_2^2)$$

Activación de la neurona de salida:

$$y = a_1^3 = f(v_1a_1^2 + v_2a_2^2 + u^3)$$



Propagación de patrones de entrada (generalización)





<https://playground.tensorflow.org/>

Deep Playground

Comprendiendo
las redes de
neuronas artificiales



Bloque B

RNA: “Aproximadores universales”.



¡Gracias por su atención!

Datos de contacto:

rcanizales@uca.edu.sv

rcanizal@colostate.edu

www.linkedin.com/in/ronaldo-canizales/

<https://x.com/ArmandoCodigos>

CREDITS: This presentation template was created by [Slidesgo](#), and includes icons by [Flaticon](#), and infographics & images by [Freepik](#)