

DIPLOMATURA EN

# NUEVAS TECNOLOGÍAS

## MÓDULO 1

# Introducción a la Programación

Docente:

Esp. Ing. Martín Polliotto





## Repaso de Semana anterior

- ✓ Estructuras de control
  - ✓ Condicionales
  - ✓ Repetitivas
- ✓ Estructuras de datos: Arreglos
  - ✓ Creación
  - ✓ Recorrido
  - ✓ Búsqueda
  - ✓ Ordenamiento



DIPLOMATURA EN

**NUEVAS  
TECNOLOGÍAS**



Ministerio de  
**PROMOCIÓN DEL EMPLEO  
Y DE LA ECONOMÍA FAMILIAR**

Ministerio de  
**CIENCIA Y  
TECNOLOGÍA**



Con el apoyo de  
**Oficina de Montevideo**  
Oficina Regional de Genios  
para América Latina y el Caribe

## Contenidos Semana 05:

- 3.1 ¿Qué es un **paradigma**?
- 3.2 **Programación Orientado a Objetos (POO)**
- 3.3 **Clases y Objetos**
- 3.4 Caso de Estudio I
- 3.5 **Objetos y referencias**
- 3.6 **Relaciones** entre clases.
- 3.7 Caso de Estudio II



DIPLOMATURA EN

**NUEVAS  
TECNOLOGÍAS**

SECRETARÍA DE  
EXTENSIÓN  
UNIVERSITARIA  
UTN - FBC

**SEU**

**UTN**  
Facultad Regional Córdoba

Ministerio de  
**PROMOCIÓN DEL EMPLEO  
Y DE LA ECONOMÍA FAMILIAR**

Ministerio de  
**CIENCIA Y  
TECNOLOGÍA**

**CBA**  
ENTRE TODOS

Organización  
de las Naciones Unidas  
para la Educación,  
la Ciencia y la Cultura

Con el apoyo de  
**Oficina de Montevideo**  
Oficina Regional de Genios  
para América Latina y el Caribe

## 3.1 ¿Qué es un paradigma?

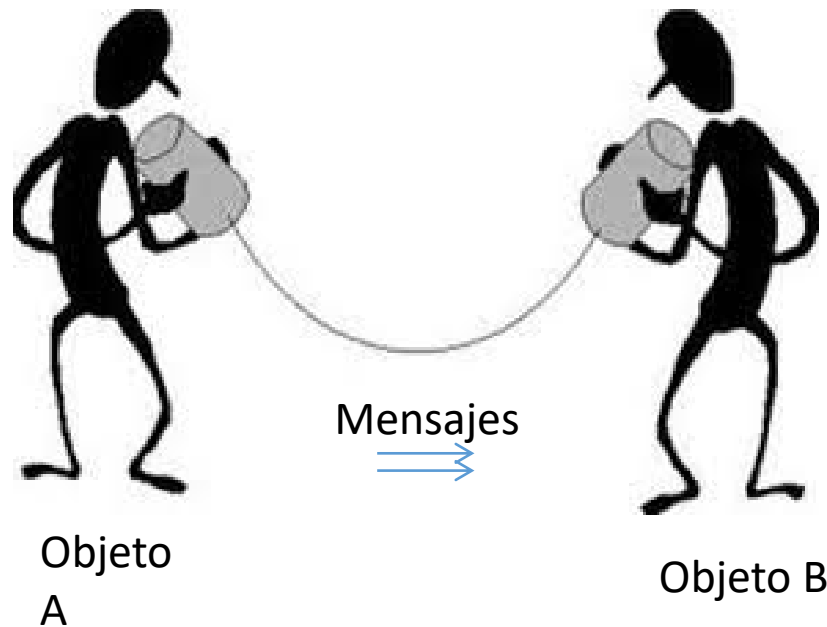


**Día H**, cambio de paradigma para los suecos...

## 3.2 Paradigmas de Programación

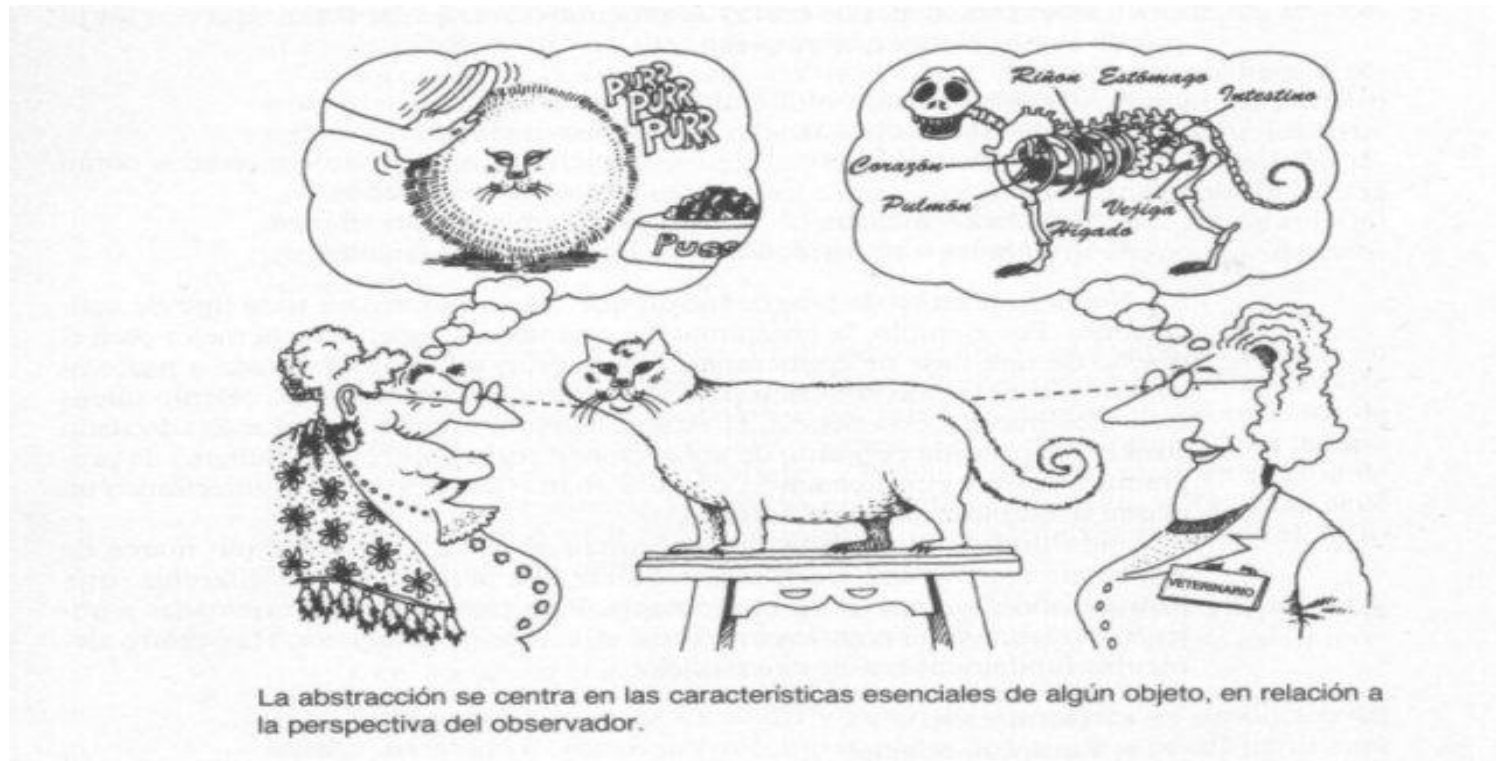


## 3.2 Programación Orientada a Objetos (POO)



En **POO** todo problema se resuelve como un conjunto de **objetos** que **colaboran** entre sí mediante el paso de **mensajes**.

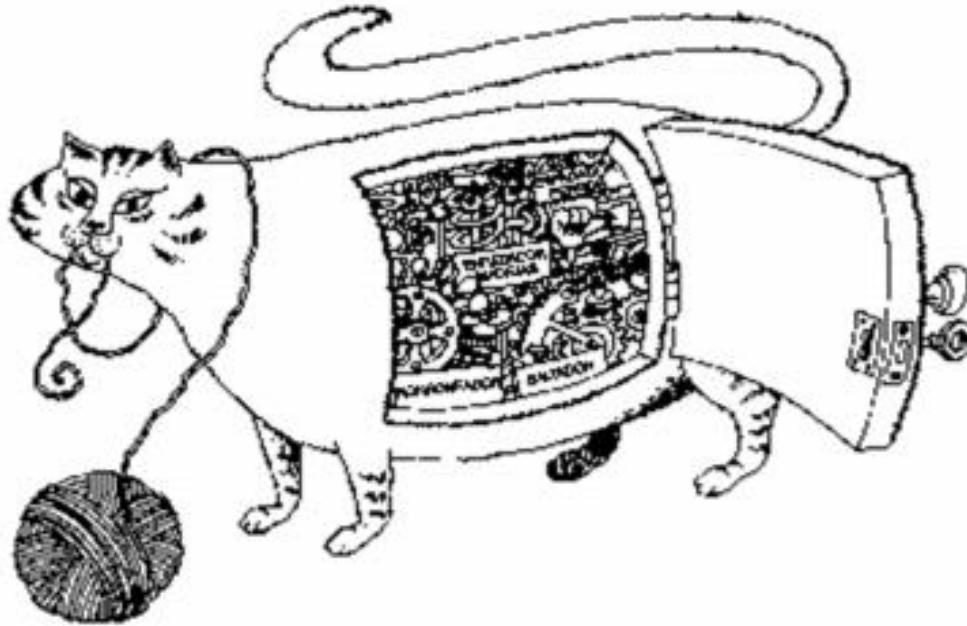
## 3.2 POO – Abstracción



Capacidad para centrarse en **las características o propiedades esenciales** de un objeto, dejando de lado otras no tan relevantes para el dominio del problema



## 3.2 POO – Encapsulamiento



Es el mecanismo mediante el cual ocultamos la implementación de un objeto. En programación se logra mediante el **Principio de Ocultamiento de Información**



## 3.2 POO – Modularidad



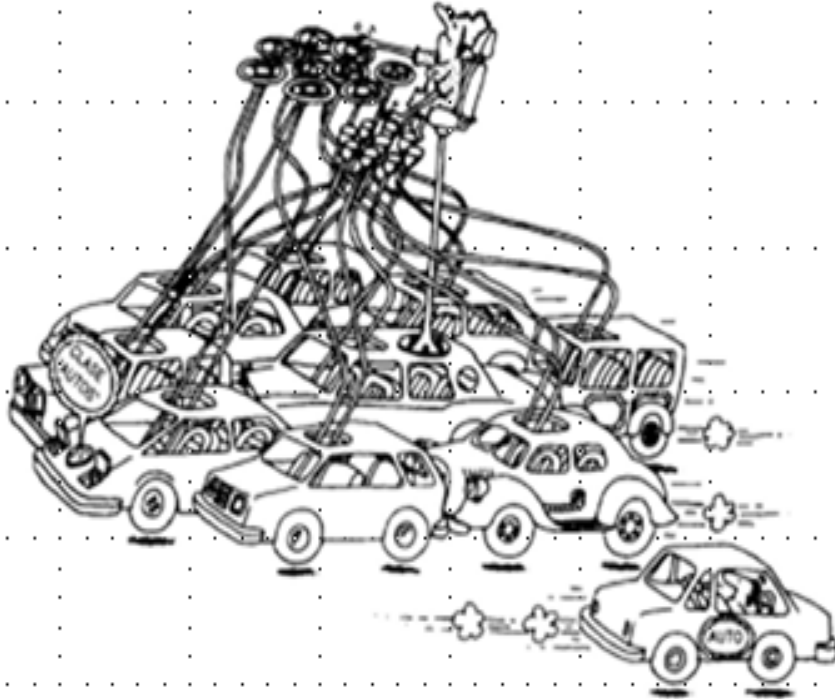
Es la propiedad que tiene un sistema que ha sido descompuesto en un conjunto de **módulos cohesivos y débilmente acoplados**

## 3.2 POO – Jerarquía



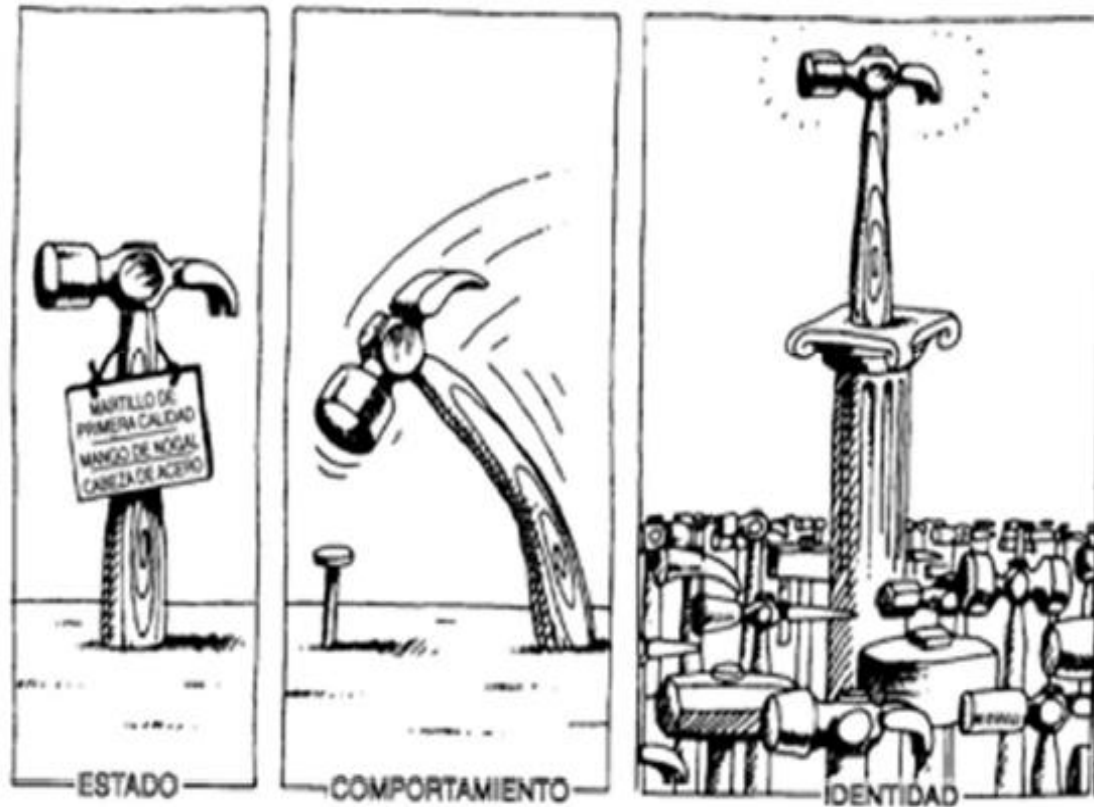
Es una **clasificación** u **ordenación** de abstracciones.  
Nuestros programas se construyen como una **jerarquía de clases**

## 3.3 Clases y Objetos



**Molde, o definición a partir de la cual se modelan las entidades.  
Constituyen nuestro bloque mínimo de código Java**

## 3.3 Clases y Objetos



Es una **instancia** de una clase que tiene: **estado**, **comportamiento** e **identidad**

## 3.3 Clases y Objetos

- Los **atributos no son accesibles** en forma directa desde el exterior de la clase, sino que se usan métodos de la misma para consultar sus valores o modificarlos.
- El **programador** desconoce los detalles de implementación interna de una clase, **solo utiliza los métodos que ésta ofrece.**
- Modificadores de acceso: **public, protected, private y default.**

## 3.3 Clases y Objetos

- Un **constructor** es un comportamiento especial que permite inicializar los atributos de un objeto luego de su creación.
  - Por defecto, con parámetros o por copia.
- Los **accesores** permiten acceder o modificar los atributos de un objeto: **Gets/Sets**

## 3.4 Caso práctico



Vamos a NetBeans  
de nuevo!



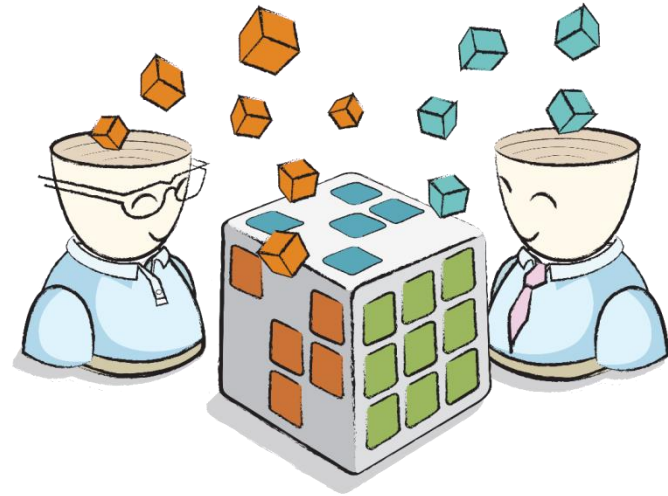
# Caso práctico 01

Se necesita modelar cuentas bancarias de una entidad crediticia, de modo tal que podamos realizar operaciones de depósito y extracción.

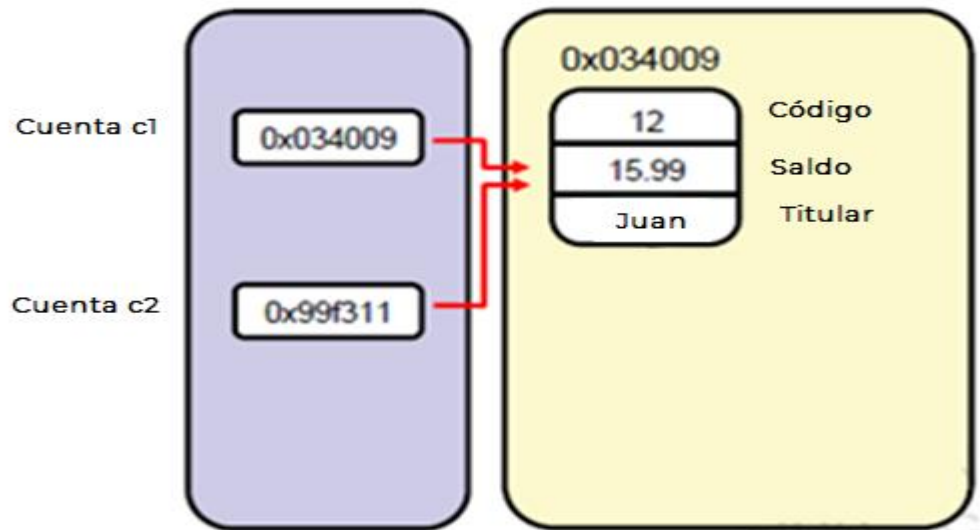
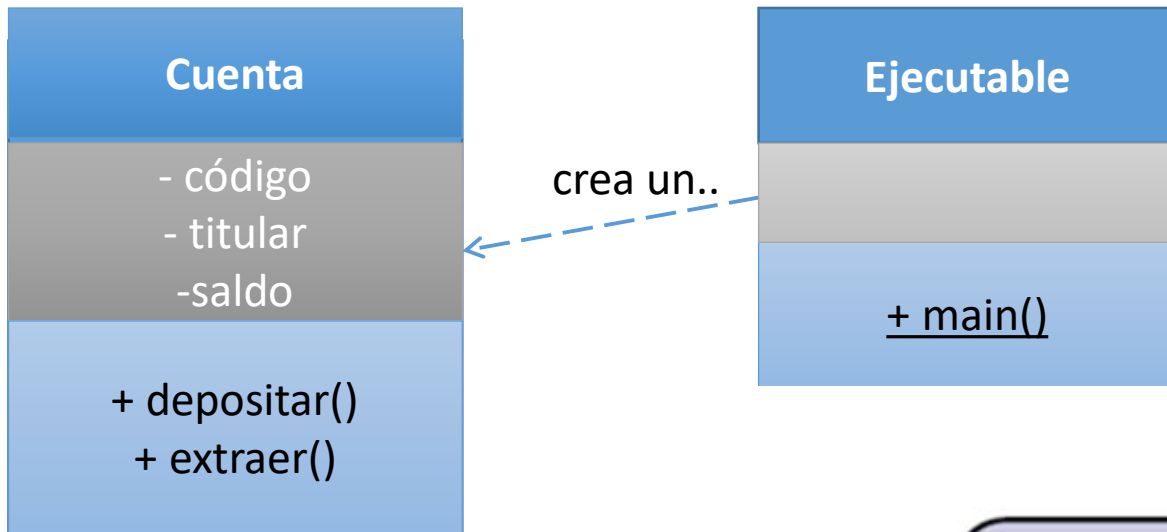
Cada cuenta tiene un código único, un titular y un saldo.

Se necesita simular el manejo de 2 cuentas y validar cuál tiene mayor saldo.

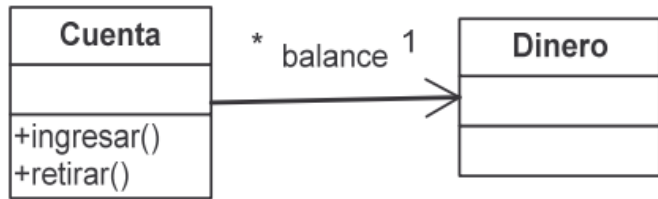
Tanto las operaciones de depósito como extracción deben ser seguras, es decir, no se puede extraer más del saldo actual, ni extraer una cantidad negativa.



## 3.5 Objetos y referencias



## 3.6 Relaciones entre clases

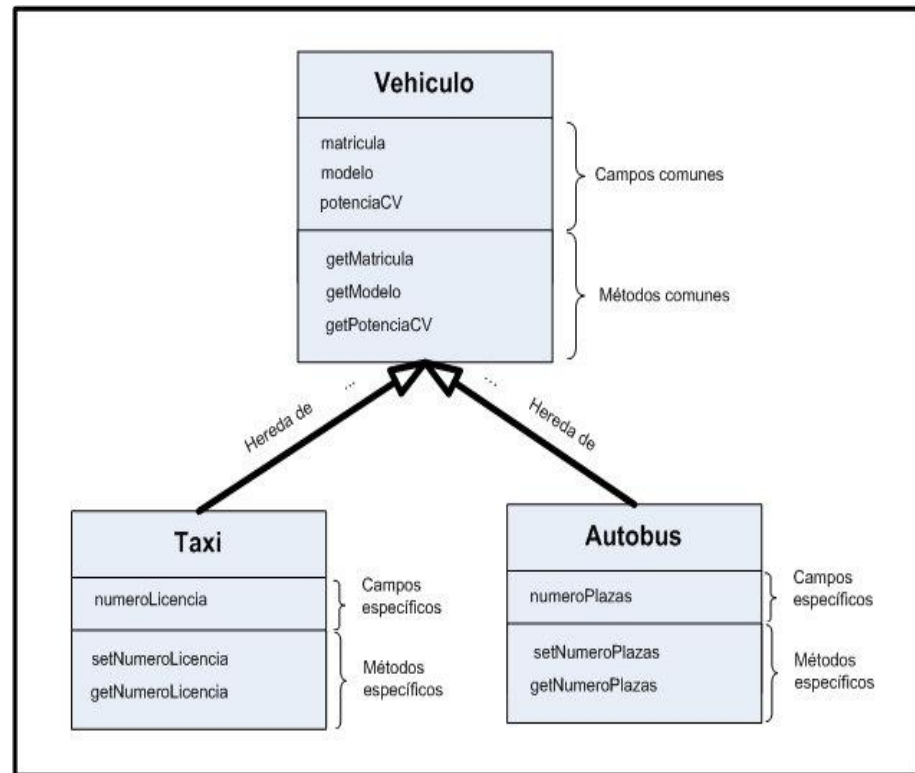


Asociaciones

Generalización /  
Especialización



$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$



## 3.7 Caso práctico



Vamos a NetBeans  
de nuevo!

# Caso práctico 02



Vamos a desarrollar una baraja de cartas españolas orientado a objetos. Una carta tiene un número entre 1 y 12 (el 8 y el 9 no los incluimos), y un palo (espadas, bastos, oros y copas). La baraja estará compuesta por un conjunto de cartas, 40 exactamente.

Las operaciones que podrá realizar la baraja son:

- **barajar**: cambia de posición todas las cartas aleatoriamente
- **siguienteCarta**: devuelve la siguiente carta que está en la baraja, cuando no haya más o se haya llegado al final, se indica al usuario que no hay más cartas.
- **cartasDisponibles**: indica el número de cartas que aún puede repartir
- **cartasMonton**: mostramos aquellas cartas que ya han salido, si no ha salido ninguna indicárselo al usuario
- **mostrarBaraja**: muestra todas las cartas hasta el final. Es decir, si se saca una carta y luego se llama al método, este no mostrara esa primera carta.

# GRACIAS!



DIPLOMATURA EN

## NUEVAS TECNOLOGÍAS

SECRETARÍA DE  
EXTENSIÓN  
UNIVERSITARIA  
UTN - FBC



UTN  
Facultad Regional Córdoba

Ministerio de  
PROMOCIÓN DEL EMPLEO  
Y DE LA ECONOMÍA FAMILIAR

Ministerio de  
CIENCIA Y  
TECNOLOGÍA



Organización  
de las Naciones Unidas  
para la Educación,  
la Ciencia y la Cultura

Con el apoyo de  
**Oficina de Montevideo**  
Oficina Regional de Gestión  
para América Latina y el Caribe