

# HERENCIA

---



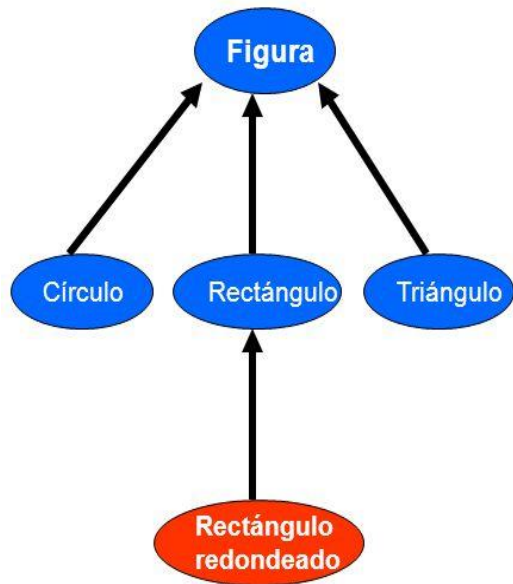
# Herencia

- Introducción:
  - Usamos las clases para representar parte del conocimiento que adquirimos del dominio de problema.
  - ¿Qué sucede cuando las clases comparten parte del conocimiento que representan?
    - Subclasificación.
  - Crear clases nuevas que se construyan tomando como base clases ya existentes.
  - Cuando se hereda, se reutilizan métodos y atributos.
  - Permite extender la funcionalidad de un objeto.

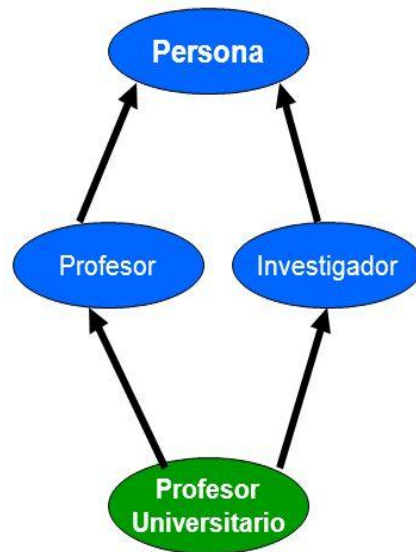
# Herencia

## TIPOS DE HERENCIA

### SIMPLE

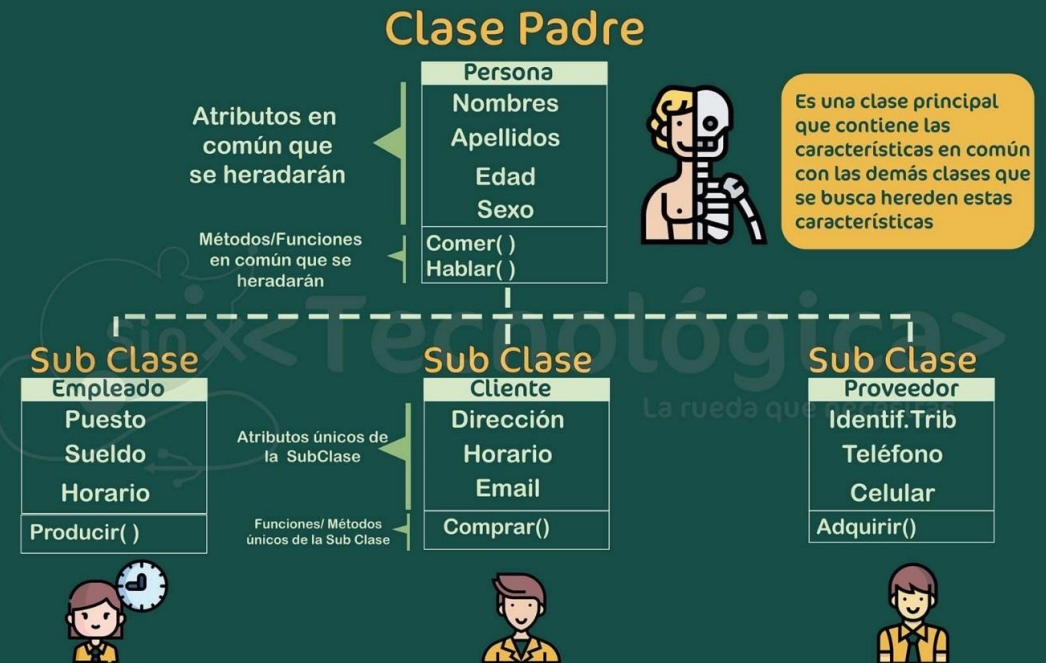


### MÚLTIPLE



# Herencia

## ¿Cómo funciona?

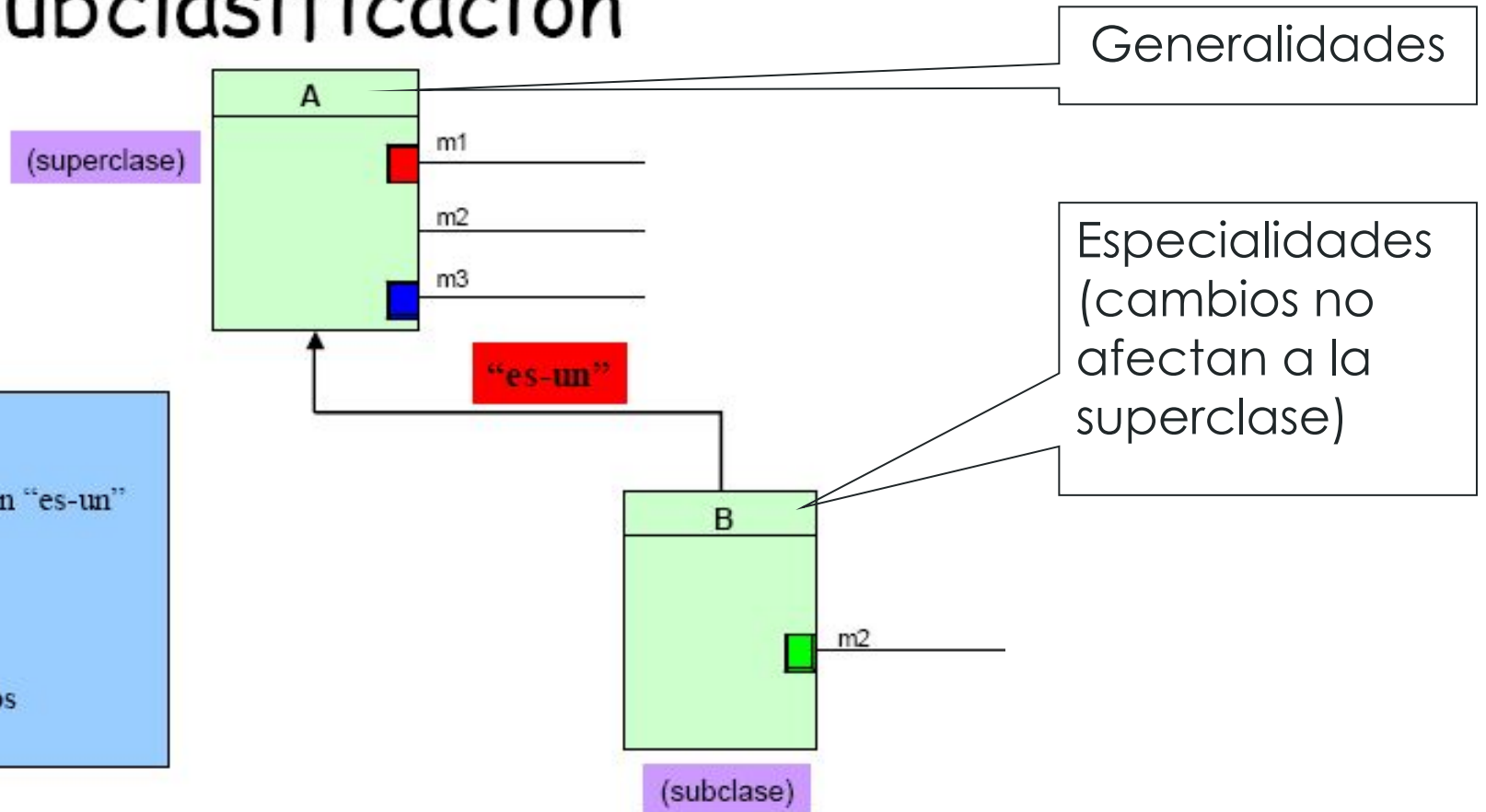


Las sub clases heredan las características de la clase padre, pero también pueden contener métodos/funciones y atributos únicos cada Sub Clase

# Herencia

## Subclasificación

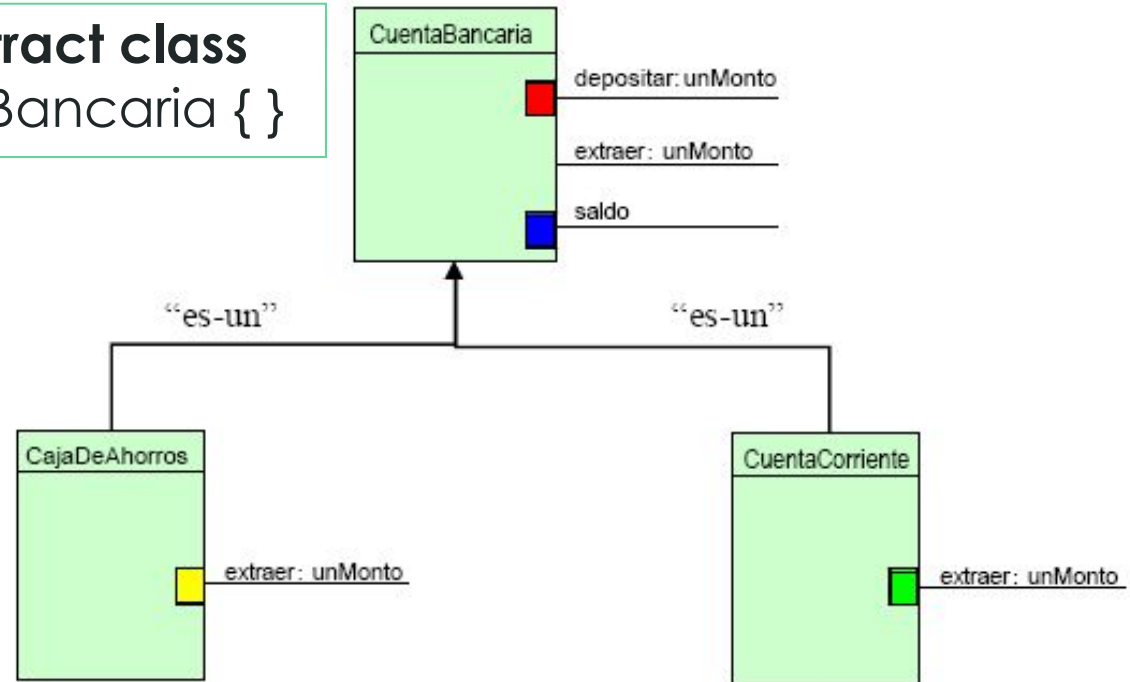
- Conceptos clave
- El significado de la relación “es-un”
  - Clases Abstractas
  - Clases Concretas
  - Jerarquías Polimórficas
  - Tipos
  - Re-definición de protocolos
  - Herencia



# Herencia

## Una jerarquía de clases

```
public abstract class  
CuentaBancaria { }
```



```
public class CajaDeAhorros  
    extends CuentaBancaria {  
}
```

```
public class CuentaCorriente  
    extends CuentaBancaria {  
}
```



# Herencia

- Constructores:
  - Los constructores NO se heredan de las superclases.
  - Si se puede invocar del constructor de la subclase a la superclase con `super()`.
  - Debe ser primera sentencia del cuerpo del constructor.

```
public class CajaDeAhorro extends CuentaBancaria {  
    public CajaDeAhorro() {  
        super();  
    }  
}
```



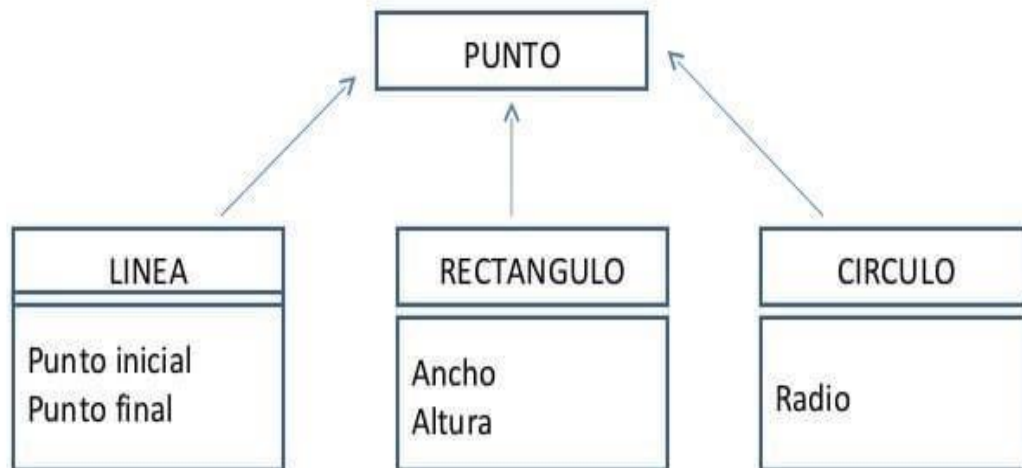
# Herencia

- Métodos:
  - Si heredo métodos que no tienen sentido en la subclase, la jerarquía fue forzada ☐ MAL DISEÑO.
  - Si heredo métodos que son insuficientes en la subclase ☐ OVERWRITING.
  - La sobreescritura puede ser PARCIAL o TOTAL.
  - Para sobrescribir un método se debe respetar la firma del mismo.  
Ejemplo:

```
public boolean equals(Object obj)
```

# Polimorfismo

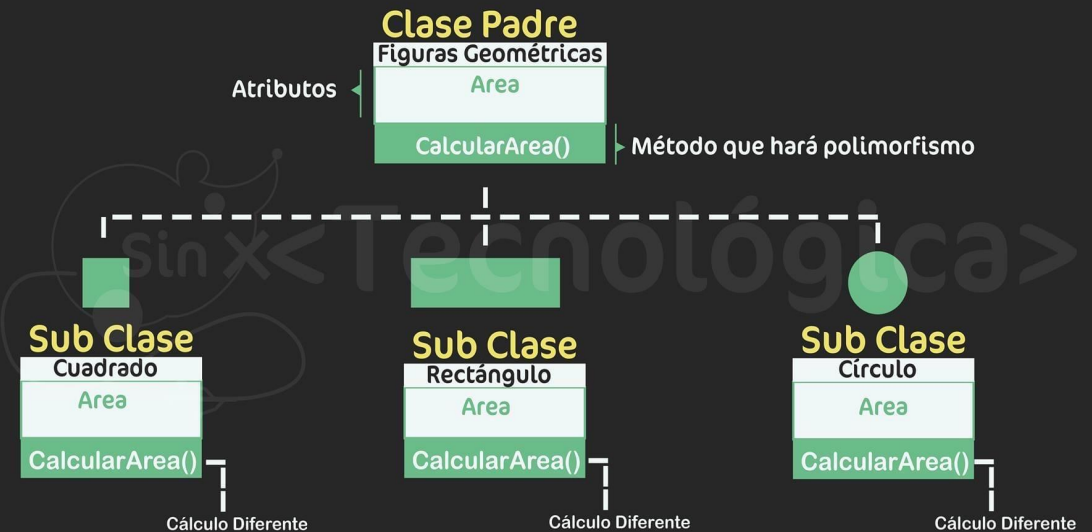
Es la capacidad que tiene los objetos de una clase de responder al mismo mensaje o evento en función de los parámetros utilizados durante su invocación.



# Polimorfismo

## ¿Cómo funciona?

Describe múltiples y posibles estados de una única propiedad, en otras palabras: **consiste en conseguir que un objeto de una clase se comporte como un objeto de cualquiera de sus subclases**



Como en cada sub clase se calcula de forma diferente el área pero bajo el mismo nombre del método, cambiaremos el comportamiento del mismo con Polimorfismo





# Polimorfismo

- Dos o más objetos son polimórficos respecto de un conjunto de mensajes, si todos pueden responder esos mensajes, aún cuando cada uno lo haga de un modo diferente.
- Objetos polimórficos corresponden a un mismo “tipo” de objeto □ mismo comportamiento esencial, independientemente de implementación.
- Permite que clases de distintos tipos puedan ser referenciadas por una misma variable:

```
CuentaBancaria cuentaBancaria;  
  
cuentaBancaria = new CajaDeAhorro();  
  
cuentaBancaria = new CuentaCorriente();
```

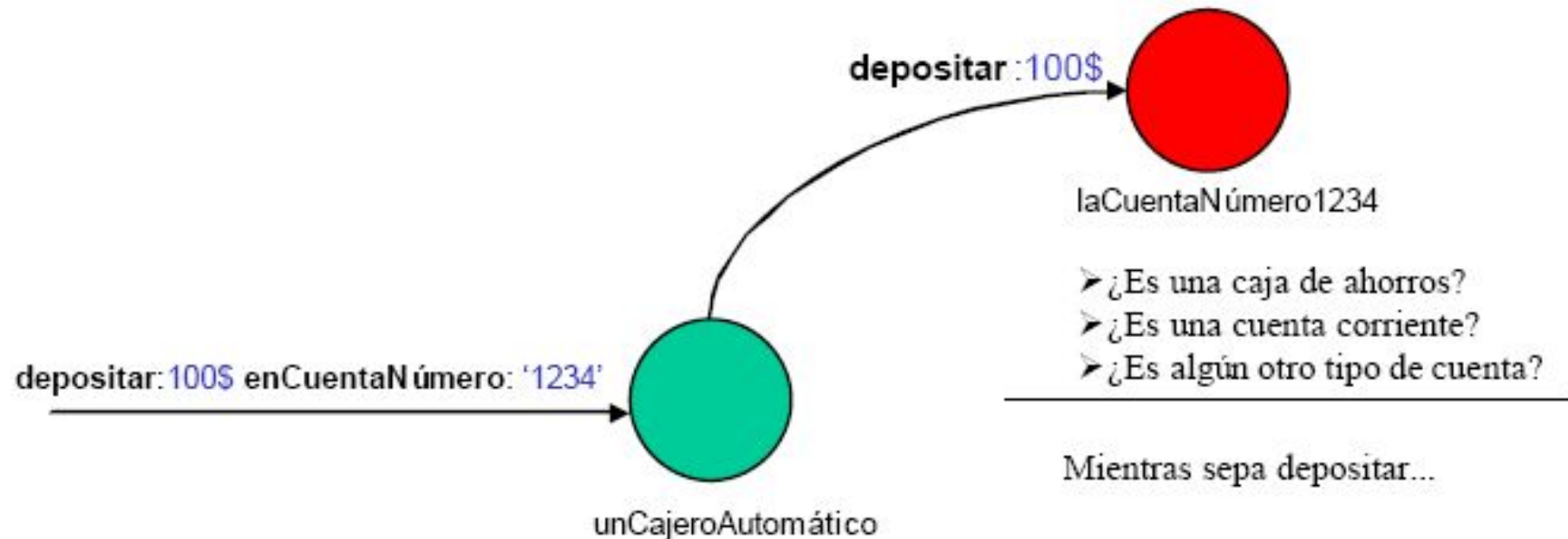
# Polimorfismo

## Dos objetos polimórficos...



# Polimorfismo

## Polimorfismo: ¿Para qué?





# Polimorfismo

- Es la clave de un buen diseño:
  - “Código” genérico.
  - Objetos desacoplados.
  - Objetos intercambiables.
  - Objetos reusables.
  - Programar por protocolo, no por implementación (buscar la esencia).

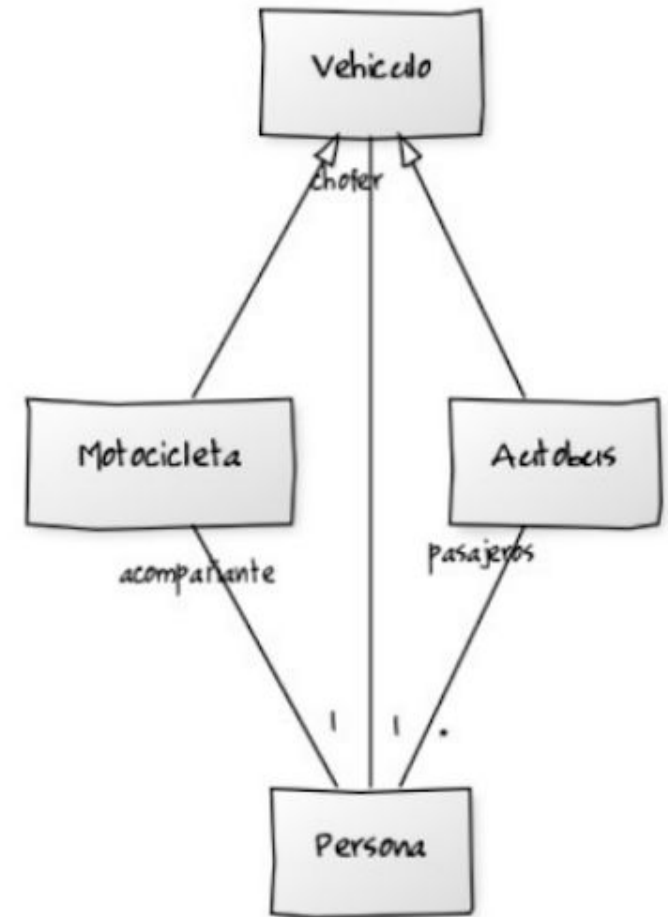


# Polimorfismo

- Tipos de Binding:
  - **Binding Temprano o Estático:** operandos y operadores son ligados en tiempo de compilación. Es el programador quién decide qué se debe ejecutar.
  - **Binding Tardío o Dinámico:** operandos y operadores son ligados en tiempo de ejecución. Es el objeto receptor del mensaje quién decide qué se debe ejecutar. La Máquina Virtual tiene que llamar a la versión del método que sea la adecuada para el tipo real del objeto al que se refiere.

# Ejercicios

- 1) Existen dos tipos de vehículos: las motocicletas, que llevan un chofer y un acompañante, y los autobuses, que llevan un chofer y varios pasajeros. Los vehículos deben conocer la cantidad de kilómetros recorridos, asignar y cambiar chofer. Cada vehículo particular deberá poder agregar un acompañante o diversos pasajeros, respectivamente. En caso del Autobús, no puede cambiar de chofer si hubiera pasajeros. En el caso de la Motocicleta, no puede hacerlo si hubiera un acompañante.





# Ejercicios

- 2) Modele una Empresa con empleados. Una empresa conoce a todos sus empleados. Los empleados pueden ser de planta permanente o temporaria, además hay gerentes, que también son empleados de planta permanente, pero siguen un régimen salarial particular. Cuando un empleado es de planta permanente cobra la cantidad de horas trabajadas por \$300, más antigüedad (\$100 por año de antigüedad), más salario familiar. Cuando es de planta temporaria, no cobra antigüedad y cobra la cantidad de horas trabajadas por \$200, más salario familiar. El salario familiar es \$200 por cada hijo, los empleados casados además cobran \$100 por su esposa/o. Un gerente cobra de manera similar a un empleado de planta permanente pero su hora trabajada vale \$400, por antigüedad se le pagan \$150 por año, mientras que el salario familiar es el mismo que el de los empleados de planta permanente y temporal. (cont....)



# Ejercicios

- 2) (cont...) Defina e implemente el mensaje `montoTotal()` en la clase `Empresa`, que retorna el monto total que la empresa debe pagar en concepto de sueldos a sus empleados. Provea una clase `TestEmpresa` para instanciar y testear su sistema. En el método `main` de esa clase cree el siguiente escenario y envíe a la empresa el mensaje `montoTotal()` para obtener la liquidación total. (cont...)





# Ejercicios

- 2) (cont...)
  - Una empresa, con el CUIT y Razón Social que desee, y con los cuatro empleados que se describen a continuación.
    - Un empleado de Planta Temporal con 80 horas trabajadas, con esposa y sin hijos.
    - Un empleado de Planta Permanente (que no sea gerente) con 80 horas trabajadas, con esposa, 2 hijos y 6 años de antigüedad.
    - Un empleado de Planta Permanente (que no sea gerente) con 160 horas trabajadas, sin esposa, sin hijos y con 4 años de antigüedad.
    - Un Gerente con 160 horas trabajadas, con esposa, un hijo y 10 años de antigüedad.