

# 14. Ejemplo de herencia Prototípica

## 1. Clase Padre: **Mamifero**

Definimos una función constructora para crear objetos con atributos y métodos compartidos:

```
function Mamifero(comestible) {  
  this.patas = 4;  
  this.cola = true;  
  this.comestible = comestible;  
}  
  
// Métodos agregados al prototipo  
Mamifero.prototype.correr = function() {  
  console.log("Este animal corre");  
};  
  
Mamifero.prototype.saltar = function() {  
  console.log("Este animal salta");  
};  
  
Mamifero.prototype.meLoComo = function() {  
  if (this.comestible) {  
    console.log("Te has comido un mamífero");  
  } else {  
    console.log("No puedes comerte este animal");  
  }  
};
```

## 2. Subclase: **AnimalDeGranja**

Heredamos de **Mamifero** usando **Object.create()** y **call()** para vincular el contexto:

```
function AnimalDeGranja(comestible) {  
  // Llamamos al constructor padre (similar a `super`)  
  Mamifero.call(this, comestible);  
  this.enChozo = false;  
}  
  
// Herencia del prototipo de Mamifero  
AnimalDeGranja.prototype = Object.create(Mamifero.prototype);  
AnimalDeGranja.prototype.constructor = AnimalDeGranja;  
  
// Métodos nuevos para AnimalDeGranja  
AnimalDeGranja.prototype.guardarEnChozo = function() {  
  this.enChozo = true;  
};
```

```
AnimalDeGranja.prototype.sacarDeChoza = function() {
  this.enChoza = false;
};

// Sobrescribir el método `meLoComo`
AnimalDeGranja.prototype.meLoComo = function() {
  if (this.comestible) {
    console.log("Te has comido un animal de granja");
  } else {
    console.log("No puedes comerte este animal");
  }
};
```

### 3. Subclase: **AnimalDomestico**

Heredamos de `AnimalDeGranja` y agregamos atributos/métodos específicos:

```
function AnimalDomestico(nombre) {
  // Llamamos al constructor padre (similar a `super`)
  AnimalDeGranja.call(this, false); // comestible = false
  this.nombre = nombre;
  this.enCasa = false;
}

// Herencia del prototipo de AnimalDeGranja
AnimalDomestico.prototype = Object.create(AnimalDeGranja.prototype);
AnimalDomestico.prototype.constructor = AnimalDomestico;

// Métodos nuevos para AnimalDomestico
AnimalDomestico.prototype.guardarEnCasa = function() {
  this.enCasa = true;
};

AnimalDomestico.prototype.sacarDeCasa = function() {
  this.enCasa = false;
};

// Sobrescribir el método `meLoComo`
AnimalDomestico.prototype.meLoComo = function() {
  console.log(`No te puedes comer a ${this.nombre}`);
  throw new Error("Este método no está disponible en la subclase");
};
```

### 4. Ejemplos de Uso

## Crear Instancias

```
// Instancia de Mamifero
const ciervo = new Mamifero(true);
ciervo.correr(); // "Este animal corre"
ciervo.meLoComo(); // "Te has comido un mamífero"

// Instancia de AnimalDeGranja
const cerdo = new AnimalDeGranja(true);
cerdo.guardarEnChoza(); // Modifica `enChoza`
cerdo.meLoComo(); // "Te has comido un animal de granja"

// Instancia de AnimalDomestico
const perro = new AnimalDomestico("Lassie");
perro.guardarEnCasa(); // Modifica `enCasa`
perro.guardarEnChoza(); // Método heredado de AnimalDeGranja
```

## 5. Cadena de Prototipos

Cada objeto tiene una cadena de prototipos que permite la herencia:

```
console.log(perro instanceof AnimalDomestico); // true
console.log(perro instanceof AnimalDeGranja); // true
console.log(perro instanceof Mamifero); // true
```