

Universidad Autónoma  
de Baja California

INGENIERÍA EN COMPUTACIÓN

PROGRAMACIÓN ORIENTADA  
A OBJETOS (541)

*Práctica 2*



Autor:  
Luis E. Galindo Amaya  
1274895

22 de febrero de 2022

## 1. Proyecto En Github

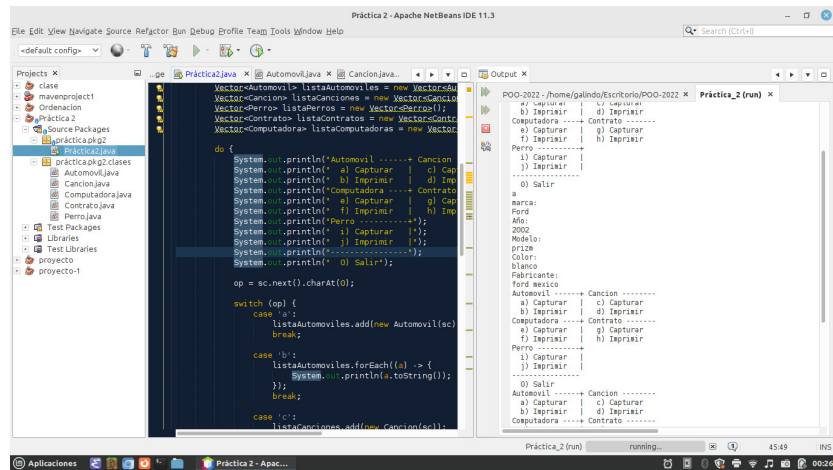
click en las listas para abrir directamente el archivo en el navegador.

■ <https://github.com/Galindo-lab/P00-2022>

- Main.java
  - Automovil.java
  - Cancion.java
  - Computadora.java
  - Contrato.java
  - Perro.java

## 2. Capturas De Pantalla

No añadí captura de cada opción del menú, son diez opciones y todas tienen la misma lógica detrás, utilice un vector para guardar los registros y un foreach para mostrar el contenido de cada uno.



```

class
{
    Vector<Automovil> listaAutomoviles = new Vector<Automovil>();
    Vector<Cancion> listaCanciones = new Vector<Cancion>();
    Vector<Perro> listaPerros = new Vector<Perro>();
    Vector<Contrato> listaContratos = new Vector<Contrato>();
    Vector<Computadora> listaComputadoras = new Vector<Computadora>();

    do {
        System.out.println("Automovil ----- Cancion -----");
        System.out.println("a) Capturar | c) Capturar");
        System.out.println("b) Imprimir | d) Imprimir");
        System.out.println("Computadora ----- Contrato -----");
        System.out.println("e) Capturar | g) Capturar");
        System.out.println("f) Imprimir | h) Imprimir");
        System.out.println("Perro -----");
        System.out.println("i) Capturar |");
        System.out.println("j) Imprimir |");
        System.out.println("0) Salir");
        op = sc.next().charAt(0);

        switch (op) {
            case 'a':
                listaAutomoviles.add(new Automovil(sc));
                break;
            case 'b':
                listaAutomoviles.forEach((a) -> {
                    System.out.println(a.toString());
                });
                break;
            case 'c':
                listaCanciones.add(new Cancion(sc));
                break;
        }
    } while (op != '0');
}

```

```

            case 'h':
                listaContratos.forEach((a) -> {
                    System.out.println(a.toString());
                });
                break;
            case 'i':
                listaPerros.add(new Perro(sc));
                break;
            case 'j':
                listaPerros.forEach((a) -> {
                    System.out.println(a.toString());
                });
                break;
            case '0':
                System.out.println("Adios :");
                break;
        }
    } while (op != '0');
}

```

```

            case 'i':
                listaPerros.forEach((a) -> {
                    System.out.println(a.toString());
                });
                break;
            case '0':
                System.out.println("Adios :");
                break;
        }
    } while (op != '0');
}

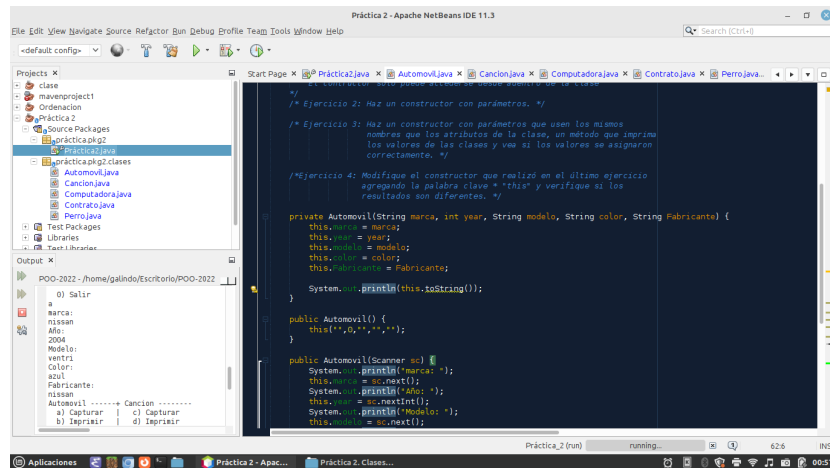
```

### 3. Ejercicios Extras

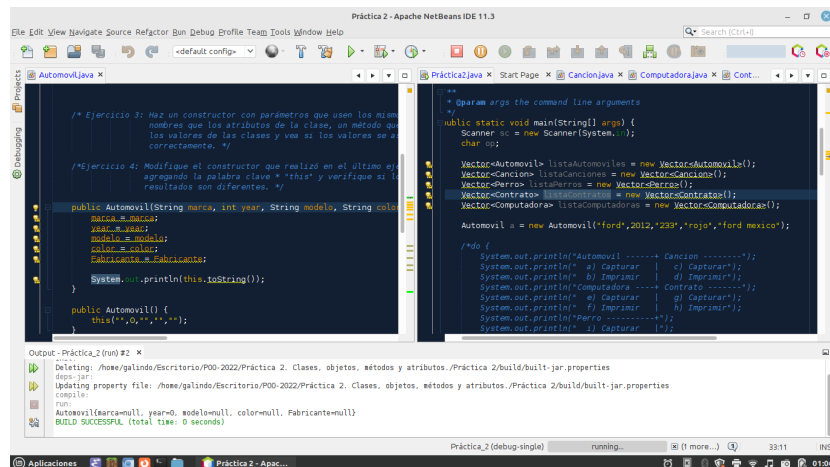
**Ejercicio 1** Implementa una clase con un constructor privado y ve que sucede.

Si es private el constructor solo se puede acceder desde dentro de la clase, en mi programa de aproveche de esto para que no se contruyeran clases espesificamente con ese constructor.

**Ejercicio 2** Haz un constructor con parámetros.



**Ejercicio 3** Haz un constructor con parámetros que usen los mismos nombres que los atributos de la clase, un método que imprima los valores de las clases y vea si los valores se asignaron correctamente.



**Ejercicio 4** Modifique el constructor que realizó en el último ejercicio agregando la palabra clave "this" y verifique si los resultados son diferentes.

