

Resumen

a) Pasos para implementar el patrón *Visitor* (se hacen *una sola vez*)

1) Hacer un interfaz **Visitor** con un método *visit* por cada tipo de nodo del árbol.

```
public interface Visitor {  
    public Object visit(Programa p, Object param);  
    public Object visit(Print p, Object param);  
    ...  
}
```

2) Añadir un método **accept** al interfaz **Nodo** (así se obliga a que lo implementen *todos* los nodos).

```
public interface Nodo {  
    Object accept(Visitor v, Object param);  
}
```

3) Hacer que todos los nodos implementen el método **accept**. En él solo tienen que invocar al método **visit**.

```
class Print implements Sentencia { // Sentencia extends Nodo
```

```
    ...  
    Object accept(Visitor v, Object param) {  
        return v.visit(this, param);  
    }  
}
```

```
class Read implements Sentencia { // Sentencia extends Nodo
```

```
    ...  
    Object accept(Visitor v, Object param) {  
        return v.visit(this, param);  
    }  
}
```

Se puede copiar y pegar en todos los nodos

b) Para implementar un nuevo recorrido del árbol

La clase que implemente el recorrido solo tiene que derivar de **Visitor** y dar implementación a todos sus métodos.

```
public class MiNuevoVisitor implements Visitor {  
    ...  
}
```

Pero no hace falta tocar los nodos!!!