

STRATEGY PATTERN


```
public class Materia {  
  
    private String nombre;  
    private int precio;  
  
    ...  
}
```

```
public class Matricula {  
  
    private Alumno alumno;  
    private List<Materia> materias;  
  
    ...  
}
```

```
public class Secretaria {  
  
    public int precio (Matricula matricula) {  
        int total = 0;  
        for (Materia materia: matricula.getMaterias()) {  
            total += materia.getPrecio();  
        }  
        return total;  
    }  
}
```


No todos los alumnos pagan lo mismo por las materias.
P.e. familia numerosa pagan un 50% del precio base


```
public interface Alumno {  
    int precioPagado (Materia materia);  
}
```

```
public class AlumnoFamiliaNumerosa implements Alumno {  
  
    @Override  
    public int precioPagado(Materia materia) {  
        return materia.getPrecio() / 2;  
    }  
}
```


Problema: el tipo de pago puede cambiar dinámicamente.

P.e. Si la secretaría considera que el número de matrículas supera una cantidad, asigna al alumno algún tipo con descuento

¡ No podemos cambiar tipos en tiempo de ejecución !

“

Define a family of algorithms,
encapsulate each one, and make them
interchangeable. Strategy lets the
algorithm vary independently from
clients that use it.

— *GoF*

”


```
public interface EstrategiaDePago {  
    int precioPagado (Materia materia);  
}
```

```
public class EstrategiaDePagoPorDefecto implements EstrategiaDePago {  
  
    @Override  
    public int precioPagado(Materia materia) {  
        return materia.getPrecio();  
    }  
}
```



```
public class Alumno {  
    private EstrategiaDePago estrategia;  
    ...  
}
```

¡ La estrategia ya no está asociada al tipo (invariable) sino a una propiedad del objeto (modificable) !


```
public class Secretaria {
```

```
    public int precio (Matricula matricula) {
```

```
        // si lo considera
```

```
        matricula.getAlumno().setEstrategia(<nueva estrategia>);
```

```
        EstrategiaDePago estrategia = matricula.getAlumno().getEstrategia();
```

```
        int total = 0;
```

```
        for (Materia materia: matricula.getMaterias()) {
```

```
            total += estrategia.precioPagado(materia);
```

```
        }
```

```
        return total;
```

```
    }
```

```
}
```


STRATEGY PATTERN