

### Punto 1

Se parte de estos dos ejemplos en Smalltalk y Prolog

#Cuenta total ^saldo := saldo + deposito	<table border="1"><tr><td>Clase Cuenta</td></tr><tr><td>nombre <b>Juan</b> saldo <b>500</b> deposito <b>100</b></td></tr></table>	Clase Cuenta	nombre <b>Juan</b> saldo <b>500</b> deposito <b>100</b>	cuenta(juan,500). deposito(Juan,100).  total(Tot,C) :- cuenta(C,Saldo), deposito(C,D), Tot is Saldo + D.
Clase Cuenta				
nombre <b>Juan</b> saldo <b>500</b> deposito <b>100</b>				

- Como se manifiesta el efecto de lado en los ejemplos anteriores? Justifique su respuesta.
  - Como modifica en los códigos anteriores para que no se produzca el efecto de lado, en ninguno de ellos?
- 

### Punto 2

Dado el siguiente código de Prolog:

```
hecho(Alberto, [1,2,3,0,5]).  
hecho(Berta, [1,3,0,5,2]).  
...  
regla(X,Y) :- hecho(X,L), primero(L,Y).  
regla(X,Y) :- hecho(X,L), ultimo(L,Y).  
  
primero([X_|_],X).  
  
ultimo([X],X).  
ultimo([_|Xs],Y) :- ultimo(Xs,Y).
```

- Escribir todos los ejemplos que muestren la variedad de consultas que se pueden hacer utilizando el predicado *regla*, con sus correspondientes respuestas.
  - Cuáles son los conceptos que se aplican? Justifique.
- 

### Punto 3

Dada la siguiente clase:

<b>Prestamo</b>
nombreLibro fechaPrestamo genero plazo socio

Para cada atributo hay un método con el mismo nombre que simplemente devuelve el atributo, por ejemplo:

```
plazo  
^plazo
```

- Suponiendo que todos los libros del mismo genero se prestan por el mismo plazo y un libro puede prestarse a diferentes socios y en diferentes oportunidades, Como modificaría la estructura de clases para que se adapte a este supuesto?
  - Modifique el método *plazo* para que se adapte a la nueva estructura de clases. Justifique.
  - Que pasa con los otros objetos que usaban los métodos anteriores del préstamo?
- 

### Punto 4

- Defina declaratividad.
- Cuáles son sus ventajas?