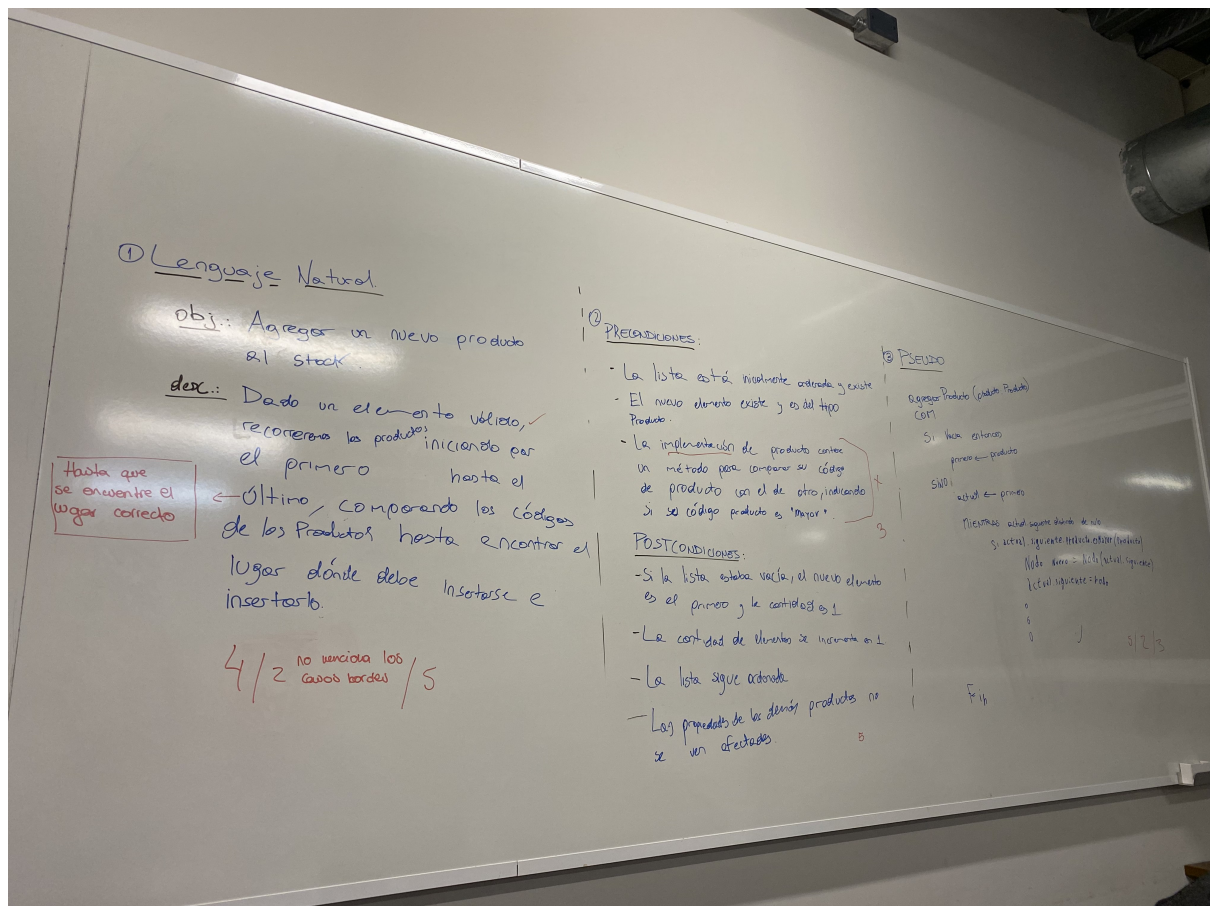


# UT3\_PD2

▼ Property	COMPLETED
📅 Date	@April 30, 2022
☰ BLOCKED	



## LENGUAJE NATURAL:

El principal problema fue no considerar las condiciones de borde, es decir, que sucedería si la lista esta vacia, o si la lista o si se agrega al principio o al final.

Ademas tambien esta el hecho de que nosotros decimos que tenemos que recorrer toda la lista, cuando en realidad solo hay que recorrer hasta que encuentre el lugar correcto

CORRECCION: Dado un elemento valido, verificaoms que los productos tengan algun producto. En caso de estar vacio agregamos como primero al elemento valido. Caso contrario recorremos productos iniciando por el primero hasta que encuentre el lugar correcto, comparando los codigos de los productos hasta encontrar el lugar donde debe insertarse e insertarlo

### **PRECONDICIONES:**

Sacaria a la ultima precondition pues no viene al caso del problema a resolver

CORRECCION:

PRECONDICIONES:

- La lista se encuentra originalmente ordenada y existe
- El nuevo elemento existe y es del tipo Producto

### **POSTCONDICIONES:**

Considero que no hay nada a correguir.

### **PSEUDO:**

Considero que no hay nada a correguir.

POSIBLES SITUACIONES:

- Lista vacia
- Lista ordenada

```
agregarProducto( Producto : producto)
COM
SI vacia =>
    primero <- producto
SINO
    actual <- primero
    MIENTRAS actual.siguiente <> nulo
        SI actual.siguiente.prodcuto.esMayor(producto)
            Nodo nuevo = Nodo (actual.siguiente)
            actual.siguiente = nodo
        FIN SI
```