



DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico 1 - Especificación

Fat Food

25 de agosto de 2015

Algoritmos y Estructura de Datos I

Grupo xx Plaza Sesamo

Integrante	LU	Correo electrónico
XXXX XXXX, XXX XXX	XXX/XX	XXX@XXX.XXX
XXXX XXXX, XXX XXX	XXX/XX	XXX@XXX.XXX
Martín Darricades, Matías Facundo	480/13	matiasamd@gmail.com
Solari Saban, Tomás León	774/14	tomassolari94@gmail.com



Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2160 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (54 11) 4576-3359

<http://www.exactas.uba.ar>

Índice

1. Tipos	3
2. Combo	3
3. Pedido	4
4. Local	5
5. Funciones Auxiliares	7
5.1. Combo	7
5.2. Pedido	7
5.3. Local	7

1. Tipos

```
tipo Empleado = String ;
tipo Energia =  $\mathbb{Z}$  ;
tipo Cantidad =  $\mathbb{Z}$  ;
tipo Bebida = Pesti Cola, Falsa Naranja, Se ve nada, Agua con Gags, Agua sin Gags ;
tipo Hamburguesa = McGyver, CukiQueFresco (Cuarto de Kilo con Queso Fresco), McPato, Big Macabra ;
```

2. Combo

```
tipo Combo {
    observador bebida (c: Combo) : Bebida ;
    observador sandwich (c: Combo) : Hamburguesa ;
    observador dificultad (c: Combo) : Energia ;

    invariante dificultadHasta100 : energiaEnRango(dificultad(c)) ;
}

problema nuevoC (b: Bebida, h: Hamburguesa, d: Energia) = res : Combo {
}

problema bebidaC (c: Combo) = res : Bebida {
}

problema sandwichC (c: Combo) = res : Hamburguesa {
}

problema dificultadC (c: Combo) = res : Energia {
}
```

3. Pedido

```
tipo Pedido {  
    observador numero (p: Pedido) :  $\mathbb{Z}$  ;  
    observador atendio (p: Pedido) : Empleado ;  
    observador combos (p: Pedido) : [Combo] ;  
  
    invariante numeroPositivo :  $numero(p) > 0$  ;  
    invariante pideAlgo :  $|combos(p)| > 0$  ;  
}  
  
problema nuevoP (n:  $\mathbb{Z}$ , e: Empleado, cs: [Combo]) = res : Pedido {  
}  
  
problema numeroP (p: Pedido) = res :  $\mathbb{Z}$  {  
}  
  
problema atendioP (p: Pedido) = res : Empleado {  
}  
  
problema combosP (p: Pedido) = res : [Combo] {  
}  
  
problema agregarComboP (p: Pedido, c: Combo) {  
}  
  
problema anularComboP (p: Pedido, i:  $\mathbb{Z}$ ) {  
}  
  
problema cambiarBebidaComboP (p: Pedido, b: Bebida, i:  $\mathbb{Z}$ ) {  
}  
  
problema elMezcladitoP (p: Pedido) {  
}
```

4. Local

```

tipo Local {
  observador stockBebidas (l: Local, b: Bebida) : Cantidad ;
    requiere  $b \in bebidasDelLocal(l)$  ;
  observador stockSandwiches (l: Local, h: Hamburguesa) : Cantidad ;
    requiere  $h \in sandwichesDelLocal(l)$  ;
  observador bebidasDelLocal (l:Local) : [Bebida] ;
  observador sandwichesDelLocal (l:Local) : [Hamburguesa] ;
  observador empleados (l: Local) : [Empleado] ;
  observador desempleados (l: Local) : [Empleado] ;
  observador energiaEmpleado (l: Local, e: Empleado) : Energia ;
    requiere  $e \in empleados(l)$  ;
  observador ventas (l: Local) : [Pedido] ;

  invariante hayBebidasYSonDistintas :  $|bebidasDelLocal(l)| > 0 \wedge distintos(bebidasDelLocal(l))$  ;
  invariante haySandwichesYSonDistintos :  $|sandwichesDelLocal(l)| > 0 \wedge distintos(sandwichesDelLocal(l))$  ;
  invariante stockBebidasPositivo :  $(\forall b \leftarrow bebidasDelLocal(l)) stockBebidas(l, b) \geq 0$  ;
  invariante stockSandwichesPositivo :  $(\forall h \leftarrow sandwichesDelLocal(l)) stockSandwiches(l, h) \geq 0$  ;
  invariante empleadosDistintos :  $distintos(empleados(l) ++ desempleados(l))$  ;
  invariante energiaHasta100 :  $(\forall e \leftarrow empleados(l)) energiaEnRango(energiaEmpleado(l, e))$  ;
  invariante empleadosQAtendieronDelLocal :  $(\forall v \leftarrow ventas(l)) atendio(v) \in empleados(l) ++ desempleados(l)$  ;
  invariante ventasCorrelativas : ... ;
  invariante combosDeLocal : ... ;
}

problema stockBebidasL (l: Local, b:Bebida) = res : Cantidad {
}

problema stockSandwichesL (l: Local, h:Hamburguesa) = res : Cantidad {
}

problema bebidasDelLocalL (l: Local) = res : [Bebida] {
}

problema sandwichesDelLocalL (l: Local) = res : [Hamburguesa] {
}

problema empleadosL (l: Local) = res : [Empleado] {
}

problema desempleadosL (l: Local) = res : [Empleado] {
}

problema energiaEmpleadoL (l: Local, e:Empleado) = res : Energia {
}

problema ventasL (l: Local) = res : [Pedido] {
}

problema unaVentaCadaUno (l:Local) = res : Bool {
}

problema venderL (l: Local, p:Pedido) {
}

problema candidatosAEmpleadosDelMesL (l: Local) = res : [Empleado] {
}

problema sancionL (l: Local, e:Empleado, n:Energia) {
}

problema elVagonetaL (l: Local) = res : Empleado {
}

```

```
}
```

```
problema anularPedidoL (l: Local, n:  $\mathbb{Z}$ ) {  
}
```

```
problema agregarComboAlPedidoL (l: Local, c: Combo, n:  $\mathbb{Z}$ ) {  
}
```

5. Funciones Auxiliares

aux distintos (ls:[T]) : Bool = $(\forall i, j \leftarrow [0..|ls|], i \neq j) ls_i \neq ls_j$;
aux energiaEnRango (e: Energia) : Bool = $0 \leq e \leq 100$;

5.1. Combo

5.2. Pedido

5.3. Local