

1. Tipos

```
tipo Id =  $\mathbb{Z}$ ;
tipo Carga =  $\mathbb{Z}$ ;
tipo Ancho =  $\mathbb{Z}$ ;
tipo Largo =  $\mathbb{Z}$ ;
tipo Parcela = Cultivo, Granero, Casa;
tipo Producto = Fertilizante, Plaguicida, PlaguicidaBajoConsumo, Herbicida, HerbicidaLargoAlcance;
tipo EstadoCultivo = ReciénSembrado, EnCrecimiento, ListoParaCosechar, ConMaleza, ConPlaga, NoSensado;
```

2. Campo

```
tipo Campo {
  observador dimensiones (c: Campo) : (Ancho, Largo);
  observador contenido (c: Campo, i, j:  $\mathbb{Z}$ ) : Parcela;
  requiere enRango :  $0 \leq i < prm(dimensiones(c)) \wedge 0 \leq j < sgd(dimensiones(c))$ ;

  invariante dimensionesValidas :  $prm(dimensiones(c)) > 0 \wedge sgd(dimensiones(c)) > 0$ ;
  invariante unaSolaCasa :  $|\{(i, j) | i \leftarrow [0..prm(dimensiones(c))], j \leftarrow [0..sgd(dimensiones(c))],$ 
     $contenido(c, i, j) == Casa\}| == 1$ ;
  invariante unSoloGranero :  $|\{(i, j) | i \leftarrow [0..prm(dimensiones(c))], j \leftarrow [0..sgd(dimensiones(c))],$ 
     $contenido(c, i, j) == Granero\}| == 1$ ;
  invariante algoDeCultivo :  $|\{(i, j) | i \leftarrow [0..prm(dimensiones(c))], j \leftarrow [0..sgd(dimensiones(c))],$ 
     $contenido(c, i, j) == Cultivo\}| \geq 1$ ;
  invariante posicionesAlcanzables :  $posicionesAlcanzablesEn100(c)$ ;
}

aux posicionesAlcanzablesEn100 (c: Campo) : Bool =
  alcanzableEn100(posicionGranero(c), prm(dimensiones(c)), sgd(dimensiones(c)));

problema crearC (posG, posC: ( $\mathbb{Z}$ ,  $\mathbb{Z}$ )) = res : Campo {
}

problema dimensionesC (c: Campo) = res : (Ancho, Largo) {
}

problema contenidoC (c: Campo, i, j:  $\mathbb{Z}$ ) = res : Parcela {
}
```

3. Drone

```

tipo Drone {
  observador id (d: Drone) : Id;
  observador bateria (d: Drone) : Carga;
  observador enVuelo (d: Drone) : Bool;
  observador vueloRealizado (d: Drone) :  $[(\mathbb{Z}, \mathbb{Z})]$ ;
  observador posicionActual (d: Drone) :  $(\mathbb{Z}, \mathbb{Z})$ ;
  observador productosDisponibles (d: Drone) : [Producto];

  invariante vuelosOk :
     $enVuelo(d) \Rightarrow (|vueloRealizado(d)| > 0 \wedge posicionActual(d) == vueloRealizado(d)_{|vueloRealizado(d)|-1} \wedge$ 
     $posicionesPositivas(d) \wedge movimientosOK(d)) \wedge \neg enVuelo(d) \Rightarrow |vueloRealizado(d)| == 0;$ 
  invariante bateriaOk :  $0 \leq bateria(d) \leq 100;$ 
}

aux posicionesPositivas (d: Drone) : Bool =  $(\forall i \leftarrow [0..|vueloRealizado(d)|]) prm(vueloRealizado(d)_i) \geq 0 \wedge$ 
 $sgd(vueloRealizado(d)_i) \geq 0;$ 
aux movimientosOK (d: Drone) : Bool =  $(\forall i \leftarrow [1..|vueloRealizado(d)|])$ 
 $prm(vueloRealizado(d)_i) == prm(vueloRealizado(d)_{i-1}) \wedge (sgd(vueloRealizado(d)_i) == sgd(vueloRealizado(d)_{i-1}) - 1 \vee$ 
 $sgd(vueloRealizado(d)_i) == sgd(vueloRealizado(d)_{i-1}) + 1) \vee sgd(vueloRealizado(d)_i) == sgd(vueloRealizado(d)_{i-1})$ 
 $\wedge (prm(vueloRealizado(d)_i) == prm(vueloRealizado(d)_{i-1}) - 1 \vee prm(vueloRealizado(d)_i) == prm(vueloRealizado(d)_{i-1}) +$ 
 $1);$ 

problema crearD (i:  $\mathbb{Z}$ , ps: [Producto]) = res : Drone {
}

problema idD (d: Drone) = res :  $\mathbb{Z}$  {
}

problema bateriaD (d: Drone) = res :  $\mathbb{Z}$  {
}

problema enVueloD (d: Drone) = res : Bool {
}

problema vueloRealizadoD (d: Drone) = res :  $[(\mathbb{Z}, \mathbb{Z})]$  {
}

problema posicionActualD (d: Drone) = res :  $(\mathbb{Z}, \mathbb{Z})$  {
}

problema productosDisponiblesD (d: Drone) = res : [Producto] {
}

problema vueloEscaleraD (d: Drone) = res : Bool {
}

problema vuelosCruzadosD (ds: [Drone]) = res :  $[(\mathbb{Z}, \mathbb{Z}), \mathbb{Z}]$  {
}

```

4. Sistema

```
tipo Sistema {
  observador campo (s: Sistema) : Campo;
  observador estadoDelCultivo (s: Sistema, i, j:  $\mathbb{Z}$ ) : EstadoCultivo;
    requiere  $enRango(dimensiones(s), i, j) \wedge contenido(campo(s), i, j) == Cultivo$ ;
  observador enjambreDrones (s: Sistema) : [Drone];

  invariante identificadoresUnicos :  $sinRepetidos([id(d) \mid d \leftarrow enjambreDrones(s)])$ ;
  invariante unoPorParcela :  $(\forall d, d' \leftarrow dronesEnVuelo(s), id(d) \neq id(d')) posicionActual(d) \neq posicionActual(d')$ ;
  invariante siNoVuelanEstanEnGranero :  $(\forall d \leftarrow enjambreDrones(s), \neg enVuelo(d))$ 
     $posicionActual(d) == posicionGranero(campo(s))$ ;
  invariante siEstanEnVueloElVueloEstaEnRango :  $(\forall d \leftarrow dronesEnVuelo(s))(\forall v \leftarrow vueloRealizado(d))$ 
     $enRango(dimensiones(campo(s), prm(v), sgd(v))$ ;
}

aux dronesEnVuelo (s: Sistema) : [Drone] =  $[d \mid d \leftarrow enjambreDrones(s), enVuelo(d)]$ ;

problema crearS (c: Campo, ds: [Drone]) = res : Sistema {
}

problema campoS (s: Sistema) = res : Campo {
}

problema estadoDelCultivoS (s: Sistema, i, j:  $\mathbb{Z}$ ) = res : EstadoCultivo {
}

problema enjambreDronesS (s: Sistema) = res : [Drone] {
}

problema crecerS (s: Sistema) {
}

problema seVinoLaMalezaS (s: Sistema, ps:  $[(\mathbb{Z}, \mathbb{Z})]$ ) {
}

problema seExpandePlagaS (s: Sistema) {
}

problema despegarS (s: Sistema, d: Drone) {
}

problema listoParaCosecharS (s: Sistema) = res : Bool {
}

problema aterrizarYCargarBateriaS (s: Sistema, b:  $\mathbb{Z}$ ) {
}

problema fertilizarPorFilas (s: Sistema) {
}

problema volarYSensarS (s: Sistema, d: Drone) {
}
```

5. Funciones Auxiliares

5.1. Campo

5.2. Drone

5.3. Sistema