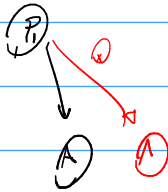


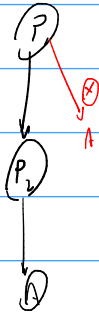
Tests

17



```
register: anAsset
  anAsset parent = self ifTrue:[ ^CantRegister signal].
  (assets includes: anAsset) ifTrue:[ ^CantRegister signal].
  assets add: anAsset
```

18

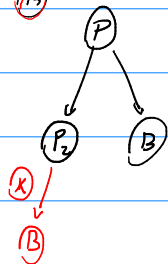


```
register: anAsset
  (assets includes: anAsset) ifTrue:[self error: self class cantRegister.].
  (self hasAlreadyTheAsset: anAsset) ifTrue:[self error: self class cantRegister.].
  assets add: anAsset
```

```
hasAlreadyTheAsset: anAssetToRegister
  ^assets anySatisfy: [:anAsset |
    anAsset = anAssetToRegister
    or:[anAsset hasAlreadyTheAsset: anAssetToRegister]]
```

Nos fijamos en el mismo nivel y los niveles inferiores

19



Vamos hasta arriba de todo y luego nos fijamos los niveles inferiores

```
register: anAsset
  (self hasAlreadyTheAsset: anAsset) ifTrue:[self error: self class cantRegister.].
  self assertCanAdd: anAsset.
  assets add: anAsset.
  (anAsset isKindOf: Portfolio) ifTrue:[anAsset setParent: self].
```

```
assertCanAdd: anAsset
  ^ (self parentHasAlreadyTheAsset: anAsset) ifTrue:[self error: self class cantRegister ]
```

```
hasAlreadyTheAsset: anAssetToRegister
  parent isNil ifTrue:[
    ^assets anySatisfy: [:anAsset | anAsset = anAssetToRegister or:[anAsset
    hasAlreadyTheAsset: anAssetToRegister]]
  ].
  parent hasAlreadyTheAsset: anAssetToRegister.
```

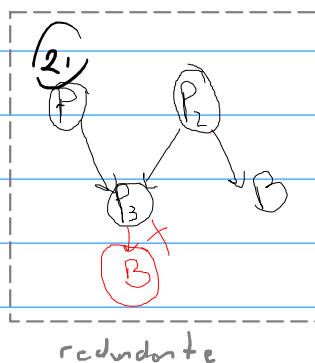
Esto significa seguir subiendo hasta una raíz

rename de
parentHas...

20



```
assertCanAdd: anAsset
  (self = anAsset or:[self parentHasAlreadyTheAsset: anAsset]) ifTrue:[self error: self
  class cantRegister ]
```



```

rename
assertCanAdd: anAsset

(self = anAsset or: [self isAlreadyTheAsset: anAsset]) ifTrue: [self error: self class
cantRegister ]

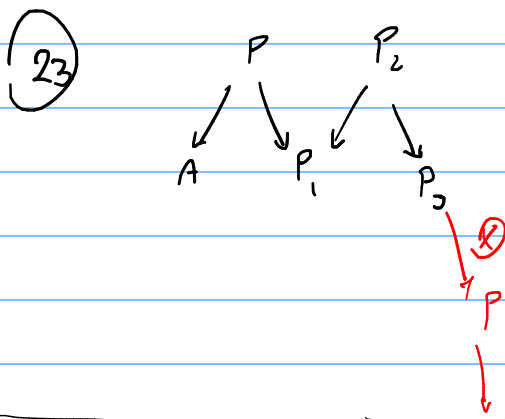
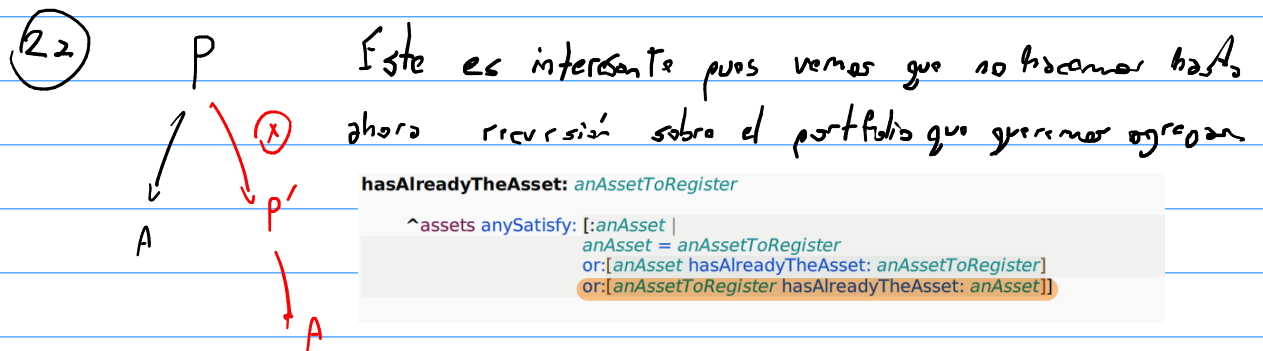
isAlreadyTheAsset: anAssetToRegister
^self rootParents anySatisfy: [:aRootParent | aRootParent hasAlreadyTheAsset:
anAssetToRegister].

```

Solo buscamos padres que son raiz

La implementacion de rootParents es igual a la de la catedral.

- Hasta test 21, decidimos refactorizar la manera de iterar sobre los padres de los portfolios. Es decir, creamos el mensaje #rootParents y a cada uno enviamos el msg. #hasAlreadyTheAsset
- Renombramos #parentHasAlreadyTheAsset por #isAlreadyTheAsset, pues verifica en todo el arbol y no solo por el padre (a pesar de que al final voy hasta las hojas y luego bajo).



tests de catedral:

