

Trabajo Práctico para Ingeniería de Software

Franco Sansone

1. Requerimientos

El usuario puede asociar un cliente (paciente si estuviese relacionado a la medicina) por nombre y DNI a un día y horario. Puede dar de baja un turno.

Se puede filtrar por fecha, nombre y DNI.

Podría ser usado por cualquiera que desarrolle sus actividades mediante turnos (médicos, docentes que den clases particulares, psicólogos, etc).

2. Especificación

Designaciones.

d es un DNI $\approx d \in DNI$

n es un nombre $\approx n \in NAME$

f es una fecha y hora $\approx f \in DATETIME$

El nombre de la persona con DNI $k \approx clientes\ k$

La fecha y hora del turno de la persona con DNI $k \approx turnos\ k$

Entonces introducimos los siguientes tipos básicos.

$[DNI, NOMBRE, DATETIME]$

Ahora podemos definir el espacio de estados de la especificación de la siguiente forma.

$AgendaDeTurnos$

$clientes : DNI \rightarrow NOMBRE$

$turnos : DATETIME \rightarrow DNI$

El estado inicial de la agenda de turnos es el siguiente.

$AgendaDeTurnosInit$

$AgendaDeTurnos$

$clientes = \emptyset$

$turnos = \emptyset$

Primero modelo el esquema que describe los predicados que deberían ser invariantes de estado.

<i>AgendaDeTurnosInv</i>
<i>AgendaDeTurnos</i>
$\text{dom clientes} = \text{Im turnos}$

Avanzo con la primera operación: asignar un turno.

<i>AsignarTurnoPrimeraVez</i>
$\Delta \text{AgendaDeTurnos}$
$\text{nombre?} : \text{NAME}$
$\text{dni?} : \text{DNI}$
$\text{fecha?} : \text{DATETIME}$
$\text{ahora?} : \text{DATETIME}$
$\text{dni?} \notin \text{dom clientes}$
$\text{fecha?} \notin \text{dom turnos}$
$\text{fecha?} > \text{ahora?}$
$\text{clientes}' = \text{clientes} \cup \{\text{dni?} \mapsto \text{name?}\}$
$\text{turnos}' = \text{turnos} \cup \{\text{fecha?} \mapsto \text{dni?}\}$

El argumento de entrada *ahora?* no necesariamente tiene que ser una entrada ingresada por el usuario. El pasaje de este argumento a la subrutina debería ser transparente para el usuario.

<i>AsignarTurnoCliente</i>
$\Delta \text{AgendaDeTurnos}$
$\text{dni?} : \text{DNI}$
$\text{fecha?} : \text{DATETIME}$
$\text{ahora?} : \text{DATETIME}$
$\text{dni?} \in \text{dom clientes}$
$\text{fecha?} \notin \text{dom turnos}$
$\text{fecha?} > \text{ahora?}$
$\text{turnos}' = \text{turnos} \cup \{\text{fecha?} \mapsto \text{dni?}\}$
$\text{clientes}' = \text{clientes}$

<i>ErrorFechaPasada</i>
$\Xi \text{AgendaDeTurnos}$
$\text{fecha?} : \text{DATETIME}$
$\text{ahora?} : \text{DATETIME}$
$\text{fecha?} \leq \text{ahora?}$

<i>TurnoYaAsignado</i>
$\Xi \text{AgendaDeTurnos}$
$\text{fecha?} : \text{DATETIME}$
$\text{fecha?} \in \text{turnos}$

$AsignarTurnoOk == AsignarTurnoPrimeraVez \vee AsignarTurnoCliente$
 $AsignarTurnoError == ErrorFechaPasada \vee TurnoYaAsignado$
 $AsignarTurno == AsignarTurnoOk \vee AsignarTurnoError$

Las siguientes operaciones corresponden a la búsqueda de turnos por nombre, DNI y fecha.

<i>BusquedaPorDNI</i> Ok
$\exists AgendaDeTurnos$
$dni? : DNI$
$resp! : \mathbb{P} DATETIME$
$dni? \in \text{dom } clientes$
$resp! = \text{dom}(turnos \triangleleft \{dni?\})$

<i>ClienteNoExisteDNI</i>
$\exists AgendaDeTurnos$
$dni? : DNI$
$dni? \notin \text{dom } clientes$

$BusquedaPorDNI == BusquedaPorDNI\text{Ok} \vee \text{ClienteNoExisteDNI}$

<i>BusquedaPorNombre</i> Ok
$\exists AgendaDeTurnos$
$nombre? : NOMBRE$
$resp! : DATETIME \rightarrow DNI$
$nombre? \in \text{ran } clientes$
$resp! = turnos \triangleright (\text{dom}(clientes \triangleright \{nombre?\}))$

<i>ClienteNoExisteNombre</i>
$\exists AgendaDeTurnos$
$nombre? : NOMBRE$
$nombre? \notin \text{Im } clientes$

$BusquedaPorNombre == BusquedaPorNombre\text{Ok} \vee \text{ClienteNoExisteNombre}$

<i>BusquedaPorFecha</i>
$\exists AgendaDeTurnos$
$dia? : \mathbb{P} DATETIME$
$resp! : DATETIME \rightarrow DNI$
$resp! = dia? \triangleleft turnos$

3. Simulaciones

La primera simulación es la siguiente:

```
agendaDeTurnosInit(S0) &
asignarTurno(S0,'Franco',37449292,201904011900,201903301900,S1) &
asignarTurno(S1,'Oscar',22099399,201904051730,201903301901,S2) &
asignarTurno(S2,'',22099399,201904011900,201903301903,S3) &
busquedaPorDni(S3,37449292,R1,S4) & busquedaPorNombre(S4,'Oscar',R2,S5).
```

cuyo primer resultado es:

```
S0 = {[clientes,{ }],[turnos,{ }]},
S1 = {[clientes,{[37449292,Franco]}],[turnos,{[201904011900,37449292]}]},
S2 = {[clientes,{[37449292,Franco],[22099399,Oscar]}],
      [turnos,{[201904011900,37449292],[201904051730,22099399]}]},
S3 = {[clientes,{[37449292,Franco],[22099399,Oscar]}],
      [turnos,{[201904011900,37449292],[201904051730,22099399]}]},
S4 = {[clientes,{[37449292,Franco],[22099399,Oscar]}],
      [turnos,{[201904011900,37449292],[201904051730,22099399]}]},
S5 = {[clientes,{[37449292,Franco],[22099399,Oscar]}],
      [turnos,{[201904011900,37449292],[201904051730,22099399]}]},
S6 = {[clientes,{[37449292,Franco],[22099399,Oscar]}],
      [turnos,{[201904011900,37449292],[201904051730,22099399]}]},
R1 = {201904011900},
R2 = {[201904051730,22099399]},
R3 = {[201904011900,37449292]},
```

Esta es la segunda simulación:

```
S0={ [turnos,
      { [201904011900,37449292],
        [201904011920, 33456787],
        [201904011940, 32890789],
        [201904031900,11235678]
      }],
      { [37449292,'Franco'],
        [33456787,'Ramiro'],
        [11235678,'Floresencia']
      }
    } &
busquedaPorFecha(S0,{201904011900,201904011920,201904011940},R,S1).
```

cuya ejecución devolvió:

```
S0 = {[turnos,
      {[201904011900,37449292],
        [201904011920,33456787],
        [201904011940,32890789],
        [201904031900,11235678]}],
      {[37449292,Franco],[33456787,Ramiro],[11235678,Floresencia]}],
```

```

R = {[201904011900,37449292],[201904011920,33456787],[201904011940,32890789]},
S1 = {[turnos,{[201904011900,37449292],[201904011920,33456787],
[201904011940,32890789],[201904031900,11235678]}],
{[37449292,Franco],[33456787,Ramiro],[11235678,Florencia]}}

```

4. Demostraciones con $\{log\}$

Primera demostración con $\{log\}$. Demuestro que *AsignarTurno* preserva el invariante *AgendaDeTurnosInv*, o sea el siguiente teorema:

theorem AsignarTurnoPI

$$AgendaDeTurnosInv \wedge AsignarTurno \Rightarrow AgendaDeTurnosInv'$$

el cual en $\{log\}$ se escribe de la siguiente forma:

```

S = {[clientes,C],[turnos,T]} &
S_ = {[clientes,C_],[turnos,T_]} &
ran(T,It) &
dom(C, It) &
asignarTurno(S,B,K,M,H,S_) &
ran(T_,It_) &
ndom(C_,It_).

```

Segunda demostración con $\{log\}$. Demuestro que *AsignarTurno* preserva el invariante $turnos \in DATETIME \rightarrow DNI$, o sea el teorema:

theorem TurnosIsPfun

$$turnos \in DATETIME \rightarrow DNI \wedge AsignarTurno \Rightarrow turnos' \in DATETIME \rightarrow DNI$$

el cual en $\{log\}$ se escribe de la siguiente forma:

```

S = {[turnos,T],[clientes,C]} &
S_ = {[turnos,T_],[clientes,C_]} &
pfun(T) &
asignarTurno(S,F,J,M,K,S_) &
npfun(T_).

```

5. Demostración con Z/EVES

Para esta prueba sustituí la precondition de $dni? \in \text{dom} \text{clientes}$ por $(dni? \mapsto \text{nombre?}) \in \text{clientes}$ en *AsignarTurnoCliente* (obviamente, agregando *nombre? \in NOMBRE* como parámetro de entrada).

theorem AsignarTurnoPI

$$AgendaDeTurnosInv \wedge AsignarTurno \Rightarrow AsignarTurnoPI'$$

proof[AsignarTurnoPI]
 invoke AsignarTurno;
 split AsignarTurnoOk;
 split AsignarTurnoPrimeraVez;
 cases;
 prove by reduce;
 next;
 split AsignarTurnoCliente;
 cases;
 prove by reduce;
 next;
 prove by reduce;
 next;
 ■