

Segundo Parcial

- **NO OLVIDAR PONER EL NOMBRE Y APELLIDO EN CADA HOJA**
- **NUMERAR LAS HOJAS CON LA FORMA <número de hoja>/<cantidad total de hojas>**
- Leer el examen completo antes de empezar a resolver los ejercicios, ya que ayuda a comprender mejor el dominio. Se puede consultar cualquier material (en papel) que haya sido escrito antes de comenzar el examen y usar sin definir todas las funciones y procedimientos vistos durante la cursada.
- Pensar bien la estrategia a seguir, consultar lo que no se entienda y recordar que el parcial es una instancia más de aprendizaje donde algún docente va a dar una devolución personalizada de todo lo que se escriba, es necesario para aprovechar esta instancia como algo más que solamente un momento para sacar una nota.

GobSalud

Gobsalud es un sistema para prestaciones medicas para obras sociales y prepagas, que actua de forma rapida y eficiente al proveer tipo de prestaciones medicas fijas. La empresa se encuentra en crecimiento y esta informatizando sus procesos para lo cual ha realizado un sistema en Gobstone, en donde se han definido los siguientes tipos.

```
type EmpresaMedica is record {
/*
Proposito: Modelar una empresa de prestaciones medicas
In Rep:
*No hay dos prestaciones en **prestaciones** con el mismo id
*MNo hay dos socios en **socios** con el mismo numero de dni
*Todas las autorizaciones de los socios en **socios** tiene un nombrePrestador y un tipo que
coincide con el nombre y el tipo de prestadora en el campo "prestaciones" que tiene el id
igual al idDePrestaciones de dicha autorizacion
*Todas las autorizaciones en los socios en "socios" tiene un numero de destino de "idDePrestacion"
que ademas coincide con el id de alguna prestacion del campo "prestaciones"
*/
field prestaciones // [Prestacion]
field socios // [socio]
}
```

Donde el campo “prestaciones” contiene la lista de servicios (medicamentos, estudios, recetas, derivaciones, que la empresa ofrece a sus afiliados el campo “socio” contiene la lista de afiliados a la empresa

De cada prestación ofrecida sabemos el nombre del prestador, la cantidad de bonos que se le pedirá al socio para atenderlo, el tipo de prestación y el identificador único que tiene dicha prestación.

```
type Prestacion is record {
/*
Proposito: Modelar una prestacion medica
In Rep:
* **nombreDePrestador** no es un string vacio
* **id** es mayor a cero
* **costoENBonos** es mayor o igual a cero
*/
field id // numero
field nombrePrestador // string
field costoEnBonos // Numero
field tipo // TipoDePrestacion
}
```

```
type Pedido is record {
/*
Proposito: Modelar un pedido de prestacion medica
*/
field tipo //TipoDePrestacion
field diagnostico // string
}
type Autorizacion is record {
/*
Proposito: Modelar una autorizacion de prestacion medica
*/
field tipo //TipoDePrestacion
field diagnostico // string
field idDePrestacion //Numero
field nombrePrestador // string
}
```

```

type Socio is record {
/*
Proposito: Modelar un socio de la empresa de salud
Inv.Rep:
* **dniSocio** no es un string vacio
* **cantidadDeBonos** es mayor o igual a cero
*/
field dniSocio      // String
field cantidadDeBonos // Numero
field solicitudesPendientes // [Pedido]
field autorizacionesDadas // [autorizaciones]
}

```

```

type TipoDePrestacion is variant {
/*
Proposito: Modelar un socio de la empresa de salud
Inv.Rep:
* **dniSocio** no es un string vacio
* **cantidadDeBonos** es mayor o igual a cero
*/
case Consulta   {}
case Receta    {}
case Derivacion {}
case Estudio   {}
}

```

De cada socio además del DNI sabemos cuántos bonos tiene disponible los pedidos de prestaciones por autorizar y las prestaciones ya autorizadas antes. Usaremos este modelo como base realice las funciones y procedimientos a continuación.

Punto 1)

En una empresa surgió que una prestadora compró a otra por lo cual debemos reemplazar todas las referencias a la vieja prestadora por referencias a la nueva esto incluye no sólo las prestaciones sino también todas las autorizaciones que contengan a la prestadora original. Para realizar esta tarea queremos definir la función **empresa_ConCambioDePrestadora_Por_** que dada una empresa de prestaciones médicas, un string con el nombre de la prestadora comprada y un string con el nombre de la prestadora que la compró, describa una nueva empresa en donde se realizó el cambio de todas las autorizaciones que eran de la prestadora comprada, por autorizaciones donde la prestadora es la que la adquirió. No olvidar también modificar el listado de prestaciones, para que todas las prestaciones figuren ahora bajo la nueva prestadora

Puede hacer uso de la siguiente función para ayudarse a solucionar el problema

```

function prestaciones_ConPrestadora_CambiadaA_(prestaciones,prestadoraVieja,prestadoraNueva){
/*
Propósito: describe la lista de prestaciones dadas donde cada prestación cambia su prestadora por **prestadoraNueva** sólo si su prestadora era **prestadoraVieja** sino queda como estaba.
Parámetros:
*prestaciones      [Prestacion]
*prestadoraVieja  String
*prestadoraNueva  String
Precondicion; ninguna
Tipo: [Prestación]
*/
}

```

Punto 2)

Escriba la función **prestacionesEn_ConDerivacionesDeSocioConDni_** que he dado una empresa médica y el DNI de un socio, describa la lista de prestaciones que tiene autorizada el socio dado que son del tipo Derivación. Tener en cuenta que puede no haber un socio con el DNI dado en cuyo caso se espera una lista vacía por resultado.

Si desea puede servirse de la siguiente primitiva para completar la tarea:

```

function socio_En_(dniSocio, empresa){
/*
Propósito: Describe el socio de la empresa **empresa** con dni **dniSocio**
Parámetros:
*dniSocio      String
*empresa       EmpresaMedica
Precondicion: hay al menos un socio en la empresa dada con el dni "dniSocio"
Tipo: Socio
*/
}

```

Punto 3)

Escriba la función **dnisDeSocioEn_ConPedidoDeTipo_ConDineroInsuficiente** que dada una empresa médica y un tipo de prestación, describa la lista de dnis de los socios que entre sus pedidos pendientes tengan alguna prestación del tipo dado y que no tienen bonos suficientes para realizar la totalidad de las prestaciones ya autorizadas.

Si lo desea puede servirse de la siguiente primitiva para completar la tarea:

```
function prestacionesAutorizadasDeSocioConDni_En_(dni, empresa){  
/*  
Propósito: Describe la lista de todas las prestaciones que el socio con el DNI dado tiene autorizadas en la empresa dada  
Parámetros:  
*dni: String  
*empresa EmpresaMedica  
Precondicion: hay al menos un socio con el dni dado en la empresa dada  
Tipo:  
*/  
}
```