

# Programación 4

## DSS Y CONTRATOS

### SOLUCIÓN LABORATORIO 2008

El sistema de gestión de reclamos debe satisfacer los requerimientos funcionales expresados por los siguientes casos de uso. Para todos ellos existe un único actor Administrador que interactúa con el sistema. Se pide realizar los Diagramas de Secuencia del Sistema para todos los casos de uso presentados (un único diagrama por caso de uso) y los contratos para las operaciones del sistema correspondientes.

Nombre	Alta de Técnico
Actores	Administrador
Sinopsis	El caso de uso comienza cuando el usuario Administrador desea ingresar un nuevo técnico en el sistema. Para ello indica la cédula, nombre y el tipo de técnico (jornalero o mensual). De ser jornalero debe ingresar la tarifa por hora y de ser empleado mensual debe ingresar el sueldo por mes. En caso de que la cédula no se encuentre registrada en el sistema, se da de alta el técnico. Finalmente, se agrega cada uno de los tipos de artículo en los cuales el técnico está capacitado para trabajar. En caso de que la cédula se encuentre registrada, el sistema genera un error.

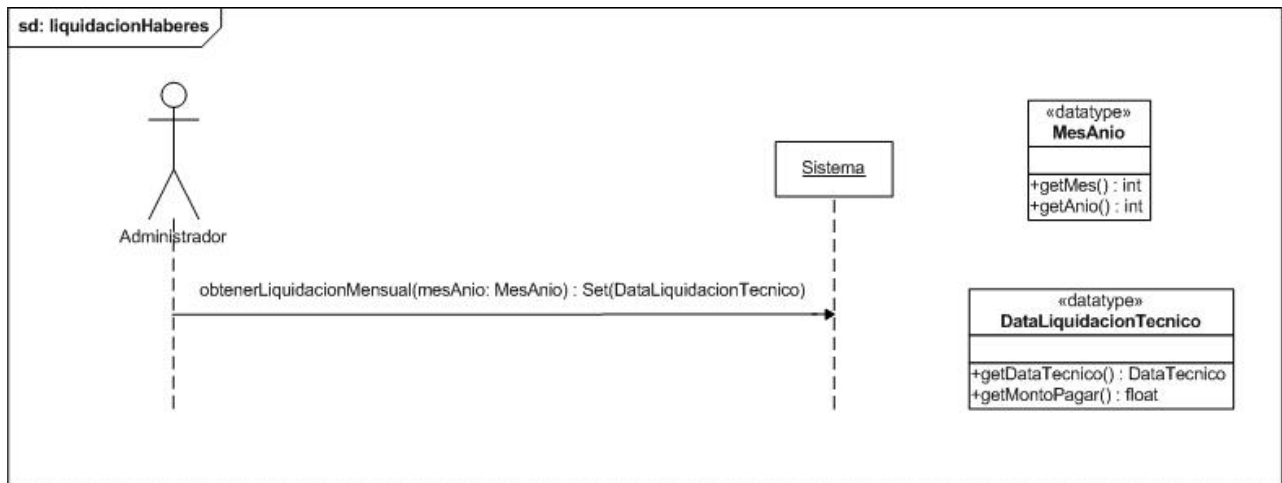
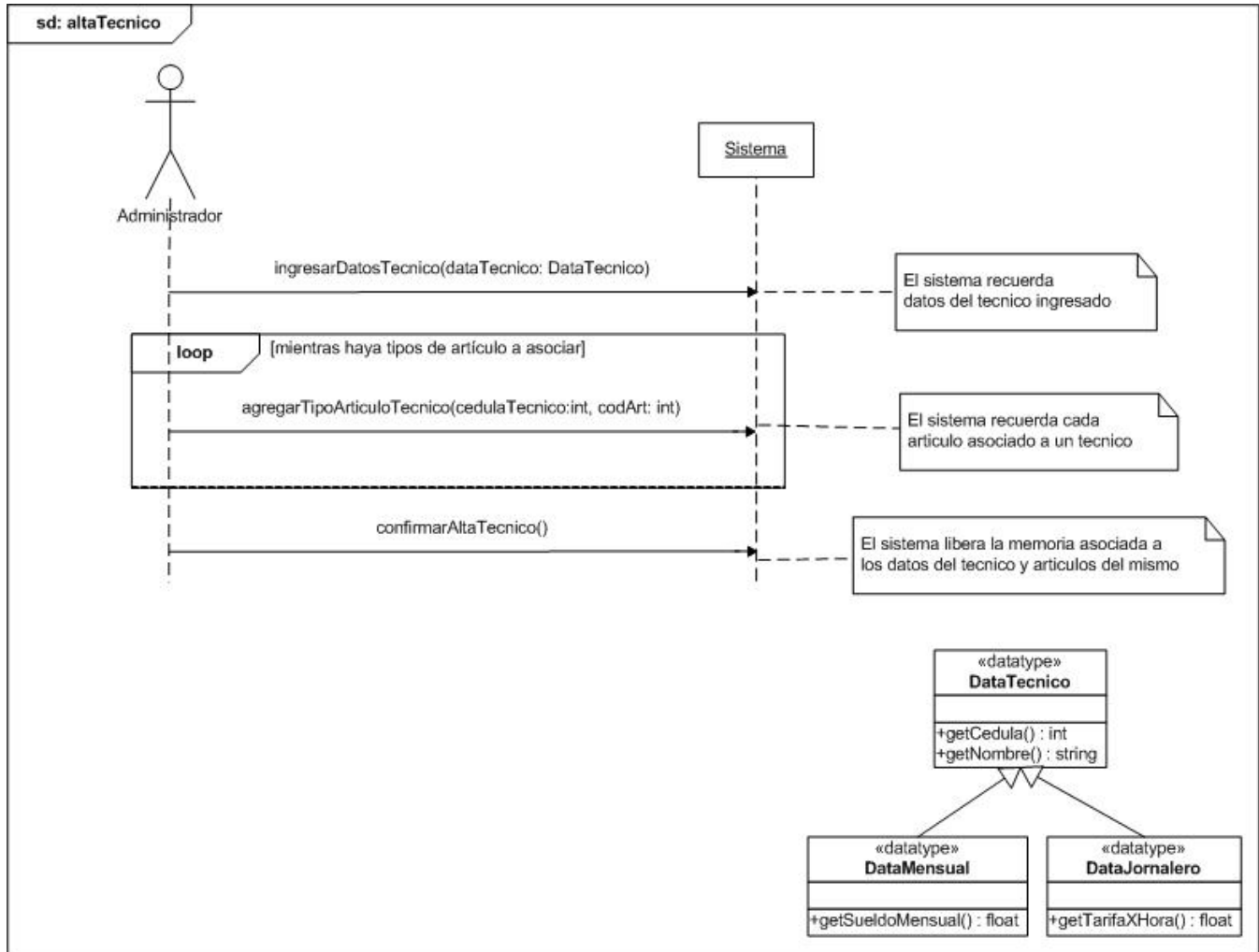
Nombre	Liquidación de haberes
Actores	Administrador
Sinopsis	El usuario ingresa el mes y año a liquidar y el sistema muestra una lista de los técnicos (ordenados alfabéticamente) y el monto a pagar en dicho mes a cada uno. Para los jornaleros dicho monto se calcula como el producto de las horas registradas en el mes y la tarifa por hora.

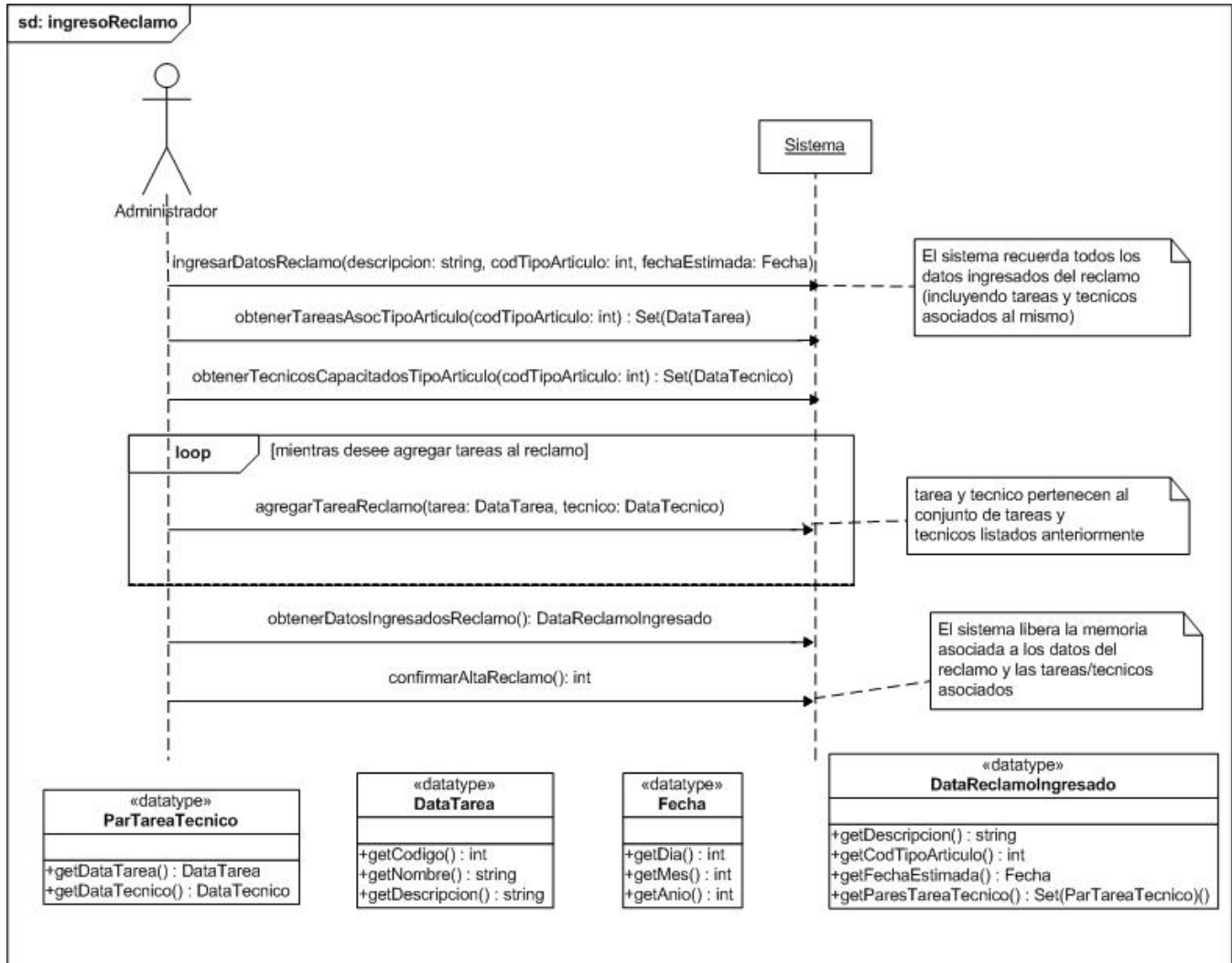
Nombre	Ingreso de Reclamo
Actores	Administrador
Sinopsis	El caso de uso comienza cuando el usuario Administrador desea ingresar un nuevo reclamo al sistema. Para ello indica la descripción del problema, el tipo de artículo a reparar y la fecha estimada de entrega. Posteriormente, el sistema muestra un listado de posibles tareas que se pueden realizar para el tipo de artículo a reparar. Luego, el usuario Administrador planifica la reparación ingresando una a una las tareas a realizar para el nuevo reclamo. El usuario deberá seleccionar, de entre los técnicos habilitados para trabajar en el tipo de artículo del reclamo, aquel que la realizará. Una vez ingresada la información del reclamo, el sistema muestra dicha información. Si el usuario confirma el ingreso, el sistema da de alta el reclamo y muestra el número generado.

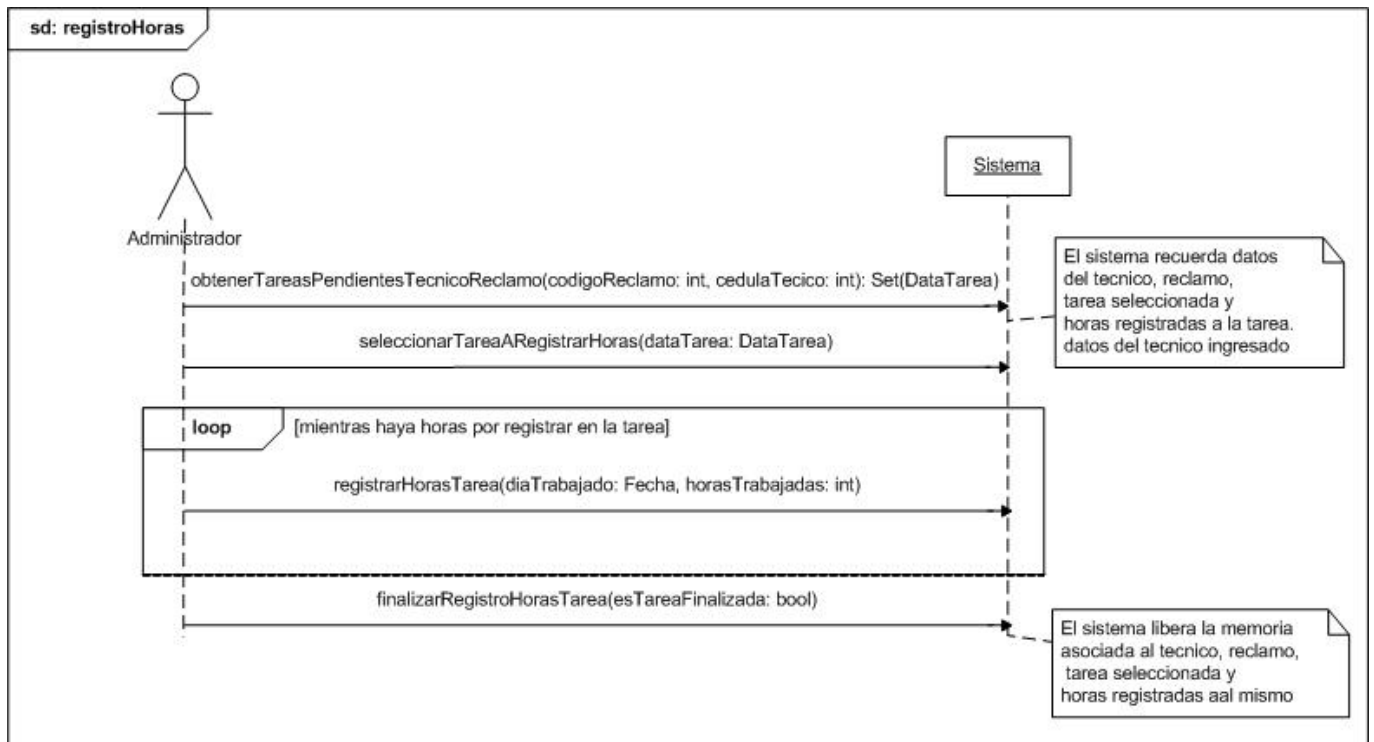
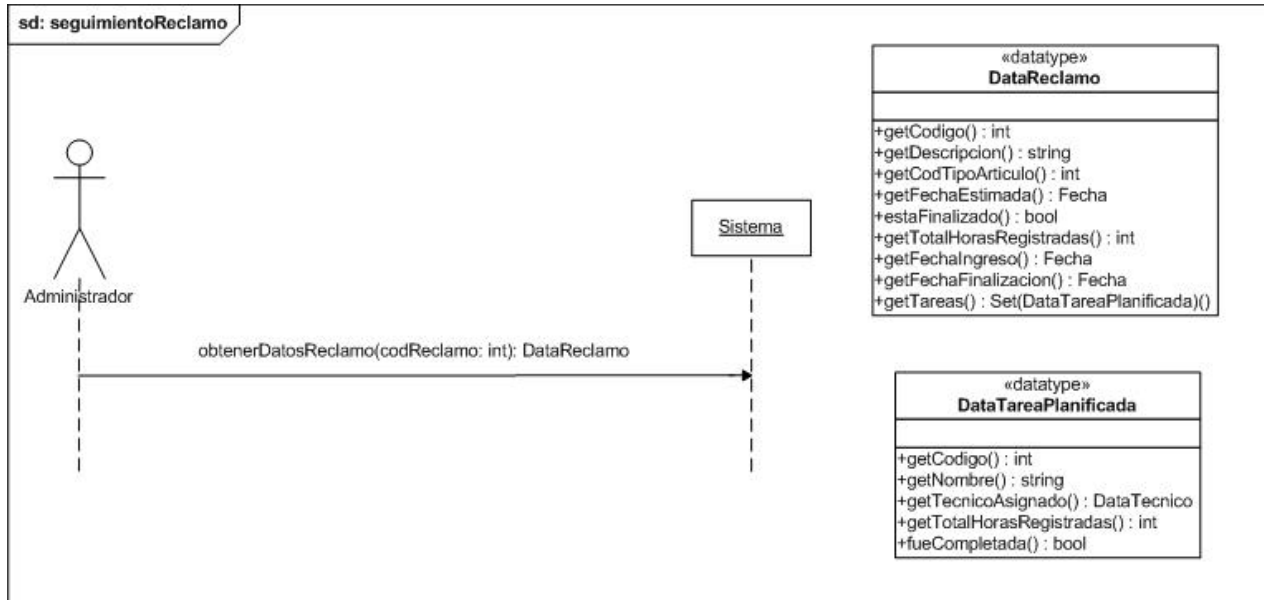
<b>Nombre</b>	<b>Cancelar Reclamo</b>
<b>Actores</b>	<b>Administrador</b>
<b>Sinopsis</b>	El caso de uso comienza cuando el usuario Administrador desea cancelar un reclamo. Para ello indica el número del mismo al sistema, que siempre y cuando no se hayan registrado horas para alguna de sus tareas, lo da de baja. En caso contrario, da un error indicando la causa. En caso de no existir un reclamo con ese número, el sistema avisa sobre ello.

<b>Nombre</b>	<b>Seguimiento de Reclamo</b>
<b>Actores</b>	<b>Administrador</b>
<b>Sinopsis</b>	El usuario ingresa el número de reclamo a consultar y el sistema muestra el número, la descripción, si fue completado o no, el total de horas registradas, la fecha de ingreso, la fecha estimada de entrega, la fecha de finalización del reclamo (si corresponde) y un listado de las tareas asignadas. Para cada tarea se muestra el código, el nombre, el técnico asignado, el total de horas registradas y si fue completada o no.

<b>Nombre</b>	<b>Registro de horas</b>
<b>Actores</b>	<b>Administrador</b>
<b>Sinopsis</b>	El usuario ingresa el número de reclamo y la cédula del técnico para el cual se desea registrar horas de trabajo. El sistema muestra una lista de tareas no completadas (en el orden que fueron asignadas) para el técnico ingresado y el usuario selecciona una de esas tareas. Luego, el usuario ingresa los pares de fecha y cantidad de horas a registrar para la tarea seleccionada. Al finalizar, el sistema pregunta si la tarea ha sido completada o no.







Nota: Se presentan solamente los contratos de los casos de uso Alta de Técnico, Registro de Horas y Cancelar Reclamo.

## Alta de Técnico

<b>Nombre</b>	Ingresar Técnico.
<b>Operación</b>	<code>ingresarDatosTecnico(data:DataTecnico)</code>
<b>Entrada</b>	<code>data</code> es una instancia de <code>DataMensual</code> o <code>DataJornalero</code> y con valores en sus atributos que indican la <code>ci</code> y nombre del técnico a ingresar así como su sueldo mensual o su tarifa por hora según corresponda.
<b>Salida</b>	No hay.
<b>Descripción</b>	Ingresar los datos del nuevo técnico que se ingresará al sistema.
<b>Excepciones</b>	Si no se cumple <code>pre1</code> se lanza la excepción <code>ExYaExisteTecnico</code> .

### Precondiciones y Postcondiciones

**Pre1:** No existe en el sistema una instancia de Técnico con cédula igual a `data.getCedula()`.

**Post1:** Se almaceno en la memoria del sistema la instancia de `DataTecnico` ingresada.

<b>Nombre</b>	Agregar tipo articulo a técnico
<b>Operación</b>	<code>agregarTipoArticuloTecnico(ciTecnico:int, codArt:int)</code>
<b>Entrada</b>	<code>ciTecnico</code> representa la cedula de un técnico que está almacenado en la memoria del sistema y <code>codArt</code> representa el código que identifica a uno de los tipos de artículos que existen ingresados en el sistema.
<b>Salida</b>	No hay.
<b>Descripción</b>	Asocia un tipo de articulo del sistema al los datos del técnico almacenado en la memoria del sistema.
<b>Excepciones</b>	Si no se cumple <code>pre1</code> se lanza la excepción <code>ExTecnicoNoSeteado</code> . Si no se cumple <code>pre2</code> se lanza la excepcion <code>ExNoExisteTipoArticulo</code> .

### Precondiciones y Postcondiciones

**Pre1:** Existe en la memoria del sistema una instancia de `DataTecnico`.

**Pre2:** Existe en el sistema una instancia de `TipoArticulo` con su atributo código `== codArt`.

**Post1:** Se asocio el valor de `codArt` a la instancia de `DataTecnico` con `ci == ciTecnico`.

<b>Nombre</b>	Confirmar el alta del técnico.
<b>Operación</b>	<code>confirmarAltaTecnico()</code>
<b>Entrada</b>	No hay.
<b>Salida</b>	No hay.
<b>Descripción</b>	Confirma el alta del técnico recordado por el sistema.
<b>Excepciones</b>	No hay.

### Precondiciones y Postcondiciones

**Pre1:** El sistema tiene almacenado en su memoria una instancia de `DataTecnico`.

**Post1:** Se creó en el sistema una instancia de `Jornalero` o de `Mensual` con los valores en su tipo y sus atributos correspondientes a la instancia de `DataTecnico` que existía en la memoria.

**Post2:** Se crearon links entre las instancias de `TipoArticulo` correspondientes a los códigos almacenados en la memoria del sistema y la instancia de `Técnico` que se creó en `Post1`.

**Post3:** Se libero la memoria del sistema.

## Registro de Horas

<b>Nombre</b>	<b>Obtener tareas pendientes del técnico en un reclamo.</b>
<b>Operación</b>	<code>obtenerTareasPendientesTecnicoReclamo(codReclamo:int, ciTecnico:int) : Set(DataTarea)</code>
<b>Entrada</b>	<code>ciTecnico</code> representa la cedula de un técnico del sistema y <code>codReclamo</code> representa el código que identifica a uno de los reclamos que existen en el sistema.
<b>Salida</b>	Un conjunto de instancias de <code>DataTarea</code> que representa las tareas pendientes del técnico ingresado.
<b>Descripción</b>	Retorna las tareas pendientes de un determinado reclamo para un determinado técnico.
<b>Excepciones</b>	Si no se cumple <code>pre1</code> se lanza la excepción <code>ExNoExisteTecnico</code> . Si no se cumple <code>pre2</code> se lanza la excepción <code>ExNoExisteReclamo</code> .

<b>Precondiciones y Postcondiciones</b>	
<p><b>Pre1:</b> Existe en la memoria del sistema una instancia de Técnico con <code>ci == ciTecnico</code>.</p> <p><b>Pre2:</b> Existe en el sistema una instancia de Reclamo con su atributo código == <code>codReclamo</code>.</p> <p><b>Post1:</b> Se construyo y retorno un conjunto de instancias de <code>DataTarea</code> con la información de las instancias de Tarea existentes en el sistema con su atributo <code>completada? == false</code> y que están linkeadas con la instancia de Reclamo con código = <code>codReclamo</code> y también con la instancia de Técnico con <code>ci = ciTecnico</code>.</p> <p><b>Post2:</b> Se almaceno en la memoria del sistema el valor de <code>codReclamo</code> y de <code>ciTecnico</code>.</p>	

<b>Nombre</b>	<b>Seleccionar tarea a registrar horas</b>
<b>Operación</b>	<code>seleccionarTareaARegistrarHoras(dataTarea:DataTarea)</code>
<b>Entrada</b>	<code>dataTarea</code> representa a la tarea del sistema a la que se le quiere agregar horas de dedicación.
<b>Salida</b>	No hay.
<b>Descripción</b>	Marca en la memoria del sistema la tarea a la que se le agregaran horas de dedicación
<b>Excepciones</b>	Si no se cumple <code>pre1</code> se lanza la excepcion <code>ExNoExisteTarea</code> .

<b>Precondiciones y Postcondiciones</b>	
<p><b>Pre1:</b> Existe en el sistema una instancia de Tarea con su atributo código == <code>codReclamo</code>.</p> <p><b>Post1:</b> Se almaceno en la memoria del sistema la instancia <code>dataTarea</code>.</p>	

<b>Nombre</b>	<b>Registrar horas de la tarea</b>
<b>Operación</b>	<code>registrarHorasTarea(dia:Fecha, horas:int)</code>
<b>Entrada</b>	<code>dia</code> representa el dia en que se realizo el trabajo y <code>horas</code> la cantidad de horas que se trabajo en la tarea.
<b>Salida</b>	No hay.
<b>Descripción</b>	Almacena en la memoria del sistema el ingreso de horas para un determinado día.
<b>Excepciones</b>	Si no se cumple <code>pre1</code> se lanza la excepción <code>ExMemoriaNoSeteada</code> .

Precondiciones y Postcondiciones	
Pre1: Existen en la memoria del sistema valores que representan un código de reclamo, una ci técnico y una instancia de DataTarea.	
Post1: Se agrego a la colección de dedicaciones el par (día,horas) en la instancia dataTarea en la memoria del sistema.	

Nombre	Finalizar registro de horas
Operación	<code>finalizarRegistroHorasTarea(finalizada:bool)</code>
Entrada	Finalizada indica que la tarea a la que se le agrego dedicación está finalizada.
Salida	No hay.
Descripción	Confirma el ingreso de las dedicaciones de horas a la tarea, reclamo y técnico elegido.
Excepciones	Si no se cumple pre1 se lanza la excepción ExMemoriaNoSeteada.

Precondiciones y Postcondiciones	
Pre1: Existen en la memoria del sistema valores que representan un código de reclamo, una ci técnico y una instancia de DataTarea.	
Post1: Se creó una instancia de la clase de asociación TareaReclamo con valor finalizada? = finalizada, entre la instancia de Reclamo con código = codReclamo (memoria) y la tarea con código = dataTarea.getCodigo() y se creó también un link entre esta a la instancia de Técnico con ci = ciTecnico (memoria); se creó una instancia de Registro por cada par (día, horas) con un link a la instancia de TareaReclamo creada.	
Post2: Se libero la memoria del sistema.	

## Cancelar Reclamo

Nombre	Cancelar reclamo
Operación	<code>cancelarReclamo(codReclamo:int)</code>
Entrada	codReclamo es el codigo del reclamo que se quiere cancelar.
Salida	No hay.
Descripción	Cancela el reclamo con codigo codReclamo existente en el sistema si el mismo no tiene horas registradas.
Excepciones	Si no se cumple pre1 se lanza la excepción ExNoExisteReclamo. Si no se cumple pre2 se lanza la excepción ExReclamoConDedicacion.

Precondiciones y Postcondiciones	
Pre2: Existe en el sistema una instancia de Reclamo con código = codReclamo	
Pre1: No existe ninguna instancia del tipo asociativo TareaReclamo que linkea al reclamo con código = codReclamo con alguna tarea.	
Post1: Se elimino el link entre el reclamo con código = codReclamo y el tipo de artículo que el reclamo tenia asociado y posteriormente se elimino del sistema la instancia del reclamo indicada.	