



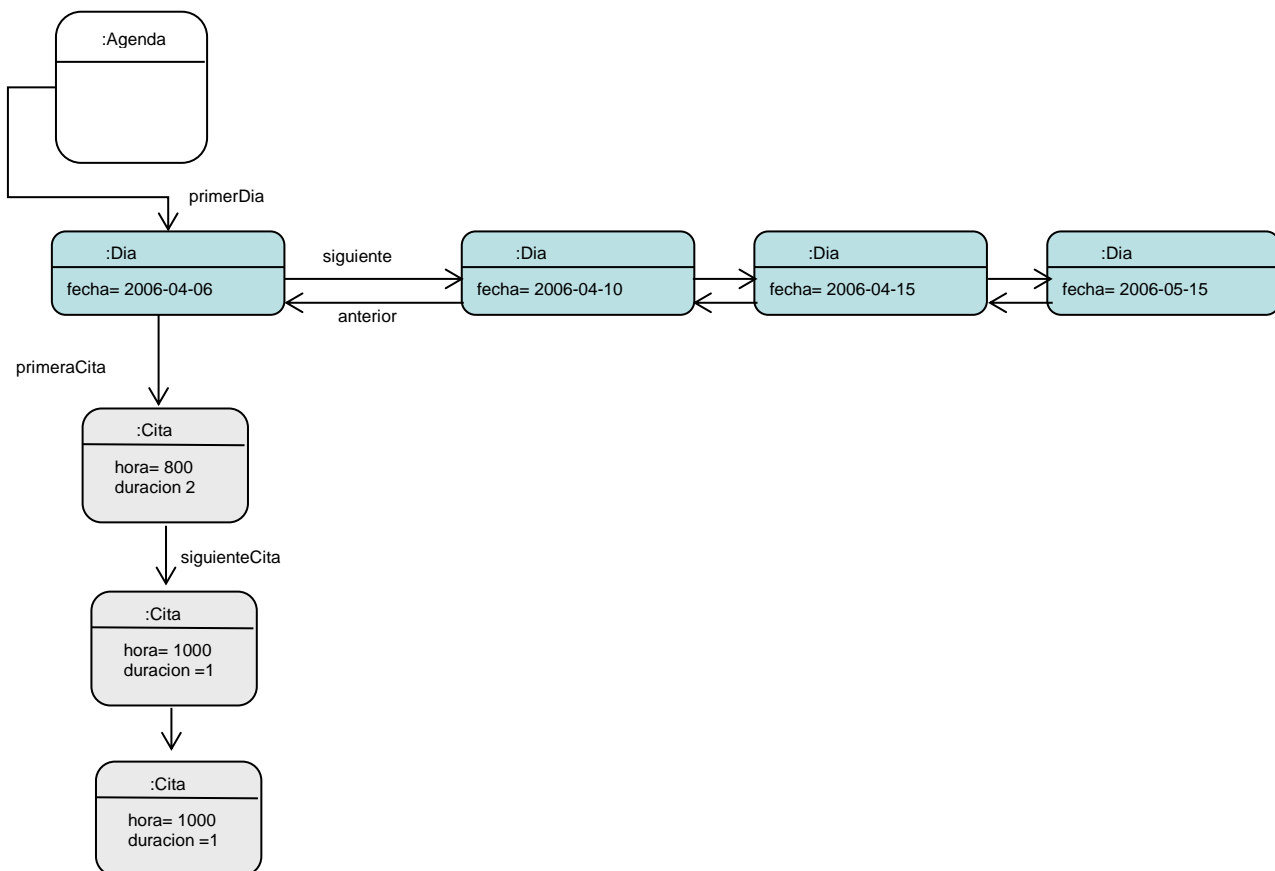
Universidad de los Andes
 Ingeniería de Sistemas y Computación
 ISIS 1205 - Algorítmica y Programación 2
 Hoja de trabajo teórica N9



La Agenda

- Estudie el enunciado del ejercicio de la agenda, el diseño de la interfaz usuario, los requerimientos funcionales y el diagrama de clases del mundo.
- Estudie el siguiente diagrama de objetos que representa gráficamente:
 - Una **agenda** que contiene **días**. Los días están almacenados en una lista doblemente encadenada.
 - Un **día** compuesto de **citas**. Las citas están almacenadas en una lista simplemente encadenada.

Ambas listas están ordenadas. La de **días** por fecha y la de **citas** por hora.




```
class Agenda {
    .....

/**
 * Crea un nuevo Dia en la agenda con la fecha indicada y lo deja en
 * su lugar dentro de la lista<br>
 * <b>pre:</b>En la agenda no hay ya un día con la fecha indicada<br>
 * <b>post:</b>Se agregó el Dia a la agenda y la lista de días siguió
 * ordenada por fecha
 * @param fecha La fecha que tendrá el día
 * @return Retorna el objeto Dia que fue creado
 */
public Dia crearDia( Date fecha )
{

}
}
```

```
class Dia {
    .....

/**
 * Agrega una nueva cita en la posición que le corresponde según la hora
 * @param nuevaCita La nueva cita que se va a agregar
 * @throws CitaInvalidaExcepcion Se lanza esta excepción si esta cita no
 * se puede poner en este día porque se cruza con otra cita
 */
public void agregarCita( Cita nuevaCita ) throws CitaInvalidaExcepcion
{

}
}
```

```
class Dia {
.....
/**
 * Elimina una cita del día dada su hora <br>
 * <b>post: </b>Se eliminó la cita que estaba programada para la hora
 * indicada
 * @param horaCita La hora de la cita que se va a eliminar
 * @throws CitaInexistenteExcepcion Se lanza esta excepción si no se
 * encontró la cita que se iba a eliminar
 */
public void eliminarCita( int horaCita ) throws CitaInexistenteExcepcion
{

}
}
```

```
class Dia
....
/**
 * Este método sirve para saber si una hora está libre para una cita de
 * una determinada duración
 * @param hora La hora a la que debería empezar la cita
 *   Dia.PRIMERA_HORA <= hora <= Dia.ULTIMA_HORA y 0 <= hora % 100 < 60
 * @param duración La duración en minutos de la cita - duracion > 0
 * @return Retorna true si hay espacio para acomodar una cita de la
 *   duración indicada en la hora indicada
 */
public boolean estaHoraLibre( int hora, int duracion )
{

}
}
```