

Programación Orientada a objetos

Técnicas Avanzadas
de
programación

- Para los métodos, elegir verbos en infinitivo o participios, con la misma convención que para los atributos y clases, pero sin colocar ninguna letra delante.
- Para los objetos, aplicamos la misma notación de los métodos en cuanto a la convención de mayúsculas y minúsculas.

Nos pareció que este estándar de nomenclatura era adecuado desde el punto de vista del cliente, que suele tratar con objetos y métodos más que con clases y atributos. Asimismo, nos pareció que cumplía una finalidad didáctica mayor que los anteriores.

De todas maneras, no importa demasiado qué estándar se utilice, sino que se utilice alguno, y sobre todo, con fundamentos y coherencia. Por otro lado, como el lenguaje que se está utilizando siempre tiene identificadores predefinidos, habría que tratar de que nuestro propio estándar no difiera demasiado del del lenguaje.

EJERCICIO RESUELTO: DIAGRAMAS DE ACTIVIDADES, CLASES Y SECUENCIA

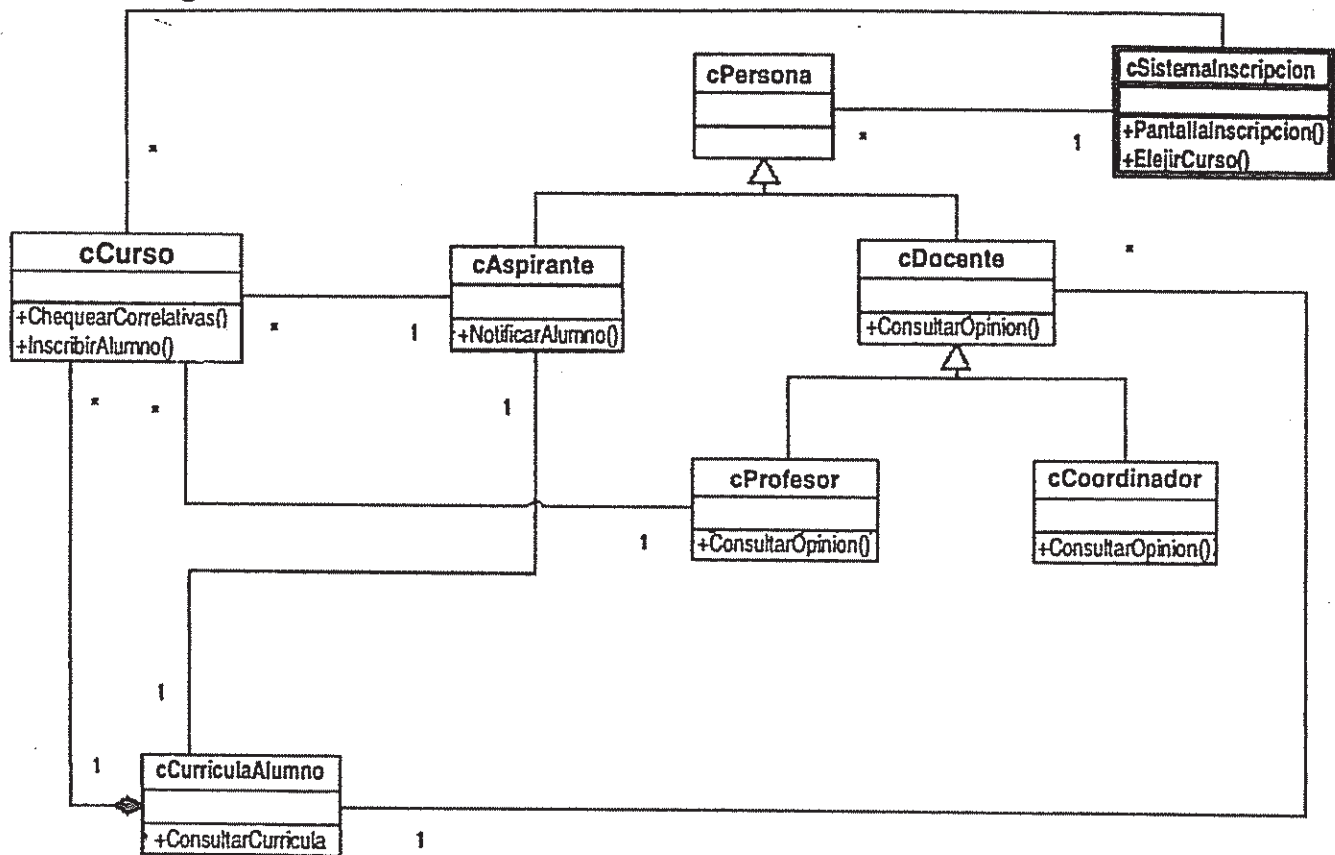
Enunciado

Hacer los diagramas de actividades (a nivel de análisis), de clases y de secuencia del proceso de anotación de un alumno en las materias de un postgrado. Para ello, se supondrá que cada alumno podrá anotarse en varios cursos, para los cuales habrá que verificar que tenga aprobadas las correlativas. En una segunda etapa, el profesor del curso y el coordinador del postgrado evaluarán la situación del alumno y aprobarán o no la solicitud, dado que pueden considerar que la orientación general que el aspirante le está dando a sus estudios no debería orientarlo en esa dirección.

Solución

Diagrama de actividades:

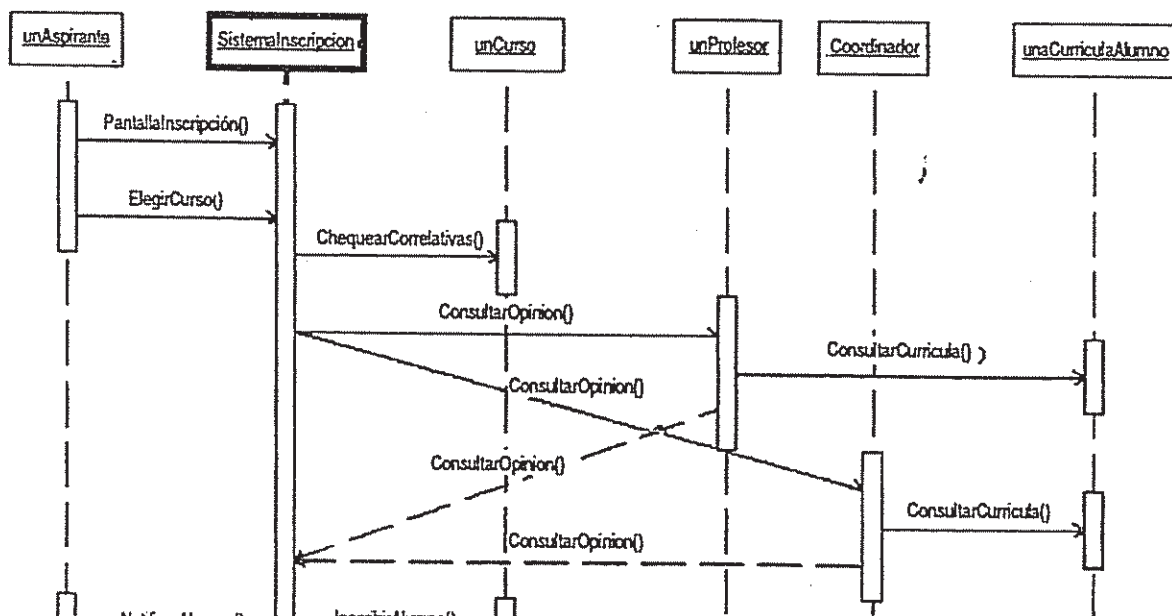
Diagrama de clases:



El contorno más grueso de clase *cSistemaInscripcion* indica que se trata de un objeto activo, que puede disparar llamadas concurrentes.

Debe notarse que, cuando el sistema de inscripción se comunica con una persona, sea ésta un aspirante o un docente, lo que hace es trabajar con un objeto cliente; que en realidad es una interfaz de usuario que ve la persona. Reconozcamos que esta relación fue la única razón que justificó la existencia de la clase *cPersona* en el diagrama.

Diagrama de secuencia:



El contorno más grueso del objeto *SistemaInscripcion* indica que se trata de un objeto activo, que puede disparar llamadas concurrentes. Precisamente, las llamadas a los dos métodos *ConsultarOpinion* se hacen asíncronas y en paralelo. Estos conceptos se estudiarán en el capítulo de concurrencia.

Nótese que no todos los estados del diagrama de actividades quedaron reflejados en el diagrama de secuencia, aunque sí los principales. También fue necesario introducir dos objetos nuevos, el curso y la currícula del alumno.

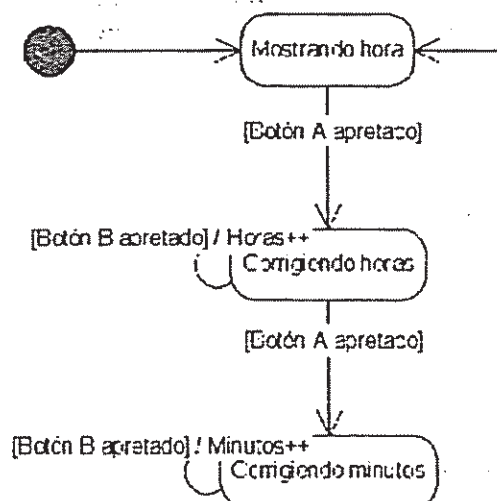
EJERCICIO RESUELTO: UN DIAGRAMA DE ESTADOS

Enunciado

Hacer el diagrama de estados - transiciones para el siguiente escenario:

- Un reloj digital tiene un visor y dos botones, el *botón A* y el *botón B*. El reloj tiene dos modos de operación, *mostrar hora* y *corregir hora*.
- En el modo mostrar hora, se muestran las horas y minutos, separados por dos puntos. El modo corregir hora tiene dos submodos, *corregir hora* y *corregir minutos*.
- El botón A es el usado para cambiar de modo. Cada vez que es presionado el modo avanza secuencialmente: mostrar hora, corregir hora, corregir minutos, etc.
- En cada submodo el botón B es usado para avanzar las horas o minutos cada vez que es presionado. Los botones deben ser soltados antes de generar algún otro evento.

Solución



***Programación
orientada a objetos
y técnicas avanzadas de
programación***

***Carlos Fontela
2003***

usar este estándar, que luego quedó en Object Pascal:

- ?? Para los tipos y clases, utilizar nombres extraídos del vocabulario del sistema, y escribirlos en minúsculas, utilizando las mayúsculas al comienzo de cada palabra y agregando una *T* delante, como en *TCanvas* y *TComponent*.
- ?? Para las interfaces se usan las mismas convenciones que para las clases, pero con una *I* en vez de la *T*.
- ?? Para los atributos, utilizar nombres de propiedades de la clase a la que pertenecen, escribiéndolos con las mismas convenciones de las clases, sin ninguna letra inicial.
- ?? Para los métodos, elegir verbos en infinitivo o participios, con la misma convención que para los atributos.
- ?? Para los objetos, la misma notación de los métodos y atributos en cuanto a la convención de mayúsculas y minúsculas.

En nuestro caso, hemos recurrido a un estándar consistente en:

- ?? Para las clases, utilizar nombres extraídos del vocabulario del sistema, y escribirlos en minúsculas, utilizando las mayúsculas al comienzo de cada palabra. Para enfatizar que se trata de una clase, hemos agregado una *c* adelante, como en *cComplejo*.
- ?? En las clases de excepción hemos reemplazado la *c* por una *e*.
- ?? Para las interfaces, usamos las mismas convenciones que para las clases, pero con una *i* minúscula en vez de la *c*.
- ?? Para los tipos que no son clases, utilizamos una *t* delante del nombre.
- ?? Para los atributos, utilizamos nombres de propiedades de la clase a la que pertenecen, escribiéndolos con las mismas convenciones de las clases, salvo que la primera letra es siempre una *a*.
- ?? Para los métodos, elegir verbos en infinitivo o participios, con la misma convención que para los atributos y clases, pero sin colocar ninguna letra delante.
- ?? Para los objetos, aplicamos la misma notación de los métodos en cuanto a la convención de mayúsculas y minúsculas.

Nos pareció que este estándar de nomenclatura era adecuado desde el punto de vista del cliente, que suele tratar con objetos y métodos más que con clases y atributos. Asimismo, nos pareció que cumplía una finalidad didáctica mayor que los anteriores.

De todas maneras, no importa demasiado qué estándar se utilice, sino que se utilice alguno, y sobre todo, con fundamentos y coherencia. Por otro lado, como el lenguaje que se está utilizando siempre tiene identificadores predefinidos, habría que tratar de que nuestro propio estándar no difiera demasiado del del lenguaje.

EJERCICIO RESUELTO: DIAGRAMAS DE ACTIVIDADES, CLASES Y SECUENCIA

Enunciado

Hacer los diagramas de actividades (a nivel de análisis), de clases y de secuencia del proceso de anotación de un alumno en las materias de un postgrado. Para ello, se supondrá que cada alumno podrá anotarse en varios cursos, para los cuales habrá que verificar que tenga aprobadas las correlativas. En una segunda etapa, el profesor del curso y el coordinador del postgrado evaluarán la situación del alumno y aprobarán o no la solicitud, dado que pueden considerar que la orientación general que el aspirante le está dando a sus estudios no debería orientarlo en esa dirección.

Solución

Diagrama de actividades:

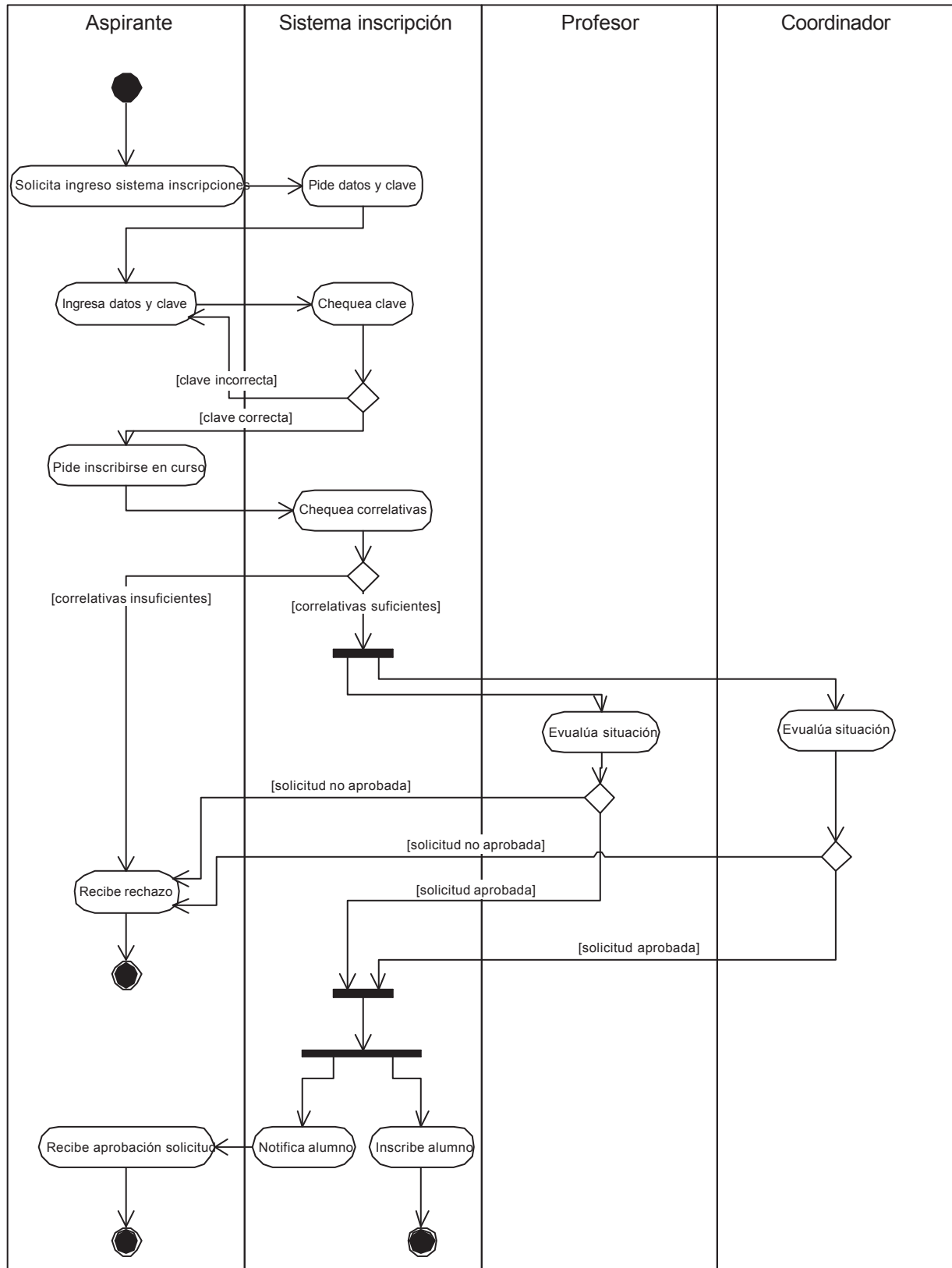
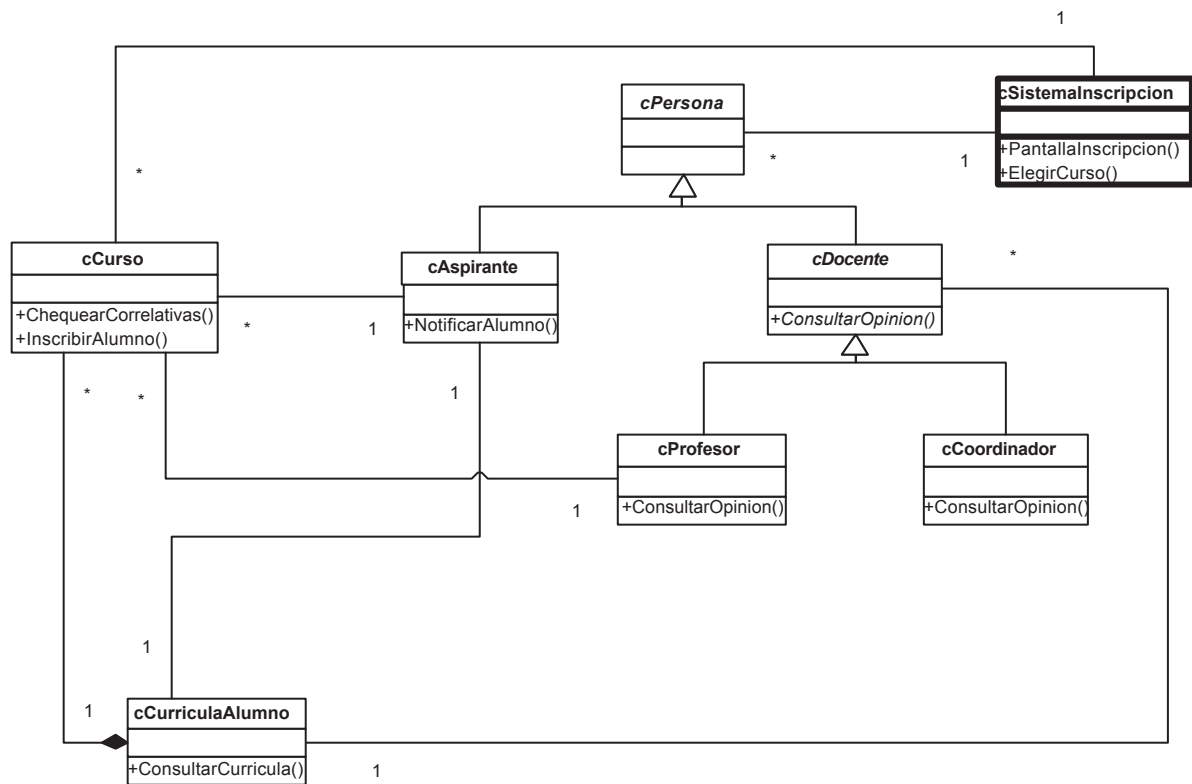


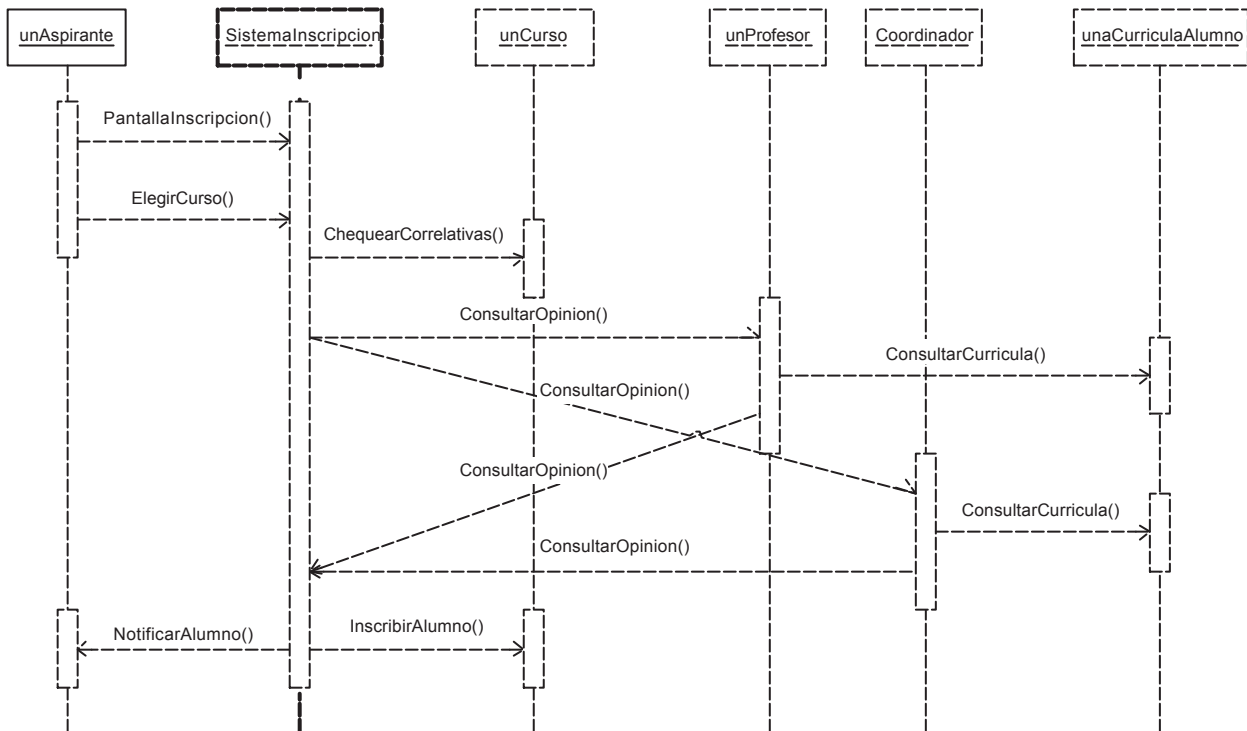
Diagrama de clases:



El contorno más grueso de clase *cSistemaInscripcion* indica que se trata de un objeto activo, que puede disparar llamadas concurrentes.

Debe notarse que, cuando el sistema de inscripción se comunica con una persona, sea ésta un aspirante o un docente, lo que hace es trabajar con un objeto cliente, que en realidad es una interfaz de usuario que ve la persona. Reconozcamos que esta relación fue la única razón que justificó la existencia de la clase *cPersona* en el diagrama.

Diagrama de secuencia:



El contorno más grueso del objeto *SistemaInscripcion* indica que se trata de un objeto activo, que puede disparar llamadas concurrentes. Precisamente, las llamadas a los dos métodos *ConsultarOpinion* se hacen asíncronas y en paralelo. Estos conceptos se estudiarán en el capítulo de concurrencia.

Nótese que no todos los estados del diagrama de actividades quedaron reflejados en el diagrama de secuencia, aunque sí los principales. También fue necesario introducir dos objetos nuevos, el curso y la currícula del alumno.

EJERCICIO RESUELTO: UN DIAGRAMA DE ESTADOS

Enunciado

Hacer el diagrama de estados - transiciones para el siguiente escenario:

- Un reloj digital tiene un visor y dos botones, el *botón A* y el *botón B*. El reloj tiene dos modos de operación, *mostrar hora* y *corregir hora*.
- En el modo *mostrar hora*, se muestran las horas y minutos, separados por dos puntos. El modo *corregir hora* tiene dos submodos, *corregir hora* y *corregir minutos*.
- El botón *A* es el usado para cambiar de modo. Cada vez que es presionado el modo avanza secuencialmente: *mostrar hora*, *corregir hora*, *corregir minutos*, etc.
- En cada submodo el botón *B* es usado para avanzar las horas o minutos cada vez que es presionado. Los botones deben ser soltados antes de generar algún otro evento.

Solución

