

# Prueba 1 de Evaluación Continua

## Análisis y Diseño de Software (2014/2015)

Contesta el ejercicio 1 en esta hoja, el 2 y el 3 en hojas separadas, cada uno en una hoja distinta

Apellidos:

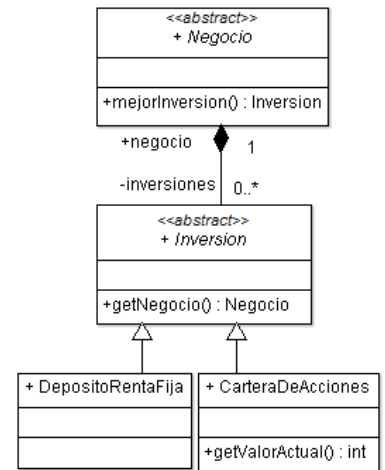
Nombre:

### Ejercicio 1: Herencia y polimorfismo (2 puntos)

Dado el diagrama de clases de la derecha, indica cuál de las siguientes líneas de pseudocódigo contiene algún **error**, e indica la razón:

1. Negocio neg1 = **crear Negocio**
2. Inversion **inv** = crear CarteraDeAcciones
3. DepositoRentaFija **dep** = **crear Inversion**
4. Negocio neg2 = **dep.getNegocio()**
5. int valor = neg2.mejorInversion().**getValorActual()**

El error en 1 y en 3 es el mismo, que no se puede instanciar una clase abstracta, y el error en 5 es que el retorno de mejorInversion() es una Inversion pero la clase Inversion no tiene definido el método getValorActual().

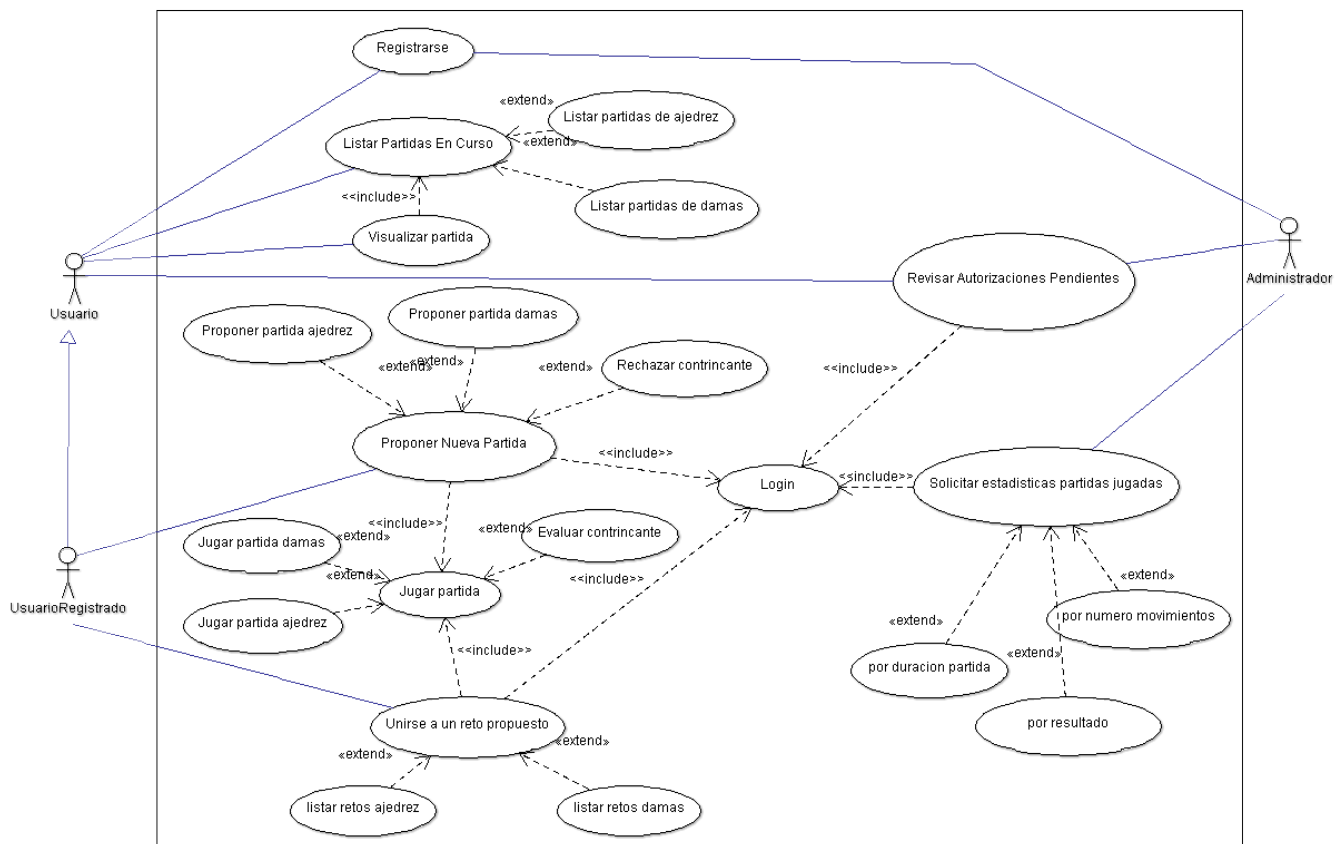


## Ejercicio 2: Diagramas de casos de uso (3.5 puntos)

Realiza el diagrama de casos de uso de una aplicación para jugar por Internet al ajedrez y a las damas.

Cualquier usuario puede listar y visualizar las partidas en curso (de ajedrez o damas), pero sólo los usuarios registrados pueden proponer una nueva partida (esto es, buscar un oponente), o empezar a jugar con otro usuario que ha empezado la búsqueda de oponente. El usuario que inicia la búsqueda de oponente puede rechazar jugar con un usuario dado. Una vez que termina la partida, cada usuario tiene la opción de valorar al contrincante (una nota del 0 al 10).

El administrador es responsable de validar las solicitudes de registro de nuevos usuarios, y para ello analizará frecuentemente las notificaciones que el sistema le envía sobre las aprobaciones pendientes. El administrador también podrá solicitar estadísticas sobre las partidas que se han jugado, que podrán organizarse por duración de la partida, resultado y número de movimientos de la partida.



### Ejercicio 3: Diagramas de clase (4.5 puntos)

Se quiere construir una aplicación para la gestión de *itinerarios turísticos* a diversas escalas territoriales. Los itinerarios más básicos se caracterizan por el sitio turístico visitado, el precio por visitante y la duración aproximada de la visita. Otros itinerarios más complejos podrán consistir en la visita en secuencia de varios itinerarios (básicos o complejos). Los sitios turísticos se identifican mediante un nombre y podrán ser de dos tipos: culturales y naturales. Para los culturales se indicará el número de días de antelación necesario para hacer la reserva y si son recomendables o no para niños menores de 7 años; para los naturales tendremos dos fechas que limitan el periodo ideal de visita y también se debe saber si incluyen o no actividades de riesgo. La aplicación también debe considerar sugerencias de pares de sitios turísticos que desaconseja realizar en un mismo itinerario, ya que se consideran muy similares; cada una de estas sugerencias se caracteriza por los dos sitios turísticos considerados similares, y además para cada uno de ellos se tiene una lista (posiblemente vacía) de sitios turísticos sugeridos como posibles sustitutos por proximidad geográfica e interés turístico complementario.

Se pide:

(a) Realiza el diagrama de clases que describe la aplicación anterior (3,75 puntos).

(b) Añade los métodos necesarios para (0,75 puntos):

1. Saber si la visita a un sitio turístico está desaconsejada dado un itinerario cualquiera.
2. Dados dos sitios turísticos, saber si se desaconseja realizarlos en el mismo itinerario.
3. Añadir un itinerario a otro previamente creado.

