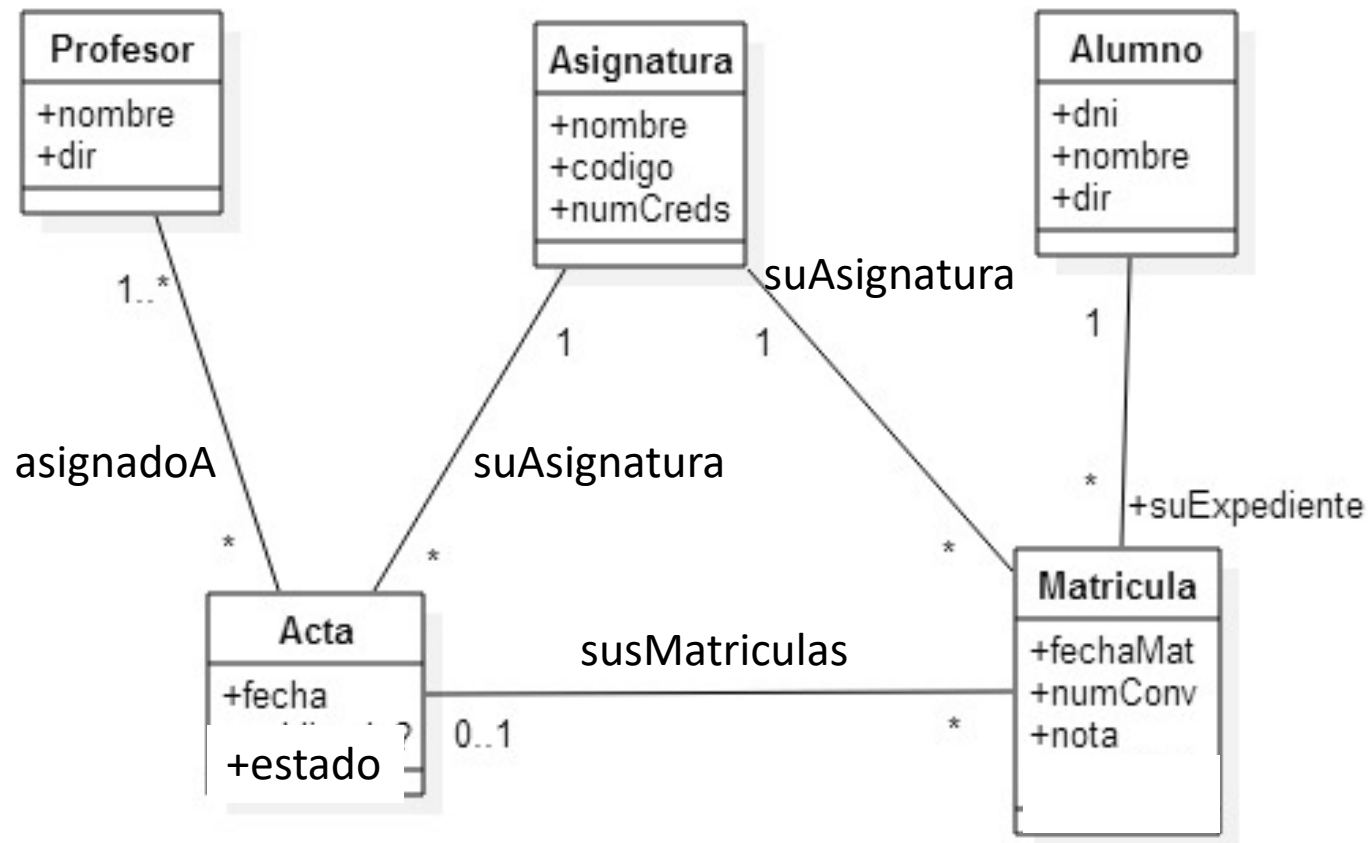


Diseño CU

“Visualizar Expediente”

Flujo de eventos: VISUALIZAR EXPEDIENTE

- El alumno introduce su dni
- Se obtienen todas las matrículas del mismo
- Por cada matrícula se muestra la asignatura con la calificación obtenida, siempre que se haya publicado el acta

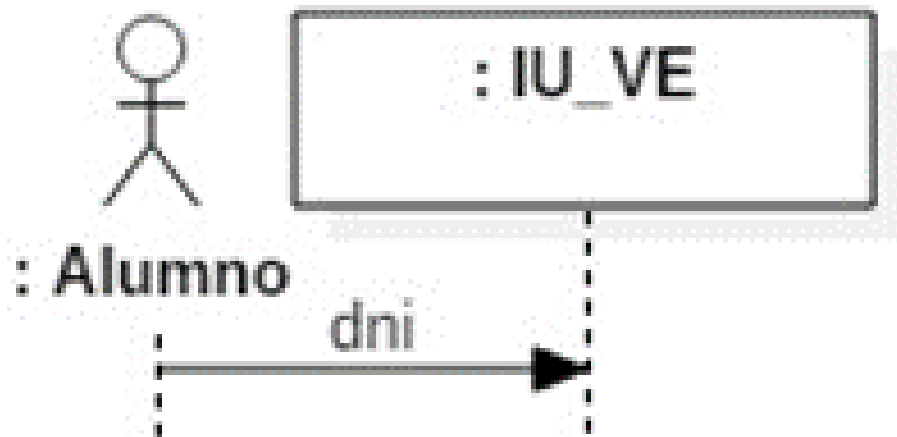


Flujo de eventos: VISUALIZAR EXPEDIENTE

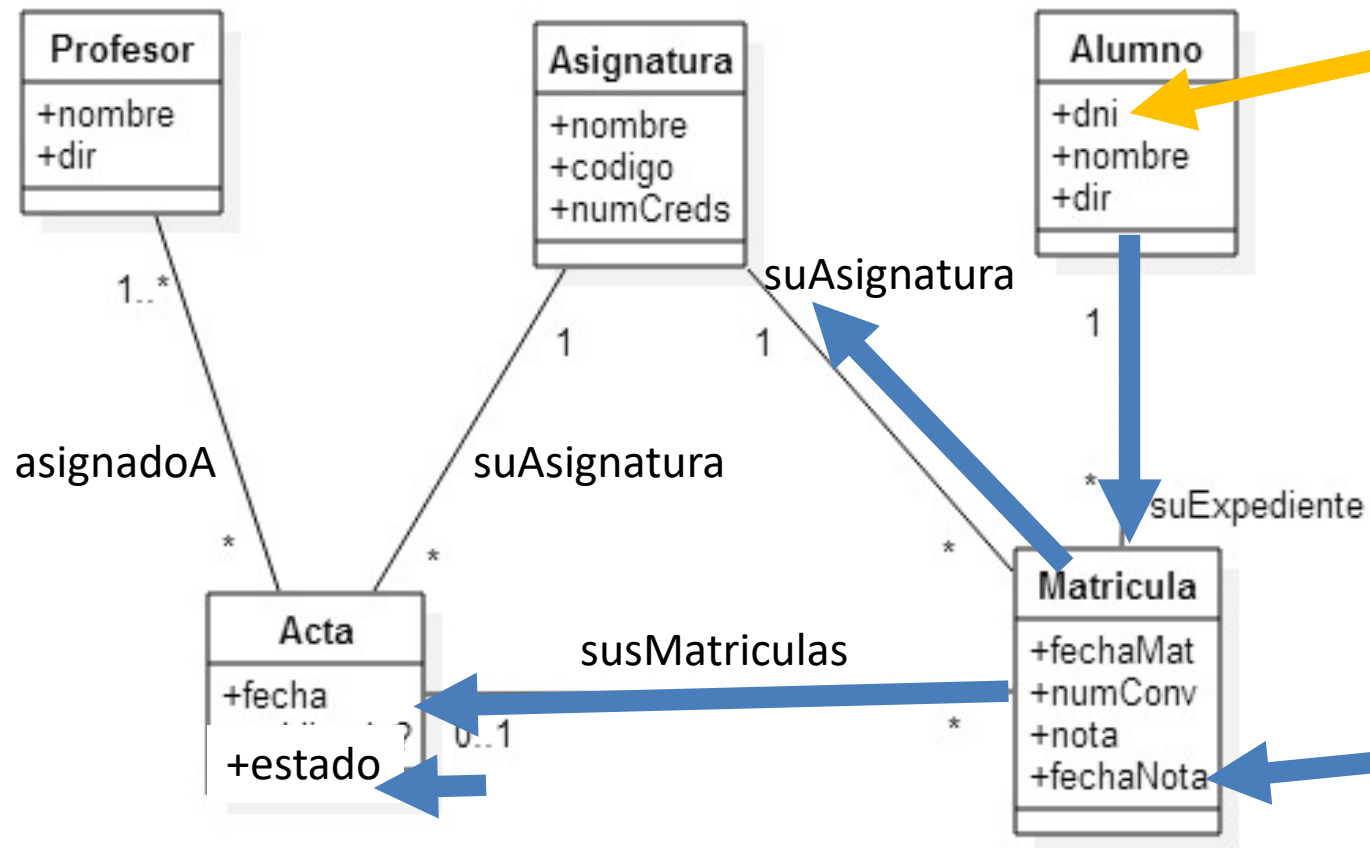
- El alumno introduce su dni

Flujo de eventos: VISUALIZAR EXPEDIENTE

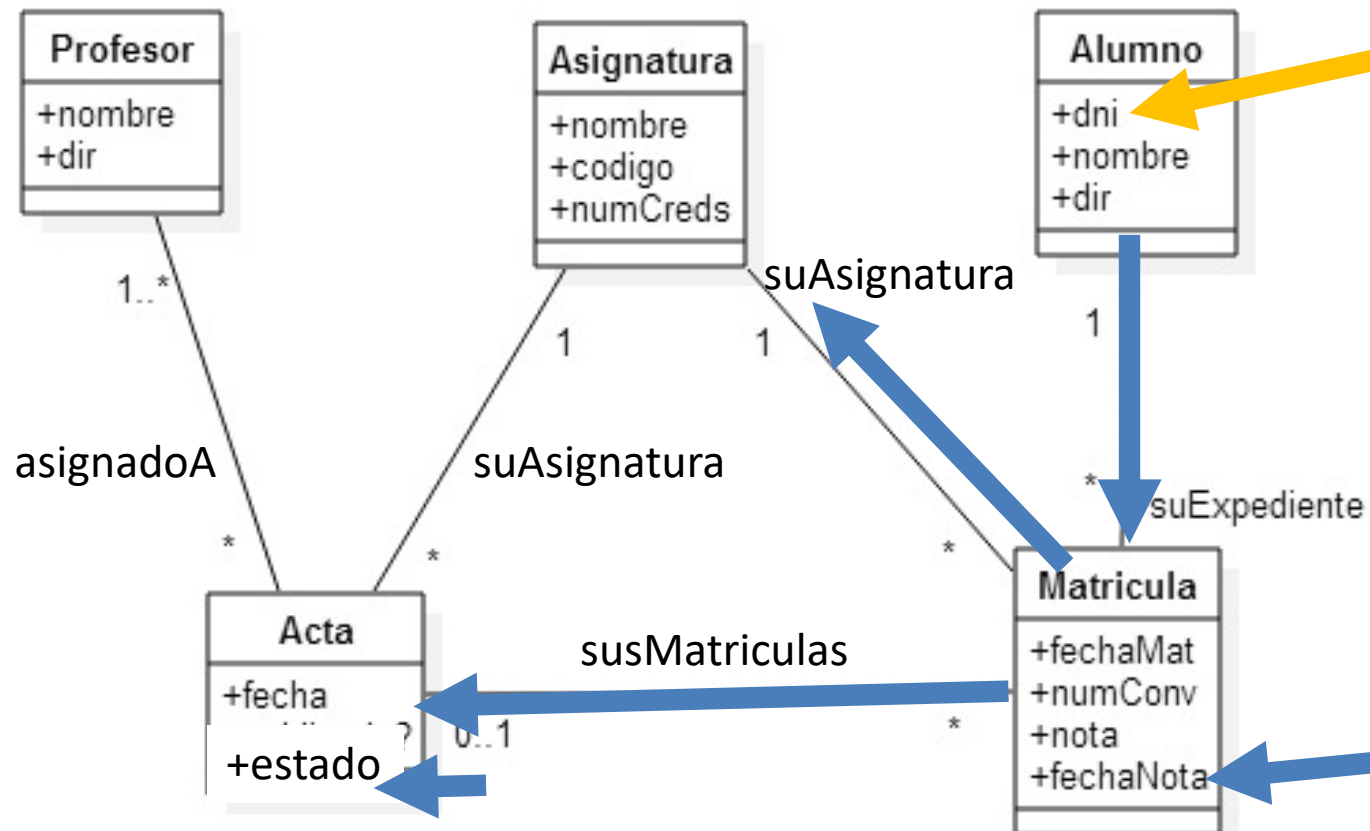
- El alumno introduce su dni



- Se obtienen todas las matrículas del mismo
- Por cada matrícula se muestra la asignatura con la calificación obtenida, siempre que se haya publicado el acta

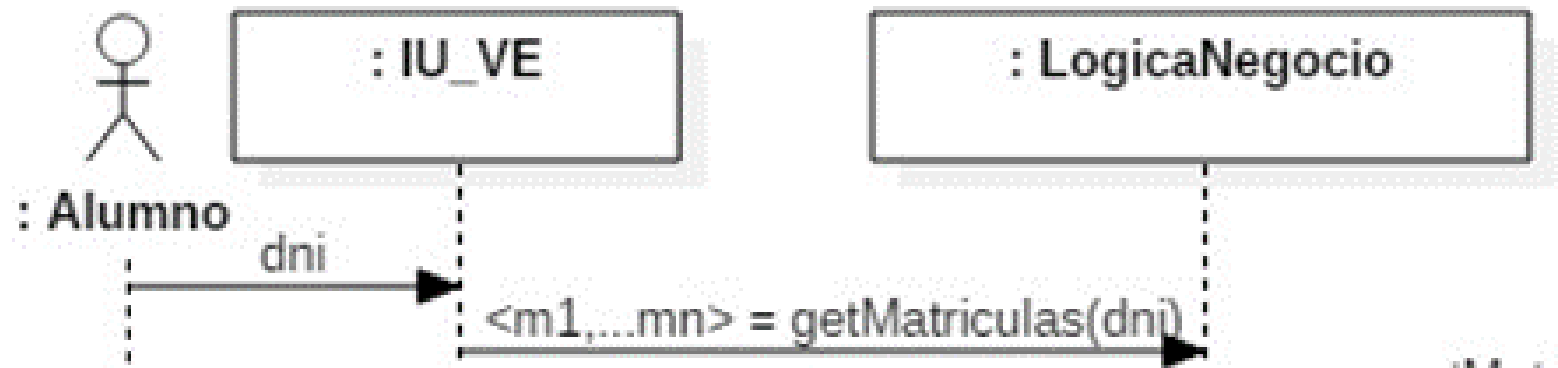


- Se obtienen todas las matrículas del mismo
- Por cada matrícula se muestra la asignatura con la calificación obtenida, siempre que se haya publicado el acta



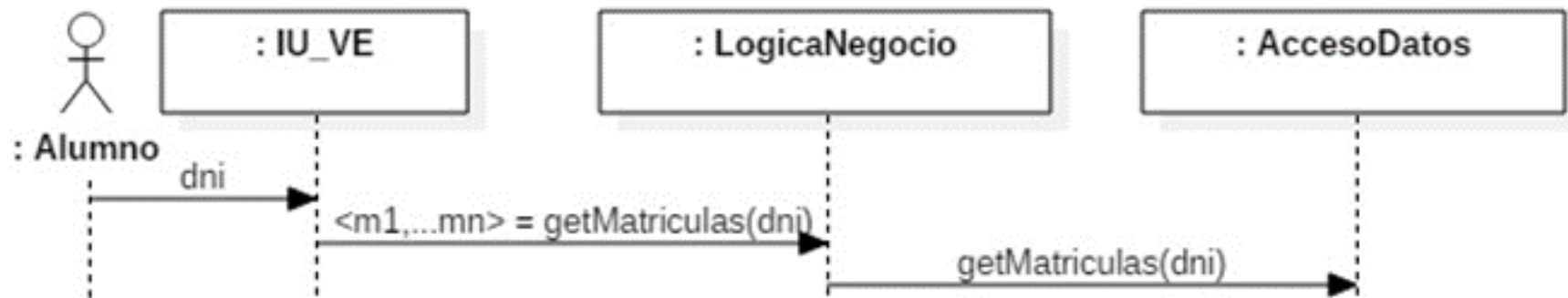
Se pueden traer todas las matrículas del alumno con ese dni, y ya obtendremos sus valores

- Se obtienen todas las matrículas del mismo
- Por cada matrícula se muestra la asignatura con la calificación obtenida, siempre que se haya publicado el acta



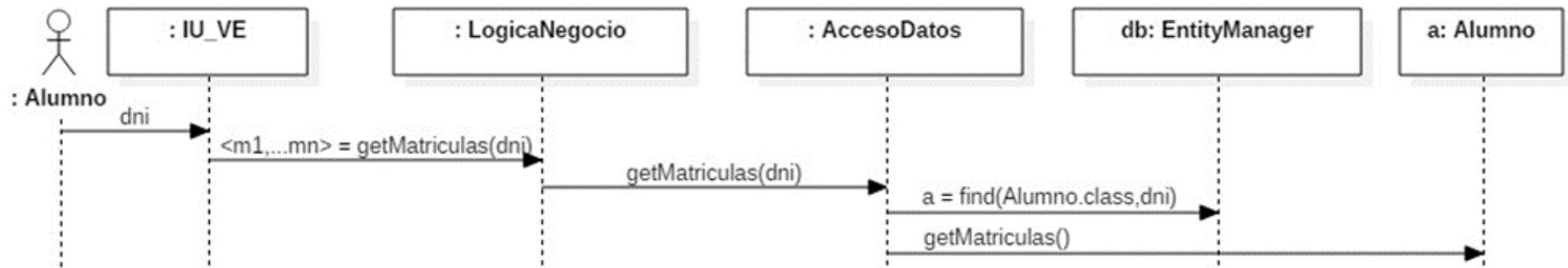
Para conseguir todas las matrículas del alumno con ese dni...

- Se obtienen todas las matrículas del mismo
- Por cada matrícula se muestra la asignatura con la calificación obtenida, siempre que se haya publicado el acta



Como se trata de buscar datos en la BD, la lógica del negocio delega en el acceso a datos, usando un método homónimo

- Se obtienen todas las matrículas del mismo
- Por cada matrícula se muestra la asignatura con la calificación obtenida, siempre que se haya publicado el acta

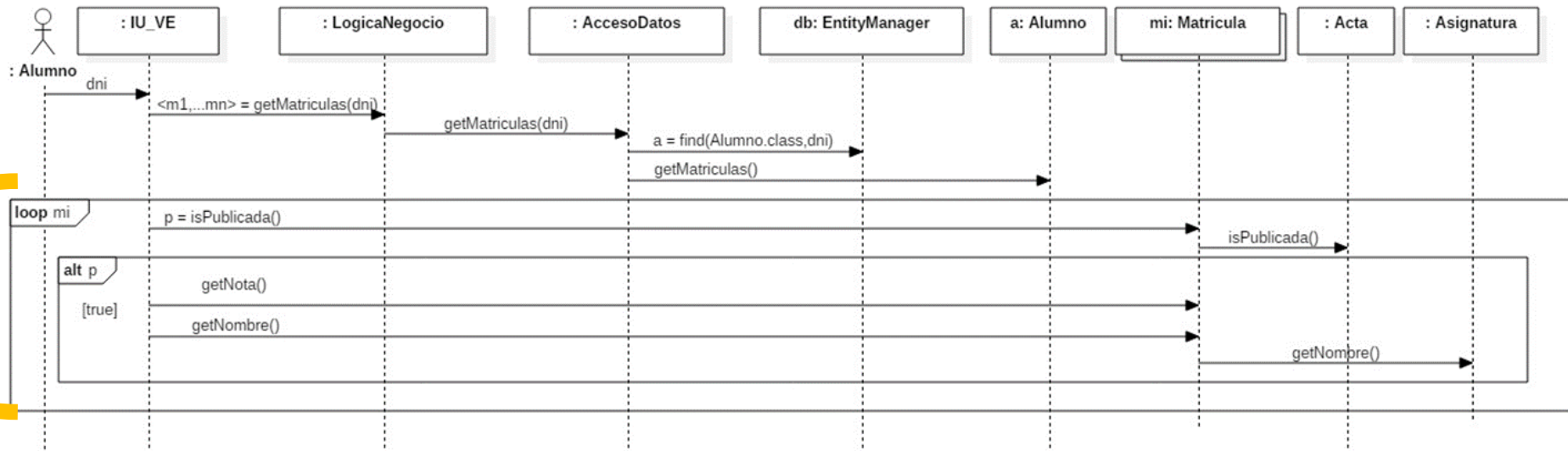


El acceso a datos busca los datos en la base de datos (EntityManager), y en los objetos del dominio (objeto a de Alumno)

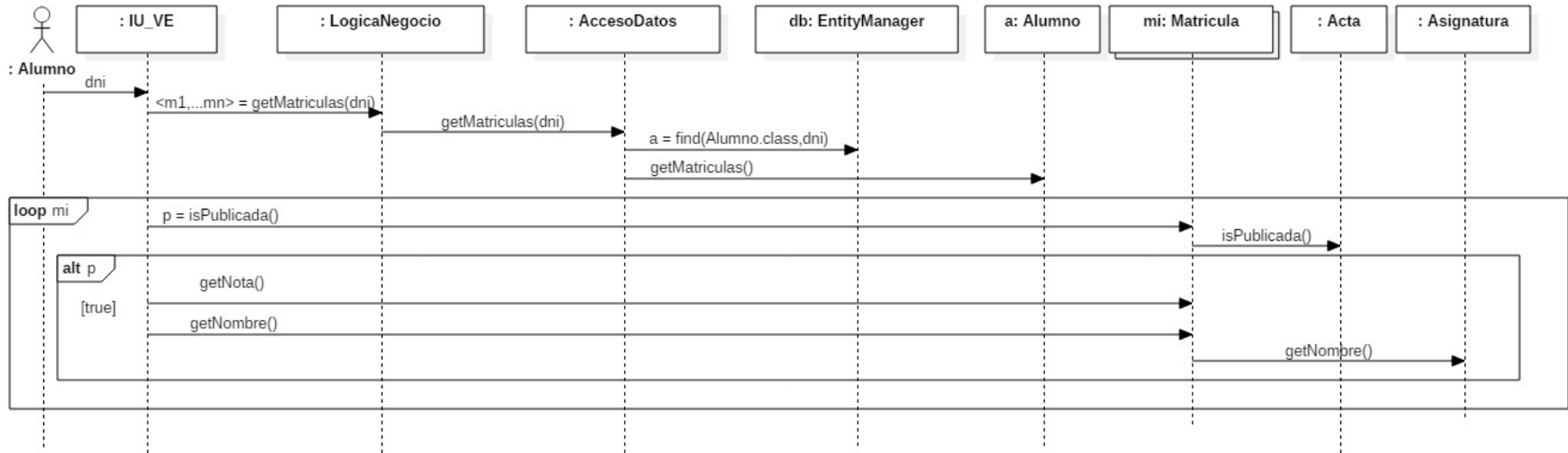
- Se obtienen todas las matrículas del mismo
- Por cada matrícula se muestra la asignatura con la calificación obtenida, siempre que se haya publicado el acta

Se pueden traer todas las matrículas del alumno con ese dni, y ya obtendremos sus valores

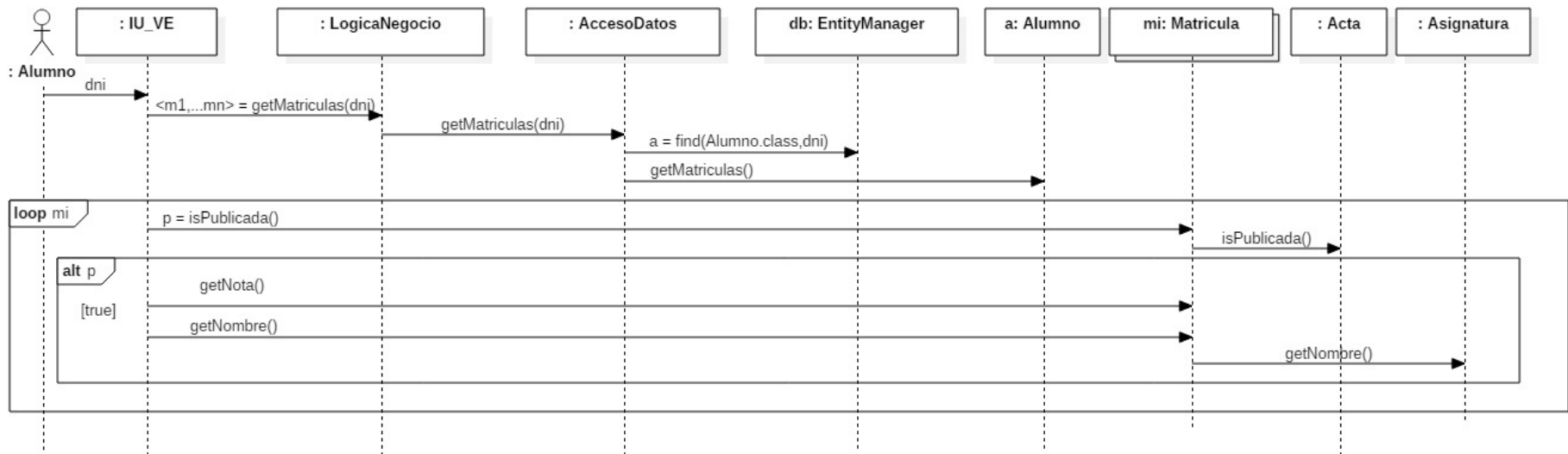
Ahora ya se puede



Versión 0



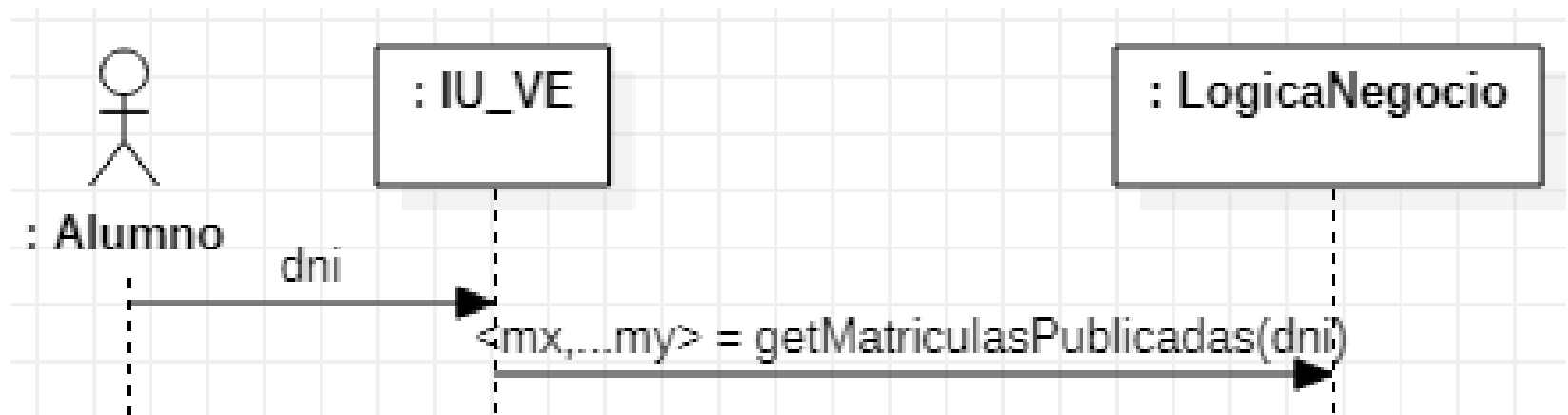
Versión 0



¿Se puede conseguir que la PRESENTACIÓN haga menos trabajo?
Esto es, que lo haga la LÓGICA DEL NEGOCIO.

Flujo de eventos: VISUALIZAR EXPEDIENTE

- El alumno introduce su dni
- Se obtienen todas las matrículas del mismo
- Por cada matrícula se muestra la asignatura con la calificación obtenida, siempre que se haya publicado el acta

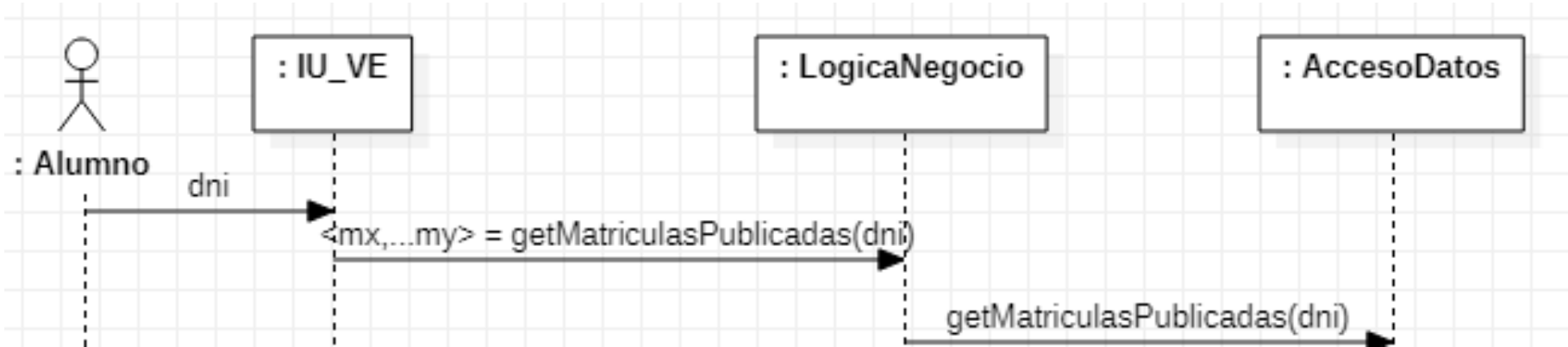


Ahora se piden solamente las matrículas con actas publicadas...

... y a ellas ya les pediremos la asignatura y calificación

Flujo de eventos: VISUALIZAR EXPEDIENTE

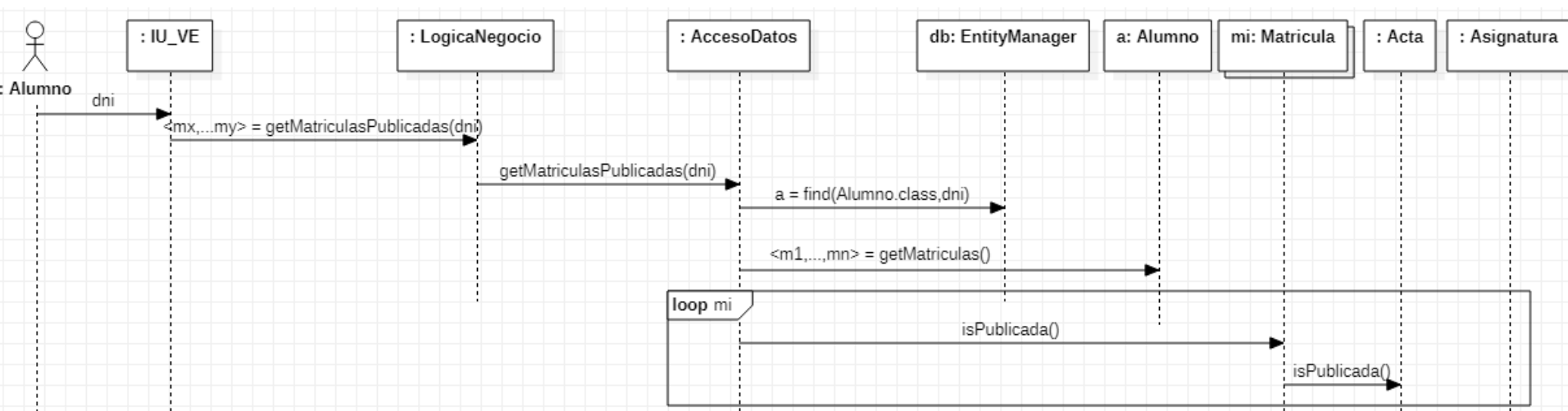
- El alumno introduce su dni
- Se obtienen todas las matrículas del mismo
- Por cada matrícula se muestra la asignatura con la calificación obtenida, siempre que se haya publicado el acta



La lógica del negocio nuevamente delega en el acceso a datos

Flujo de eventos: VISUALIZAR EXPEDIENTE

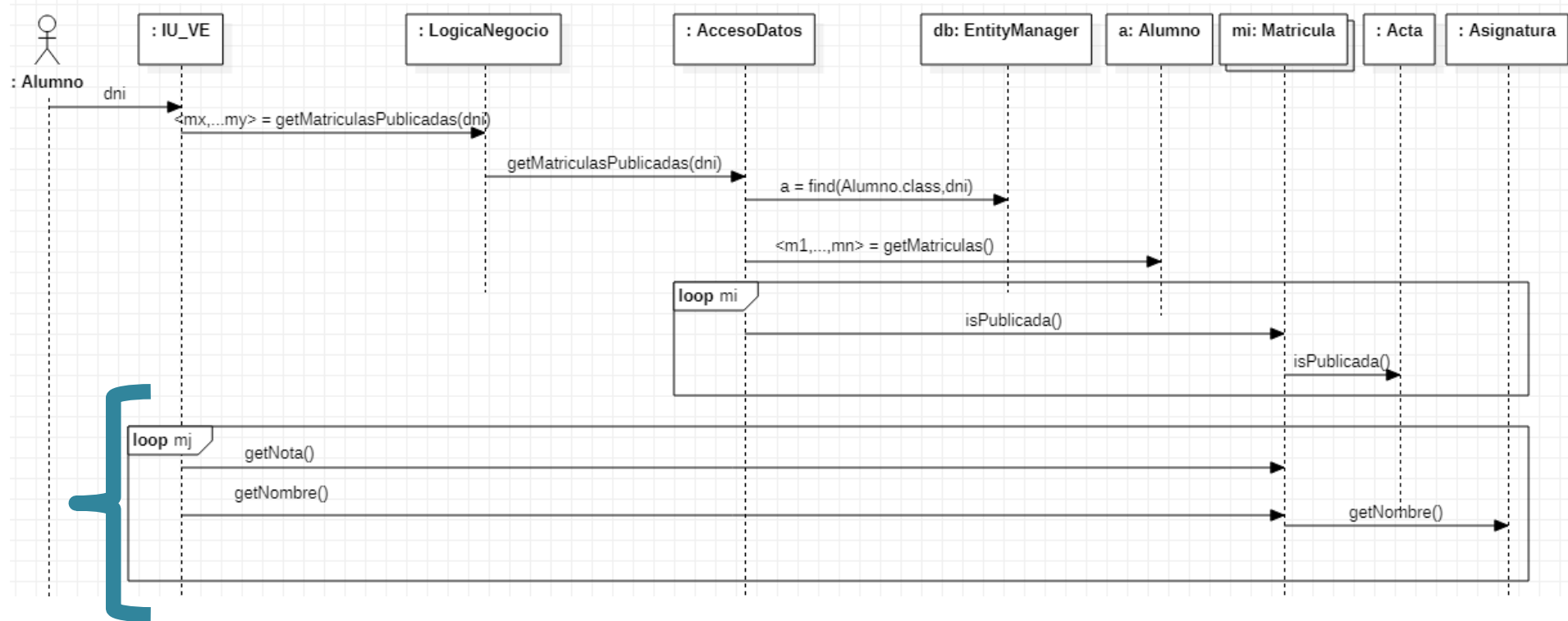
- El alumno introduce su dni
- Se obtienen todas las matrículas del mismo
- Por cada matrícula se muestra la asignatura con la calificación obtenida, siempre que se haya publicado el acta



Y el acceso a datos busca los datos en la base de datos (EntityManager), y navega por los objetos del dominio (objeto a de Alumno y objetos mi de Matricula)

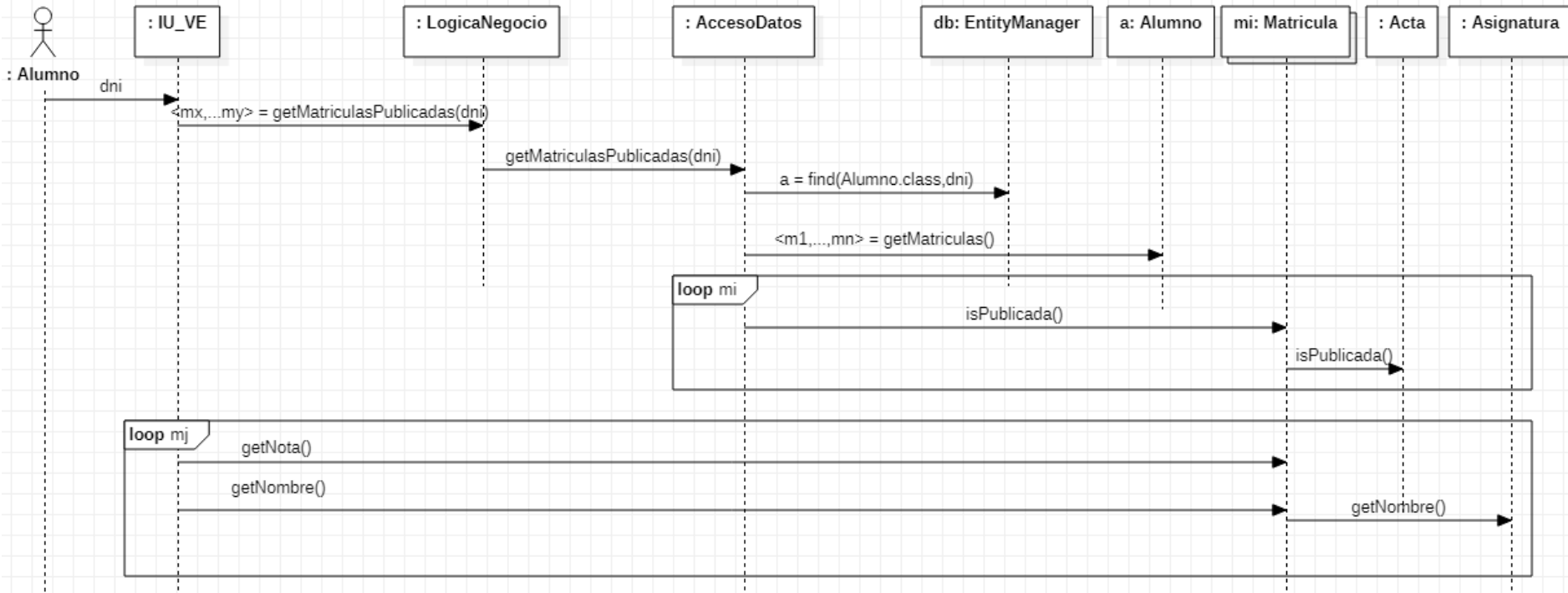
Flujo de eventos: VISUALIZAR EXPEDIENTE

- El alumno introduce su dni
- Se obtienen todas las matrículas del mismo
- Por cada matrícula se muestra la asignatura con la calificación obtenida, siempre que se haya publicado el acta

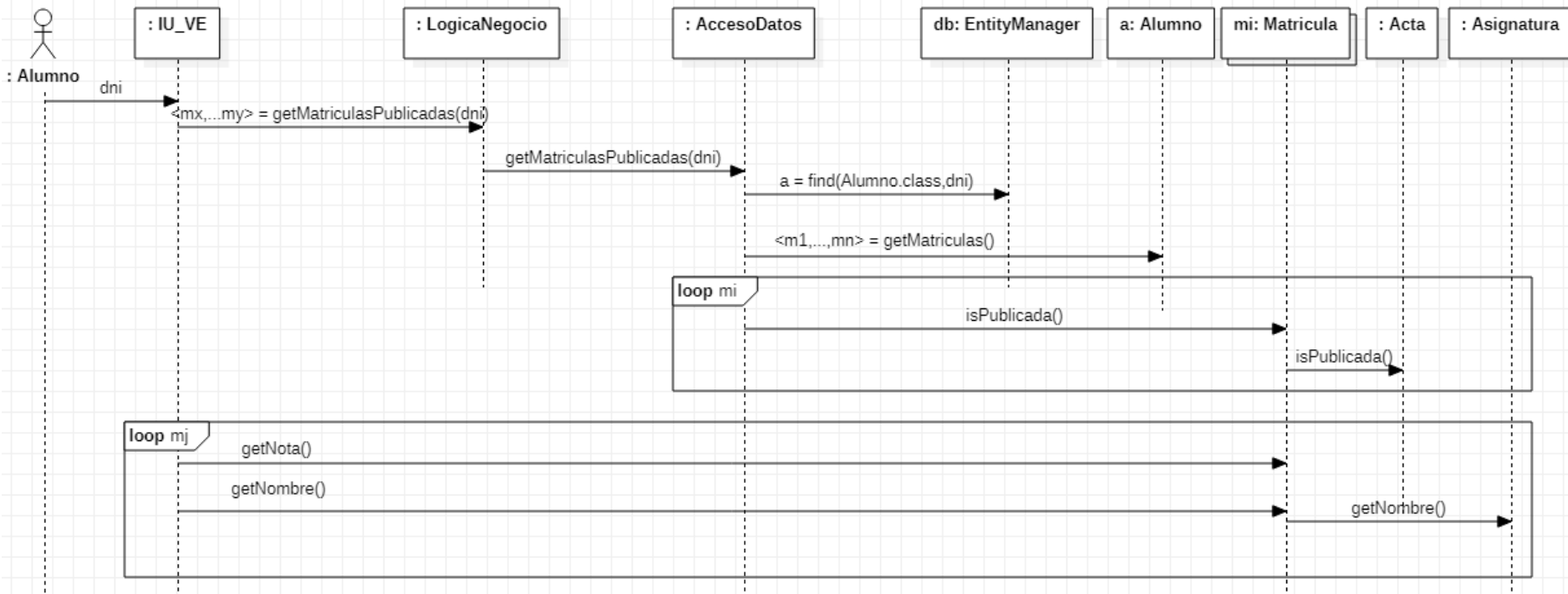


... y a las matrículas se les pide el nombre de la asignatura y calificación

Versión 1: se pide más a la LN para que la presentación haga menos trabajo...



Versión 1: se pide más a la LN para que la presentación haga menos trabajo...



¿Se puede conseguir que el ACCESO A DATOS haga menos trabajo? Esto es, que lo haga la BD (EntityManager)

Sí se puede



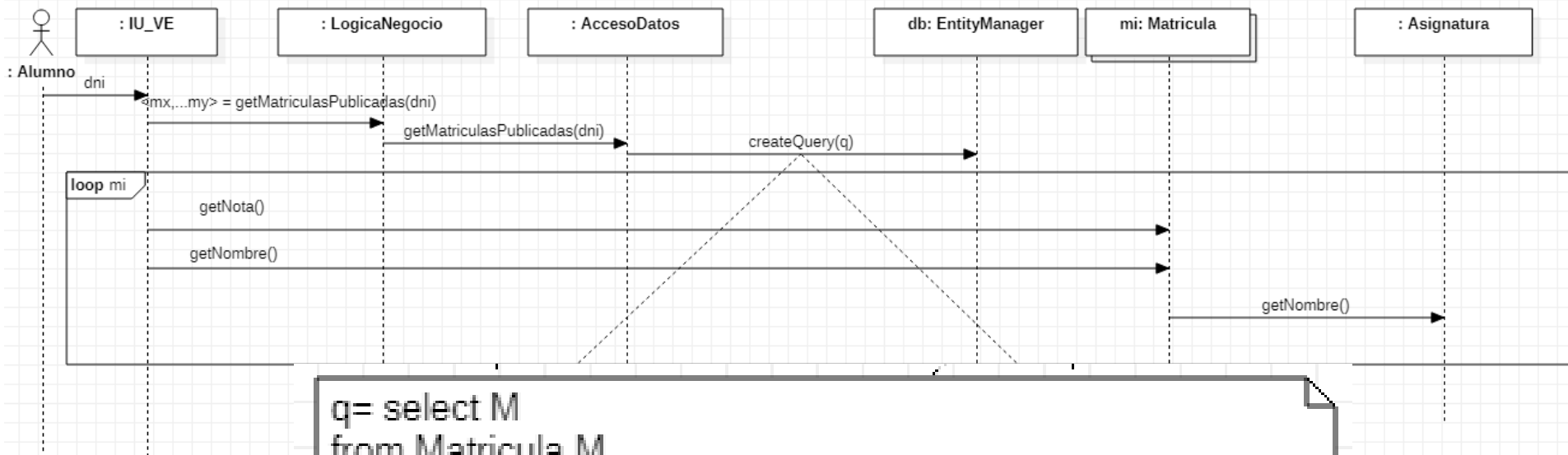
En vez de un `find(Alumno.class,dni)` y luego navegar pidiendo `getMatriculas()` al alumno, y `isPublicada()` a cada matrícula, se puede enviar la sentencia más específica al EntityManager

```
q= select M
from Matricula M
where M.acta.estado="publicada" and M.alumno.dni=%dni
```

```
q= select M
from Alumno A join A.suExpediente M
where M.acta.estado="publicada" and A.dni=%dni
```

2 versiones de consulta JPQL: dependiendo de si recorre Matricula o Alumno se necesitará usar “join”

Versión 2; en vez de “consultar” a los objetos, el AD afina la consulta a la BD



q= select M
from Matricula M
where M.acta.estado="publicada" and M.alumno.dni=%dni

q= select M
from Alumno A join A.suExpediente M
where M.acta.estado="publicada" and A.dni=%dni