

## Enunciado del primer ejercicio de mínimos

Debe implementarse una aplicación web responsive que ofrezca las siguientes funcionalidades:

- listado de asignaturas (CU\_1)
- añadir alumno en una asignatura (CU\_2)
- ver detalle de una asignatura (CU\_3)
- ver detalle de un alumno dentro de una asignatura (CU\_4)

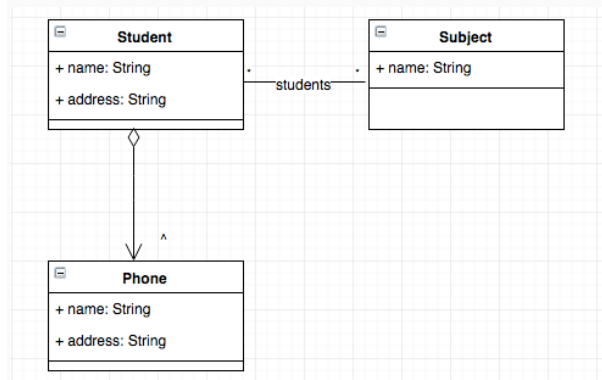
El conjunto de decisiones que definen el estilo arquitectónico propuesto son las siguientes:

- Los usuarios deben acceder a la aplicación sin instalar ningún software especial, únicamente un navegador ya sea en un dispositivo móvil como en un equipo de escritorio
- La aplicación debe implementarse como una single-page que consumirá una API Restful implementada en nodeJS y express
- La base de datos (mongodb) debe almacenar dos entidades de información: Student y Subject

Se propone el siguiente guión:

### 1.- Definición de los modelos de mongoose

El punto de vista de la información que define el problema se muestra a continuación por medio del siguiente diagrama de clases:



A modo de ejemplo se muestran unos posibles datos de las colecciones Students y Subjects:

```
{
  "_id" : ObjectId("56f91864639184069c145734"),
  "name" : "Juan",
  "address" : "c/Rue del percebe ...",
  "phones" : [{"home" : "+34 66666666"}, {"work" : "+34 93555555"}]
}
{
  "_id" : ObjectId("56f919bd639184069c145735"),
  "name" : "Eduard",
  "address" : "c/Rue del cangrejo ...",
  "phones" : [{"home" : "+34 77777777"}, {"work" : "+34 99999999"}]
}
{
  "_id" : ObjectId("56f91b3d8a5c95822af39f8f"),
  "name" : "David",
  "address" : "c / Rue del cargol ...",
  "phones" : [{"home" : "+34 1111111"}, {"work" : "+34 2222222"}]
}
```

```

{
  "_id" : ObjectId("56f91a418a5c95822af39f8e"),
  "name" : "EA",
  "students" : [
    ObjectId("56f91864639184069c145734"),
    ObjectId("56f919bd639184069c145735")
  ]
}
{
  "_id" : ObjectId("56f91bca8a5c95822af39f90"),
  "name" : "IOT",
  "students" : [
    ObjectId("56f91b3d8a5c95822af39f8f")
  ]
}

```

## 2.- Definición de rutas o *Endpoints* del API

Identificar las rutas o endpoints necesarios para implementar los requerimientos descritos anteriormente de manera similar a la tabla que se presenta a continuación y posteriormente realizar su implementación

HTTP	URL	Descripción
GET	/api/todos	Devuelve todas las tareas de la BD
POST	/api/todos	Crea una tarea
DELETE	/api/todos/:todo	Borra una tarea

## 5.- implementación single page

La aplicación angularJS debe implementarse teniendo en cuenta las siguientes premisas:

- Uso de bower
- Uso de \$routeProvider o \$stateProvider
- La página del formulario para añadir un alumno a una asignatura debe realizar alguna validación de cliente (por ejemplo, debe ser obligatorio indicar el nombre del alumno)

### NOTA:

- **Antes de realizar la entrega y abandonar el aula debéis enseñar al profesorado de EA vuestras contribuciones**
- **Vuestra solución se debe publicar en github i el enlace en atenea**