Manual Funcionamiento de la Aplicación.

Antes de compilar y ejecutar la aplicación se debe asegurar que el archivo application.properties ubicado en la carpeta src/main/resources contenga los datos adecuados para la conexión a su base de datos creada previamente.

Los datos a modificar en el archivo application.properties son:

```
Spring.datasource.url
Spring.datasource.username
Spring.datasource.password
```

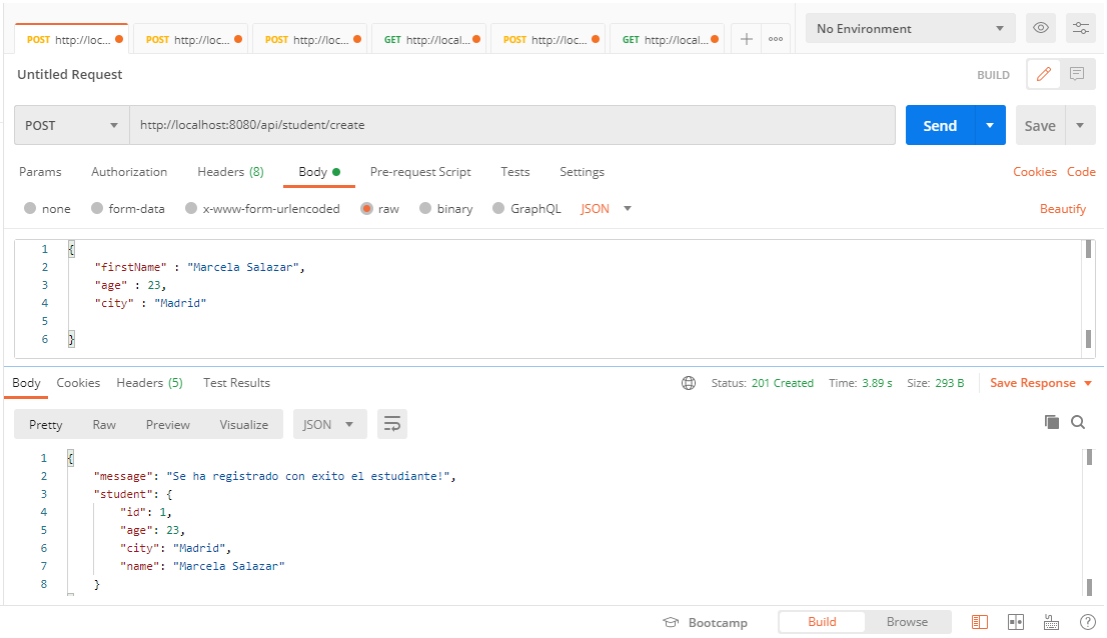
Para una base de datos de Mysql que corresponderán a la URL , nombre de usuario y password de la base de datos, según la configuración de su entorno.

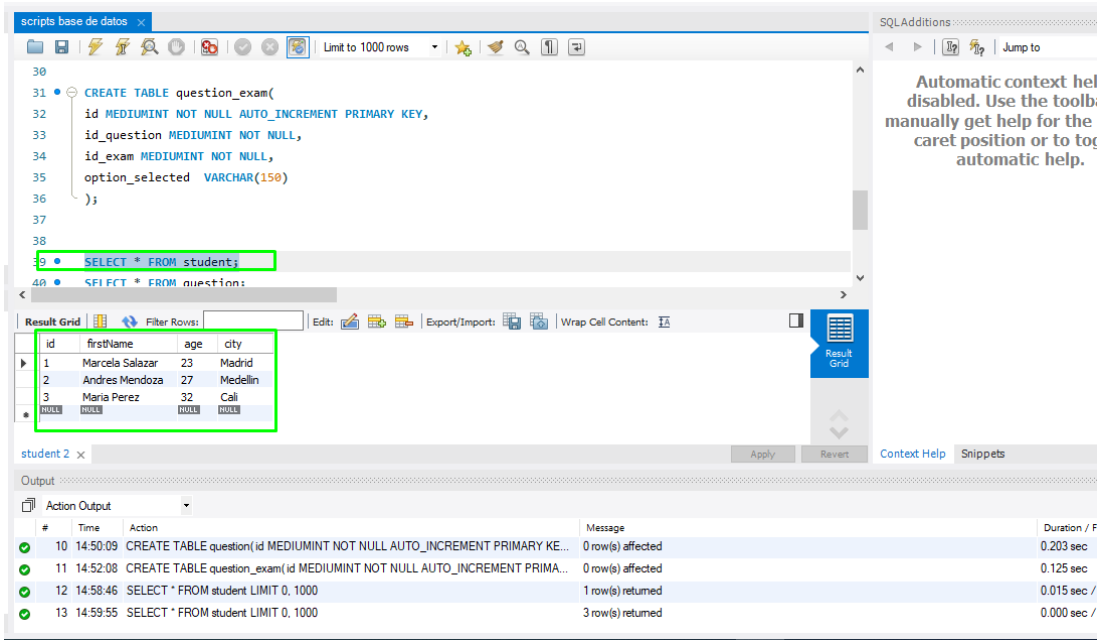
Una vez configurada la conexión a la base de datos ejecutamos la aplicación como una aplicación de springboot.

La colección de peticiones se encontrarán en el repositorio enviado.

1. Crear un conjunto de estudiantes.

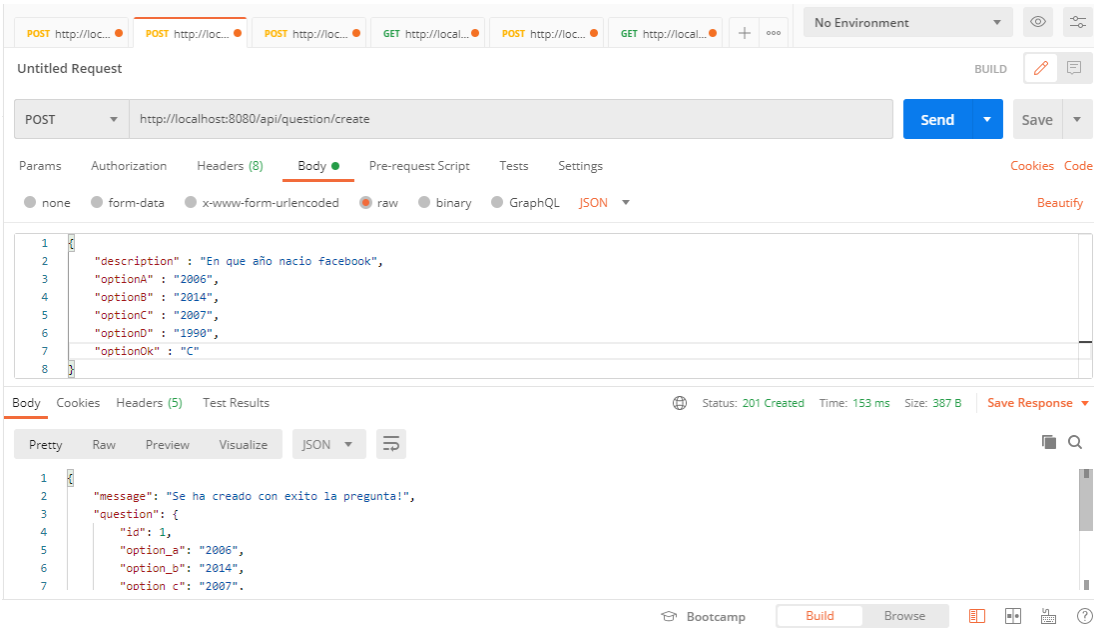
Por medio de la petición que se muestra en la imagen se puede crear varios estudiantes modificando los campos que se muestran en el json a conveniencia.

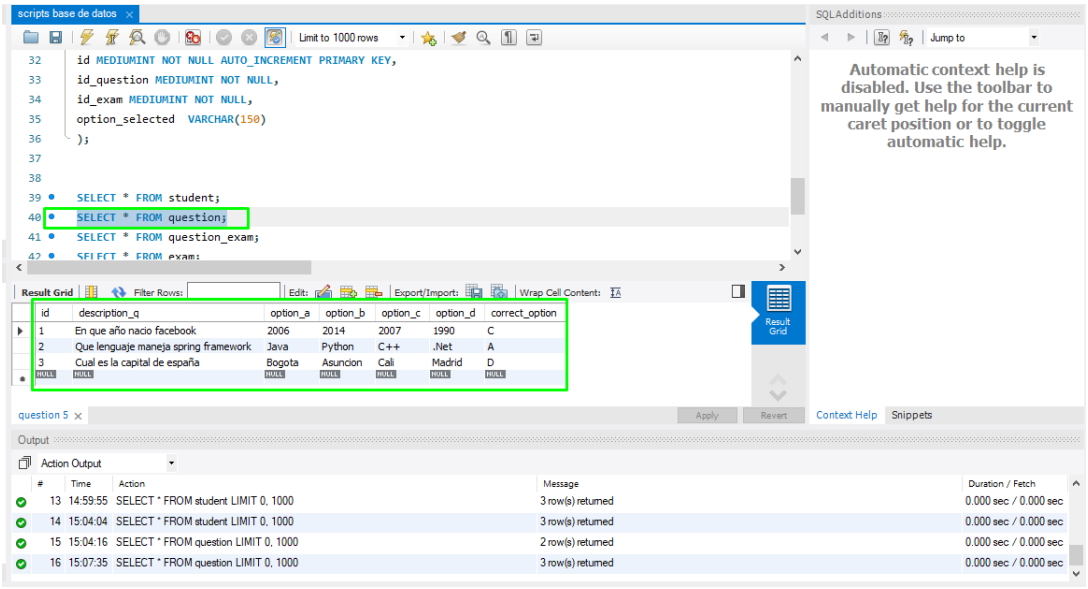




2. Crear un conjunto de preguntas.

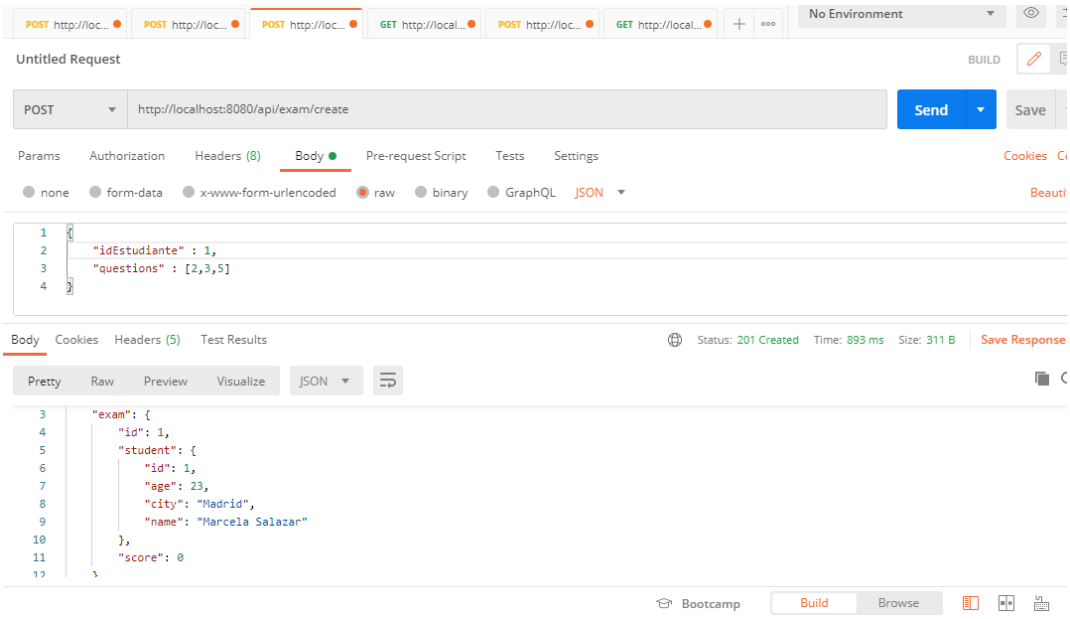
Por medio de la petición que se muestra en la imagen se puede crear una pregunta y registrarla modificando los campos que se muestran en el json a conveniencia.

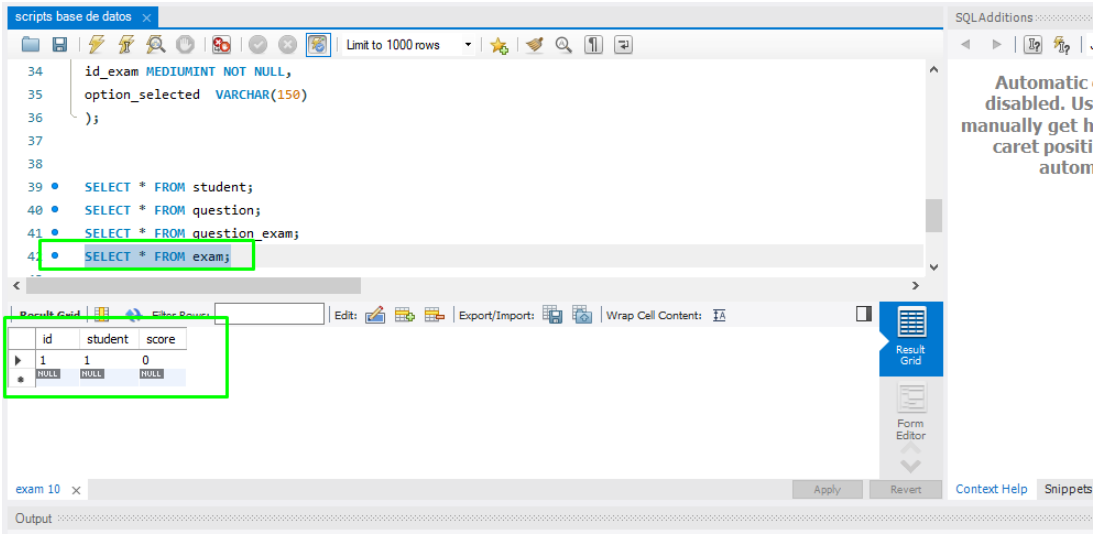




3. Crear un examen.

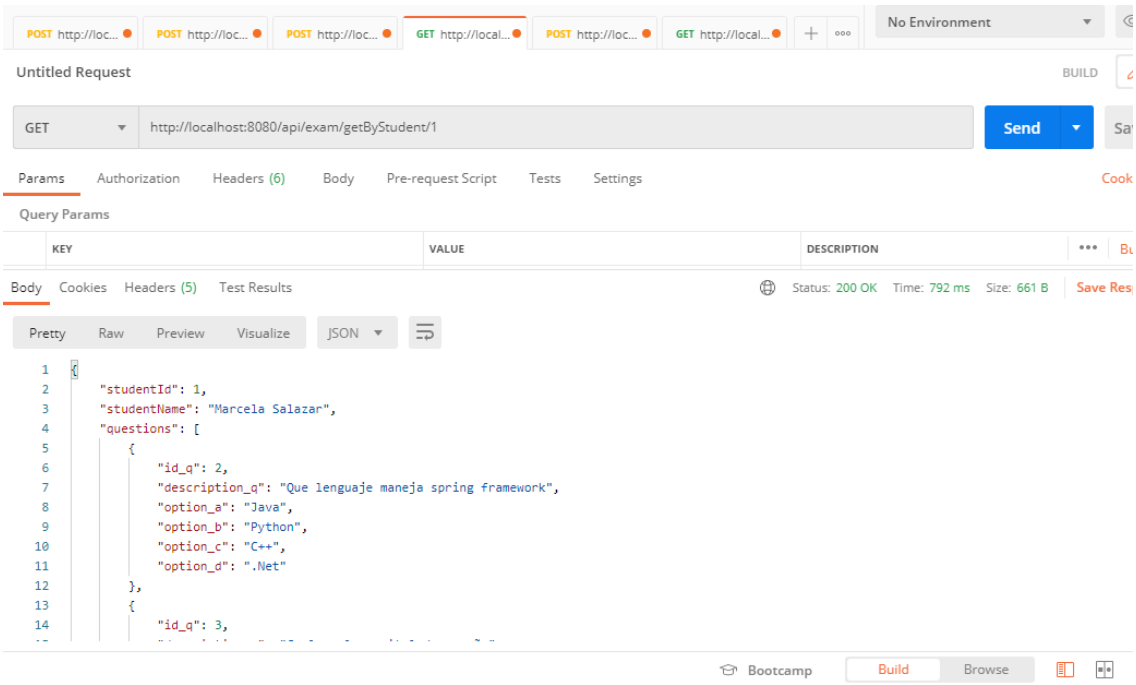
Por medio de la petición que se muestra en la imagen se puede crear un examen, registrarlo y asignarlo a un estudiante modificando los campos que se muestran en el json , donde la lista question debe contener los ids de cada una de las preguntas que desee asignar a estudiante, idEstudiante corresponde al id del estudiante asignado.





4. Obtener un examen creado especificando un id de estudiante.

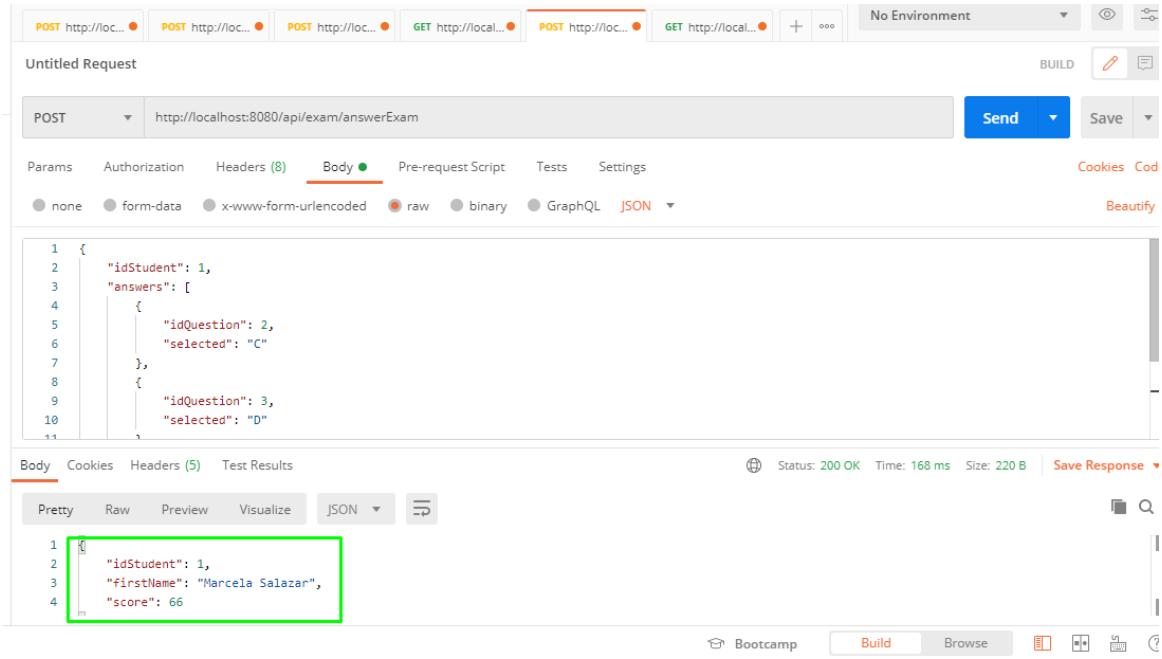
Por medio de la petición que se muestra en la imagen se puede consultar el examen registrado enviando como parámetro en la URL el id del estudiante a consultar.

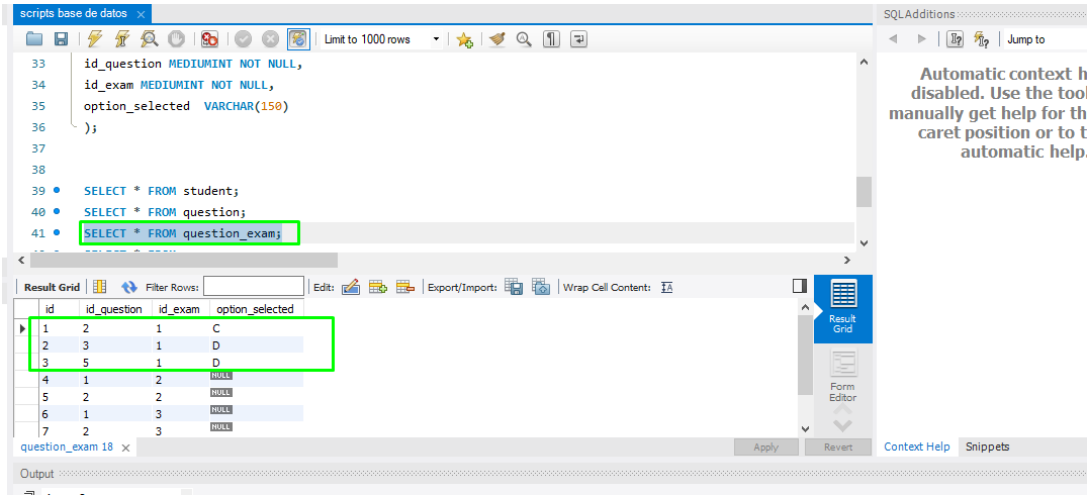


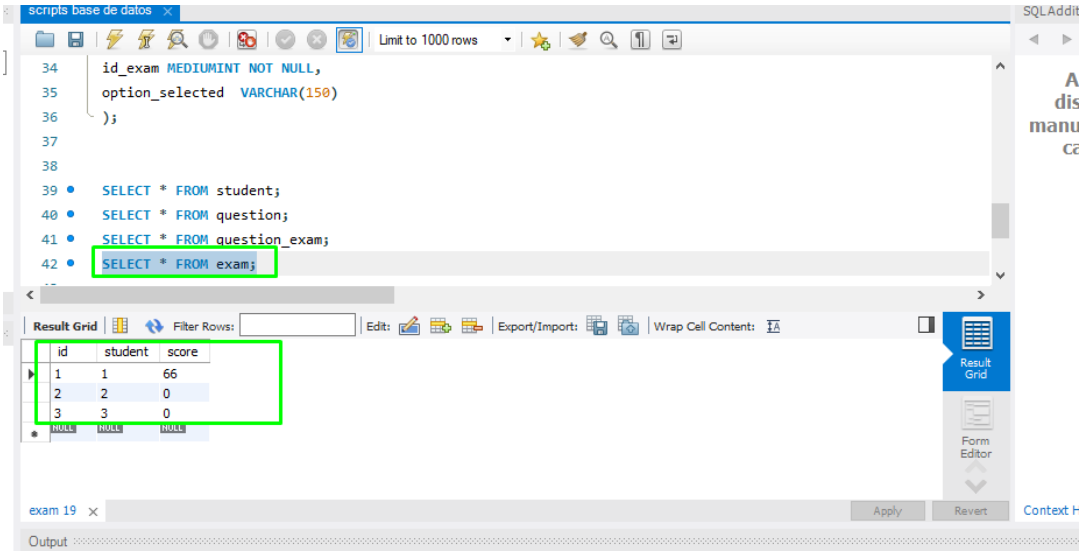
5. Responder un examen.

Por medio de la petición que se muestra en la imagen se puede responder el examen registrado, nótese que el Json enviado contiene el id del estudiante y una lista que contiene las distintas preguntas asignadas, estas contienen el id de la pregunta y la opción que seleccionó el estudiante.

El servicio responde con el nombre del estudiante y la calificación del examen.


Untitled Request
POST
http://localhost:8080/api/exam/answerExam
Send
Save
Params
Authorization
Headers (8)
Body
Pre-request Script
Tests
Settings
raw
JSON
{
"idStudent": 1,
"answers": [
{
"idQuestion": 2,
"selected": "C"
},
{
"idQuestion": 3,
"selected": "D"
Body
Cookies
Headers (5)
Test Results
Status: 200 OK
Time: 168 ms
Size: 220 B
Save Response
Pretty
Raw
Preview
Visualize
JSON
{
"idStudent": 1,
"firstName": "Marcela Salazar",
"score": 66


scripts base de datos
id_question MEDIUMINT NOT NULL,
id_exam MEDIUMINT NOT NULL,
option_selected VARCHAR(150)
);
SELECT * FROM student;
SELECT * FROM question;
SELECT * FROM question_exam;
Result Grid
id
id_question
id_exam
option_selected
1 2 1 C
2 3 1 D
3 5 1 D
4 1 2 NULL
5 2 2 NULL
6 1 3 NULL
7 2 3
question_exam 18
Output


scripts base de datos
id_exam MEDIUMINT NOT NULL,
option_selected VARCHAR(150)
);
SELECT * FROM student;
SELECT * FROM question;
SELECT * FROM question_exam;
SELECT * FROM exam;
Result Grid
id
student
score
1 1 66
2 2 0
3 3 0
exam 19
Output

6. Recopilar las respuestas enviadas por un estudiante.

Por medio de la petición que se muestra en la imagen se puede obtener todas las respuestas seleccionadas por un estudiante, enviando como parámetro en la URL el id del estudiante.

El servicio responde con el nombre del estudiante, un conjunto de preguntas y la opción seleccionada, la calificación del examen.