**PHP Homework**

Provide an api to handle users entities. The api should allow the user to:

- Get his information - buscar

- Delete his profile - borrar

- Change his profile data - update

- Upload an image

Agrege insert

-- To upload an image, use this resource (on any **php library** of your choice):

$ curl -XPOST "<https://api.olx.com/v1.0/users/images>" -F "[file=@img.jpg](mailto:file%3D@img.jpg)" Comando de consola

Response example:

{"id":1956350,"url":"ui/50/97/16/64/o\_1471879056\_200d9c0e4f5c597d7b12d07938de58df.jpg"}

*Just prepend ht*[***tps://images01.olx-st.com/***](https://images01.olx-st.com/) ***ui/50/97/16/64/o\_1471879056\_200d9c0e4f5c597d7b12d07938de58df.jpg*** *to the* ***returned url*** *to be able to access the image.*

User's table:

id int(11) not null primary key auto\_increment,

name varchar(100) not null,

picture varchar(200),

address text

- Rules:

-- You can use any Module you want, but you can't use a framework as a whole (Example: you could use symfony/dependency-injection, but you can't use symfony as a whole)

-- Data security, architecture and scalability will be considered please take this into account, before you start coding.

Puedo usar una libreria de un framework pero no un framework entero

- Code Delivery:

-- Use GitLab, in a private repository and when you finish it, please share the Developers access with:

\* Jesús Farfan: @jesus-farfan-olx

\* Gerardo Herz: @gerardo-herz-olx

\* Nicolas Cajelli: @nicolas-cajelli

\* Federico Kandus: @fkandus-olx

- Bonus points:

-- Unit tests

-- Code Style (You could choose any Standard, as long as you respect it, just let us know which) indicar standard

-- Provide a docker env to test your code

-- Improvement ideas / feedback