Prueba PAGO FACIL

Descripción Técnica

Lenguaje de programación: PHP versión 7
Framework de desarrollo: Laravel versión 5.5
Base de Datos: Postgres *
Versionador: GIT

*Se implemento mediante migraciones por lo cual solo se debe cambiar la configuración para implementarlo en mysql

```
|
| Here you may specify which of the database connections below you wish
| to use as your default connection for all database work. Of course
| you may use many connections at once using the Database library.
|
*/

'default' => env('DB_CONNECTION', 'pgsql'),
/*
|--------------------------------------------------------------------------
| Database Connections
|--------------------------------------------------------------------------
|
| Here are each of the database connections setup for your application.
| Of course, examples of configuring each database platform that is
```

Cambiar por **mysql** y configurar los parámetros de la siguiente imagen.

```
'mysql' => [
    'driver' => 'mysql',
    'host' => env('DB_HOST', '127.0.0.1'),
    'port' => env('DB_PORT', '3306'),
    'database' => env('DB_DATABASE', 'forge'),
    'username' => env('DB_USERNAME', 'forge'),
    'password' => env('DB_PASSWORD', ''),
    'unix_socket' => env('DB_SOCKET', ''),
    'charset' => 'utf8mb4',
    'collation' => 'utf8mb4_unicode_ci',
    'prefix' => '',
    'strict' => true,
    'engine' => null,
],
```

Instalación

Requerimientos

PHP, composer, postgresql o mysql, GIT

1. Descomprimir el archivo schools.tar.gz en el directorio de su preferencia
2. Ingresar a la carpeta school
3. Dar el siguiente comando composer install
4. Php artisan migrate
5. Php db:seed (carga en bd de registros para probar la herramienta).
6. php artisan serve –host= TUIP

API REST

Configuración de rutas

/var/www/html/school/routes.php

```
20
21  Route::get('courses',  ['uses' => 'StudentController@index']);
22  Route::get('student/{id}', ['uses' => 'StudentController@show']);
23   //$router->post('student/{id}/qualification', ['uses' =>
     'StudentController@addQualification']);
24  Route::post('student/{id}/qualification', ['uses' => 'StudentController@addQualification']);
25  Route::put('test/', ['uses' => 'StudentController@update']);
26   Route::delete('test/{id}', ['uses' => 'StudentController@destroy']);
27   Route::post('register', 'Api\AuthController@register');
```

Modelos

/var/www/html/school/app

Student Modelo que se encarga de integrarse con con la tabla t_alumnos

Courses Modelo que se encarga con integrarse con la tabla t_materias

Califcations Modelo que se encarga con integrarse con la tabla t_calificaciones

Controladores

school/app/Http/Controllers

StudentController.php Controlador que se encarga de realizar y resolver las peticiones de la prueba

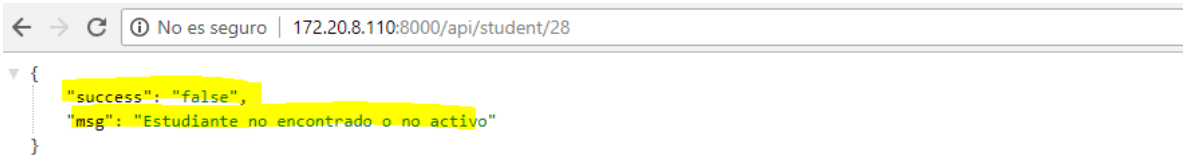
RESULTADOS

GET

URL http:// TUIP:8000/api/student/2

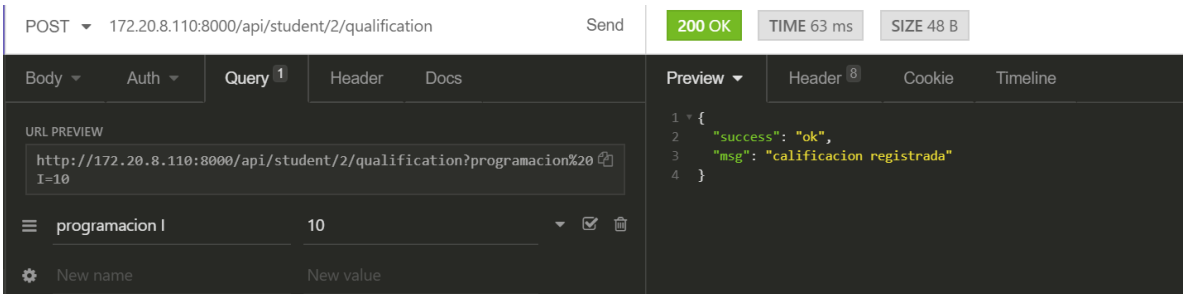
Obtiene mediante el get el id del estudiante y muestra el resultado conforme a la prueba.

Extra: valida que el estudiante exista o cuente con calificaciones



POST

172.20.8.110:8000/api/student/2/qualification



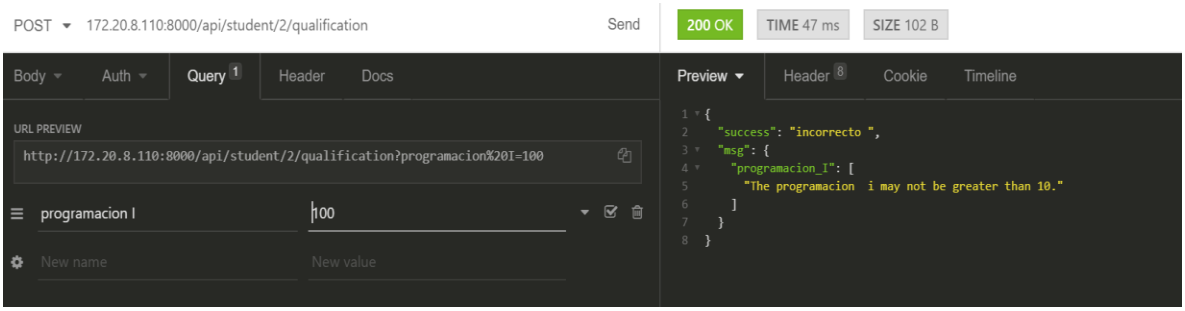
Se envía mediante Get el id del estudiante y por post se envía el nombre de la materia y su calificación

EXTRA validaciones:

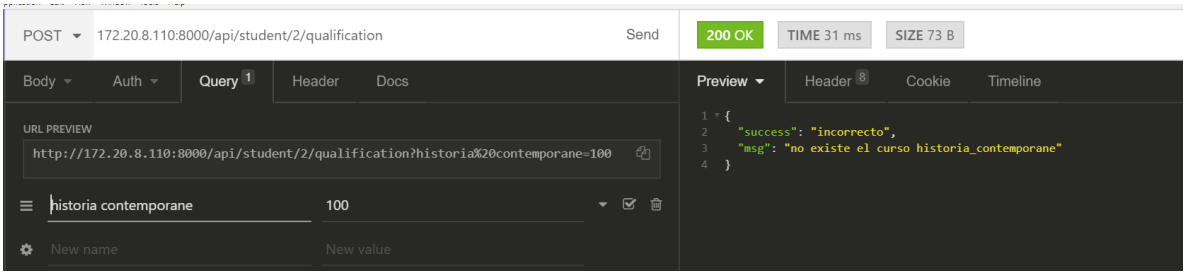
1. **valida que el nombre de la materia exista en la bd**
2. **valida que el alumno exista en la bd**
3. **valida que la calificación sea flotante entre 0 y 10**

EJEMPLO

3

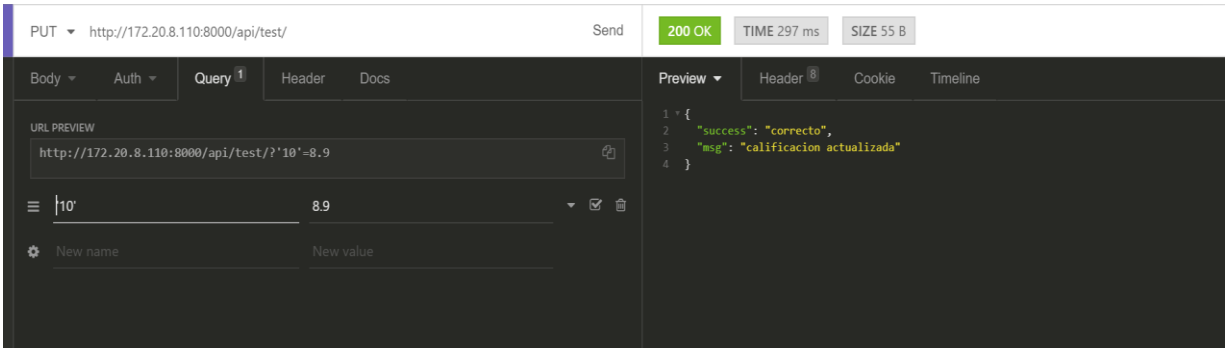


2



PUT

url http://172.20.8.110:8000/api/test/

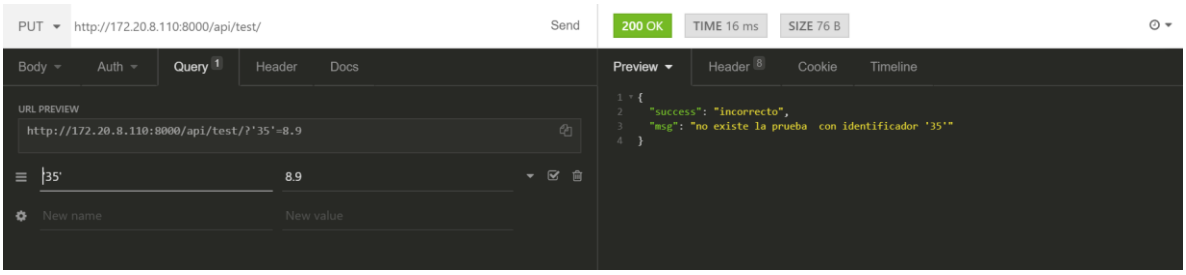


Recibe como parámetro el id de la prueba entre comilla simpre ‘10’ el valor por el que se va actualizar

Extra

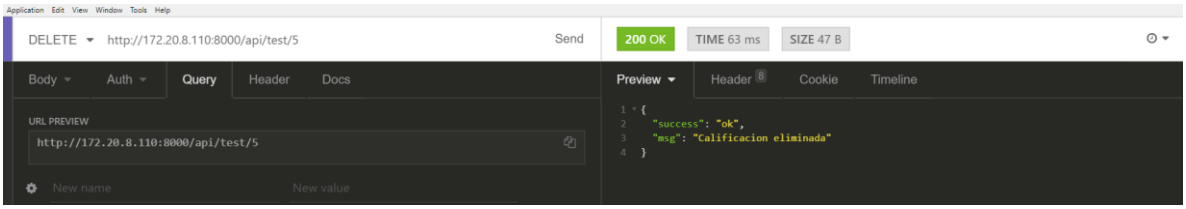
Validación que exista la calificación y el alumno

Validación que sea entre 1 y 10



DELETE

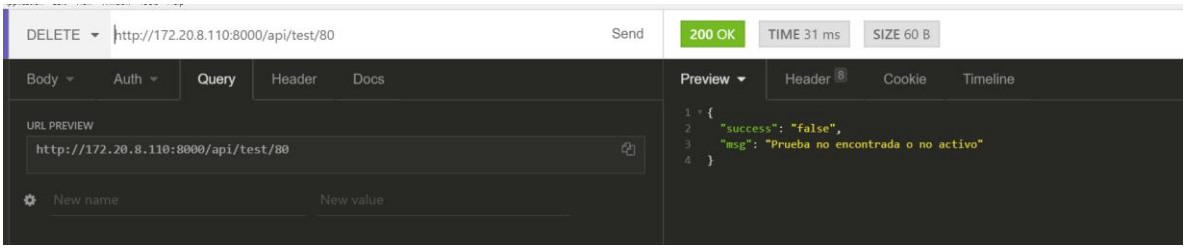
URL: http://172.20.8.110:8000/api/test/34



Recibe mediante parámetro el id de la prueba a eliminar si es satisfactorio manda mensaje y elimina registro de la base de datos

EXTRA:

Valida que exista la prueba si la prueba existe



Implementación de método de seguridad

Se instalo Laravel Passport que tiene se integra con oauth 2,

```
{
    "name": "laravel/laravel",
    "description": "The Laravel Framework.",
    "keywords": ["framework", "laravel"],
    "license": "MIT",
    "type": "project",
    "require": {
        "php": ">=7.0.0",
        "fideloper/proxy": "~3.3",
        "laravel/framework": "5.5.*",
        "laravel/tinker": "~1.0",
        "laravel/passport": "4.0.*"

    },
    "require-dev": {
```