

FUNCIONAMIENTO DE LA APLICACIÓN

----- AppTeachers -----

APIS PUBLICAS:

Si no se llama al middleware checkToken antes de acceder a la ruta

GET /api -> pantalla ppal. Selecciona materia y CP y click boton "Buscar profesores cerca y con mejor puntuación", lee una lista de materias para el dropdown, -> Procedimiento/Funcion "Teachers-Zona" al ir a GET /api/teachers/teachersnear

GET /api/users/register (al hacer click sobre "Registrarse en la Aplicación") -> introducir nombre, email, password, tipo de usuario y nombre de usuario -> POST /api/users/register -> encripta password e inserta en tabla 'users'.

A) Si es de tipo 'TEACHER' POST /api/teachers (nuevo teacher vacío con el id de usuario en su clave foránea), Se abre un formulario para introducir los campos de profesor y actualiza la tabla 'teachers' con id del teacher creado antes.

UPDATE /api/teachers/:teach_id

B) Si es de tipo 'STUDENT' POST /api/students (nuevo students vacío con el id de usuario en su clave foránea), Se abre un formulario para introducir los campos de student y actualiza la tabla 'students' con id del student creado antes.

UPDATE /api/students/:student_id

No se dan de alta administradores via web, ya está creado con usuario ADMIN y contraseña ADMIN. Los subjects se crean via Wordbench y no por la Web.

PANTALLA de LOGIN al hacer click sobre "Entrar en el sistema") -> introducir email y password -> POST /api/users/login -> valida usuario, password y DEVUELVE UN TOKEN al cliente con una duración e incluye el req.user[0][0].user_type para control de acceso a cada ruta. Falla el login, si el campo del usuario tiene el user_deleted a 'SI'. Si es profesor el teach_validated debe ser 'SI'

APIS POR TIPO DE USUARIO

Ejecuta el middleware checkToken y obtenemos req.user[0][0]

Después se llama a un segundo middleware que controla el acceso a cada ruta:

Si req.user[0][0].user_type es 'STUDENT' y la ruta tiene el middleware accesoStudents-> next()

Si req.user[0][0].user_type es 'ADMIN' y la ruta tiene el middleware accesoAdministrator-> next()

Si req.user[0][0].user_type es 'TEACHER' y la ruta tiene el middleware accesoTeachers-> next()

Si req.user[0][0].user_type es 'TEACHER' o 'ADMIN' y la ruta tiene el middleware accesoAdministratorORTeachers-> next()

Redireccionar: <https://expressjs.com/es/4x/api.html#res.redirect>

a) si es 'STUDENT' va al panel de Automatrícula (redirección automática del login a GET /api/students/board (sólo accesoStudents), donde leemos las materias para dropdown de materias). Rellenar campos y click sobre 'Buscar Teachers' -> Procedimiento/Funcion "Teachers-Zona" al ir a GET /api/teachers/teachersnear

*Proced/Funcion "Teachers-Zona"-> SELECT teach_name as Profesor, teach_city as Ciudad, teach_zip as "Código Postal", avg(enroll_assessment) as Valoración FROM teachers WHERE teacher_delete='NO' and teacher_validated='SI' and teach_zip=<cp indicado>

JOIN subjects ON subjects.subj_id=teach_id_subject and subjects.subj_id=<subject seleccionado>

JOIN enrollments ON enrollments.enroll_id_teacher=teachers.teach_id

GROUP BY Profesor, Ciudad, "Código Postal"

ORDER BY Valoración DESC;

Esta lista la devolvermos en formato JSON una lista de objetos al cliente con status=200

Si falla devolvemos status=500

El resultado de la consulta aparece en 'Resultado de la búsqueda'. Introducir fechas de inicio y fin y click en 'Realizar Matrícula' -> POST /api/enrollments. También se puede realizar la Valoración y comentarios de un profesor en el mismo panel pero posteriormente. Al hacer click sobre 'Realizar Valoración'. -> PATCH /api/enrollments (campos de Valoración y Comentarios).

b) si es 'ADMIN' va al panel de Administración (redirección automática de login a esta ruta GET api/admin/board con

dos botones b.1) y b.2))

b.1) BORRAR TEACHERS. Leer materias para dropdown de materias. Rellenar campos y click sobre 'Buscar Teachers' -> al ir a GET api/teachers/novalidados

*Proced/Funcion "Buscar-Teachers"

```
SELECT teach_name as Profesor, teach_city as Ciudad, teach_zip as "CP", avg(enroll_assessment) as Valoracion,  
teach_experinece_years as 'Experiencia', user_created as 'Fecha alta' FROM teachers WHERE teach_deleted='NO'  
JOIN users ON user_id=teach_id_user  
JOIN enrollments ON enroll_id_teacher=teach_id  
ORDER BY Valoracion DESC;
```

El resultado de la consulta aparece en 'Resultado de la búsqueda'. Se puede borrar un Teacher con click sobre 'Borrar' -> PATCH /api/teachers/:teacher_id y PATCH /api/users/:user_id(=teacher_id_user) el campo 'deleted' se pone a 'SI'.

b.2) VALIDAR TEACHERS . Lista de los teachers registrador recientemente ordenado por fecha de alta (user_created de la tabla user).

```
SELECT teach_name as Profesor user_created as 'Fecha alta' FROM teachers WHERE teach_validated='NO' AND  
teach_deleted='NO'
```

```
JOIN users ON user_id=teach_id_user
```

```
ORDER BY user_created DESC;
```

Al hacer click en el boton 'Validar Teacher' -> PATCH /api/teachers/:teacher_id y actualizamos el campo 'Validado' a 'SI'.

c) si es un 'TEACHER' va al panel de Teachers (redirección automática del login a /api/teachers/board). Puede tener dos botones c.1) y c.2)

En c.1) GET /api/enrollments/teachers consulta los estudianes que tiene matriculado entre dos fechas determinadas.

En c.2) PATCH /api/teachers/:teacher_id el campo de teacher_deleted a 'SI' y user_deleted correspondiente a 'SI'