Modalidades

Las modalidades se encuentran en la carpeta *controllers/modalidades.controller.js* y tienen la siguiente estructura: modalidad (nombre de la modalidad), e identificador (ID).

```
let modalidades = [
    {
        "modalidad": "Incapacidad",
        "id": "e7b26ae3-ee9a-48ac-97d6-c824641627c0"
    },
    {
        "modalidad": "Soluciones Industriales",
        "id": "186571f2-cf9a-43e2-bb0b-946ad3f7c622"
    }
];
```

En el mismo archivo se encuentra la clase *ModalidadesController* que tienen todos los métodos para interactuar con el arreglo de modalidades:

- ingresar: Este método tiene como parámetro el nombre de la modalidad, luego crea un identificador único con la librería uuid y luego guarda la modalidad en el arreglo para luego retornar el identificador de la nueva modalidad.
- mostrar: Retorna el arreglo de modalidades.
- existeModalidad: Tiene como parámetro el identificador de la modalidad (idModalidad). Se declara un arreglo “existe” que servirá para indicar si existe la modalidad en el arreglo, luego retorna un boolean a partir de un operador ternario.

  Al final del archivo se exporta una instancia de la clase *ModalidadesController*.

```
class ModalidadesController {
    ingresar(modalidad) {
        modalidad.id = uuidv4();
        modalidades.push(modalidad);

        return modalidad.id;
    }

    mostrar() {
        return modalidades;
    }

    existeModalidad(idModalidad) {
        let existe = [];

        modalidades.find((m) => {
            if (m.id === idModalidad) {
                existe.push(true);
            }
        });

        return (existe.length)? true : false;
    }
}
```

Las rutas para acceder a la información de las modalidades están en el archivo *routes/modalidades.router.js* y estas son:

- GET localhost:3000/modalidades/view: Obtiene todas las modalidades a partir del método *mostrar* del controlador *ModalidadesController* y luego renderiza el archivo *views/modalidades.jade* pasándole los datos de “title” y el arreglo de modalidades.
- GET localhost:3000/modalidades: Devuelve un JSON con el arreglo de modalidades.
- POST localhost:3000/modalidades: Pide en el body de la petición el nombre de la modalidad en un JSON con el atributo “modalidad”. Devuelve un JSON con el identificador de la nueva modalidad.

```
const ModalidadesController = require('../controllers/modalidades.controller');
var express = require('express');

var router = express.Router();


router.get('/view', function(req, res, next) {
    const modalidades = ModalidadesController.mostrar();
    res.render('modalidades', {
        title: 'Modalidades',
        modalidades
    });
});

router.get('/', function(req, res, next) {
    const modalidades = ModalidadesController.mostrar();
    res.send(modalidades);
});

router.post('/', function(req, res, next) {
    const idModalidad = ModalidadesController.ingresar(req.body);
    res.json({ idModalidad });
})

module.exports = router;
```

Categorías

Las categorías se encuentran en la carpeta controllers/categorias.controller.js y tienen la siguiente estructura: identificador (ID), categoria (nombre de la categoría), idModalidad (identificador de la modalidad a la que pertenece).

```
let categorias = [
    {
        "id": "50bcfbdf-f87e-46f3-835b-55d90e037bd1",
        "categoria": "Zumo",
        "idModalidad": "e7b26ae3-ee9a-48ac-97d6-c824641627c0"
    },
    {
        "id": "19eb4bdb-7819-4e92-ace2-8fb025b82bac",
        "categoria": "Boxeo",
        "idModalidad": "e7b26ae3-ee9a-48ac-97d6-c824641627c0"
    },
    {
        "id": "463ef141-2e3e-43ab-8ee4-1ac8cc7064e8",
        "categoria": "Base",
        "idModalidad": "186571f2-cf9a-43e2-bb0b-946ad3f7c622"
    }
];
```

El controlador de categorías, *CategoriasController*, tiene los siguentes métodos:

- ingresar: Tiene como parámetros el nombre de la categoría (categoria) y el identificador de la modalidad (idModalidad). Primero comprueba que la modalidad exista para después crear el identificador único con la librería uuid y finalmente agregar un JSON con los datos de la categoría en el arreglo de categorías. Retorna el identificador de la nueva categoría.
- editar: Tiene como parámetros el identificador de la categoría (id), el nombre de la categoría (categoria) y el identificador de la modalidad (idModalidad). Primero verifica

que exista la modalidad y luego busca la categoría en el arreglo a partir del identificador y una vez encontrada reemplaza los datos de la categoría por los nuevos datos.

- eliminar: Tiene como parámetro el identificador de la modalidad. Primero busca la categoría y luego utiliza el índice del bucle for para utilizar el método splice del objeto array para poder eliminar la categoría del arreglo.
- mostrar: Retorna el arreglo de categorías.
- existe: Tiene como parámetro el identificador de la categoría. El proceso es el mismo en todos los métodos "existe". Retorna un boolean a partir de un operator ternario.

Al final del archivo se exporta una instancia de la clase *CategoriasController*.

```
class CategoriasController {
    ingresar(categoria, idModalidad) {
        if (ModalidadesController.existeModalidad(idModalidad)) {
            const id = uuidv4();
            categorias.push({ id, categoria, idModalidad });

            return id;
        }
    }

    editar(id, categoria, idModalidad) {
        if (ModalidadesController.existeModalidad(idModalidad)) {
            for (const c of categorias) {
                if (c.id === id) {
                    c.categoria = categoria;
                    c.idModalidad = idModalidad;
                }
            }
        }
    }
```

```
    eliminar(id) {
        for (let i = 0; i < categorias.length; i++) {
            const categoria = categorias[i];

            if (categoria.id === id) {
                categorias.splice(i, 1);
            }
        }
    }

    mostrar() {
        return categorias;
    }

    existe(id) {
        let existe = [];

        categorias.find((c) => {
            if (c.id === id) {
                existe.push(true);
            }
        });

        return (existe.length)? true : false;
    }
}

module.exports = new CategoriasController();
```

Las rutas para acceder a la información de las categorías están en el archivo *routes/categorias.router.js* y estas son:

- GET localhost:3000/categorias: Devuelve la lista de categorías.
- POST localhost:3000/categorias: Pide en el body de la petición el nombre de la categoría (categoria) y el identificador de la modalidad (idModalidad). Retorna el identificador de la nueva categoría.

- PUT localhost:3000/categorias/:id: “:id” es el identificador de la categoría que se desea modificar. Pide en el body de la petición el nombre de la categoría (categoria) y el identificador de la modalidad (idModalidad). Devuelve la lista de categorías actualizada.
- DELETE localhost:3000/categorias/:id: “:id” es el identificador de la categoría que se desea eliminar. Devuelve la lista de categorías actualizada.

```
router.get('/', function(req, res, next) {
    const categorias = CategoriasController.mostrar();
    res.send(categorias);
});

router.post('/', function(req, res, next) {
    const { categoria, idModalidad } = req.body;
    const idCategoria = CategoriasController.ingresar(categoria, idModalidad);
    res.json({ idCategoria });
});

router.put('/:id', function(req, res, next) {
    const { categoria, idModalidad } = req.body;
    CategoriasController.editar(req.params.id, categoria, idModalidad);
    res.send(CategoriasController.mostrar());
});

router.delete('/:id', function(req, res, next) {
    CategoriasController.eliminar(req.params.id);
    res.send(CategoriasController.mostrar());
});
```

Equipos

Los equipos se encuentran en el archivo *controllers/equipos.controller.js* y tienen la siguiente estructura: id, integrantes (arreglo de strings), categorias (arreglo de identificadores). El controlador *EquiposController* tiene los siguientes métodos:

- insertar: Tiene los integrantes y las categorías como parámetros, estos son arreglos como se puede ver en los equipos ya registrados. Primero verifica la existencia de todas las

categorías y luego crea el identificador uuid para poder registrar el equipo en el arreglo de equipos.

- mostrar: Retorna el arreglo de equipos.
- editar: Tiene como parámetros el identificador del equipo (idEquipo), la lista de integrantes y la lista de categorías. Primero verifica que todas las categorías estén registradas y luego busca el equipo en el arreglo para después de encontrar el equipo actualizar los datos del equipo.
- eliminar: Tiene como parámetro el identificador del equipo (id). Busca el equipo con un bucle for y gracias al índice del bucle se elimina el equipo con el método splice del objeto array.
- mostrarPorCategoria: Tiene como parámetro el identificador de la categoría (idCategoria). Primero verifica que exista la categoría para luego guardar en un arreglo aquellos equipos que tengan registrada la categoría en sus categorías. Retorna el arreglo de equipos inscritos.
- eliminarInscripcion: Tiene como parámetros el identificador del equipo (idEquipo) y el identificador de la categoría cuya inscripción se desea eliminar (idCategoria). Primero verifica que la categoría esté ya registrada para luego buscar el equipo obtener el índice de la inscripción en el arreglo de categorías del equipo por el método indexOf del objeto array y eliminar el registro con el método splice del objeto array.

```
class EquiposController {
    insertar(integrantes, categorias) {
        let existenCategorias = [1];

        for (const categoria of categorias) {
            if (!CategoriasController.existe(categoria)) {
                existenCategorias.splice(0, 1);
                break;
            }
        }

        if (existenCategorias.length) {
            const id = uuidv4();
            equipos.push({ id, integrantes, categorias });
            return id;
        }
    }

    mostrar() {
        return equipos;
    }

    editar(idEquipo, integrantes, categorias) {
        let existenCategorias = [1];

        for (const categoria of categorias) {
            if (!CategoriasController.existe(categoria)) {
                existenCategorias.splice(0, 1);
                break;
            }
        }
```

```
        if (existenCategorias.length) {
            equipos.find((e) => {
                if (e.id === idEquipo) {
                    e.integrantes = integrantes;
                    e.categorias = categorias;
                }
            });
        }
    }

    eliminar(id) {
        for (let i = 0; i < equipos.length; i++) {
            const equipo = equipos[i];

            if (equipo.id === id) {
                equipos.splice(i, 1);
            }
        }
    }

    mostrarPorCategoria(idCategoria) {
        if (CategoriasController.existe(idCategoria)) {
            let equiposInscritos = [];

            for (const equipo of equipos) {
                for (const categoria of equipo.categorias) {
                    if (categoria === idCategoria) {
                        equiposInscritos.push(equipo);
                    }
                }
            }
```

```
            return equiposInscritos;
        }
    }

    eliminarInscripcion(idEquipo, idCategoria) {
        if (CategoriasController.existe(idCategoria)) {
            for (const equipo of equipos) {
                if (equipo.id === idEquipo) {
                    const i = equipo.categorias.indexOf(idCategoria);
                    console.log('i ' + i);
                    equipo.categorias.splice(i, 1);
                }
            }
        }
    }

    existe(id) {
        let existe = [];

        equipos.find((e) => {
            if (e.id === id) {
                existe.push(true);
            }
        });

        return (existe.length)? true : false;
    }
}

module.exports = new EquiposController();
```

Las rutas para acceder a la información de los equipos están en el archivo *routes/equipos.router.js* y estas son:

- GET localhost:3000/equipos/view: Obtiene el arreglo de equipos y renderiza el archivo *views/equipos.jade* pasándole los datos equipos y el “title”.
- GET localhost:3000/equipos: Obtiene el arreglo de equipos y devuelve el arreglo.

- GET localhost:3000/equipos/:idCategoria: “:idCategoria” es el identificador de la categoría para saber aquellos equipos inscritos en la categoría. Devuelve el arreglo con los equipos inscritos.
- POST localhost:3000/equipos: Pide la lista de integrantes y la lista de categorías en el body de la petición, luego devuelve un JSON con el identificador del nuevo equipo.
- PUT localhost:3000/equipos/:id: “:id” es el identificador del equipo que se desea modificar. Pide en el body de la petición la lista de integrantes y la lista de categorías. Devuelve la lista de equipos actualizada.
- DELETE localhost:3000/equipos/:id: “:id” es el identificador del equipo que se desea eliminar. Devuelve la lista de equipos actualizada.
- DELETE localhost:3000/equipos/:idEquipo/:idCategoria: “:idEquipo” es el identificador del equipo, “:idCategoria” es el identificador de la categoría. Devuelve la lista de eupos actualizada.

```
router.get('/view', function(req, res, next) {
    const equipos = EquiposController.mostrar();
    res.render('equipos', {
        equipos: equipos,
        title: 'Equipos'
    });
});

router.get('/', function(req, res, next) {
    const equipos = EquiposController.mostrar();
    res.send(equipos);
});

router.get('/:idCategoria', function(req, res, next) {
    const equipos = EquiposController.mostrarPorCategoria(req.params.idCategoria);
    res.send(equipos);
});

router.post('/', function(req, res, next) {
    const { integrantes, categorias } = req.body;
    const idEquipo = EquiposController.insertar(integrantes, categorias);
    res.send({ idEquipo });
});

router.put('/:id', function(req, res, next) {
    const { integrantes, categorias } = req.body;
    EquiposController.editar(req.params.id, integrantes, categorias);
    res.send(EquiposController.mostrar());
});


router.delete('/:id', function(req, res, next) {
    EquiposController.eliminar(req.params.id);
    res.send(EquiposController.mostrar());
});

router.delete('/:idEquipo/:idCategoria', function(req, res, next) {
    const { idEquipo, idCategoria } = req.params;
    EquiposController.eliminarInscripcion(idEquipo, idCategoria);
    res.send(EquiposController.mostrar());
})
```