LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

PROYECTO BINGO FOPRE

PRUEBAS DE FUNCIONALIDAD PARA LA BASE DE DATOS Y APLICATIVO CONSUMIDAS EN POSTMAN

JUAN CAMILO RODRIGUEZ RAMIREZ
UNIVERSIDAD DE LOS ANDES COLOMBIA – DIRECCIÓN DE RELACIONAMIENTO
2023

CONTENIDO

LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

INTRODUCCIÓN

El siguiente manual de Pruebas del Bingo Fopre desarrollado en el framework Laravel v10, donde se desarrollaron las pruebas utilizando Postman. Este manual está diseñado para proporcionar una guía completa sobre cómo evaluar y verificar la funcionalidad de la aplicación, así como para asegurar su calidad y fiabilidad.

OBJETIVO

El objetivo principal de este manual es ofrecer a los miembros del equipo de desarrollo, QA y cualquier otra persona interesada en el proyecto, una referencia detallada de las pruebas que se han llevado a cabo. Además, proporciona información sobre cómo ejecutar estas pruebas utilizando Postman, una herramienta popular para probar API´s.

REQUISITOS PREVIOS

Antes de comenzar a utilizar este manual, es necesario tener instalado y configurado Postman en su entorno de desarrollo. Asegúrese de contar con una copia funcional de la aplicación Laravel y de disponer de acceso a la API para realizar pruebas.

ESTRUCTURA DEL MANUAL

Este manual se organiza en secciones que corresponden a las áreas mencionadas anteriormente. Cada sección contiene instrucciones detalladas sobre cómo realizar las pruebas, así como ejemplos de solicitudes y respuestas.

SECCIONES

En cada sección se podrá evidenciar el código necesario y la petición para realizar cada una de las pruebas con Postman.

CONFIGURACIÓN DE ESTADOS

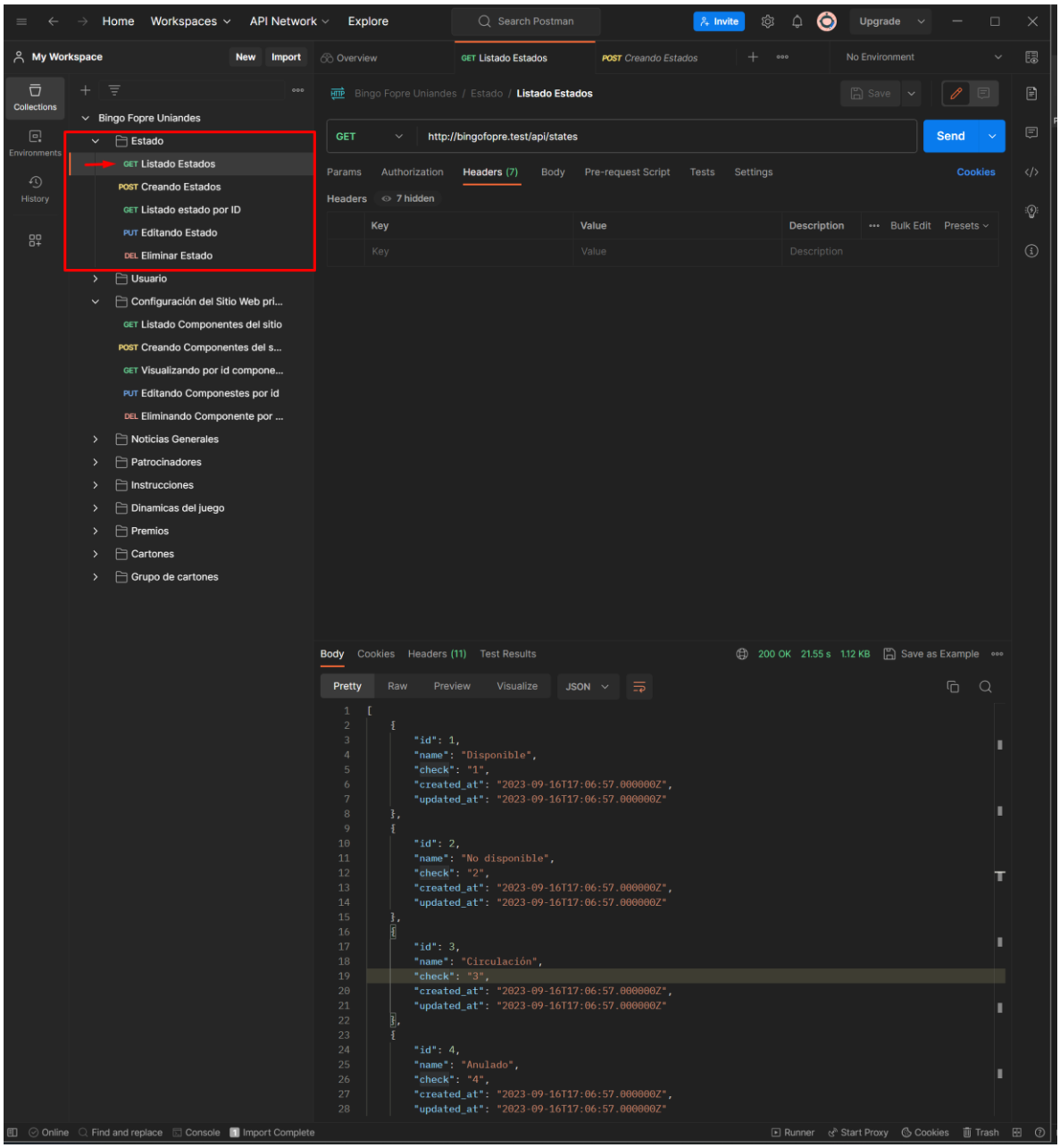
Ruta: esto lo podrá encontrar App/Http/Controller/Api/States/StatesController.php

Función para listar todos los estados.

```php
class StatesController extends Controller
{
    /**
     * Display a listing of the resource.
     */
    public function index()
    {
        try {
            $states = State:: all();
            return response()->json($states, 200);
        } catch (\Throwable $th) {
            return response()->json([
                'errors' => $th
            ], 400);

        }

    }
```

Petición Postman: aquí se recibe la petición de todos los estados.

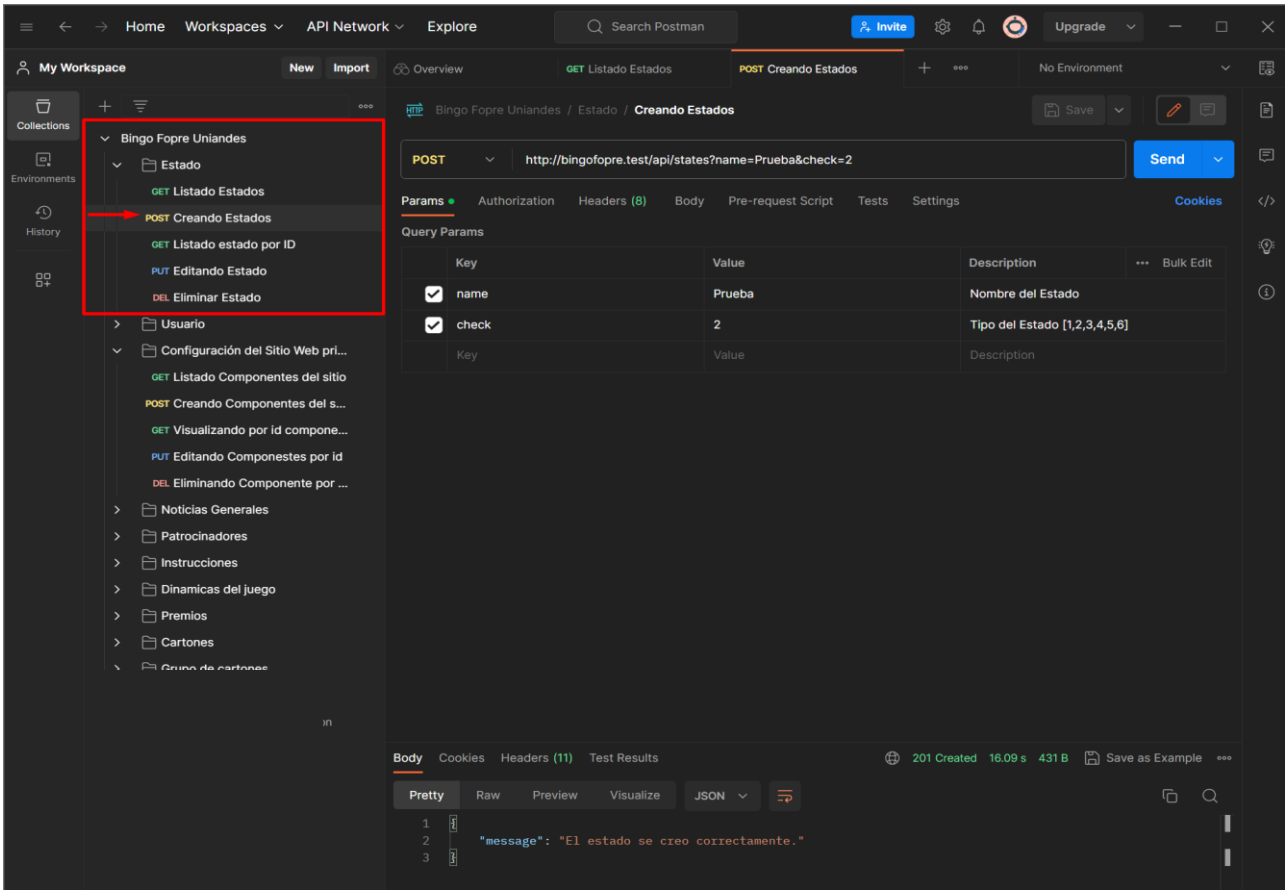
LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ



LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

Función para crear estados

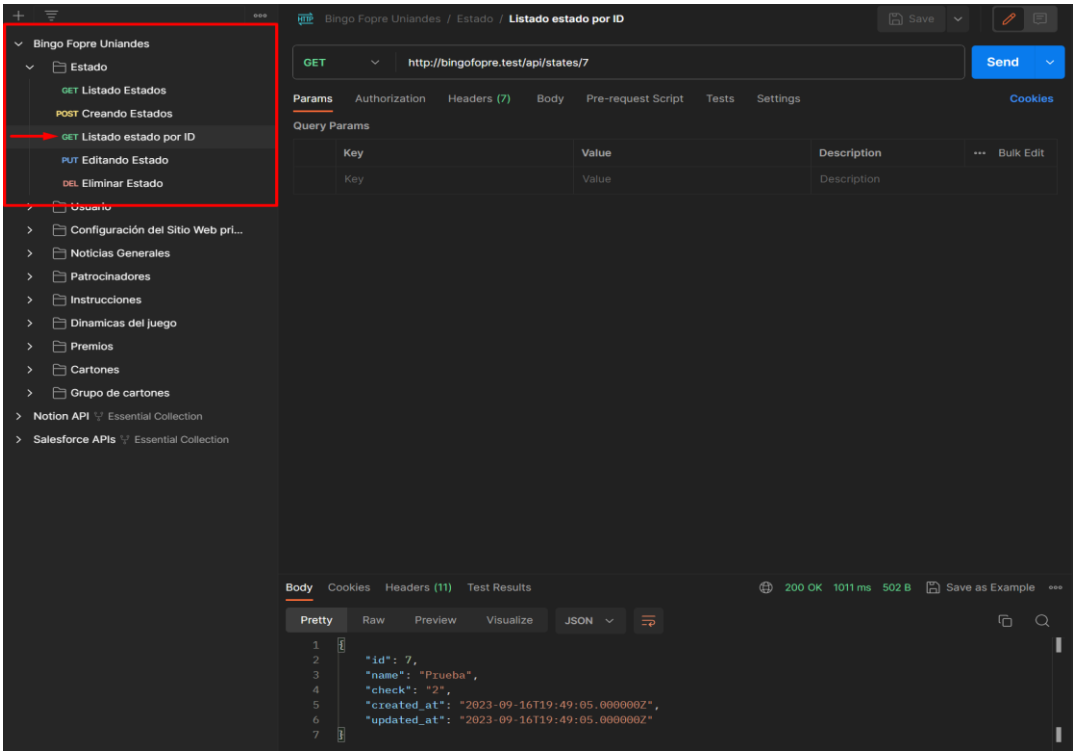
```php
/**
 * Show the form for creating a new resource.
 */


/**
 * Store a newly created resource in storage.
 */
public
function store(Request $request)
{
    try {
        $state = State:: create($request->all());
        return response()->json(['message'=> 'El estado se creo correctamente.'],201);
    } catch (\Throwable $th) {
        return response()->json([
            'errors' => $th
        ], 400);
    }
}
```



LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

Función para visualizar estados por ID

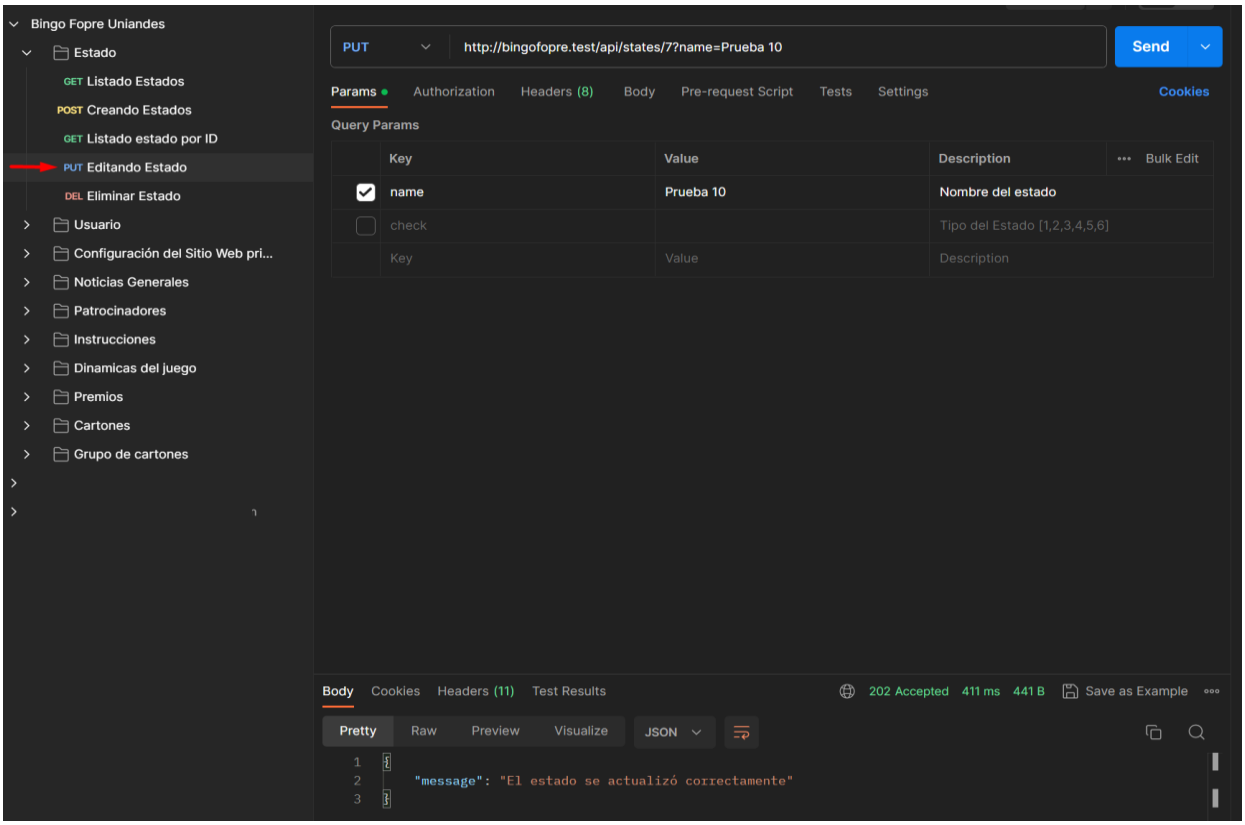
```php
/**
 * Display the specified resource.
 */
public
function show(string $id)
{
    try {
        $state = State:: find($id);
        return response()->json($state, 200);
    } catch (\Throwable $th) {
        return response()->json([
            'errors' => $th
        ], 400);

    }
}
```



LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

Función para editar estado por ID

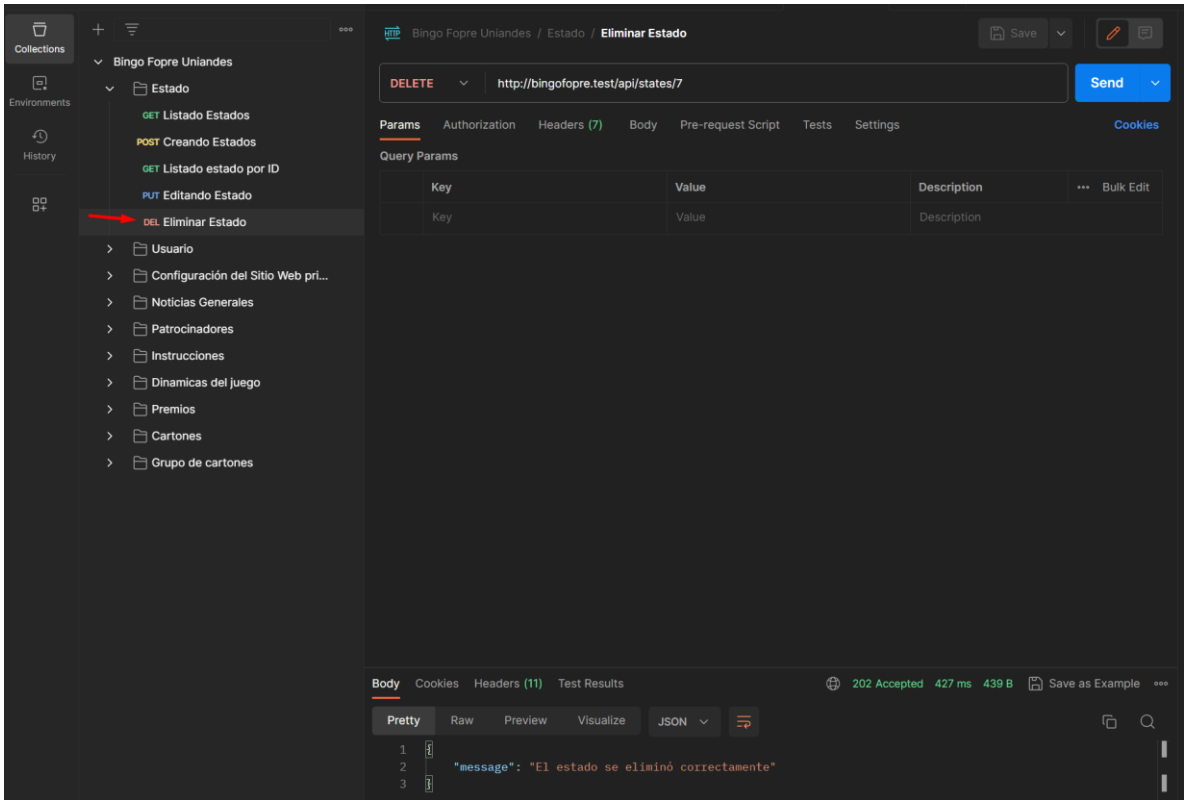
```php
/**
 * Show the form for editing the specified resource.
 */


/**
 * Update the specified resource in storage.
 */
public
function update(Request $request, string $id)
{
    try {
        $state = State:: find($id)->update($request->all());
        return response()->json([ 'message' => 'El estado se actualizó correctamente'],202);
    } catch (\Throwable $th) {
        return response()->json([
            'errors' => $th
        ], 400);
    }
}
```



LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

Función para eliminar estados

```php
/**
 * Remove the specified resource from storage.
 */
public
function destroy(string $id)
{
    try {
        $state = State:: find($id)->delete();
        return response()->json([ 'message' => 'El estado se eliminó correctamente'],202);
    } catch (\Throwable $th) {
        return response()->json([
            'errors' => $th
        ], 400);

    }
}
```



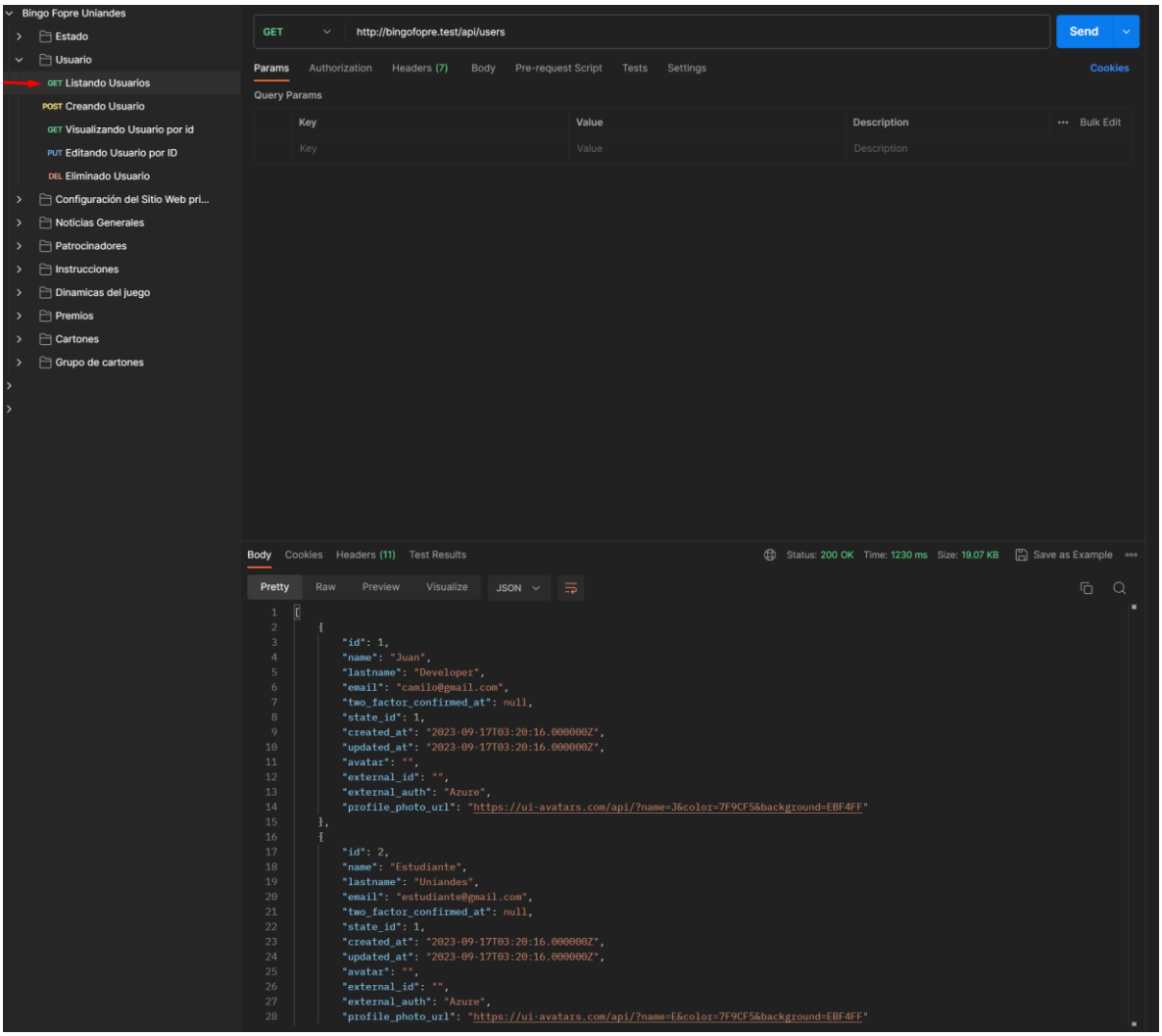
CONFIGURACIÓN DEL USUARIOS

Ruta: /App/Http/Controllers/Api/Users/UsersController.php

Función para listar usuarios

```php
class UsersController extends Controller
{
    /**
     * Display a listing of the resource.
     */
    public function index()
    {
        try {
            $users = User::all();
            return response()->json($users, 200);
        } catch (\Throwable $th) {
            return response()->json([
                'errors' => $th
            ], 400);

        }

    }
```

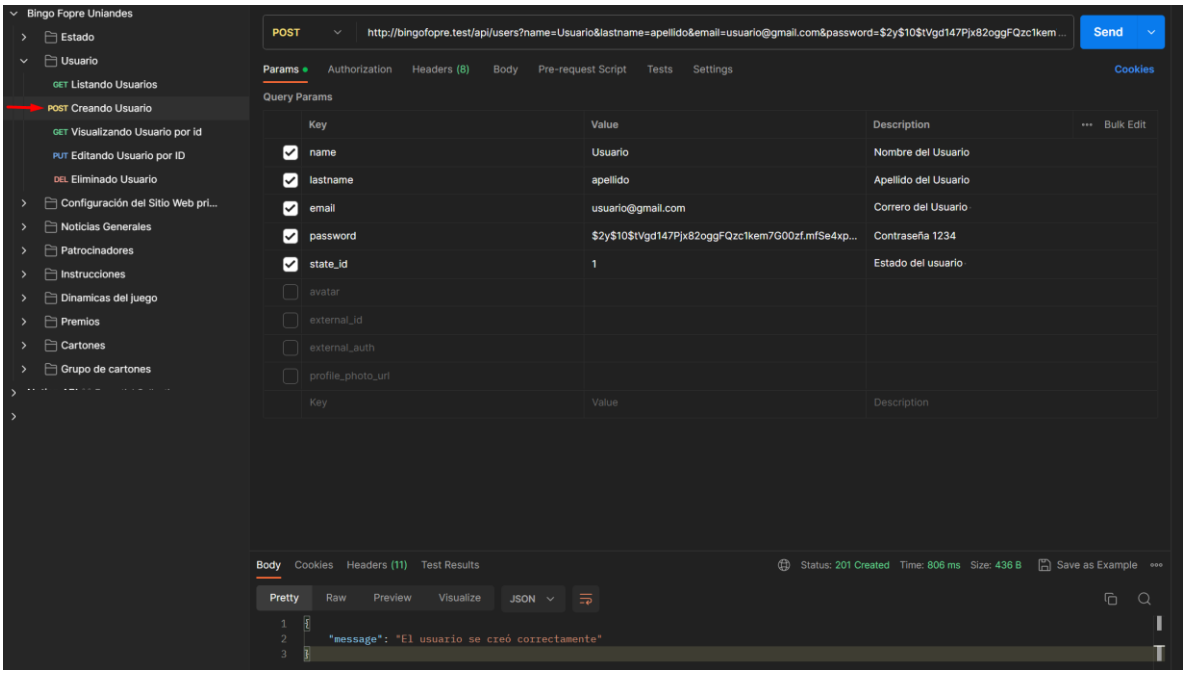
LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ



LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

Función para crear usuarios

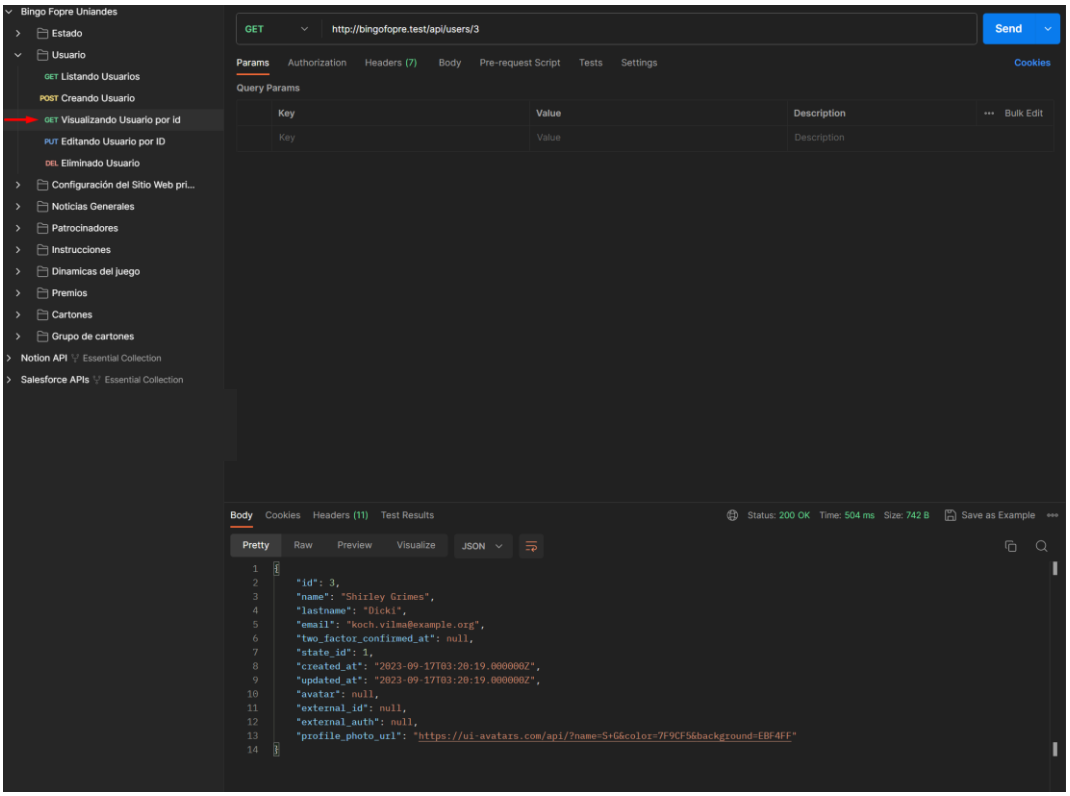
```php
    /**
     * Show the form for creating a new resource.
     */


    /**
     * Store a newly created resource in storage.
     */
    public function store(Request $request)
    {
        try {
            $user = User:: create($request->all());
            return response()->json(['message' => 'El usuario se creó correctamente'], 201);
        } catch (\Throwable $th) {
            return response()->json([
                'errors' => $th
            ], 400);
        }
    }
```



LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

Función para visualizar usuarios por ID

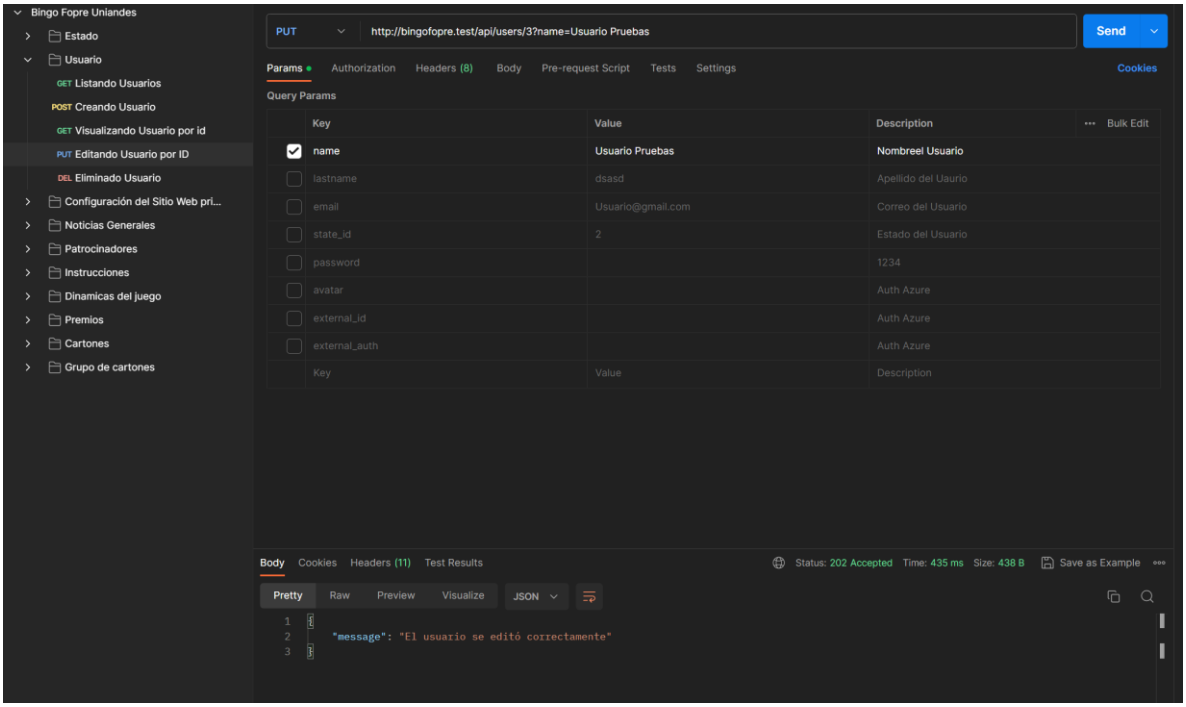
```php
/**
 * Display the specified resource.
 */
public
function show(string $id)
{
    try {
        $user = User:: find($id);
        return response()->json($user, 200);
    } catch (\Throwable $th) {
        return response()->json([
            'errors' => $th
        ], 400);

    }
}
```



LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

Función parar editar usuarios por id

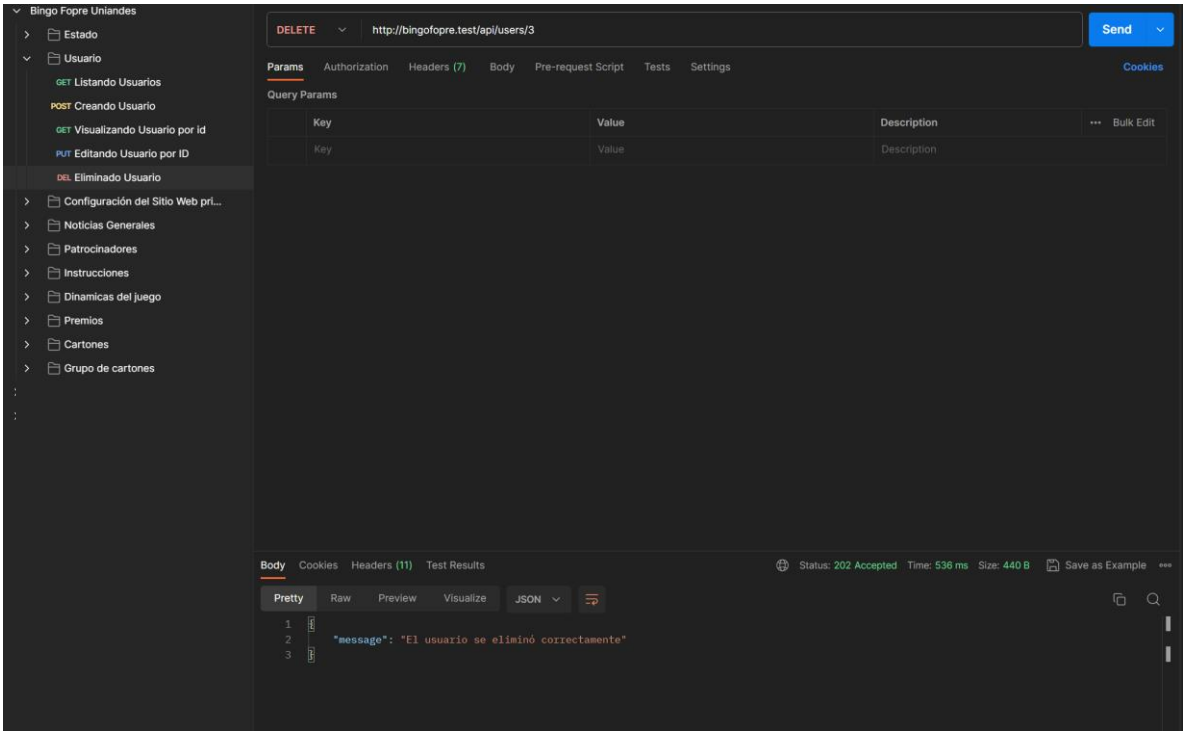
```php
/**
 * Show the form for editing the specified resource.
 */


/**
 * Update the specified resource in storage.
 */
public
function update(Request $request, string $id)
{
    try {
        $user = User:: find($id)->update($request->all());
        return response()->json(['message' => 'El usuario se editó correctamente'], 202);
    } catch (\Throwable $th) {
        return response()->json([
            'errors' => $th
        ], 400);
    }
}
```



LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

Función para eliminar de usuarios por id

```php
/**
 * Remove the specified resource from storage.
 */
public
function destroy(string $id)
{
    try {
        $user = User:: find($id)->delete();
        return response()->json(['message' => 'El usuario se eliminó correctamente'], 202);
    } catch (\Throwable $th) {
        return response()->json([
            'errors' => $th
        ], 400);

    }
}
```



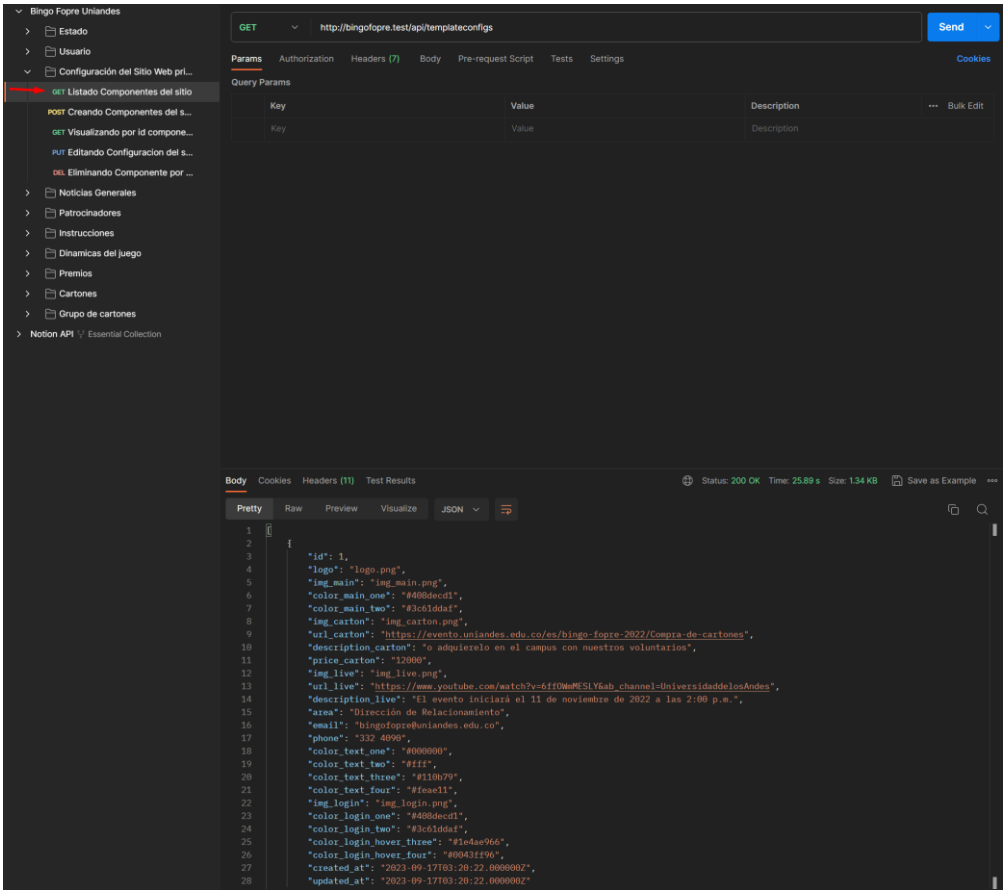
LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

CONFIGURACIÓN DEL SITIO WEB

Ruta: /App/Http/Controllers/Api/TemplateConfigs/TemplateconfigController.php

Función para listar componentes de sitio

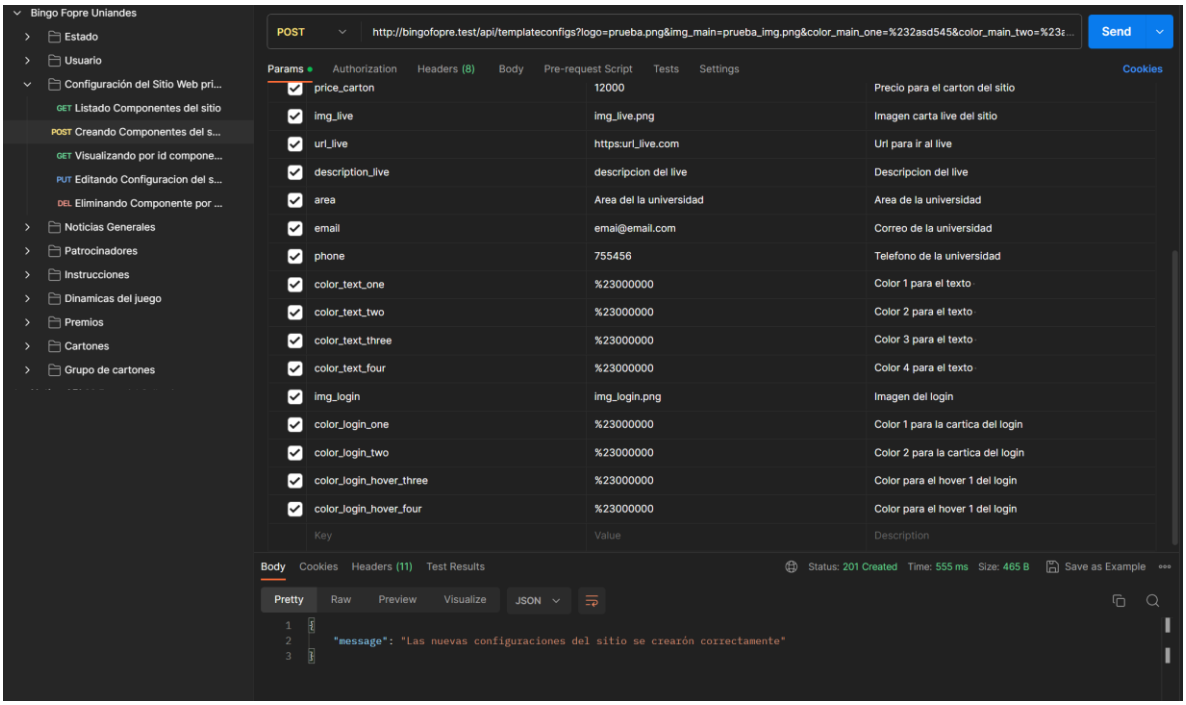
```php
class TemplateConfigsController extends Controller
{
    /**
     * Display a listing of the resource.
     */
    public function index()
    {
        try {
            $templateconfigs = TemplateConfig:: all();
            return response()->json($templateconfigs, 200);
                  \Throwable $th) {
                  n response()->json([
                'errors' => $th
            ], 400);

        }

    }
```



LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

Función para crear componente del sitio

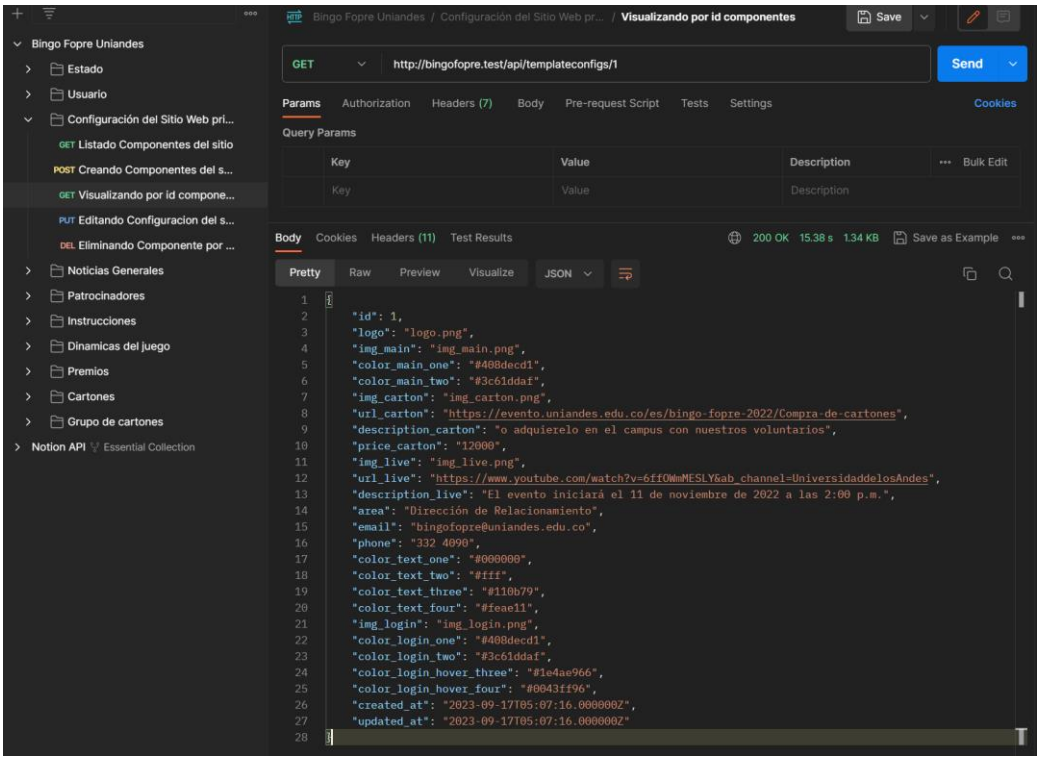
```php
/**
 * Show the form for creating a new resource.
 */


/**
 * Store a newly created resource in storage.
 */
public
function store(Request $request)
{
    try {
        $templateconfig = TemplateConfig:: create($request->all());
        return response()->json(['message'=> 'Las nuevas configuraciones del sitio se crearón correctamente'],201);
    } catch (\Throwable $th) {
        return response()->json([
            'errors' => $th
        ], 400);
    }
}
```



LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

Función para visualizar id de componentes

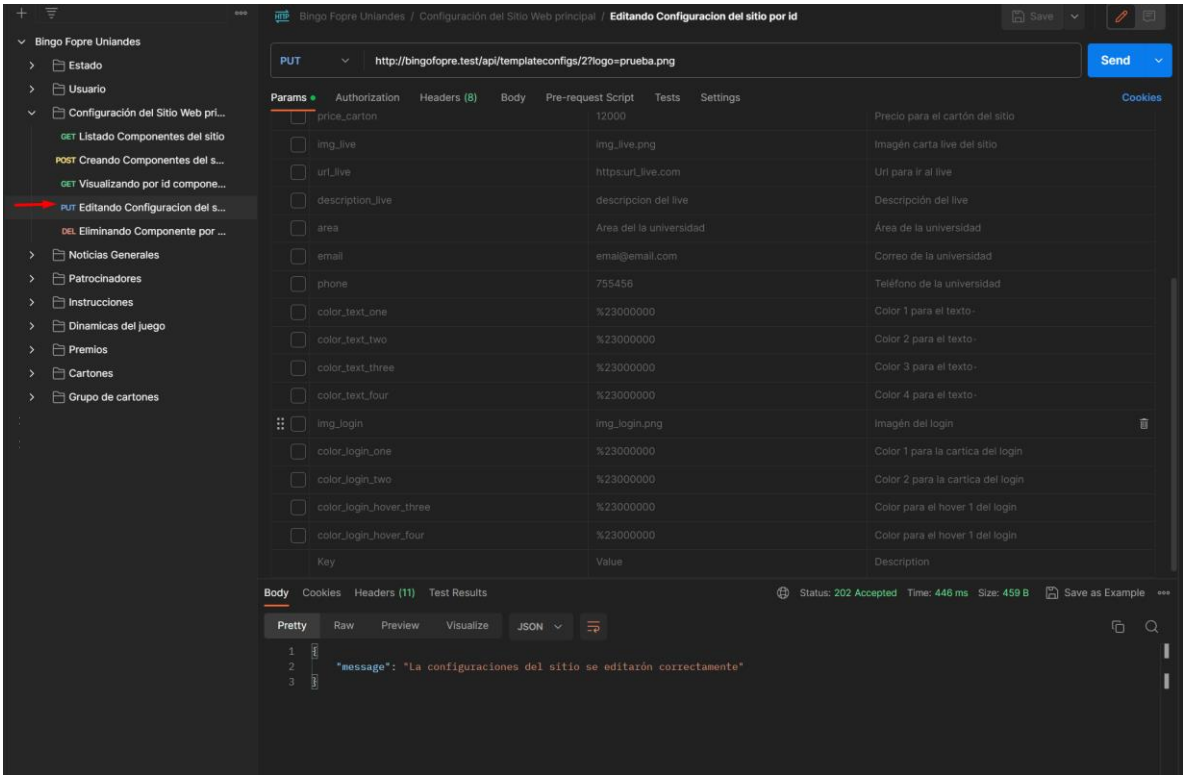
```php
/**
 * Display the specified resource.
 */
public
function show(string $id)
{
    try {
        $templateconfig = TemplateConfig:: find($id);
        return response()->json($templateconfig, 200);
    } catch (\Throwable $th) {
        return response()->json([
            'errors' => $th
        ], 400);

    }
}
```



LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

Función para editar componentes

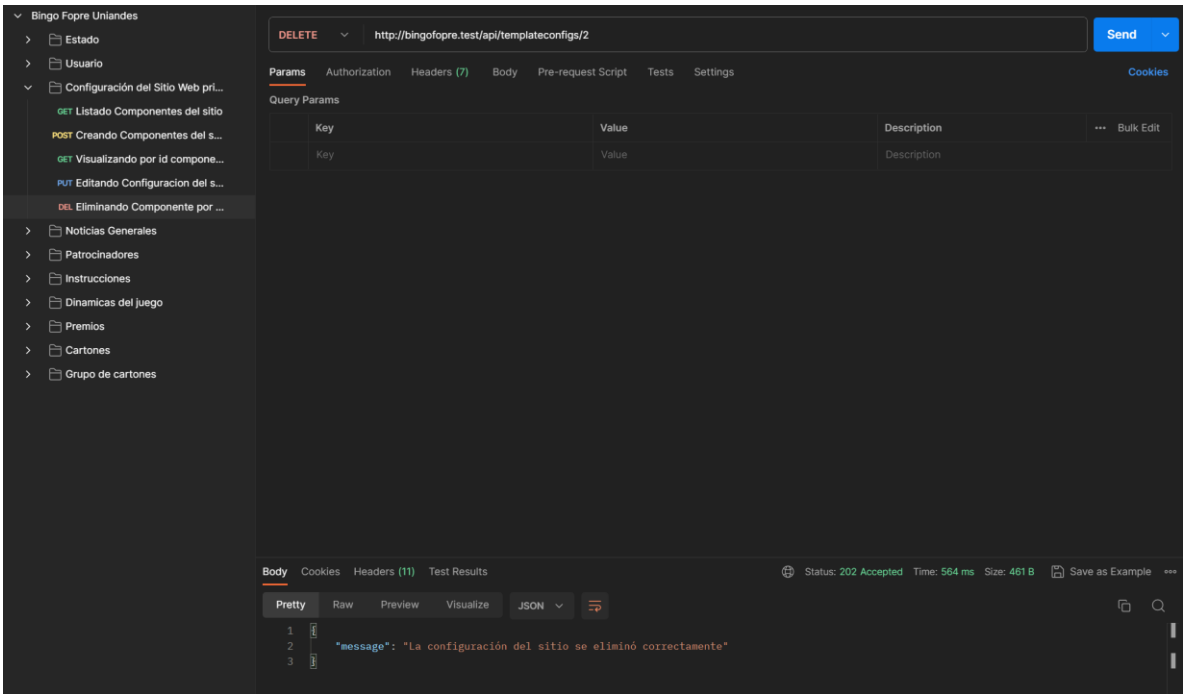
```php
/**
 * Show the form for editing the specified resource.
 */


/**
 * Update the specified resource in storage.
 */
public
function update(Request $request, string $id)
{
    try {
        $templateconfig = TemplateConfig:: find($id)->update($request->all());
        return response()->json([ 'message' => 'La configuraciones del sitio se editarón correctamente'],202);
    } catch (\Throwable $th) {
        return response()->json([
            'errors' => $th
        ], 400);
    }
}
```



LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

Función para eliminación de componentes

```php
/**
 * Remove the specified resource from storage.
 */
public
function destroy(string $id)
{
    try {
        $templateconfig = TemplateConfig:: find($id)->delete();
        return response()->json([ 'message' => 'La configuración del sitio se eliminó correctamente'],202);
    } catch (\Throwable $th) {
        return response()->json([
            'errors' => $th
        ], 400);

    }
}
```



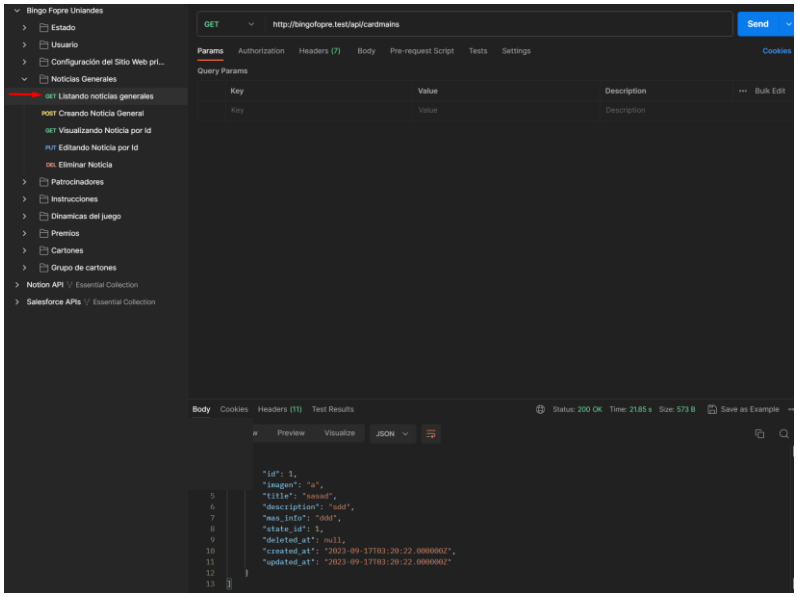
LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

NOTICIAS GENERALES

Ruta: /App/Http/Controllers/Api/CardMains/CardMainsController.php

Función para listar noticias

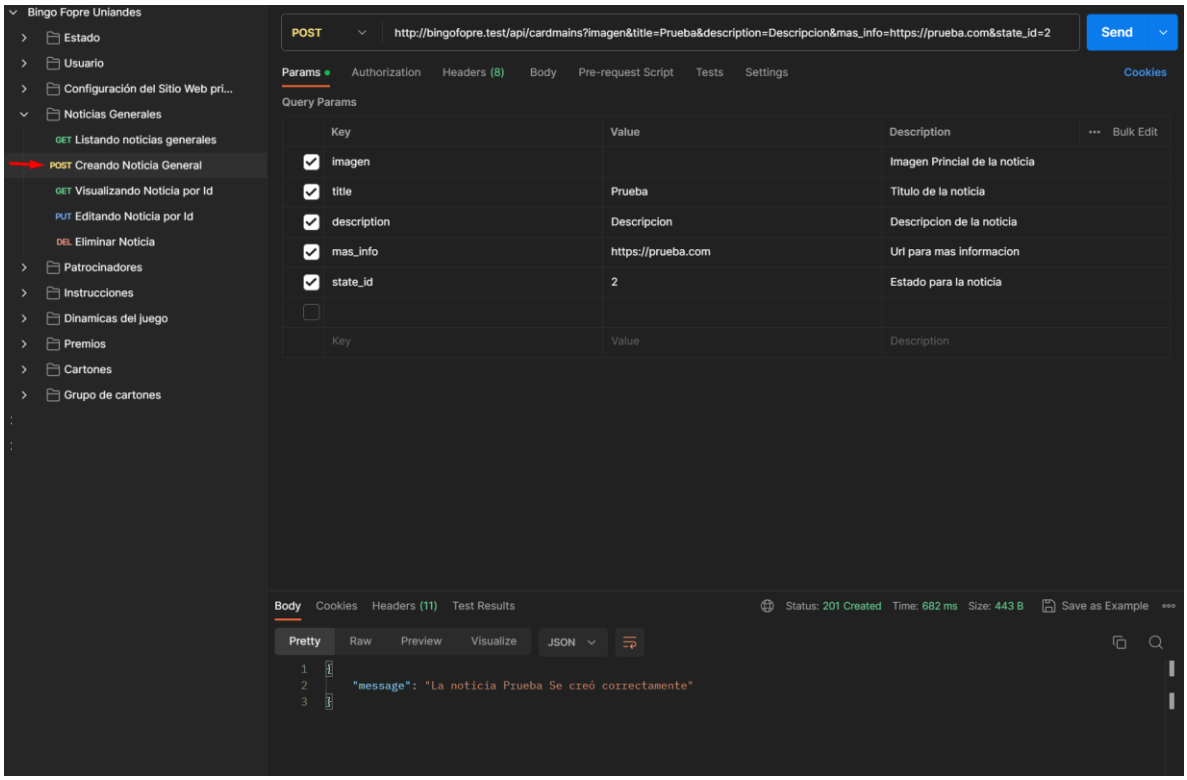
```php
class CardMainsController extends Controller
{
    /**
     * Display a listing of the resource.
     */
    public function index()
    {
        try {
            $cardmains = CardMain::all();
            return response()->json($cardmains, 200);
        } catch (\Throwable $th) {
            return response()->json([
                'errors' => $th
            ], 400);

        }

    }
```



LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

Función para crear noticias

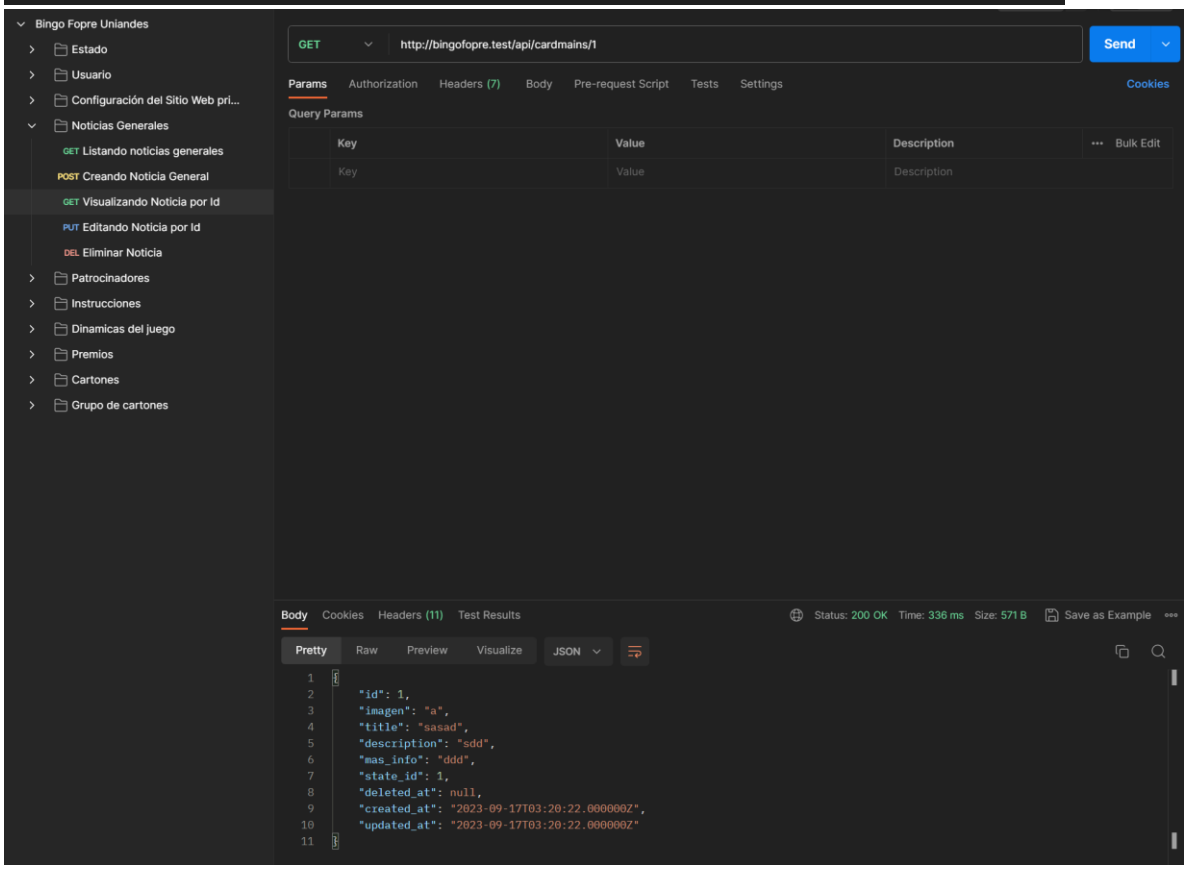
```php
/**
 * Show the form for creating a new resource.
 */


/**
 * Store a newly created resource in storage.
 */
public function store(Request $request)
{
    try {
        $cardmain = CardMain:: create($request->all());
        return response()->json(['message' => 'La noticia ' . $cardmain->title . ' Se creó correctamente'], 201);
    } catch (\Throwable $th) {
        return response()->json([
            'errors' => $th
        ], 400);
    }
}
```



LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

Función para Visualizar noticias

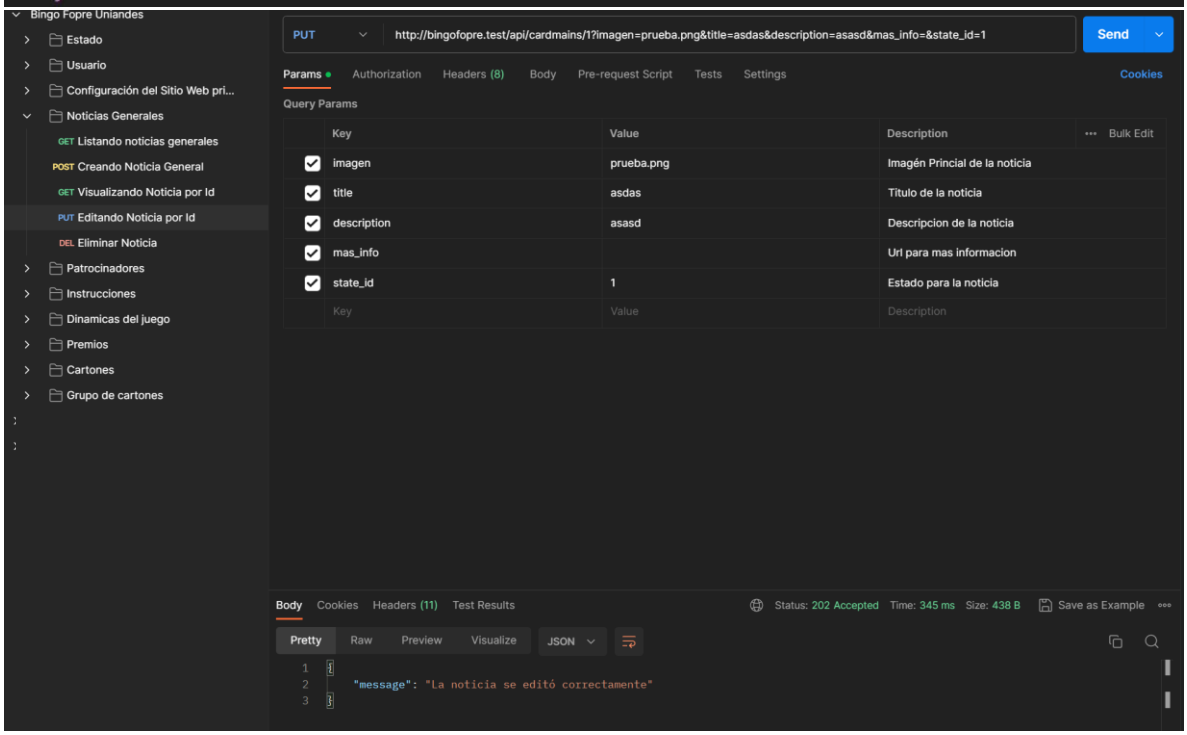
```php
/**
 * Display the specified resource.
 */
public
function show(string $id)
{
    try {
        $cardmain = CardMain:: find($id);
        return response()->json($cardmain, 200);
    } catch (\Throwable $th) {
        return response()->json([
            'errors' => $th
        ], 400);

    }
}
```



LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

Función para editar noticias general

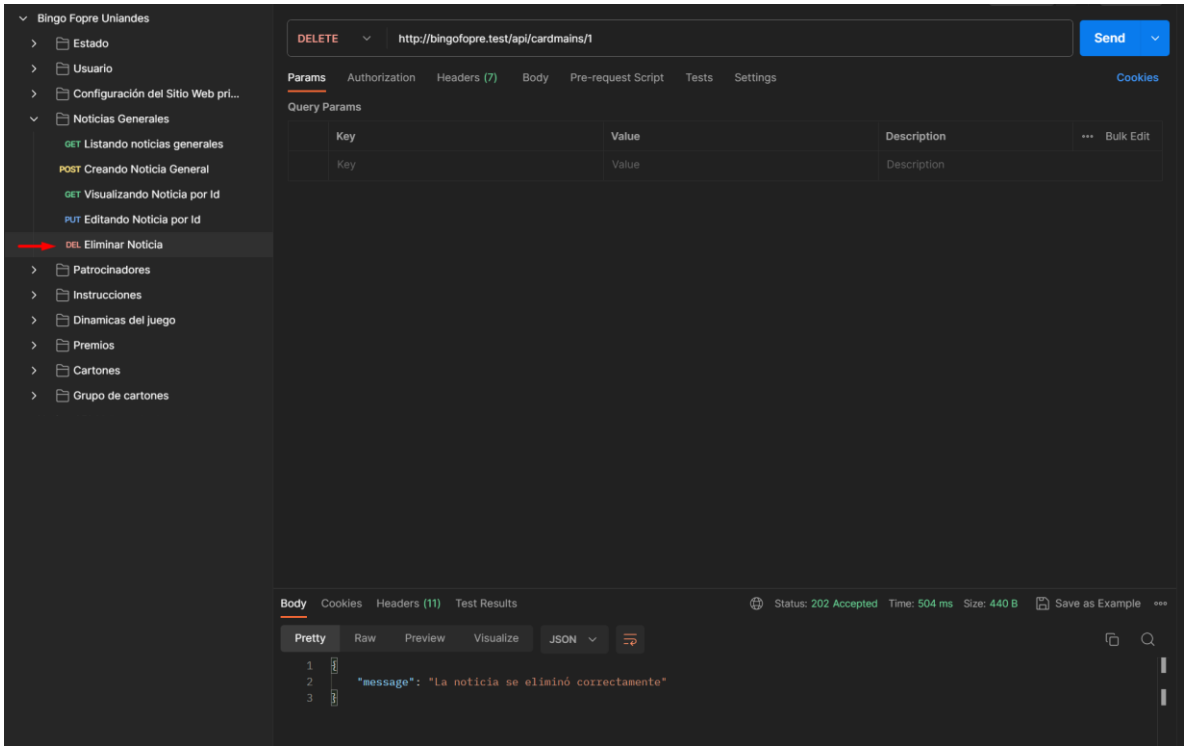
```php
/**
 * Show the form for editing the specified resource.
 */


/**
 * Update the specified resource in storage.
 */
public
function update(Request $request, string $id)
{
    try {
        $cardmain = CardMain:: find($id)->update($request->all());
        return response()->json(['message' => 'La noticia se editó correctamente'], 202);
    } catch (\Throwable $th) {
        return response()->json([
            'errors' => $th
        ], 400);
    }
}
```



LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

Función para eliminar noticias general

```php
/**
 * Remove the specified resource from storage.
 */
public
function destroy(string $id)
{
    try {
        $cardmain = CardMain:: find($id)->delete();
        return response()->json(['message' => 'La noticia se eliminó correctamente'], 202);
    } catch (\Throwable $th) {
        return response()->json([
            'errors' => $th
        ], 400);

    }
}
```



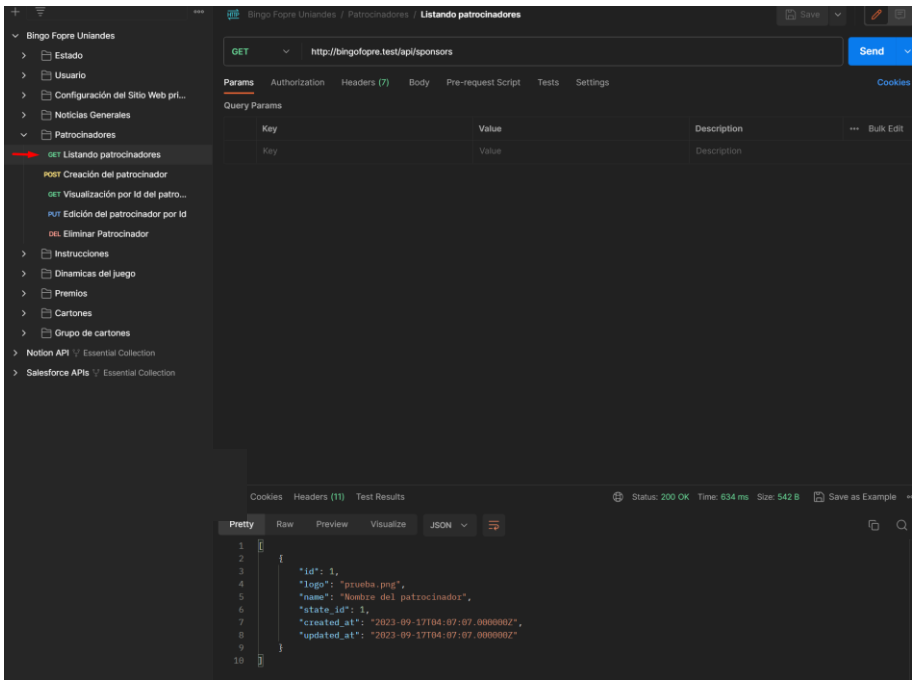
LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

PATROCINADORES

Ruta: /App/Http/Controllers/Api/Sponsor/SponsorController.php

Función para listar patrocinadores

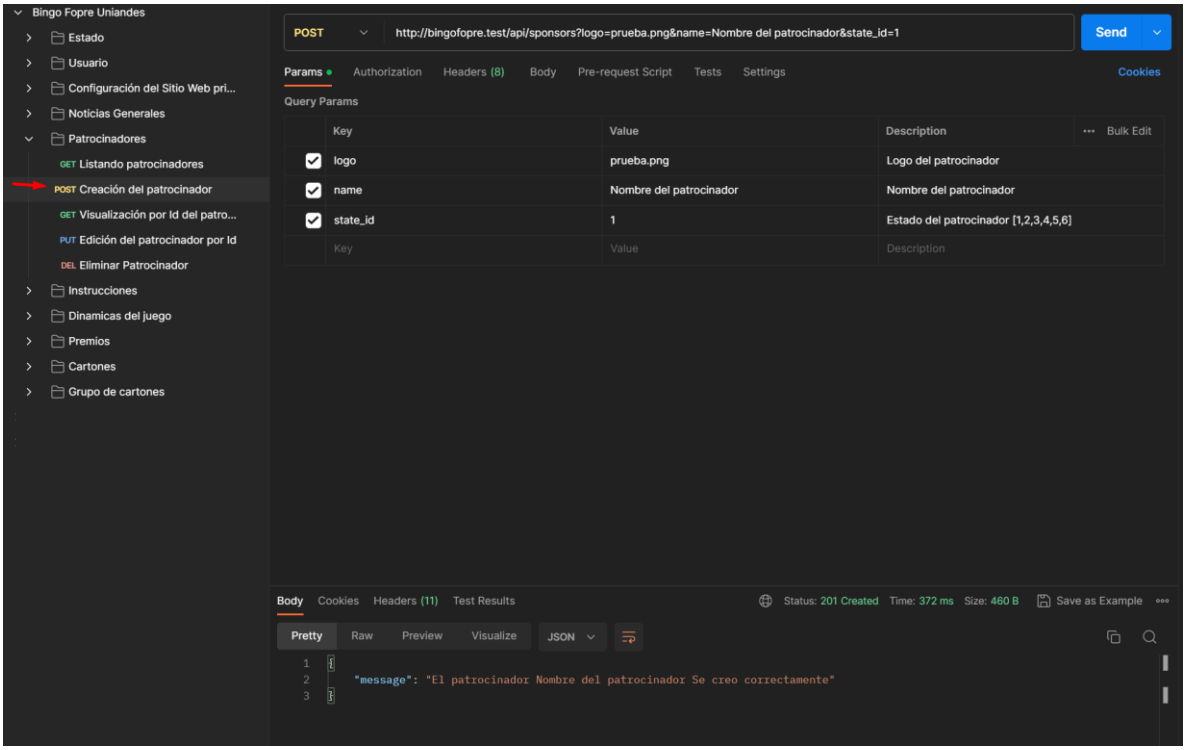
```php
class SponsorsController extends Controller
{
    /**
     * Display a listing of the resource.
     */
    public function index()
    {
        try {
            $sponsors = Sponsor::all();
            return response()->json($sponsors, 200);
        } catch (\Throwable $th) {
            return response()->json([
                'errors' => $th
            ], 400);

        }

    }
```



LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

Función para crear patrocinadores

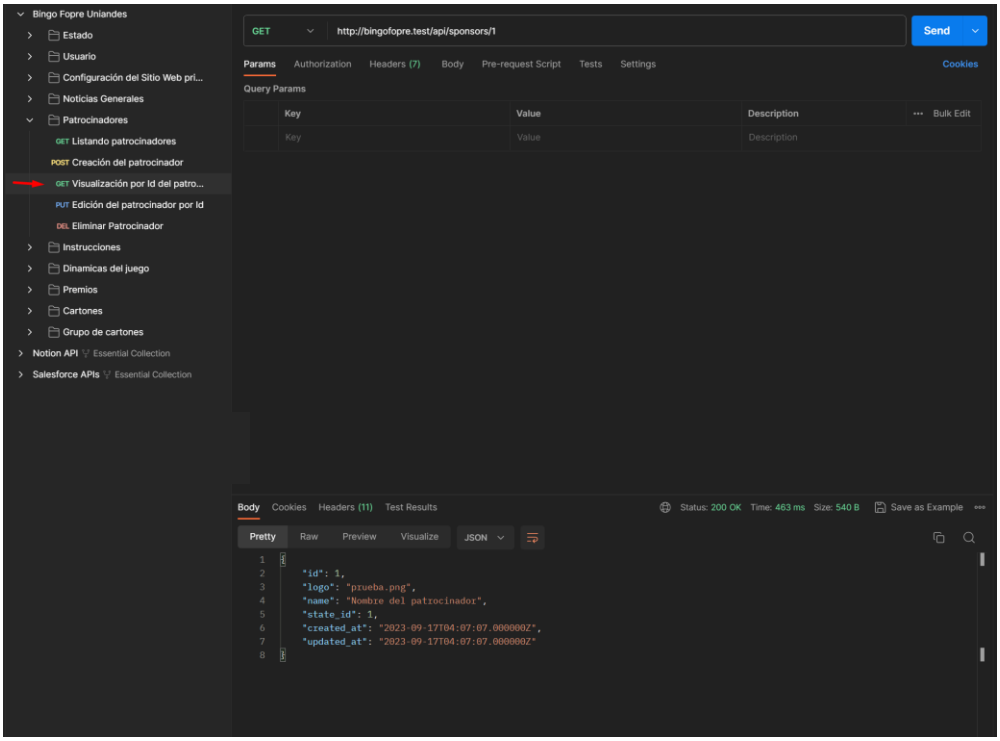
```php
public function store(Request $request)
{
    try {
        $sponsor = Sponsor:: create($request->all());
        return response()->json(['message' => 'El patrocinador ' . $sponsor->name . ' Se creo correctamente'], 201);
    } catch (\Throwable $th) {
        return response()->json([
            'errors' => $th
        ], 400);
    }
}
```



LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

Función para visualizar patrocinadores por ID

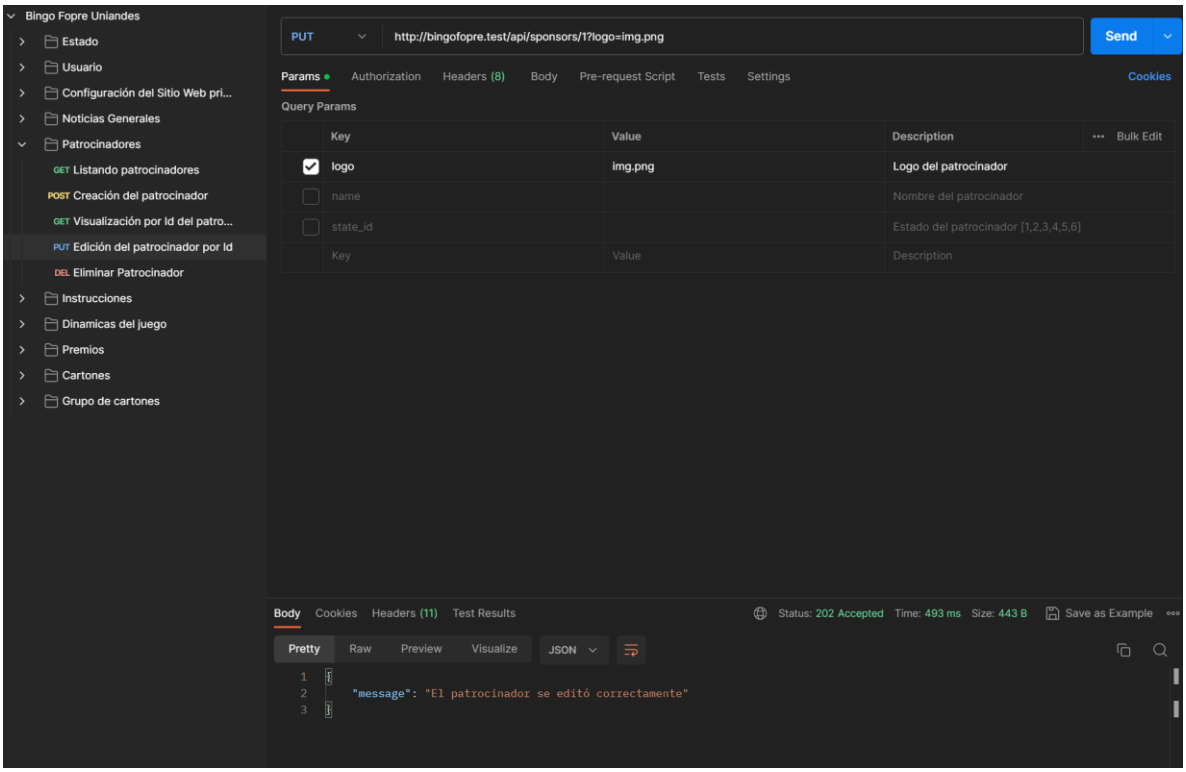
```php
/**
 * Display the specified resource.
 */
public
function show(string $id)
{
    try {
        $sponsor = Sponsor:: find($id);
        return response()->json($sponsor, 200);
    } catch (\Throwable $th) {
        return response()->json([
            'errors' => $th
        ], 400);

    }
}
```



LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

Función para edicitar patrocinador por id

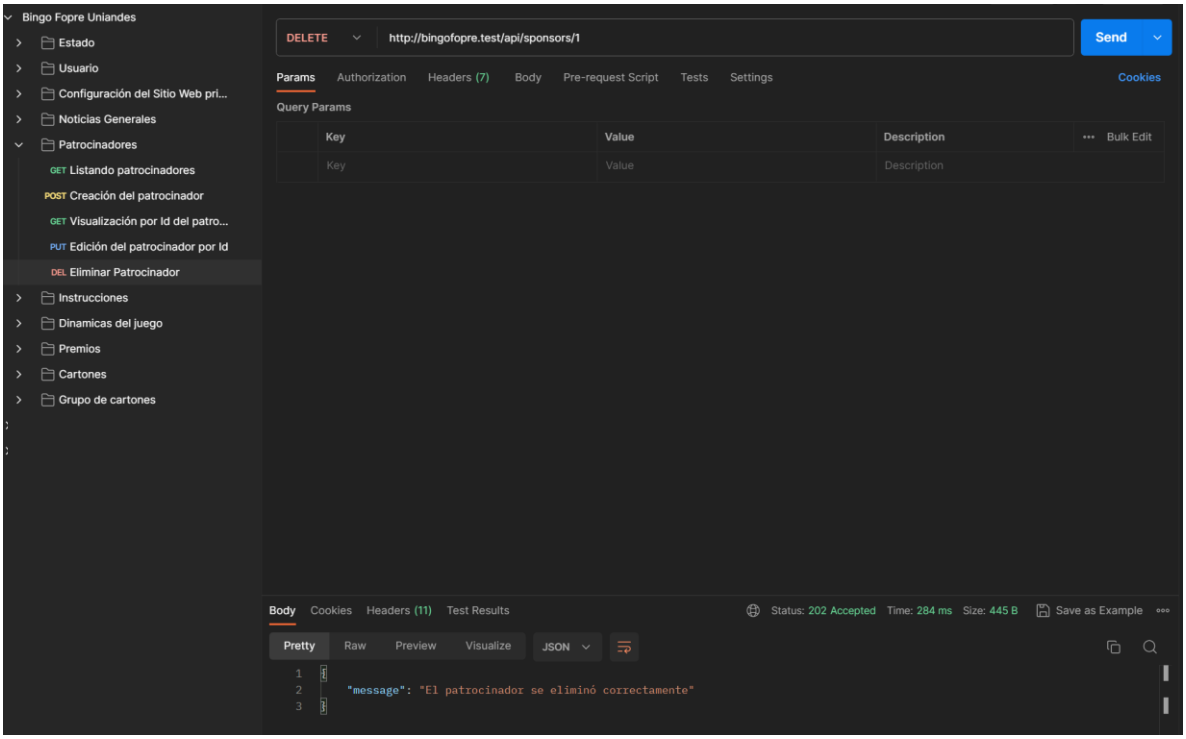
```php
/**
 * Show the form for editing the specified resource.
 */


/**
 * Update the specified resource in storage.
 */
public
function update(Request $request, string $id)
{
    try {
        $sponsor = Sponsor:: find($id)->update($request->all());
        return response()->json(['message' => 'El patrocinador se editó correctamente'], 202);
    } catch (\Throwable $th) {
        return response()->json([
            'errors' => $th
        ], 400);
    }
}
```



LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

Función para eliminar patrocinador por id

```php
/**
 * Remove the specified resource from storage.
 */
public
function destroy(string $id)
{
    try {
        $sponsor = Sponsor:: find($id)->delete();
        return response()->json(['message' => 'El patrocinador se eliminó correctamente'], 202);
    } catch (\Throwable $th) {
        return response()->json([
            'errors' => $th
        ], 400);

    }
}
```



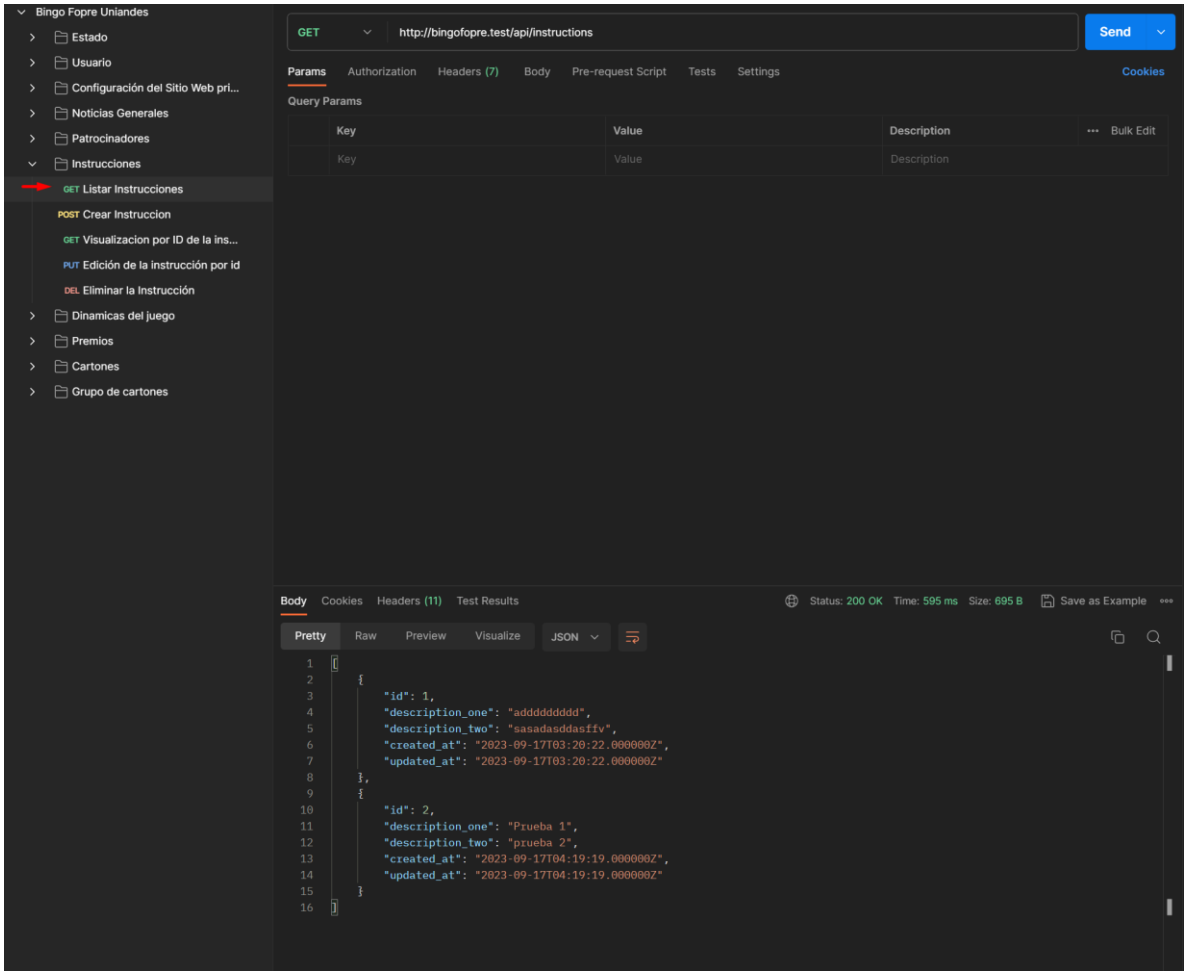
LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

INSTRUCCIONES

Ruta: /App/Http/Controllers/Api/Instructions/InstructionController.php

Función para listar instrucciones

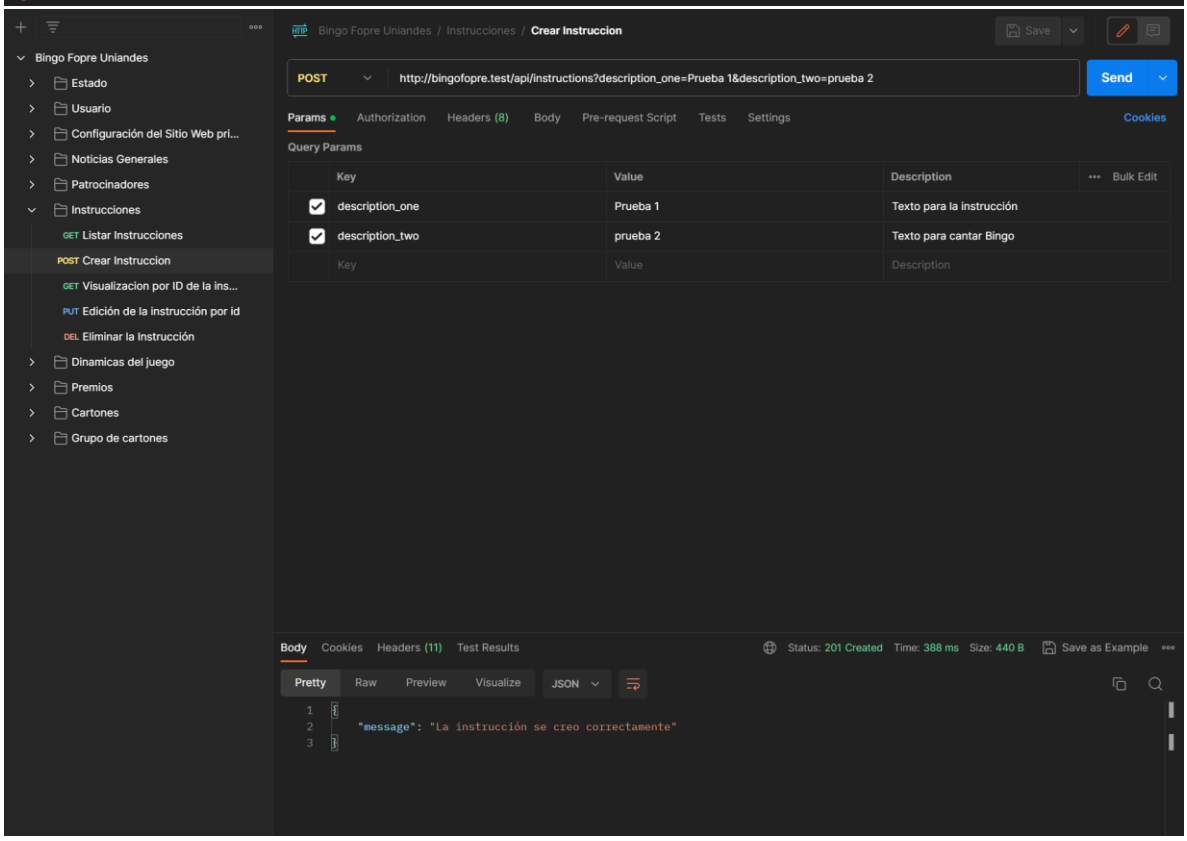
```php
/**
 * Display a listing of the resource.
 */
public function index()
{
    try {
        $instructions = Instruction::all();
        return response()->json($instructions, 200);
    } catch (\Throwable $th) {
        return response()->json([
            'errors' => $th
        ], 400);

    }

}
```



LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

Función para crear instrucciones

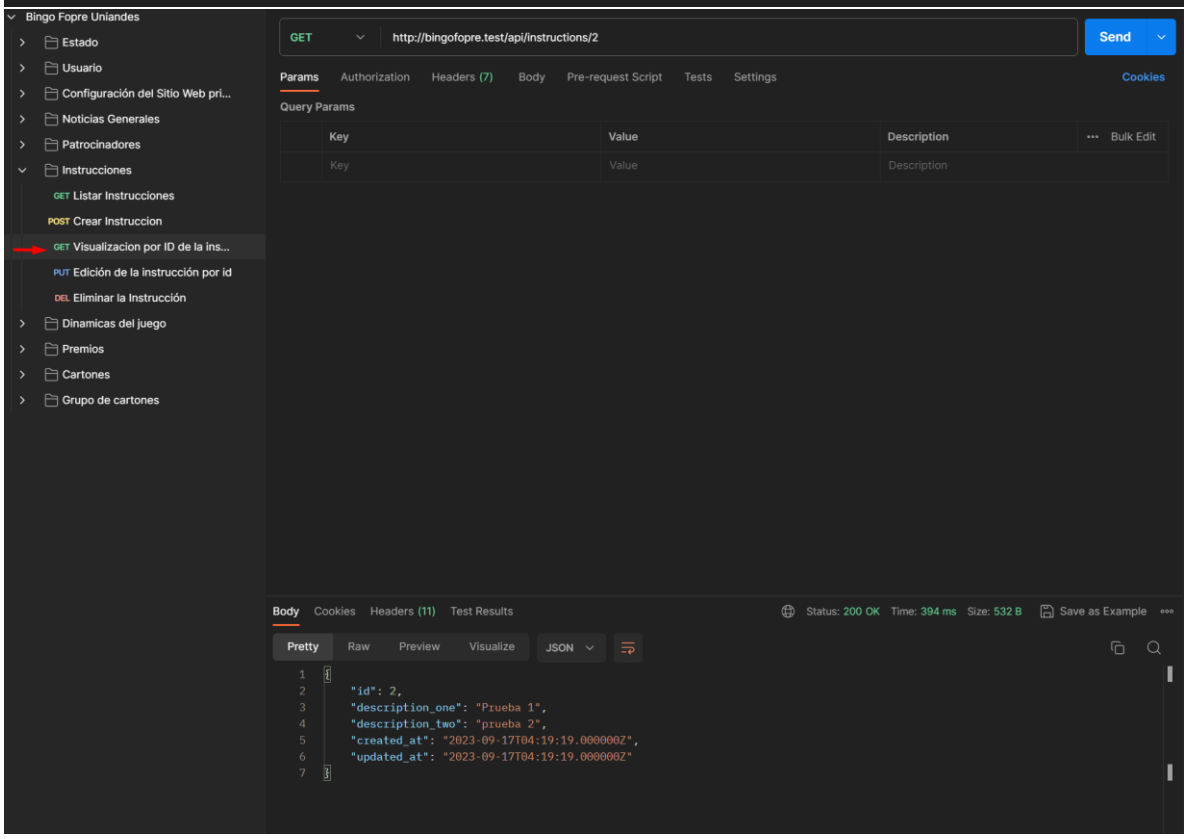
```php
/**
 * Show the form for creating a new resource.
 */


/**
 * Store a newly created resource in storage.
 */
public function store(Request $request)
{
    try {
        $instruction = Instruction:: create($request->all());
        return response()->json(['message' => 'La instrucción se creo correctamente'], 201);
    } catch (\Throwable $th) {
        return response()->json([
            'errors' => $th
        ], 400);
    }
}
```



LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

Función para visualizar instrucciones por id

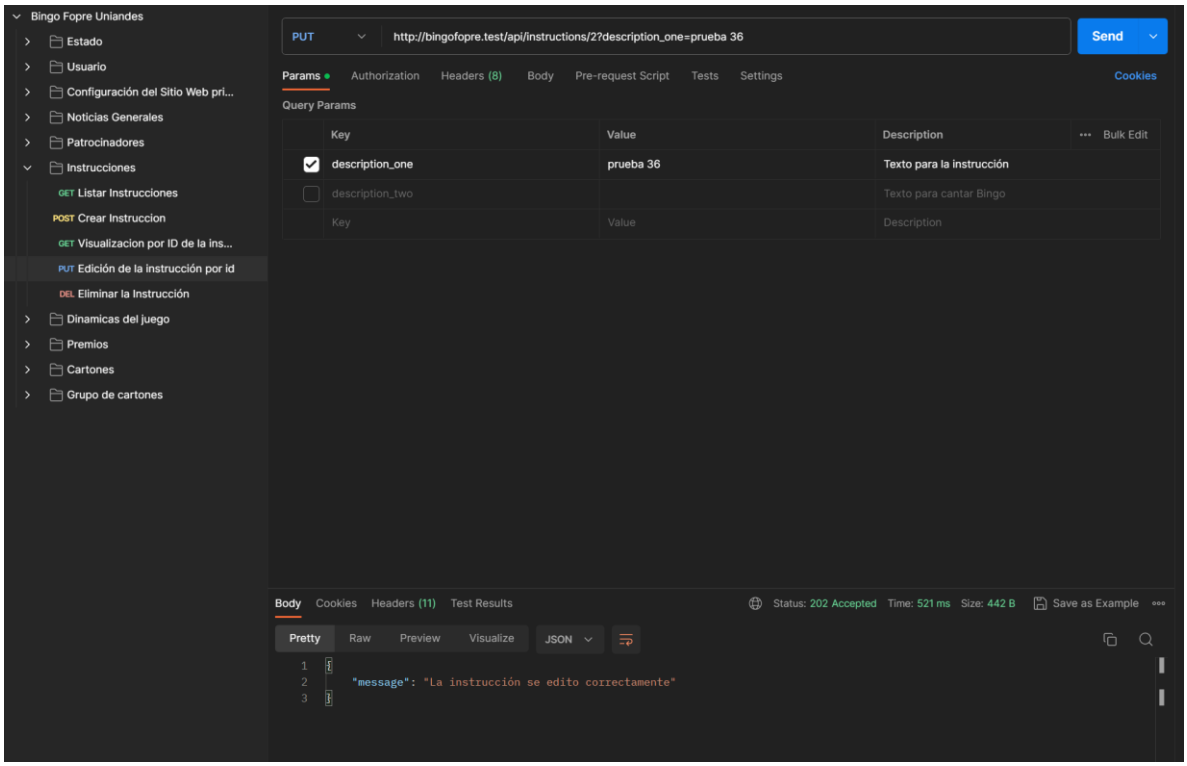
```php
/**
 * Display the specified resource.
 */
public
function show(string $id)
{
    try {
        $instruction = Instruction:: find($id);
        return response()->json($instruction, 200);
    } catch (\Throwable $th) {
        return response()->json([
            'errors' => $th
        ], 400);

    }
}
```



LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

Función para editar instrucciónes por ID

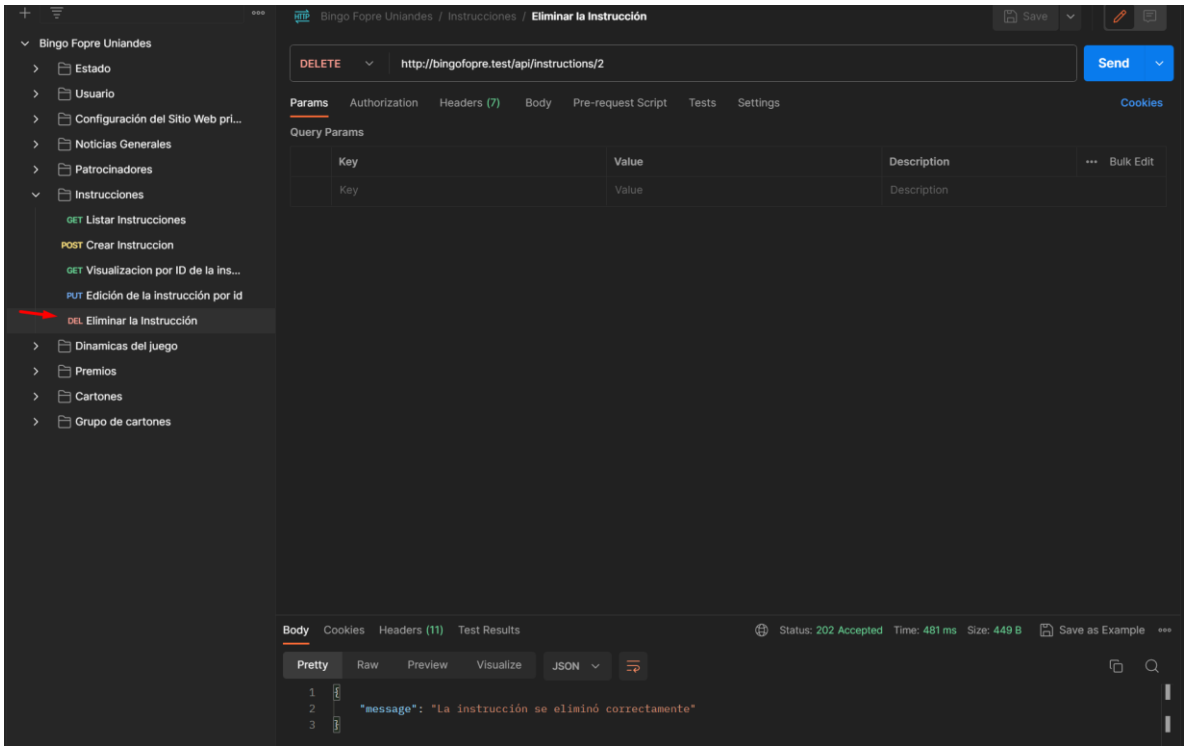
```php
/**
 * Show the form for editing the specified resource.
 */


/**
 * Update the specified resource in storage.
 */
public
function update(Request $request, string $id)
{
    try {
        $instruction = Instruction:: find($id)->update($request->all());
        return response()->json(['message' => 'La instrucción se edito correctamente'], 202);
    } catch (\Throwable $th) {
        return response()->json([
            'errors' => $th
        ], 400);
    }
}
```



LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

Función para eliminar instrucción por ID

```php
public
function destroy(string $id)
{
    try {
        $instruction = Instruction:: find($id)->delete();
        return response()->json(['message' => 'La instrucción se eliminó correctamente'], 202);
    } catch (\Throwable $th) {
        return response()->json([
            'errors' => $th
        ], 400);

    }
}
```



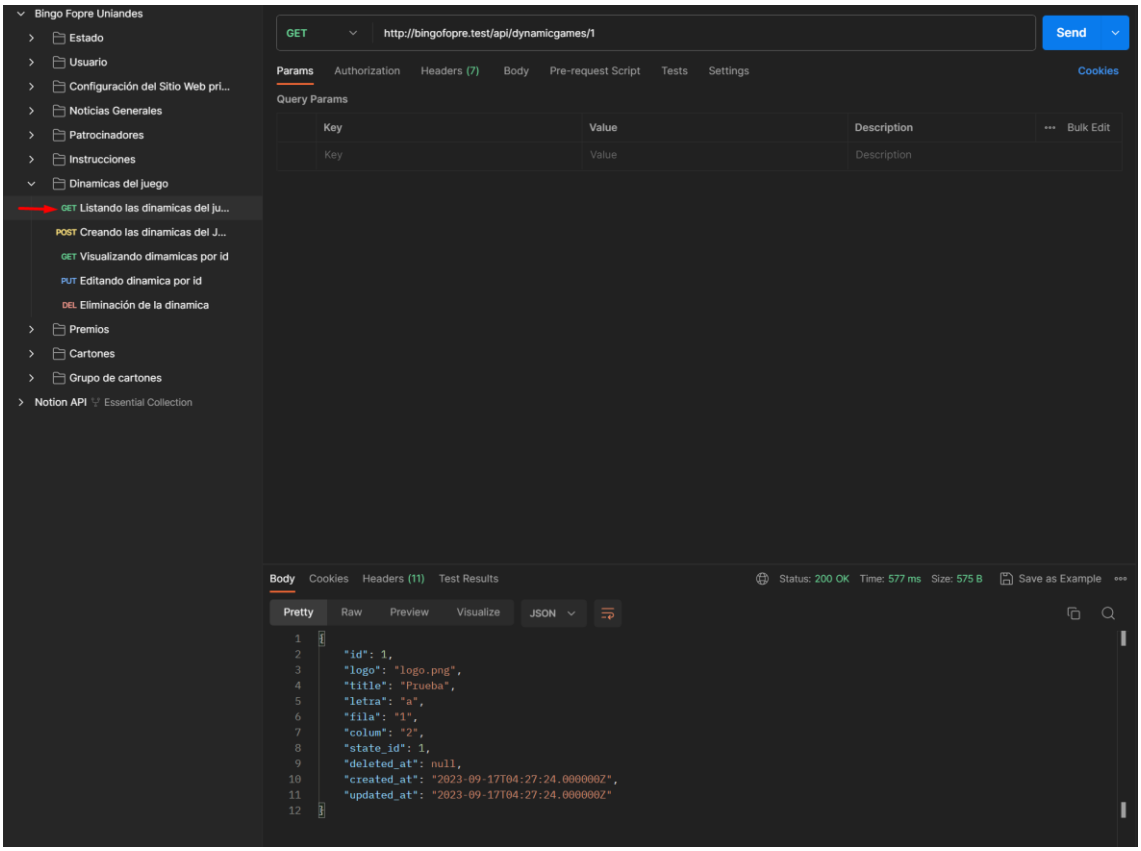
LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

DINÁMICAS DEL JUEGO

Ruta: /App/Http/Controllers/Api/DynamicGames/DynamicGameController.php

Función para listar las dinámicas del juego

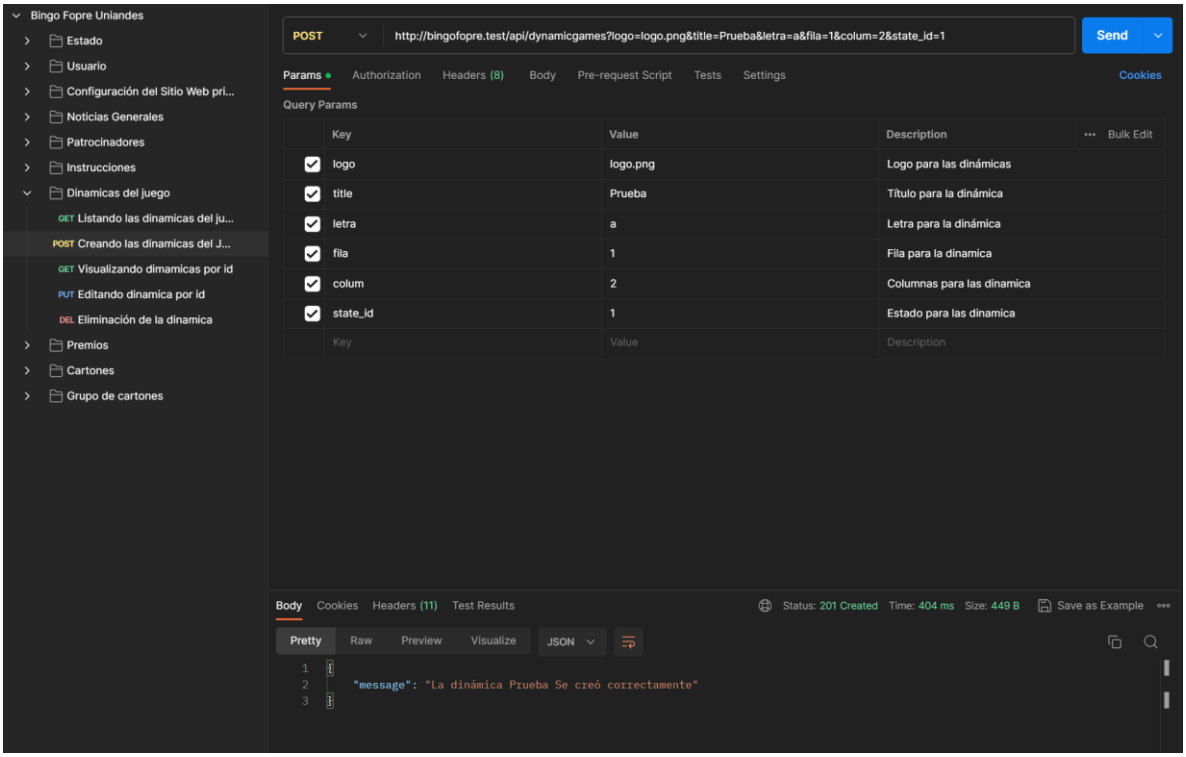
```php
/**
 * Display a listing of the resource.
 */
public function index()
{
    try {
        $dynamicgames = DynamicGame::all();
        return response()->json($dynamicgames, 200);
    } catch (\Throwable $th) {
        return response()->json([
            'errors' => $th
        ], 400);

    }

}
```



LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

Función para crear dinámicas del juego

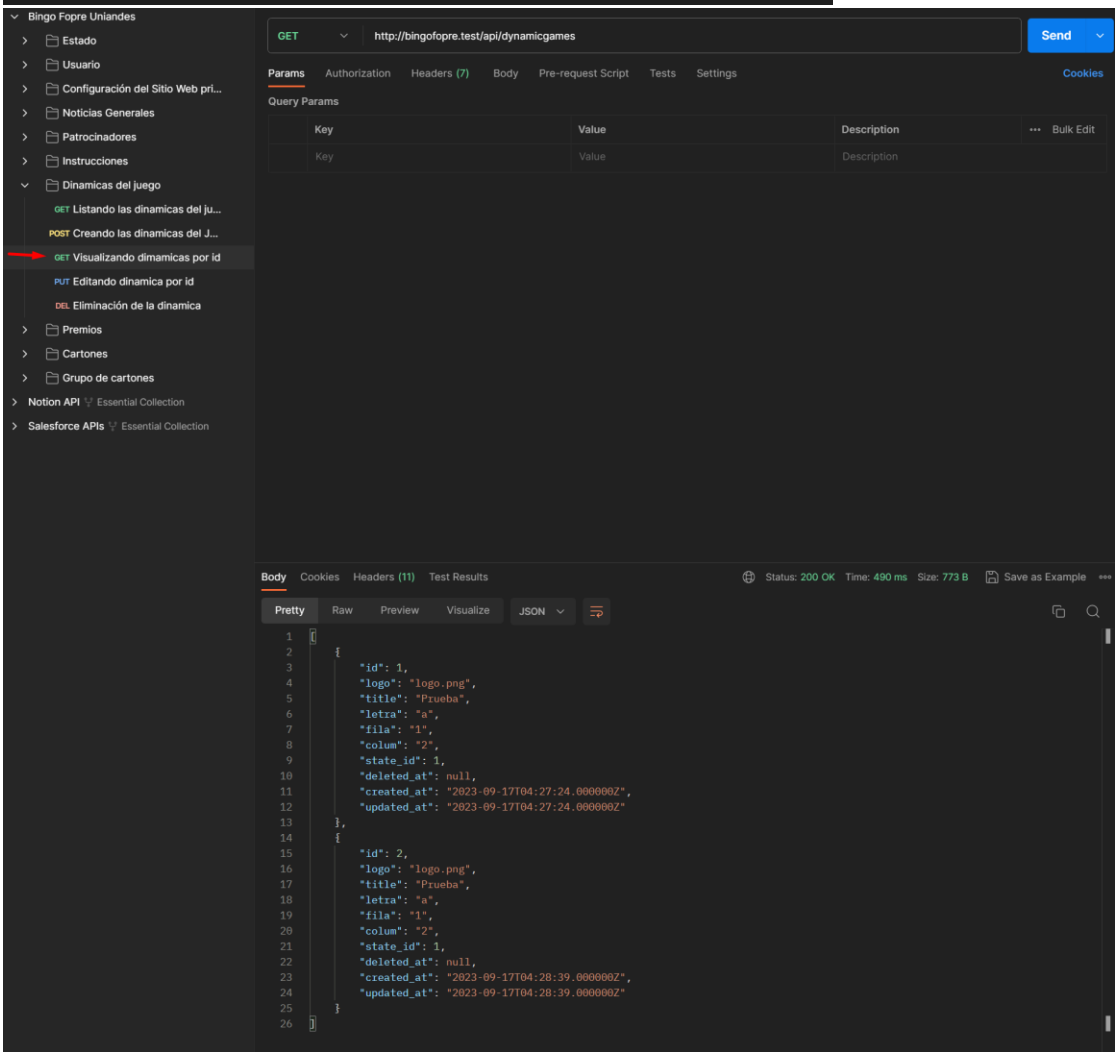
```php
/**
 * Show the form for creating a new resource.
 */


/**
 * Store a newly created resource in storage.
 */
public function store(Request $request)
{
    try {
        $dynamicgame = DynamicGame:: create($request->all());
        return response()->json(['message' => 'La dinámica ' . $dynamicgame->title . ' Se creó correctamente'], 201);
    } catch (\Throwable $th) {
        return response()->json([
            'errors' => $th
        ], 400);
    }
}
```



LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

Función para Visualizar dinámicas del juego por ID

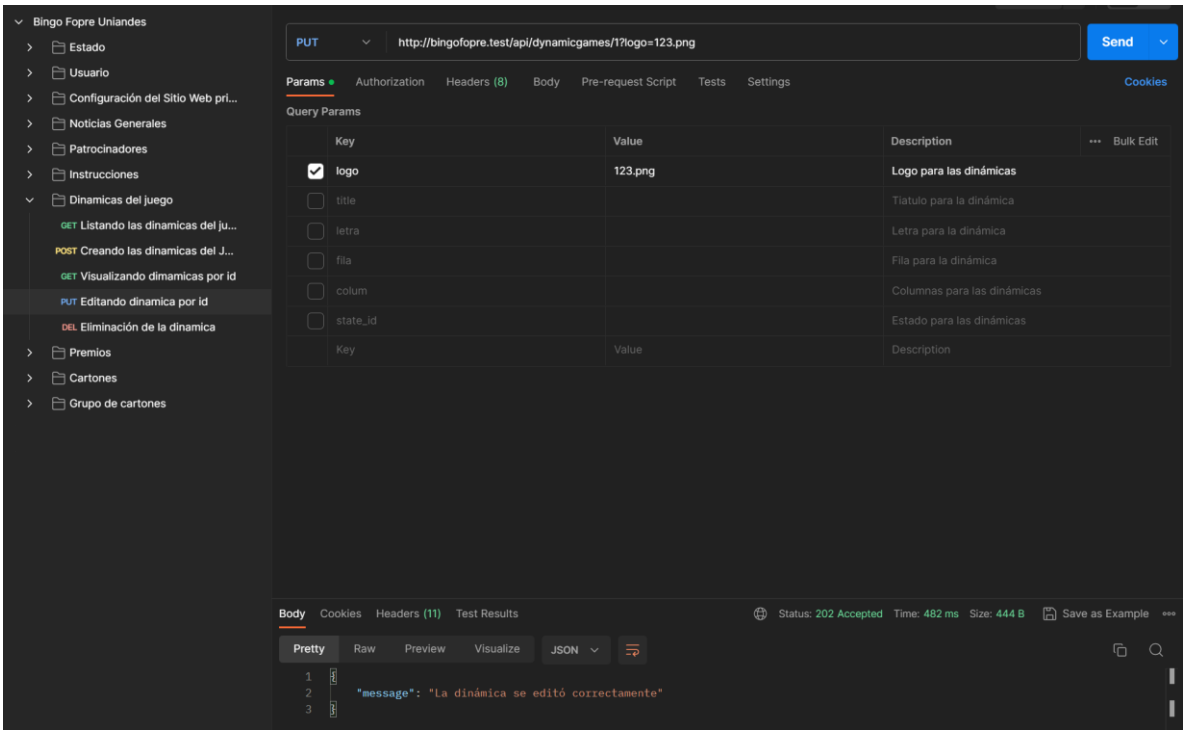
```php
/**
 * Display the specified resource.
 */
public
function show(string $id)
{
    try {
        $dynamicgame = DynamicGame:: find($id);
        return response()->json($dynamicgame, 200);
    } catch (\Throwable $th) {
        return response()->json([
                'rors' => $th
                );


    }
}
```



LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

Función para editar dinámica del juego por ID

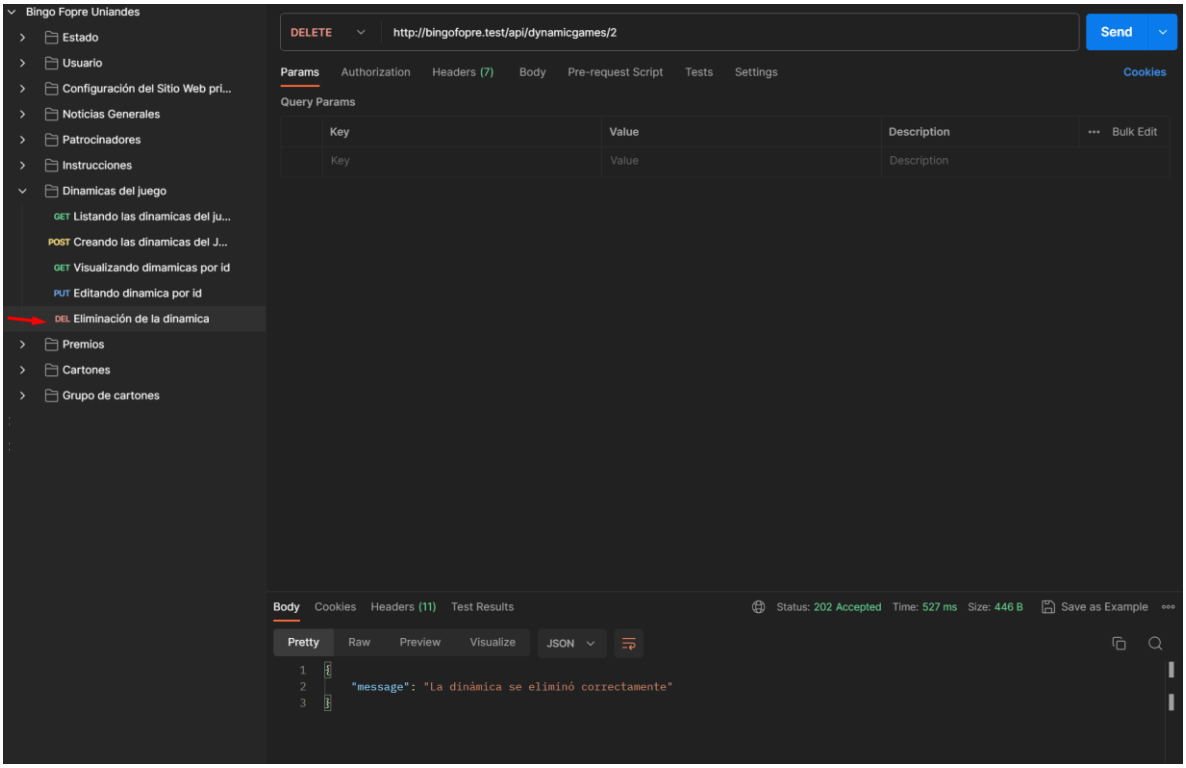
```php
/**
 * Show the form for editing the specified resource.
 */


/**
 * Update the specified resource in storage.
 */
public
function update(Request $request, string $id)
{
    try {
        $dynamicgame = DynamicGame:: find($id)->update($request->all());
        return response()->json(['message' => 'La dinámica se editó correctamente'], 202);
    } catch (\Throwable $th) {
        return response()->json([
            'errors' => $th
        ], 400);
    }
}
```



LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

Función para eliminar dinámica de juego por id.

```php
/**
 * Remove the specified resource from storage.
 */
public
function destroy(string $id)
{
    try {
        $dynamicgame = DynamicGame:: find($id)->delete();
        return response()->json(['message' => 'La dinámica se eliminó correctamente'], 202);
    } catch (\Throwable $th) {
        return response()->json([
            'errors' => $th
        ], 400);

    }
}
```



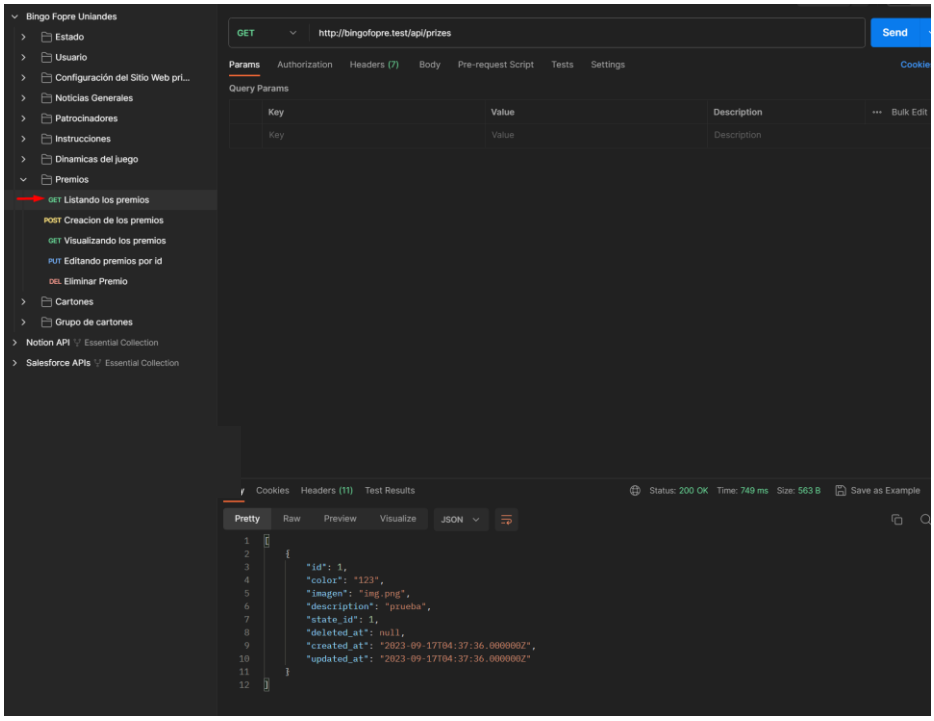
LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

PREMIOS

Ruta: /App/Http/Controllers/Api/Prizes/PrizeController.php

Función para listar de premios

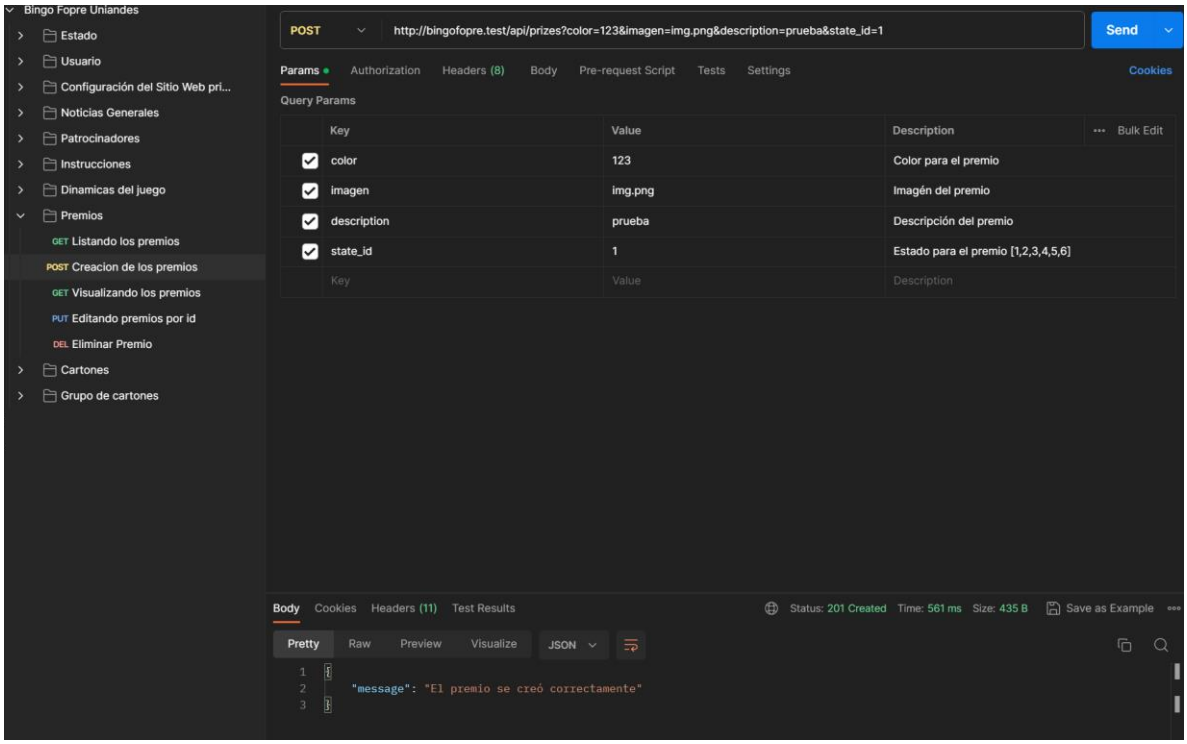
```php
/**
 * Display a listing of the resource.
 */
public function index()
{
    try {
        $prizes = Prize::all();
        return response()->json($prizes, 200);
    } catch (\Throwable $th) {
        return response()->json([
            'errors' => $th
        ], 400);

    }

}
```



LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

Función para crear premios

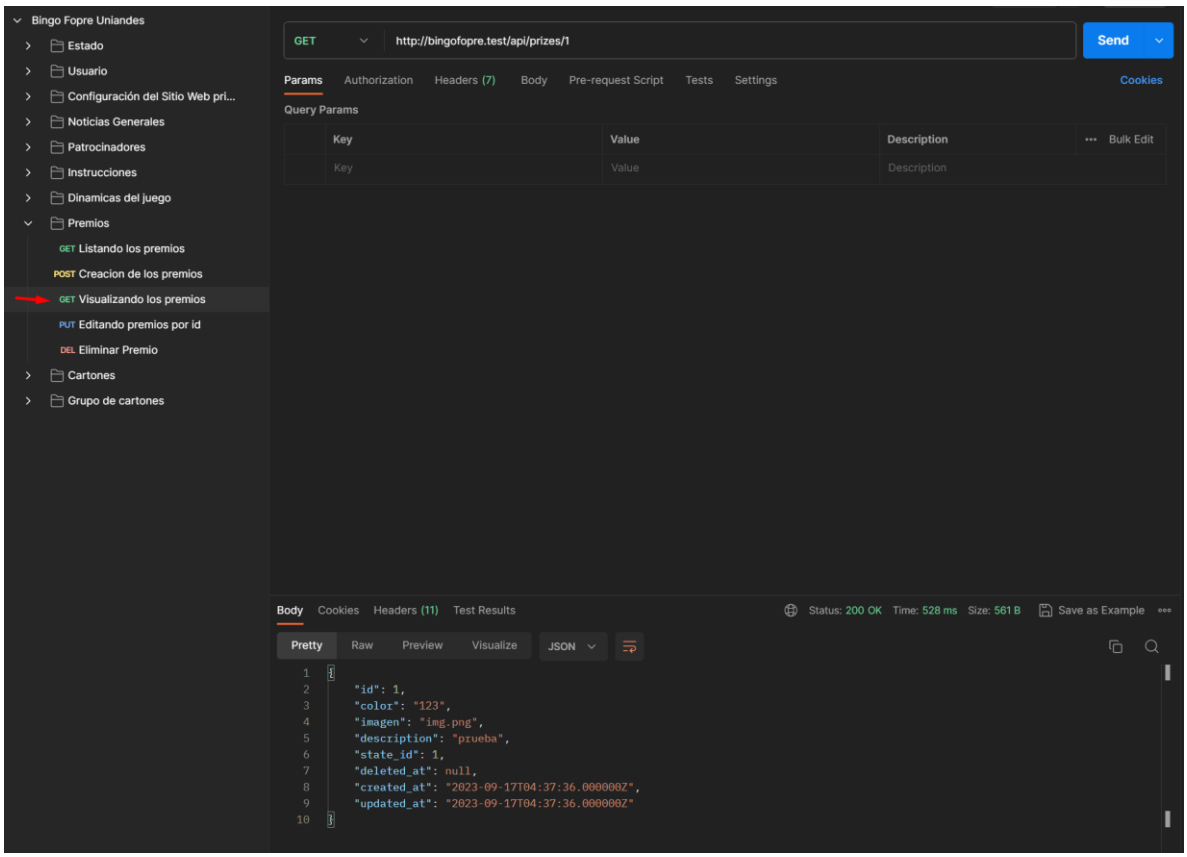
```php
/**
 * Show the form for creating a new resource.
 */


/**
 * Store a newly created resource in storage.
 */
public function store(Request $request)
{
    try {
        $prize = Prize:: create($request->all());
        return response()->json(['message' => 'El premio se creó correctamente'], 201);
    } catch (\Throwable $th) {
        return response()->json([
            'errors' => $th
        ], 400);
    }
}
```



LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

Función para visualizar premios por ID

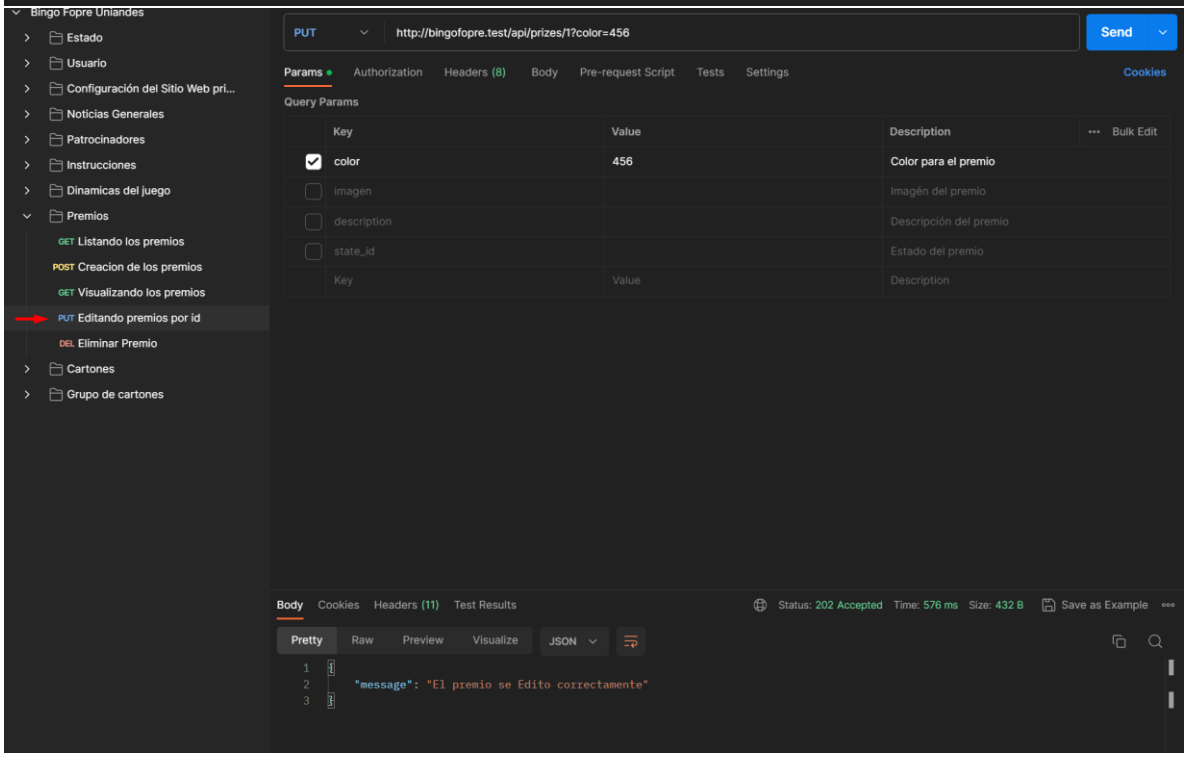
```php
/**
 * Display the specified resource.
 */
public
function show(string $id)
{
    try {
        $prize = Prize:: find($id);
        return response()->json($prize, 200);
    } catch (\Throwable $th) {
        return response()->json([
            'errors' => $th
        ], 400);

    }
}
```



LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

Función para editar premios

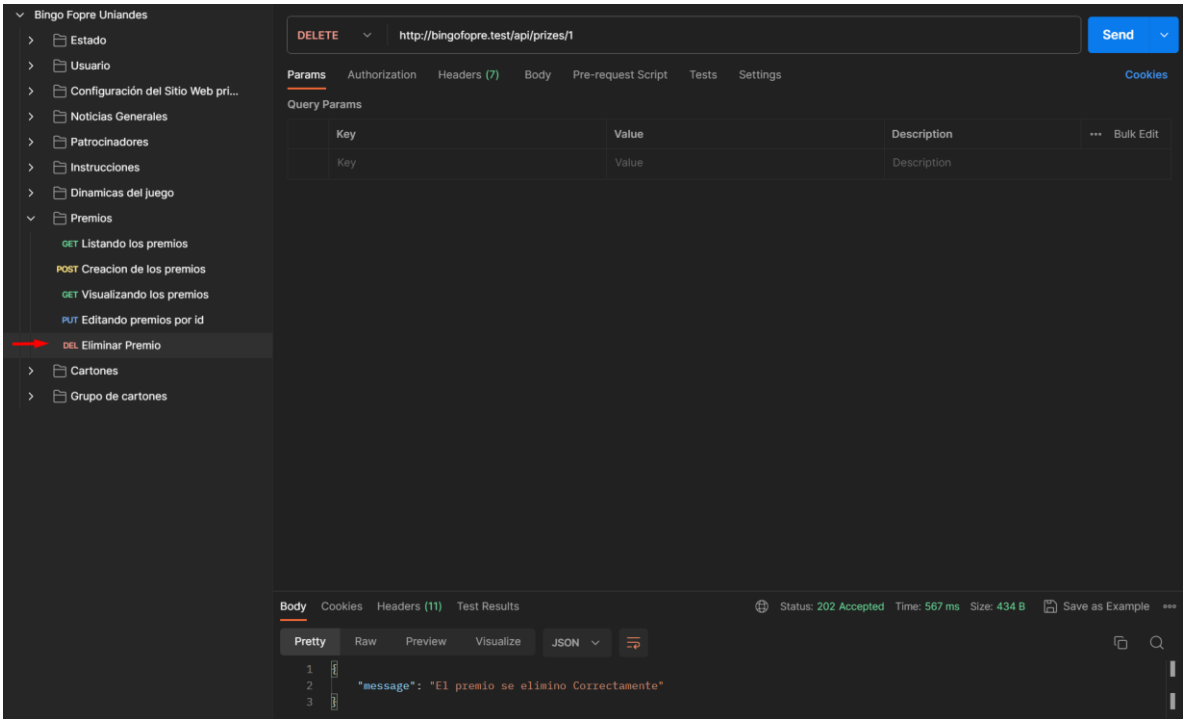
```php
/**
 * Show the form for editing the specified resource.
 */


/**
 * Update the specified resource in storage.
 */
public
function update(Request $request, string $id)
{
    try {
        $prize = Prize:: find($id)->update($request->all());
        return response()->json(['message' => 'El premio se Edito correctamente'], 202);
    } catch (\Throwable $th) {
        return response()->json([
            'errors' => $th
        ], 400);
    }
}
```



LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

Función para eliminar de premio por id

```php
/**
 * Remove the specified resource from storage.
 */
public
function destroy(string $id)
{
    try {
        $prize = Prize:: find($id)->delete();
        return response()->json(['message' => 'El premio se elimino Correctamente'], 202);
    } catch (\Throwable $th) {
        return response()->json([
            'errors' => $th
        ], 400);

    }
}
```



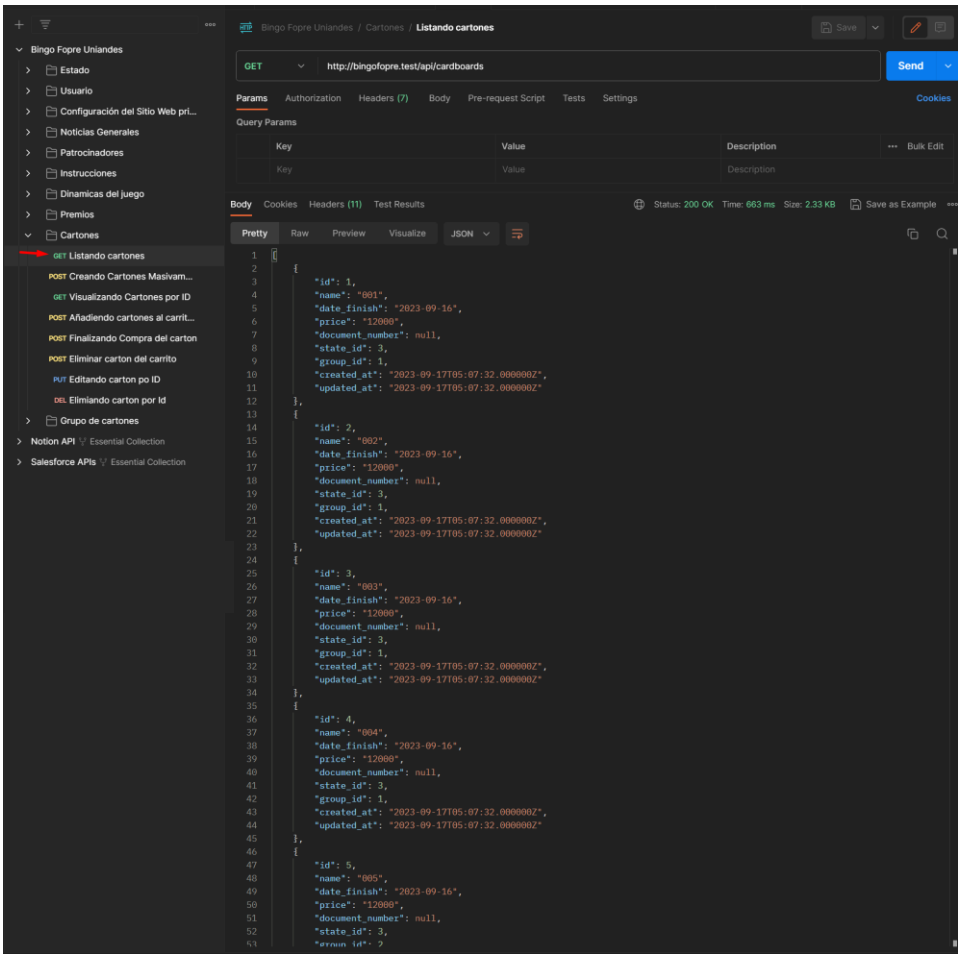
LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

CARTONES

Ruta: /App/Http/Controllers/Api/Cartons/CartonController.php

Función para listar todos los cartones

```php
/**
 * Display a listing of the resource.
 */
public function index()
{
    try {
        $cardboards = Cardboard::all();
        return response()->json($cardboards, 200);
    } catch (\Throwable $th) {
        return response()->json([
            'errors' => $th
        ], 400);

    }

}
```

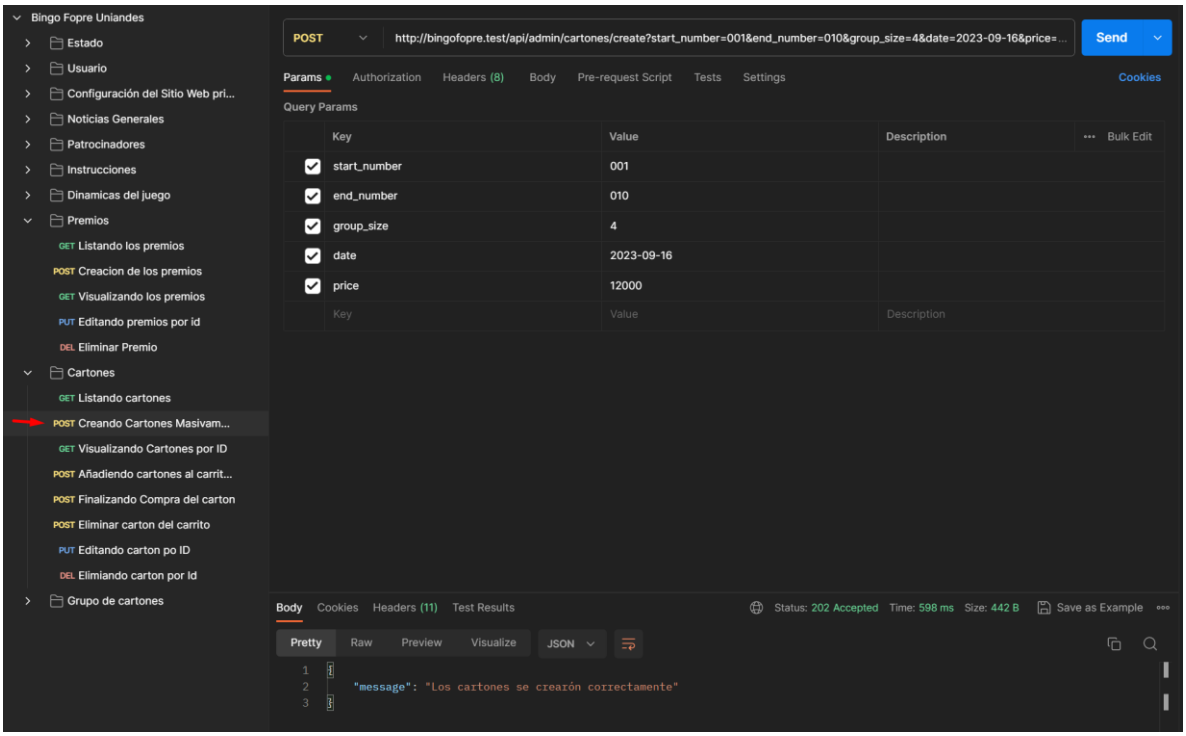


LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

Función para crear cartones masivamente

```php
public function create(Request $request)
{
    try {
        $startNumber = strval($request->input('start_number')); // Convierte a cadena
        $endNumber = strval($request->input('end_number')); // Convierte a cadena
        $groupSize = $request->input('group_size');
        $date = $request->input('date');
        $price = $request->input('price');

        //$user_id = auth()->user()->id; // Obtener el ID del usuario autenticado

        $group = null;
        $groupCount = 0;

        for ($i = $startNumber; $i <= $endNumber; $i++) {
            if ($groupCount % $groupSize === 0) {
                $group = CartonGroup::create(['user_id' => null]);
            }

            // Formatea el nombre del cartón con ceros a la izquierda
            $formattedName = str_pad($i, strlen($endNumber), '0', STR_PAD_LEFT);

            $cardboard = Cardboard::create([
                'name' => $formattedName,
                'date_finish' => $date,
                'price' => $price,
                'state_id' => 3, // Reemplaza con el estado correcto
                'group_id' => $group ? $group->id : null,
            ]);

            $groupCount++;
            // Asigna el group_id al cartón
            $cardboard->group_id = $group ? $group->id : null;
            $cardboard->save();
        }
        return response()->json(['message' => 'Los cartones se crearón correctamente'], 202);

    } catch (ValidationException $e) {
        // Manejo de excepciones de validación
        return response()->json(['error' => 'Error de validación'], 400);
    } catch (QueryException $e) {
        // Manejo de excepciones de consulta de base de datos
        return response()->json(['error' => 'Error de base de datos'], 500);
    } catch (Exception $e) {
        // Manejo de otras excepciones no previstas
        return response()->json(['error' => 'Error interno del servidor'], 500);
    }
```

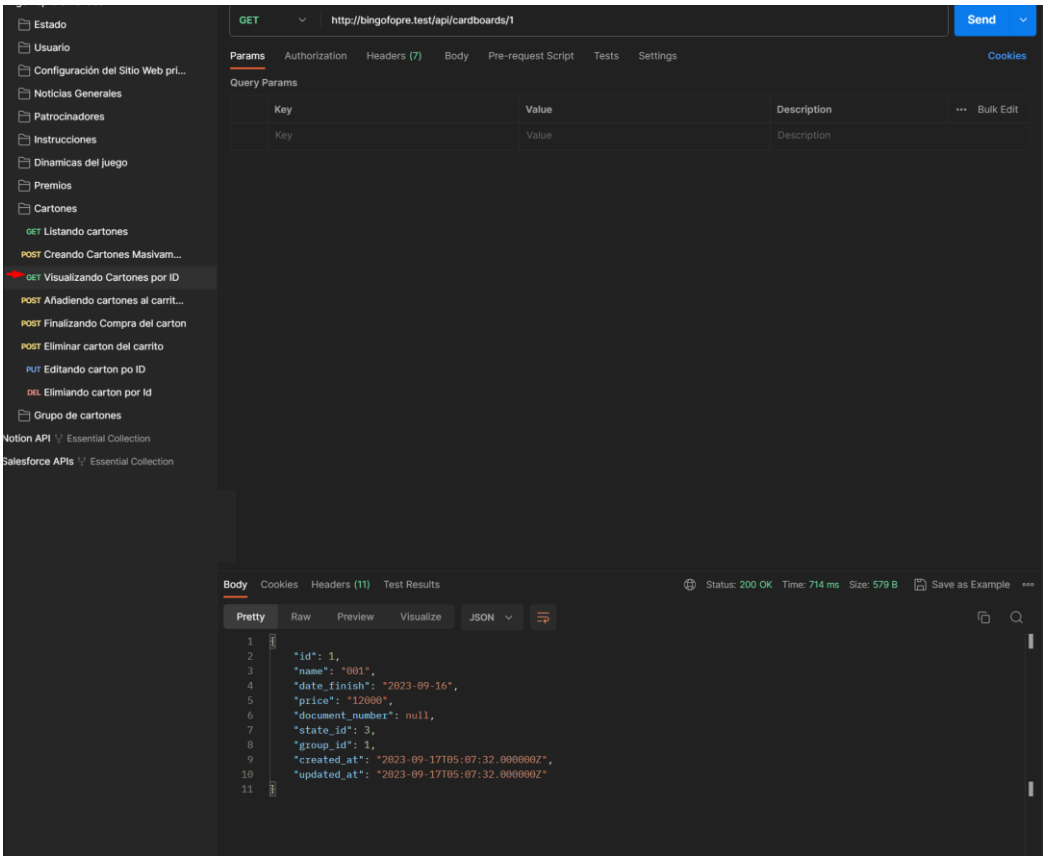
LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ



LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

Función para visualizar cartones por ID

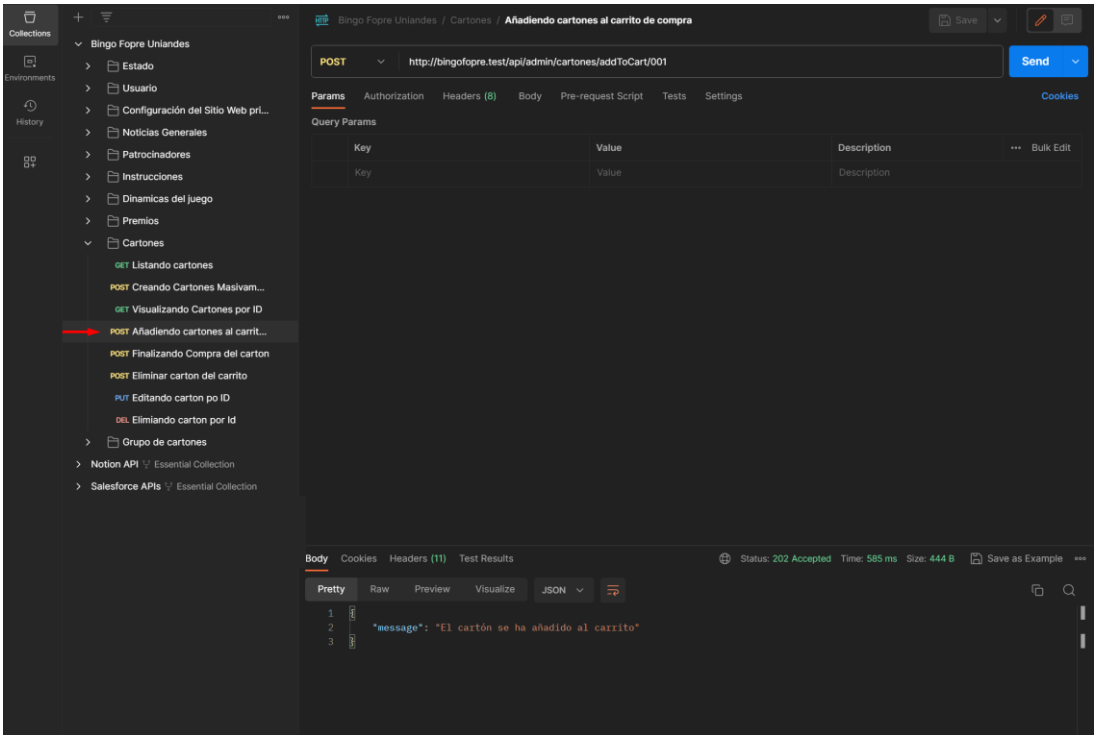
```php
/**
 * Display the specified resource.
 */
public
function show(string $id)
{
    try {
        $cardboard = Cardboard:: find($id);
        return response()->json($cardboard, 200);
    } catch (\Throwable $th) {
        return response()->json([
            'errors' => $th
        ], 400);

    }
}
```



LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

Función para añadir un carton al carrito

```php
public function addToCart($name)
{
    try {
        // Buscar el cartón por el nombre en lugar del ID
        $carton = Cardboard::where('name', $name)->first();

        // Verificar si se encontró el cartón
        if (!$carton) {
            return response()->json(['error' => 'Ocurrió un error inesperado'], 500);
        }

        $cart = session()->get('cart');

        $cart[$carton->id] = [
            'name' => $carton->name,
            'quantity' => 1,
            'date_finish' => $carton->date_finish,
            'price' => $carton->price,
            'state_id' => $carton->state_id,
            'user_id' => $carton->user_id,
            'document_number' => $carton->document_number,
        ];

        session()->put('cart', $cart);
        return response()->json(['message' => 'El cartón se ha añadido al carrito'], 202);

    } catch (Exception $e) {
        // Manejo de excepciones
        return response()->json(['error' => 'Ocurrió un error inesperado'], 500);
    }
}
```

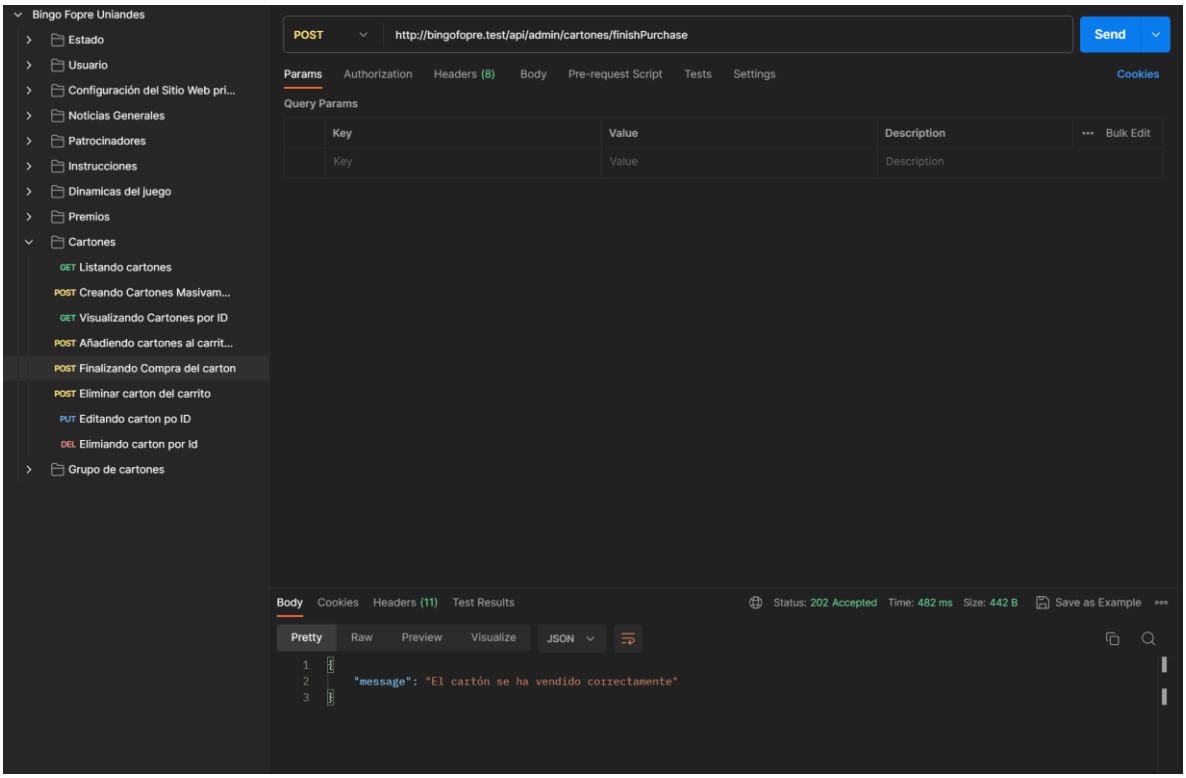


LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

Función para finalizer compra del carton

```php
public function finishPurchase(Request $request)
{
    try {
        // Obtener el carrito desde la sesión
        $cart = Session::get('cart', []);

        // Obtener datos de estado y usuario de cada cartón desde el formulario
        $cartonData = $request->input('cartons');
        $documentoComprador = $request->input('document_number');

        // Iterar a través de los elementos del carrito y actualizar la base de datos
        foreach ($cart as $cartonId => $carton) {
            // Verificar si el cartón existe en la base de datos
            $cartonDB = Cardboard::find($cartonId);

            if ($cartonDB) {
                // Actualiza el estado y el campo user_id del cartón en la base de datos
                $cartonDB->state_id = $cartonData[$cartonId]['state_id'];
                $cartonDB->document_number = $documentoComprador;
                $cartonDB->save();
            }
        }

        // Limpia el carrito en la sesión
        Session::forget('cart');

        return response()->json(['message' => 'El cartón se ha vendido correctamente'], 202);
    } catch (ValidationException $e) {
        // Manejo de excepciones de validación
        return response()->json(['error' => 'Error de validación'], 400);
    } catch (QueryException $e) {
        // Manejo de excepciones de consulta de base de datos
        return response()->json(['error' => 'Error de base de datos'], 500);
    } catch (Exception $e) {
        // Manejo de otras excepciones no previstas
        return response()->json(['error' => 'Ocurrió un error inesperado'], 500);
    }
}
```

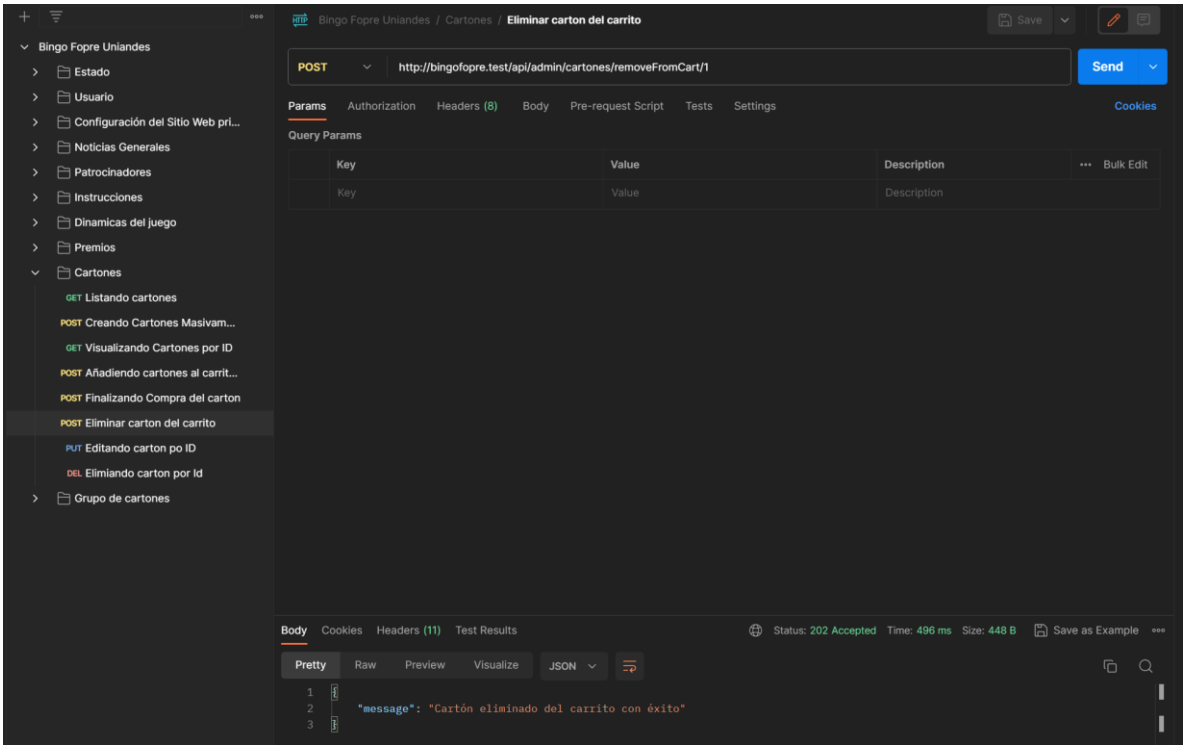
LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ



LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

Función para eliminar un carton en el carrito

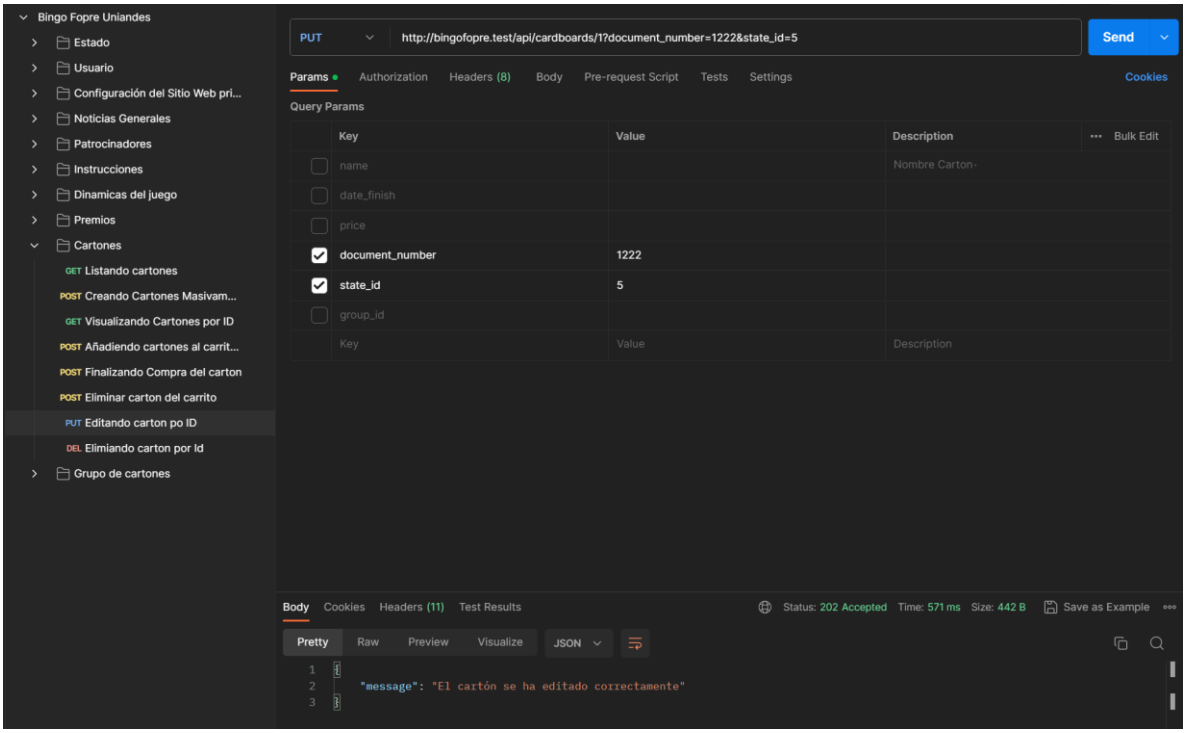
```php
public function removeFromCart($cartonId)
{
    try {
        // Obtener el carrito desde la sesión
        $cart = Session::get('cart', []);

        // Verificar si el cartón existe en el carrito
        if (isset($cart[$cartonId])) {
            // Elimina el cartón del carrito
            unset($cart[$cartonId]);

            // Actualiza el carrito en la sesión
            Session::put('cart', $cart);

            return redirect()->route('user.cart.index')->with('success', 'Cartón eliminado del carrito con éxito.');
        }

        return response()->json(['message' => 'Cartón eliminado del carrito con éxito'], 202);


    } catch (Exception $e) {
        // Manejo de excepciones
        return response()->json(['error' => 'Ocurrió un error inesperado.'], 400);

    }
}
```



LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

Función para editar carton por ID

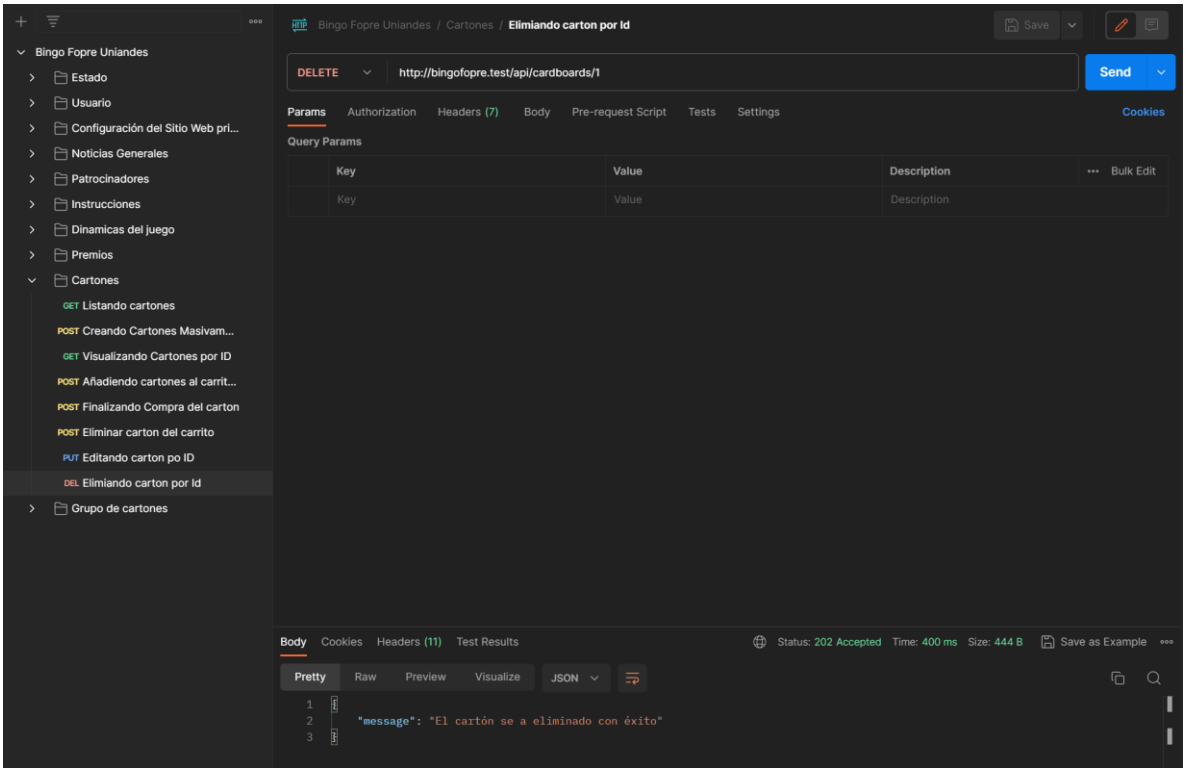
```php
/**
 * Show the form for editing the specified resource.
 */


/**
 * Update the specified resource in storage.
 */
public
function update(Request $request, string $id)
{
    try {
        $cardboard = Cardboard:: find($id)->update($request->all());
        return response()->json(['message' => 'El cartón se ha editado correctamente'], 202);
    } catch (\Throwable $th) {
        return response()->json([
            'errors' => $th
        ], 400);
    }
}
```



LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

Función para eliminar carton por id

```php
/**
 * Remove the specified resource from storage.
 */
public
function destroy(string $id)
{
    try {
        $cardboard = Cardboard:: find($id)->delete();
        return response()->json(['message' => 'El cartón se a eliminado con éxito'], 202);
    } catch (\Throwable $th) {
        return response()->json([
            'errors' => $th
        ], 400);

    }
}
```



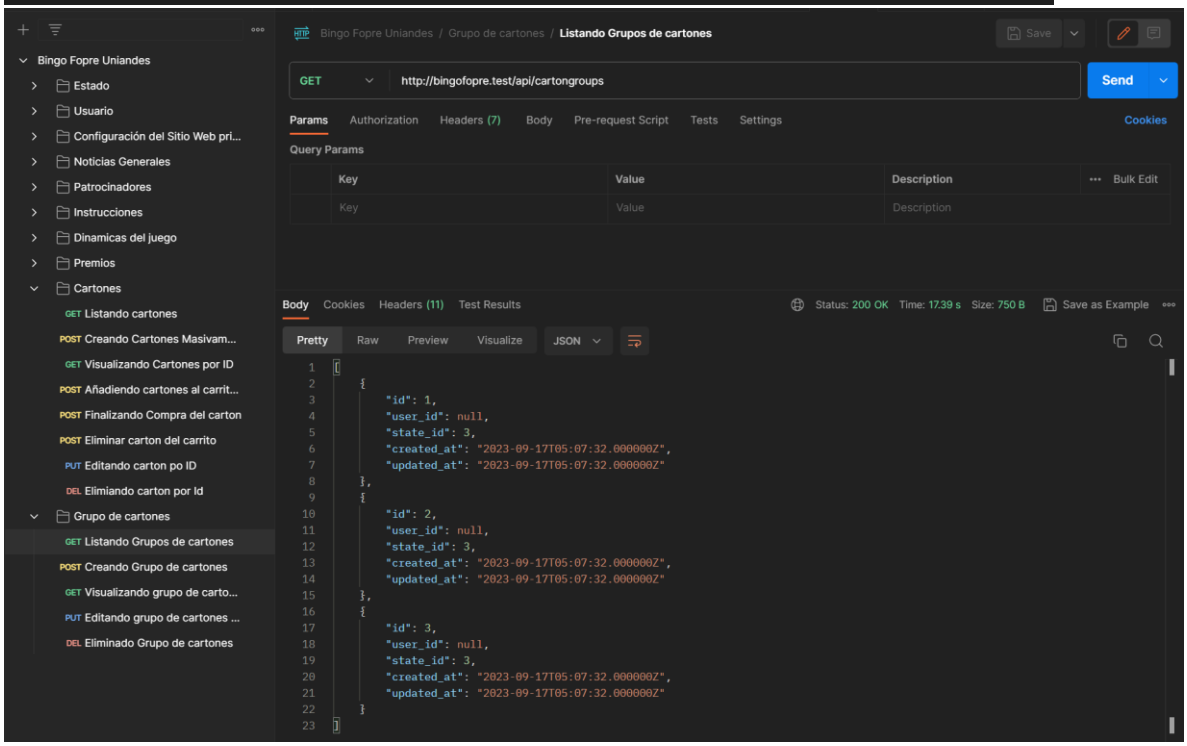
LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

GRUPO DE CARTONES

Ruta: /App/Http/Controllers/Api/CartonGroups/CartonGroupsController.php

Función para listar grupo de cartones

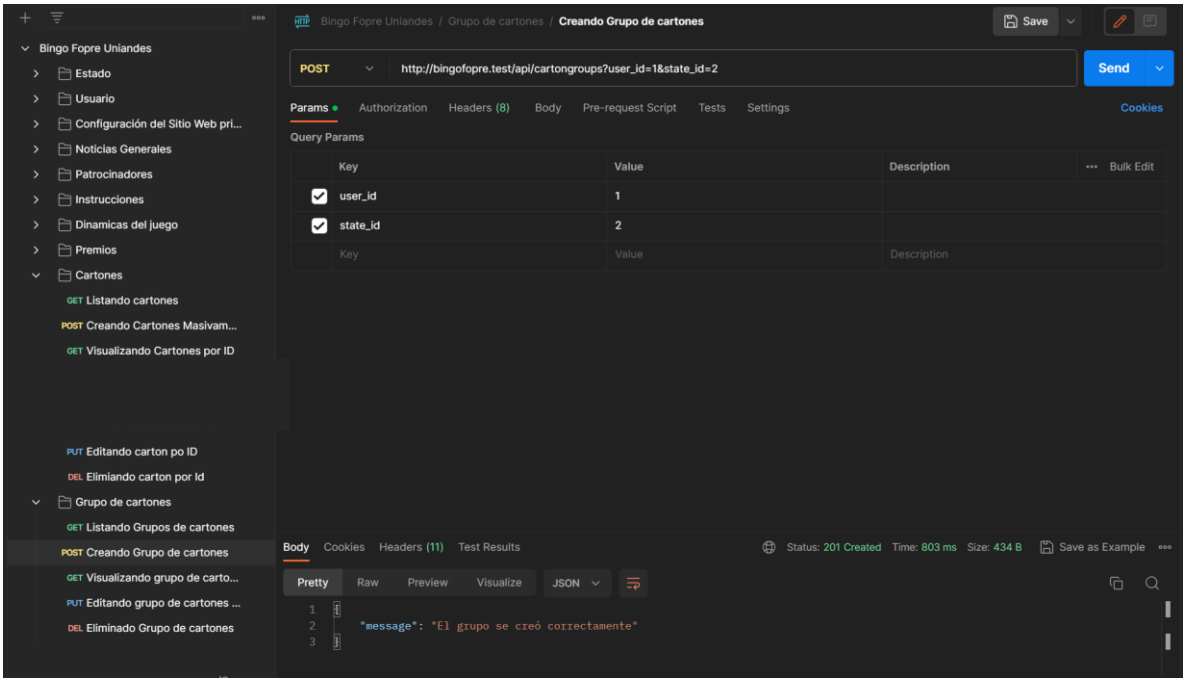
```php
/**
 * Display a listing of the resource.
 */
public function index()
{
    try {
        $cartonGroups = CartonGroup::all();
        return response()->json($cartonGroups, 200);
    } catch (\Throwable $th) {
        return response()->json([
            'errors' => $th
        ], 400);

    }

}
```



LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

Función para crear un grupo carton

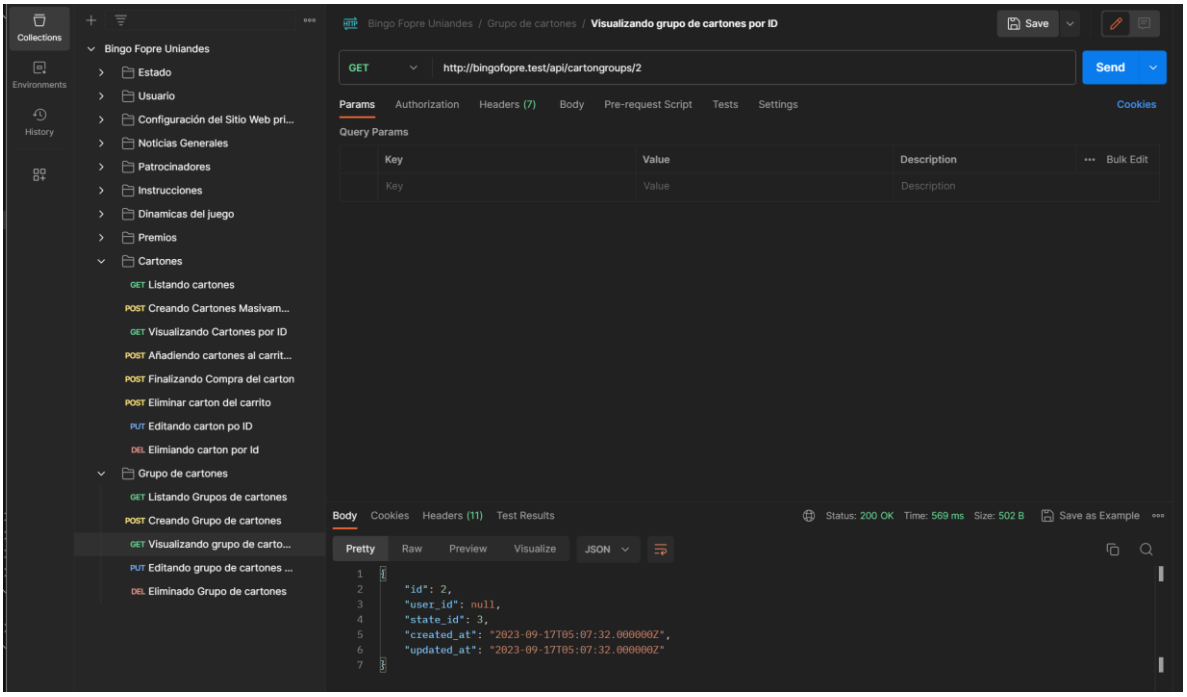
```php
/**
 * Show the form for creating a new resource.
 */


/**
 * Store a newly created resource in storage.
 */
public function store(Request $request)
{
    try {
        $cartonGroup = CartonGroup:: create($request->all());
        return response()->json(['message' => 'El grupo se creó correctamente'],
    } catch (\Throwable $th) {
        return response()->json([
            'errors' => $th
        ], 400);
    }
}
```



LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

Función para visualizar un grupo de carton por id

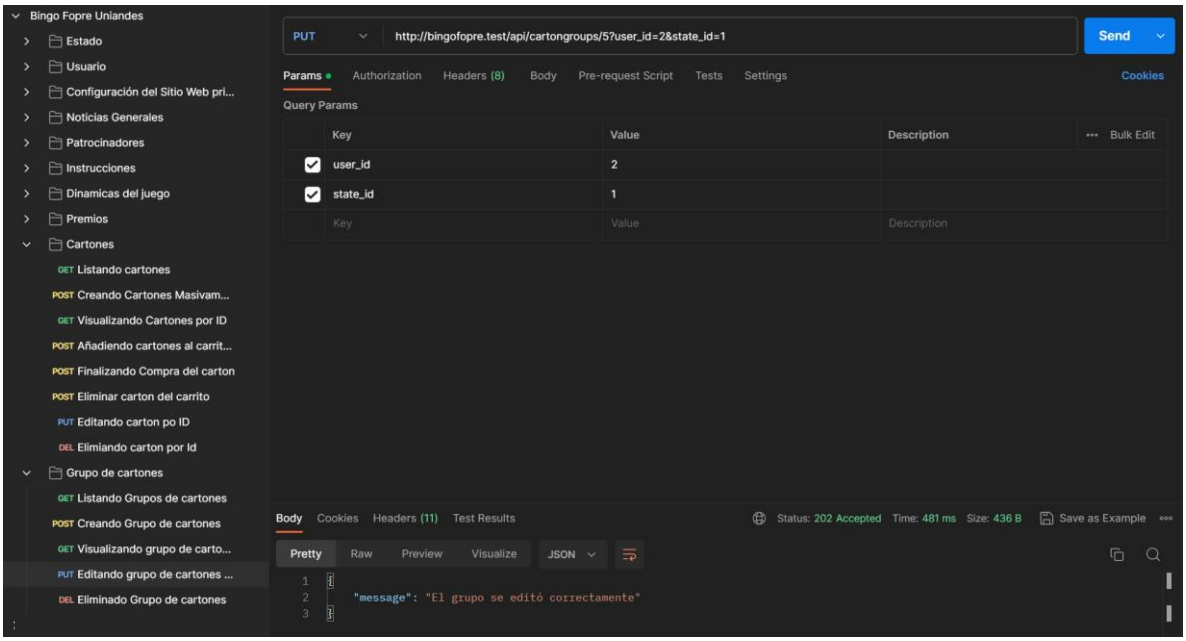
```php
/**
 * Display the specified resource.
 */
public
function show(string $id)
{
    try {
        $cartonGroup = CartonGroup:: find($id);
        return response()->json($cartonGroup, 200);
    } catch (\Throwable $th) {
        return response()->json([
            'errors' => $th
        ], 400);

    }
}
```



LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

Función para editar grupo de cartones

```php
/**
 * Show the form for editing the specified resource.
 */


/**
 * Update the specified resource in storage.
 */
public
function update(Request $request, string $id)
{
    try {
        $cartonGroup = CartonGroup:: find($id)->update($request->all());
        return response()->json(['message' => 'El grupo se editó correctamente'],
    } catch (\Throwable $th) {
        return response()->json([
            'errors' => $th
        ], 400);
    }
}
```



LCKM INNOVATY
JUAN CAMILO RODRIGUEZ RAMIREZ

Función para eliminar grupo de cartones

```php
/**
 * Remove the specified resource from storage.
 */
public
function destroy(string $id)
{
    try {
        $cartonGroup = CartonGroup:: find($id)->delete();
        return response()->json(['message' => 'El grupo se eliminó correctamente'], 202);
    } catch (\Throwable $th) {
        return response()->json([
            'errors' => $th
        ], 400);

    }
}
```