

Evidencia día 4 Semana 14

- Termino de modelo entidad relación de la aplicación.
- Empiezo el desarrollo de la API para la conexión de APP con la base de datos creada.
- Se crea Modulo Usuarios, para la creación del usuario en app y pueda ser controlado en BackEnd que estoy desarrollando en conjunto con la API. Adjunto imagen de Código fuente de API desarrollada.

```
* @return \Illuminate\Http\Response
*/
public function index()
{
    //
    return Clientes::all();
}

/**
 * Store a newly created resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @return \Illuminate\Http\Response
 */
public function store(Request $request)
{
    //$request->validated();

    //$user = Auth::user();
    //echo $request->input('comuna_id'); exit;
    $cliente = new Clientes();
    $cliente->Cliente_nombres = $request->input('nombres');
    $cliente->Cliente_apepat = $request->input('apepat');
    $cliente->Cliente_apemat = $request->input('apemat');
    $cliente->Cliente_run = $request->input('run');
    $cliente->Cliente_direccion = $request->input('direccion');
    $cliente->Cliente_telefono = $request->input('telefono');
    $cliente->Cliente_mail = $request->input('mail');
    $cliente->Cliente_password = $request->input('password');
    $cliente->comuna_id = $request->input('comuna_id');
    $fecha_nac = str_replace("/", "-", $request->input('f_nac'));
    $cliente->Cliente_fechanac = date('Y-m-d', strtotime($fecha_nac));
    //echo "<pre>"; print_r($cliente); echo "</pre>"; exit;
    $res = $cliente->save();

    if($res){
        return response()->json(['message' => 'Cliente creado'], 201);
    }
    return response()->json(['message' => 'Cliente no creado'], 500);
}

/**
```

```

namespace App\Http\Requests\V1;

use Illuminate\Foundation\Http\FormRequest;
use Illuminate\Support\Facades\Auth;

class ClienteRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return Auth::check();
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            'nombre' => 'required|max:80',
            'apepat' => 'required|max:80',
            'apemat' => 'required|max:80',
            'run' => 'required|max:15',
            'f_nac' => 'max:30',
            'comuna_id' => 'required|numeric|max:11',
            'direccion' => 'required|max:100',
            'mail' => 'required|max:80',
            'password' => 'required|max:50',
            'telefono' => 'max:30'
        ];
    }
}

```

POST 190.113.12.97/gat... X + ... No Environme

GatoApi / 190.113.12.97/gatobackend/public/api/v1/Cientes Save

POST 190.113.12.97/gatobackend/public/api/v1/Cientes

Params Authorization Headers (7) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL

	KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/>	nombres	ClaudioClaudio	
<input checked="" type="checkbox"/>	aopat	Blanco	
<input checked="" type="checkbox"/>	apemat	Vasquez	
<input checked="" type="checkbox"/>	run	167756727	
<input checked="" type="checkbox"/>	f_nac	28/10/1987	
<input checked="" type="checkbox"/>	comuna_id	1	
<input checked="" type="checkbox"/>	direccion	direeeeeectrefid	
<input checked="" type="checkbox"/>	mail	cla.blancov	

Body Cookies Headers (11) Test Results Status: 201 Created Time: 1105 ms Size: 379 B

Pretty Raw Preview Visualize JSON

```
1
2  "message": "Cliente creado"
3
```

- Se detalla prueba postman para correcta conexión de api en mi servidor personal.
- Apoyo a mis compañeros de trabajo en lo que es la programación de activitys, además de la integración de API en modulo Usuarios, en la creación de los mismos.