


API PARA MILACNIC

**Módulos, Clientes, Backend, Buenas prácticas,
documentación...**

ELEVATOR PITCH

Para los grandes proveedores de Internet en LATAM que necesitan mantener actualizada la información de clientes, IPs, ASNs y DNS reversos, el API para MILACNIC, es un sistema que permite integrar de manera eficiente los software de administración de recursos de Internet que ya utilizan estos proveedores

MILACNIC



Sergio Rojas Astigarraga
SRA

UY-OPPL-LACNIC

INFORMACIÓN

IP / ASN

ESTADO DE CUENTA

FACTURAS

RPKI

DOCUMENTOS

ESTADO DE RPKI

PAGOS

OTRAS ORGANIZACIONES

Inicio / Organización / IP / ASN

201.219.252.0/22

Sistemas Autónomos

AS52266

IP LACNIC

201.219.252.0/22

2800:700::/32

201.219.252.0 - 201.219.255.255 (AS28000)

ALLOCATED

DNS reverso

No delegado

Subasignaciones

Subasignado

Disponibile

Organización


Fecha Asignación


Fecha Actualización

LACNIC DEBOGON TESTER (UY-OPPL-LACNIC)

15-ago-2014

03-ago-2017

 TECNICO
JOC145

 ABUSO
SRA

Subasignar

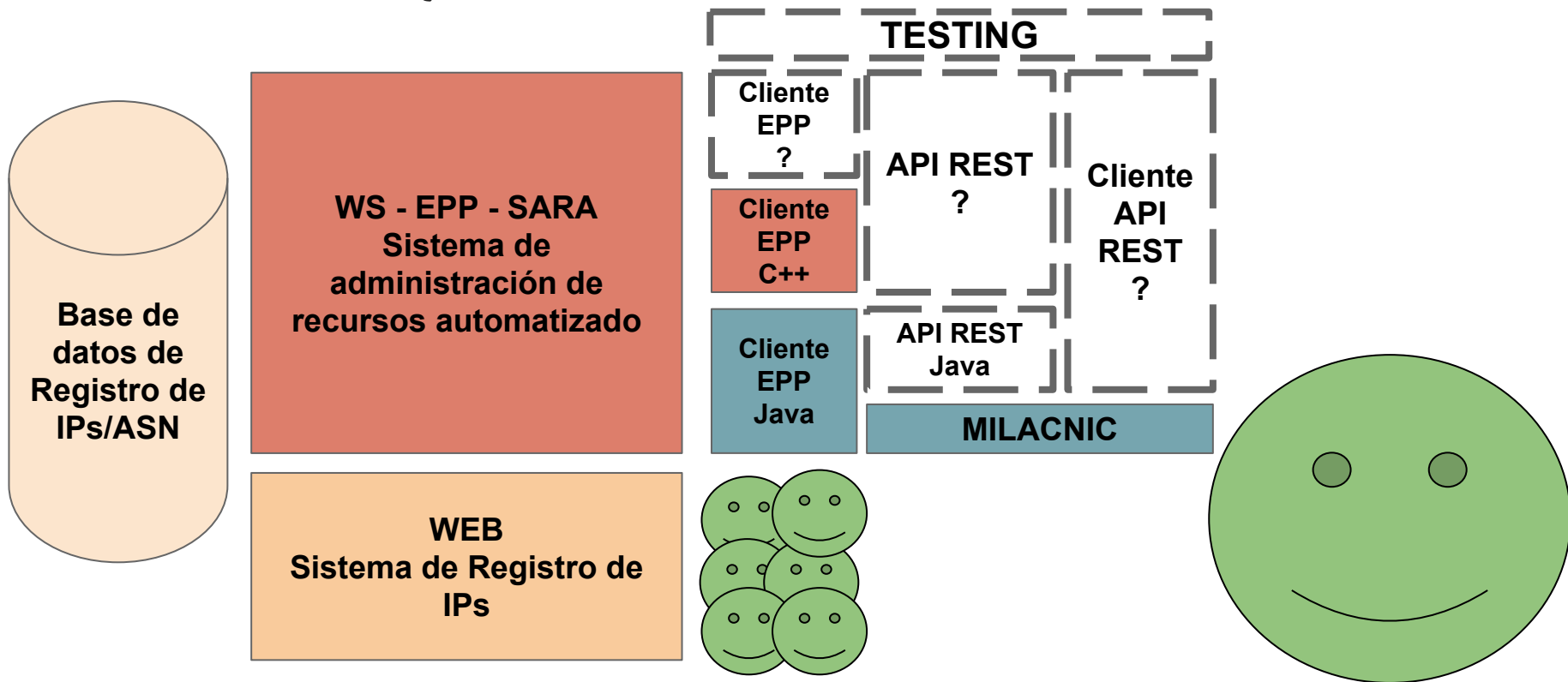
Delegar (rDNS)

Editar Contactos

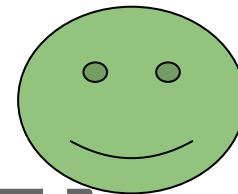
ASN de Origen

Ver Subasignaciones ...

ARQUITECTURA DE SISTEMAS ACTUAL



DONDE PODEMOS TRABAJAR



EPP

**S
A
R
A**

CLIENTE EPP

- NODE.JS
- PYTHON
- C#
- ?

API REST

- NODE.JS
- PYTHON
- C#
- ?

CLIENTE API REST

- JAVA
- ANGULAR
- NODE.JS
- PYTHON
- C++
- PHP
- C#
- ?

**Cliente EPP
C++**

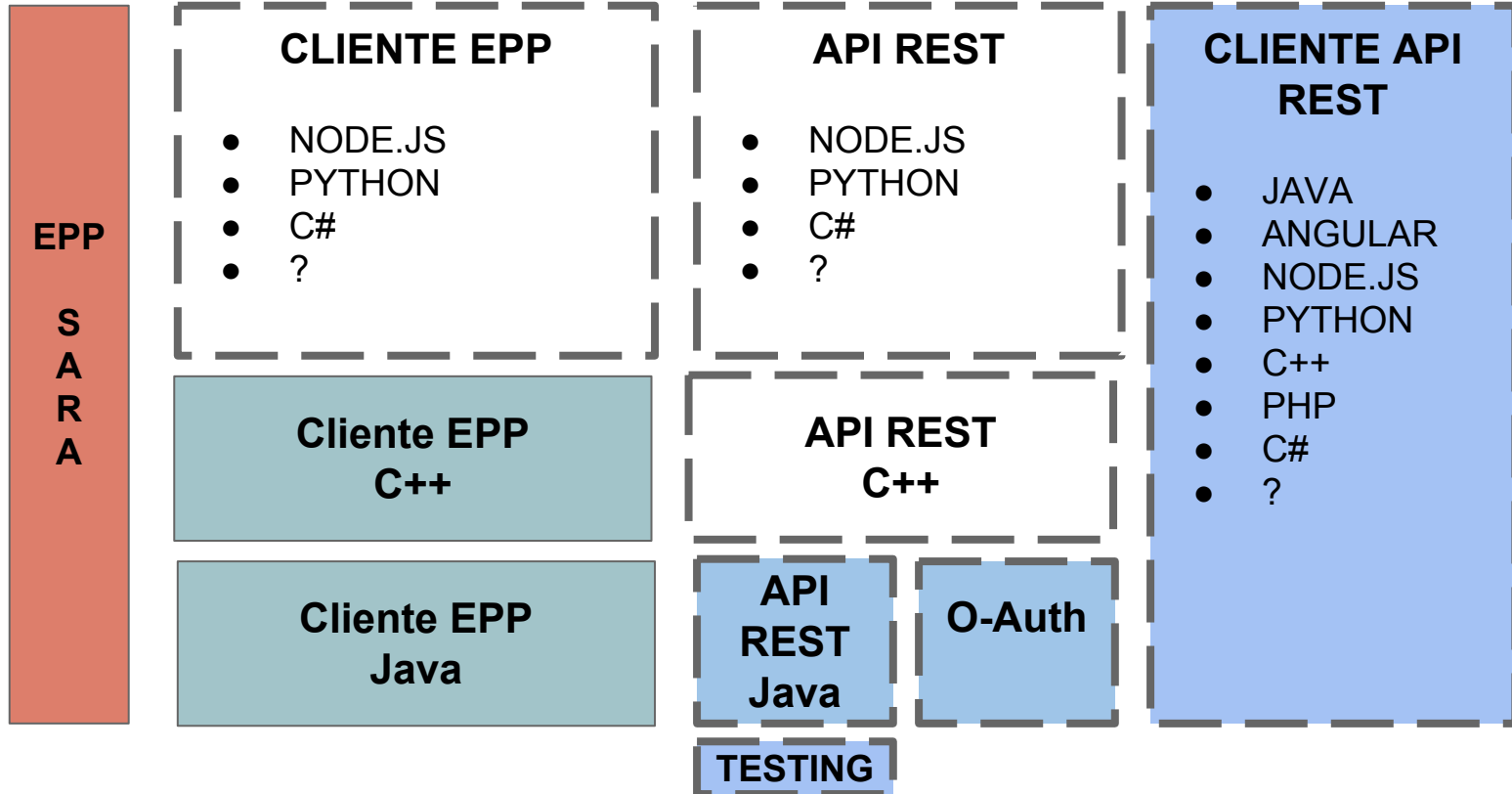
**API REST
C++**

**Cliente EPP
Java**

**API REST
Java**

TESTING

DONDE TRABAJAMOS



DIVISIÓN DEL TRABAJO

API REST JAVA

NICO FIUMARELLI
GERARDO RADA

- Creación de usuarios
- Modificación de usuarios
- Creación de organizaciones
- Modificación de organizaciones

O - AUTH

NICO BUTLER

- Definición de métodos de autenticación

CLIENTE API REST

CARLOS ORTIZ

- Cliente postman

TESTING

JWT (JSON WEB TOKEN)

- Open standard (RFC 7519)
- Info firmada
 - HMAC - usando secreto
 - RSA - private/public key

Enviada a través de una URL, parámetro POST, o en HTTP header

- Estructura
 - Header, Payload ,Signature (xxxxx.yyyyyy.zzzzz)

```
1 HMACSHA256(  
2   base64UrlEncode(header) + "." +  
3   base64UrlEncode(payload),  
4   secret)
```


Browser

Server

1. POST /users/login with username and password

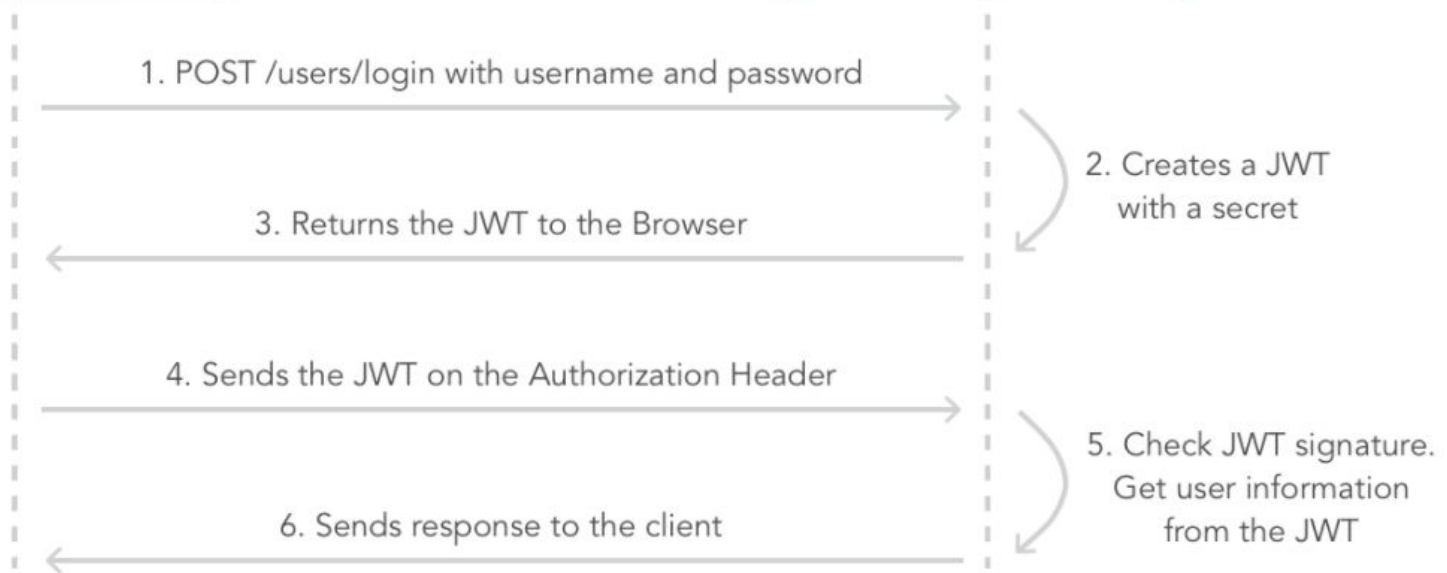
2. Creates a JWT
with a secret

3. Returns the JWT to the Browser

4. Sends the JWT on the Authorization Header

5. Check JWT signature.
Get user information
from the JWT

6. Sends response to the client



CLIENTE POSTMAN (API REST MILACNIC)

The screenshot shows the Postman interface with the following details:

- Runner** and **Import** buttons are visible in the top left.
- Builder** and **Team Library** tabs are in the top center.
- dev_lacnic** is selected in the top right dropdown.
- Filter** and **History** buttons are in the left sidebar.
- Collections** tab is active in the left sidebar, showing a list of collections including **API Rest MI Lacnic** (12 requests), **Users**, **Recursos**, **IPs**, **ASN**, **Organizaciones**, **Domain**, and **Postman Echo** (37 requests).
- create-user** is selected in the **Users** collection.
- POST** method is selected for the **create-user** endpoint.
- Body** tab is active, showing a JSON payload for creating a user.
- JSON (application/json)** is selected as the body type.
- Send** button is visible in the top right of the endpoint area.
- Response** tab is visible at the bottom of the endpoint area.

JSON Payload:

```
1 {
2   "name": "Nicolas",
3   "surname": "Fiumarelli",
4   "email": "nicocamarao@gmail.com",
5   "password": "pepito",
6   "language": "ES",
7   "streetAddress": "Rambla Republica de Mexico",
8   "numberAddress": 2165,
9   "complementAddress": "ap 101",
10  "city": "Montevideo",
11  "state": "Montevideo",
12  "pc": 11800,
13  "country": "Uruguay",
14  "countryCode": 598,
15  "areaCode": 2,
16  "phoneNumber": 2099432,
17  "phoneExtension": 1234,
18  "additionalEmails": [
19    "nicocamarao@aamil.com".
```

Hit the Send button to get a response.