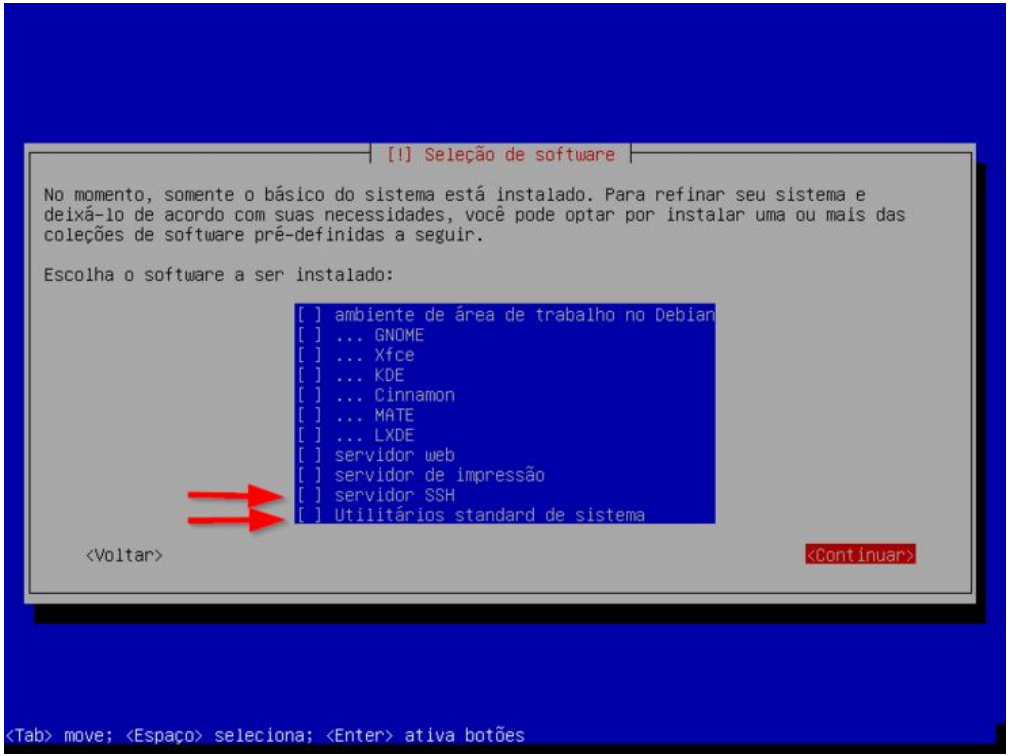
DNS Autoritativo

DNS (Domain Name System) é o sistema responsável pelas traduções de nome legíveis por nós humanos, para endereços IP usados na comunicação através da Internet e vice versa. Sem o DNS seria como termos que decorar CPFs, ao invés dos nomes das pessoas.

O DNS não para por aí. É através dele e de extensões, que usamos outros tipos de registros para aumentar a segurança, confiabilidade, autenticidade e privacidade na troca de dados entre sistemas e em acessos de visitantes. Exemplos de outros usos do DNS:

- **DNSSEC** (Domain Name System Security Extensions) é uma extensão do DNS que adiciona autenticação criptográfica às respostas DNS, garantindo que os dados não foram alterados ou forjados, aumentando confiabilidade e autenticidade dos acessos.
- No combate ao **SPAM**. O DNS com registros de **DKIM** (DomainKeys Identified Mail), **SPF** (Sender Policy Framework), **DMARC** (Domain-based Message Authentication, Reporting, and Conformance) e **TLSA** (TLS Authentication) para adoção do **DANE** (DNS-based Authentication of Named Entities) que garante que apenas o seu servidor de e-mail terá autoridade para enviar mensagens de seus domínios através de assinatura digital, com troca dos dados de forma segura (encriptada).
- Confirmação de autoridade para geração de certificados digitais de um (sub).domínio usando Let's Encrypt na modalidade **DNS-01** usando registros **TXT**.

Para o nosso treinamento utilizaremos o programa NSD da NLnet Labs, iremos configurar um DNS Autoritativo Primário e o Secundário como exemplos. Também utilizaremos um GNU/Linux Debian 12 codinome Bookworm instalação simplificada com apenas estes pacotes selecionados:



O NSD não faz recursividade e não possui cache. Ele foi criado apenas para servir como DNS Autoritativo, assim como o Unbound foi desenvolvido para ser apenas DNS Recursivo. Como boa prática de segurança esses serviços não devem estar juntos. Alguns sysadmins podem ficar tentados a usar um BIND9 no mesmo servidor rodando como Autoritativo e Recursivo. Porque não deveríamos fazer isso:

- DNS Autoritativo precisa estar com porta 53/UDP/TCP aberta para a Internet e por isso pode sofrer ataques diretos no serviço. Já o DNS Recursivo não necessita de estar aberto para consultas com origem a Internet e por isso consultas com destino ao seu DNS Recursivo nas portas 53/UDP/TCP podem ser bloqueadas.
- Um DNS Autoritativo Primário quando fica indisponível, ainda pode transmitir as informações através do DNS Autoritativo Secundário, que pode estar em outro local e inclusive em outro Sistema Autônomo. O impacto, se ambos caírem, seria a indisponibilidade dos domínios contidos neles.
- Um DNS Recursivo Primário quando indisponível não causa impactos, desde que o DNS Recursivo Secundário esteja funcionando. Contudo se ambos caírem, o impacto será a indisponibilidade de Internet para todos os clientes utilizando esses servidores.

Para o nosso lab teremos 2 VMs com IPv4 e IPv6, onde faremos o DNS Autoritativo Primário e o Secundário:

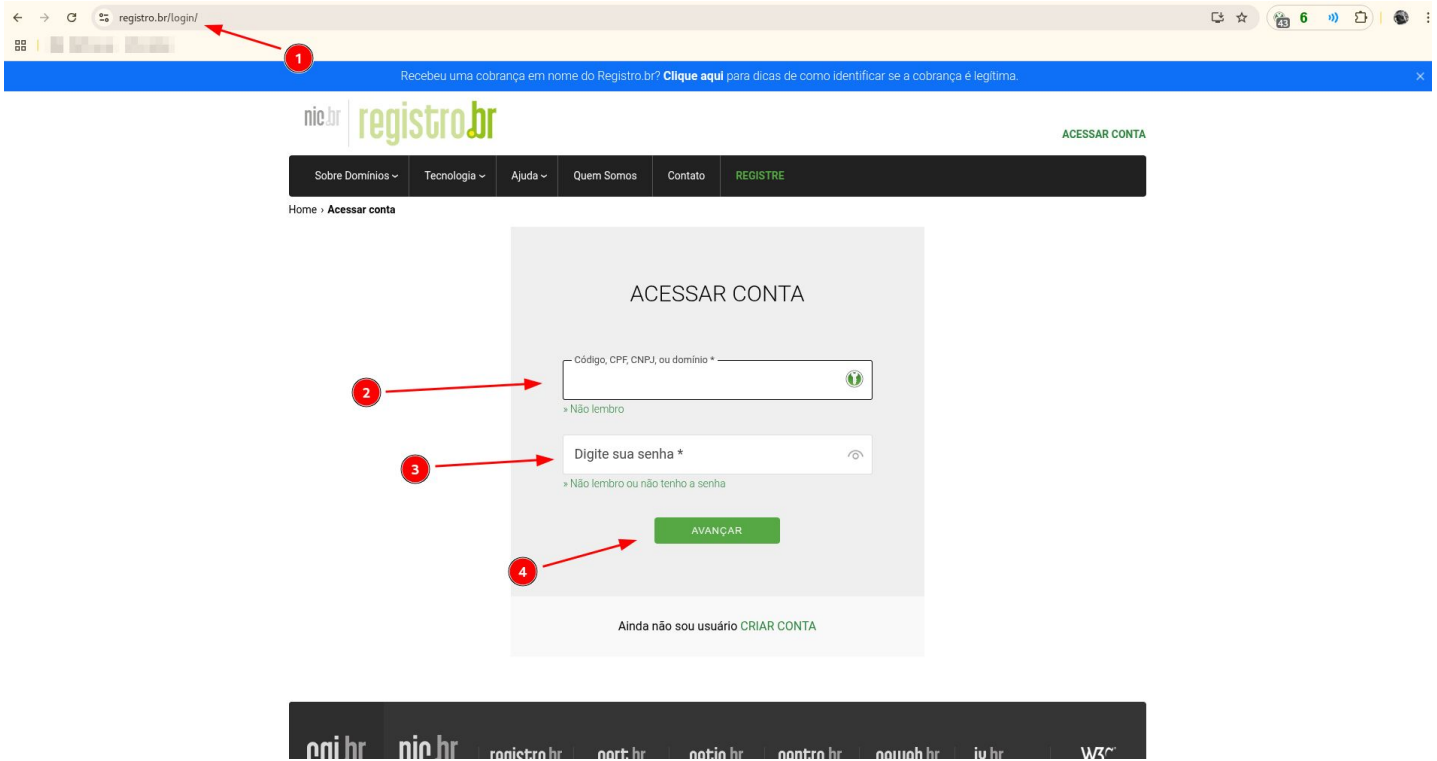
- hostname: spo-nsd-01
- IPv4: 179.190.3.54
- IPv6: 2804:ad4:ff18::6

- hostname: spo-nsd-02
- IPv4: 179.190.3.55
- IPv6: 2804:ad4:ff18::7

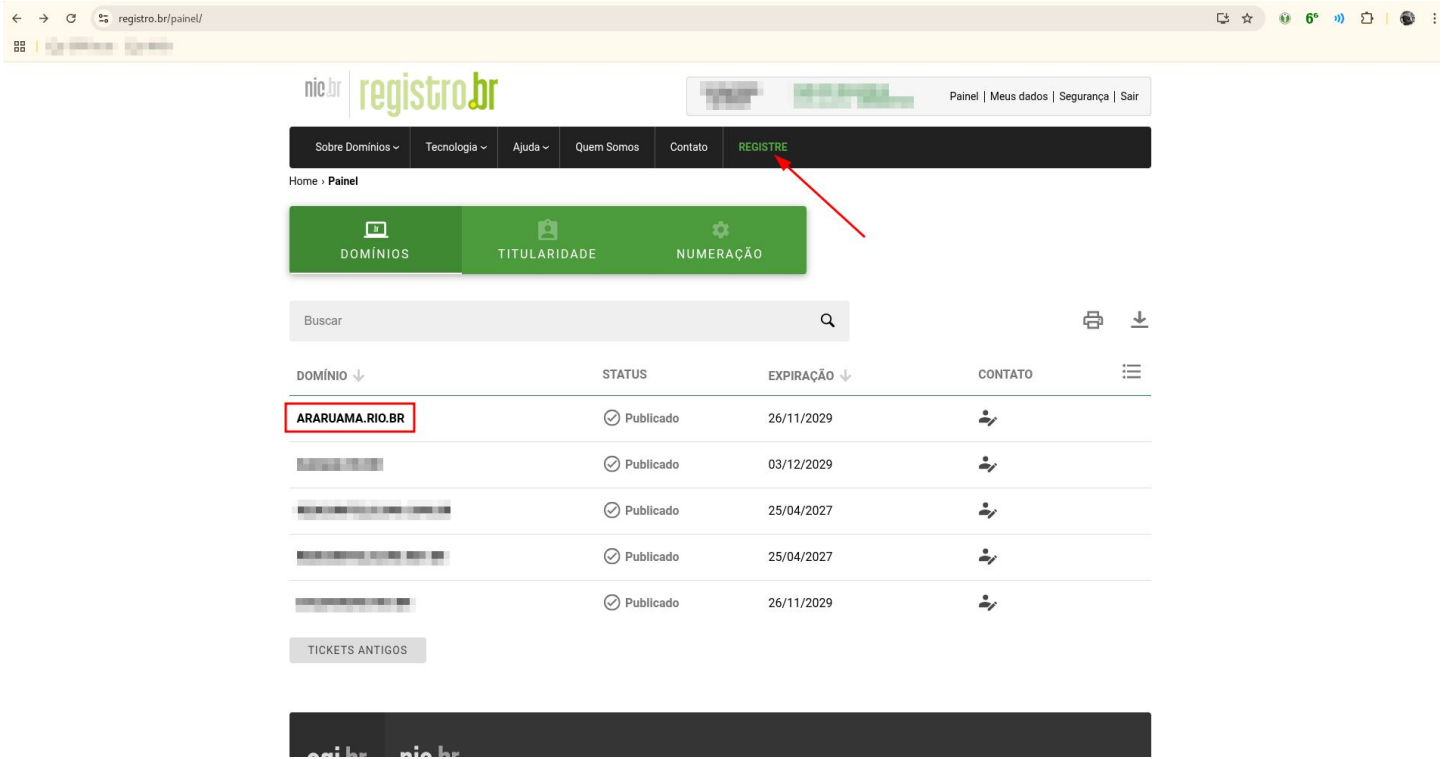
Vamos também além de configurar nossos DNS(s) Autoritativos para servir o domínio **araruama.rio.br**, iremos fazer o DNSSEC assinando o arquivo de zona.

Registro.br

Antes de iniciar você precisa ter seu domínio registrado em algum lugar, como estaremos usando o domínio **araruama.rio.br** , faremos isso no Registro.br:

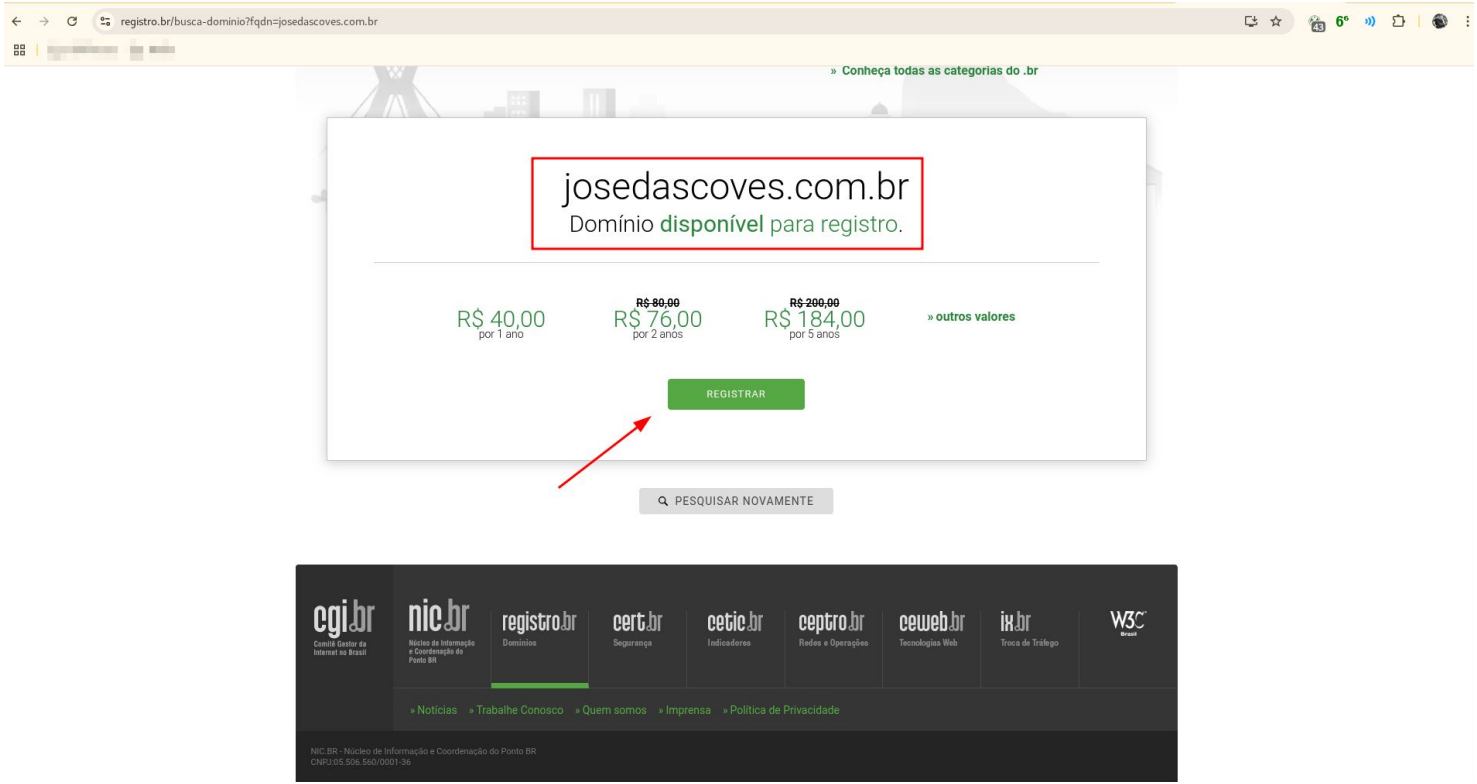


No nosso caso já temos o domínio registrado mas caso não tenha, só clicar em **REGISTRE**:

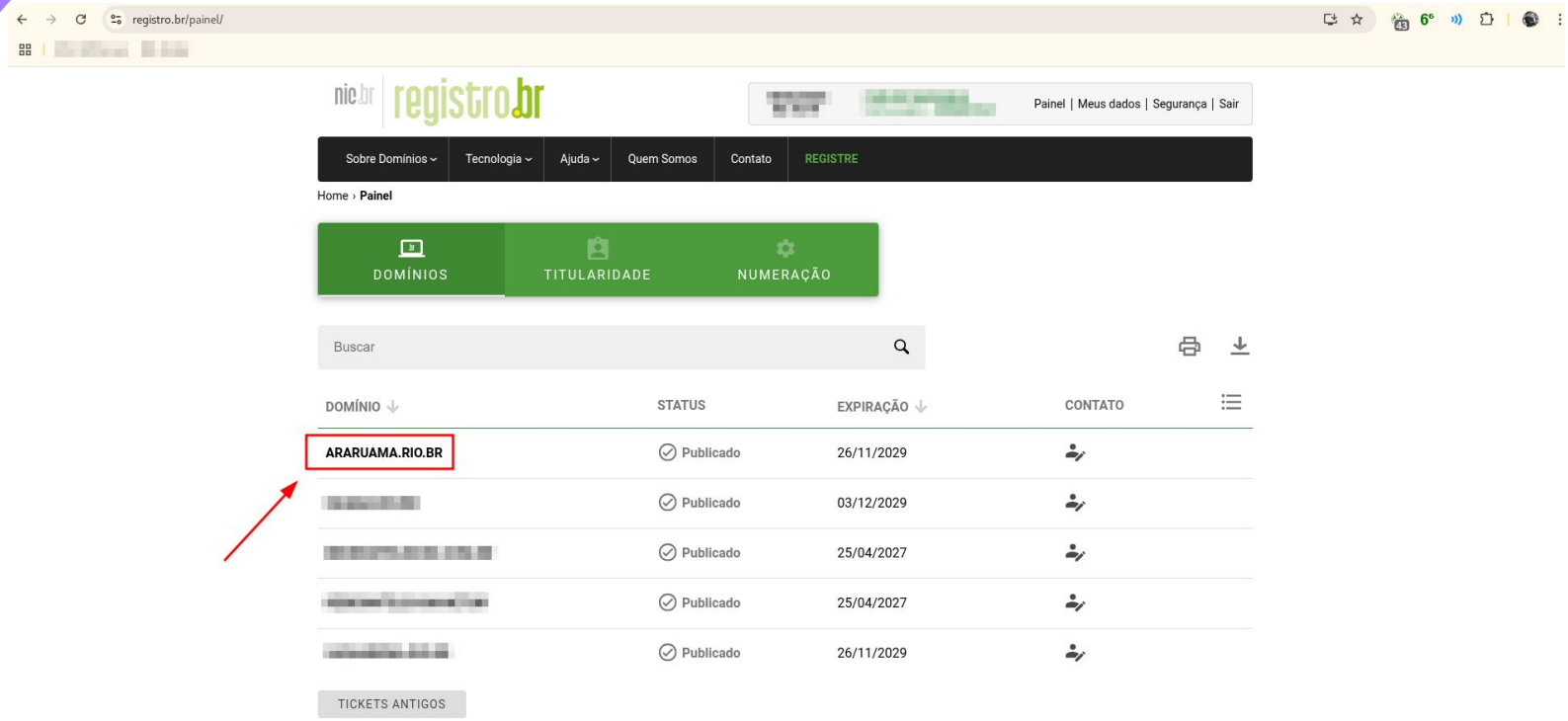


Na próxima tela você fará uma busca pelo domínio desejado e se tiver disponibilidade…

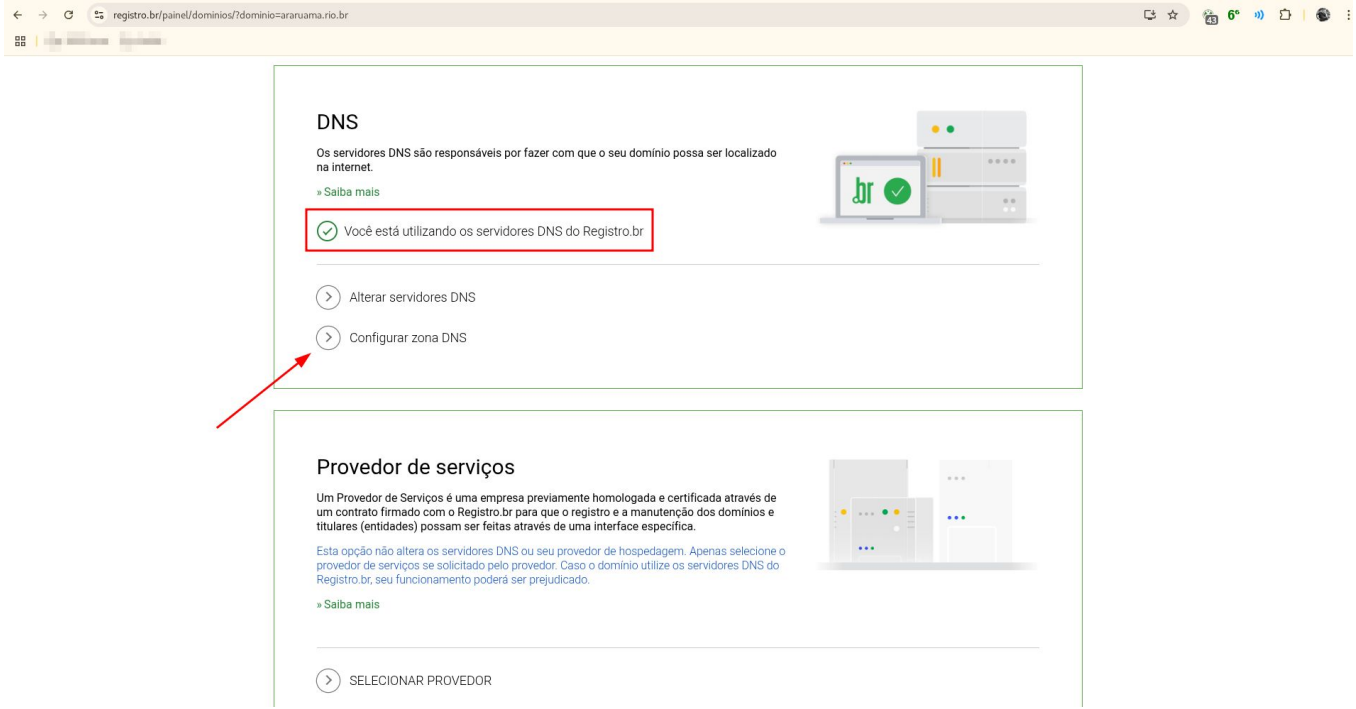
Basta clicar em **REGISTRAR** e seguir o passo a passo até o final:



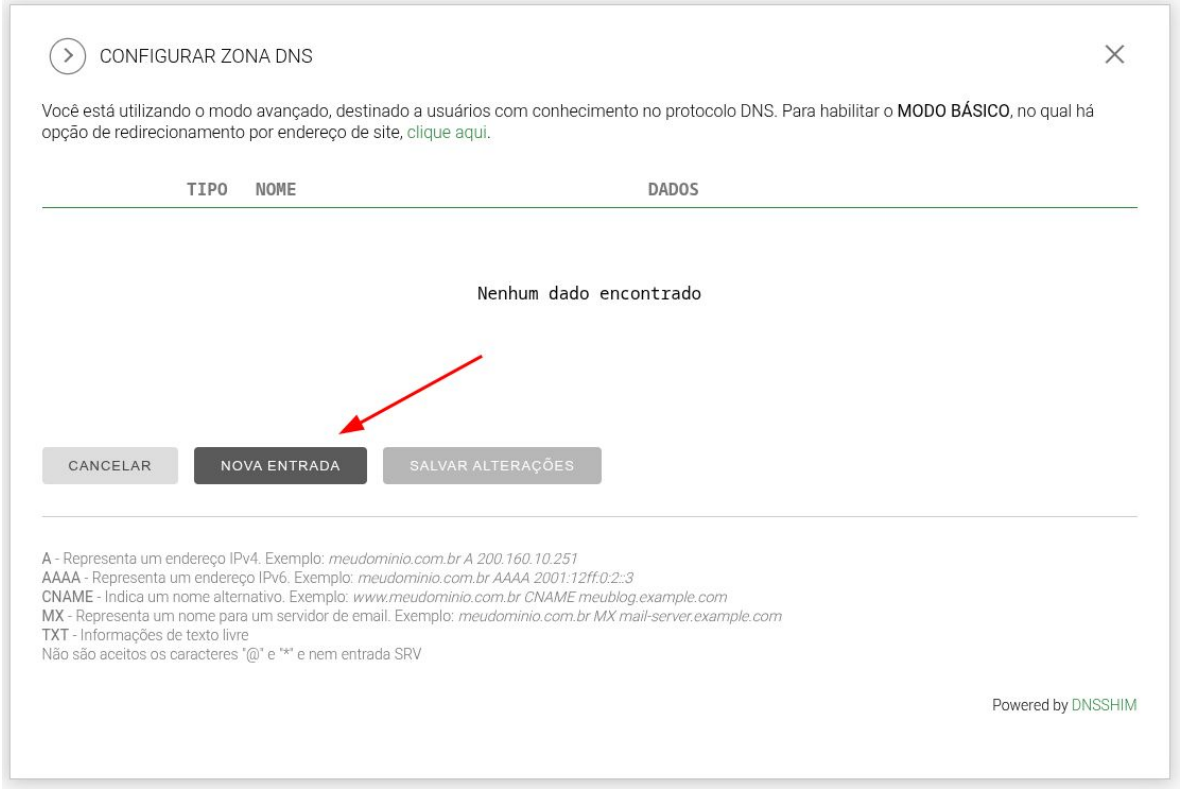
Depois de registrado e pago, voltamos nessa tela e clicamos no domínio que vamos trabalhar:



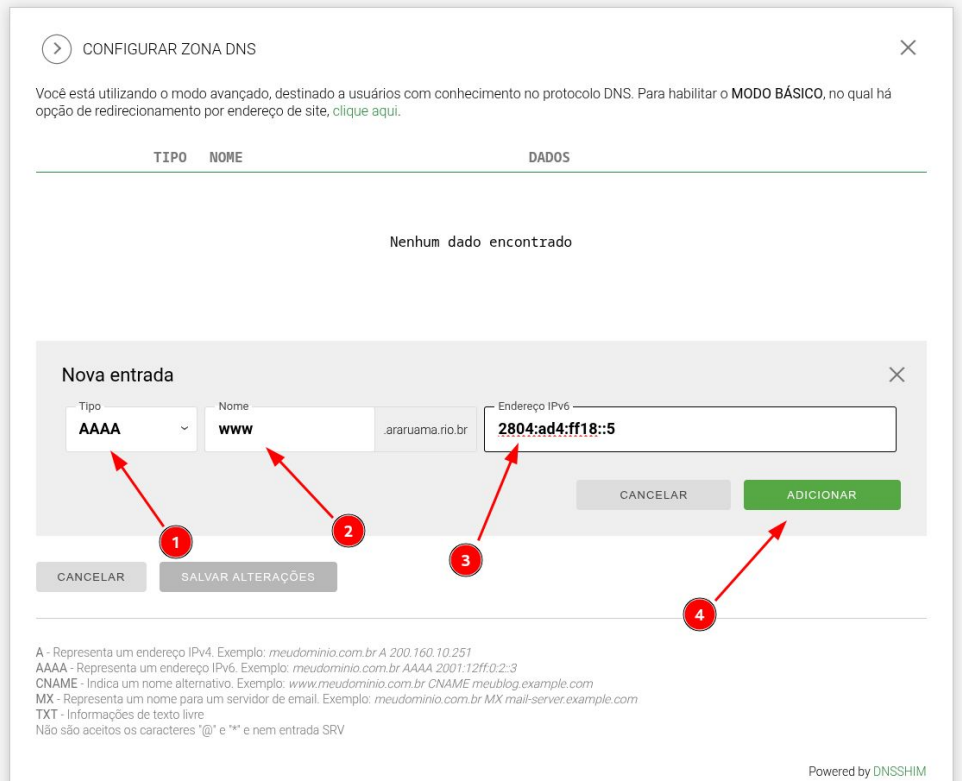
Rolamos a tela para baixo até chegarmos em DNS e vamos clicar em **Configurar zona DNS** se caso formos usar os servidores Autoritativos do **Registro.br**. Se não, deixaremos como está por enquanto. O **Registro.br** já oferece automaticamente DNSSEC para o seu domínio:



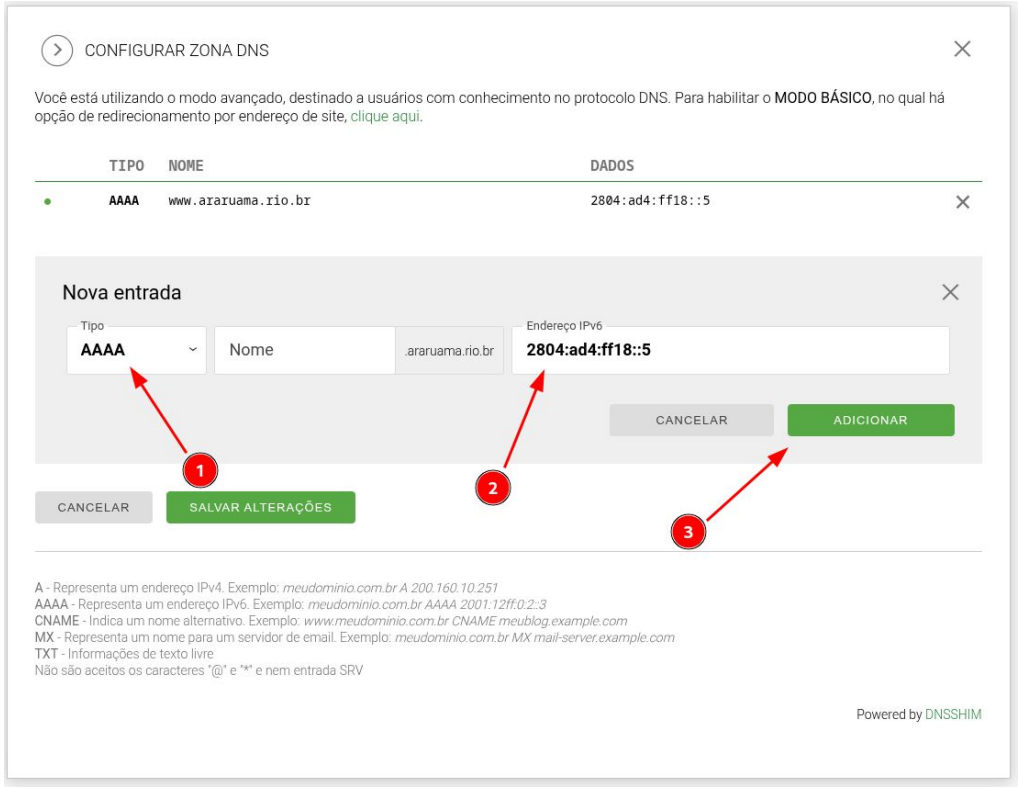
Clique em NOVA ENTRADA:



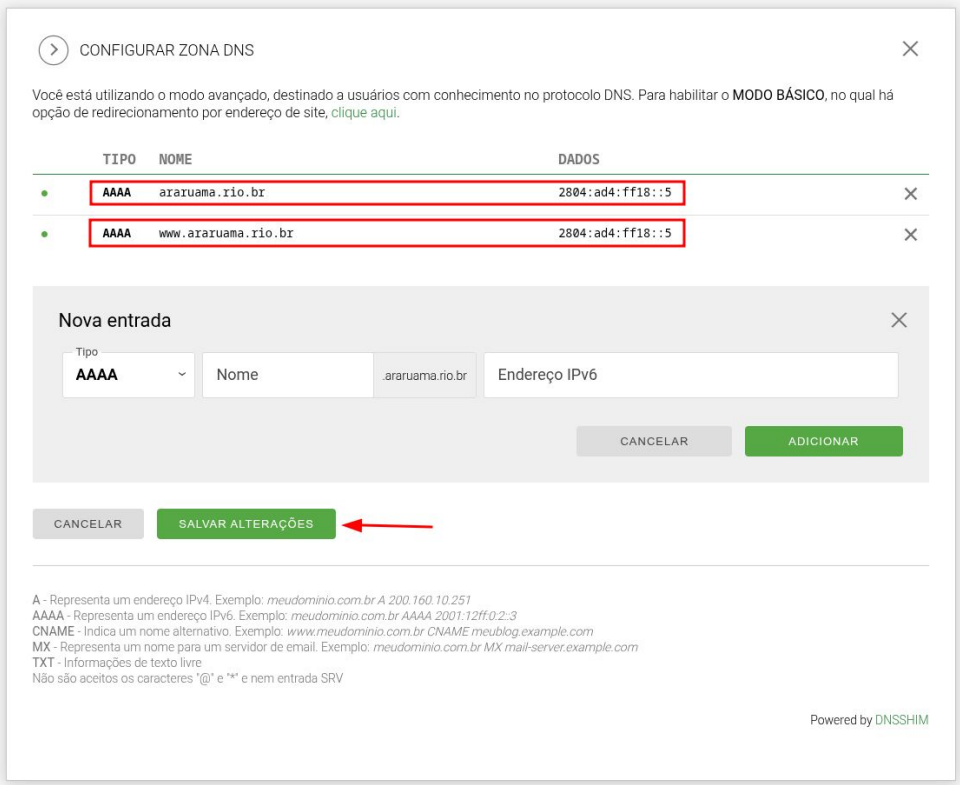
Vamos adicionar o primeiro registro **www.araruama.rio.br**:



Vamos adicionar o registro **araruama.rio.br** também apontando pro nosso site:



Criados os registros, clicamos em **SALVAR ALTERAÇÕES** e aguardamos a publicação:



NSD
(Name Server Daemon)

Instalando os pacotes que usaremos nos 2 servidores, primário e secundário:

```
root@spo-nsd-01:~# apt install nsd ldnsutils haveged
Lendo listas de pacotes... Pronto
Construindo árvore de dependências... Pronto
Lendo informação de estado... Pronto
The following additional packages will be installed:
  dns-root-data libevent-2.1-7 libfstrm0 libhavege2 libldns3 libpcap0.8 libprotobuf-c1
Os NOVOS pacotes a seguir serão instalados:
  dns-root-data haveged ldnsutils libevent-2.1-7 libfstrm0 libhavege2 libldns3 libpcap0.8
  libprotobuf-c1 nsd
0 pacotes atualizados, 10 pacotes novos instalados, 0 a serem removidos e 0 não atualizados.
É preciso baixar 1.351 kB de arquivos.
Depois desta operação, 4.847 kB adicionais de espaço em disco serão usados.
Você quer continuar? [S/n] S
```

Processos do NSD rodando mas sem nenhuma configuração ainda.

```
    460 ?        Ss     0:00 /usr/bin/dbus-daemon --system --address=systemd: --nofork --nopidfile -
    462 ?        Ss     0:03 /lib/systemd/systemd-logind
    505 tty1     Ss     0:00 /bin/login -p --
    585 tty1     S      0:00  \_ -bash
    588 tty1     S      0:00      \_ su -
    589 tty1     S+     0:00          \_ -bash
    578 ?        Ss     0:00 /lib/systemd/systemd --user
    579 ?        S      0:00  \_ (sd-pam)
  34785 ?        Ss     0:00 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups
  88865 ?        Ss     0:00  \_ sshd: root@pts/0
  88871 pts/0    Ss     0:00      \_ -bash
  88893 pts/0    R+     0:00          \_ ps afx
  62800 ?        Ss     0:01 /lib/systemd/systemd-journald
  62907 ?        Ssl    0:02 /lib/systemd/systemd-timesyncd
  62938 ?        Ss     0:00 /lib/systemd/systemd-udevd
  87400 ?        Ss     0:00 /lib/systemd/systemd --user
  87401 ?        S      0:00  \_ (sd-pam)
  88787 ?        Ss     0:00 /usr/sbin/nsd -d -P
  88788 ?        S      0:00  \_ /usr/sbin/nsd -d -P
  88789 ?        S      0:00      \_ /usr/sbin/nsd -d -P
  88843 ?        Ss     0:00 /usr/sbin/haveged --Foreground --verbose=1
root@spo-nsd-01:~#
```

Toda parte estrutural do NSD se localiza em **/etc/nsd** e criaremos o diretório onde ficarão os arquivos das zonas:

```
# mkdir -p /etc/nsd/zones
```

Criaremos nosso arquivo de configuração base para o nosso DNS Autoritativo Primário conforme abaixo em **/etc/nsd/nsd.conf.d/local.conf**:

```
server:
        server-count: 1
        ip-address: 179.190.3.54
        ip-address: 2804:ad4:ff18::6
        ip-address: 127.0.0.1
        ip-address: ::1
        do-ip4: yes
        do-ip6: yes
        username: nsd
        zonesdir: "/etc/nsd"
        database: "/var/lib/nsd/nsd.db"
        logfile: "/var/log/nsd.log"
        pidfile: "/run/nsd/nsd.pid"
        hide-version: yes
        refuse-any: yes
        hide-identity: yes
        minimal-responses: yes
```

```
remote-control:
        control-enable: yes
        control-interface: 127.0.0.1
        control-interface: ::1
        control-port: 8952
        server-key-file: "/etc/nsd/nsd_server.key"
        server-cert-file: "/etc/nsd/nsd_server.pem"
        control-key-file: "/etc/nsd/nsd_control.key"
        control-cert-file: "/etc/nsd/nsd_control.pem"

key:
        name: "serverkey"
        algorithm: sha256
        secret: "6KCbnCFUwdeDgeC0cO5ihbdeVVjBWnm2h0PQYFRC5Rw="
```

Os parâmetros em vermelho são os que iremos alterar conforme abaixo:

- **server-count** - quantidade de cores do servidor.
- **ip-address** - IPv4 e IPv6 do servidor.
- **name** - nome de referência da chave secreta. Deve ser usando em ambos servidores.
- **secret** - chave secreta. Usada para a comunicação entre o servidor Primário e o Secundário.

Para gerar essa chave secreta é bem simples. Execute o comando abaixo:

```
# dd if=/dev/random of=/dev/stdout count=1 bs=32 | base64
6KCbnCFUwdeDgeC0cO5ihbdeVVjBWnm2h0PQYFRC5Rw=
1+0 records in
1+0 records out
32 bytes copied, 0,000164304 s, 195 kB/s
```

Após criarmos nosso **/etc/nsd/nsd.conf.d/local.conf**, vamos reiniciar o serviço:

```
# systemctl restart nsd.service
```

Agora vamos configurar nosso arquivo de configuração da zona. Para que possamos ter uma organização de fácil administração, iremos criar um arquivo por domínio. Faremos então nosso arquivo **/etc/nsd/nsd.conf.d/araruama.rio.br.conf** com o seguinte conteúdo:

```
zone:
    name: "araruama.rio.br"
    zonefile: "/etc/nsd/zones/araruama.rio.br.zone.signed"
    notify: 2804:ad4:ff18::7 serverkey
    provide-xfr: 2804:ad4:ff18::7 serverkey
```

Na sequência vamos criar o arquivo **/etc/nsd/zones/araruama.rio.br.zone** e assinar ele. É importante o **;** **serial** e veremos o porquê lá na frente:

```
$TTL        3600
@       IN      SOA     ns1.araruama.rio.br. contato.araruama.rio.br. (
                        2025061904        ; serial (ano + mes + dia + versao do dia)
                        28800              ; refresh (8 horas)
                        7200              ; retry (2 horas)
                        604800              ; expire (7 dias)
                        3600 )            ; minimum (1 hora)
              IN NS ns1.araruama.rio.br.
              IN NS ns2.araruama.rio.br.

ns1.araruama.rio.br.    IN   A       179.190.3.54
ns2.araruama.rio.br.    IN   A       179.190.3.55
ns1.araruama.rio.br.    IN   AAAA    2804:ad4:ff18::6
ns2.araruama.rio.br.    IN   AAAA    2804:ad4:ff18::7
araruama.rio.br.        IN   AAAA    2804:ad4:ff18::5
www.araruama.rio.br.    IN   CNAME   araruama.rio.br.
```

Vamos assinar o nosso **/etc/nsd/zones/araruama.rio.br.zone**:

```
# cd /etc/nsd/zones

# ldns-keygen -a ECDSAP256SHA256 -b 1024 araruama.rio.br
Kararuama.rio.br.+013+14362

# ldns-keygen -k -a ECDSAP256SHA256 -b 2048 araruama.rio.br
Kararuama.rio.br.+013+57764

# rm *.ds

# ldns-signzone -b -o araruama.rio.br /etc/nsd/zones/araruama.rio.br.zone
/etc/nsd/zones/Kararuama.rio.br.+013+14362 /etc/nsd/zones/Kararuama.rio.br.+013+57764

# ls -l
total 28
-rw-r--r-- 1 root root  710 jun 18 01:23 araruama.rio.br.zone
-rw-r--r-- 1 root root 3342 jun 18 01:41 araruama.rio.br.zone.signed
-rw-r--r-- 1 root root  158 jun 18 01:37 Kararuama.rio.br.+013+14362.key
-rw------- 1 root root  114 jun 18 01:37 Kararuama.rio.br.+013+14362.private
-rw-r--r-- 1 root root  158 jun 18 01:38 Kararuama.rio.br.+013+57764.key
-rw------- 1 root root  114 jun 18 01:38 Kararuama.rio.br.+013+57764.private

# systemctl restart nsd.service
```

Na listagem do slide anterior, podemos ver que agora existe o arquivo **araruama.rio.br.zone.signed** que significa que nosso domínio **araruama.rio.br** está assinado.

Sempre alterar alguma coisa na configuração:

```
# nsd-control reconfig
```

Quando alterar algo no domínio:

```
# nsd-control reload araruama.rio.br
```

Para testarmos se nosso domínio está assinado com DNSSEC vamos instalar mais um pacote no Debian, que servirá com ferramentas de testes onde teremos o **dig** e o **host**:

```
# apt install dnsutils
```

Testando se nosso domínio de testes está OK:

```
# dig DNSKEY araruama.rio.br. @localhost +multiline +norec

; <<>> DiG 9.18.33-1~deb12u2-Debian <<>> DNSKEY araruama.rio.br. @localhost +multiline +norec
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY,  status: NOERROR, id: 11659
;; flags: qr aa; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;araruama.rio.br.      IN DNSKEY

;; ANSWER SECTION:
araruama.rio.br.       3600 IN DNSKEY 256 3 13 (
                               slq04OEddTAjTcCHMu7IaNXaVaRP3KDdsKQHEmuVm+Nm
                               kfmVjpUmhOc8hzCz0zPcNLzHMDseS/DARUitL0uVVg==
                               ) ; ZSK; alg = ECDSAP256SHA256 ; key id = 14362
araruama.rio.br.       3600 IN DNSKEY 257 3 13 (
                               1KBOr95P2EkftHroDxKT1AkFJMt66o+KVPRmnCTVsc07
                               XFP9/IdRNVTNxUqcLcxBR6SEUV6MjQee5w6EYCzW8A==
                               ) ; KSK; alg = ECDSAP256SHA256 ; key id = 57764

;; Query time: 0 msec
;; SERVER: ::1#53(localhost) (UDP)
;; WHEN: Wed Jun 18 10:23:13 -03 2025
;; MSG SIZE  rcvd: 204
```

Todo esse processo de assinatura tem uma validade e precisa ser renovado periodicamente. Por isso para facilitar nossa vida fiz um script que colocaremos no **/etc/crontab** para rodar de tempos em tempos. Para isso vamos preparar o ambiente:

```
# mkdir -p /root/scripts
```

O script quando for executado, irá procurar no diretório **/etc/nsd/zones** por arquivos de zona **.zone** e que tenham assinatura **.zone.signed**. Importante mantermos essa estrutura para que o script funcione conforme esperado mas se desejar alterar a estrutura, precisará modificar o script. O script também faz um backup das zonas para **/root/backup** antes de qualquer alteração.

Lembram que comentei sobre o **; serial**? Esse comentário é necessário porque nosso script vai buscar essa linha para fazer a incrementação do serial, importante para indicar que houve mudança no arquivo de zona.

Abaixo o nosso script **/root/scripts/dnszonesigner.sh**:

```
#!/bin/bash
/usr/sbin/nsd-control reconfig
PDIR=`pwd`
DATA_ATUAL=$((`date +%Y%m%d`*100))
mkdir -p /root/backup
ZONEDIR="/etc/nsd/zones" # localizacao dos arquivos de zona
cd $ZONEDIR
for domains in `ls *.zone`; do
   DOMAIN=$(basename $domains .zone)
   if [ -a $ZONEDIR/$DOMAIN.zone.signed ]; then
      cp $ZONEDIR/$DOMAIN.zone /root/backup/
      DATA_ZONA=`grep -i "serial" $domains | awk '{ print $1 }'`
      if [ $DATA_ZONA -lt $DATA_ATUAL ]; then
         sed -i "s/$DATA_ZONA/$DATA_ATUAL/g" $DOMAIN.zone
      else
         DATA_NOVA=$(($DATA_ZONA+1))
         sed -i "s/$DATA_ZONA/$DATA_NOVA/g" $DOMAIN.zone
      fi
       KSK=$(basename $(grep -r "`grep '(ksk)' $DOMAIN.zone.signed | cut -f3-10`" K$DOMAIN.*.key | cut
-d':' -f1) .key)
       ZSK=$(basename $(grep -r "`grep '(zsk)' $DOMAIN.zone.signed | cut -f3-10`" K$DOMAIN.*.key | cut
-d':' -f1) .key)
        /usr/bin/ldns-signzone -b -s $(head -n 1000 /dev/random | sha1sum | cut -b 1-16) -o $DOMAIN
$ZONEDIR/$DOMAIN.zone $ZONEDIR/$ZSK $ZONEDIR/$KSK
      /usr/sbin/nsd-control reload $DOMAIN
      /usr/sbin/nsd-control notify $DOMAIN
   fi
done
cd $PDIR
```

Adicionamos assim nosso script no **/etc/crontab**:

```
00 1    * * 7   root    /root/scripts/dnszonesigner.sh
```

Todo domingo às 01:00, o script renovará as assinaturas de todas as zonas.

Agora que temos nosso servidor **DNS Autoritativo Primário**, chegou a hora de configurarmos o secundário. A configuração é bem mais simples porque o secundário apenas sincroniza as zonas que estão no primário e armazena em uma base de dados dele, para as consultas.

Instalem os mesmos programas que instalamos no servidor primário:

```
# apt install nsd ldnsutils haveged dnsutils
```

Na sequência criaremos nosso **/etc/nsd/nsd.conf.d/local.conf** com a configuração referente ao nosso DNS Autoritativo Secundário.

Nosso **/etc/nsd/nsd.conf.d/local.conf**

```
server:
        server-count: 1
        ip-address: 179.190.3.55
        ip-address: 2804:ad4:ff18::7
        ip-address: 127.0.0.1
        ip-address: ::1
        do-ip4: yes
        do-ip6: yes
        username: nsd
        zonesdir: "/etc/nsd"
        database: "/var/lib/nsd/nsd.db"
        logfile: "/var/log/nsd.log"
        pidfile: "/run/nsd/nsd.pid"
        hide-version: yes
        refuse-any: yes
        hide-identity: yes
        hide-version: yes
        minimal-responses: yes

remote-control:
        control-enable: yes
        control-interface: 127.0.0.1
        control-interface: ::1
        control-port: 8952
        server-key-file: "/etc/nsd/nsd_server.key"
        server-cert-file: "/etc/nsd/nsd_server.pem"
        control-key-file: "/etc/nsd/nsd_control.key"
        control-cert-file: "/etc/nsd/nsd_control.pem"

 key:
        name: "serverkey"
        algorithm: sha256
        secret: "5KCbnCFUwdeDgeC0cO5ihbdeVVjBWnm2h0PQYFRC5Rw="
```

Também vamos organizar os domínios por arquivos separados e por isso criaremos o nosso **/etc/nsd/nsd.conf.d/araruama.rio.br.conf** com o seguinte conteúdo:

```
zone:
    name: "araruama.rio.br"
    allow-notify: 2804:ad4:ff18::6 serverkey
    request-xfr: AXFR 2804:ad4:ff18::6 serverkey
```

Na sequência vamos forçar a atualização do domínio **araruama.rio.br** no servidor secundário com o seguinte comando:

```
# nsd-control reconfig
# nsd-control force_transfer araruama.rio.br
```

Será que funcionou? Vamos testar nosso secundário se recebeu a zona do domínio **araruama.rio.br**?

```
# dig DNSKEY araruama.rio.br. @localhost +multiline +norec

; <<>> DiG 9.18.33-1~deb12u2-Debian <<>> DNSKEY araruama.rio.br. @localhost +multiline +norec
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY,  status: NOERROR, id: 2496
;; flags: qr aa; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;araruama.rio.br.    IN DNSKEY

;; ANSWER SECTION:
araruama.rio.br.     3600 IN       DNSKEY 256 3 13 (
                           slq04OEddTAjTcCHMu7IaNXaVaRP3KDdsKQHEmuVm+Nm
                           kfmVjpUmhOc8hzCz0zPcNLzHMDseS/DARUitL0uVVg==
                           ) ; ZSK; alg = ECDSAP256SHA256 ; key id = 14362
araruama.rio.br.     3600 IN       DNSKEY 257 3 13 (
                           1KBOr95P2EkftHroDxKT1AkFJMt66o+KVPRmnCTVsc07
                           XFP9/IdRNVTNxUqcLcxBR6SEUV6MjQee5w6EYCzW8A==
                           ) ; KSK; alg = ECDSAP256SHA256 ; key id = 57764

;; Query time: 0 msec
;; SERVER: ::1#53(localhost) (UDP)
;; WHEN: Wed Jun 18 11:55:22 -03 2025
;; MSG SIZE  rcvd: 204
```

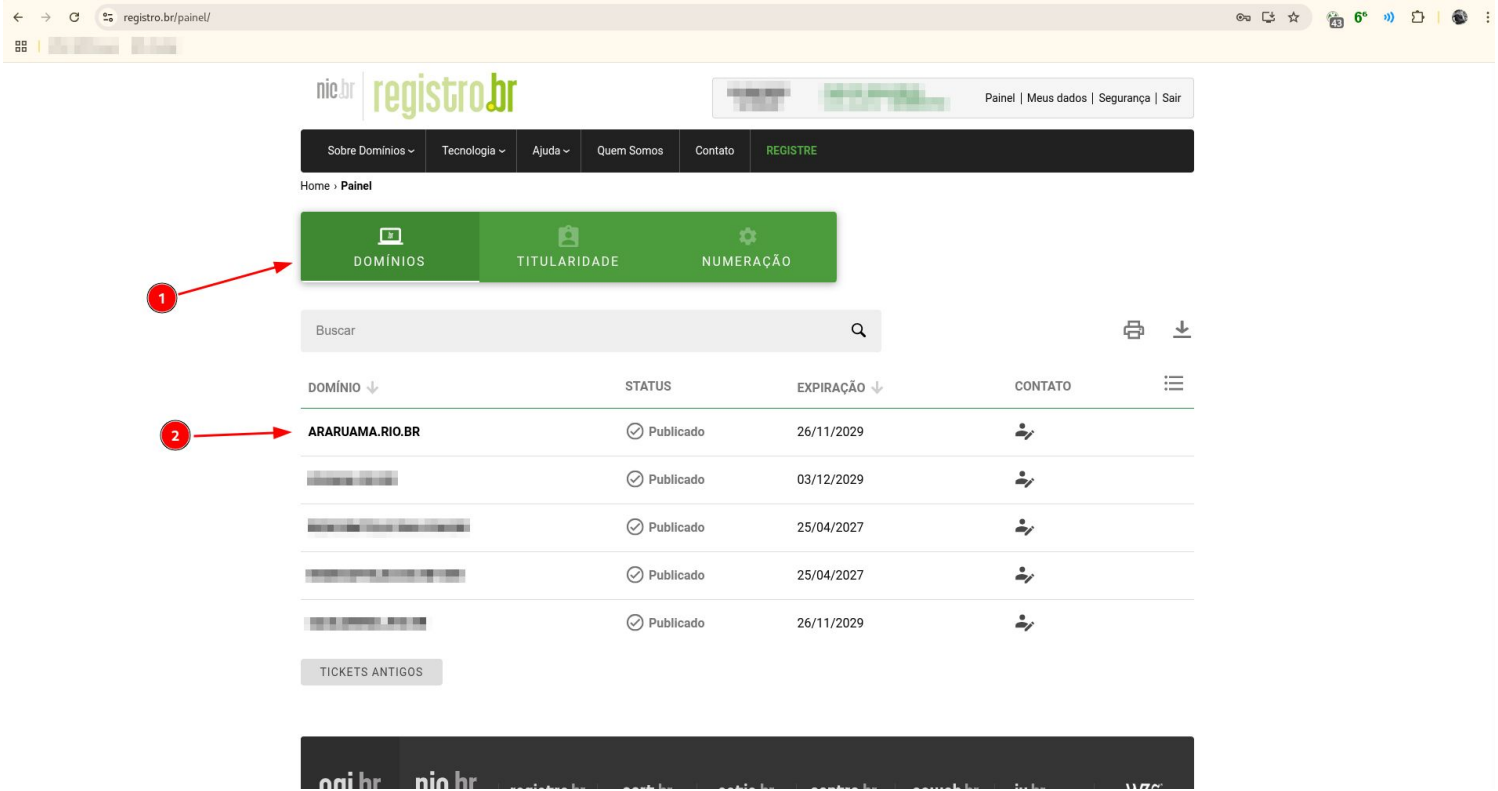
Agora que temos os servidores funcionando, precisamos configurá-los para uso no **Registro.br**. Mas antes precisamos extrair algumas informações para usarmos na configuração.
No servidor primário façamos o seguinte:

```
# ldns-key2ds -n -2 /etc/nsd/zones/araruama.rio.br.zone.signed
araruama.rio.br. 3600    IN      DS      57764 13 2 fe65a2d3c9ebbd383aacd71e63846bf9060edae927cdf32cfe9a919d9b6a85ab
```

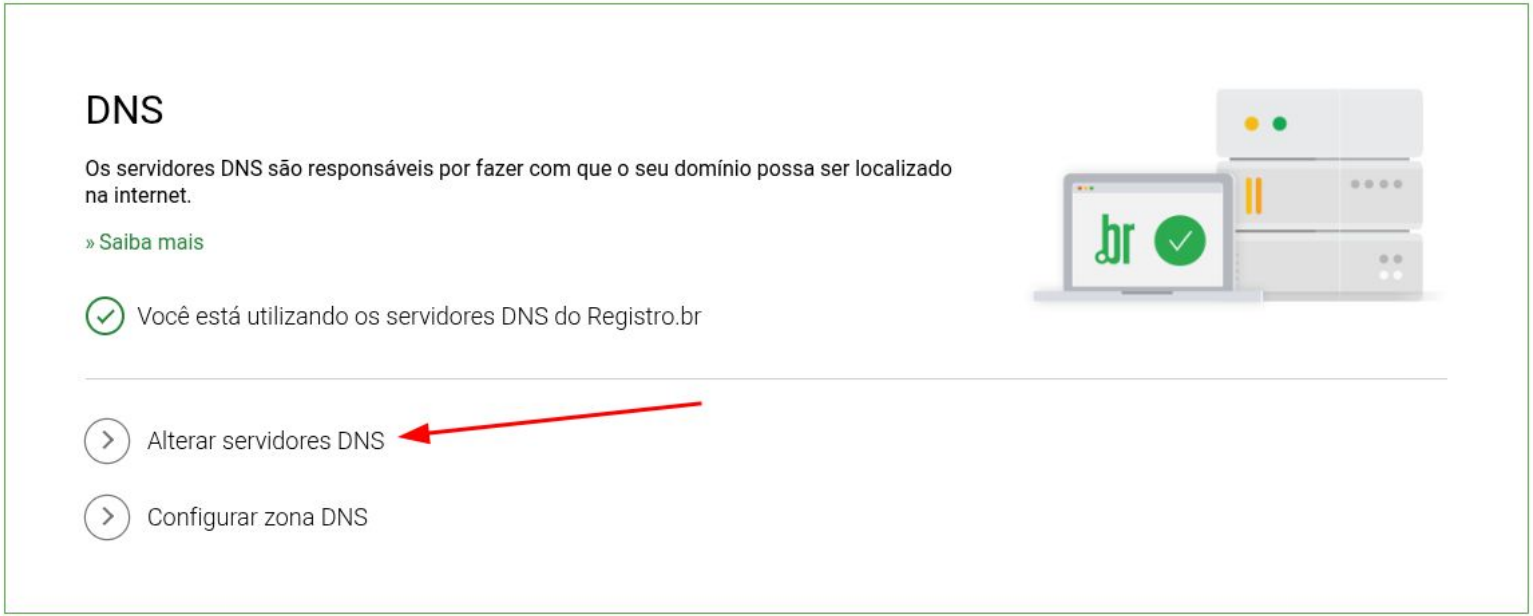
Nós precisaremos do **key tag 57764** e do **digest fe65a2d3c9ebbd383aacd71e63846bf9060edae927cdf32cfe9a919d9b6a85ab**

De posse desses dois valores podemos ir ao Registro.br fazer a configuração dos servidores DNS e do DNSSEC para o domínio **araruama.rio.br**.

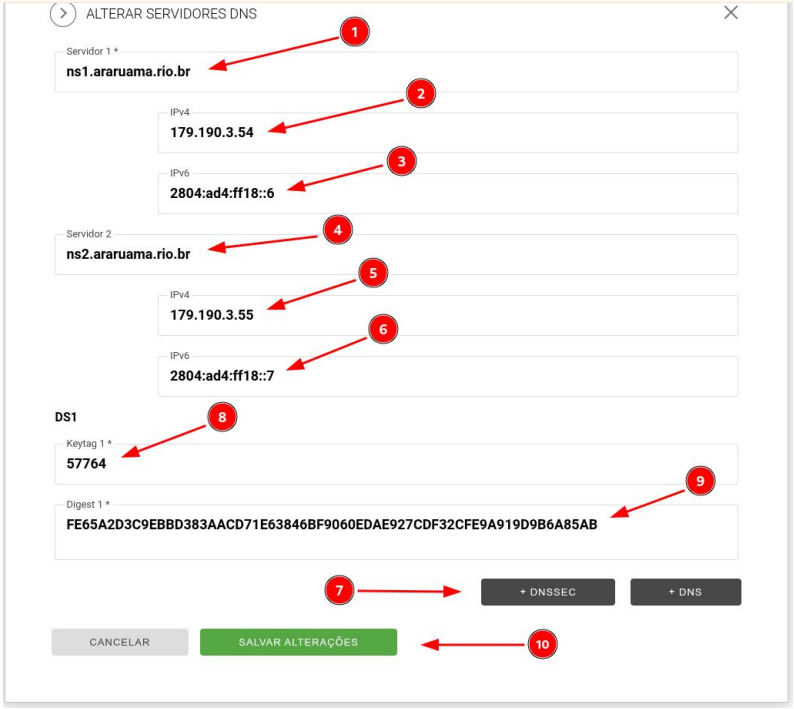
Entramos no site e selecionamos o nosso domínio de lab:



Rolamos tela abaixo até chegarmos na configuração de DNS e vamos clicar em **Alterar servidores DNS** dessa vez:



Na próxima tela configuramos nossos servidores primário e secundário tanto IPv4 quanto o IPv6 e o DNSSEC:



Um aviso para que tudo esteja OK antes de clicar em **CONFIRMAR**:

Nesta tela vemos que está tudo OK com as respostas dos servidores e temos que aguardar o tempo de publicação de aproximadamente 2h.

Se instalarmos o programa whois e checarmos veremos que os servidores ainda são os do **Registro.br**:

```
# apt install whois
# whois araruama.rio.br | egrep "nserver|ds"
nserver:     e.sec.dns.br
nserver:     f.sec.dns.br
```

Após a publicação dos nossos servidores de DNS Autoritativos veremos algo assim.

```
# whois araruama.rio.br | egrep "nserver|ds"
nserver:     ns1.araruama.rio.br 179.190.3.54 2804:ad4:ff18::6
nserver:     ns2.araruama.rio.br 179.190.3.55 2804:ad4:ff18::7
dsrecord:    57764 ECDSA-SHA-256
FE65A2D3C9EBBD383AACD71E63846BF9060EDAE927CDF32CFE9A919D9B6A85AB
dsstatus:    20250618 DSOK
dslastok:    20250618
```

Aqui finalizamos os nossos servidores de DNS mas antes ainda tenho um script para deixar. Alguém deve ter se perguntado sobre o quão trabalhoso é, assinar uma zona com os comandos abaixo e ainda criar a configuração para o domínio:

- ldns-keygen
- ldns-signzone

Por isso deixarei um script para facilitar este processo. Crie o **/root/scripts/sign.sh** no **DNS Primário** com o seguinte conteúdo do próximo slide. Vamos precisar alterar apenas a variável **DNS_SECUNDARIO** colocando o IP do nosso DNS Autoritativo Secundário e a **KEY** com o nome da chave.

```
#!/usr/bin/env bash
if [ ! -s /etc/nsd/zones/$1.zone ]; then
   echo "Falta arquivo de zona $1.zone"
   exit
fi
DNS_SECUNDARIO="2804:ad4:ff18::7"
KEY="serverkey"
cd /etc/nsd/zones
rm K$1* 2> /dev/null
rm $1.zone.signed 2> /dev/null
var1="`ldns-keygen -a ECDSAP256SHA256 -b 1024 $1`"
var2="`ldns-keygen -k -a ECDSAP256SHA256 -b 2048 $1`"
rm *.ds 2> /dev/null
ldns-signzone -b -o $1 /etc/nsd/zones/$1.zone /etc/nsd/zones/$var1 /etc/nsd/zones/$var2

cat << EOF > /etc/nsd/nsd.conf.d/$1.conf
zone:
    name: "$1"
    zonefile: "/etc/nsd/zones/$1.zone.signed"
    notify: $DNS_SECUNDARIO $KEY
    provide-xfr: $DNS_SECUNDARIO $KEY
EOF
nsd-control reconfig
nsd-control reload $1

dig DNSKEY $1. @localhost +multiline +norec

ldns-key2ds -n -2 $1.zone.signed
```

Após a criação do script façamos a alteração de permissões:

```
# chmod 700 /root/scripts/sign.sh
```

Antes de executar você só precisa ter o arquivo de zona daquele domínio por exemplo o **/etc/nsd/zones/araruama.rio.br.zone**:

```
$TTL        3600
@       IN      SOA     ns1.araruama.rio.br. contato.araruama.rio.br. (
                        2025061904      ; serial (ano + mes + dia + versao do dia)
                        28800             ; refresh (8 horas)
                        7200            ; retry (2 horas)
                        604800            ; expire (7 dias)
                        3600 )          ; minimum (1 hora)
              IN NS ns1.araruama.rio.br.
              IN NS ns2.araruama.rio.br.

ns1.araruama.rio.br.    IN   A       179.190.3.54
ns2.araruama.rio.br.    IN   A       179.190.3.55
ns1.araruama.rio.br.    IN   AAAA    2804:ad4:ff18::6
ns2.araruama.rio.br.    IN   AAAA    2804:ad4:ff18::7
araruama.rio.br.        IN   AAAA    2804:ad4:ff18::5
www.araruama.rio.br.    IN   CNAME   araruama.rio.br.
```

Na sequência só executar:

```
# /root/scripts/sign.sh araruama.rio.br
reconfig start, read /etc/nsd/nsd.conf
ok
ok

; <<>> DiG 9.18.33-1~deb12u2-Debian <<>> DNSKEY araruama.rio.br. @localhost +multiline +norec
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 29634
;; flags: qr aa; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;araruama.rio.br.IN DNSKEY

;; ANSWER SECTION:
araruama.rio.br. 3600 IN  DNSKEY 256 3 13 (
                                YgTDnI90GzRUxy7JDVIvI4kn5PkSeLam/62L0lahEVjG
                                cL2fcn3f+y/ThhUqp14YT9knu5jiKybtdOzn255qcQ==
                                ) ; ZSK; alg = ECDSAP256SHA256 ; key id = 12371
araruama.rio.br. 3600 IN  DNSKEY 257 3 13 (
                                ICN14lLQT5U6wzEo/rxEDYufDwr3+fQEYNFdU4ZgS2u7
                                dsQnPbglbRcx6EwG/9h5+OtWI8bikpZkMgDU3AlAbQ==
                                ) ; KSK; alg = ECDSAP256SHA256 ; key id = 60808

;; Query time: 0 msec
;; SERVER: ::1#53(localhost) (UDP)
;; WHEN: Wed Jun 18 15:39:13 -03 2025
;; MSG SIZE  rcvd: 204

araruama.rio.br. 3600     IN       DS       60808 13 2 097ab41652c12c10215afa10a161f689c8b06d6f41f007fb5a310b3a55804bdf
```

No **DNS Autoritativo Secundário** também teremos um script mas um pouco diferente chamado **/root/scripts/replica.sh**:

```
#!/usr/bin/env bash
DNS_PRIMARIO="2804:ad4:ff18::6"
KEY="serverkey"

cat << EOF > /etc/nsd/nsd.conf.d/$1.conf
zone:
    name: "$1"
    allow-notify: $DNS_PRIMARIO $KEY
    request-xfr: AXFR $DNS_PRIMARIO $KEY
EOF
nsd-control reconfig
nsd-control force_transfer $1

dig DNSKEY $1. @localhost +multiline +norec
```

Mudaremos a permissão e a execução se dá da mesma maneira que o script anterior:

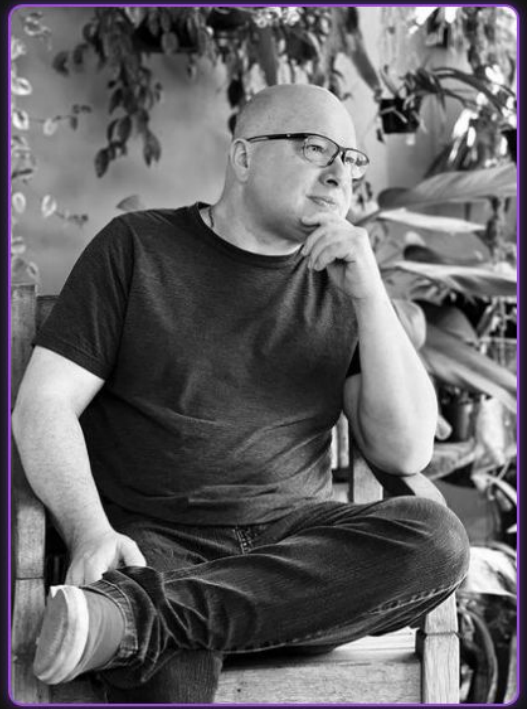
```
# chmod 700 /root/scripts/replica.sh
# /root/scripts/replica.sh araruama.rio.br
reconfig start, read /etc/nsd/nsd.conf
ok
ok

; <<>> DiG 9.18.33-1~deb12u2-Debian <<>> DNSKEY araruama.rio.br. @localhost +multiline +norec
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 44325
;; flags: qr aa; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;araruama.rio.br.IN DNSKEY

;; ANSWER SECTION:
araruama.rio.br. 3600 IN  DNSKEY 256 3 13 (
                                YgTDnI90GzRUxy7JDVIvI4kn5PkSeLam/62L0lahEVjG
                                cL2fcn3f+y/ThhUqp14YT9knu5jiKybtdOzn255qcQ==
                                ) ; ZSK; alg = ECDSAP256SHA256 ; key id = 12371
araruama.rio.br. 3600 IN  DNSKEY 257 3 13 (
                                ICN14lLQT5U6wzEo/rxEDYufDwr3+fQEYNFdU4ZgS2u7
                                dsQnPbglbRcx6EwG/9h5+OtWI8bikpZkMgDU3AlAbQ==
                                ) ; KSK; alg = ECDSAP256SHA256 ; key id = 60808

;; Query time: 0 msec
;; SERVER: ::1#53(localhost) (UDP)
;; WHEN: Wed Jun 18 15:58:10 -03 2025
;; MSG SIZE  rcvd: 204
```

Finalizamos nosso conteúdo sobre DNS Autoritativo utilizando o NSD. Para mais informações sugerimos também que vejam a **Semana da Capacitação 7 - Curso - Configurando o seu DNS de forma simples e segura**.

https://www.youtube.com/watch?v=JYLwzLh7BJc&list=PLQq8-9yVHyOapTWvl4OG_Lwe904xhkA5M

Por curiosidade se fizermos uma consulta da versão dos DNS(s) Root Servers teremos:

```
# for lista in `dig . NS @"a.root-servers.net" +short`; do echo "$lista --- versão: `dig @$lista version.bind chaos txt +short`"; done
l.root-servers.net. --- versão: "NSD 4"
j.root-servers.net. --- versão: "NSD"
f.root-servers.net. --- versão: "9.18.37"
h.root-servers.net. --- versão: "NSD 4.x"
d.root-servers.net. --- versão: "NSD 4"
b.root-servers.net. --- versão: "knot 3.x"
k.root-servers.net. --- versão: "Knot DNS"
i.root-servers.net. --- versão: "contact info@netnod.se"
m.root-servers.net. --- versão: "9.16"
e.root-servers.net. --- versão:
g.root-servers.net. --- versão: "I-AM-GROOT"     <==== Fã da Marvel? :D
c.root-servers.net. --- versão: "c-root"
a.root-servers.net. --- versão: "ATLAS"
```



Marcelo Gondim da Cunha

Especialista em redes e segurança, com experiência desde os anos 1990. Atuou como desenvolvedor, consultor de sistemas GNU/Linux e foi CTO da Nettel Telecom, onde implantou IPv6 em 2013. Contribui com o projeto MANRS. Também liderou o SOC da Brasil TecPar entre 2022 e 2025, focando em defesas contra DDoS, boas práticas e onde desenvolveu uma rede de DNS(s) Recursivos Anycast certificada pelo KINDNS.

- Administração de Sistemas Unix-Like desde 1996.
- Consultor na Conectiva S/A - Unidade Rio em 2000.
- Direção do AS53135 - Nettel Telecomunicações entre 2003 e 2021 atingindo a marca de 41.000 assinantes.
- Diversas palestras em eventos da área de Redes e Serviços e artigos técnicos publicados.