

# Tarefa Prática 2 (OpenSSL e Apache)

## Nomes:

Bruno Aurélio Rôzza de Moura Campos (14104255)

Caio Cargnin Cardoso (09138003)

## PARTE 1.OpenSSL

2. (Entregar) Gere a chave pública a partir da chave privada com os comandos abaixo (guardar a chave pública no arquivo seunome.publica.pem). Explique a saída obtida em cada um dos comandos. **Guarde o arquivo gerado e envie o arquivo da sua chave pública junto nas respostas da tarefa.**

a. Explique a saída obtida no seguinte comando:

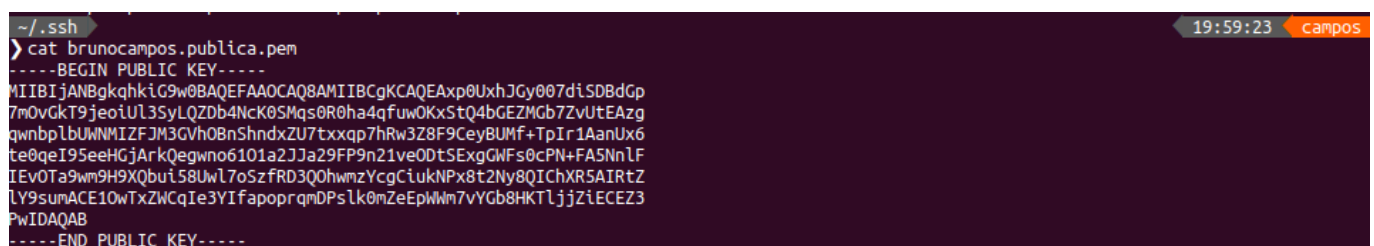
```
openssl rsa -in seunome.privada.pem -pubout -out seunome.publica.pem
```

## Resposta:

É necessário gerar a chave privada primeiro, então:

```
openssl genrsa -aes256-out brunocampos.privada.pem 2048
```

Em seguida o comando `openssl rsa -in seunome.privada.pem -pubout -out seunome.publica.pem` irá pegar a chave privada e criará a chave pública codificada em base64.



```
~/ssh  
cat brunocampos.publica.pem  
-----BEGIN PUBLIC KEY-----  
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxp0UxhJGy007diSDBdGp  
7m0vGkT9jeoiU13SylQZDb4NcK0SMqs0R0ha4qfuwOKxStQ4bGEZMgb7ZvUteAzg  
qwnbplbUWNMIZFJM3GVh0BnShndxZU7txxqp7hRw3Z8F9CeyBUMf+TpIr1AanUx6  
te0qeI95eeHGjArkQegwno6101a2JJJa29FP9n21ve0DtSExgGWFs0cPN+FA5NnLF  
IEv0Ta9wm9H9XQbui58UwL7oSzfRD3Q0hwmzYcgCiukNPx8t2Ny8QIchXR5AIRtZ  
LY9sumACE10wTxZWCqIe3YIfapoprqmDPsLk0mZeEpWm7vYVGb8HKTljZiECEZ3  
PwIDAQAB  
-----END PUBLIC KEY-----
```

- Nota: arquivo gerado `brunocampos.publica.pem` em anexo.

3. (Entregar) Digite o seguinte comando e depois abra o arquivo `seunome.publica.componentes`. Explique os componentes que constam nesse arquivo(<https://tools.ietf.org/html/rfc3447#appendix-A>).

Comando:

```
openssl rsa -in seunome.privada.pem -out seunome.publica.componentes -text -noout
```

**Resposta:**

Output do comando:

```

~/.ssh
> cat brunocampos publica.componentes
RSA Private-Key: (2048 bit, 2 primes)
modulus:
 00:c6:9d:14:c6:12:46:cb:4d:3b:76:24:83:05:d1:
 a9:ee:63:af:1a:44:fd:8d:ea:22:52:5d:d2:c8:b4:
 19:0d:be:0d:70:ad:12:32:ab:34:47:48:5a:e2:a7:
 ee:c0:e2:b1:4a:d4:38:6c:61:19:30:66:fb:66:f5:
 2d:10:0c:e0:ab:09:db:a6:56:d4:58:d3:08:64:52:
 4c:dc:65:61:38:19:d2:86:77:71:65:4e:ed:c7:1a:
 a9:ee:14:70:dd:9f:05:f4:27:b2:05:43:1f:f9:3a:
 48:af:50:1a:9d:4c:7a:b5:ed:2a:78:8f:79:79:e1:
 c6:8c:0a:e4:41:e8:30:9e:8e:b5:3b:56:b6:24:96:
 b6:f4:53:fd:9f:6d:6f:78:e0:ed:48:4c:60:19:61:
 6c:d1:c3:cd:f8:50:39:36:79:45:20:4b:ce:4d:af:
 70:9b:d1:fd:5d:06:ee:8b:9f:14:c2:5e:e8:4b:37:
 d1:0f:74:0e:87:09:b3:61:c8:02:8a:e9:0d:3f:1f:
 2d:d8:dc:bc:40:80:a1:5d:1e:40:21:1b:59:95:8f:
 6c:ba:60:02:13:53:b0:4f:16:56:0a:a2:1e:dd:82:
 1f:6a:9a:29:ae:a9:83:3e:c9:64:d2:66:5e:12:95:
 96:9b:bb:d8:19:bf:07:29:39:63:8d:98:84:08:46:
 77:3f
publicExponent: 65537 (0x10001)
privateExponent:
 64:a4:4a:57:88:01:59:99:7c:2d:04:99:64:04:77:
 20:76:60:cf:65:f0:39:ce:4f:af:ff:1d:05:58:c5:
 6d:42:45:db:37:c8:05:e6:dd:29:d5:cf:40:15:2a:
 95:91:09:97:ca:6c:00:f7:3e:e5:58:f9:c1:95:31:
 4d:75:c9:0e:c2:83:0c:09:e9:b6:4c:d1:6e:cc:89:
 68:10:f3:2f:93:5e:2b:87:30:ed:ce:0e:0d:1b:44:
 ca:80:8f:24:ae:25:3d:30:34:34:18:6c:86:44:f3:
 79:ea:94:61:ea:d4:2f:85:cb:44:a6:27:3f:0c:9a:
 28:72:88:71:a1:2f:a9:d7:ee:ae:41:fa:6e:49:b1:
 18:ae:39:54:49:ab:48:6a:59:8d:6f:a9:c8:7d:ff:
 a5:ab:c9:43:ab:9c:75:e3:fa:7b:8e:2a:bc:6d:15:
 49:12:42:e0:9f:80:11:38:0b:88:1a:70:41:02:8b:
 d9:f0:ca:89:be:8b:e9:7e:fa:9a:91:d1:2f:03:f8:
 5d:35:22:ba:bd:e5:10:c6:7c:8b:38:b9:47:f9:ec:
 e5:75:98:0d:99:91:95:9d:d9:14:da:88:1a:a2:95:
 b0:d2:ae:60:ea:c3:0f:4f:df:e6:b9:42:7c:87:b5:
 3b:38:cc:b6:84:5a:73:de:30:2e:5e:ff:fe:8e:c1:
 f9
prime1:
 00:fb:eb:69:f3:97:0d:73:55:9d:b7:c0:6a:cb:67:
 f7:ec:86:7b:46:96:6e:9b:d5:51:7f:c5:73:e3:8c:
 46:bc:d7:03:96:0e:ac:52:f1:36:16:c2:d9:d7:76:
 90:70:bf:8b:e7:a4:21:d4:dd:ce:0d:a5:e4:68:50:
 09:78:e7:76:ca:6c:a7:7f:0e:6d:3e:47:69:15:24:
 ee:4a:05:d2:44:75:17:b2:f0:3b:d0:b8:d8:80:1d:
 35:71:b1:97:d3:52:dc:dc:d8:fd:49:1b:7e:4f:ae:
 53:07:17:e0:59:fb:b0:34:0b:a0:b0:19:a5:56:09:
 fb:f5:2d:c0:8a:43:d8:12:0d
prime2:
 00:c9:d4:a2:37:fe:62:39:5b:53:cb:24:d2:20:13:
 7c:db:6a:e4:27:6c:b3:e5:7c:7a:54:ad:9f:a0:8f:
 28:cf:0a:71:6b:b6:81:c7:d7:fc:77:ff:79:55:06:
 35:92:61:45:a8:e8:7f:06:92:e9:9c:fc:3d:c5:6c:
 1d:66:9b:0c:5a:6b:80:20:ee:d3:1e:d9:12:9d:98:
 f9:65:76:37:43:83:40:f6:b7:75:93:ee:76:1b:55:
 61:b5:24:46:da:fa:21:c6:ea:d0:cd:46:77:2f:d1:
 39:5f:71:6b:8e:d3:53:ca:a7:9e:17:05:f5:60:4d:
 3f:a2:ae:82:95:3d:a9:37:7b
exponent1:
 00:83:de:ad:6d:a7:8a:90:ef:26:4a:43:dd:23:70:
 df:24:df:18:b4:d0:96:41:d8:9e:7a:e5:df:4e:23:
 e5:fb:80:0a:0e:88:cb:c7:f7:20:3b:35:f0:56:8b:
 67:fc:bd:27:fd:2f:bb:cd:f3:f5:a2:cb:4e:0f:14:
 a6:80:b5:99:47:49:2d:3c:a3:4e:a8:25:35:6a:ae:

```

```

14:56:87:49:94:30:3f:21:9a:03:95:b4:cd:0f:f3:
3a:40:b3:98:28:34:de:0c:75:41:d0:fd:25:57:8c:
87:45:d5:47:c9:92:a8:9f:f5:de:3d:90:8f:2d:c4:
b8:31:95:ab:8e:35:09:6c:19
exponent2:
00:ad:25:94:b1:12:b3:f3:5e:cd:09:06:a2:99:4f:
fe:9a:42:1e:4f:50:2b:18:e1:ec:14:7f:0a:e5:74:
4b:5f:2b:27:58:6f:ae:f0:e1:f3:3e:82:d5:f5:42:
29:6d:55:b3:ac:0f:21:02:63:c0:b4:a3:94:de:ac:
3c:a0:cc:bd:11:49:0e:17:b2:ab:3d:d8:9d:e6:c3:
d8:98:d9:8c:d9:87:5d:91:0a:9c:7c:f7:63:2d:59:
d7:43:ce:46:57:0f:a5:30:80:3b:f7:0e:cf:ab:1f:
03:e8:44:66:30:96:4d:59:1a:e9:3d:f9:27:a3:a1:
41:c9:6f:8a:a9:3b:c7:d2:c7
coefficient:
44:88:69:5c:77:e7:42:12:54:d8:93:bc:c9:60:7c:
6f:7a:d8:6c:36:d6:7e:37:13:68:2b:68:1d:55:d1:
3b:66:cf:c1:50:21:ea:51:db:9d:03:88:69:d9:ea:
fc:b6:72:91:a7:1c:ff:17:3b:d9:23:a3:c5:b4:ef:
24:05:fb:64:dd:88:ae:b3:98:07:cf:2f:f8:29:45:
6f:41:c2:27:3c:02:d1:da:63:69:10:95:17:c2:08:
1e:89:05:55:f9:c7:36:ad:56:81:1d:6d:0f:e7:a3:
7e:ea:9a:09:20:d9:25:38:e1:8f:3f:86:11:ab:ac:
d3:a9:61:c2:49:d3:7a:70

```

Segundo o site <https://tools.ietf.org/html/rfc3447#appendix-A>, uma chave privada RSA deve ser representada com o tipo ASN.1 RSAPrivateKey:

```

RSAPrivateKey ::= SEQUENCE {
    versão versão,
    modulo INTEGER, - n
    publicExponent INTEGER, - e
    privateExponent INTEGER, - d
    prime1 INTEGER, - p
    prime2 INTEGER, - q
    exponent1 INTEGER, - d mod (p-1)
    exponent2 INTEGER, - d mod (q-1)
    coeficiente INTEGER, - (inverso de q) mod p
    otherPrimeInfos OtherPrimeInfos OPCIONAL
}

```

Já os campos do tipo RSAPrivateKey têm os seguintes significados:

- **modulus** é o módulo RSA n.
- **publicExponent** é o expoente público da RSA e.
- **privateExponent** é o expoente privado da RSA d.
- **prime1** é o fator primo p de n.
- **prime2** é o fator primo q de n.
- **expoent1** é d mod (p - 1).
- **expoent2** é d mod (q - 1).
- **coeficient** é o coeficiente CRT  $q^{-1} \bmod p$ .

Nota: arquivo gerado **brunocampos publica.componentes** em anexo.

## PARTE 2. Assinatura Digital



```
openssl rand -out chaveSecretaNomeAluno.bin -base64 128
```

**Resposta:**

```
~/ssh
> cat chave_secreta_brunocampos.bin
QzaMyJLIv9J4ILOSfersaFtYBH2dhqQtW6+lhcnRLXSHFEY2h/KhBbvj4kYCJRQi
s4CG/8QN2XqhZvbFwf9Nu7smdVSSJehDqwq4IX36riQM61RuKURhupPK3FatfHI+
35K/bxw1opn/BiXawiFQ9TlPeNRctJ/SyNP+5S+KduA=
```

- Nota: arquivo `chave_secreta_brunocampos.bin` em anexo.

7. (Entregar) Cifrar o arquivo `msgPlana.txt` com a chave secreta criada na questão anterior:

```
openssl enc -aes-128-ctr \
    -in msgPlana.txt \
    -out msgCifrada \
    -pass file:./chaveSecretaNomeAluno.bin
```

**Resposta:**

```
~/ssh 21:24:48 campos
> cat msgCifrada
salted__i'v>E+s++%v++B]p5e0++9++*+++++)++;+pm@+9:++G].0+
!+++v!++e++Ss++N++]l++m++W++;z$S
he}+++7+_B +xt++m++2-80+:2t++
weyGQ++\++++ ++=++q++)H'++J++U=++++q{+F++L+g++++(K+z++%++++D+§9++%I+h++Z!+ +,Z0nZ+FK
++Z`q>h0++
++B`+K+++S+p++++t)++6++++++v!++#|++++P++?5we++&E++s++i++11R{++++3+_!+F++t++X++N++ ^S3+0+5+.gS+@oxjiwn+5++++!+'l+Zb+'G++D>+Q++|f+
[Pq++++P+AG++S++++p++++S++0+h++&+++S
=X-X+Xl+HqY++8++'++G8j+xfvd,=llV+++P-CC+lg.Y+2++++++E1+:+&1XJX++++ ~/ssh 21:27:09 campos
```

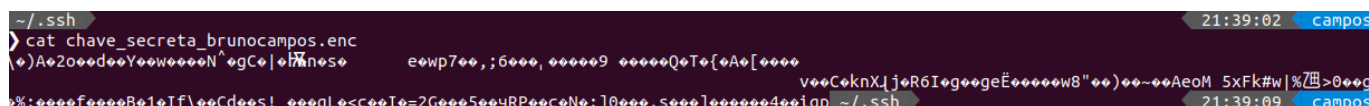
- Nota: arquivo `msgCifrada` em anexo.

8. (Entregar) Cifrar a sua chaveSecreta (`chaveSecretaNomeAluno.bin`) usando o meu certificado (certificado de Carla):

```
openssl rsautl -encrypt -oaep \
    -inkey certificadoCarla.crt \
    -certin \
    -in chaveSecretaNomeAluno.bin \
    -out chaveSecretaNomeAlunoCifrada.enc
```

**Resposta:**

```
openssl rsautl -encrypt -oaep \
    -inkey
../projetos/seguranca/trabalho_02_sem_implementacao/certificadoCarla.crt \
    -certin \
    -in chave_secreta_brunocampos.bin \
    -out chave_secreta_brunocampos.enc
```



```

~/.ssh
> cat chave_secreta_brunocampos.enc
A2o0d0Y0w000N^0gC0|0h0n0s0      e0wp700,;6000,00009 0000Q0T0{0A0[0000
v00C0knX|j0R6I0g00geE0000w8"00)00~00AeoM 5xFk#w|0%70>000g
0%:0000f0000B0!0If\00Cd00s! 000qL0<c00I0=2G0005004RP00c0N0:|0000.s000]00000400iqp ~/.ssh

```

- Nota: arquivo `chave_secreta_brunocampos.enc` em anexo.

9. (Entregar) Explique o que foi feito nas questões 6, 7 e 8. Explique também como será feito o processo de decifragem.

**Observação:** vou verificar com os comandos abaixo!!

```

openssl rsautl -decrypt -oaep \
    -inkey chavePrivadaCarla.key \
    -in chaveSecretaNomeAlunoCifrada.enc \
    -out chaveSecretaNomeAlunoDecifrada.bin

```

```

openssl enc -aes-128-ctr -d \
    -in msgCifrada \
    -pass file:./chaveSecretaAlunoDecifrada.bin

```

**Resposta:**

Na questão 6 foi gerado uma chave secreta usando o parâmetro `rand` que gera um pseudo-random em bytes e lançado esse valor no arquivo `chave_secreta_brunocampos.bin` com encoding na base64.

Na questão 7, foi cifrado o arquivo `msgPlana.txt` com a chave secreta (Passphrase source) `chave_secreta_brunocampos.bin` e lançado essa cifragem no arquivo `msgCifrada`.

Por fim, na questão 8 foi cifrado a chave secreta `chave_secreta_brunocampos.bin` com o certificado `certificadoCarla.crt` e lançado no arquivo `chave_secreta_brunocampos.enc`.

Decifragem: De forma inversa, para decifrar é necessário informar a chave privada do certificado, o certificado e qual o arquivo de saída.

## GERAR SEU CERTIFICADO NA ICPEDU

10. (Entregar)

- Acessar o site <https://p1.icpedu.rnp.br/default/public/default> e gerar o seu certificado digital pessoal.
- Clique em "Emitir".
- Logue pela Federação Café na UFSC.
- Depois de autenticar com o email e senha do idufsc, você obterá a tela da figura 1.
- Coloque uma senha para proteger o arquivo PKCS12 que será gerado.
- NOTA: Documento com screenshots o processo. Depois de emitir, você obterá a tela da figura 2.

**Resposta:**



**icpedu** Infraestrutura de Chaves Públicas para Ensino e Pesquisa | **eduID**

Ajuda | Área Restrita | Idioma

Página inicial | Certificado | Verificação de Atributos | Repositório | O eduID | Fale Conosco

Home > Emitir Certificado

## Emitir Certificado

Abaixo encontram-se os dados cadastrados na sua instituição. Para emitir um certificado, escolha uma senha e clique em "submeter".

Nome: Bruno Aurelio Rozza de Moura Campos

E-mail: moura.campos@grad.ufsc.br

Página inicial | Certificado | Verificação de Atributos | Repositório | O eduID | Fale Conosco

Home > Baixar certificado

## Baixar certificado

✓ Seu certificado foi emitido com sucesso.

DOWNLOAD DO CERTIFICADO

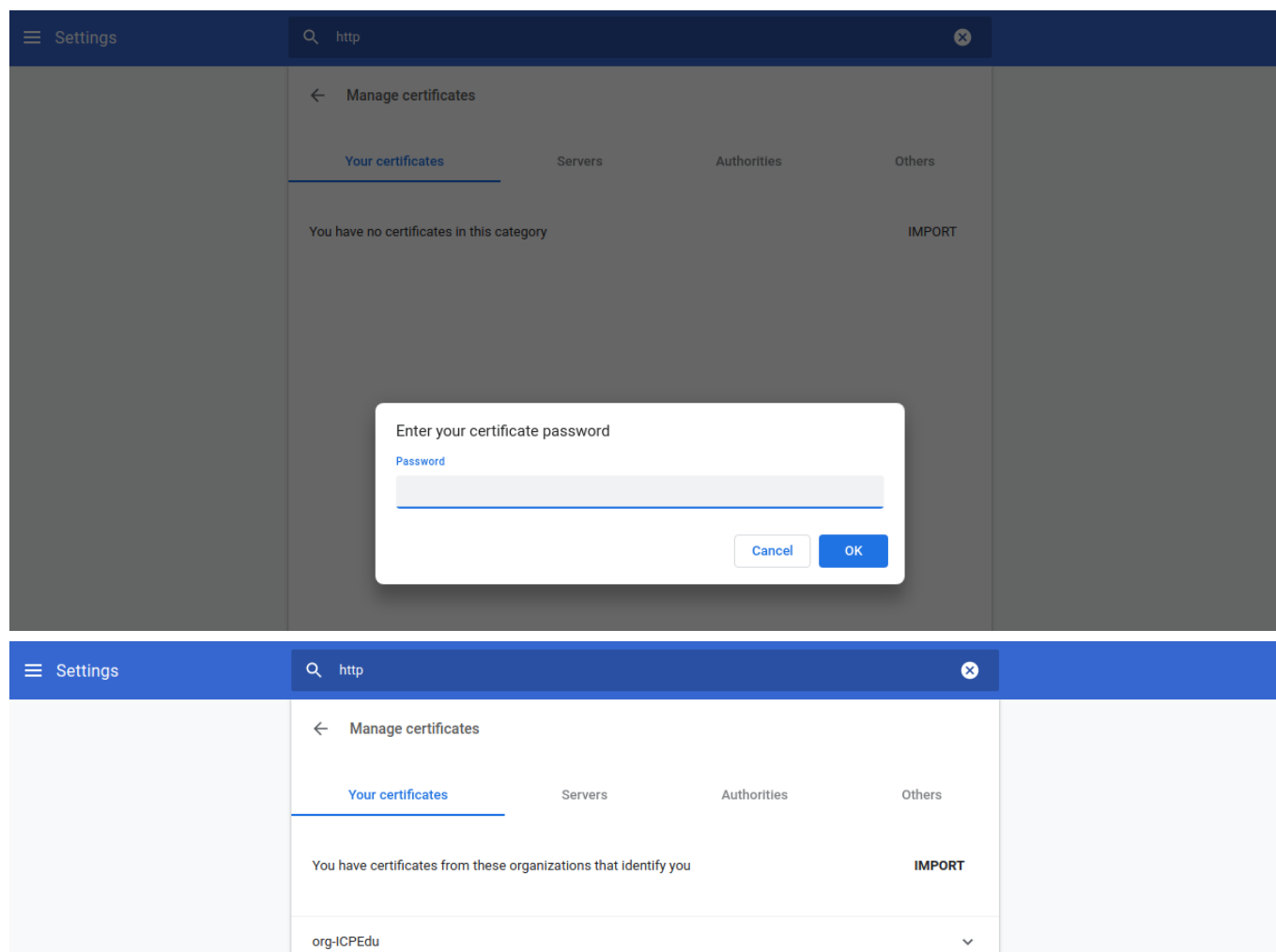
SAEC P1 - Br....p12

- Nota: arquivo **SAEC P1 - Bruno Aurelio Rozza de Moura.p12** em anexo.



11. (Entregar) Agora, clique em <https://p1.icpedu.rnp.br/index/howto> e instale o seu certificado no navegador. Documente com screenshots.

**Resposta:**



12. (Entregar) Explique o formato deste certificado:

a) Qual é o formato?

**Resposta:**

É uma arquivo com extensão **.p12**

b. Onde ficam a chave pública e a chave privada?

**Resposta:**

Ficam no arquivo **'SAEC P1 - Bruno Aurelio Rozza de Moura.p12'**. É possível extrair a chave privada do arquivo com os comandos:

**Private**

```
openssl pkcs12 -in
../projetos/seguranca/trabalho_02_sem_implementacao/'SAEC P1 - Bruno
Aurelio Rozza de Moura.p12' \
```

```
-nocerts -nodes \  
| openssl rsa > id_rsa_trab_02
```

- Nota: arquivos `id_rsa_trab_02` em anexo.

### c. Quem é a autoridade certificadora que assinou o certificado?

#### Resposta:

AC Raiz da ICPEDU V2

14. (Entregar) Agora o certificado X.509 AUTO-ASSINADO será efetivamente criado (assinado por você mesmo, usando a SUA chave privada), usando o comando:

```
openssl x509 -req -days 90 -sha512 \  
-in certificado.csr \  
-signkey seunome.privada.pem \  
-out certificado.crt
```

#### Resposta:

```
openssl x509 -req -days 90 -sha512 \  
-in \  
../projetos/seguranca/trabalho_02_sem_implementacao/'SAEC P1 - Bruno  
Aurelio Rozza de Moura.p12' \  
-signkey brunocampos.privada.pem \  
-out \  
../projetos/seguranca/trabalho_02_sem_implementacao/certificado_brunocampos  
.crt
```



- Nota: arquivos `certificado_brunocampos.crt` em anexo.

## Apache2

16-35

← → ↺ 🏠

Not secure | https://localhost

🔍 ☆ ABP

Apps N 25 📞 🔊

UFSC data\_science DevOps data\_structure cursos de TI jobs études SI



# ubuntu

## Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in [/usr/share/doc/apache2/README.Debian.gz](#)**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/` counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf`. See their respective man pages for detailed information.
- The binary is called `apache2`. Due to the use of environment variables, in the default configuration, `apache2` needs to be started/stopped with `/etc/init.d/apache2` or `apache2ctl`. **Calling `/usr/bin/apache2` directly will not work** with the default configuration.

Document Roots

By default, Ubuntu does not allow access through the web browser to *any* file apart of those located in `/var/www`, `public_html` directories (when enabled) and `/usr/share` (for web applications). If your site is using a web document