# Practica 5: Mensajería de última milla II

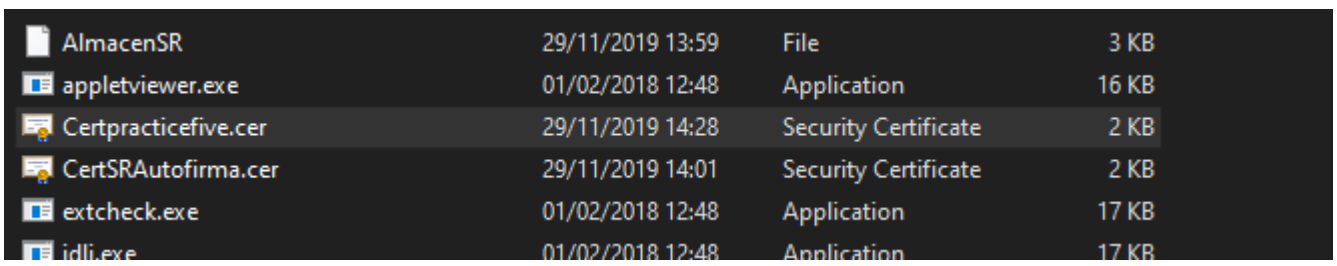
## Parte 3: Aplicar funcionalidad SSL

1. A continuación, se muestra la creación del certificado de seguridad para la practica 5

```
C:\WINDOWS\system32>cd C:\Program Files\Java\jdk1.8.0_161\bin

C:\Program Files\Java\jdk1.8.0_161\bin>keytool -genkey -alias practicefive -keyalg RSA -validity 365 -keystore practicefive -
keypass 123456 -storepass 123456
What is your first and last name?
  [Unknown]:  Alvaro Suarez
What is the name of your organizational unit?
  [Unknown]:  axxa
What is the name of your organization?
  [Unknown]:  axxa
What is the name of your City or Locality?
  [Unknown]:  Madrid
What is the name of your State or Province?
  [Unknown]:  Madrid
What is the two-letter country code for this unit?
  [Unknown]:  ES
Is CN=Alvaro Suarez, OU=axxa, O=axxa, L=Madrid, ST=Madrid, C=ES correct?
  [no]:  yes


Warning:
The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS12 which is an industry standard format using "keytoo
l -importkeystore -srckeystore practicefive -destkeystore practicefive -deststoretype pkcs12".

C:\Program Files\Java\jdk1.8.0_161\bin>keytool -list -v -keystore practicefive
Enter keystore password:
Keystore type: JKS
Keystore provider: SUN

Your keystore contains 1 entry

Alias name: practicefive
Creation date: 29-nov-2019
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=Alvaro Suarez, OU=axxa, O=axxa, L=Madrid, ST=Madrid, C=ES
Issuer: CN=Alvaro Suarez, OU=axxa, O=axxa, L=Madrid, ST=Madrid, C=ES
Serial number: 2886f6e5
Valid from: Fri Nov 29 14:23:23 CET 2019 until: Sat Nov 28 14:23:23 CET 2020
Certificate fingerprints:
         MD5:  62:EC:58:86:F9:60:EF:94:E1:1D:BB:B1:F1:B2:6C:43
         SHA1: 73:F5:12:CD:4B:75:63:22:1B:A1:30:4A:3F:DC:5C:B5:D6:AE:61:5B
         SHA256: 1B:1C:9E:DD:6D:BD:B7:7D:6F:CA:35:AD:E5:ED:6B:06:8A:03:3C:F6:CF:E0:A6:2D:E0:50:BC:E9:39:CD:27:8E
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 2048-bit RSA key
Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 35 4B FB 64 1F 20 E0 5E   6A 4B A5 28 8D 3D 5C DB  5K.d. .^jK.(.=\.
0010: D1 E4 BC B9                                        ....
]
]



*******************************************
*******************************************



Warning:
The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS12 which is an industry standard format using "keytoo
l -importkeystore -srckeystore practicefive -destkeystore practicefive -deststoretype pkcs12".

C:\Program Files\Java\jdk1.8.0_161\bin>keytool -list -v -keystore practicefive
```

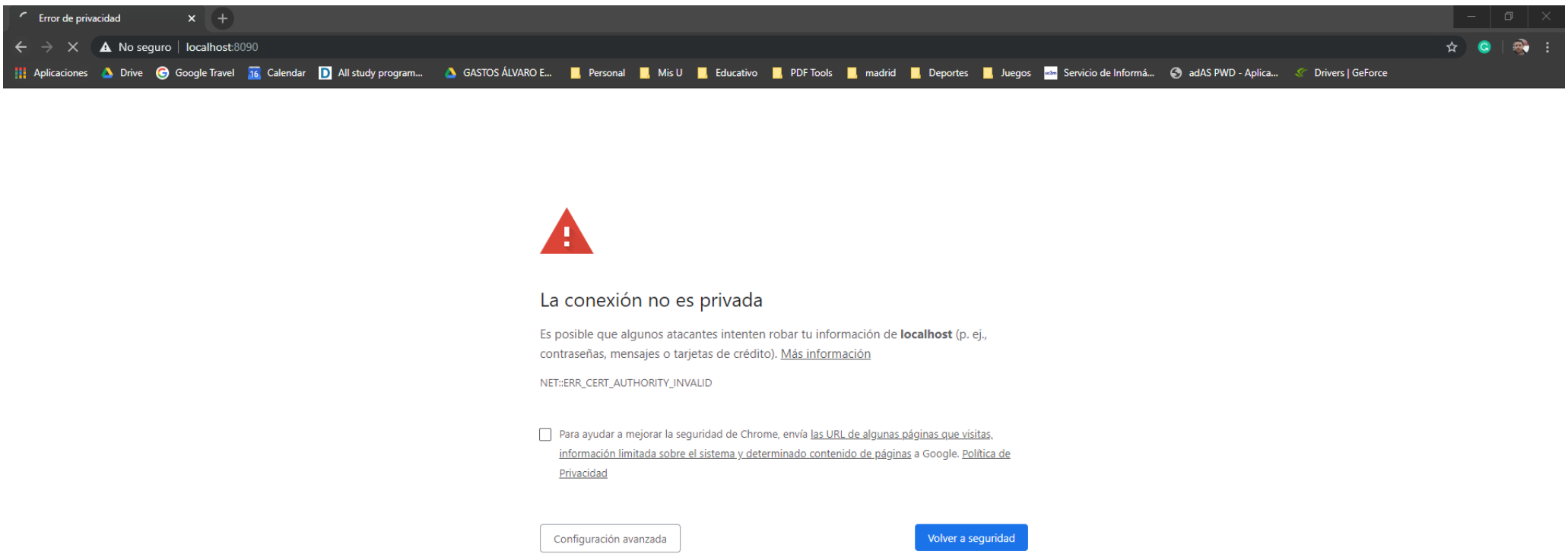
```
Enter keystore password:
Keystore type: JKS
Keystore provider: SUN

Your keystore contains 1 entry

Alias name: practicefive
Creation date: 29-nov-2019
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=Alvaro Suarez, OU=axxa, O=axxa, L=Madrid, ST=Madrid, C=ES
Issuer: CN=Alvaro Suarez, OU=axxa, O=axxa, L=Madrid, ST=Madrid, C=ES
Serial number: 2886f6e5
Valid from: Fri Nov 29 14:23:23 CET 2019 until: Sat Nov 28 14:23:23 CET 2020
Certificate fingerprints:
         MD5:  62:EC:58:86:F9:60:EF:94:E1:1D:BB:B1:F1:B2:6C:43
         SHA1: 73:F5:12:CD:4B:75:63:22:1B:A1:30:4A:3F:DC:5C:B5:D6:AE:61:5B
         SHA256: 1B:1C:9E:DD:6D:BD:B7:7D:6F:CA:35:AD:E5:ED:6B:06:8A:03:3C:F6:CF:E0:A6:2D:E0:50:BC:E9:39:CD:27:8E
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 2048-bit RSA key
Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 35 4B FB 64 1F 20 E0 5E   6A 4B A5 28 8D 3D 5C DB  5K.d. .^jK.(.=\.
0010: D1 E4 BC B9                                        ....
]
]



*******************************************
*******************************************



Warning:
The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS12 which is an industry standard format using "keytoo
l -importkeystore -srckeystore practicefive -destkeystore practicefive -deststoretype pkcs12".

C:\Program Files\Java\jdk1.8.0_161\bin>keytool -export -alias practicefive -keystore practicefive -rfc -file Certpracticefive.cer
Enter keystore password:
Certificate stored in file <Certpracticefive.cer>

Warning:
The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS12 which is an industry standard format using "keytoo
l -importkeystore -srckeystore practicefive -destkeystore practicefive -deststoretype pkcs12".

C:\Program Files\Java\jdk1.8.0_161\bin>
```

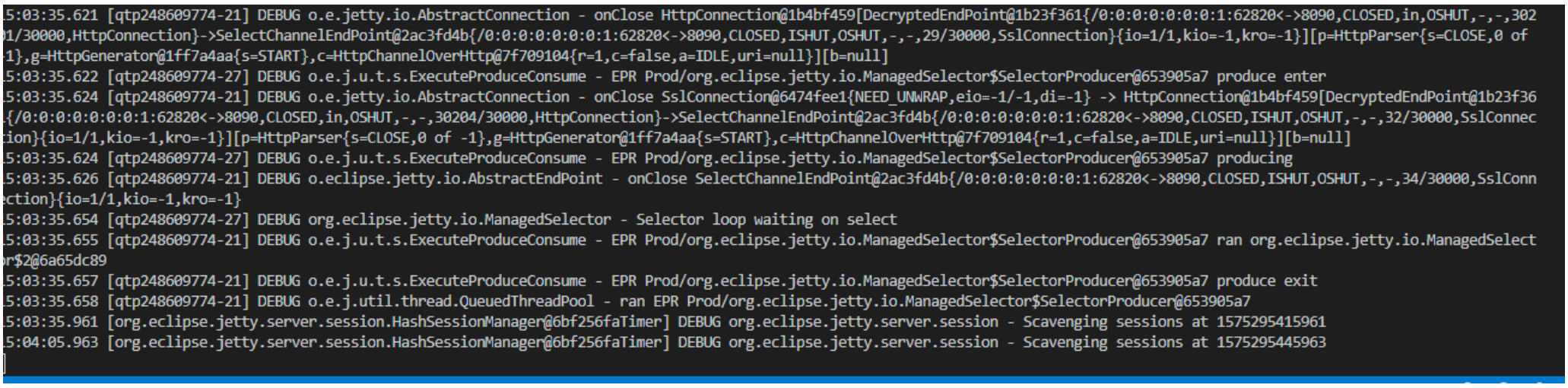
Y el certificado esta creado en el path de keytool

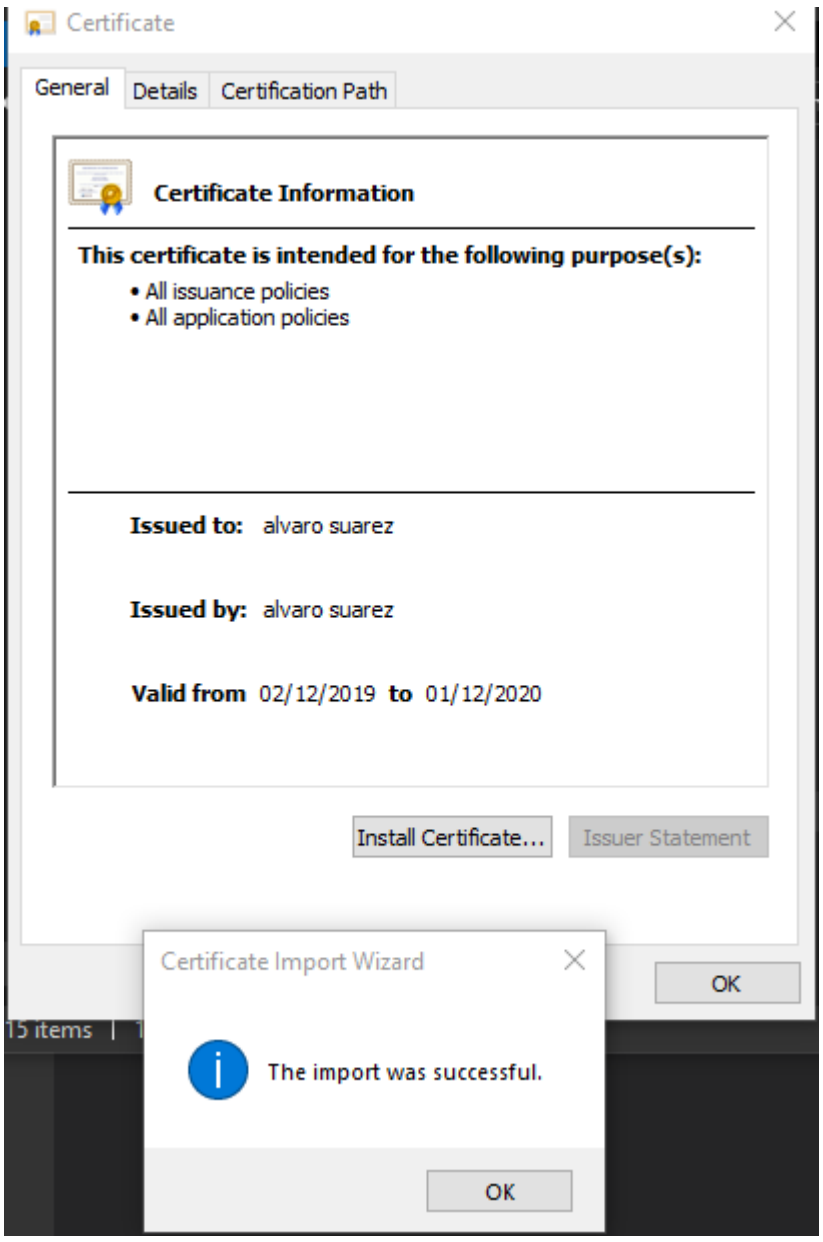


| | | | |
|---|---|---|---|
| AlmacenSR | 29/11/2019 13:59 | File | 3 KB |
| appletviewer.exe | 01/02/2018 12:48 | Application | 16 KB |
| Certpracticefive.cer | 29/11/2019 14:28 | Security Certificate | 2 KB |
| CertSRAutofirma.cer | 29/11/2019 14:01 | Security Certificate | 2 KB |
| extcheck.exe | 01/02/2018 12:48 | Application | 17 KB |
| idlj.exe | 01/02/2018 12:48 | Application | 17 KB |

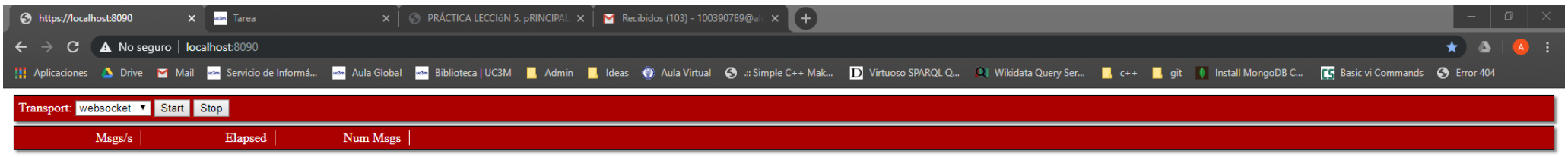
2. Certificado instalado en la aplicación



Después de importado el certificado el cliente abre sin problema en el navegador




3. Medición y comparación con tiempos en http

La siguiente es la medición tomada del cliente

| SleepTime | Msg/s | | Time | Msgs | | Transport |
|---|---|---|---|---|---|---|
| | http | https | | http | https | |
| 0 | 130 | 67 | 10 | 1305 | 667 | webSocket |
| 0 | 18 | 6 | 10 | 176 | 64 | long-polling |
| 1 | 145 | 71 | 10 | 1447 | 714 | webSocket |
| 1 | 21 | 9 | 10 | 206 | 91 | long-polling |
| 10 | 90 | 72 | 10 | 901 | 723 | webSocket |
| 10 | 17 | 9 | 10 | 174 | 87 | long-polling |

Como conclusión el TRADE off de usar una comunicación segura es el rendimiento, tanto con websocket como con long-polling la relación en rendimiento es de 2:1, es decir, si con http puedo enviar dos mensajes, con https puedo enviar un mensaje.

## Parte 4: Cifrado y Descifrado simétrico/asimétrico

En modo de cifrado simétrico se envia “Hello world!”:

LauncherCryptoAES.java

```java
        // Create a new instance of AESCrypto
        final AESCrypto aesCrypto = AESCrypto.createNewInstance() ;

        // Encode directly to the array
        final byte[] msgEncoded   = aesCrypto.encode(msg.getBytes()) ;

        // Printout the encoded message
        System.out.println("Message encoded: " + new String(msgEncoded)) ;

        // Decode the encoded message
        final byte[] msgDecoded   = aesCrypto.decode(msgEncoded) ;

        // Printout the decoded message
        System.out.println("Message decoded: " + new String(msgDecoded)) ;
    }

    /**
     * @param args with the input arguments
     * @throws Lesson5Exception with an occurred exception
     */
    public static void main(String[] args) throws Lesson5Exception
    {
        final LauncherCryptoAES launcherCryptoAES = new LauncherCryptoAES() ;

        launcherCryptoAES.doExample() ;
    }
}
```

Markers | Properties | Servers | Data Source Explorer | Snippets | Console | Progress | Debug

```
<terminated> LauncherCryptoAES [Java Application] C:\Program Files\Java\jdk1.8.0_161\bin\javaw.exe (2 dic. 2019 17:01:29)
Message to be encoded: Hello world!
Message encoded: ♦5♦x♦♦h♦@♦♦E♦T
Message decoded: Hello world!
```

En modo de cifrado asimétrico se envia “Hello world!”:

LauncherCryptoRSA.java

```java
package com.cnebrera.uc3.tech.lesson5;

import com.cnebrera.uc3.tech.lesson5.util.Lesson5Exception;

/**
 * Launcher class - Crypto - AES
 * -------------------------------------
 * @author Francisco Manuel Benitez Chico
 * -------------------------------------
 */
public class LauncherCryptoRSA
{
    /**
     * @throws Lesson5Exception with an occurred exception
     *
     */
    private void doExample() throws Lesson5Exception
    {
        // Message to encode/decode
        final String msg           = "Hello world!" ;

        // Printout the message
        System.out.println("Message to be encoded: " + msg) ;

        // Create a new instance of RSACrypto
        final RSACrypto rsaCrypto  = new RSACrypto() ;

        final byte[] msgEncoded    = rsaCrypto.encodeWithPubKey(msg.getBytes()) ;

```

Markers | Properties | Servers | Data Source Explorer | Snippets | Console | Progress | Debug

```
<terminated> LauncherCryptoRSA [Java Application] C:\Program Files\Java\jdk1.8.0_161\bin\javaw.exe (2 dic. 2019 17:13:56)
Message to be encoded: Hello world!
Message encoded: yp0◆◆◆◆◆◆◆6◆◆◆Q◆◆◆◆u◆◆00k◆b◆◆3◆◆◆◆◆◆◆◆00S-◆◆Bf._[◆◆:)*◆◆◆>'0◆◆◆0◆l◆P◆"◆◆j◆◆\b00c◆q◆◆x◆_%0>◆i◆S◆T/00◆+◆]◆Л0d e◆0◆q◆◆◆P◆
Message decoded: Hello world!
```