Universitat Oberta de Catalunya

Memòria

# Estudi, disseny i implementació d'un *Sports Tracker Segur*

**PFC - *Àrea de seguretat informàtica***

**Segon cicle d'enginyeria informàtica**

*Alumne:* Javier Díaz Espejo

*Consultora:* Cristina Pérez Solà

*Data:* 04 de gener de 2016

JAVIER DIAZ ESPEJO

Firmado digitalmente por JAVIER DIAZ ESPEJO
Fecha: 2016.01.26 17:19:31 +01'00'

**Dedicatòria i agraïments:**

*A la Olga, als meus pares i la resta de família pel seu suport constant en aquesta última etapa d'un llarg període d'estudis.*

*A la Cristina com a consultora que m'ha guiat durant tot el projecte i facilitat la realització d'aspectes claus d'aquest.*

# Resum

El projecte tracta sobre el desenvolupament d'una aplicació web segura per gestionar activitats esportives anomenada *Sports Tracker Segur*. En l'actualitat la pràctica d'esports és un hàbit molt comú per moltes persones que volen mantenir un estil de vida saludable. La societat actual compta amb una gran diversitat de dispositius tecnològics que en els últims anys han experimentat un creixement exponencial. Dispositius com els telèfons i els rellotges intel·ligents s´han convertit en una eina de gran ajuda alhora de monitoritzar la practica d'esports.

En el mercat actual existeixen un gran ventall d'aplicacions comercials pel seguiment d'activitats esportives, tant per a dispositius mòbils com per a ordinadors. Entre els desenvolupadors d'aquestes aplicacions ens trobem amb els mateixos fabricants de dispositius específics de monitoratge d'esports, com són els rellotges pulsòmetres que existeixen des de fa temps. Aquestes aplicacions treballen amb dades molt sensibles dels usuaris, ja que poden contenir dades sobre la salut, fisiològiques i personals dels usuaris. Una pèrdua d'informació per part d'una aplicació d'aquest tipus de dades pot generar un gran perjudici a una persona.

L'aplicació desenvolupada en el projecte té com a objectiu principal l'assegurament de totes les dades personals que els usuaris desen en l'aplicació. S'ha realitzat un estudi previ dels possibles punts febles del sistema a través d'analitzar una varietat d'atacs que podria sofrir una aplicació similar com la desenvolupada. Els protocols criptogràfics és l'eina bàsica utilitzada per poder assegurar la confidencialitat de les dades dels usuaris, tant a nivell de connexió com d'emmagatzematge de les dades. Les dades que disposa l'aplicació estan protegides tant per un atacant aliè al sistema com per un possible atacant intern d'aquest.

# Paraules clau

*Sports Tracker*, criptografia, aplicació web, activitats esportives, seguretat, confidencialitat, connexions segures, xifratge, privacitat i dades sensibles.

# Àrea del projecte

Seguretat informàtica.

# Abstract

The project deals with the development of a secure web application to manage sports called *Sports Tracker Segur*. Currently playing sports is a very common habit for many people who want to maintain a healthy lifestyle. Today's society has a wide variety of technological devices in recent years has grown exponentially. Devices like smart phones and watches have become a helpful tool when monitoring the practice of sports.

In today's market there are a wide range of commercial applications for tracking sports activities for both mobile devices to computers. Among the developers of these applications we find the same manufacturers of specific sports monitoring devices such as heart rate monitors watches that have long existed. These applications work with highly sensitive data of users, as they may contain data on the health, physiological and personal users. A loss of information from an application of this type of data may generate a great disservice to a person.

The application developed in the project's main objective is the assurance of all personal data that users stored in the application. We carried out a preliminary study of possible weaknesses in the system by analyzing a variety of attacks that could suffer a similar application as developed. Cryptographic protocols are the basic tool used to ensure the confidentiality of data users, both in terms of connection and data storage. The data available to the application are protected for both an outside attacker to the system as a potential attacker inside it.



# Key words

Sports Tracker, cryptography, web application, sports, security, confidentiality, secure connections, encryption, privacy and sensitive data.


# Project area

Information security.

# Índex de contingut

# Índex d'il·lustracions

# Índex de taules

# Capítol 1. Introducció

## 1.1 Justificació i context

En l'actualitat Internet experimenta un creixement exponencial respecte al volum de dades personals que els usuaris dipositen en aquesta gran xarxa. L'aparició de dispositius mòbils junt tot un ventall d'aplicacions per aquests, les xarxes socials, el comerç electrònic, l'emmagatzematge de dades en el núvol... Tot aquest conjunt de tecnologies basats en la xarxa d'Internet han modificat el comportament i l'ús de les persones en moltes activitats quotidianes. Així, tasques quotidianes de les persones passen per interactuar amb les noves tecnologies, i per tant, que aquestes tractin i emmagatzemin dades personals dels usuaris; com per exemple anar al metge, realitzar la compra, fer esports, comunicar-se amb els amics,...

El desenvolupament d'aquest projecte pretén focalitzar-se en la seguretat de les dades de les aplicacions que els usuaris utilitzen per la realització d'esports. La pràctica d'esports en l'actualitat està en plena expansió per moltes persones, i amb el sorgiments dels dispositius mòbils (*smartphones*, *smartwatches*,...), ràpidament han sorgit les aplicacions per gestionar i compartir tota una sèrie de dades i resultats dels esportistes. Aquestes aplicacions tracten dades sobre aspectes fisiològics i de salut de les persones, per tant manipulen dades que són altament sensibles. S'ha de posar molta atenció a l'hora de donar accés a aquesta informació a través de la xarxa, aplicant-hi les mesures de seguretat necessàries per impedir que terceres persones puguin accedir a les dades personals d'un usuari.

Al mercat actual existeixen un ventall d'aplicacions per cobrir les necessitats dels esportistes. Entre les principals tenim les de les grans marques (Polar, Garmin,...) fabricants de dispositius mòbils per esports, com els rellotges pulsòmetres, i altres dispositius específics per a una gran diversitat d'esports. El desenvolupament del projecte ve enfocat en poder assegurar que una aplicació d' *sports tracker* sigui capaç de garantir la seguretat de les dades que gestiona, permetent a l'usuari tindre la suficient confiança que certes dades personals mai arribaran a mans de tercers.

## 1.2 Objectius

L'objectiu principal d'aquest projecte és el disseny i implementació d'un *sports tracker* segur, a través d'una aplicació web, que ofereixi un seguit de característiques per assegurar la total seguretat i confidencialitat de les dades personals que els usuaris dipositin en l'aplicació.

Com a objectiu secundari, l'aplicació web treballarà amb un conjunt de variables fisiològiques i personals de l'usuari, que s'utilitzaran per al seguiment de diferents esports que l'usuari practiqui, a més de poder compartir certes dades amb la resta de la comunitat d'usuaris de l'aplicació.

## 1.3 Enfocament i metodologia

El desenvolupament d'aquest projecte es basa en la metodologia de desenvolupament de programari. Més concretament en el model en cascada, el qual consta d'un pla de treball, d'un anàlisi i estudi d'un model d'atacant, del disseny basant-se en la fase anterior, de la implantació d'un prototip bàsic, d'unes proves de validació i finalment de la redacció de la memòria i la presentació del mateix.

Tampoc ens hem d'oblidar de la gestió de projecte pròpiament dita, que qualsevol projecte ha de seguir per poder arribar amb èxit als seus objectius. La gestió de projectes constarà d'un cicle de vida amb les següents fases: iniciació, planificació, execució, seguiment i control i tancament. Les dues primeres fases coincidiran amb la fase de pla de treball de la metodologia de desenvolupament de programari. La d'execució inclouria la d'anàlisi, disseny i implantació, i per la fase de tancament les fases de proves, memòria i presentació. La fase de seguiment i control es duu a terme durant tot el cicle de vida del projecte. En la taula 1 és fa una resum de com quedaria la metodologia a seguir en el projecte.

| Gestió de projecte | | Metodologia de desenvolupament de programari (model en cascada) | Descripció |
|---|---|---|---|
| **Seguiment i control** | **Iniciació Planificació** | Pla de treball | Aquesta fase constarà: explicació detallada del problema, objectius, metodologia, tasques, planificació, i revisió de l'estat de l'art. |
| | **Execució** | Anàlisi i estudi d'un model d'atacant | Estudi de les diferents tècniques conegudes que un atacant utilitzarà contra el tipus d'aplicació a desenvolupar. |
| | | Disseny de l'aplicació | Disseny de l'aplicació amb els protocols criptogràfics en base a la fase anterior. |
| | | Implantació d'un prototip | Implementació d'un prototip amb funcionalitats bàsiques i els protocols criptogràfics analitzats en la fase anterior. |
| | **Tancament** | Proves | Estudi de la seguretat que ofereix l'aplicació desenvolupada. |
| | | Memòria | Redacció del document final de la memòria. |
| | | Presentació projecte | Creació de la presentació i presentació virtual. |

*Taula 1. Resum metodologia del projecte*

## 1.4 Planificació del projecte

### 1.4.1 Tasques desenvolupades

Per la realització d'aquest projecte, s'han establert una sèrie de tasques. Dins d'aquestes podem distingir una sèrie de tasques primàries, les quals es composen d'altres tasques més detallades:

- Definició del pla de treball d'aquest projecte:
  - Resum, objectius, metodologia i requeriments.
  - Planificació temporal i elaboració del cronograma.
  - Estat de l'art.

- Estudi i anàlisi d'un model d'atacant:
  - Escolta de la xarxa.
  - Suplantació del lloc web.
  - Robatori de dades de la base de dades del servidor de l'aplicació.
  - Robatori de les credencials d'usuari.
  - Atac d'home en el mig (MITM).
  - Atac de denegació de servei (DoS).

- Disseny:
  - Anàlisis de les diferents tecnologies web per a dissenyar l'aplicació.
  - Definició de l'arquitectura de l'aplicació i dels agents que hi interactuaran.
  - Disseny del diagrama UML de l'aplicació.
  - Protocols criptogràfics:
    - Connexions segures:
      - El protocol SSL.
      - Autoritats de certificació.
    - Confidencialitat de les dades:
      - Xifratge de la base de dades.
      - Gestió de claus.
      - Gestió de permisos.
      - Altres.
    - Autenticació dels usuaris:
      - Sistema d'usuari i contrasenya.
      - Verificació en dues passes.
  - Esquema del model de l'aplicació amb els diferents protocols de seguretat adoptats.
  - Diagrames de casos d'ús i de seqüencia de l'aplicació.

- Implementació:
  - Implementació d'un prototip inicial amb funcions bàsiques.
  - Preparació de la base de dades amb totes les variables i la seguretat requerida.
  - Adequació d'un entorn de simulació de servidor amb la base de dades i un servidor d'aplicacions web.

  - Aplicació al prototip i a l'entorn de simulació els protocols criptogràfics analitzats en la part de disseny.

- Proves:
  - Realització de proves de funcionament del prototip.
  - Anàlisi de la seguretat implantada a l'aplicació segons el model de l'atacant modelitzat.

- Memòria:
  - Redacció de la memòria.
  - Conclusions i perspectiva de treball futur.
  - Entrega de la memòria junt amb el prototip testat.

- Presentació:
  - Creació de la presentació.
  - Presentació virtual del projecte.

- Gestió del projecte (realitzat des de l'inici fins al final del projecte).

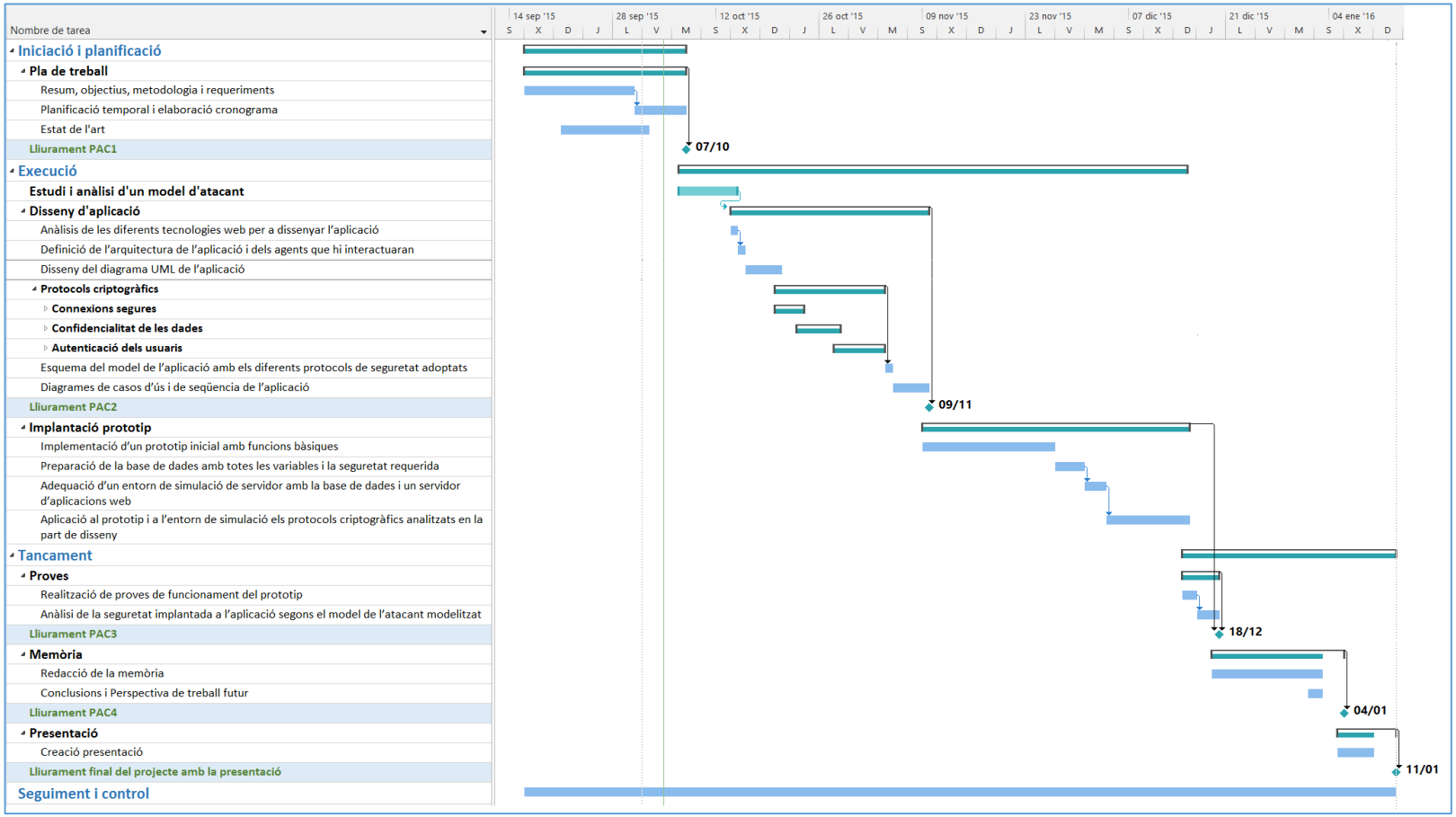
## 1.4.2 Planificació temporal

Per la realització de la planificació temporal s'ha utilitzat l'eina Microsoft Project. En la il·lustració 1 es mostra la taula de tasques amb la planificació detallada, i les diferents fites corresponents a les diferents PAC's que composen el projecte.

| Id | Modo de tarea | Nombre de tarea | Duración | Comienzo | Fin | Predecesoras |
|---|---|---|---|---|---|---|
| 1 | | **Iniciació i planificació** | **16 días** | **mié 16/09/15** | **mié 07/10/15** | |
| 2 | | **Pla de treball** | **16 días** | **mié 16/09/15** | **mié 07/10/15** | |
| 3 | | Resum, objectius, metodologia i requeriments | 11 días? | mié 16/09/15 | mié 30/09/15 | |
| 4 | | Planificació temporal i elaboració cronograma | 5 días? | jue 01/10/15 | mié 07/10/15 | 3 |
| 5 | | Estat de l'art | 10 días? | lun 21/09/15 | vie 02/10/15 | |
| 6 | | **Lliurament PAC1** | 0 días | mié 07/10/15 | **mié 07/10/15** | 2 |
| 7 | | **Execució** | **49 días** | **mié 07/10/15** | **lun 14/12/15** | |
| 8 | | **Estudi i anàlisi d'un model d'atacant** | **6 días** | **mié 07/10/15** | **mié 14/10/15** | |
| 9 | | **Disseny d'aplicació** | **19 días** | **mié 14/10/15** | **lun 09/11/15** | 8 |
| 10 | | Anàlisis de les diferents tecnologies web per a dissenyar l'aplicació | 1 día? | mié 14/10/15 | mié 14/10/15 | |
| 11 | | Definició de l'arquitectura de l'aplicació i dels agents que hi interactuaran | 1 día? | jue 15/10/15 | jue 15/10/15 | 10 |
| 12 | | Disseny del diagrama UML de l'aplicació | 3 días? | vie 16/10/15 | mar 20/10/15 | |
| 13 | | **Protocols criptogràfics** | **11 días** | **mar 20/10/15** | **mar 03/11/15** | |
| 14 | | **Connexions segures** | **4 días** | **mar 20/10/15** | **vie 23/10/15** | |
| 17 | | **Confidencialitat de les dades** | **4 días** | **vie 23/10/15** | **mié 28/10/15** | |
| 22 | | **Autenticació dels usuaris** | **5 días** | **mié 28/10/15** | **mar 03/11/15** | |
| 25 | | Esquema del model de l'aplicació amb els diferents protocols de seguretat adoptats | 1 día? | mié 04/11/15 | mié 04/11/15 | 13 |
| 26 | | Diagrames de casos d'ús i de seqüencia de l'aplicació | 3 días? | jue 05/11/15 | lun 09/11/15 | |
| 27 | | **Lliurament PAC2** | 0 días | lun 09/11/15 | **lun 09/11/15** | 9 |
| 28 | | **Implantació prototip** | **26 días** | **lun 09/11/15** | **lun 14/12/15** | |
| 29 | | Implementació d'un prototip inicial amb funcions bàsiques | 14 días? | lun 09/11/15 | jue 26/11/15 | |
| 30 | | Preparació de la base de dades amb totes les variables i la seguretat requerida | 2 días? | vie 27/11/15 | lun 30/11/15 | |
| 31 | | Adequació d'un entorn de simulació de servidor amb la base de dades i un servidor d'aplicacions web | 3 días | mar 01/12/15 | jue 03/12/15 | 30 |
| 32 | | Aplicació al prototip i a l'entorn de simulació els protocols criptogràfics analitzats en la part de disseny | 7 días | vie 04/12/15 | lun 14/12/15 | 31 |
| 33 | | **Tancament** | **21 días** | **lun 14/12/15** | **lun 11/01/16** | |
| 34 | | **Proves** | **5 días** | **lun 14/12/15** | **vie 18/12/15** | |
| 35 | | Realització de proves de funcionament del prototip | 2 días? | lun 14/12/15 | mar 15/12/15 | |
| 36 | | Anàlisi de la seguretat implantada a l'aplicació segons el model de l'atacant modelitz | 3 días? | mié 16/12/15 | vie 18/12/15 | 35 |
| 37 | | **Lliurament PAC3** | 0 días | vie 18/12/15 | **vie 18/12/15** | 28;34 |
| 38 | | **Memòria** | **12 días** | **vie 18/12/15** | **lun 04/01/16** | |
| 39 | | Redacció de la memòria | 11 días? | vie 18/12/15 | vie 01/01/16 | |
| 40 | | Conclusions i Perspectiva de treball futur | 2 días? | jue 31/12/15 | vie 01/01/16 | |
| 41 | | **Lliurament PAC4** | 0 días | lun 04/01/16 | **lun 04/01/16** | 38 |
| 42 | | **Presentació** | **6 días** | **lun 04/01/16** | **lun 11/01/16** | |
| 43 | | Creació presentació | 5 días? | lun 04/01/16 | vie 08/01/16 | |
| 44 | | **Lliurament final del projecte amb la presentació** | 0 días | lun 11/01/16 | **lun 11/01/16** | 42 |
| 45 | | **Seguiment i control** | **84 días?** | **mié 16/09/15** | **lun 11/01/16** | |

*Il·lustració 1. Planificació de les tasques i fites.*

## 1.4.3 Diagrama de Gantt




*Il·lustració 2. Diagrama de Gantt.*

## 1.5 Estat de l'art

En l'actualitat en el mercat hi ha una gran diversitat d'aplicacions *sports tracker*, a causa del gran volum de dispositius mòbils intel·ligents apareguts en els últims anys, sobretot als telèfons i als rellotges intel·ligents. Els principals fabricants de dispositius esportius, com els rellotges pulsòmetres, disposen d'aplicacions tant per a web com per a dispositius mòbils, que ofereixen als seus client una gestió integral de les dades relacionades amb els esports que practiquen juntament amb una comunitat d'usuaris per compartir-los. A més, gràcies als *smartphones* que incorporen una gran diversitat de sensors que poden ser utilitzats per monitorar l'activitat física dels usuaris, als mercats d'aplicacions dels sistemes mòbils tals com *iOS* i *Android* existeixen una gran varietat d'aplicacions d'*sports tracker*.

Exemples d'aplicacions existents al mercat són: Connect de Garmin, Polar Flow de Polar, Runtastic, Endomondo, Runkeeper i un llarg etc. Totes aquests exemples ofereixen aplicacions per a les diferents plataformes mòbils com aplicacions web, i les grans marques, dispositius com són els rellotges seguidors d'esports. A continuació es descriuen tres de les aplicacions esmentades anteriorment, ja que són unes de les més conegudes en el mercat català.

### 1.5.1 Connect de Garmin

Segons el fabricant[1] és una eina d'entrenament en línia que permet emmagatzemar, analitzar i compartir totes les activitats físiques. És una gran comunitat de milions d'usuaris que practiquen esports com el córrer, natació o el caminar. L'aplicació té una interacció perfecta amb els dispositius Garmin per ajudar als usuaris a aconseguir els seus objectius.

L'aplicació és capaç de mostrar a l'usuari un mapa amb les rutes, la temperatura, les voltes, una gran varietat de gràfics i la possibilitat d'afegir notes a les activitats. Tot això, depenent dels dispositius que s'utilitzin en l'entrenament. A més, l'usuari té un control del progrés amb ajuda de dotzenes d'informes predeterminats en el quals es poden controlar variables com la velocitat, ritme, cadència, ritme cardíac,...

L'aplicació també ofereix plans d'entrenament professionals dissenyats per entrenadors experts, a on l'usuari pot definir el nivell de l'activitat i el seu objectiu, i l'aplicació omplirà el calendari del l'usuari amb un seguit de sessions d'entrenament personalitzades. Hi ha la possibilitat de transferir les sessions de l'entrenament a alguns dispositius de la marca per poder rebre instruccions pas a pas.

La comunitat social que ofereix l'aplicació, permet als usuaris veure les rutes d'altres usuaris i quines activitats s'han practicat en aquella ruta. Així un usuari pot buscar noves rutes en base a altres usuaris, i veure les activitats practicades per poder comparar el seu rendiment amb altres. Addicionalment, els usuaris podran estar connectats uns amb els altres per poder seguir i comentar els diferents entrenaments o intercanviar missatges. Existeix la possibilitat de compartir les activitats amb altres xarxes socials com són FaceBook, Twitter i altres xarxes

[1] Dades consultades en la pàgina web de l'aplicació: https://connect.garmin.com/es-ES/features/

socials. L'usuari té la possibilitat de configurar un bloc de privacitat per a què les seves dades no siguin mostrades als altres usuaris.

Sobre la seguretat implementada en aquesta aplicació web, en l'apartat de privacitat de la seva pàgina web, ens trobem un apartat sobre seguretat que ens diu[2]:

> *"Garmin pren mesures de seguretat raonables per ajudar a protegir contra la pèrdua, mal ús, accés no autoritzat i la divulgació no autoritzada o alteració de la informació personal sota el seu control. En alguns dels nostres llocs pot crear un compte per a participar o obtenir beneficis addicionals. La transmissió de la informació que proporcioni a Garmin durant els processos de registre està xifrat utilitzant la tecnologia Secure Socket Layer (SSL). Si vostè té raons per creure que la seva interacció amb nosaltres ja no és segura (per exemple, si vostè sent que la seguretat de qualsevol compte que pugui tenir amb nosaltres s'ha compromès), si us plau notifiqui immediatament del problema posant-se en contacte amb nosaltres d'acord amb el paràgraf 10 descrit més avall."*

Tal i com diu el paràgraf anterior, es pot observar que la pàgina d'inici es connecta al servidor amb una connexió segura (https) i que posseeix un certificat signat per una autoritat certificadora.

### 1.5.2 Endomondo

L'aplicació transforma el mòbil del l'usuari en un entrenador personal, dissenyat per a practicar esports com el córrer, ciclisme, caminar i altres activitats esportives. A més, ofereix la possibilitat de ser una xarxa social permetent connectar-se amb altres usuaris per compartir activitats i poder rebre estímuls d'altres. Permet la integració amb una amplia gamma de rellotges i sensors per a millorar l'experiència de l'usuari i proporcionar-li dades d'entrenament més detallades, com estadístiques de freqüència cardíaca.

En la pàgina web de l'aplicació[3] se'ns descriu una taula de funcionalitats que ofereix l'aplicació, tant en la versió gratuïta com en la de pagament. Seguidament es realitza un llistat de les diferents funcions més importants:

- Entrenador personal:
  - Seguiment GPS y Mapa en temps real.
  - Informació d'entrenador en forma de veu.
  - Historial d'entrenaments.
  - Objectius d'entrenaments.
  - Etiquetat i fotos.
  - Sensors de freqüència cardíaca.
  - Personalització de l'aplicació, tal com la pantalla d'entrenament amb dades com distancia, duració, ritme, freqüència cardíaca, calories, etc.
  - Varis esports.
  - Escoltar la teva música favorita durant l'entrenament.

[2] Informació sobre la privacitat de Garmin: http://www.garmin.com/en-US/legal/privacy-statement
[3] Dades consultades en la pàgina web de l'aplicació: https://www.endomondo.com/features

- Motivador social:
  - Comunitat social esportiva i global en tot el mon.
  - Seguir als teus amics, inspirar-se amb les seves activitats, animals i veure que està passant en la comunitat.
  - Enviar i rebre paraules d'ànim d'amics durant l'entrenament.
  - Trobar motivació participant en els desafiaments, o fixa el teu propi objectiu i competeix amb els teus amics o col·legues.
  - Descobrir i crear noves rutes en els teus voltants i converteix-te en el campió de ruta.
  - Compartir els teus entrenaments en la teva xarxa social preferida (FaceBook, Twitter,...).
  - L'aplicació es pot configurar en una gran diversitat d'idiomes.

- Funcionalitats Premium (versió de pagament):
  - Pla d'entrenaments personalitzats.
  - Estadístiques avançades.
  - Entrenament d'intervals.
  - Gràfics interactius.
  - Zones de freqüència cardíaca.
  - Informació meteorològica.
  - Configuració personalitzada.
  - Objectius d'entrenament addicionals.
  - Historial de rècords personals.
  - Comparació d'entrenaments.
  - Estadístiques comparatives amb altres usuaris.
  - Sense anuncis.
  - Assistència VIP. Rebre ajuda per correu electrònic en 24 hores laborals.

Sobre la compartició d'informació també es disposa d'un bloc de privacitat que l'usuari pot definir al seu gust, per decidir la informació que es mostra a la resta de la comunitat. I en el cas de la seguretat, la connexió a la pàgina principal de l'aplicació es realitza a través d'una connexió segura (https) amb certificat signat per una autoritat certificadora. En l'apartat de privacitat de la seva pàgina web no es troba cap apartat que parli de la seguretat aplicada sobre les dades personals que gestiona l'aplicació.

### 1.5.3 RunKeeper

L'aplicació es presenta[4] com una gran comunitat de corredors, i que ofereix ser una gran motivadora per als usuaris. Ens assegura tindre una comunitat d'usuaris superior als 45 milions de corredors.

[4] Dades consultades en la pàgina web de l'aplicació: https://runkeeper.com/#where-to-start-module

Les funcionalitats que ens ofereix són molt semblants a les aplicacions anteriors. Entre les més destacades tenim els plans d'entrenament que pot seguir l'usuari, creació de rutines personalitzades per assolir objectius, el seguiment dels resultats per analitzar la progressió cap als objectius i l'assoliment de grans resultats en esdeveniments esportius pot generar premis per part de grans marques esportives.

Més concretament, ofereix a l'usuari funcionalitats com:

- La distància, el temps i el ritme.
- Seguiment del progrés amb altres usuaris amics de la comunitat.
- Gravar, descobrir i crear noves rutes per córrer.
- Motivació a llarg termini a través de desafiaments.
- Comparació d'entrenaments.
- Assignació d'un objectiu realista per a l'usuari per part de l'aplicació.
- Llistes de reproducció de música per l'entrenament.

En l'anàlisi d'aquesta aplicació, val la pena fer un incís sobre la política de privacitat que ofereix als seus usuaris. Segons la seva pàgina web, si ens anem a l'apartat de polítiques de privacitat, es pot trobar un apartat que fa referència a la seguretat. Aquest apartat de seguretat diu literalment[5]:

> *"Prenem mesures raonables per protegir les dades personals facilitades amb ajuda de diferents serveis, per evitar la pèrdua, el mal ús i l'accés no autoritzat, la revelació, alteració o destrucció d'aquestes. Aquestes mesures inclouen revisions internes de les nostres pràctiques de recopilació de dades, emmagatzematge i processament i les mesures de seguretat, així com mesures de seguretat física per protegir contra l'accés no autoritzat als sistemes en què emmagatzemem les dades personals."*

Tot i això, fan constar que no poden assegurar la seguretat de la informació al cent per cent alhora d'utilitzar Internet, ja que en cas de rebre informació a través d'internet amb aplicacions de tercers, tal com un correu electrònic, no poden assegurar la seva privacitat.

> *"Nosaltres restringim l'accés a les dades personals dels nostres empleats, contractistes i agents que necessiten conèixer aquesta informació per operar, desenvolupar o millorar els nostres serveis. Aquests individus estan obligats per normes de confidencialitat i poden estar subjectes a la disciplina d'aquestes si no les compleixen."*

Per acabar de parlar sobre la seguretat, tal com succeeix amb les altres aplicacions analitzades, la pàgina principal de l'aplicació es connecta al servidor amb una connexió segura (https) i posseeix un certificat signat per un autoritat certificadora.

[5] Informació sobre la privacitat de RunKeeper: https://runkeeper.com/privacypolicy

## 1.6 Requeriments

Per a portar a terme aquest projecte, s'han definit una sèrie de requeriments tant funcionals com de seguretat bàsics, per al desenvolupament del prototip del projecte:

1. ***Funcionals:***

- Selecció del següent conjunt de variables sobre la salut: pes, mida, pulsacions cardíaques, estat físic (a nivell esportiu).
- Selecció del següent conjunt de pràctiques esportives: caminar, córrer i ciclisme.
- Selecció del següent conjunt de dades personals: calories cremades durant els diferents esports, edat i hàbits que poden afectar de forma positiva o negativa alhora de realitzar exercicis físics.
- L'oferiment als usuaris d'una aplicació web, on disposaran d'un àrea privada amb un perfil amb les seves dades personals, i les diferents pantalles de les pràctiques esportives que l'usuari disposa.
- L'usuari tindrà la possibilitat de compartir certes parts del seu perfil, com dels resultats obtinguts en les diferents pràctiques esportives, amb la resta de la comunitat d'usuaris de l'aplicació.
- Els usuaris per poder seguir les seves activitats esportives, tindran que disposar d'algun dispositiu o telèfon intel·ligent que recopili les diferents variables, i amb opció d'exportar les dades a traves d'arxius *tcx.*

2. ***De seguretat:***

- L'accés a l'àrea privada per part dels usuaris es realitzarà a través d'un sistema segur d'autenticació, tal com un nom d'usuari junt amb una contrasenya.
- El servidor de l'aplicació assegurarà la integritat, confidencialitat i l'accés no autoritzat a les dades d'un usuari per part d'altres usuaris com de tercers.
- L'usuari alhora de connectar-se a l'aplicació web amb el seu navegador, ho farà a través d'un canal segur xifrat, a més de poder assegurar l'autenticació del lloc web a on l'usuari està connectat.

## 1.7 Productes obtinguts

El desenvolupament del projecte ha donat com a resultat una aplicació web anomenada *Sports Tracker* amb una sèrie de funcionalitats bàsiques i que compleix els objectius finals del projecte. Aquesta pròpia memòria amb l'estudi i anàlisi de models d'atacant, del disseny i dels detalls d'implementació de l'aplicació és un altre dels productes obtinguts del projecte.

## 1.8 Descripció de la resta de capítols

En el següent capítol es realitza un estudi i anàlisi de possibles diferents models d'atacant que una aplicació web com la desenvolupada pot sofrir. Es composa de sis apartats, els quals cadascun d'ells fa referència a un tipus de metodologia d'atac diferent i per a cada cas s'intenta establir les mesures de protecció adequades per evitar-los.

En el capítol 3, un cop estudiat els diferents models d'atacant i mesures preventives, es realitza el disseny de l'aplicació. Els primers apartats defineixen la tecnologia, l'arquitectura i diferents diagrames de desenvolupament de l'aplicació. En l'apartat de protocols criptogràfics es defineixen tres grans grups claus per l'aplicació: les connexions segures, l'autenticació d'usuaris i la confidencialitat de les dades. La resta d'apartats fan referència a diversos diagrames de disseny de l'aplicació, entre ells el de la base de dades.

El capítol 4 descriu els detalls més significatius de la implementació de l'aplicació. Els primers apartats descriuen la preparació de l'entorn, l'estructura de fitxers del projecte i una descripció de les diferents pàgines web de l'aplicació desenvolupada. L'últim apartat detalla la implementació dels protocols criptogràfics en tres blocs igual que en el capítol de disseny. En el capítol 5 es realitzen les proves de prototip obtingut en aquest capítol.

En l'últim capítol −el número 6− es descriuen les conclusions finals que s'han arribat un cop desenvolupat el projecte, juntament amb unes línies de treball futur d'aquest.

# Capítol 2. Estudi i anàlisi de models d'atacant

L'objectiu principal d'aquest projecte és el disseny i implementació d'una aplicació web d'*sports tracker* segura. Així, un cop revisat l'estat de l'art d'aquest tipus d'aplicacions i abans de començar el disseny de l'aplicació, s'ha de fer un estudi d'un possible model d'atacant; per més tard en el disseny poder aplicar les mesures adequades.

La seguretat en aplicacions web, les quals es basen en l'ús de la xarxa d'Internet, estan exposades a una infinitat de riscos. El tràfic de dades entre el client de l'aplicació web i el servidor a on resideix aquesta, es realitza a través d'una xarxa que és pública i que té accés tothom. Per tant, quan les dades que transmet el client són dades altament sensibles, s'ha de prendre les mesures adequades per impedir que un tercer es pugui apoderar d'aquestes.

No es pot assegurar al cent per cent la seguretat d'una aplicació web, però sí en un alt percentatge prenent les mesures adequades per a totes aquelles vulnerabilitats conegudes. En aquest apartat s'intenta fer un estudi de les diferents tècniques que un possible atacant té al seu abast, per intentar apoderar-se, manipular o esborrar informació sensible dels usuaris de l'aplicació.

En l'actualitat existeixen empreses en el mercat dedicades a l'àrea de seguretat, les quals realitzen proves de penetració (*pen-testing*) en sistemes per a descobrir possibles vulnerabilitats i corregir-les el més aviat possible. Hi ha molta literatura i informació per la xarxa a l'abast de qualsevol a on es descriuen tècniques, eines i vulnerabilitats de protocols. Clar, sempre s'ha de tindre en compte que hi pot existir alguna vulnerabilitat encara no coneguda pel públic en general i sí per alguna o algunes persones amb intencions no massa bones que vulguin aprofitar-se d'aquest avantatge. Una nova vulnerabilitat trobada i encara no coneguda pel públic en general se l'anomena atac de dia zero (*zero-day*). En el document *Zero-day (computing)* de la Viquipèdia de la referència bibliogràfica [9] s'explica aquest tipus d'atac.

A continuació es descriuen una sèrie de vulnerabilitats que una aplicació web està exposada. Aquestes vulnerabilitats es basen en l'ús d'eines de xarxes, vulnerabilitats de protocols, enginyeria social, i tot un seguit de tècniques que juntes poden ocasionar una bretxa en un sistema produint una fuita de dades personals dels usuaris.

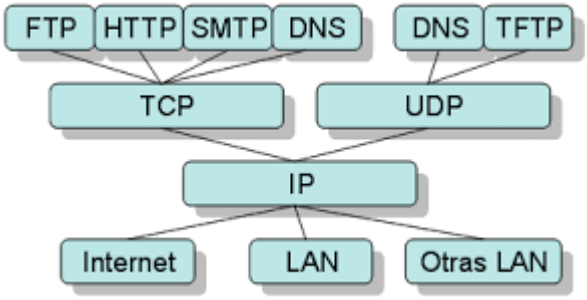
En els apartats següents es descriuen una sèrie de diferents atacs que un atacant podria utilitzar per intentar extreure informació de l'aplicació a dissenyar, a més es proposa una sèrie de mesures per mitigar-los o evitar-los completament amb ajuda de protocols criptogràfics.

## 2.1 Escolta de xarxa

L'escolta de xarxa consisteix a què un atacant amb ajuda d'una eina especifica és capaç d'interceptar tot el tràfic de dades (normalment paquets de la xarxa) que es produeix entre la víctima i el servidor de l'aplicació. A aquest tipus d'atacant se'l coneix com atacant passiu, ja que només escolta i no intenta interferir en la connexió. L'eina que un atacant utilitza en aquest cas se l'anomena *sniffer* de xarxa, la qual és una aplicació d'escriptori que posa la targeta de xarxa de l'equip en mode promiscu. Aquest mode permet capturar tot el tràfic que passa per la xarxa

a la qual està connectat l'atacant, independentment els paquets vagin dirigit a ell o no. L'aplicació a través d'unes finestres mostra tots els paquets que van circulant per la xarxa, especificant els diferents protocols de xarxa detectats.

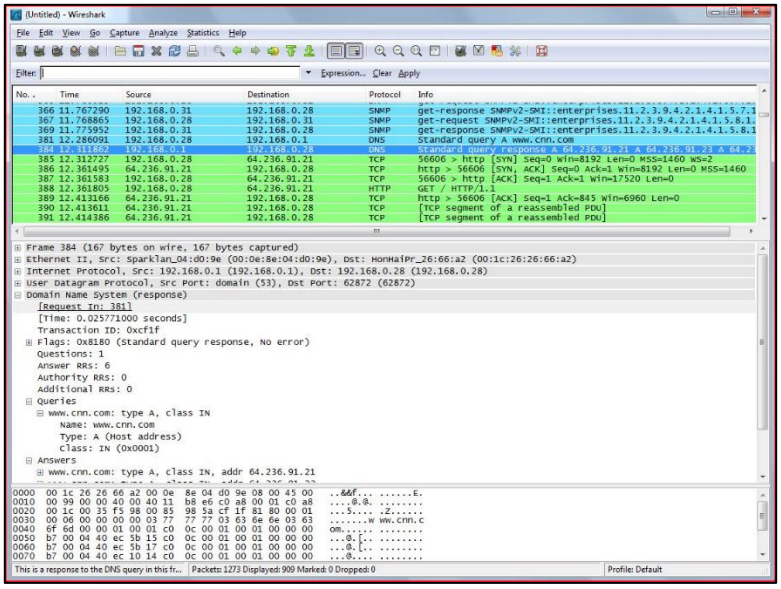
Així, un atacant amb aquesta eina, podrà interceptar tots els paquets que intercanvia una víctima amb al servidor. Aquests paquets a nivell de xarxa, utilitzen el protocol IP per al transport de dades, que és el protocol utilitzat en Internet i xarxes LAN. El protocol IP, a la vegada encapsula altres protocols de transport com TCP i UDP, i aquests a la vegada protocols d'aplicació. En la il·lustració 3 es mostren exemples de protocols utilitzats en les diferents capes de nivell per al transport de dades.




*Il·lustració 3. Protocols utilitzats en els diferents nivells de l'arquitectura TCP/IP*

Un client a l'hora d'utilitzar una aplicació web a on les dades personals circulen per xarxes públiques o compartides, està exposat a que un atacant capturi tot el tràfic de xarxa i pugui accedir a les seves dades personals. Si no es pren cap mesura per evitar-ho, les dades enviades circulen en text pla, i per tant, qualsevol persona que capturi els paquets pot veure la informació que transporten de forma clara i immediata.

Com a exemple d'eina que podria emprar un atacant per escoltar la xarxa, tenim la coneguda aplicació *WireShark*. En la il·lustració 2 és mostra l'eina amb un conjunt de paquets que ha capturat d'una xarxa.




*Il·lustració 4. Eina WireShark amb un exemple de captura de paquets*

Per tant, per evitar aquest tipus d'atac bàsic quan el client utilitzi una aplicació web, haurem de xifrar les dades que circulin entre el client i el servidor de l'aplicació. Per això ens ajudarem dels protocols criptogràfics dissenyats per assegurar connexions segures entre un client i un servidor a través de la xarxa.

El protocol SSL (*Security Socket Layer*) i el TLS (*Transport Layer Security*) són els protocols criptogràfics de seguretat d'ús comú per a connexions amb pàgines web, per establir una connexió segura entre un client i un servidor en una xarxa compartida o Internet. En el llibre *Security technologies for the Word Wide Web* de la referència bibliogràfica [7] es detalla aquests protocols.

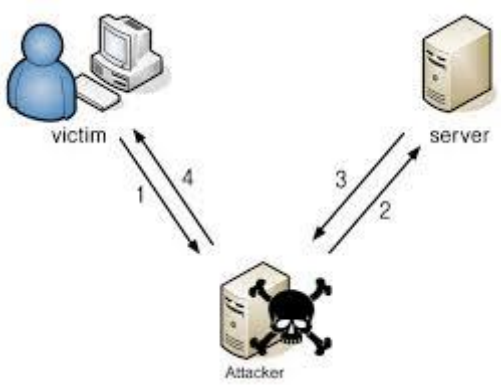
## 2.2 Suplantació del lloc web

Un atacant es podria plantejar realitzar una clonació de pàgines web de la nostra aplicació, amb l'únic objectiu de fer creure a la víctima que està interactuant amb l'aplicació vertadera, per apoderar-se de totes les dades personals que la víctima introdueix en les pàgines falses o envia al servidor de l'aplicació.

Aquesta tècnica se l'ha coneix com a *Web Spoofing*, la qual es basa en l'ús de tècniques que un atacant utilitza per fer-se passar per una entitat diferent a través de la falsificació de les dades en una comunicació. A nivell més global, la tècnica de *Spoofing*, pot englobar qualsevol tecnologia de xarxa que sigui susceptible de sofrir una suplantació d'identitat.

L'atacant el primer que fa és encaminar la connexió de la víctima als seus servidors quan aquesta tecleja la direcció de la web al navegador. Per exemple, l'atacant podria modificar l'arxiu HOST del sistema operatiu Windows per a què quan al navegador se l'introdueixi una direcció web concreta apunti a una *IP* fraudulenta; o també ho podria realitzar a través de *DNS spoofing*. Indistintament del mètode utilitzat per l'atacant per encaminar la connexió fraudulenta, la víctima quan tecleja la direcció de l'aplicació realment està fent la petició al servidor de l'atacant.

L'atacant, a la vegada haurà preparat una o varies pàgines falses idèntiques a l'original, normalment formularis o pàgines de *login* idèntiques. La pàgina suplantada actua com a un servidor *proxy*, les peticions que rep l'atacant de la víctima les encamina cap als servidors originals. L'atacant té la capacitat de poder veure totes les dades de les pàgines suplantades, i per tant la capacitat de controlar i/o modificar les dades de la víctima. En aquest casos, inclús una connexió segura (SSL) no impedeix a l'atacant portar a terme l'atac, ja que a la víctima se li ofereix una connexió segura per a què no sospiti, però no amb el servidor original si no amb el servidor de l'atacant. L'atacant, a més, tindrà en compte aspectes com modificar la URL de la barra de direcció, controlar la barra d'estat del navegador i diferents petits detalls d'aquest per tal de no donar cap sospita a la víctima. La il·lustració de la pàgina següent mostra l'esquema de *web spoofing*.




*Il·lustració 5. Esquema de web spoofing.*

Exemples d'atacs d'aquest tipus han estat els que han sofert les entitats bancàries, molts cops a la víctima se li mostrava una pàgina de *login* falsa, demanant més dades del habitual com era la d'introduir totes les claus de la targeta de coordenades. En el cas de l'aplicació del projecte, un atacant podria suplantar la pàgina de *login*, per apoderar-se de les credencials de l'usuari (*phishing*), o de la pàgina del perfil per extreure directament les dades personals.

En l'actualitat és un tipus d'atac bastant perillós i difícilment detectable, tot i què es poden prendre algunes mesures de prevenció com les següents:

- Controlar l'ús del *JavaScript* en el navegador, permetent-lo en llocs web de confiança.
- Assegurar-se que la barra de navegació del navegador està activa.
- Prestar molta atenció a les URLs que es mostren en la barra d'estat, observant que les llocs web assenyalats siguin els esperats.
- Evitar l´ús de *Active-X* i altres llenguatges en llocs web que no siguin de confiança.
- No permetre per defecte *Scripts* i *Cookies* de cap classe.
- L'ús de connexions segures i verificar els certificats d'autenticació que apuntin a la direcció web esperada.
- No fiar-se de les redireccions, comprovar-les sempre.
- Conèixer la URL legítima dels llocs visitats.
- L'ús d'un navegador web actualitzat i de confiança.

## 2.3 Accés i modificació de la base de dades del servidor

Un punt crític de seguretat és a on s'emmagatzemen les dades dels usuaris. Normalment, en una aplicació web hi haurà un servidor a on residirà la bases de dades d'aquesta. És de vital importància controlar l'accés de forma completa de qui pot i a quines dades pot accedir. En el cas de l'aplicació d'aquest projecte, la base de dades contindrà dades dels usuaris altament sensibles. Per tant, s'haurà de prendre mesures per evitar un possible accés a aquestes per part de terceres persones des de la xarxa d'Internet, però també de les pròpies persones que puguin gestionar o tindre accés directament al servidor de la base de dades. Així, el primer objectiu és assegurar que solament l'usuari de les pròpies dades tingui accés a aquestes.

Temps enrere, unes de les prioritats per als serveis d'informàtica era assegurar els accessos a la xarxa a través de *firewalls*[6], IDS/IPS[7] i antivirus. Sense prestar massa atenció a la seguretat respecte a les bases de dades a on normalment poden contenir dades sensibles. En l'actualitat aquest fet està canviant, i ara es pren més importància a la seguretat de les bases de dades protegint-les contra intrusions i canvis no autoritzats. En el *Whitepaper: Base de datos y sus vulnerabilidades más comunes* de la referència bibliogràfica [12] s'explica amb més detall la seguretat en les bases de dades.

A continuació es detallen les vulnerabilitats més comunes en bases de dades:

- ***Utilització d'usuari/clau en blanc o massa feble***: s'ha de ser conscient que molts usuaris encara en l'actualitat, per facilitat d'ús, posem noms d'usuaris i claus massa fàcils i previsibles. Com per exemples, usuari *admin* i clau *1234*, el qual a través de processos automàtics es podrien descobrir fàcilment.
- ***Preferència de privilegis d'usuari per privilegis de grup***: en moltes ocasions es tendeix a donar privilegis en forma de grup als usuaris per facilitat i comoditat, però això pot generar que usuaris es trobin amb permisos que mai utilitzaran. En canvi, algun usuari maliciós podria aprofitar per portar a terme un atac aquest excés de permisos.

- ***Característiques de la base de dades habilitades sense cap ús***: les bases de dades es composen normalment per mòduls, els quals ofereixen certes característiques al servidor de la base de dades. Per norma general, mai s'aprofiten o es donen ús a totes les característiques que ofereix les bases de dades, i molts mòduls es queden activitats sense donar-li cap ús. Aquest fet genera un augment d'inseguretat, ja que qualsevol vulnerabilitat que es descobreixi d'un mòdul podria oferir una porta d'entrada a un atacant. Per tant, si no es fan servir certs mòduls cal desactivar-los i reduirem la probabilitat d'un atac al fer funcionar menys mòduls del total disponibles. A més, que solament ens haurem de preocupar d'actualitzar aquells mòduls que emprem d'una forma prioritària, deixant de banda o sense molta prioritat els no utilitzats.

- ***Desbordament de buffer***: a l'hora d'entrar dades a una aplicació que rebrà el servidor per introduir-les en la base de dades, es pot donar el cas que rebi molta més informació de la que espera. Això pot generar un desbordament del buffer de la base de dades, ocasionant pèrdua d'informació i modificacions. Com a exemple, es pot produir quan l'usuari emplena un formulari i en un camp que s'espera rebre solament dos números introdueix un text d'una gran llargària, i alhora de l'enviament del formulari no es comproven les dades introduïdes.

- ***Falta d'actualitzacions de la base de dades***: com a qualsevol altre programari del servidor, una política fonamental és la d'actualitzacions. S'ha d'estar informat i fer un seguiment de les possibles actualitzacions de la bases de dades per part del fabricant, sobretot per assabentar-se el més aviat possible de qualsevol nova actualització que corregeixi una possible bretxa de seguretat que ens pot afectar.

- ***Dades altament sensibles sense xifrar***: s'ha de posar èmfasi que les dades sensibles d'una base de dades han d'estar sempre xifrades, incloent-hi les claus dels usuaris que

[6] Tallafocs: programari per controlar el tràfic de xarxa entrant i sortint d'una xarxa.
[7] IDS/IPS: Sistemes de detecció d'intrusions

utilitzen per identificar-se. Aquestes últimes, les claus, es poden emmagatzemar amb ajuda de funcions resum (*hash*) com SHA[8]. Això impediria que en cas que un atacant accedeixi a la base de dades conèixer la clau original.

- ***Injeccions SQL***: aquest tipus d'atac es bastant perillós, i en l'actualitat és un dels més utilitzats pels atacants, ja que pot donar accés complet sense cap restricció a la base de dades. L'atac es porta a terme a través de l'entrada de dades de l'aplicació, com pot ser qualsevol tipus de formulari. Si en aquest formulari no es realitza una comprovació i neteja de les dades introduïdes en els diferents camps, pot arribar a permetre l'execució de codi maliciós SQL[9] directament a la base de dades. Molts proveïdors de base de dades ja ho tenen en compte en les últimes versions, per això són molt importants les actualitzacions.

Un cop detallades les vulnerabilitats més comunes en les bases de dades, també es pot donar una sèrie de recomanacions per a protegir una base de dades, a part de les que es poden extreure directament de les recomanacions anteriors:

- ***Proves de seguretat***: per a implementar les mesures que s'han de prendre en una base de dades, primer s'ha de conèixer exactament els que s'ha de protegir o corregir. Per tant, s'haurien de realitzar proves contra la base de dades per poder identificar possibles forats de seguretat, i així poder reforçar els punts febles.

- ***Configuració i avaluació de vulnerabilitats***: s'hauria de revisar la configuració completa de la base de dades junt la del sistema operatiu, per a què no hi hagi cap element amb més privilegis o accessos dels esperats a la base de dades o al mateix servidor. Això ens pot ajudar a esbrinar possibles vulnerabilitats del sistema. Els privilegis dels usuaris són un punt important a tindre en compte.

- ***Auditar***: durant el transcurs del temps s'ha d'assegurar que la configuració realitzada en un principi es manté sense cap variació, en cas contrari una supervisió continua ens ajudarà a detectar-lo.

- ***Monitorització de la base de dades***: un control continu de la base de dades, monitoritzant-la, ens servirà per poder descobrir certs usos mal intencionats o inclús intrusos dintre d'aquesta.

- ***Autenticació, control d'accés i gestió de drets***: és molt important realitzar una jerarquització dels usuaris, no tots els usuaris tindran accés a les mateixes dades. Com ja s'ha comentat més amunt, els permisos dels usuaris han de ser els justos per mantenir una seguretat mínima. A més, l'ús de xifratge en les dades sensibles que solament l'usuari legítim pugui veure en clar, i cap persones més inclús accedint a la base de dades.

[8] SHA (Secure Hash Algorithm): algoritmes de funcions resum de xifrat dissenyat per NIST dels EUA.
[9] SQL (Structured Query Language): llenguatge declaratiu d'accés a les bases de dades relacionals.

En l'actualitat existeix un projecte anomenat OWASP Top Ten[10], en el qual s'aglutina una llista de les 10 falles de seguretat més comunament explotades en aplicacions web. En la llista publicada l'any 2013, es pot observar que les tres primeres posicions vénen donades per falles d'injecció en el costat del servidor. Les tècniques d'injecció SQL apareixen sistemàticament en atacs contra llocs web, tals com el ***Bit9*** o els dels ***premis Goya***, juntament amb la injecció de comandes de sistema operatiu i les injeccions *LDAP*[11]. Els atacs a llocs web molt coneguts i les diferents tècniques utilitzades es detallen en el bloc *Un informático en el lado del mal* de la referència bibliogràfica [11]. Per tant, la seguretat en la base de dades d'una aplicació web es clau, i s'han de prendre totes les mesures necessàries i conegudes existents fins al moment.

## 2.4 Enginyeria social: robatori de les credencials d'usuari

Un possible atac pot dirigir-se a aconseguir les credencials dels usuaris a través dels mateixos usuaris, ja que són un punt dèbil de seguretat en molts dels casos. En aquests tipus d'atac, a part de basar-se en l'ús de qualsevol tipus de programari maliciós, és habitual l'ús de l'enginyeria social. Aquest últim concepte, fa referència als atacs que es basen en intentar fer creure alguna cosa a l'usuari per a què aquest de forma voluntària li faciliti les pròpies credencials a l'atacant. L'usuari es pensa que ha establert una comunicació amb una persona coneguda, però realment és l'atacant que s'està fent passar per aquesta persona.

El robatori de credencials dels usuaris també pot vindre a través d'atacs de les bases de dades, d'escolta de la xarxa i de molts altres els quals intenten aconseguir accedir a informació personal el usuari. Però alguns d'aquests tipus d'atacs ja s'han analitzat en els apartats anteriors d'aquest capítol, i s'han vist les mesures habituals a prendre per evitar-los o pal·liar-los.

Sobre la utilització de l'enginyeria social per realitzar atacs per aconseguir credencials o altra informació sensible, existeix el que es coneix com a *phishing* o suplantació d'identitat. L'atacant, anomenat *phisher*, es fa passar per una persona o empresa de confiança a través d'una comunicació oficial electrònica. Normalment les comunicacions es basen en correu electrònic, programari de missatgeria instantània, xarxes socials i missatges SMS/MMS entre altres; basant-se en l'existència d'un programari maliciós que ha afectat al sistema.

Els bancs són un gran exemple d'entitats que pateixen atacs de *phishing* indiscriminats. En la web *Info Spyware* de la referència bibliogràfica [13] es descriu aquest tipus d'atac i es pot trobar molta més informació. Usuaris bancaris reben correus electrònics falsejats amb una aparença semblant als originals del banc. En aquests se'ls demana fer clic a un enllaç adjunt al correu −fraudulent− que els envia a una web de l'atacant per a què introdueixi les claus d'accés al banc, amb l'excusa d'uns problemes de seguretat que ha tingut el banc i que obliga a introduir claus per canviar-les. En general, un cop que l'usuari introdueix les claus a la web fraudulenta, se li diu que en aquell moment no funciona el servei i que ho intenti més tard; però l'atacant ja s'ha apoderat de les claus. En la il·lustració de la pàgina següent es mostra en forma de resum els principals mitjans de propagació i quin tipus d'informació té com a objectiu aquest tipus d'atac.

[10] OWASP: Comunitat lliure i oberta sobre seguretat en aplicacions.
[11] LDAP (Lightweight Directory Access Protocol): és un protocol per a realitzar consultes i modificacions de directoris que corren sobre TCP/IP.

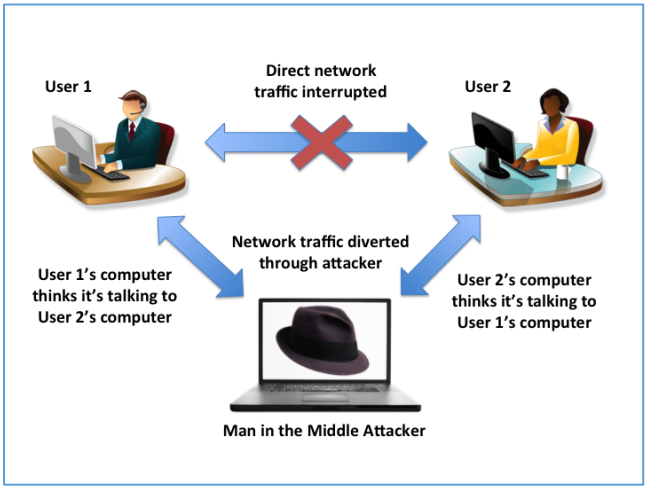

*Il·lustració 6. Medis i objectius del phishing (font: www.infospyware.com)*

Aquest atac es basa en la part del client, i per tant l'aplicació web del servidor no podrà fer gaire per impedir-lo. A part d'informar el millor possible a l'usuari per a què sàpiga que mai se'l demanarà que introdueixi la clau, a cap lloc diferent que la pàgina de registre per entrar en l'aplicació. Igualment, tot seguit es redacten unes mesures que tot usuari hauria de tindre en compte per poder detectar i evitar aquest tipus d'atac:

- Mai s'ha d'enviar dades personals a través de correu electrònic; entitats que manipulen dades sensibles mai les reben per aquest mitjà de comunicació.
- Els enllaços web inclosos en correus electrònics de no confiança s'han de bloquejar amb ajuda del client pesat de correu o ignorar-los.
- En cas de voler accedir a un enllaç d'un correu de no confiança, escriure la direcció directament en la barra de navegació del navegador.
- En cas de rebre un correu d'aquest tipus, si es té dubte de la veracitat, el primer que s'ha de fer es posar-se en contacte amb l'organització implicada.
- En cas de detectar un atac d'aquest tipus per qualsevol mitjà electrònic de comunicació, directament s'ha d'esborrar el missatge.
- Si s'ha teclejat la direcció en el navegador per entrar a la web indicada a l'enllaç, comprovar si es tracta d'una connexió segura amb el seu certificats corresponent.
- En cas d'entrar a la direcció web indicada a l'enllaç, s'ha de verificar que el lèxic i la sintaxis de la direcció coincideix amb l'original esperada, sense cap variació de cap lletra.
- En cas de sofrir un atac de *phishing*, canviar immediatament les claus i comunicar-ho a l'administrador de l'aplicació web.

## 2.5 Atac d'home en el mig (MITM)

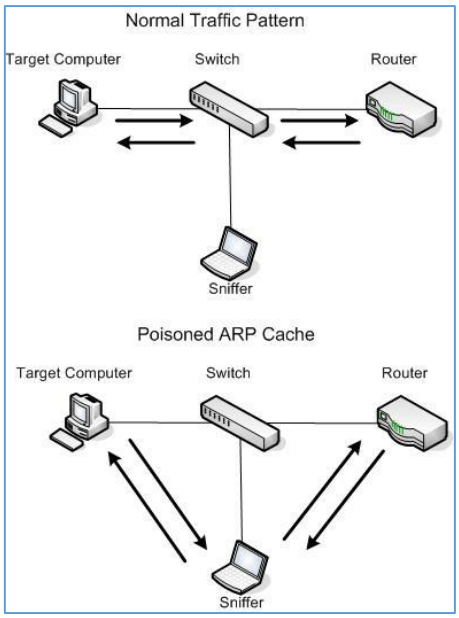
Aquest tipus d'atac es basa en que un atacant és capaç de desviar el tràfic de la víctima per a què passi pel seu propi ordinador, i tornar a redirigir-lo al seu destí original. En cap moment la víctima se n'adona de res, és un atac realment silenciós. Per si mateix l'atac no té cap efecte sobre la víctima, i se sol acompanyar d'altres tècniques d'atac. Per tant, un atacant podria arribar a controlar i/o manipular totes les dades de la víctima enviades al servidor sense que ella se n'adoni.



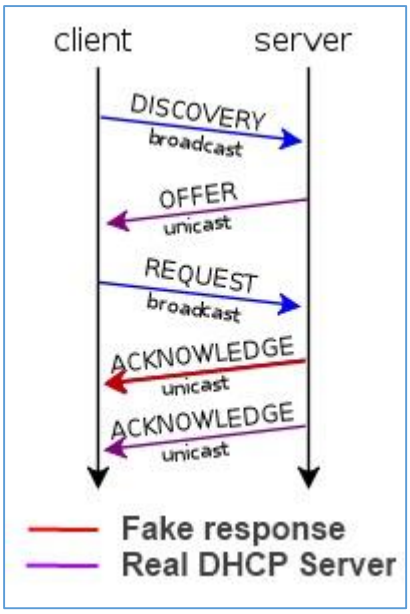

*Il·lustració 7. Atac d'home en el mig.*

A nivell d'una xarxa local, l'atac es pot realitzar a través de l'enverinament ARP[12]. Aquest atac es basa en modificar la cache ARP de la víctima per fer-li creure que la MAC de la porta d'enllaç és la direcció MAC de l'equip atacant; d'aquest forma la màquina de l'atacant es situa entre l'equip de la víctima i la porta d'enllaç. Per a aquesta tècnica d'atac, al mercat ja existeixen moltes eines per a detectar-lo, tals com *ArpON* i *Patriot* NG. A part d'aquesta tècnica existeix una altra menys coneguda, que es basa en la utilització del protocol DHCP[13]. A través de crear un servidor fals DHCP, o a través d'injectar un ACK modificat al final de la comunicació establerta pel servidor real DHCP. A continuació en la pàgina següent es mostren unes il·lustracions de les diverses tècniques vistes fins ara.

[12] ARP (Address Resolution Protocol): protocol a nivell de xarxa de resolució de direccions IP/MAC.
[13] DHCP (Dynamic Host Configuration Protocol): protocol de xarxa que permet als clients d'una xarxa IP obtenir els seus paràmetres de configuració automàticament.




*Il·lustració 8. Enverinament de ARP.*




*Il·lustració 9. Atac d'injecció de ACK en DHCP.*

Tampoc s'ha d'oblidar les connexions sense fils que avui dia són molt habituals. Un atacant pot crear un punt d'accés (AP) maliciós amb un SSID copiat d'un punt d'accés original (aquesta variant se l'anomena *Rogue APs*). El mateix usuari, o per norma general els mateixos dispositius de forma automàtica, es connectaran a questa xarxa sense fils pensant que és la legítima. En aquest moment l'atacant podrà redirigir el tràfic al seu destí legítim, però tindrà el control de totes les dades que passin pel punt d'accés. Aquesta tècnica se sol utilitzar molt en xarxes sense fils públiques que molts comerços i llocs públics ofereixen, i a on els usuaris acostumen a connectar-se de forma molt fàcil i ràpida.


*Il·lustració 10. Símbol habitual en llocs públics que ofereixen connexió a Internet.*

Una nova tècnica d'atac bastant recent és la del *Web Proxy Auto-Discovery*. Els navegadors d'Internet per defecte venen configurats per cercar el servidor *Web Proxy Auto-Discovery* de la xarxa mitjançant un registre DNS anomenat WPAD[14]. Aquest servidor és el que conté informació d'on s'ubica realment el servidor *web proxy*. El registre WPAD és cercat automàticament per la xarxa a través del protocol LLMNR[15], i es realitza de forma *multicast*.

Un atacant podria capturar la petició *multicast* i retornar a la víctima una confirmació de que és el servidor *Web Proxy Auto-Discovery* de la xarxa. La víctima llavors sol·licitarà a l'atacant el fitxer que conté les dades d'on s'ubica el servidor *web proxy* de la xarxa, ja que pot estar en una ubicació diferent del *Web Proxy Auto-Discovery*. L'atacant enviarà el fitxer WPAD.PAC modificat

[14] WPAD (Web Proxy Auto-Discovery): és un mètode utilitzat pels clients de servidors *proxy* per a localitzar el URI d'un arxiu de configuració, utilitzant mètodes de descobriment a través de DHCP i DNS.
[15] LLMNR (Link-Local Multicast Name Resolution): protocol que permet la resolució de noms en escenaris en que no es possible utilitzar DNS, perquè no existeix cap servidor o perquè no es està a l'abast.

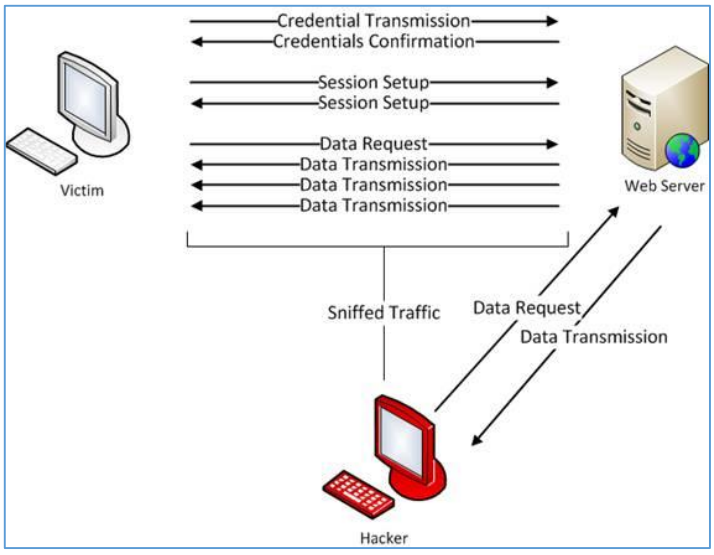
amb la seva direcció IP, que és el que conté les dades d'ubicació del servidor *web proxy*. Un cop la víctima el rebi, utilitzarà el servidor *web proxy* que l'atacant ha implementat en el seu ordinador. L'atacant en aquest moment estarà en el mig de les comunicacions entre la víctima i els servidors externs.

L'atac MITM no solament es pot realitzar en xarxes locals, si no que també es pot realitzar basant-se en altres protocols i altres arquitectures de xarxa. Aquest atac es pot realitzar fins i tot en xarxes WAN (tenint el maquinari necessari). A continuació s'han descrit a tall d'exemple una sèrie d'atacs complementaris per realitzar juntament amb l'atac MITM, i que representen un gran perill per als usuaris.

### 2.5.1 Hijacking

A través del que es coneix com a *Hijacking,* es pot segrestar una sessió, i l'atacant pot situar la seva màquina entre la comunicació de la víctima i el servidor. El *Hijacking* de sessió consisteix en l'explotació d'una sessió entre dispositius, i s'anomena segrest de sessió. Quan ens referim a una sessió, estem parlant d'una connexió entre els dispositius en què no hi ha estat. És a dir, hi ha un diàleg que estableix una connexió formalment, la connexió es manté en el temps, i un procés definit s'ha d'utilitzar per acabar la connexió.

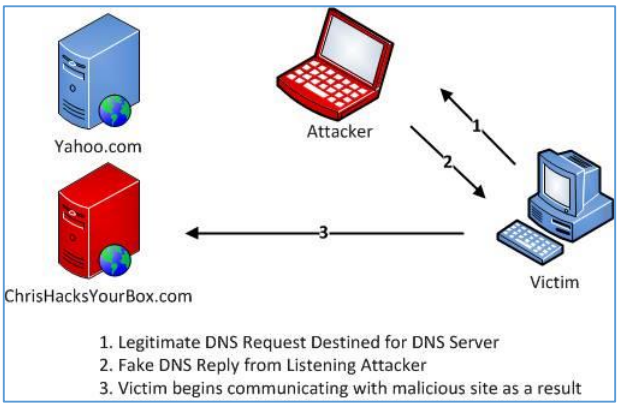
Per exemple, es pot definir el segrest de sessió a través del robatori de *cookies*, que implica sessions *http*. En els casos de llocs web amb requeriments de credencials d'accés, aquest són connexions orientades a sessions, i és necessari estar autenticat per formalitzar la sessió. La principal forma de segrest de sessió, és la que es pot interceptar certes parts de l'establiment de sessió, per utilitzar aquestes dades per fer-se passar per una de les parts involucrades en la comunicació. En el cas de robatori de *cookies*, es podria capturar la *cookie* que es fa servir per mantenir l'estat de sessió entre el navegador de la víctima i el lloc web que ha iniciat sessió. Aquesta *cookie* robada es podria presentar al servidor web, i fer-se passar per la connexió de la víctima.




*Il·lustració 11. Hijacking de sessió.*

### 2.5.2 Spoofing DNS

També existeix el *Spoofing*[16] *DNS*, terme vist en un altre apartat d'aquest capítol. En la pràctica hi ha diverses tècniques per a portar-ho a terme, aquí es detalla la tècnica anomenada *DNS ID Spoofing*. Cada consulta DNS que s'envia per la xarxa conté un número d'identificació generat automàticament, a on el seu propòsit és identificar les consultes per poder associar-hi una resposta. Així, es pot donar el cas que l'atacant intercepti el paquet de la consulta DNS, i falsifiqui un paquet de resposta que envia a la víctima. D'aquesta forma, la víctima quan vol visitar una pàgina web fa una petició al servidor DNS. L'atacant intercepta la petició, i retorna a la víctima una resposta falsa indicant una IP fraudulenta d'una pàgina de l'atacant. Per poder interceptar les peticions es podria realitzar una atac d'enverinament ARP. L'atacant d'aquesta forma, pot fer que quan l'usuari vol visitar una pàgina web concreta, com la d'un banc, redirigeixi la petició cap a una pàgina fraudulenta del banc.




*Il·lustració 12. Atac utilitzant mètode DNS ID Spoofing.*

### 2.5.3 Spoofing SSL

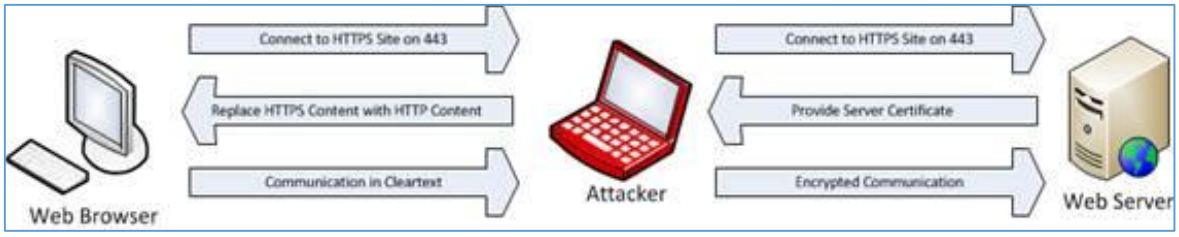
Ara es detalla una tècnica que intenta atacar a les connexions segures que utilitzen normalment el protocol SSL, anomenada *Spoofing SSL*. En aquest cas ens basem en exemplificar un atac sobre una pàgina web que utilitza *https* (protocol *http* que implementa SSL). L'atac no implica trencar la seguretat SSL, però sí el pont entre les comunicacions segures i no segures. La major part del temps, una connexió segura SSL per norma general s'inicia a través de *https*, ja sigui per un codi resposta *http 302* o fent clic a un enllaç. La idea és atacar la transició d'una connexió no segura a segura, en aquest cas de *http* a *https*. L'investigador de seguretat Moxie Moulinsart va desenvolupar una eina anomenada *SSLstrip*, per a portar a terme aquest atac amb eficàcia. En el document de la Viquipèdia Moxie Marlinspike de la referència bibliogràfica [15] es troba una petita bibliografia d'aquest autor, i en la referència bibliogràfica [16] la pàgina web *Understanding Man-In-The-Middle Attacks* a on es pot trobar la descripció de funcionament de l'eina SSLstrip. El procés que seguiria l'atac seria el següent:

- S'intercepta tot el tràfic entre el client i el servidor web.
- Quan es detecta una petició a una pàgina *https*, l'eina *SSLstrip* la reemplaça amb un enllaç *http* i es manté un mapeig dels canvis.

[16] *Spoofing*, en termes de seguretat de xarxes fa referència a l'ús de tècniques de suplantació d'identitat, generalment amb usos maliciosos o d'investigació.

- La màquina de l'atacant subministra els certificats al servidor web i suplanta al client.
- L'atacant per un canal segur rep el tràfic del servidor web, i el reenvia per un canal no segur a la víctima.

Com es pot observar, l'atacant reenvia el tràfic del servidor a la víctima per un canal no segur, i per tant veu tota la informació en clar. A més, el servidor no nota cap diferència un cop executat l'atac, ja que per part seva el tràfic es realitza per un canal segur. L'única diferència visible per part de la víctima, és que el transit amb la pàgina web no es a través de *https*, i per tant, la víctima podria ser conscient que existeix alguna cosa que no funciona correctament. A continuació es mostra el procés simplificat en la següent il·lustració:




*Il·lustració 13. Procés de l'atac Spoofing SSL.*

### 2.5.4 Heartbleed

Per finalitzar la descripció d'atacs complementaris, es descriu el famós *Heartbleed*. Aquest és una vulnerabilitat pel descobriment d'una falla de seguretat en *OpenSSL*, i ha fet que sigui una de les grans falles que afecten a Internet globalment. *OpenSSL* són un conjunt d'eines d'administració i de biblioteques relacionades amb la criptografia de codi obert. A principis d'abril de 2014 es va donar a conèixer un forat de seguretat que afecta a les versions 1.0.1 i 1.0.1f, i per tant afectava a dos terços de les comunicacions segures que s'efectuen en Internet. En l'article *5 cosas que debes saber sobre Heartbleed* publicat el 9 d'abril de 2014 de la referència bibliogràfica [17] especifica com s'obté el percentatge d'afectació de llocs web actius per aquesta falla. El forat de seguretat estava en el codi d'*OpenSSL* des de desembre de 2011, a on Neel Mehta de l'equip de seguretat de Google el va descobrir en desembre de 2013. Fins aquest moment, qualsevol persona amb el coneixement de la vulnerabilitat, l'havia pogut fer servir per a portar a terme atacs maliciosos sobre connexions que suposadament eren segures.

La falla radicava en la funció encarregada de gestionar els missatges *heartbeat*. Aquests són anomenats *keep-alive*, ja que és una forma de comunicar-li al servidor que es segueix connectat, per a què no tanqui la connexió. Els missatges enviats al servidor poden contenir dades (*payload*), com pot ser la data d'enviament. El servidor al rebre un missatge respon al client amb el mateix contingut que ha rebut del client. El client quan envia un missatge, ha d'informar de la longitud de les dades que ha afegit en el missatge, en un camp del mateix missatge per a què el servidor sàpiga on finalitzen les dades.

*OpenSSL* està escrit en C, i en aquest llenguatge les variables realment són apuntadors de la memòria, o sigui que indiquen la posició de la memòria on està la dada emmagatzemada i no la dada en si. Aprofitant aquest característica del llenguatge i la dels missatge *heartbeat*, es pot donar el cas que al enviar un missatge se li indica que té una llargària de dades de molts més bytes que realment té les dades. Això provocarà que el servidor al rebre el missatge llegeixi un

segment de dades de la memòria molt més gran del que realment ocupa les dades que conté el missatge, i retorni un missatge amb tot un segment de dades de la memòria del servidor que conté moltes més coses. En el segment de dades retornat, hi haurà informació del propi servidor, a vegades serà brossa però altres vegades pot arribar a contenir les claus privades de la connexió segura.

Això es produeix perquè la reserva de memòria no es determinista, i per tant pot haver-hi absolutament de tot. El problema rau en el fet que el programari no verifica si la longitud que indica el camp del missatge, es correspon realment amb la longitud real de les dades. Repetint molt cops l'atac, cada cop es pot recuperar fins a un màxim de 64kb de la memòria del servidor. Al final, es pot acabant trobant dades tant importants com les mateixes claus privades, que dintre de la memòria són fàcilment identificables.

La vulnerabilitat de *Heathbleed* compromet la funció del protocol SSL d'autenticar una o ambdues parts d'una comunicació a través d'un certificat de seguretat confiable, deixant la porta oberta per realitzar un atac MITM. Si un atacant aconsegueix capturar la clau de la connexió SSL no té cap impediment per realitzar l'atac.

Aquesta vulnerabilitat en els últimes versions de *OpenSSL* ja s'ha corregit, i per tant aquest tipus d'atac dependrà de la falta d'actualització dels sistemes en el diferents llocs web. Tot i què, en la trobada ***Black Hat Asia 2015***[17] es va presentar un nou treball d'investigació sobre les vulnerabilitats de les connexions SSL/TLS quan aquestes utilitzen l'algoritme de xifratge RC4[18]. El treball presentat explica l'atac per robar sessions SSL/TLS amb protocol RC4, i que han denominat *Bar Mitzvah*. Aquest és un algoritme criptogràfic comunament utilitzat en el xifrat de comunicacions digitals, i d'ell se'l coneix vulnerabilitats que el fan totalment insegur des de fa més de tres anys.

### 2.5.5 Reflexions sobre mesures preventives per un atac MITM

I finalment com a reflexió després de la descripció dels diferents atacs per realitzar un MITM i dels atacs complementaris per acompanyar-lo, es poden descriure unes possibles mesures preventives:

- L'ús de llocs web amb connexions segures *https* (SSL/TLS), sempre comprovant la direcció en la barra de direccions del navegador i que el certificat sigui vàlid i correspongui al domini. Així s'evita els atacs menys sofisticats, però el client continua sent vulnerable.
- Els llocs web amb connexions segures que es basen en *OpenSSL*, han d'haver actualitzat la versió com a prioritat número u per evitar la vulnerabilitat *Heartbleed*.
- L'ús de la verificació en dues passes en tots els serveis web que ho permeti. D'aquesta forma s'afegeixen barreres que dificultaran els processos de l'atacant.

[17] *Black Hat Asia 2015*: edició de trobada celebrada en Àsia de les conferències sobre seguretat informàtica que reuneixen a tot tipus de persones interessades en la ciberseguretat.
[18] RC4: sistema de xifratge de flux més utilitzat en TLS/SSL i WEP. Aquest va ser exclòs ràpidament dels estàndards de alta seguretat pels criptògrafs, ja que alguns modes d'ús de l'algoritme el feien molt insegur, incloent-hi el seu ús en WEP.

- Utilització de VPN (Virtual Private Network), que és una tecnologia que permet l'extensió segura d'una xarxa local (LAN) sobre una xarxa pública, com Internet.
- Evitar a tota costa l'ús d'encaminadors que ofereixen accés a Internet a través d'una xarxa sense fils oberta. En cas de necessitat d'utilitzar-los, es recomanable utilitzar una VPN.
- Desactivar dels navegadors l'opció d'autoconfiguració de xarxes locals.
- Configuració manual de la xarxa local (LAN) i deixar d'utilitzar el servidor DHCP.
- Ús de servidors DNS segurs.
- Utilització i configuració de tallafocs, IDS i programes de monitoratge.

## 2.6 Atac de denegació de servei (DoS)

Aquest tipus d'atac té com objectiu principal un sistema de computadors o xarxa per a què els seus serveis siguin inaccessibles als usuaris legítims. Per norma general, provoca la pèrdua de la connectivitat de la xarxa pel consum d'ample de banda de la xarxa de la víctima, o sobrecarregant els recursos computacionals del sistema de la víctima. Es genera mitjançant la saturació dels ports amb un flux d'informació, fent que el servidor es sobrecarregui i no pugui seguir prestant serveis.

L'atac no té com a objectiu el robatori de dades, i per tant no posa en perill les dades d'una aplicació web. Però sí que ocasiona l'impediment d'un ús normal pels usuaris de l'aplicació web, i segons quina pot deixar sense servei a uns quants milions d'usuaris. Molts cops es combina amb una atac MITM. Una forma de combatre aquest tipus d'atac són els sistemes distribuïts, els quals es basen en components distribuïts i replicats en diferents màquines virtuals com físiques en la xarxa. Tot i què s'ha de prendre mesures addicionals per poder prevenir, detectar i mitigar en cas d'atac. A través de tallafocs, IDS, limitació de la taxa de tràfic provinent d'un únic host, limitació del nombre de connexions concurrents al servidor, restricció de l'ús de l'ample de banda per hosts sospitosos i la realització d'un monitoratge constant del tràfic per detectar patrons d'atac.

Tot i què s'han vist diversos tipus d'atacs MITM i alguns de complementaris en aquest capítol, són solament una mostra dels més coneguts a mode d'exemple. A nivell de seguretat, cada dia que passa apareixen noves tècniques o vulnerabilitats que aprofiten els atacants. Per tant, la seguretat d'un servei prestat a través d'una xarxa pública com Internet, requereix una permanent monitorització durant tota la vida que el servei està operatiu, tant a nivell de programari, de maquinari com de xarxa.

# Capítol 3. Disseny d'aplicació

## 3.1 Tecnologia web per al desenvolupament de l'aplicació

Per al desenvolupament del prototip s'utilitzarà el llenguatge Java, ja que ha estat el llenguatge principal d'ús en el transcurs dels estudis que conclouen amb aquest projecte. El prototip a implementar és una aplicació web, per tant el client accedirà a l'aplicació a través d'una pàgina web des del seu navegador.

Per la construcció de la pàgina web es farà servir un *framework* anomenat JSF (JavaServer Faces). És una tecnologia i *framework* per aplicacions Java basades en web que simplifica el desenvolupament d'interfícies d'usuari en aplicacions Java EE. En la referència bibliogràfica [24] es descriu i es pot descarregar la tecnologia JSF. Aquesta tecnologia inclou els següents elements:

- Conjunt de *APIs* per representar components d'una interfície d'usuari i administrar el seu estat, manipular esdeveniments, validar entrada, definir un esquema de navegació de les pàgines i donar suport per la internacionalització i accessibilitat.
- Conjunt per defecte de components per la interfície d'usuari.
- Dues biblioteques d'etiquetes personalitzades per *JavaServer Pages* que permeten expressar una interfície *JavaServer Faces* dintre d'una pàgina JSP.
- Un model d'esdeveniments en el costat del servidor.
- Administració d'estats.
- *JavaBeans* administrats.

A més de l'ús d'aquest *framework* per la construcció de les interfícies web, s'utilitzarà com és lògic el llenguatge HTML 5 juntament amb fulles d'estils en cascada (CCS 3) per donar disseny a les pàgines web. En la pàgina web de w3schools de la referència bibliogràfica [25] es pot consultar la descripció i l'ús de diferents tecnologies web actuals.

Per la implementació de la lògica de negoci s'utilitzarà classes de Java POJO (Plain Old Java Object), que són classes simples i que no depenen de cap *framework* en especial. Addicionalment, per la base de dades s'utilitzarà la coneguda base de dades relacional MySQL que es pot descarregar i consultar documentació des de la pàgina web de la referència bibliogràfica [26].

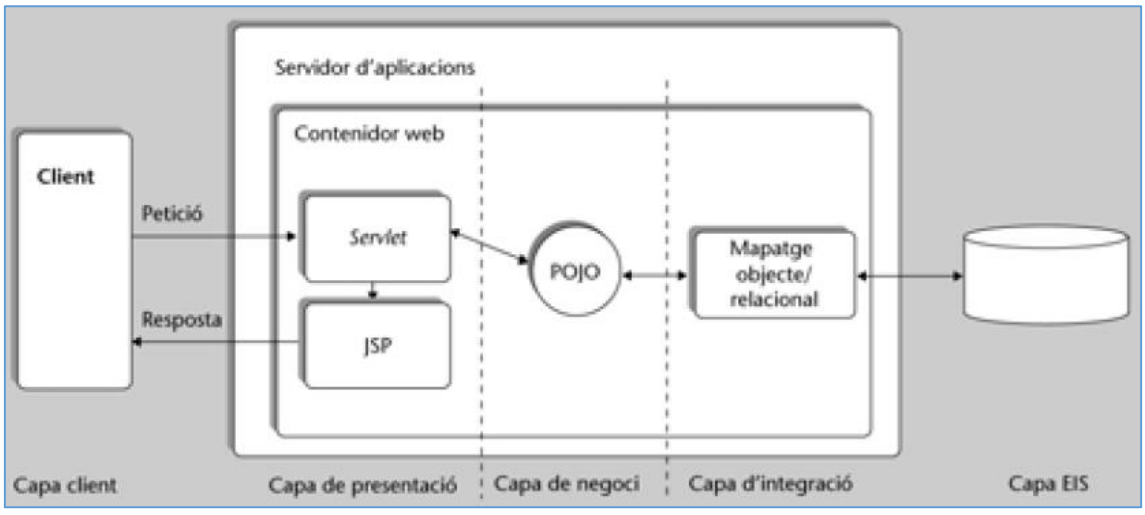
Tot el conjunt de components que es composarà el prototip a implementar es desplegarà en un servidor d'aplicacions web anomenat *GlassFish*. Aquest és un servidor d'aplicacions de programari lliure desenvolupat per *Sun Microsystems*, que implementa les tecnologies definides en la plataforma Java EE i permet executar aplicacions que segueixen aquesta especificació. Per més informació sobre el servidor d'aplicacions consultar la pàgina web de la referència bibliogràfica [27].

## 3.2 Definició de l'arquitectura de l'aplicació i dels agents que hi interactuaran

L'aplicació web segueix una arquitectura heterogènia. Per una part seguirà una arquitectura client-servidor, ja que tindrem al client que és l'usuari que accedirà a través del seu navegador web a l'aplicació fent peticions a un servidor. I per una altra part el servidor que satisfà totes les peticions que rep del navegador del client. D'una manera més formal, es pot dir que l'aplicació es modela com un conjunt de components servidors que ofereixen uns serveis, i un conjunt de clients que utilitzen aquest serveis.

Un altra arquitectura és l'arquitectura en nivells que representen una organització jeràrquica dels elements del sistema. Cada capa de l'arquitectura proporciona serveis als elements de la capa immediatament anterior, i se serveix dels serveis que li ofereixen els elements de la capa immediatament següent. En el desenvolupament de l'aplicació, els components que resideixen en la part del servidor estan distribuïts en diferents capes. Una capa de presentació que interactuarà amb el client, gestionant les peticions que rebi com servir-ne les respostes adequades al client. Una capa lògica de negoci que integra tots els elements encarregats de la implementació de la pròpia lògica de l'aplicació. I finalment, la capa de persistència formada pels elements encarregats de la gestió dels recursos d'informació externs, en el cas de l'aplicació una base de dades relacional.

A continuació es mostra un diagrama de com queden distribuïts els diferents elements o sistemes per l'aplicació web:



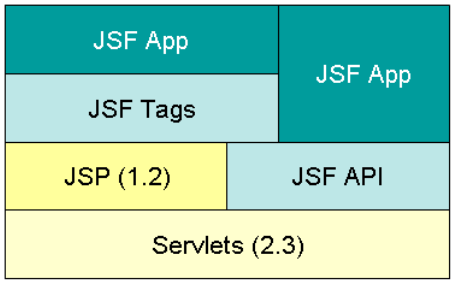

*Il·lustració 14. Arquitectura de programari de l'aplicació web.*

En el servidor d'aplicacions hi resideixen les tres capes lògiques, a nivell físic tots els components estan en la mateixa màquina, la separació solament és a nivell lògic. La capa client mostrada en la il·lustració anterior, representa a l'usuari que accedeix a l'aplicació a través d'un navegador web, el qual fa les peticions i rep les respostes del servidor. La capa EIS està formada per una base de dades relacional.

La capa de presentació és la encarregada de modelar les diferents interfícies que es presentaran a l'usuari. En aquesta capa s'aplica un patró de disseny anomenat Model-Vista-Controlador (MVC). En el mòdul *J2EE Una plataforma de components distribuïda* de l'assignatura Enginyeria

del programari de components i sistemes distribuïts de la UOC amb la referència bibliogràfica [1] es descriu el patró MVC junt l'arquitectura presentada en aquest apartat. El controlador i la vista residiran en aquesta capa, pel contrari el model resideix en la capa lògica. Addicionalment, en la capa de presentació s'aplicarà el patró de controlador únic (*FrontController*) per totes les operacions d'un component, i a la vegada s'aplica un esquema basat en el patró *Command*, fent que la implementació de cada acció estigui desacoblada entre elles però totes coordinades a través del controlador.

En el diagrama anterior la capa de presentació mostra una arquitectura d'aplicació web Model-2, a on els *servlets* i les *JSP* col·laboren per a implementar la capa de presentació de l'aplicació web. Però, per facilitar la construcció de l'aplicació web en la capa de presentació, s'utilitzarà un *framework* basat en l'arquitectura Model-2 anomenat JSF (*Java Server Faces*). La capa presentació o també anomenada capa web es crearà utilitzant o estenent les classes del JSF. Per tant, a l'utilitzar el *framework* es farà ús de les tecnologies mostrades en la següent il·lustració.




*Il·lustració 15. Tecnologia que es basa JSF.*

Addicionalment, aquest *framework* ens oferirà una sèrie d'avantatges alhora de la construcció web:

- Desacobla la capa de presentació de la capa de negoci en components separats.
- Simplifica i estandarditza la validació del paràmetres d'entrada.
- Simplifica la gestió del flux de navegació de l'aplicació.
- Proporciona un punt central de control.
- Permet un nivell molt alt de reutilització.
- Imposa la mateixa arquitectura per a tots els desenvolupaments.
- Simplifica moltes tasques repetitives.

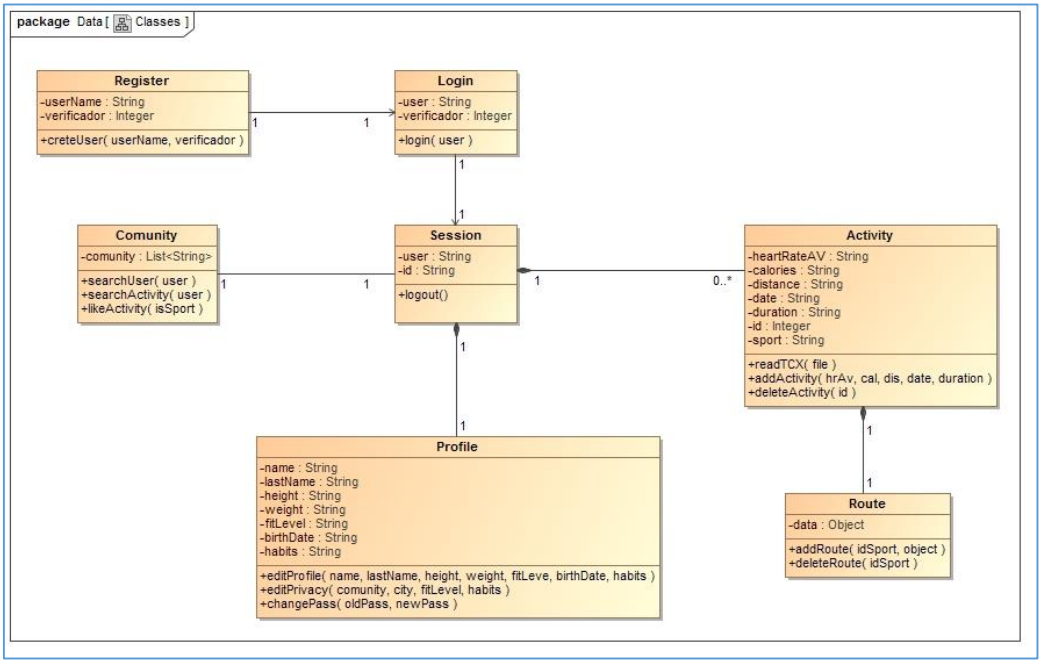
La capa de negoci s'implementarà amb classes Java POJO (*Plain Old Java Object*). En aquesta capa és a on s'implementarà el model del patró MVC. En la capa d'integració o persistència s'utilitzarà classes Java simples seguint el patró DAO (*Data Access Object*). El patró DAO servirà com una interfície comuna entre l'aplicació i la base de dades relacional.

Els agents identificats que interactuaran amb l'aplicació són els següents:

- Els usuaris a través dels seus navegadors web.
- La base de dades relacional (*MySQL*).
- El servidor d'aplicacions a on s'executa l'aplicació (*Glassfish*).

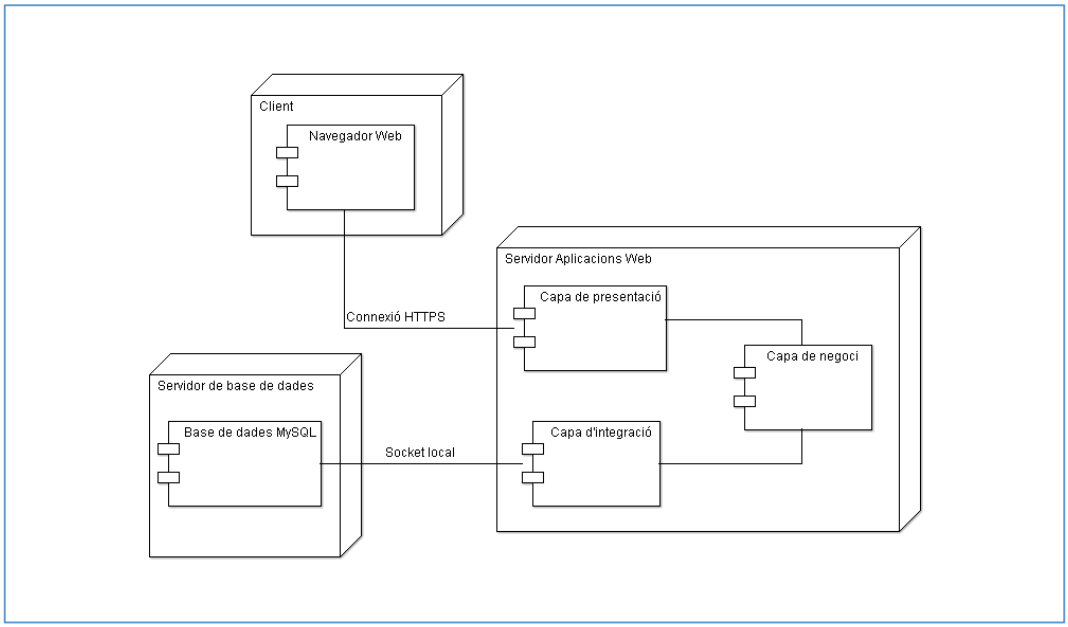
## 3.3 Disseny del diagrama de classes i de desplegament del l'aplicació

El diagrama de classes en alt nivell de l'aplicació és el definit en la següent il·lustració:




*Il·lustració 16. Diagrama de classes de l'aplicació.*

Addicionalment, s'ha dissenyat el diagrama de desplegament de l'aplicació en alt nivell. Tal i com s'ha definit l'arquitectura en l'apartat anterior, la capa intermèdia es realitzarà en tres grans blocs lògics per separat. La capa de presentació, la capa lògica de negoci i la capa d'integració.




*Il·lustració 17. Diagrama de desplegament de l'aplicació.*

## 3.4 Protocols criptogràfics

En aquest apartat s'especifiquen els diferents protocols criptogràfics seleccionats per a complir els requeriments de seguretat de l'aplicació. La selecció dels diferents protocols es basa en l'anàlisi i estudi realitzat d'un model d'atacant en l'apartat 2.1. L'apartat està dividit en tres grans blocs que fan referència a punts crítics dins del sistema.

El primer bloc tracta sobre les connexions segures i el seu objectiu és resoldre el problema de la falta de privacitat en les comunicacions entre client i servidor. Una comunicació a través de la xarxa d'Internet per si mateixa és pública, i totes les dades que es transfereixen a través d'ella estan exposades a la resta d'usuaris de la xarxa. En l'aplicació a desenvolupar això es vol evitar i per tant s'ha de fer ús d'alguna capa de protecció per assegurar la privadesa de les comunicacions entre client i servidor. Les connexions segures són l'eina bàsica per a portar-ho a terme.

Els segon bloc tracta sobre l'autenticació dels usuaris. L'aplicació a desenvolupar necessita conèixer d'alguna forma que el client amb el que ha establert comunicació realment és qui diu ser. Aquest apartat té l'objectiu d'estudiar algun mètode per poder autenticar la identitat d'un usuari de forma segura, i que el servidor de l'aplicació tingui la certesa de que és l'usuari legítim. Els mètodes d'autenticació feble o els forts entre altres donaran resposta a la necessitat de l'aplicació.

Com a últim bloc tenim la confidencialitat de les dades. La gestió de dades personals molt sensibles per part de l'aplicació a desenvolupar fa necessari prendre mesures per evitar qualsevol pèrdua d'informació dels usuaris. La base de dades de l'aplicació emmagatzemarà totes les dades dels usuaris i si no es pren cap mesura de seguretat qualsevol pot tenir accés a aquestes, tant des de l'exterior com del propi personal autoritzat que té accés a la base de dades. El protocols criptogràfics de xifratge de dades ens permetrà donar-hi aquesta capa de seguretat a les dades dels usuaris, impedint l'accés no autoritzat per tercers.

### 3.4.1 Connexions segures

L'assegurament de les connexions entre el client i el servidor de l'aplicació és fonamental per tal d'evitar certs atacs d'un possible atacant, entre altres destaquem l'escolta de xarxa, la suplantació web o un MITM.

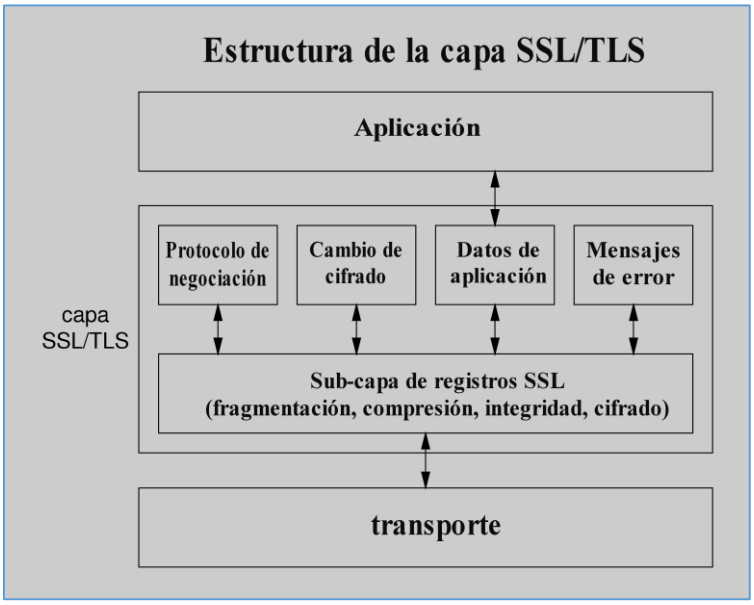
Per a portar a terme la connexió segura entre el client i el servidor web ens basarem en l'ús del protocol SSL/TLS. Amb ajuda del paquet d'eines d'administració i biblioteques relacionades amb la criptografia anomenat OpenSSL de codi obert, en la seva última versió estable i amb la vulnerabilitat *Heartbleed* corregida.

El protocol estàndard SSL va néixer en la dècada dels 90 de la mà de l'empresa Netscape Communication, amb la finalitat de proporcionar comunicacions segures en la xarxa. La primera versió d'aquest protocol àmpliament difosa i implementada va ser la 2.0. Temps més tard, Netscape va publicar la versió 3.0 amb molts canvis respecte l'anterior, actualment la versió 3.0 està en desús.

Addicionalment, com una evolució del protocol SSL va sorgir el protocol TLS (*Transport Layer Security*), el qual millora el protocol SSL en la protecció davant de nous atacs. L'especificació de TLS va ser elaborada per la IETF (*Internet Engineering Task Force*). La versió 1.0 del protocol TLS està publicada en el document RFC 2246. És pràcticament equivalent a SSL 3.0 amb petites diferencies, ocasionant que en certs contexts es considera el TLS 1.0 com si fos el protocol SSL 3.1.

La versió més moderna del protocol TLS amb les extensions recomanades pot considerar-se suficientment forta davant els atacs coneguts. Els atacs que vulneren la seva seguretat es centren especialment en enganyar a l'usuari amb la direcció a la que es connecta o amb el certificat digital que autentifica al servidor.

El protocol SSL/TLS s'implementa per damunt del protocol de transport i per sota del protocol d'aplicació. En el cas de les pàgines web segures, el protocol utilitzat és el *https* que indica que el protocol *http* està implementat per sobre del protocol SSL/TLS. En la il·lustració de la pàgina següent es mostra l'estructura de la capa SSL/TLS.




*Il·lustració 18. Estructura de la capa SSL/TLS.*

El protocol criptogràfic SSL/TLS ens ajuda a proporcionar confidencialitat, autenticitat, i integritat en una comunicació client-servidor:

- ***Confidencialitat***: els paquets que s'intercanvien el client i el servidor transporten les dades xifrades mitjançant claus simètriques (una clau per rebre i un altra per enviar), les quals s'acorden a l'inici de sessió. L'ús d'una xifra pública durant les negociacions a l'inici de sessió permet intercanviar de forma segura les claus de les xifres simètriques. També, a l'inici de les negociacions s'acorden les diferents xifres a utilitzar per la connexió entre el client i el servidor.

- ***Autenticació d'entitat***: l'ús de certificats digitals permet al client autenticar la identitat del servidor. Per l'autenticació del client, es pot realitzar a través del protocol SSL/TLS, però generalment la mateixa aplicació s'encarrega de fer-ho.

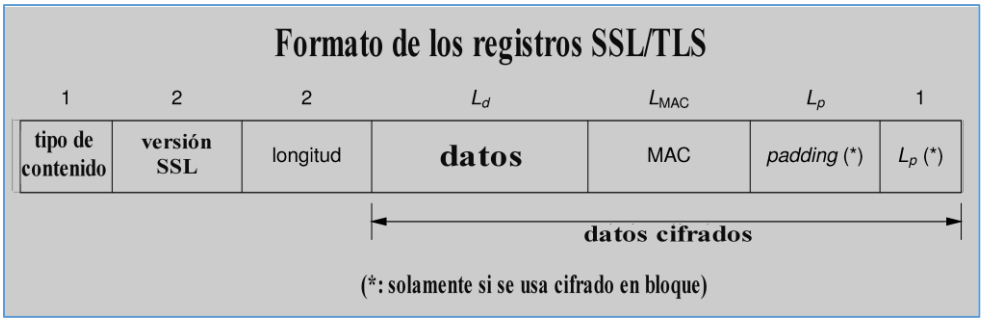
- ***Autenticació de missatge***: a part de xifrar les dades de cada paquet, aquest pot incorporar un codi MAC que servirà al destinatari per a comprovar que ningú a modificat el contingut de les dades. El codi MAC utilitza dues claus secretes, una per enviaments i una altra per rebre, i són pactades en la negociació inicial de la connexió.

L'establiment d'una comunicació segura amb el protocol SSL/TLS es composa de dos protocols:

- El protocol SSL/TLS *Handshake* o de negociació: ens ajuda a la negociació de paràmetres de seguretat per facilitar la confidencialitat, integritat, i autenticitat en una comunicació entre client i servidor.
- EL protocol SSL/TLS *Record* o de registres: especifica la forma d'encapsular les dades transmeses i rebudes, incloses les de negociació.

A continuació es descriu el protocol de registres:

La informació que s'intercanvia entre el client i el servidor en una connexió SSL/TLS s'empaqueta en registres, els quals tenen el format mostrat en la il·lustració següent:



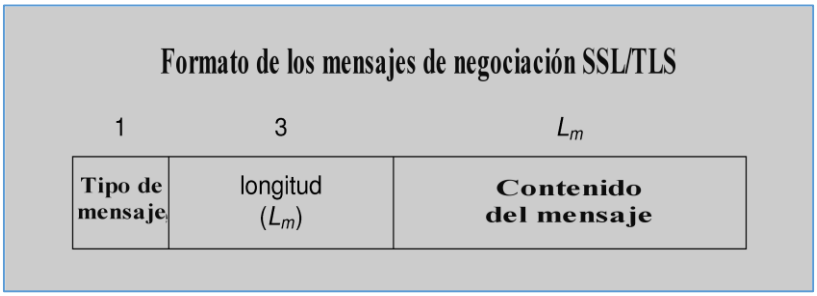

*Il·lustració 19. Format dels registres SSL/TLS.*

- El primer camp identifica el tipus de contingut de les dades, que poden ser: missatge del protocol de negociació, notificació de canvi de xifratge, missatge d'error o dades d'aplicació.
- El segon camp són dos bytes que indiquen la versió del protocol: un tres i un zero fa referència al protocol SSL 3.0, i un 3 i un u el protocol és el TLS 1.0.
- El tercer camp indica la longitud de la resta del registre: $L_d + L_{MAC}$ i en cas de xifratge de dades amb un algorisme en bloc $L_p + 1$.
- El quart camp són les dades que poden anar comprimides en cas d'utilitzar algun algorisme de comprensió.
- El cinquè camp és el codi d'autenticació (MAC): per calcular-ho s'utilitza la clau MAC, un nombre de seqüència implícit de 64bits (s'incrementa en cada registre) i el contingut del registre. La longitud del camp dependrà de l'algorisme utilitzat, pot ser zero en cas d'algorisme nul quan encara no se n'ha acordat cap.
- El sisè i setè camp són els bytes addicionals necessaris per tindre un nombre total que sigui múltiple de la longitud del bloc, en cas d'ús de xifratge en bloc de les dades. L'últim camp és un byte que indica el nombre de bytes addicionals inserits.

Una diferència entre SSL i TLS són en els bytes addicionals. En SSL el seu valor és indiferent excepte l'últim que indica el nombre, i en TLS tots els bytes addicionals tenen que tindre el mateix valor que l'últim.

El protocol de registres SSL/TLS s'encarrega de formar cada registre amb els seus camps corresponents, calcular el MAC, i xifrar les dades, el MAC i els bits addicionals amb els algorismes. En els primers passos de l'algorisme de negociació no existeix cap xifratge i autenticació per falta d'acord en els algorismes, tot i així al final tot el procés de negociació queda autenticat *a posteriori*.

Els missatges del protocol de negociació, com tots els missatges SSL/TLS, s'inclouen dintre del camp de dades dels registres SSL/TLS. La seva estructura és la següent:




*Il·lustració 20. Format dels missatges del protocol de negociació SSL/TLS.*

Un exemple de negociació bàsica del protocol d'autenticació de servidor (*SSL/TLS Handshake Protocol*) és el següent:

1. En general és el client qui inicia la negociació, però el servidor també pot enviar un *HelloRequest* al client per indicar-li que està preparat per la negociació. També, serveix per a què el servidor reinici una nova negociació.

2. El client envia un missatge *ClientHello* especificant la versió més alta del protocol TLS suportada, un nombre aleatori i una llista d'algoritmes d'autenticació, xifratge, MAC[19] i algorismes de comprensió.

3. El servidor respon amb un missatge *ServerHello* indicant la versió del protocol seleccionat, el qual és la versió més alta que suporta el client i el servidor, un nombre aleatori, els algoritmes seleccionats dels enviats pel client i el seu certificat digital mitjançant un missatge *Certificate* per poder-lo autenticar.

4. El client verifica el certificat rebut pel servidor a través d'una autoritat de confiança o mitjançant PKI. Llavors el client respon al servidor amb un missatge *ClientKeyExchange*, el qual conté una *PreMasterSecret* (un nombre secret) amb informació per a generar la clau de sessió. En el cas d'ús de l'algorisme RSA el missatge és xifra amb la clau pública del servidor i el nombre secret seleccionat pel client serà de 48 bytes de llargada.

5. Tant el client com el servidor utilitzen els nombres aleatoris intercanviats juntament amb la *PreMasterSecret*, aquesta última el servidor la recupera després de desxifrar-la amb ajuda de la seva clau privada. Amb aquests nombres intercanviats client i servidor calculen un secret comú anomenat *MasterSecret*. A partir d'aquest moment, totes les

[19] MAC (*Message Authentication Code*): és una porció d'informació utilitzada per a autenticar un missatge, el qual es calcula mitjançant l'aplicació d'una funció *hash* criptogràfica amb clau secreta k, que solament coneixen el remitent i el destinatari.

claus primàries com secundàries es deriven de la *MasterSecret* a través de la funció pseudoaleatòria establerta.

6. A continuació el client envia un registre *ChangeCipherSpec*, i indica al servidor que a partir d'aquest moment tota la informació intercanviada és autenticada i xifrada. Per a què sigui xifrada ho ha tingut que establir el servidor.

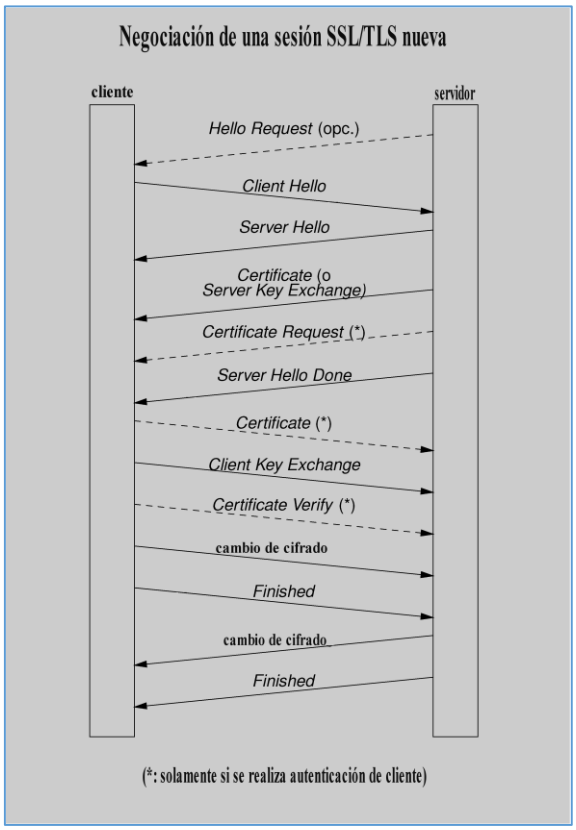
7. Per acabar, el client envia un missatge *Finished* signat i xifrat que conté un *hash* i MAC dels missatges negociats anteriorment.

8. El servidor rep el missatge *Finished* enviat pel client, el desxifrarà per verificar el *hash* i el MAC. Per finalitzar la connexió amb èxit la verificació del servidor té que ser satisfactòria, en cas contrari no s'estableix la connexió.

9. Si en el pas anterior s'ha produït una verificació satisfactòria, el servidor retorna al client un *ChangeCipherSpec* indicant al client que a partir d'aquest moment totes les dades enviades estaran signades i xifrades si és el cas. Seguidament el servidor també envia un missatge *Finished* signat i xifrat, contenint un *hash* i MAC dels missatges negociats anteriorment. Aquest missatge serà validat pel client.

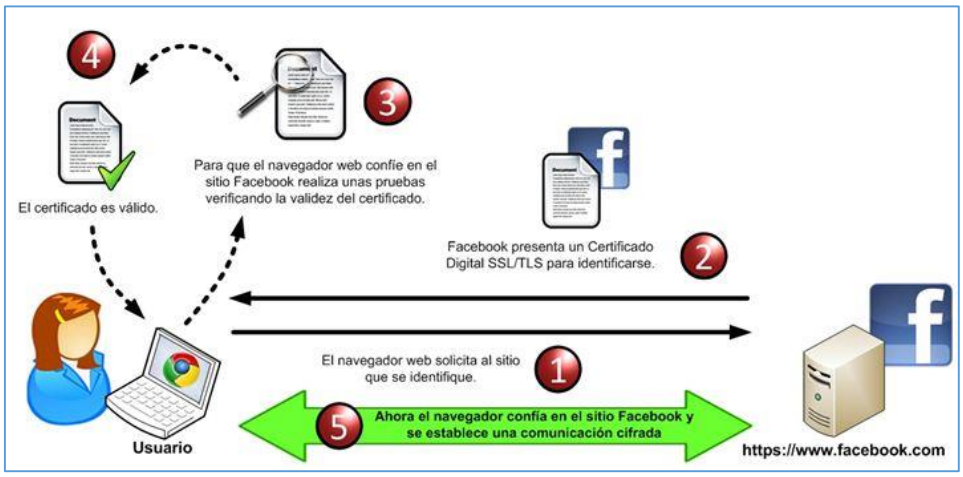
10. En aquest punt finalitza la fase de negociació, permetent l'intercanvi de missatges entre el client i el servidor autenticats, i xifrats si així s'ha establert per part del servidor.

Seguidament es mostra un diagrama que resumeix els missatges intercanviats durant la fase de negociació SSL/TLS en una sessió nova:




*Il·lustració 21. Diagrama d'intercanvi de missatges de la fase de negociació SSL/TLS.*

Com a exemple il·lustratiu del funcionament general del protocol SSL/TLS en que es basarà a l'aplicació web a desenvolupar en el projecte, es mostra els següent dibuix amb l'exemple de la realització d'una connexió segura a la pàgina web de la xarxa social FaceBook:




*Il·lustració 22. Funcionament general del protocol SSL/TLS.*

Segons les versions dels protocols SSL i TLS tindrem a l'abast un conjunt d'algorismes per a portar a terme l'intercanvi de claus, el xifratge de dades i la integritat de les dades. En les taules següents es mostren els diferents algorismes disponibles per les diferents versions, en el cas del xifratge a més s'indica si és segur o no.

| Algorisme | SSL 2.0 | SSL 3.0 | TLS 1.0 | TLS 1.1 | TLS 1.2 | Estat |
|---|---|---|---|---|---|---|
| RSA | Sí | Sí | Sí | Sí | Sí | |
| DH-RSA | No | Sí | Sí | Sí | Sí | |
| DHE-RSA (forward secrecy) | | | | | | |
| ECDH-RSA | No | No | Sí | Sí | Sí | |
| ECDHE-RSA (forward secrecy) | | | | | | |
| DH-DSS | No | Sí | Sí | Sí | Sí | Definit per a TLS 1.2 en RFC |
| DHE-DSS (forward secrecy) | | | | | | |
| ECDH-ECDSA | No | No | Sí | Sí | Sí | |
| ECDHE-ECDSA (forward secrecy) | | | | | | |
| DH-ANON (insegur) | No | No | Sí | Sí | Sí | |
| ECDH-ANON (insegur) | No | No | Sí | Sí | Sí | |
| GOST R 34.10-94/34.10-2001 | No | No | Sí | Sí | Sí | Proposat en esborranys RFC |

*Taula 2. Algorismes disponibles per autenticació i intercanvi de claus segons la versió SSL/TLS.*

| Algorisme de xifrat | SSL 2.0 | SSL 3.0 | TLS 1.0 | TLS 1.1 | TLS 1.2 |
|---|---|---|---|---|---|
| AES CBC | — | — | depèn | Segur | Segur |
| AES GCM | — | — | — | — | Segur |
| AES CCM | — | — | — | — | Segur |
| Camellia CBC | — | — | depèn | Segur | Segur |
| Camellia GCM | — | — | — | — | Segur |
| SEED CBC | — | — | depèn | Segur | Segur |
| ChaCha20+Poly1305 | — | — | — | — | Segur |
| IDEA CBC | insegur | depèn | depèn | Segur | — |
| 3DES CBC | insegur | depèn | depèn | depèn | depèn |
| DES CBC | insegur | insegur | insegur | insegur | — |
| RC2 CBC | insegur | insegur | insegur | insegur | — |
| RC4 | insegur | insegur | insegur | insegur | insegur |

*Taula 3. Algorismes disponibles i nivell de seguretat per al xifratge per blocs segons la versió SSL/TLS.*

| Algorisme | SSL 2.0 | SSL 3.0 | TLS 1.0 | TLS 1.1 | TLS 1.2 | Estat |
|---|---|---|---|---|---|---|
| **HMAC-MD5** | Sí | Sí | Sí | Sí | Sí | Definit per a TLS 1.2 en RFC |
| **HMAC-SHA1** | No | Sí | Sí | Sí | Sí | |
| **HMAC-SHA256/384** | No | No | No | No | Sí | |
| **AEAD** | No | No | No | No | Sí | |
| **GOST 28147-89 IMIT** | No | No | Sí | Sí | Sí | Proposat en esborranys de RFC |
| **GOST 34.11-94** | No | No | Sí | Sí | Sí | |

*Taula 4. Algorismes disponibles per la integritat de les dades segons la versió SSL/TLS.*

Per tant, un cop coneguts els diferents algorismes que pot implementar el protocol SSL/TLS en les diferents versions, i el nivell de seguretat en els casos dels algorismes pel xifratge, es pot establir quin d'aquests és la millor opció a aplicar en el desenvolupament de l'aplicació.

Primer es descriu la propietat *Forward secrecy* dels sistemes criptogràfics, la qual fa referència a la garantia que una clau de sessió derivada d'un conjunt de claus públiques i privades no es veurà compromesa si una de les claus privades es veu compromesa en un futur. Sense aquesta propietat, si la clau privada del servidor es dóna a conèixer es veurien compromeses les sessions xifrades futures com les anteriors que utilitzaven aquesta clau.

La millor opció per al desenvolupament de l'aplicació és la selecció del protocol TLS a partir de la versió 1.1. Per garantir la propietat de *Forward secrecy* s'ha d'exigir l'ús de l'intercanvi de claus Diffie-Hellman efímeres per establir les claus de sessió. Per tant, per l'autenticació es pot utilitzar la xifra pública RSA junt l'algorisme DHE.

Els mecanismes de gestió de sessió en el servidor poden afectar a la propietat *Forward secrecy*. El protocol SSL/TLS compta amb dos sistemes per escurçar el temps de negociació quan el client es connecta al servidor, memoritzant els seu estat. Els mecanismes són *Sessions IDs* i *Session tickets*. El mètode d'us de *tickets* de sessió està actiu per defecte en OpenSSL, i és el servidor el que signa i envia l'estat xifrat al client. El client rep un *ticket* de sessió del servidor i un cop validat el fa servir per a un altre *front-end* diferent, així evita de començar de nou les negociacions de connexió amb un servidor.

Aquests *tickets* que té els clients són com una mena d'entrades temporals de sessió per estalviar temps en les connexions. Els *tickets* es generen i es xifren de forma independent, i TLS utilitza un algorisme molt més feble. Per tant, l'ús de *tickets* de sessió ocasiona que el secret sigui tan fort com el del xifrat durant l'intercanvi d'aquests *tickets*, independentment del xifrat de la informació real. Addicionalment, la informació per a xifrar els *tickets* no canvia molt en períodes de temps llargs, i es solen generar en la memòria del servidor. Si es disposa d'un sol servidor SSL que maneja tota la informació, això pot ocasionar que un mateix *ticket* de sessió es mantingui en memòria des de l'arrencada del servei fins que aquest sigui reiniciat. Per això existeix una recomanació en la construcció dels *tickets* de sessió (RFC 5077), que aconsella l'ús de l'algorisme AES128 en mode CBC i del HMAC-SHA-256.

Amb tot el descrit fins ara podem concloure que la suite de xifrat recomanada a utilitzar en el servidor de l'aplicació és ***DHE_RSA_WITH_AES_CBC_SHA***.

En l'actualitat els navegadors més importants que molts usuaris utilitzen suporten perfectament totes les versions del protocol TLS. Per part de l'aplicació s'hauria de recomanar als usuaris quins navegadors són els que poden fer ús per disposar d'accés a l'aplicació, i així poder garantir-ne la seguretat. El protocol TLS ofereix unes mesures de seguretat que milloren la seguretat respecte als protocols anteriors:

- Protecció contra una degradació del protocol a una versió anterior o a un conjunt de xifrat més dèbil.
- Ús d'un nombre de seqüència en els registres d'aplicació posteriors utilitzat en els codis d'autenticació de missatges (MAC).
- Utilització d'una funció *hash* millorat amb una clau.
- Enviament d'un resum de tots les missatges intercanviats durant el protocol de negociació al finalitzar aquest.
- La funció pseudoaleatòria divideix les dades d'entrada en dos i processa cada part amb un algorisme *hash* diferent, seguidament amb el resultat de les dues parts fa una operació OR exclusiva per a crear el MAC. Això proporciona una protecció extra en cas que algun dels dos algorismes emprats resulti ser vulnerable.

### 3.4.2 Autenticació dels usuaris

Quan un usuari és connecta a l'aplicació web, el servidor s'autentica a través d'un certificat digital mitjançant el protocol SSL/TLS que envia a l'usuari. En canvi, l'usuari no utilitza el mateix mètode per autenticar-se davant el servidor, si no que s'usarà un mètode d'autenticació de repte-resposta.

Tot i què ja s'implementa el protocol de connexió segura SSL/TLS, que estableix un canal segur per a on intercanviar informació tal com podria ser una contrasenya, s'ha preferit l'ús d'una tècnica d'autenticació forta com són els protocols de repte-resposta. Qualsevol atac que posés en perill la connexió segura establerta entre client i servidor, encara ens quedaria una capa de seguretat per impedir a l'atacant que s'apoderés de les credencials de l'usuari.

El protocol SRP (*Secure Remote Password*) permet que un usuari demostri el coneixement d'una contrasenya sense que en cap moment qui la comprova tingui coneixement de la mateixa o d'un valor derivat de la mateixa després de l'aplicació d'una funció resum. La pàgina web de la referència bibliogràfica [28] mostra una descripció completa d'aquest protocol per Thomas Wu del departament de ciències de la computació de la Universitat de Stanford (EUA). Aquest protocol va ser creat en 1998 en la Universitat de Stanford i és un estàndard que forma part del protocol TLS a través del RFC5054.

El SRP estableix entre la contrasenya de l'usuari i un verificador amb unes propietats concretes una relació asimètrica similar a les xifres de clau pública. D'aquesta forma la contrasenya de l'usuari mai s'envia al servidor, ni tan sols xifrada o amb un resum. Amb l'ús d'aquest protocol l'usuari pot demostrar al servidor el coneixement de la contrasenya sense revelar-la. En el món de la criptografia aquest fet se'l coneix com a "proves de coneixement nul o de coneixement zero".

Per al desenvolupament del prototip es pot utilitzar una versió bàsica d'aquest protocol que alhora també suporta una autenticació bàsica per al servidor. La descripció del protocol a implementar es compon de dos punts: el registre de l'usuari i l'autenticació d'aquest davant el servidor.

*Registre del l'usuari:*

L'usuari abans d'interaccionar amb l'aplicació web s'haurà de registrar per establir un nom d'usuari i una contrasenya. Aplicant el protocol SRP, en la part del client es calcula:

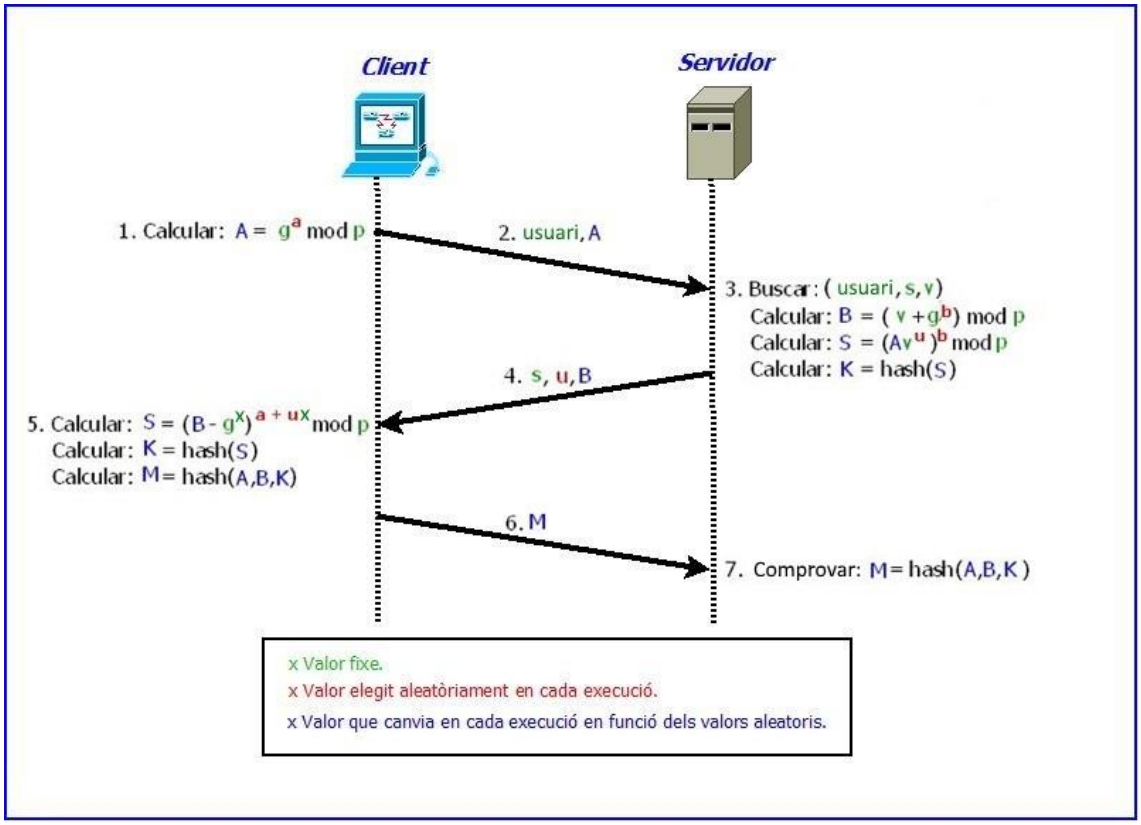
$x = resum ( contrasenya + s )$, on $s$ és un valor aleatori utilitzat com a bits de sal.

$v = g^x \bmod p$, on $p$ es un nombre primer suficientment gran i $g$ es un generador de $Z_p$.

Per a completar el registre un cop calculats els valors anteriors, el client envia $s$ i $v$ al servidor. Aquest parell de valors enviats al servidor se'l coneix com el verificador de l'usuari. Fent una analogia amb les xifres de clau pública la contrasenya de l'usuari és la clau privada i el parell ($s$,$v$) és la clau pública.

*Autenticació de l'usuari:*

Un usuari a l'hora d'autenticar-se en l'aplicació web ha de demostrar que coneix la contrasenya a partir d'executar el protocol de la següent il·lustració:




*Il·lustració 23. Diagrama del protocol SRP d'autenticació.*

Per tant, cada cop que un usuari vol autenticar-se en l'aplicació es realitzen els següents passos:

1. Calcula el valor $A = g^a \bmod p$ i l'envia al servidor juntament amb el nom d'usuari. El valor $a$ s'obté aleatòriament en cada autenticació i sempre és diferent en cadascuna d'elles.
2. El servidor busca el verificador de l'usuari (parell $s$,$v$) i calcula $B$, $S$ i $K$ tal i com s'especifica en el pas 3 de la il·lustració anterior, utilitzant uns valors $b$ i $u$ aleatoris i diferents en cada execució del protocol. Els bits de sal $s$ juntament amb els valors $u$ i $B$ són enviats a l'usuari.
3. L'usuari és l'única persona que coneix $x$ (depèn de la contrasenya) i que s'ha definit durant el registre d'aquest. Per tant, l'usuari serà l'únic capaç de derivar l'element $M$ tal com es mostra en la il·lustració en el pas 5. El valor $M$ calculat per l'usuari s'envia al servidor, i aquest podrà contrastar el resum rebut amb el generat per ell. A partir d'aquí el servidor sap amb tota certesa que l'usuari coneix la contrasenya correcta, tot i què el servidor no la coneix.

Com a funció *hash* per al protocol presentat s'utilitzarà l'algorisme SHA-256, ja que és un algorisme molt utilitzat i amb una seguretat contrastada. Per al bits de sal s'utilitzarà una cadena de bytes aleatòria d'almenys 32 caràcters.

#### 3.4.3.1 Creació, fortalesa i canvi de contrasenya

L'elecció de la contrasenya per part de l'usuari no és un tema trivial, i s'ha de definir uns criteris bàsics per assegurar un nivell seguretat adient. En la pàgina de procediments de Microsoft Windows de la referència bibliogràfica [29] es donen una sèrie de consells per a crear una contrasenya segura. De forma abreviada una contrasenya segura compleix unes condicions com les següents:

- Longitud mínima de vuit caràcters.
- No esta formada pel nom de l'usuari o pel nom real d'aquest.
- No conté una paraula completa.
- És significativament diferent a altres contrasenyes anteriors.
- Es composa per caràcters de cada una de les següents categories:
  - Caràcters (lletres majúscules i lletres minúscules).
  - Nombres (conjunt dels nombres naturals).
  - Símbols especials i espais (` ~ ! @ # $ % ^ & * ( ) _ - + = { } [ ] \ | : ; " ' < > , . ? /).

Per això, quan un usuari es dóna d'alta en el sistema o canviï la contrasenya s'exigirà que la nova contrasenya compleixi els quatre primers punts anteriors, i a més almenys dos de les tres categories del cinquè punt.

Per a calcular la fortalesa d'una contrasenya s'ha de parlar d'entropia, aquest terme s'engloba dins del camp de la teoria de la informació i intenta mesurar la incertesa d'un valor aleatori. En la pàgina *Password strength* de la Viquipèdia de la referència bibliogràfica [30] s'explica amb molt més detall el concepte de contrasenya forta.

La fortalesa d'una contrasenya es mesura en bits d'entropia i es calcula a través d'una fórmula matemàtica basada en l'entropia de Shannon que mesura la quantitat d'informació d'un valor aleatori. Un atacant de mitjana tindrà que provar almenys la meitat de les combinacions possibles d'una contrasenya abans de trobar la correcta. L'addició d'un bit a l'entropia d'una contrasenya duplica el nombre de conjectures necessàries, fent que la tasca d'un atacant sigui dues vegades més difícil. La següent il·lustració mostra la fórmula de Shannon per poder calcular l'entropia d'una contrasenya, on L és la longitud de la contrasenya i N és el conjunt de possibles símbols de la contrasenya.

$$H = \log_2 N^L = L \log_2 N = L \frac{\log N}{\log 2}$$

*Il·lustració 24. Fórmula de Shannon pel càlcul de l'entropia d'una contrasenya.*

Com a exemple es mostra la següent taula que busca l'entropia de dues contrasenyes:

| Contrasenya | Conjunt de símbols (N) | Entropia (bits) |
|---|---|---|
| ***abcdefghijklmnopq*** | 28 (alfabet minúscules) | $17 * \left(\frac{\log 28}{\log 2}\right) = 82$ |
| ***1234AbCd*** | 10+28+28 (alfabet minúscules, majúscules i números) | $8 * \left(\frac{\log 66}{\log 2}\right) = 49$ |

*Taula 5. Exemple de càlcul de l'entropia de contrasenyes.*

En la taula de la pàgina anterior es pot observar com tot i què la segona contrasenya té números i lletres majúscules i minúscules no arriba a ser tant forta com la primera que solament conté lletres minúscules.

Addicionalment, a part de complir amb els requisits en la construcció de les contrasenyes també és recomanable aplicar una caducitat a aquestes. Una bona pràctica és que passats 90 dies s'obligui a l'usuari a canviar la contrasenya per una nova. L'aplicació és l'encarregada de fer les comprovacions adients sobre les contrasenyes que elegeixen els usuaris, i portar a terme la política de caducitat d'aquestes.

#### 3.4.3.2 Verificació en dues passes

La verificació en dues passes ens permet afegir un segon factor de seguretat en l'autenticació dels usuaris. És un mètode que en l'actualitat està en molt ús per part de les aplicacions web de la grans empreses. Es tracta d'utilitzar una segona clau d'un sol ús que s'envia al client, o que s'obté d'una tercera aplicació, en el moment de l'autenticació.

Una de les vies més comunes per rebre la segona clau és a través de missatges de text en el mòbil. En el cas de la plataforma web *iCloud* de Apple, utilitzen directament els dispositius mòbils *iOS* per rebre aquesta clau temporal. En la pàgina web de suport d'Apple de la referència bibliogràfica [31] es detalla una sèrie de preguntes freqüents sobre el seu sistema de verificació en dues passes.

Com alternativa existeix l'aplicació mòbil d'autenticació *Google Authenticator*, que permet a aplicacions que no implementin l'enviament de missatges o que no es vulgui dependre de la xarxa mòbil, l'obtenció de la segona clau. En la referència bibliogràfica [32] es pot trobar la pàgina web per la seva instal·lació i configuració. Per a utilitzar-la s'ha de vincular l'aplicació mòbil amb l'aplicació que té implementada la verificació en dues passos. Això s'aconsegueix a través d'un codi QR que l'aplicació a protegir genera i que es llegeix amb el mòbil. Un cop que *Google Authenticator* té vinculada l'aplicació, identificada amb un nom per relacionar-la fàcilment per part de l'usuari, genera un codi temporal que cada certs segons canvia. En el moment que l'usuari accedeix a una aplicació que li demana aquest segon codi, simplement obre l'aplicació mòbil, mira el codi que li mostra en aquell mateix moment i l'introdueix a l'aplicació. En la il·lustració següent es mostra la pantalla de l'aplicació *Google Authenticator* i en la que es pot veure l'exemple d'aplicacions com *Dropbox*, *WordPress* i *Google* on podem utilitzar aquest mètode de verificació de dues passes. Al costat de cada codi, identificat amb l'aplicació al qual correspon, hi ha una mena de temporitzador que indica el temps restant per a què el codi canviï.


*Il·lustració 25. Pantalla de l'aplicació Google Authenticator.*

### 3.4.3 Confidencialitat de les dades

En el desenvolupament de l'aplicació web un dels punt crítics són les dades dels usuaris. Aquestes pel tipus d'aplicació són altament sensibles, per això s'ha de tenir en compte tot un conjunt de mesures de seguretat per impedir una possible fuga d'informació. L'ús de protocols criptogràfics en la gestió de la base de dades és l'eix fonamental per a protegir les dades de terceres persones.

Quan es parla de protegir les dades de terceres persones, a part de referir-nos a possibles atacants aliens al sistema, també s'ha de pensar en protegir les dades contra les mateixes persones que tenen un accés autoritzat a la gestió (*insider*) o a l'ús de l'aplicació web com els usuaris.

S'ha de pensar que el mateix administrador de l'aplicació web segurament tindrà accés a la base de dades, o si és el cas el propi administrador de base de dades. Tot i què segurament a nivell contractual i de diferents normatives legals els administradors estan obligats a mantenir el secret i la privadesa de les dades, s'han de prendre mesures tècniques per a què les dades dels usuaris puguin ser totalment confidencials sense que cap persona les pugui arribar a veure. D'aquesta forma s'assegura un nivell de protecció màxim, i els mateixos administradors del sistema es treuen una important responsabilitat.

Per part dels usuaris, aquests posseïssin accés a l'aplicació web per al seu ús i emprant algun mètode d'autenticació segur s'autentiquen en el servidor. Un usuari amb intencions malicioses podria aprofitar-se del seu accés al servidor de l'aplicació per a portar a terme un atac. Per això és important acotar al màxim l'accés a la base de dades per part dels usuaris, i que solament puguin accedir i/o modificar el que realment necessiten.

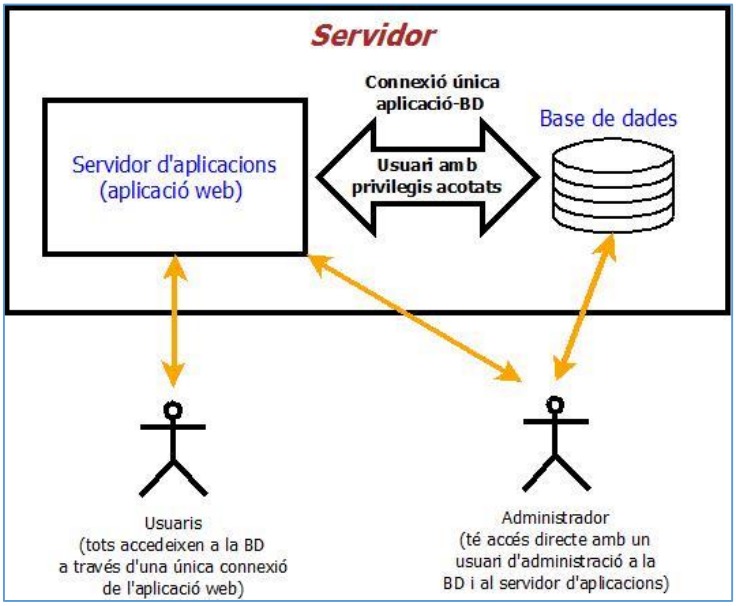
En els següents subapartats s'especifiquen els diferents aspectes de seguretat que s'aplicaran a la base de dades de l'aplicació web, amb ajuda de tècniques de seguretat i protocols criptogràfics.

#### 3.4.2.1 Gestió de permisos

Un dels punts importants és la gestió de permisos en la base de dades. Un usuari a l'hora d'interaccionar amb l'aplicació web necessita emmagatzemar dades personals com les dades dels diferents esports que ha practicat. Per tant, l'usuari pot crear i/o modificar les dades de les diferents taules de la base dades.

Per seguretat, l'usuari solament tindrà els permisos justos per poder accedir a l'esquema de taules de la base de dades corresponent a l'aplicació, i amb permisos solament per inserir, actualitzar o esborrar dades de les taules. En la base de dades es donarà d'alta un usuari amb els privilegis citats anteriorment, i aquest usuari utilitzarà l'aplicació web per connectar-se a la base de dades. En principi els usuaris seran tots del mateix perfil i no existiran diferents perfils, com podria ser un perfil *premium* per als usuaris que paguessin una quota i disposessin d'uns serveis extres. Així, tots les usuaris utilitzaran la mateixa estructura de taules amb els mateixos privilegis.

Per tant, un usuari no tindrà accés directe a la base de dades, sinó que serà l'aplicació web la que accedirà a la base de dades. Així, tots els usuaris de la web en realitat faran servir un mateix usuari de la base de dades, i és aquest usuari el que tindrà uns permisos restringits. En la il·lustració següent es mostra un diagrama dels diferents rols i de les seves formes d'accés a la base de dades:




*Il·lustració 26. Diagrama d'accés a la base de dades.*

L'administrador accedeix directament a la base de dades, físicament a través del servidor o de forma remota segura (per exemple SSH), i disposa d'un usuari amb privilegis d'administrador. En canvi, un usuari accedeix a través del seu navegador web a l'aplicació web, i aquesta a la vegada disposa d'una única connexió amb la base de dades a través d'un usuari amb els privilegis acotats. L'usuari té un accés indirecte a la base de dades, mitjançant l'aplicació web i amb els privilegis acotats i accés únicament a taules d'un esquema concret.

### 3.4.2.2 Xifratge de dades

Un cop s'ha limitat l'accés i l'ús en la base de dades, és el moment d'assegurar les dades de tal forma que cap persona que arribés a accedir a la base de dades les pugui arribar a llegir de forma clara. La solució és l'aplicació d'un algorisme criptogràfic per a xifrar totes les dades que s'emmagatzemen d'un usuari.

Per aquest cas l'ús de xifres simètriques és el més adequat, ja que proporcionen molta rapidesa a l'hora de xifrar i desxifrar comparat amb les xifres asimètriques o de clau pública. Les xifres simètriques posseeixen una clau única que és coneguda tant per l'emissor com pel receptor. Per al cas de l'aplicació es pot utilitzar una clau específica per al xifratge de dades diferent a la contrasenya que utilitza l'usuari per autenticar-se, i que es pot generar quan un usuari es dóna d'alta en el sistema. La gestió d'aquesta clau es veu més endavant en l'apartat de gestió de claus.

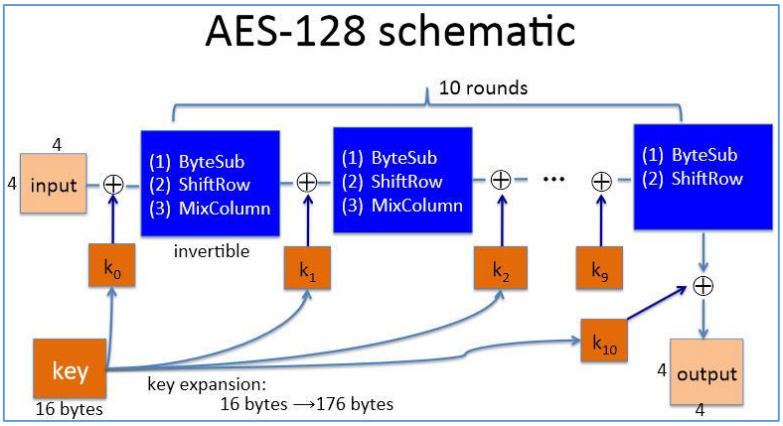
Per a la xifra de clau compartida o simètrica seleccionem les xifres de bloc, aquestes actuen sense memòria i el text xifrat només pot dependre del text en clar i de la clau. Això últim ocasiona que dos textos en clar iguals es xifren de la mateixa forma quan s'utilitza la mateixa

clau. S'ha de tenir en compte aquest fet per poder-ho corregir i que el sistema no sigui vulnerable. En l'actualitat la xifra de clau compartida de bloc més utilitzada i aconsellada és AES (*Advanced Encryption Standard*). La pàgina web *Advanced Encryption Standard* de la Viquipèdia de la referència bibliogràfica [23] mostra una breu descripció i historia d'aquest algorisme.

Per a usar els xifratges de bloc en textos llargs i que poden contenir determinats patrons que es repeteixen s'ha d'aplicar algun dels següents modes de funcionament: CBC, CFB o OFB. De tots aquests modes el més utilitzat és el CBC (*Cipher Block Chaining*). La pàgina web *Block cipher mode of operation* de la Viquipèdia amb referència bibliogràfica [33] descriu amb detall els diferents modes d'operació. El mode CBC consisteix en l'encadenament dels blocs per al xifratge, ocasionant una dependència del xifratge de cada bloc amb el bloc immediatament anterior. Abans de xifrar cada bloc s'aplica l'operació XOR amb el bloc anterior xifrat, per al primer bloc a xifrar s'utilitza un vector d'inicialització. A causa de l'encadenament de tots el blocs xifrats, l'últim bloc pot actuar com a signatura digital o *checksum* de la resta de blocs, permetent certificar que el xifratge no ha estat alterat. En aquest mode d'operació un error en el text xifrat tan sols afecta al desxifrat de dos blocs. Per tant, aquest mode ens protegeix respecte a la substitució de blocs i és resistent als errors, per contra la seva principal desavantatge és ser seqüencial i no poder funcionar en paral·lel.

L'algorisme AES ofereix tres versions (AES-128, AES-192 i AES-256) en funció de la longitud de la seva clau. A priori una clau de major longitud ofereix més seguretat, però autoritats en la matèria com Bruce Schneier[20] recomanen utilitzar millor AES-128 en lloc de AES-256. La grandària del bloc de AES és fixa independentment de la longitud de la seva clau i és de 128 bits. AES és ràpid tant en programari com en maquinari, és relativament fàcil d'implementar, requereix poca memòria i opera en una matriu de quatre per quatre bytes.

Com a conclusió podem determinar que l'algorisme de xifratge més adequat per a xifrar les dades de l'aplicació és AES-128 en mode CBC, utilitzant una clau mestra generada de forma aleatòria en el moment que un usuari es dóna d'alta en el sistema. El xifratge de les dades es realitzarà en la part del client, d'aquesta forma les dades que rep o envia el servidor sempre estan xifrades i fent que el servidor no conegui en cap moment la clau de xifrar dades. En l'esquema següent es mostra el funcionament bàsic de l'algorisme AES-128 que s'utilitzarà en l'aplicació:




*Il·lustració 27. Diagrama de funcionament de AES-128.*

[20] Bruce Schneier: és un criptògraf expert en seguretat informàtica i escriptor. És l'autor de diversos llibres de seguretat informàtica i criptografia, i és el fundador i cap tecnològic de *Counterpane Internet Security*.

#### 3.4.2.3 Gestió de claus

L'autenticació de l'usuari en el servidor de l'aplicació es realitzarà a través d'un mètode d'autenticació forta, el qual es detalla en l'apartat corresponent. La base de dades emmagatzemarà el verificador d'usuari (parell de valors s,v) per a cada usuari. Segons l'algorisme que es detalla en l'apartat 3.4.2 d'autenticació, els usuaris mai enviaran les seves contrasenyes al servidor i aquest no tindrà mai cap coneixement de cap contrasenya. Això afegeix una capa de seguretat en cas que la connexió segura amb el servidor es vegi compromesa. Per tant, el servidor sempre treballarà amb els verificadors dels usuaris per autenticar als usuaris en el sistema.

En l'apartat anterior sobre el xifratge de dades (apartat 3.4.2.2) s'especifica que la clau a utilitzar en el xifratge de les dades és una segona clau generada quan l'usuari es dóna d'alta en el sistema. Aquesta clau s'emmagatzemarà en l'ordinador del client i per seguretat es xifrarà amb AES amb la contrasenya de l'usuari.

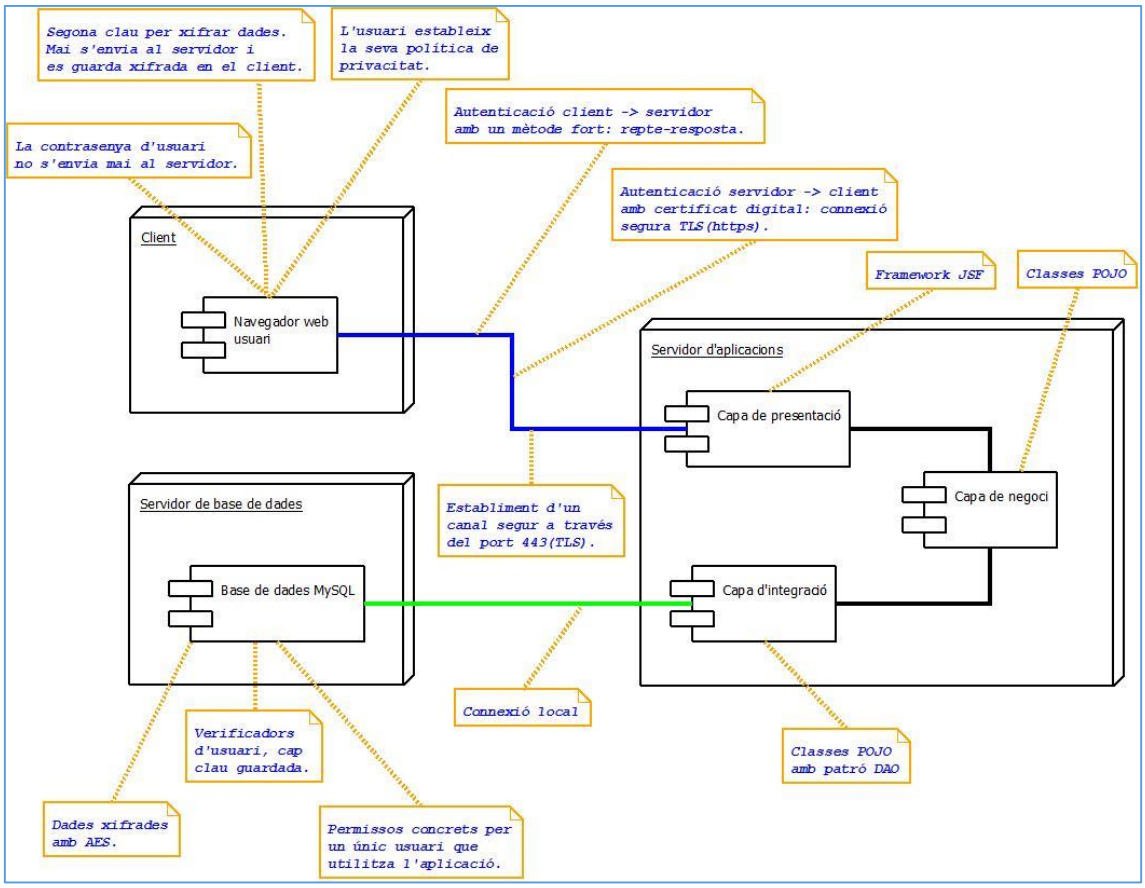
Tant la contrasenya de l'usuari com la clau de xifrar dades mai s'envien al servidor. El servidor no té constància de cap de les dues claus mai, ni tan sols en el procés de registre de l'usuari. El disposar de dos claus diferents, una per autenticar i l'altra per xifrar dades, possibilita que un canvi de contrasenya d'usuari no afecti en res a les dades xifrades de l'usuari de la BD, únicament en el xifratge de la clau de xifrar dades en la part del client.

#### 3.4.2.4 Privacitat de les dades

Els usuaris de l'aplicació web a més de poder gestionar les seves pràctiques esportives també poden participar en una comunitat social, on podran visualitzar els resultats d'activitats esportives i certes dades personals d'altres usuaris. Hi haurà dades personals que per defecte sempre seran confidencials, però d'altres dependrà de la decisió de l'usuari si vol mostrar algunes o cap dada en la comunitat. Per tant, l'aplicació haurà de llegir algunes dades concretes de la base de dades que no estaran xifrades per mostrar-les en el panell de la comunitat. S'establirà algun camp indicador en la taula corresponent de la base de dades per a indicar-hi a l'aplicació la configuració de la privacitat d'un usuari. La taula dedicada a les dades de la comunitat no tindrà cap dada xifrada, en canvi la taula dels perfils solament si la dada personal es comparteix amb la comunitat.

## 3.5 Diagrama de components amb especificacions de seguretat i construcció

En la il·lustració següent es mostra un diagrama dels diferents components de l'aplicació web, juntament amb un seguit de notes que identifiquen totes les mesures de seguretat i de construcció preses.



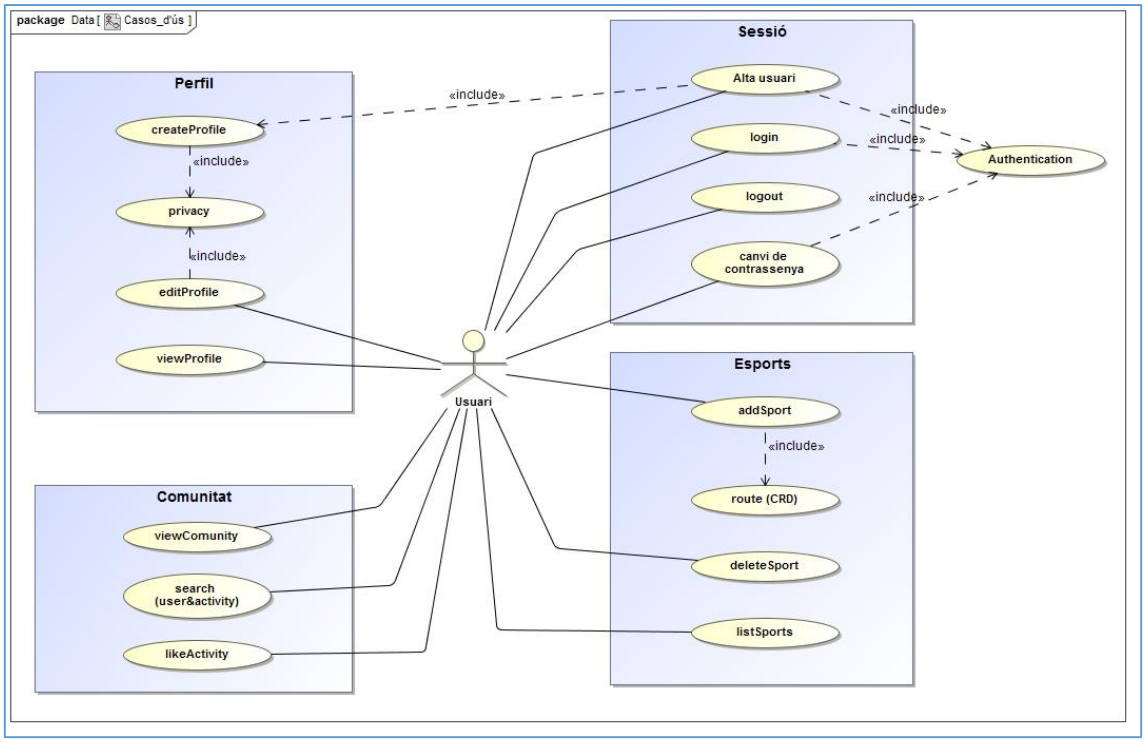

*Il·lustració 28. Diagrama de components amb especificacions de seguretat i construcció.*

## 3.6 Diagrames de casos d'ús i de seqüència de l'aplicació

En aquest apartat es presenten dos dels diagrames UML més comuns per definir les diferents funcionalitats de l'aplicació. Es poden distingir les diferents accions amb que l'usuari interactuarà amb aquesta i els seus diferents passos d'execució.

### 3.6.1 Diagrama de casos d'ús

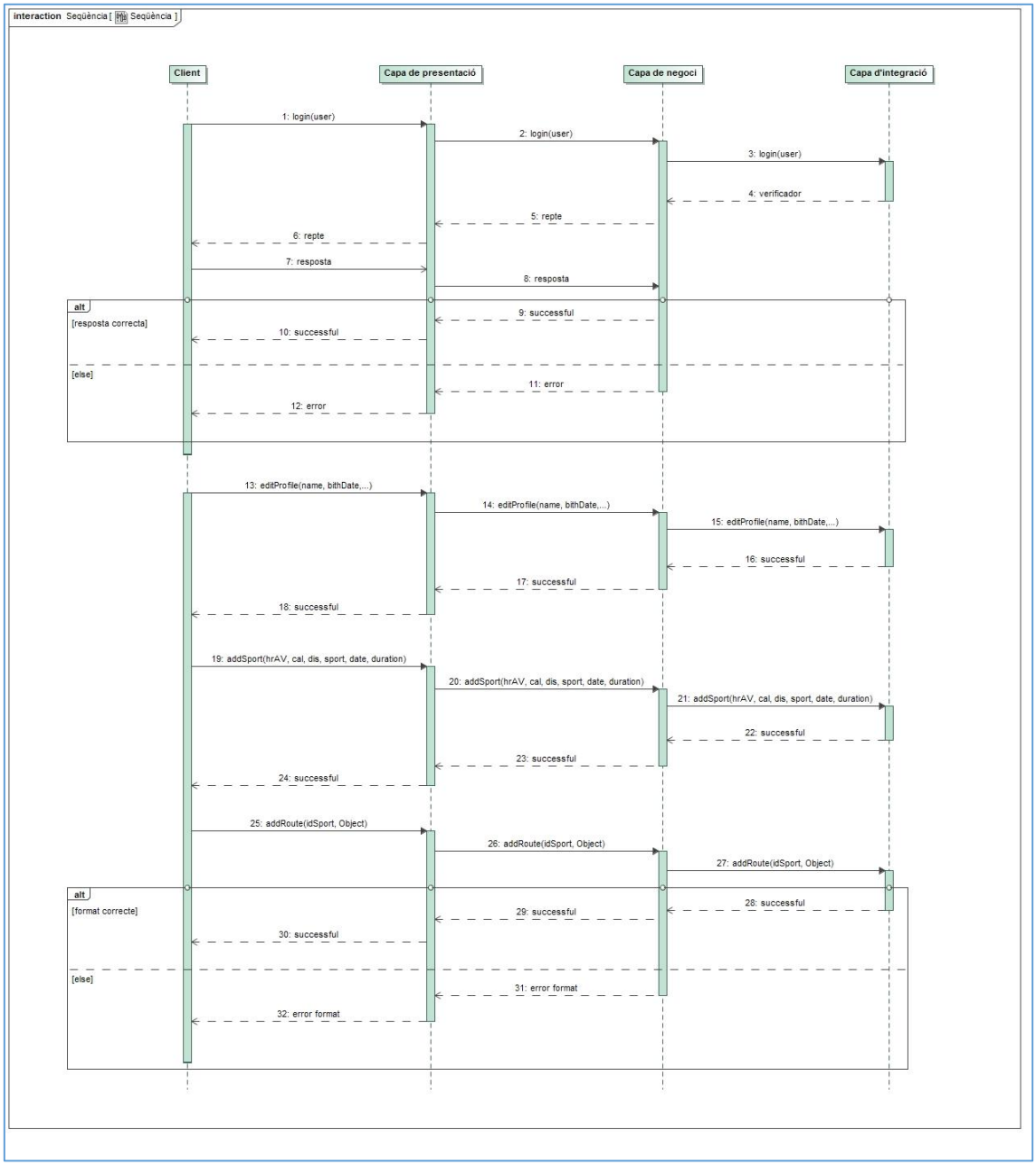
La il·lustració següent mostra el diagrama de casos d'ús, amb les diferents accions agrupades en diferents categories:




*Il·lustració 29. Diagrama de casos d'ús de l'aplicació.*

### 3.6.2 Diagrama de seqüència

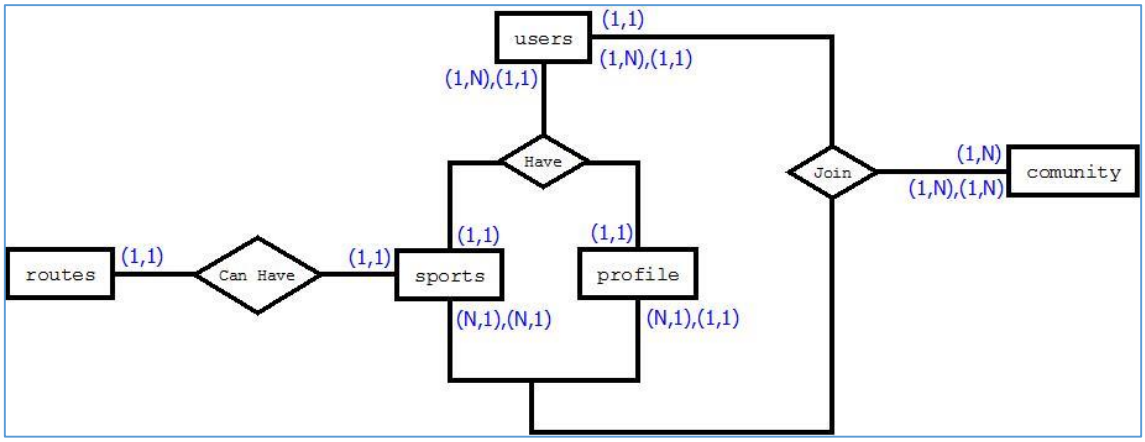
La següent il·lustració mostra el diagrama de seqüència de les accions: *login* d'un usuari, edició del perfil d'usuari, afegiment d'un esport i inserció d'una ruta en un esport. La resta d'accions no mostrades, com l'edició d'un esport o la creació del perfil d'usuari, segueixen una seqüència similar i la seva estructura no té cap diferencia amb les presentades.




*Il·lustració 30. Diagrama de seqüència de l'aplicació.*

## 3.7 Diagrama ER de la base de dades

L'esquema d'entitat-relació ha aplicar en la base de dades és el mostrat en la següent il·lustració:




*Il·lustració 31. Diagrama ER de la base de dades.*

A continuació és realitza una breu descripció de les diferents taules del diagrama anterior:

- La taula users emmagatzema els noms dels usuaris i els verificadors de cada usuari, segons el sistema d'autenticació descrit en apartats anteriors. La clau primària és el nom de l'usuari i aquesta serà clau forana d'altres taules. Un usuari solament tindrà un perfil, en canvi pot tenir un o molts esports.

- La taula profile guarda els diferents camps de les dades personals d'un usuari, i té com a clau primària el nom de l'usuari que a la vegada és clau forana de la taula users.

- La taula sports emmagatzema les diferents variables de les activitats esportives i com a clau primària té un nombre enter generat automàticament per la BD, també conté la clau forana del nom d'usuari de la taula users. Cada usuari pot tindre més d'una activitat guardada, i aquestes es relacionen amb els usuaris a través de la clau forana.

- La taula routes contindrà les dades de la ruta d'una activitat, cada activitat solament pot tindre una ruta com a màxim. Com a clau primària utilitza la clau forana de l'identificador de l'activitat de la taula sports.

- La taula comunity guarda les dades de l'última activitat practicada per cada usuari, si aquest participa en la comunitat. Com a clau primària utilitza el nom de l'usuari que a la vegada és clau forana de la taula users.

# Capítol 4. Implementació de prototip

En aquest capítol es desenvolupa la implementació d'un prototip amb un conjunt de funcionalitats bàsiques. El prototip mostra el funcionament de l'aplicació amb el conjunt de protocols criptogràfics desenvolupats en la part de disseny. En els següents apartats es descriuen les diferents etapes per al desenvolupament de l'aplicació, incloent-hi un apartat específic per a cada un dels tres punts principals dels protocols criptogràfics especificats en l'apartat de disseny.

## 4.1 Preparació de l'entorn de treball

Tal i com s'ha especificat en la part de disseny, la tecnologia a emprar en el desenvolupament és la base de dades *MySQL*, el servidor d'aplicacions *GlassFish Server* i el *framework JSF* per aplicacions Java.

Es realitza la instal·lació de *MySQL Community Server (GPL) 5.7.9*, creant un usuari administrador, a més d'instal·lar l'eina visual *MySQL Workbench*[21] que ens permetrà manipular la base de dades amb facilitat. Amb ajuda d'aquesta última eina es dóna d'alta un nou usuari anomenat "usuari", el qual és el que utilitzarà l'aplicació per connectar-se a la base de dades. A partir del *script* anomenat *sportsTracker.sql* generat per al projecte, es crea la base de dades amb totes les taules necessàries per l'aplicació. Els servidor d'aplicacions instal·lat és el *GlassFish Server Open Source Edition 4.1 (build 13)*.

Per al desenvolupament del prototip s'utilitza *NetBeans* com a entorn de desenvolupament integrat i lliure. Aquest ens permet afegir els *frameworks* que necessitem alhora de crear el projecte de l'aplicació web, a més d'executar l'aplicació directament en el servidor d'aplicacions web. En els següents apartats es descriu les diferents configuracions de la base de dades com del servidor d'aplicacions segons els requeriments del prototip.

## 4.2 Estructura de fitxers del prototip

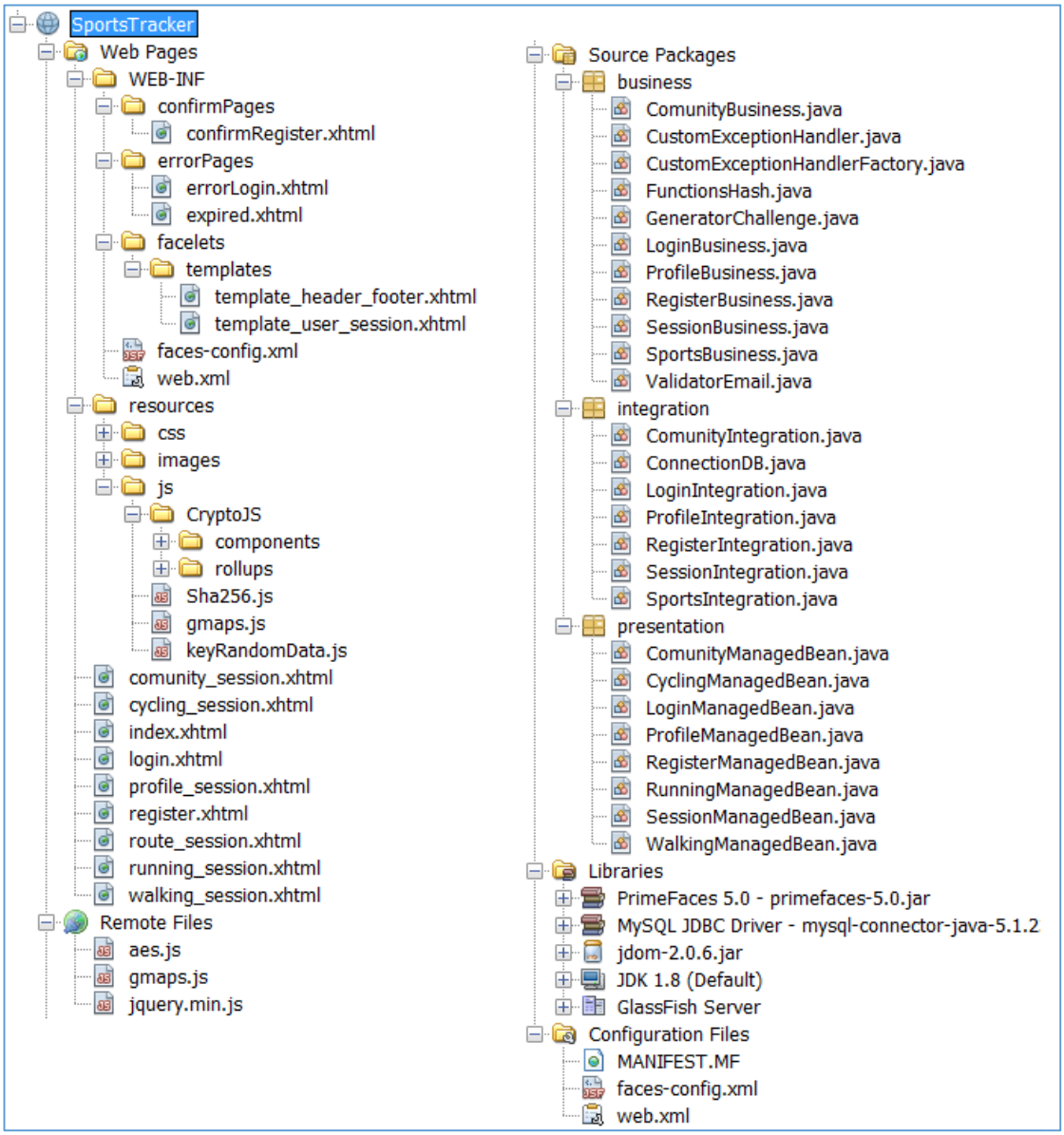
En la imatge de la següent pàgina es mostra l'estructura de fitxers que es composa el prototip desenvolupat. Es poden distingir tres grups principals: pàgines web, paquets de codi d'origen i llibreries.

La carpeta de pàgines web conté totes les pàgines *xhtml* que s'executaran en el client. Dins d'aquesta la carpeta WEB-INF conté les pàgines d'error, confirmació i plantilles utilitzades en l'aplicació. Aquesta ubicació d'aquestes últimes pàgines ens permet que no puguin ser accedides des de el navegador del client. L'arxiu web.xml conté les definicions necessàries per l'aplicació,

[21] *MySQL Workbench* és una eina visual de disseny de bases de dades que integra desenvolupament de programari, administració de bases de dades, disseny de bases de dades, creació i manteniment pel sistema de base de dades *MySQL*.

tot i què en la majoria de casos s'ha utilitzat les anotacions directament en els *beans*. L'altre arxiu faces-config.xml conté l'especificació de la factoria per manejar les excepcions produïdes en el servidor d'aplicacions.

L'altra carpeta que conté la de pàgines web és la de recursos, a on es troben les imatges de l'aplicació, el fitxer d'estils *css*[22] i la biblioteca de *JavaScript*[23] utilitzada per executar codi dins les pàgines web en el client. En la biblioteca de *JavaScript* es pot trobar l'algorisme de xifratge AES, la funció resum SHA i un generador de claus aleatori entre altres.




*Il·lustració 32. Estructura de fitxers del prototip.*

[22] Fulla d'estil en cascada o CSS (sigles en anglès de *cascading style sheets*) és un llenguatge utilitzat per a definir i crear la presentació d'un document estructurat escrit en HTML, XML o XHTML.

[23] *JavaScript* (abreviat comunament "JS") és un llenguatge de programació interpretat, dialecte de l'estàndard *ECMAScript*. Es defineix com orientat a objectes, basat en prototips, imperatiu, tipatge dèbil i dinàmic. S'utilitza principalment en la seva forma del costat del client, implementat com part d'un navegador web permetent millores en la interfície d'usuari i pàgines web dinàmiques.

La carpeta de paquets de codi d'origen conté totes les classes Java de les diferents capes de l'aplicació. En la capa de presentació estan les classes Java *bean* les quals són les que interactuen amb les diferents pàgines xhtml per intercanviar la informació entre client i servidor. En la capa de negoci es pot destacar les classes per manejar les excepcions del servidor, la de funció resum SHA, un validador de format de correu electrònic entre altres. En la capa d'integració existeix una classe per cada objecte de funcionalitat i una classe principal que és la connexió a la base de dades.

Per últim la carpeta de llibreries conté totes les llibreries necessàries per al prototip, entre aquestes destaquem la del servidor *GlassFish*, el fitxer *jar jdom* per poder llegir els arxius *tcx*, el connector *JDBC* per connectar amb la BD *MySQL* i la llibreria *PrimeFaces*[24] que ens facilita la creació de les interfícies web per al client. Addicionalment, a part de totes les carpetes mostrades en la imatge anterior, el projecte inclou una carpeta anomenada "sql" que conté el *script* per la creació de la base de dades i totes les taules necessàries per l'aplicació, també s'inclou la carpeta "test" que conté varis fitxers *tcx* de proves i els certificats i contenidors de claus del servidor Galssfish creats per al prototip.

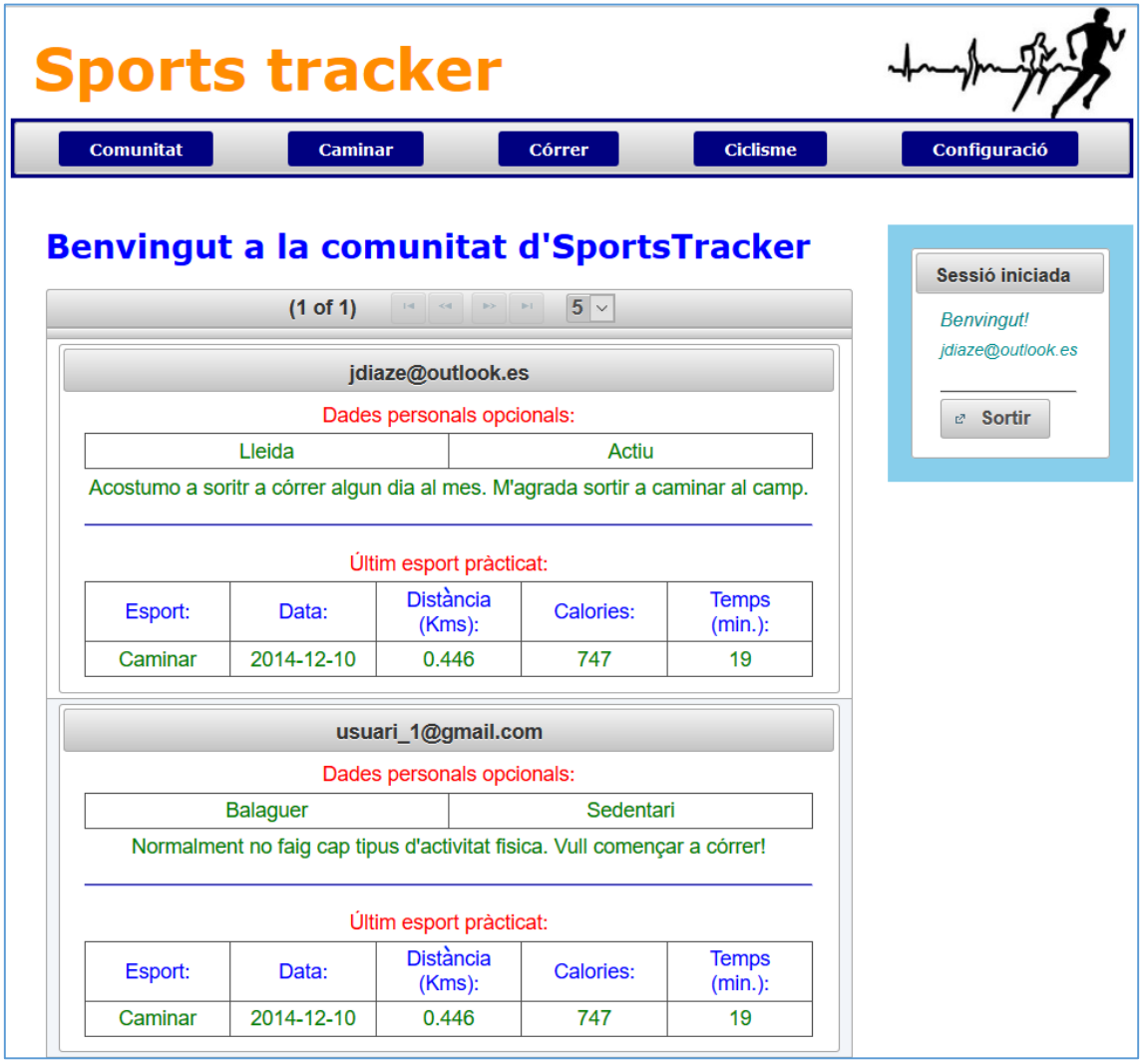
## 4.3 Estructura i funcionalitat de les pàgines web

En aquest apartat es vol mostrar les diferents pàgines web que composen l'aplicació i les seves funcionalitats. La primera pàgina que mostra l'aplicació és l'índex, el qual s'accedeix a través de la direcció *localhost/SportsTracker*. Aquesta ruta no es pot modificar sense tindre en compte que s'haurien d'actualitzar els enllaços de les pàgines d'error i de confirmació.


*Il·lustració 33. Pàgina web principal del prototip.*

[24] *PrimeFaces* és un component per a *Java Server Faces (JSF)* de codi obert que compte amb un conjunt de components enriquits que faciliten la creació de les aplicacions web.

Un cop l'usuari s'autentica des de la pàgina web d'inici, aquest inicia sessió en l'aplicació i es mostra la pàgina web de la comunitat d'usuaris amb una barra de menú superior amb les diferents opcions de l'aplicació. La configuració de l'aplicació defineix el temps de sessió en 30 minuts, un cop transcorregut aquest temps en cas d'inactivitat la sessió de l'usuari expira i es tanca. La il·lustració següent mostra la pantalla inicial que l'usuari veurà un cop inicia sessió.

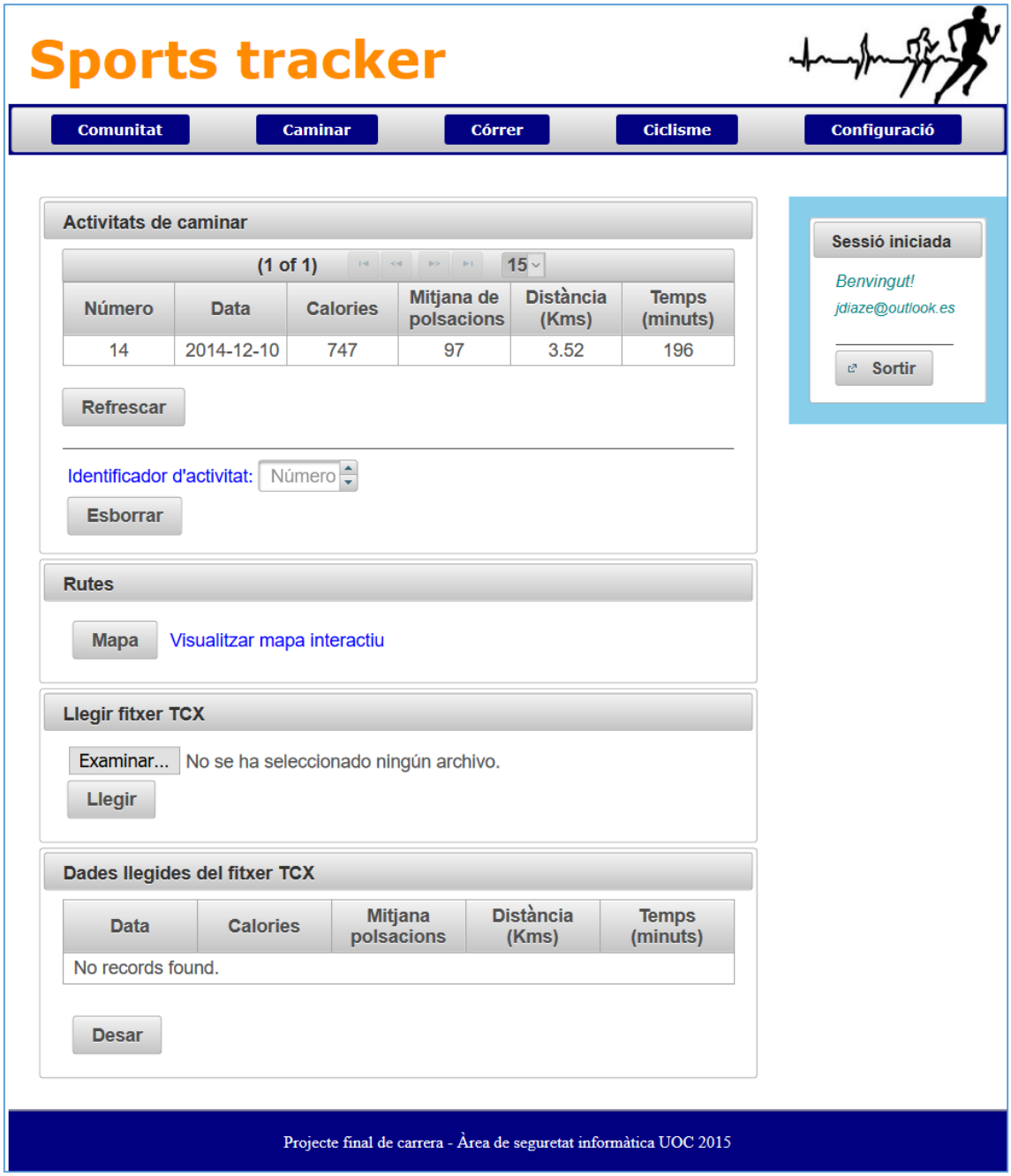



*Il·lustració 34. Pàgina web de la comunitat d'usuaris del prototip.*

La barra de menú superior, a part de la comunitat d'usuaris, ens ofereix tres activitats esportives diferents i la configuració del perfil de l'usuari. Els tres esports tenen el mateix esquema de pàgina web i de funcionalitat. La il·lustració de la pàgina següent mostra la pàgina web de l'activitat esportiva caminar.

En la pàgina d'una activitat esportiva l'usuari veurà el llistat de totes les activitats concretes a la pàgina on està i que ha desat en l'aplicació. Té l'opció d'esborrar l'activitat que vulgui a través de l'identificador de cada activitat. Addicionalment s'ha afegit l'opció de ruta, que redirigeix a l'usuari a una pàgina web nova amb un mapa interactiu de Google que pot utilitzar per planificar les seves activitats. L'opció de guardar les rutes en l'aplicació i mostrar-les en aquest mapa no s'ha implementat en aquest prototip.

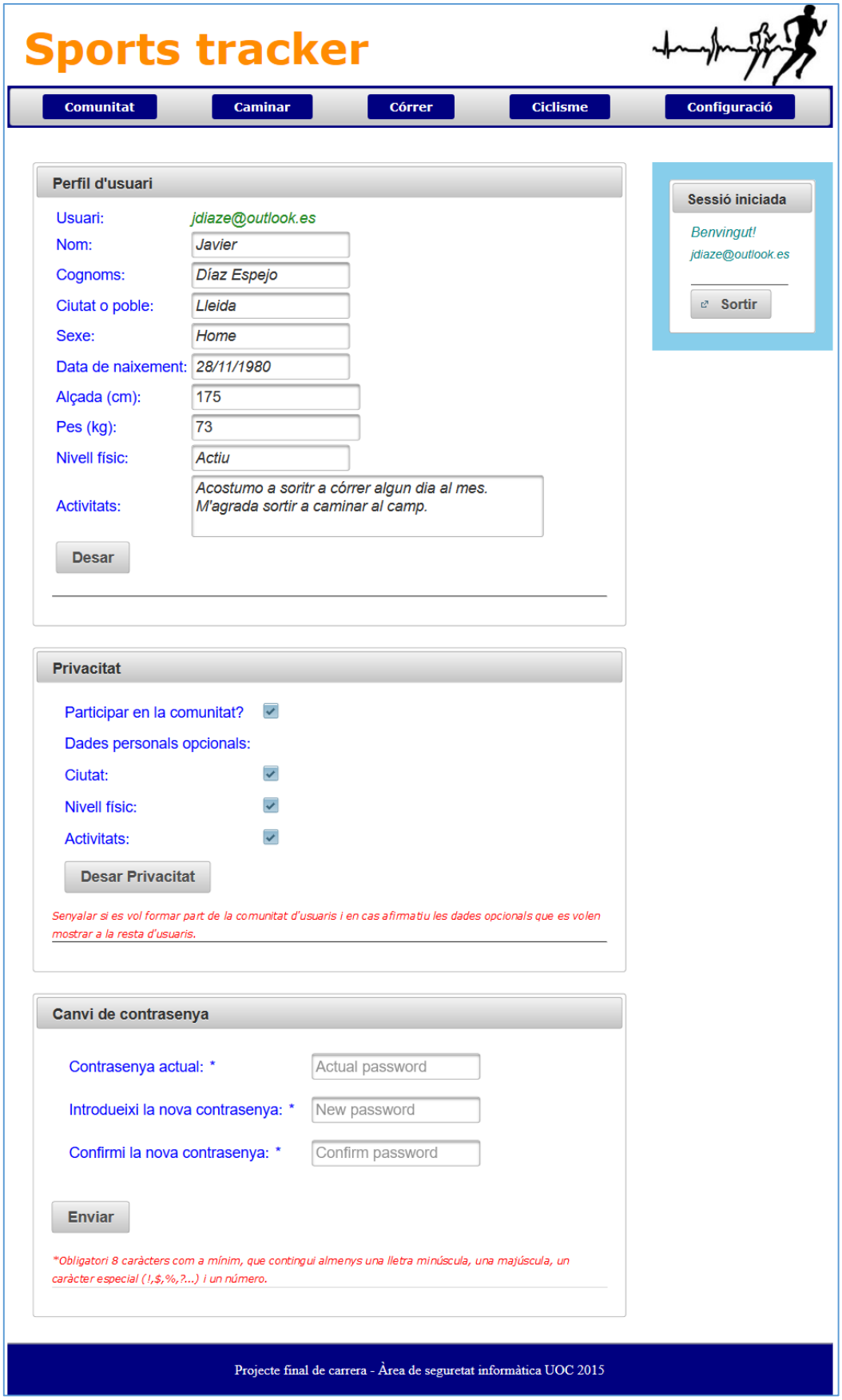
Per afegir una activitat nova en la pàgina d'una activitat esportiva, primer s'ha de carregar un fitxer *tcx* amb les dades de l'activitat. Un cop llegit el fitxer, l'usuari pot visualitzar les dades llegides del fitxer i si vol pot desar-les en l'aplicació xifrades. El fitxer *tcx* contindrà solament una activitat esportiva, en cas de contenir més d'una solament es llegirà la primera.




*Il·lustració 35. Pàgina web de l'activitat caminar del prototip.*

En l'apartat de configuració l'usuari pot afegir una sèrie de dades personals, canviar la seva privacitat per participar en la comunitat d'usuaris i mostrar certes dades personals en aquesta. També en aquest apartat té la possibilitat de canviar la contrasenya d'inici de sessió de l'aplicació.

La següent il·lustració mostra la pàgina web de configuració de l'usuari:




*Il·lustració 36. Pàgina web de configuració d'usuari del prototip.*

L'usuari a l'hora de desar les seves dades personals, abans d'enviar-les al servidor es xifren i el servidor emmagatzema a la base de dades totes les dades xifrades de l'usuari. En cas que l'usuari vulgui participar en la comunitat i marqui alguna de les caselles per mostrar alguna de les dades personals opcionals en la comunitat, la dada personal opcional a mostrar en la comunitat es desxifra i s'envia en clar al servidor per a què la guardi en el lloc de la xifrada. Si un usuari participa en la comunitat, en el moment que desi qualsevol activitat esportiva es desarà a la vegada en la taula comunitat de la BD sense xifrar.

Participar en la comunitat significa mostrar el nom d'usuari, l'última activitat esportiva pujada a l'aplicació (esport, data, distància, calories i temps) i segons l'usuari les dades personals opcionals que ha configurat en el seu perfil. Si un usuari deixa de participar en la comunitat, s'esborra totes les dades de l'últim esport practicat de la taula comunitat de la BD i si es mostrava alguna dada personal opcional es torna a xifrar per desar-la en la BD. En cas contrari, quan un usuari participa en la comunitat, l'últim esport practicat serà el que pugi a l'aplicació a partir del moment que participa en la comunitat.

L'usuari pot també canviar la seva contrasenya d'inici de sessió sense preocupar-se per les dades que s'han xifrat, ja que la clau per xifrar les dades és un altra que es guarda xifrada en el mateix ordinador de l'usuari.

## 4.4 Aplicació dels protocols criptogràfics

En aquest apartat es detalla la implementació en el prototip dels diferents protocols criptogràfics analitzats en l'apartat de disseny. Es composa de tres apartats principals: connexió segura, autenticació d'usuari i confidencialitat de les dades.

### 4.4.1 Connexió segura

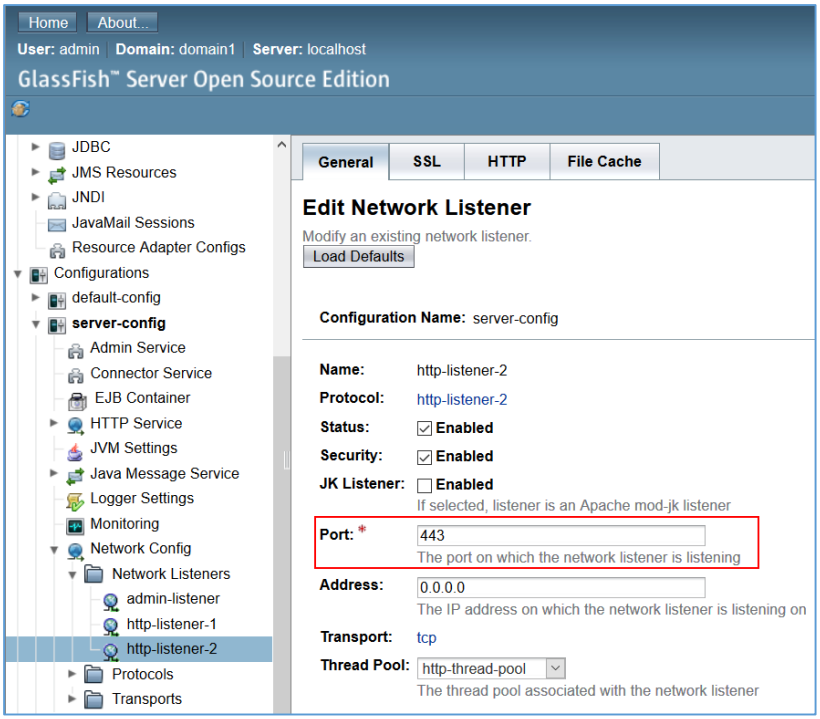
Per configurar la connexió segura entre el navegador del client i l'aplicació web es fa ús d'una connexió *https* amb ajuda del protocol criptogràfic SSL/TLS. El servidor d'aplicacions *Glassfish* es configura per a què escolti solament pel port 443, a més s'instal·la un certificat vàlid per la direcció *localhost* que a la vegada està signat per un certificat d'una autoritat de confiança que es pot instal·lar en el navegador web del client.

El primer pas és la creació dels certificats vàlids, són dos, un el certificat del propi servidor i un segon que fa d'autoritat de confiança per a signar el primer i que instal·lat en el navegador web fa que la connexió amb la pàgina web no doni cap avís de certificat no vàlid. En la creació dels dos certificats s'han introduït una sèrie de dades personals, seguit del camp CN (*common name*) que és el més important ja que fa referència a la direcció de la pàgina web. En el cas del prototip aquest camp figura el valor *localhost*, ja que les proves s'han realitzat en una màquina física a nivell local.

El servidor *Glassfish* emmagatzema les claus i els certificats en uns fitxers contenidors anomenats *keystore.jks* i *cacerts.jks*. Pel maneig d'aquests contenidors s'empra l'eina de Java Keytool[25], la qual permet la creació de parells de claus, generació de certificats de petició (*csr*), signar certificats de petició,... En el cas del prototip s'utilitza aquesta eina per generar el parell de claus al contenidor corresponent, generar el certificat de petició (*csr*) i signar aquest certificat amb el certificat de l'autoritat certificadora creat per al propòsit. Per la creació del certificat de l'autoritat certificadora s'utilitza l'eina *OpenSSl*. Per configurar el servidor que utilitzi el certificat i les claus corresponents, alhora de crear el parell de claus com del certificat se'ls assigna un àlies. Aquest àlies per facilitat s'ha deixat amb el mateix nom que el certificat auto- signat que porta per defecte el servidor, comportant la no necessitat de canviar certs punt de configuració del servidor.

En l'apartat "*Creació i gestió de certificats*" de l'annex es defineixen el punts a seguir per la generació dels dos certificats i parell de claus per al servidor. Com a resultat de l'execució dels punts definits en l'annex amb les diferents eines, el servidor ja disposa d'un parell de claus amb el seu certificat que està signat per un altre certificat que fa d'autoritat de confiança. El certificat de confiança a la vegada s'instal·la en el navegador web a utilitzar amb l'aplicació.

Seguidament es configura el servidor per a què escolti pel port 443. El servidor *Glassfish* disposa per defecte de tres escoltador de xarxa: un a través del port 4848 per accedir a la consola d'administració del propi servidor, un altre a través del port 8080 i un últim a través del port 8181 amb la seguretat SSL/TLS habilitada (*http-listener2*). Es modifica el port 8181 de l'escoltador *http-listener2* pel port 443. La il·lustració següent mostra la pantalla de configuració amb el port modificat (a través de la consola d'administració que s'accedeix des de localhost:4848).

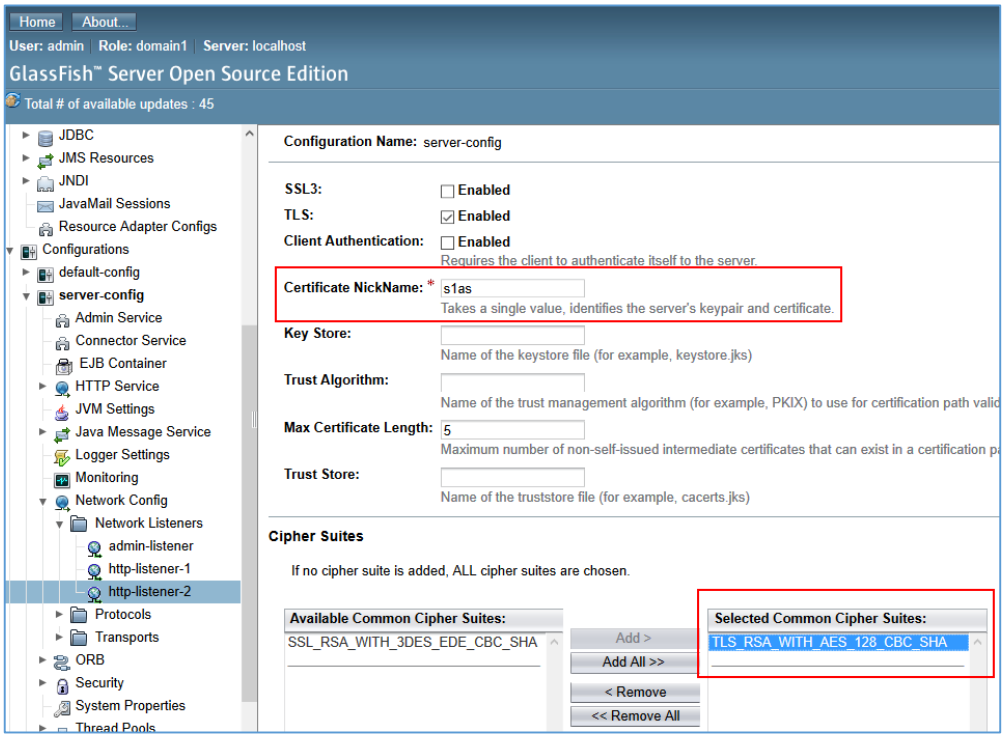



*Il·lustració 37. Configuració del port d'escolta de Glassfish.*

[25] Eina que incorpora *Java KeyStore* (JKS) que és un repositori de certificats de seguretat de Java.

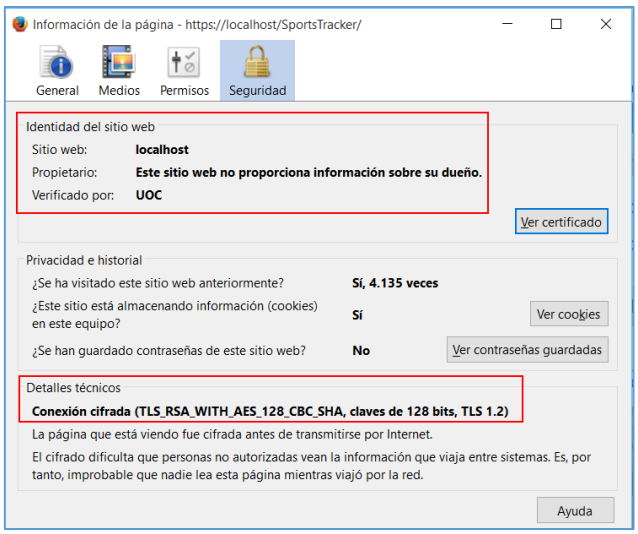
I per últim, es comprova el nom del àlies del certificat que utilitza el servidor per aquesta connexió *http* i es selecciona la suite de xifratge per defecte TLS_RSA_WITH_AES_128_CBC_SHA. Recordem que al generar el certificats del servidor se li ha donat el mateix nom d'àlies que el certificat per defecte que porta el servidor (s1as), evitant la necessitat d'actualitzar aquest paràmetre en els diversos camps.

La il·lustració següent mostra la pantalla de configuració del protocol SSL de l'escoltador *http-listener2* del servidor Glassfish:



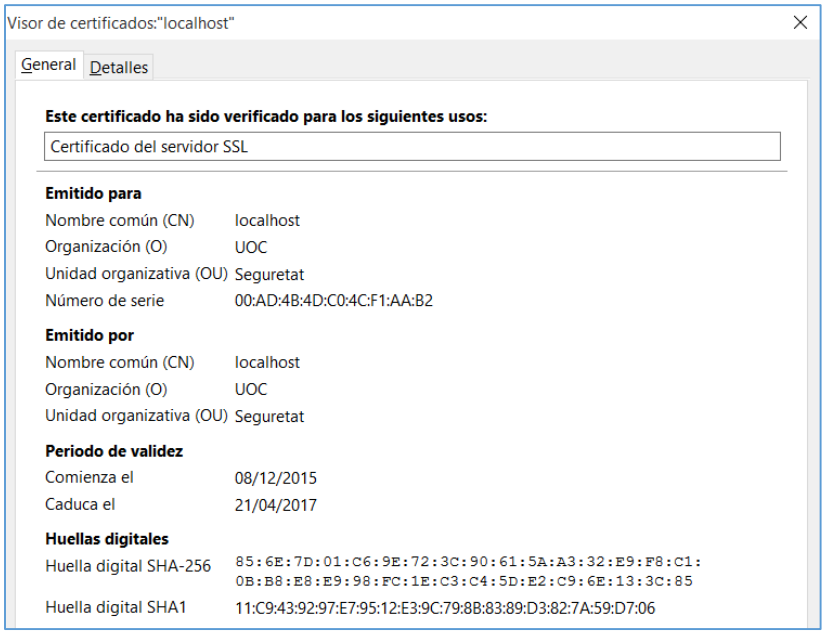

*Il·lustració 38. Configuració de SSL de Glassfish.*

Un cop configurat el servidor d'aplicacions es comprova la connexió que funciona correctament i les dades del certificat generat per al prototip. La següent il·lustració mostra la informació de la connexió del lloc web de l'aplicació des del navegador web (https://localhost/SportsTracker).




*Il·lustració 39. Informació de la connexió web del prototip.*

En el navegador web s'ha instal·lat el certificat de l'autoritat certificadora (rootCA.crt) generat per a signar el certificat de l'aplicació, en l'apartat de certificats de confiança del navegador. D'aquesta forma el navegador no dóna cap tipus d'avís de certificat no confiable. La il·lustració següent mostra la informació del certificat del servidor d'aplicacions:




*Il·lustració 40. Informació del certificat de la connexió web del prototip.*

### 4.4.2 Autenticació dels usuaris

L'autenticació d'usuaris en el prototip s'ha implementat a base d'un mètode d'autenticació feble. L'usuari introduirà la seva clau en la pàgina principal de l'aplicació juntament amb el seu nom d'usuari. Com a nom d'usuari s'ha implementat la utilització del correu electrònic del propi usuari, d'aquesta forma a l'usuari li resulta molt més fàcil recordar-se d'aquest nom i a la vegada serveix de mitjà de comunicació entre els usuaris de la comunitat de l'aplicació del prototip.

Un cop l'usuari introdueix les seves credencials a la pàgina d'inici s'envia el nom d'usuari i la seva clau al servidor aprofitant la connexió segura entre client i servidor. El servidor el primer pas que realitza és la comprovació de l'existència del nom d'usuari en la base de dades. Si aquest existeix, llegeix el resum de la clau de l'usuari que té emmagatzemada en la BD i seguidament fa el resum de la clau que li ha enviat l'usuari. Un cop té els resums de ambdues claus, la de l'usuari i la emmagatzemada pel servidor, els compara i si són iguals dóna per vàlida l'autenticació de l'usuari. En cas de no existir l'usuari o que els resums de les claus no coincideixin es retorna un error d'autenticació a l'usuari, indicant-li que el nom d'usuari o contrasenya no és vàlid. El missatge d'error no especifica quin camp és el incorrecte, d'aquesta forma no és dóna cap pista a un possible atacant que vulgui saber si un nom d'usuari està donat d'alta en l'aplicació.

Per seguretat, en el moment que un usuari és dóna d'alta en l'aplicació, el servidor emmagatzema un resum de la clau en la BD i en cap moment es guarda la contrasenya en clar en el servidor. D'aquesta forma en cas que un atacant exterior o inclús un d'interior tingués accés a la BD no podria veure les contrasenyes dels usuaris en clar.

Un altre punt que s'ha tingut en compte és l'afegiment de bits de sal en les contrasenyes dels usuaris abans de fer un resum d'aquestes. Això permet que en un cas remot que dos usuaris elegeixin una mateixa contrasenya, a l'hora de realitzar el resum d'aquestes i emmagatzemar-les en la BD tinguin resums diferents. Els bits de sal es creen a l'hora de registrar-se un usuari, i s'emmagatzemen juntament amb un resum de la clau més els bits de sal. Quan el servidor ha d'autenticar un usuari, llegeix els bits de sal d'aquest usuari i els suma a la clau facilitada pel usuari, seguidament realitza el resum d'aquesta suma i la compara amb la que té emmagatzemada en la BD.

En el prototip a l'usuari alhora de registrar-se en l'aplicació se'l exigeix que la contrasenya que elegeixi tingui les següent condicions:

*"Obligatori 8 caràcters com a mínim, que contingui almenys una lletra minúscula, una majúscula, un caràcter especial (!,$,%,?...) i un número"*

L'usuari té la possibilitat de canviar la contrasenya en qualsevol moment, amb les mateixes condicions exigides en el registre d'un nou usuari. En el prototip no s'ha implementat cap sistema de caducitat de contrasenyes per obligar a l'usuari a canviar-la.

En l'aplicació en el moment de registre d'un usuari, s'executa en el mateix client una funció *JavaScript* que genera una cadena de 32 caràcters aleatoris que formaran els bits de sal assignats a l'usuari, i es realitza el resum de la contrasenya d'usuari més els bits de sal generats. Aquesta funció fa ús de llibreries de *JavaScript* dels recursos web que disposa l'aplicació per la generació dels bits de sal i la realització del resum de contrasenya més bits de sal. La il·lustració següent mostra el codi *JavaScript* que s'executa en el client quan un usuari clica en el botó de registre:

```
var contrasenya;
var usuari;
function gestioContrasenya() {
    usuari = jQuery("[id$='usuari']").val();
    contrasenya = jQuery("[id$='contrasenya']").val();
    bitsSal = keyRandomData.generate(32,false);
    contrasenya = contrasenya + bitsSal;
    var contrasenyaHash = Sha256.hash(contrasenya);
    jQuery("[id$='passHash']").val(contrasenyaHash);
    jQuery("[id$='bitsSal']").val(bitsSal);
}
```

*Il·lustració 41. Codi de la funció JavaScript executada en el client en cas de registre d'un usuari.*

En el cas d'un usuari ja registrat i que es vol autenticar en el sistema, el client envia la contrasenya introduïda pel usuari al servidor. El servidor busca en la BD el resum de la clau del usuari i els seus bits de sal, realitza el resum de la clau rebuda pel client més els bits de sal llegits de la BD i fa la comparació del resum calculat amb el resum llegit de la BD. En la pàgina següent es mostra una il·lustració amb el codi del mètode *login* de la classe Java *LoginBusiness* de la capa de negoci, executat en el servidor i encarregat de realitzar l'autenticació d'un usuari que vol iniciar sessió en l'aplicació web.

```java
/**
 * Mètode de login d'un usuari.
 * @param usuari
 * @param clau
 * @return boolean - si el resum de la clau d'usuari es igual al resum obtingut de la BD s'inicia la sessió d'usuari (true).
 */
public boolean login(String usuari, String clau)  {

    String[] clauHashBitsSal = loginIntegration.searchUser(usuari);
    functionsHash = new FunctionsHash();

    String bitsSal = clauHashBitsSal[1];
    String clauHash = functionsHash.getStringMessageDigest(clau+bitsSal, "SHA-256");

    String clauHashServer = clauHashBitsSal[0];
    //Es comprova si són iguals els dos resums hash.
    if(clauHash.equals(clauHashServer)) {
        System.out.print("Autenticació d'usuari correcta: "+usuari);
        return true;    //s'inicia sessió
    } else {
        return false;   //error d'autenticació
    }
}
```

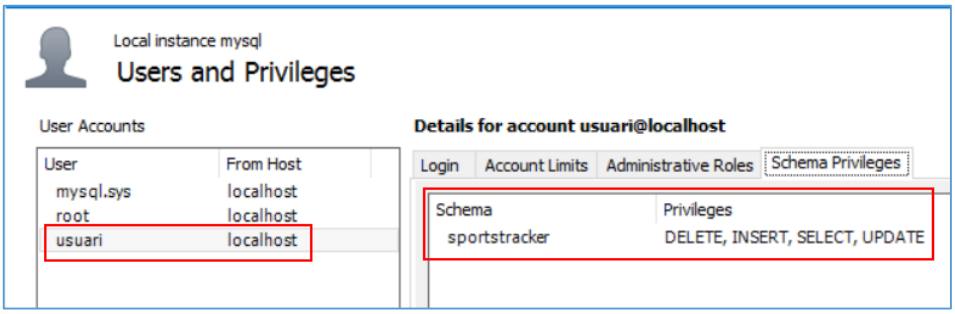
*Il·lustració 42. Codi del mètode login de la classe LoginBusiness executat en el servidor per autenticar un usuari.*

### 4.4.3 Confidencialitat de les dades

En aquest apartat es realitza la descripció de la implementació en el prototip dels diferents punts tractats en l'apartat de disseny sobre la confidencialitat de les dades.

#### 4.4.3.1 Gestió de permisos

Un cop s'ha instal·lat la base de dades MySQL, s'executa el script *sportsTracker.sql* per la creació de la BD anomenada *sportstracker* amb totes les taules corresponents que la composen per al funcionament de l'aplicació web. Tot seguit es realitza la creació d'un usuari a la BD anomenat "*usuari*" i se'l dóna permisos per seleccionar, inserir, actualitzar i esborrar files sobre totes les taules de l'esquema *sportstracker* de la BD. L'usuari de la BD creat per l'aplicació solament es podrà accedir a ell a nivell local. Els usuaris de l'aplicació no tenen accés directe a la BD, és solament l'aplicació la que interacciona amb la BD amb una única connexió i a través de l'usuari creat específicament per aquesta. La següent il·lustració mostra la pantalla de configuració d'usuaris i privilegis de l'administrador *Workbench* de MySQL, a on figura el compte d'usuari creat i els privilegis que disposa sobre l'esquema *sportstracker*.




*Il·lustració 43. Privilegis de l'usuari de l'aplicació de la base de dades.*

#### 4.4.3.2 Xifratge de dades

Les dades que un usuari introdueix o llegeix del prototip són xifrades en la part del client i el servidor les emmagatzema en la BD. Per realitzar aquesta tasca el prototip utilitza una llibreria *JavaScript* anomenada *CryptoJS*. Aquesta llibreria inclou un conjunt d'eines criptogràfiques per xifrar, realitzar resums i canvi de base de codificació.

De la llibreria *CryptoJS* s'utilitza la funció AES. La configuració de la longitud de la clau a 128 bits de l'algorisme AES segons el que s'ha especificat en la part de disseny, ho realitza la mateixa funció de la llibreria automàticament segons la clau que s'especifiqui alhora de xifrar i desxifrar. Per defecte l'algorisme de la funció AES de la llibreria utilitza el mode CBC, per tant no s'ha tingut de canviar cap configuració. En les il·lustracions següents es mostren exemples de codi *JavaScript* utilitzat en la part del client per a xifrar i desxifrar les dades de l'usuari.

```
var getNom = jQuery("[id$='nom']").val();
var nom = CryptoJS.AES.encrypt(getNom,sessionStorage.getItem(usuari));

var getCognoms = jQuery("[id$='cognoms']").val();
var cognoms = CryptoJS.AES.encrypt(getCognoms,sessionStorage.getItem(usuari));
```

*Il·lustració 44. Codi JavaScript d'exemple de xifratge de dades.*

```
var getNom = jQuery("[id$='nomX']").val();
var nom = CryptoJS.AES.decrypt(getNom,sessionStorage.getItem(usuari));

var getCognoms = jQuery("[id$='cognomsX']").val();
var cognoms = CryptoJS.AES.decrypt(getCognoms,sessionStorage.getItem(usuari));
```

*Il·lustració 45. Codi JavaScript d'exemple de desxifrar dades.*

En aquestes il·lustracions es mostra el codi de xifrat i de desxifrat de dos camps de dades personals (nom i cognoms) del perfil de l'usuari. La clau utilitzada per la xifra AES no és la contrasenya de l'usuari, és una clau generada aleatòriament durant el registre d'un usuari amb una llargària adequada per a què la funció AES de la llibreria criptogràfica treballi amb una longitud de clau de 128 bits. Aquesta clau per xifrar dades és manté guardada en l'ordinador de l'usuari i de la seva gestió es parla en l'apartat següent.

S'ha de recordar que una xifra de clau compartida o simètrica de xifres de bloc, actuen sense memòria i el text xifrat només pot dependre del text en clar i de la clau. Això últim ocasiona que dos textos en clar iguals es xifren de la mateixa forma quan s'utilitza la mateixa clau. La funció criptogràfica AES de la llibreria utilitza vectors d'inicialització, que en cas de no indicar-li un de propi ella mateixa genera un de nou cada cop que xifra. D'aquesta forma al xifrar una mateixa dada diverses vegades el resultat del xifrat és diferent, a més que dos usuaris tinguin les mateixes claus de xifrar les dades són d'una probabilitat molt baixa, la clau de xifrar dades són onze caràcters que es generen de forma aleatòria.

#### 4.4.3.3 Gestió de claus

Segons l'apartat anterior i l'apartat d'autenticació d'usuaris, es pot concloure que per la implementació del prototip s'han utilitzat dos tipus de claus. La primera és la contrasenya d'usuari que ens serveix per l'autenticació d'aquest davant l'aplicació i una segona clau utilitzada per a xifrar les dades personals de l'usuari.

En l'apartat d'autenticació s'ha especificat clarament com l'aplicació guarda un resum de la contrasenya d'usuari més uns bits de sal a la BD, i aquest resum és el utilitzat per verificar la identitat d'un usuari. Les contrasenyes d'usuari no es guarden en la BD en format de text en clar, en el seu lloc hi ha els resums d'aquestes que a demés incorporen l'afegit dels bits de sal.

La clau de xifrar les dades personals d'un usuari, es genera de forma transparent en el moment que un usuari es dóna d'alta en l'aplicació. A través d'una funció de *JavaScript* en el client es genera una cadena d'onze caràcters aleatoris. Un cop generada la clau per xifrar dades, s'emmagatzema en l'ordinador de l'usuari a través de l'espai d'emmagatzemament *localStorage* que ens ofereix HTML5 utilitzant com a identificador el nom de l'usuari. La clau de dades es xifrarà amb la contrasenya de l'usuari abans de guardar-la. Aquest espai d'emmagatzemament que ofereix HTML5 té les següents característiques:

- Té un espai assignat entre 5 i 10 MB, depenent del navegador web.
- La informació emmagatzemada amb *localStorage* no és enviada al servidor en cada petició.
- No existeix una caducitat per a *localStorage*, la informació queda emmagatzemada fins que s'elimini expressament; el tancament del navegador web no afecta en res.

La gestió de la clau de xifrar dades fa que l'usuari per l'ús de l'aplicació hagi d'utilitzar el mateix ordinador amb el mateix navegador web que va utilitzar per a donar-se d'alta en l'aplicació. La clau de dades xifrada que emmagatzema el navegador web es pot arribar a recuperar amb eines externes capaces de llegir el contingut de *localStorage* d'alguns navegadors web. Això permetria canviar de navegador web com d'ordinador personal, tot i què no és un procés que qualsevol usuari podria arribar a realitzar.

Quan un usuari s'autentica en l'aplicació, la mateixa aplicació busca la clau de dades xifrada a través del mateix nom d'usuari en *localStorage*. Un cop llegida la clau la desxifra amb la contrasenya que ha facilitat l'usuari per autenticar-se, i la clau de dades en clar s'emmagatzema en l'espai *sessionStorage* que ens ofereix HTML5 durant la sessió d'usuari. L'aplicació durant la sessió d'usuari fa ús de la clau de dades en clar que té guardada en *sessionStorage* per a xifrar i desxifrar les dades, un cop l'usuari tanca sessió o aquesta expira la clau de dades en clar s'esborra.

La il·lustració de la pàgina següent mostra la funció *JavaScript* que s'executa en el client quan un usuari es dóna d'alta en l'aplicació. Es pot observar en el codi de la il·lustració com es genera una clau aleatòria d'onze caràcters que és transforma en base 64 (16 bytes / 128 bits), es xifra amb la contrasenya de l'usuari i finalment s'emmagatzema en l'espai *localStorage*.

```
function gestioClau() {
    var clauRandomData =  keyRandomData.generate(11,false);
    var key = CryptoJS.enc.Base64.parse(clauRandomData);
    var clauDataXifrada = CryptoJS.AES.encrypt(key,contrasenya);
    localStorage.setItem(usuari,clauDataXifrada);
}
```

*Il·lustració 46. Codi de la funció JavaScript executada per generar la clau de xifrar dades.*

En la il·lustració següent mostra la funció *JavaScript* que s'executa en el client quan un usuari s'autentica en l'aplicació. El primer que es realitza és llegir la clau de dades xifrada de l'usuari de l'espai *localStorage* a través d'utilitzar el nom d'usuari com a identificador, tot seguit es desxifra amb ajuda de la contrasenya de l'usuari i la clau en clar es guarda en l'espai *sessionStorage* amb el nom d'usuari com a identificador.

```
function getKeyData() {
    var usuari = jQuery("[id$='usuari']").val();
    var pass = jQuery("[id$='clau']").val();
    var clau = CryptoJS.AES.decrypt(localStorage.getItem(usuari),pass);
    sessionStorage.setItem(usuari,clau);
}
```

*Il·lustració 47. Codi de la funció JavaScript executada per recuperar la clau de xifrar dades.*

#### 4.4.3.4 Privacitat de les dades

En el prototip desenvolupat s'utilitza un camp de la taula de perfils d'usuari de la BD per indicar-hi si l'usuari participa en la comunitat i quins camps de les dades personals que són opcionals vol mostrar en la comunitat.

Quan un usuari participa en la comunitat, automàticament en el moment que puja una activitat a l'aplicació es guarda en la taula comunitat de la BD sense xifrar com a última activitat practicada. Si ja existia una activitat en la taula comunitat es substitueix per la nova pujada a l'aplicació. Les activitats d'un usuari sempre es van guardant xifrades en la taula d'esports de la BD, la taula comunitat solament inclourà l'última activitat pujada a l'aplicació pel usuari en cas de participar a la comunitat. Com a última activitat es té en compte l'última pujada a l'aplicació a partir del moment que l'usuari participa en la comunitat, en cas de no haver-hi pujat cap activitat a l'aplicació des de que participa no es mostrarà cap a la comunitat.

Per les dades personals que són opcionals a mostrar en la comunitat, aquestes no es guarden en la taula de la comunitat de la BD, si no que es deixen en clar en la taula del perfil d'usuari. Per tant, totes les dades personals del perfil d'un usuari estan xifrades, excepte les tres dades opcionals que l'usuari pot mostrar a la comunitat que segons la configuració poden estar guardades en clar.

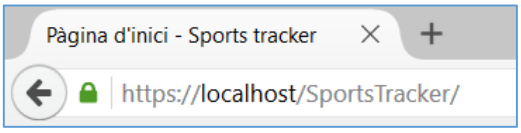
El camp utilitzat en la taula perfil de la BD per gestionar la comunitat no està xifrat, tot i què podria ser modificat per un possible atacant alhora de carregar les dades a la comunitat es trobaria que les dades personals estan xifrades i que a la taula de la comunitat no hi ha cap activitat emmagatzemada de l'usuari o usuaris afectats.

# Capítol 5. Proves de prototip

Un cop es té el prototip en funcionament i amb l'entorn de proves configurat adequadament es realitzen una sèrie de comprovacions:

- *Comprovació del tipus de connexió entre client i servidor:*

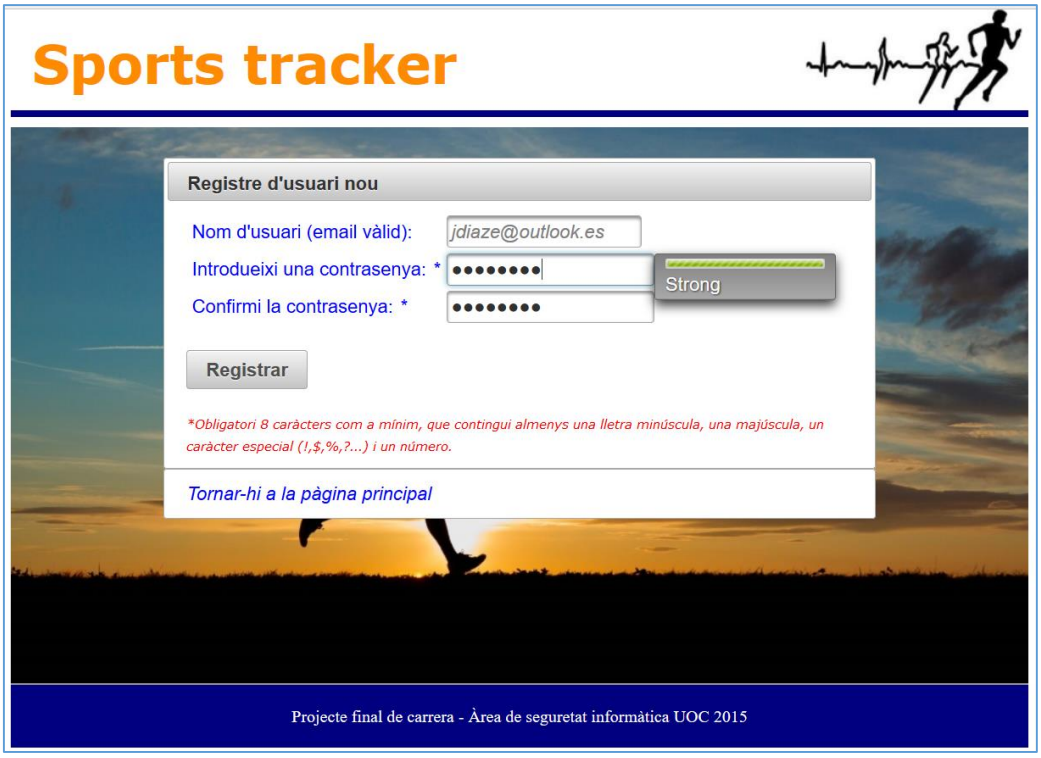
Tal i com s'ha descrit en l'apartat de connexió segura de l'apartat anterior el prototip desenvolupat utilitza solament el port 443 (*https*) per la connexió entre client i servidor. L'ús de la connexió *http* (port 80) no està habilitada. La següent il·lustració mostra la barra de direcció del navegador alhora de connectar-se amb l'aplicació, la qual hi figura el símbol de pàgina segura i la URL comença amb *https*. La carpeta "proves" del projecte inclou els certificats i els contenidors de claus i certificats *jks* del servidor *Glassfish* utilitzats en l'entorn de proves.




*Il·lustració 48. URL amb connexió segura del prototip.*

- *Creació d'un nou usuari de l'aplicació:*

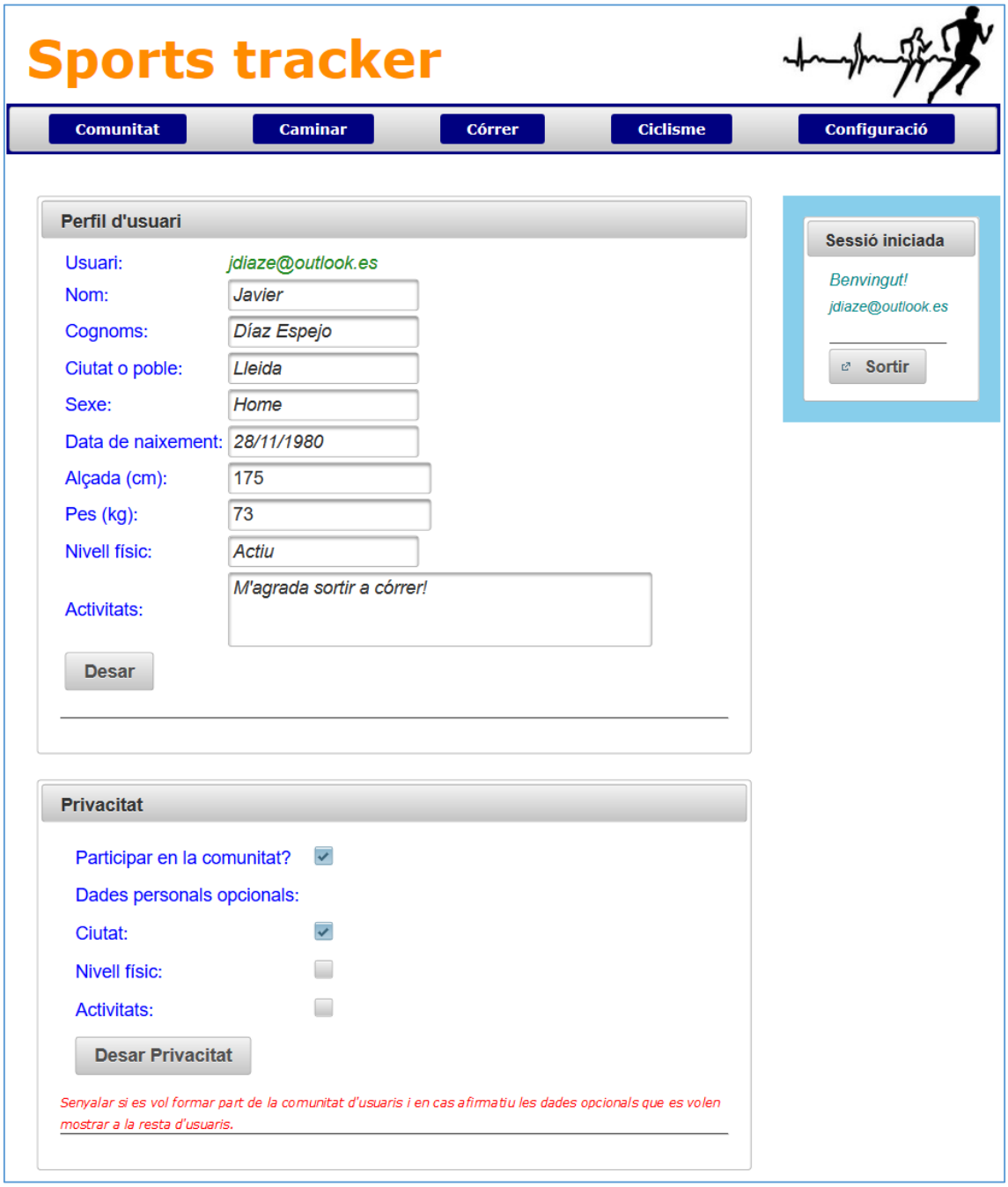
Es dóna d'alta en l'aplicació un nou usuari amb el nom d'usuari *jdiaze@outlook.es* i amb la contrasenya *1234ASd$* que compleix els requisits de l'aplicació. Alhora d'introduir la contrasenya la mateixa aplicació ens retorna un *feedback* sobre la fortalesa d'aquesta, tal com es pot visualitzar en la il·lustració següent:




*Il·lustració 49.Creació d'un nou usuari per realització de proves.*

- *Modificació de les dades personals del perfil del nou usuari creat:*

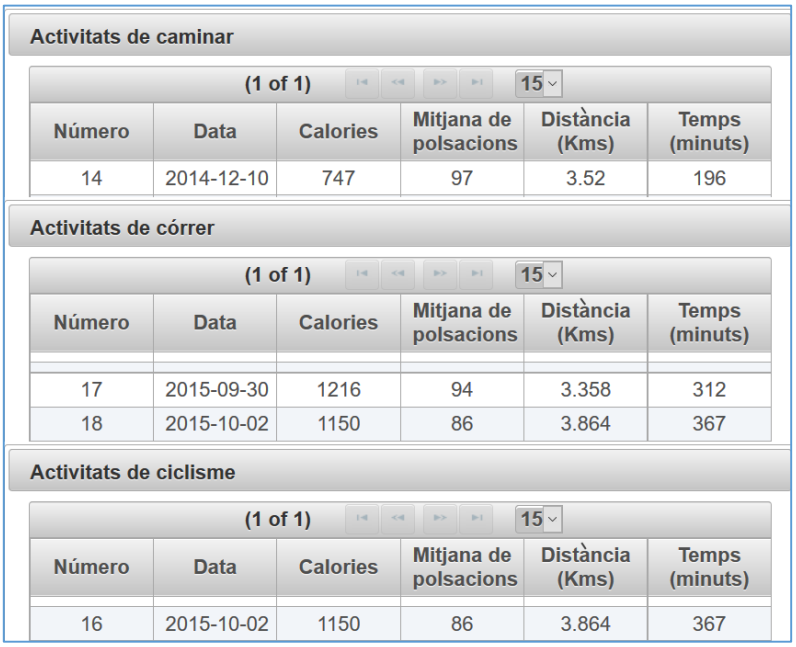
Un cop creat l'usuari s'autentica des de la pàgina d'inici i es visualitza la pàgina de configuració. S'omple tots els camps de dades personals i es clica el botó de desar. Seguidament, en l'apartat de privacitat s'assenyala l'opció de participar en la comunitat i a més de mostrar la dada personal de ciutat, es desa la privacitat. A continuació la següent il·lustració mostra el resultat final:




*Il·lustració 50. Dades del perfil d'usuari creat per a proves.*

- *Inserció d'activitats esportives per a l'usuari en les tres modalitats diferents:*

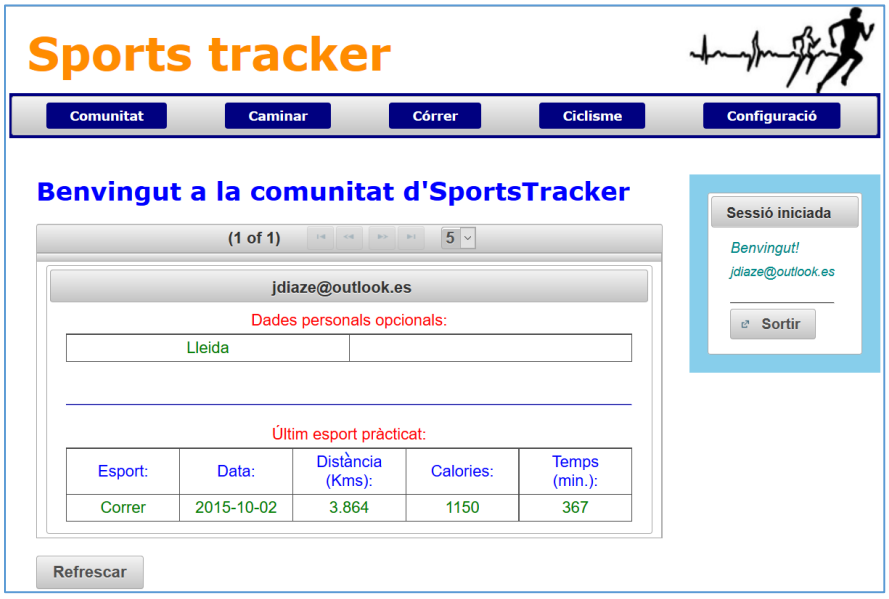
Amb ajuda de tres fitxers de proves *tcx* generats amb una tercera aplicació comercial s'insereix per a cada tipus d'esport almenys una activitat. Aquest fitxers es troben en la carpeta "proves" del projecte(*test_1.tcx, test_2.tcx i test_3.tcx*). La següent il·lustració mostra el resultat de les taules de les tres activitats esportives de l'aplicació:




*Il·lustració 51. Taules d'activitats de l'usuari de proves.*

- *Comprovació de les dades mostrades en la comunitat:*

Segons la privacitat que anteriorment s'ha configurat i desat per a l'usuari de proves, aquest participa a la comunitat i a més vol mostrar de la ciutat d'on és. La il·lustració següent comprova que en la pàgina de la comunitat és mostra l'usuari amb l'últim esport pujat a l'aplicació i mostrant el camp ciutat.




*Il·lustració 52.Pagina de la comunitat de l'usuari de proves.*

- *Realització de consultes en les diferents taules de la BD:*

Amb ajuda de l'eina de gestió *Workbench* de MySQL es visualitzen les dades que té emmagatzemades les diferents taules de l'usuari creat per a les proves. La il·lustració següent mostra la consulta realitzada a cadascuna de les quatre taules de la BD utilitzades per l'aplicació.

**Taula d'usuaris:**

| user | password | bits_sal |
|---|---|---|
| jdiaze@outlook.es | 719e9433f81cc82674d07b79162d7c77467cffb88d524f3438bf48f3db6f8de6 | y8TEbdzl0SPnXak0TsJMX4r3luAcC8kj |

**Taula del perfil:**

| userId | name | lastName | city | sex | birthday | height | weight | fitLevel | activities | comunity |
|---|---|---|---|---|---|---|---|---|---|---|
| jdiaze@outlook.es | U2FsdGV... | U2FsdGVk... | Lleida | U2FsdG... | U2FsdGVkX... | U2FsdGVkX1... | U2FsdGVkX... | U2FsdG... | U2FsdGVkX... | 1100000 |

**Taula d'activitats esportives:**

| userId | id | heartRateAv | calories | distance | sport | date | duration |
|---|---|---|---|---|---|---|---|
| jdiaze@outlook.es | 14 | U2FsdGVkX184... | U2FsdGVkX1+5bZ... | U2FsdGVkX19Cpy... | U2FsdGVkX1+HgGLW... | U2FsdGVkX18VvwtpjtN... | U2FsdGVkX1/gfNEo/P7... |
| jdiaze@outlook.es | 16 | U2FsdGVkX1/4i... | U2FsdGVkX1/d4O... | U2FsdGVkX1/fTMrt... | U2FsdGVkX18bnkKrKm... | U2FsdGVkX1+GDpcgoR... | U2FsdGVkX19d2XfZYK... |
| jdiaze@outlook.es | 17 | U2FsdGVkX18A... | U2FsdGVkX191Pbj... | U2FsdGVkX1/7NPL... | U2FsdGVkX19snI2eS/... | U2FsdGVkX1+K+m3Sv... | U2FsdGVkX1/P8IGpHZ8... |
| jdiaze@outlook.es | 18 | U2FsdGVkX1/O... | U2FsdGVkX19NWe... | U2FsdGVkX1+d8J... | U2FsdGVkX19l//Vhshy... | U2FsdGVkX18ofchsJ9ip... | U2FsdGVkX19oM7PjAV... |

**Taula de la comunitat:**

| userId | sport | date | distance | calories | duration |
|---|---|---|---|---|---|
| jdiaze@outlook.es | Correr | 2015-10-02 | 3.864 | 1150 | 367 |

*Il·lustració 53. Consultes a les taules de la BD de l'usuari de proves.*

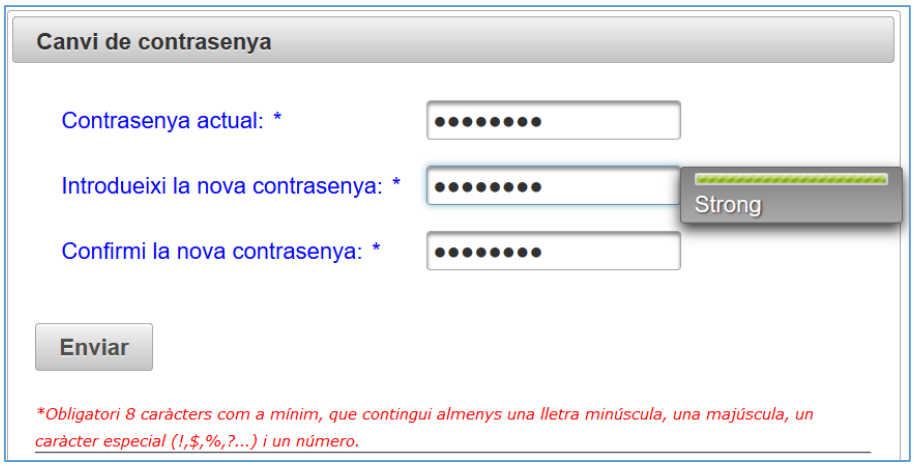
En la taula d'usuaris es pot observar com el camp *password* és un resum que s'ha generat a partir de la contrasenya més uns bits de sal. Els bits de sal de l'usuari són 32 caràcters generats aleatòriament en el moment del registre.

En la taula del perfil es comprova que totes les dades estan xifrades, excepte una que és el camp *city* i el camp *comunity*. El camp de la ciutat està en clar ja que l'usuari l'ha volgut compartir com a dada personal a mostrar en la comunitat de l'aplicació. El camp *comunity* és un camp utilitzat per la pròpia aplicació i no comporta la visualització de cap dada personal.

En la taula d'activitats esportives, excepte els identificadors únics que són nombre enters i claus primàries, la resta de camps estan tots xifrats. En canvi en la taula de la comunitat a on s'emmagatzema l'última activitat practicada per l'usuari estan totes les dades en clar, ja que són les que es visualitzen en la comunitat a tota la resta d'usuaris.

- *Canvi de contrasenya:*

Es realitza un canvi de contrasenya de l'usuari per verificar que al tornar a autenticar-se en l'aplicació les dades es continuen llegint correctament, i per tant les dades xifrades no depenen de la contrasenya de l'usuari. En la il·lustració de la pàgina següent és mostra l'apartat de canvi de contrasenya ubicat en la pàgina de configuració, i com en el cas del registre per a la nova contrasenya l'aplicació ens retorna un *feedback* sobre la fortalesa d'aquesta.




*Il·lustració 54. Canvi de contrasenya de l'usuari de proves del prototip.*

- *Cost computacional de l'aplicació web segura en el servidor:*

El cost computacional de l'aplicació en el servidor web no s'incrementa respecte altres aplicacions similars no segures, tot i què l'aplicació desenvolupada inclou la capa de seguretat del xifratge de les dades. A causa que el xifrar i el desxifrar s'executa en els ordinadors dels clients, el servidor queda lliure de realitzar aquesta tasca i únicament serveix la informació que té emmagatzemada en la BD. Per tant, la càrrega del servidor dependrà principalment del nombre d'usuaris que tingui l'aplicació web, sense grans variacions comparada amb qualsevol altra aplicació similar no segura.

# Capítol 6. Conclusions finals

Un cop realitzat el disseny i desenvolupat el prototip bàsic de l'aplicació web del *Sports Tracker*, se'n poden extreure unes conclusions finals del treball desenvolupat. L'objectiu principal del projecte és desenvolupar una aplicació web segura per gestionar les activitats esportives que practiquen els usuaris. En el disseny com en la implementació s'ha utilitzat les eines adequades per poder assegurar aquest objectiu, i per tant es pot dir que l'objectiu principal s'ha assolit amb èxit.

L'aplicació web desenvolupada treballa amb les dades personals dels usuaris a través dels perfils d'aquests com amb les dades dels diferents esports practicats i que han desat en l'aplicació. Per poder assegurar aquestes dades el projecte s'ha basat en tres pilars fonamentals respecte la seguretat de l'aplicació: la connexió segura, l'autenticació dels usuaris i la confidencialitat de les dades.

L'aplicació web dissenyada i desenvolupada segueix una arquitectura híbrida, per una part la de client-servidor i per l'altra part una arquitectura per capes. En la part del servidor s'emmagatzema totes les dades personals dels usuaris xifrades, excepte d'algunes concretes que l'usuari vol compartir amb la resta de la comunitat de l'aplicació. La transferència de dades entre el client i el servidor es realitza a través d'un canal segur sobre la xarxa pública d'Internet, el qual permet verificar la identitat del servidor per part de l'usuari. En canvi, el servidor verifica la identitat dels usuaris a través del mètode d'autenticació nom d'usuari i contrasenya. La seguretat implementada intenta impedir atacs exteriors però també possibles atacs des de dintre del mateix sistema.

Els protocols criptogràfics són les eines bàsiques que han permès afegir les diferents capes de seguretat en l'aplicació web. En l'actualitat, l'aplicació desenvolupada pot resistir una gran diversitat d'atacs com els especificats en el capítol 2 d'aquest projecte. Tot i així hi ha una part de la seguretat que encara recau en l'usuari i dependrà d'aquest i dels seus actes que un possible atacant es beneficiï. Addicionalment, no es pot assegurar que una aplicació pugui ser cent per cent segura, ja que en el món de la seguretat cada dia es troben falles que poden obrir bretxes de seguretat dins d'una aplicació. Per això és bàsic una actualització constant dels diferents protocols i tècniques criptogràfiques utilitzades. Una falla de seguretat pot ser aprofitada per un atacant que l'hagi descobert fins que es faci pública, i per tant els atacants sempre aniran un pas endavant en la seguretat.

Com a conclusió final, es pot dir que una aplicació segura disposa d'un conjunt de capes de seguretat que intenten fer invulnerable el sistema, tot i què una fiabilitat del cent per cent en seguretat és impossible. Es tracta de posar les màximes traves possibles als atacants per a què el nostre sistema sigui el menys fàcil i rentable d'atacar.

## 6.1 Línies de treball futur

El desenvolupament del projecte s'ha basat en el disseny i implementació d'una aplicació web d'*Sports Tracker* on l'usuari pot pujar les seves activitats esportives i desar certes dades personals. Una evolució natural del sistema desenvolupat, tal com ofereixen moltes aplicacions comercials, és el desenvolupament d'una aplicació mòbil per a les diferents plataformes existents al mercat. L'aplicació mòbil, a més d'oferir la pròpia gestió de l'aplicació similar a la versió web, aprofitaria l'ús dels sensors dels dispositius mòbils per enregistrar directament les activitats esportives d'un usuari. Per tant, l'ús d'una tercera aplicació per enregistrar les activitats esportives i generar els fitxers *tcx* o similars per importar les dades a l'aplicació, passaria a ser una característica complementaria.

En el cas de l'aplicació web desenvolupada encara tindria un ventall de possibilitats de noves característiques i millores. L'aplicació desenvolupada en l'apartat de la implementació és un prototip, i per tant la seva missió principal és la de mostrar una base sòlida respecte a la seguretat a implementar en l'aplicació. Un treball futur sobre el prototip és l'acabament de polir certes característiques i l'afegiment d'altres com: gràfiques, filtres de cerca, rutes d'activitats, la importació d'altres extensions de fitxers,...

El projecte és una base per la construcció d'un futur sistema complet d'aplicacions d'*Sports Tracker* que ofereixin un plus de seguretat als usuaris i que les aplicacions comercials actuals no hi tenen.

# Glossari

**Criptografia:** terme d'origen grec que prové dels mots *krypto* ("amagar") i *grapho* ("escriure"). Podem dir que la criptografia és la ciència i l'estudi de l'escriptura secreta.

**Xifratge:** procés de transformació d'un text en clar en un text xifrat.

**Clau:** paràmetre, normalment secret, que controla els processos de xifratge i/o de desxifratge.

**Xifra simètrica:** xifra en què tant l'emissor com el receptor comparteixen una sola clau que fan servir tant per a xifrar com per a desxifrar.

**Xifra de clau pública:** xifra on l'emissor disposa de dues claus, una pública i una privada, i el receptor de dues més. El receptor solament coneix la clau pública de l'emissor i viceversa.

**Funció unidireccional:** funció unidireccional invertible que és fàcil de calcular en sentit directe i difícil de calcular en sentit invers.

**Funció *hash* unidireccional:** funció que dóna com a sortida un resum de longitud fixa a partir d'una entrada consistent en un missatge arbitràriament llarg, i que a més és unidireccional.

**Certificat digital:** estructura de dades que vincula una identitat amb una clau pública. El certificat és emès per una autoritat de certificació.

**Autoritat de certificació:** autoritat de confiança que emet certificats. Sigla: CA

**Atac:** estratègia o mètode que té per objectiu descobrir la clau de xifratge o bé el text en clar. Els atacs criptoanalítics exploten les febleses dels algorismes de xifra. Els atacs als sistemes informàtics i de comunicacions exploten les vulnerabilitats d'aquests sistemes.

**Autentificació:** comprovació de l'autenticitat.

**Integritat:** propietat de no haver sofert, en relació amb la informació, modificacions ni supressions parcials no autoritzades.

**Privacitat:** dret de les persones a salvaguardar la seva intimitat, especialment pel que fa a les dades de què disposen les entitats públiques o privades.

**AES** (*Advanced Encryption Standard*): criptosistema Rijndael, que xifra blocs de 128 bits per mitjà d'una clau que pot variar la longitud entre 128, 192 o 256 bits.

**RSA:** criptosistema de clau pública molt utilitzat, publicat per Rivest, Shamir i Adleman l'any 1978; es basa en el problema de la factorització.

**CBC** (*Cipher Bloc Chaining*): mode de xifratge de bloc en què es crea un encadenament dels blocs, de manera que el xifratge d'un bloc depèn de l'anterior per mitjà d'un bloc inicial aleatori per al xifratge.

**Entropia de Shannon:** entropia d'una variable aleatòria, que mesura, en bits, la incertesa sobre el valor que prendrà la variable.

**XML:** acrònim de l'expressió anglesa *extensible mark-up language*. Especificació que defineix una sintaxi i unes regles sobre l'ús d'etiquetes per a estructurar la informació.

**TCX:** és un format d'intercanvi de dades introduït en 2007 com a part del producte *Garmin's Training Center*. TCX estableix l'estructura per a la transferència de la freqüència cardíaca, la cadència de carrera, la cadència de bicicleta, les calories... També proporciona dades de resum en forma de voltes.

**OpenSSL:** consisteix en un robust paquet d'eines d'administració i biblioteques relacionades amb la criptografia, que subministren funcions criptogràfiques a altres paquets com OpenSSH i navegadors web.

**SHA** (*Secure Hash Algorithm*): és una família de funcions *hash* publicades per l'Institut Nacional d'Estàndards i Tecnologia (NIST) dels EUA.

**HTML** (*HyperText Mark-up Language*): és un estàndard que serveix de referència per a l'elaboració de pàgines web, defineix una estructura bàsica i un codi per a la definició de contingut d'una pàgina web, com text, imatges, vídeos, entre d'altres.

**JavaScript** (*abreujat comunament "JS"*): és un llenguatge de programació interpretat, dialecte de l'estàndard ECMAScript. Es defineix com orientat a objectes, basat en prototips, imperatiu, dèbilment tipat i dinàmic.

**JSF** (*JavaServer Faces*): és una tecnologia i *framework* per a aplicacions Java basades en web que simplifica el desenvolupament d'interfícies d'usuari en aplicacions Java EE.

**GlassFish:** és un servidor d'aplicacions de programari lliure desenvolupat per *Sun Microsystems*, companyia adquirida per Oracle Corporation, que implementa les tecnologies definides en la plataforma Java EE i permet executar aplicacions que segueixen aquesta especificació.

**Framework:** en el desenvolupament de programari, un *framework* o infraestructura digital, és una estructura conceptual i tecnològica de suport definit, normalment amb artefactes o mòduls concrets de programari, que pot servir de base per a l'organització i desenvolupament de programari.

# Bibliografia

[1] **Project Management Institute, Inc**. *A guide to the project management body of knowledge (PMBOK Guide)*. ISBN: 978-1-933890-51-7. PMI Publications, EUA. Quarta edició: 2008.

[2] **Camps i Riba, JM**. *J2EE Una plataforma de components distribuïda*. P06/11059/01150. Editorial FUOC. Primera edició: setembre 2006.

[3] **Herrera Joancomartí, J**. *Xifres de clau compartida: xifres de bloc*. P05/05024/00979. ISBN: 84-9788-388-8. Editorial FUOC. Tercera edició: febrer 2006.

[4] **Domingo Ferrer, J**. *Xifres de clau pública*. P05/05024/00980. ISBN: 84-9788-388-8. Editorial FUOC. Tercera edició: febrer 2006.

[5] **Rifà Pous, H**. *Infraestructura de clau pública PKI*. P05/05024/00982. ISBN: 84-9788-388-8. Editorial FUOC. Tercera edició: febrer 2006.

[6] **Perramon, X**. *Mecanismes de protecció*. P07/05070/02624. Editorial FUOC. Segona edició: febrer 2008.

[7] **Alfred J. Menezes**, **Paul C. van Oorschot** i **Scott A**. **Vanstone**. *Handbook of Applied Cryptography*. ISBN: 0-8493-8523-7. CRC Press. Cinquena edició: agost 2001.

[8] **Oppliger, R**. *Security technologies for the Word Wide Web*. Artech House. 2000

[9] **Viquipèdia**. *Zero-day (computing)*. [en línia] Pàgina web: https://en.wikipedia.org/wiki/Zero-day_(computing). Data de consulta: 18 d'octubre de 2015.

[10] **OWASP**. *Comunitat lliure i oberta sobre seguretat en aplicacions*. [en línia] Pàgina web: https://www.owasp.org/index.php/MainPage. Data de consulta: 9 d'octubre de 2015.

[11] **Chema Alonso**. *Un informático en el lado del mal*. [en línia] Pàgina web: http://www.elladodelmal.com. Data de consulta: del 9 al 14 d'octubre de 2015.

[12] **Acens Technologies** (a Telefonica company). *Whitepaper: Base de datos y sus vulnerabilidades más comunes*. [en línia] Pàgina web: http://www.acens.com/wp-content/images/2015/03/vuln erabilidades-bbdd-wp-acens.pdf. Data de consulta: 10 d'octubre de 2015.

[13] **Info Spyware**. *¿Qué es el Phising?* Article: novembre 2008. [en línia] Pàgina web: https://www.in fospyware.com/articulos/que-es-el-phishing/. Data de consulta: 12 d'octubre de 2015.

[14] **Viquipèdia**. *Phishing*. [en línia] Pàgina web: https://es.wikipedia.org/wiki/Phishing#Origendelt.C3.A9rmino. Data de consulta: 12 d'octubre de 2015.

[15] **Viquipèdia**. *Moxie Marlinspike*. [en línia] Pàgina web: https://en.wikipedia.org/wiki/Moxie_Marlinspike. Data de consulta: 19 d'octubre de 2015.

[16] **Sanders, C**. (WindowSecurity.com) *Understanding Man-In-The-Middle Attacks* (Part 1 to Part 4). Article: juny de 2010. [en línia] Pàgina web: http://www.windowsecurity.com/articles-tutorials/authentication_and_encryption/Understanding-Man-in-the-Middle-Attacks-ARP-Part1.html. Data de consulta: 13 d'octubre de 2015.

[17] **Rodríguez Varela, J**. *5 cosas que debes saber sobre Heartbleed*. WeLiveSecurity. Publicat: abril de 2014 [en línia] Pàgina web: http://www.welivesecurity.com/la-es/2014/04/09/5-cosas-debes-saber-sobre-heartbleed/. Data de consulta: 19 d'octubre de 2015.

[18] **BitEndian. Enginyers informàtics**. *¿Qué es el man-in-the-middle?* Publicat: març de 2015 [en línia] Pàgina web: http://www.bitendian.com/es/que-es-el-man-in-the-middle/. Data de consulta: 14 d'octubre de 2015.

[19] **Dr. Alfonso Muñoz** (UPM). *Lección 9: Introducción al protocolo SSL.* intypedia: Information Security Encyclopedia. R&D Security Researcher. T>SIC Group — UPM. [en línia] Pàgina web: http://www.criptored.upm.es/intypedia/docs/es/video9/DiapositivasIntypedia009.pdf. Data de consulta: 22 d'octubre de 2015. Publicat: juliol de 2011; Madrid.

[20] **Ramírez López, D.O.**; **Espinosa Madrigal, C.C.** (.Seguridad) *El cifrado Web (SSL/TLS)*. Revista de la universitat nacional autònoma de Mèxic. Publicat: edició número 10 de maig de 2011. [en línia] Pàgina web: http://revista.seguridad.unam.mx/numero-10/el-cifrado-web-ssltls. Data de consulta: 22 d'octubre de 2015.

[21] **Viquipèdia**. *Transport Layer Security*. [en línia] Pàgina web: https://es.wikipedia.org/wiki/Trans port_Layer_Security. Data de consulta: 23 d'octubre de 2015.

[22] **De los Santos, S**. (ElevenPaths blog offers) *Perfect Forward Secrecy: ¿Existe el secreto perfecto y permanente?* Publicat: maig de 2014 [en línia] Pàgina web: http://blog.elevenpaths.com/2014/05/perfect-forward-secrecy-existe-el.html. Data de consulta: 23 d'octubre de 2015.

[23] **Viquipèdia**. *Advanced Encryption Standard*. [en línia] Pàgina web: https://es.wikipedia.org/wiki/Advanced_Encryption_Standard. Data de consulta: 25 d'octubre de 2015.

[24] **Oracle Corporation**. *JavaServer Faces Technology*. [en línia] Pàgina web: http://www.oracle.com/technetwork/java/javaee/javaserverfaces-139869.html. Data de consulta: 26 d'octubre de 2015.

[25] **w3schools**. *HTML, CSS and JavaScript*. [en línia] Pàgina web: http://www.w3schools.com. Data de consulta: 10 de novembre de 2015.

[26] **Oracle Corporation and/or its affiliates**. *MySQL*. [en línia] Pàgina web: https://www.mysql.com. Data de consulta: 10 de novembre de 2015.

[27] **Oracle Corporation and/or its affiliates**. *GlassFish - World's first Java EE 7 Application Server*. [en línia] Pàgina web: https://glassfish.java.net. Data de consulta: 10 de novembre de 2015.

[28] **Wu, T**. *The Secure Remote Password Protocol*. Computer Science Department. Stanford University EUA. [en línia] Pàgina web: http://srp.stanford.edu/ndss.html. Data de consulta: 10 de desembre de 2015.

[29] **Microsoft**. *Sugerencias para crear una contraseña segura*. [en línia] Pàgina web: http://windows.microsoft.com/es-es/windows-vista/tips-for-creating-a-strong-password. Data de consulta: 2 de novembre de 2015.

[30] **Viquipèdia**. *Password strength*. [en línia] Pàgina web: https://en.wikipedia.org/wiki/Password_strength. Data de consulta: 3 de novembre de 2015.

[31] **Apple**. *Preguntas frecuentes acerca de la verificación en dos pasos del ID de Apple* [en línia] Pàgina web: https://support.apple.com/es-es/HT204152. Data de consulta: 3 de novembre de 2015.

[32] **Google**. *Verificación en dos pasos*. [en línia] Pàgina web: https://support.google.com/accounts/topic/28786?hl=es&ref_topic=3382253. Data de consulta: 3 de novembre de 2015.

[33] **Viquipèdia**. *Block cipher mode of operation*. [en línia] Pàgina web: https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation. Data de consulta: 25 d'octubre de 2015.

[34] **Garmin**. *Definició d'estructura de fitxers TCX*. [en línia] Pàgina web: http://www8.garmin.com/xmlschemas/TrainingCenterDatabasev2.xsd. Data de consulta: 12 de novembre de 2015.

# Annexos

## Creació i gestió de certificats

Seguidament es defineixen el punts a seguir per la generació dels dos certificats i parell de claus per al servidor.

1. **Amb l'eina Keytool:**

Generació del parell de claus per al certificat del servidor (s'emmagatzema en keystore.jks): keytool -genkeypair -keyalg RSA -keystore ./config/keystore.jks -validity 1000 -alias s1as

Creació d'un certificat de petició per al servidor: keytool -certreq -alias s1as -filesportsTracker.csr -keystore ./config/keystore.jks

2. **Amb l'eina OpenSSL:**

Generació d'un parell de claus per CA: openssl genrsa -out rootCA.key 1024

Creació d'un certificat de petició per CA: openssl req -new -key rootCA.key -out rootCA.csr

Auto-signat del certificat de petició CA: openssl x509 -req -days 3650 -in rootCA.csr -signkey rootCA.key -out rootCA.crt (s'obté rootCA.crt)

Signament del certificat de petició del servidor amb el certificat anterior CA: openssl x509 -req -days 500 -in sportsTracker.csr -CA rootCA.crt -CAkey rootCA.key -out sportsTracker.crt –Cacreateserial (s'obté sportsTracker.crt)

3. **Amb l'eina Keytool:**

Importació del certificat de confiança (rootCA.crt) al contenidor de CA's: keytool -import -v -trustcacerts -alias rootCA -file rootCA.crt -keystore ./config/cacerts.jks

Importació del certificat de confiança (rootCA.crt) al contenidor de claus privades: keytool -import -v -trustcacerts -alias rootCA -file rootCA.crt -keystore ./config/keystore.jks

Importació del certificat del servidor (sportsTracker.crt) al contenidor de claus privades: keytool -import -v -trustcacerts -alias s1as -file sportsTracker.crt -keystore ./config/keystore.jks

# Estructura del fitxer TCX

A continuació s'especifica una estructura d'exemple dels fitxers *tcx* utilitzats en les proves del prototip:

```
<?xml version="1.0" encoding="UTF-8"?>
<TrainingCenterDatabase xmlns="http://www.garmin.com/xmlschemas/TrainingCenterDatabase/v2">
    <Activities>
        <Activity Sport="Other">
            <Id>2014-12-10T15:59:13.000Z</Id>
            <Lap StartTime="2014-12-10T15:59:13.000Z">
                <TotalTimeSeconds>1579.0</TotalTimeSeconds>
                <DistanceMeters>446.79998779296875</DistanceMeters>
                <MaximumSpeed>8.499999046325684</MaximumSpeed>
                <Calories>747</Calories>
                <AverageHeartRateBpm><Value>97</Value></AverageHeartRateBpm>
                <MaximumHeartRateBpm><Value>119</Value></MaximumHeartRateBpm>
                <Intensity>Active</Intensity>
                <TriggerMethod>Manual</TriggerMethod>
                <Track>
                    <Trackpoint>
                        <Time>2014-12-10T16:00:21.000Z</Time>
                        <Position>
                            <LatitudeDegrees>41.61667283</LatitudeDegrees>
                            <LongitudeDegrees>0.661049</LongitudeDegrees>
                        </Position>
                        <AltitudeMeters>0.0</AltitudeMeters>
                        <DistanceMeters>0.0</DistanceMeters>
                        <HeartRateBpm><Value>113</Value></HeartRateBpm>
                        <SensorState>Present</SensorState>
                    </Trackpoint>
                </Track>
            </Lap>
            <Training xmlns=http://www.garmin.com/xmlschemas/TrainingCenterDatabase/v2
                VirtualPartner="false">
                <Plan Type="Workout" IntervalWorkout="false">
                    <Name>Caminata</Name>
                    <Extensions></Extensions>
                </Plan>
            </Training>
            <Creator xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="Device_t">
                <Name>Polar M400</Name>
                <UnitId>0</UnitId>
                <ProductID>22</ProductID>
            </Creator>
        </Activity>
    </Activities>
    <Author xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="Application_t">
        <Name>no application</Name>
    </Author>
</TrainingCenterDatabase>
```

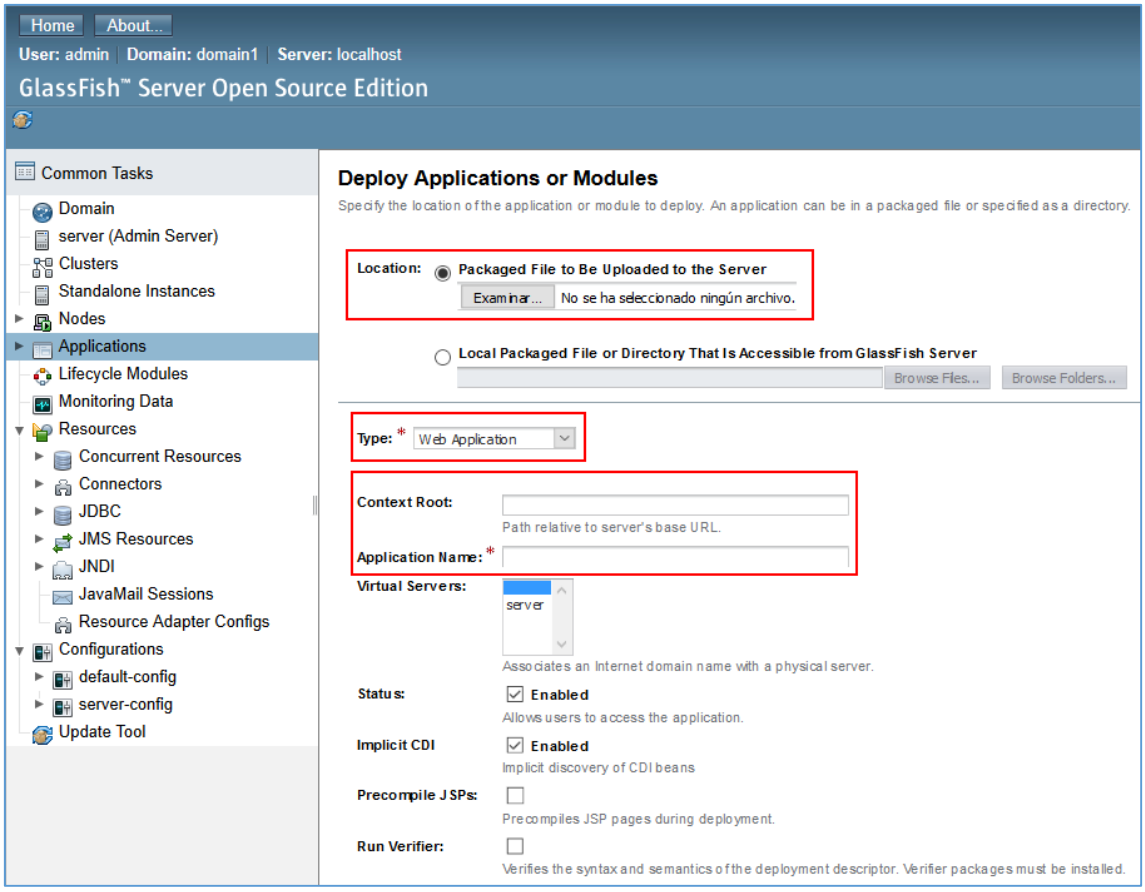
Per visualitzar l'estructura completa dels fitxers *tcx* es pot visitar el lloc web[26] de Garmin, a on es defineix amb exactitud l'estructura completa de la versió 2 d'aquest tipus de fitxers *xml*.

[26] Visitar la referència bibliogràfica [34].

# Desplegament de l'aplicació en el servidor

En el conjunt de carpetes del projecte lliurat existeix una carpeta anomenada *dist*, la qual conté un fitxer amb extensió .war (SportsTracker.war). Aquest tipus de fitxer és un arxiu JAR utilitzat per a distribuir una col·lecció de diferents elements com *JavaServer Pages*, *servlets*, classes Java, arxius XML, llibreries d'etiquetes i pàgines web estàtiques. El conjunt de tots aquests elements formen una aplicació web.

Per tant, per al desplegament de l'aplicació desenvolupada en el projecte s'ha de desplegar amb ajuda del servidor d'aplicacions *Glassfish* i l'arxiu *war*. La següent il·lustració mostra la pantalla de desplegament d'aplicacions del servidor amb l'opció de seleccionar un fitxer *war* per al desplegament d'una aplicació i la configuració bàsica.




*Il·lustració 55. Desplegament d'aplicació en Glassfish.*