

SPRINT 5

Index

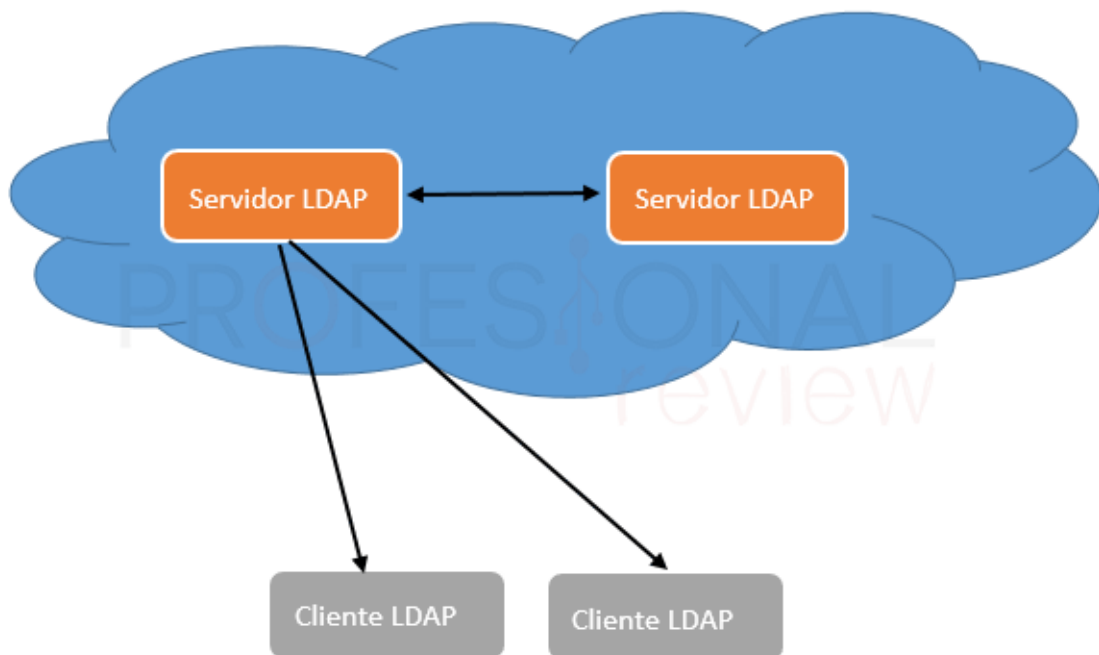
1. Servidor DNS
2. Servidor LDAP
3. Codi Embegut
4. API Externa
5. API RESTful
6. Control d'errors i exeptions
7. DOM
8. Vue

Introducció

LDAP

Que es?

Es tracta d'un conjunt de protocols de llicència oberta que són utilitzades per accedir a la informació que està emmagatzemada de forma centralitzada en una xarxa. Ldap es un protocol basat en la connexió entre client-servidor. En el servidor s'almacenaràn les dades relatives al directori.



Quan el client LDAP es connecta amb el servidor, podrà realitzar dues accions bàsiques, bé consultar i obtenir informació de directori, o modificar-la.

- **Si un client consulta la informació del servidor LDAP** pot connectar-la directament si tenen un directori allotjat en ell, o redirigir la sol·licitud a un altre servidor
- **Si un client vol modificar la informació de directori**, el servidor comprovarà si l'usuari que està accedint a aquest directori té permisos d'administrador o no

La informació a LDAP es pot almacenar de diferents formes.

- **Entrades**, trucades objectes en Active Directory. Aquestes entrades són col·leccions d'atributs amb un nom Distingit (DN).
- **Atributs**: els quals posseeixen un tipus identificador i els corresponents valors. Els tipus s'utilitzen per identificar els noms d'atributs, per exemple "mail", "name",

- **LDIF:** el Format d'Intercanvi de Dades de LDAP és la representació en text ASCII de les entrades LDAP.
- **Arbres:** És l'organització jerarquitzada d'entrades.

DNS

Qué significa DNS.

Quan es vol accedir a una pàgina web a Internet es necessita l'adreça IP del servidor on està emmagatzemada, però, per regla general, l'usuari només coneix el nom del domini. La raó no és altra que la complicació en recordar les sèries numèriques de l'tipus 93.184.216.34 que les componen, que són les que, precisament, constitueixen la base de la comunicació a Internet. És per aquest motiu pel qual les adreces IP es "tradueixen" en noms que puguem recordar, aquests noms els anomenem dominis.

Exemple:

Direcció IP: 86.231.213.55

Domini: www.exemple.cat

Que son les peticions al DNS?

Quan s'introdueix l'adreça d'una pàgina web (URL) en el camp de recerca de el navegador, aquest realitza una petició a l'anomenat **resolver**, un component especial de sistema operatiu que té la funció emmagatzemar en la memòria caché adreces IP ja sol·licitades anteriorment, i proporcionar-les quan l'aplicació client (navegador o programa de correu) la demana. Si l'adreça IP sol·licitada no es troba en la memòria caché del **resolver**, aquest redirigeix la petició a servidor DNS que correspongui, que, en general, es tracta de servidor DNS de proveïdor d'Internet.

Si un servidor DNS no pot respondre a una petició amb la informació de què disposa a la base de dades, pot sol·licitar la informació a un altre servidor o reenviar la petició al servidor DNS que li correspongui. Aquesta resolució es pot realitzar de dues formes:

- **Resolució recursiva:** és la que es produeix quan el servidor DNS no pot respondre per si mateix a una petició i pren la informació sobre un altre servidor. El **resolver** transfereix la petició completa al seu servidor DNS, que proporciona al seu torn la resposta al·resoldre amb el nom de domini, si s'ha resolt.
- **Resolució iterativa:** quan el servidor DNS no pot resoldre la petició, envia com a resposta la direcció de el següent servidor DNS de la jerarquia. L'·resoldre ha d'enviar ell mateix una nova petició i repetir la maniobra fins que es resol el nom de domini.

Instal·lar i configurar servidor DNS.

Després d'actualitzar els repositoris executarem la comanda per a instal·lar bind9.

```
alumne@pc-alumne:~$ sudo apt-get install bind9
```

També instal·larem els paquets que ens suggereix la instal·lació.

```
Paquetes sugeridos:  
bind9-doc resolvconf rblcheck python-ply-doc
```

Un cop tot instal·lat comprovarem que el servei DNS s'ha iniciat.

```
alumne@pc-alumne:~$ ps -ef | grep named  
bind      11110      1  0 16:25 ?        00:00:00 /usr/sbin/named -f -u bind  
alumne    12644    9160  0 16:32 pts/0    00:00:00 grep --color=auto named
```

També podem comprovar si el servei de DNS està a l'escolta dels ports 53 TCP i 53 UDP.

```
alumne@pc-alumne:~$ netstat -ltun | grep :53  
tcp        0      0 192.168.224.108:53      0.0.0.0:*        ESCUCHAR  
tcp        0      0 127.0.0.1:53            0.0.0.0:*        ESCUCHAR  
tcp        0      0 127.0.0.53:53          0.0.0.0:*        ESCUCHAR  
tcp6       0      0 :::53                   :::*          ESCUCHAR  
udp        0      0 192.168.224.108:53      0.0.0.0:*  
udp        0      0 127.0.0.1:53            0.0.0.0:*  
udp        0      0 127.0.0.53:53          0.0.0.0:*  
udp        0      0 224.0.0.251:5353       0.0.0.0:*  
udp        0      0 224.0.0.251:5353       0.0.0.0:*  
udp        0      0 0.0.0.0:5353           0.0.0.0:*  
udp6       0      0 :::53                   :::*  
udp6       0      0 :::5353                 :::*
```

A partir de Ubuntu 18 amb l'aparició de Netplan va canviar la forma en la que es configuraven els adaptadors de xarxa.

Accedirem l'arxiu de configuració de Netplan mitjançant aquesta comanda per a modificar-lo.

```
alumne@pc-alumne:~$ sudo nano /etc/netplan/01-network-manager-all.yaml
```

Un cop dins editarem els paràmetres corresponents.

```
# Let NetworkManager manage all devices on this system
network:
  version: 2
  renderer: networkd
  ethernets:
    wlp2s0:
      dhcp4: no
      dhcp6: no
      addresses: [192.168.224.203/24]
      gateway4: 192.168.224.1
      nameservers:
        addresses: [8.8.8.8, 8.8.4.4]
```

Guardarem els canvis amb (ctrl+o) i al sortir amb (ctrl+x) executarem la comanda que aplicara els canvis a Netplan.

```
alumne@pc-alumne:~$ sudo netplan apply
```

Per a comprovar que la nova ip estàtica esta configurada correctament podem fer ús de.

```
alumne@pc-alumne:~$ networkctl status
●      State: routable
    Address: 192.168.224.203 on wlp2s0
             fe80::c8ff:3298:60a5:aa7e on wlp2s0
    Gateway: 192.168.224.1 on wlp2s0
         DNS: 8.8.8.8
             8.8.4.4
```


VUE

Es un framework progressiu per a construir interfícies d'usuari. La llibreria central està enfocada només a la capa de visualització, i és fàcil d'utilitzar i integrar amb altres llibreries o projectes existents.



Instal·lació de Vue mitjançant un CLI

```
# Instalación con NPM
npm i -g @vue/cli @vue/cli-service-global

# Instalación con Yarn
yarn global add @vue/cli @vue/cli-service-global
```

Afegirem la primera comanda en cas de voler-lo instal·lar amb el gestor de paquets Npm, també podries utilitzar la segona comanda per a fer-ho amb Yarn encara que no es tan habitual.

Per a crear un projecte usarem la comanda:

```
alume@marcesp:~/Escritorio/tutorial-vue-main$ vue create nom-projecte
```

Un cop creat, entrarem a la carpeta del projecte e iniciarem el servidor.

```
alume@marcesp:~/Escritorio/tutorial-vue-main$ npm run serve
```

Sentencias de vue utilizadas

- v-model: serveix per a enllaçar variables am el nom indicat.
- v-for: es un bucle per a recorre algun arrai o el que vulguesem.
- v-if i v-else: es paregut al if else de java que li podem indicar sobre l'html.
- @click: es com el onclick de JS pero amb l'arroba davant i sense el on.
- \$emit: per a enviar informacio a un altre fitxer.
- import: per a importar fitxers amb les seues respectives variables
- let: per a crear variables igual que en JS
- filter: per a que deixe la informació com estava menos la que coincideix.

Exemple d'un codi de vue

```
<input
  ref="nombre"
  v-model="persona.nombre"
  type="text"
  class="form-control"
  :class="{ 'is-invalid': procesando && nombreInvalido }"
  @focus="resetEstado"
  @keypress="resetEstado"
/>
```

Resultat de una aplicació creada amb vue

Personas

Nombre Apellido Email

Nombre	Apellido	Email	Acciones	
<input type="text" value="Jon"/>	<input type="text" value="Nieve"/>	<input type="text" value="jon@email.com"/>	<input type="button" value="Guardar"/>	<input type="button" value="Cancelar"/>
Tyrion	Lannister	tyrion@email.com	<input type="button" value="Eliminar"/>	<input type="button" value="Editar"/>
Daenerys	Targaryen	daenerys@email.com	<input type="button" value="Eliminar"/>	<input type="button" value="Editar"/>

DOM

QuerySelector("exemple")

Agafa el primer element amb el nom indicat dins del parentesis.

Cambiar el color de fondo a blau del element amb la clase .canvi

```
document.querySelector(".canvi").style.backgroundColor = "blue";
```

QuerySelectorAll("exemple")

Agafa tots els elements amb el nom indicat dins del parentesis

A la x es guarden tots els elements seleccionats

```
x = document.querySelectorAll(".canvi");
```

Crear elements

Per a cobar element utilizarem la funcio cloneNode.

```
//Clonar elements
function addPara(mode){
    var p1 = document.getElementById("para1");
    var oldPara=p1.firstChild;
    var oldSalto=p1.childNodes[1];
    newPara=oldPara.cloneNode(mode);
    newSalto=oldSalto.cloneNode(mode);
    p1.appendChild(newPara);
    p1.appendChild(newSalto);
}
```

Clonar estils d'elements node

Per això agafarem el atribut style del node i el clonarem amb la funcio cloneNode.

I li assignarem amb setAttributeNode.

```
var p1Style=p1.getAttributeNode("style");
var cloneP1Style=p1Style.cloneNode(mode);
p2.setAttributeNode(cloneP1Style);
```

Crear nodes de text i elements

Per a crear nodes de text utilizarem la funcio createTextNode()

Per a crear elements utilizarem createElement().

```
var txt = document.createTextNode("Text creat amb JS!!");
var para2 = document.createElement("p");
```

Esborrar elements

Per a esborrar elements primer identificarem el que volem eliminar i després amb `removeChild(div.Mid.childNodes[0]);`

```
var divMid = document.getElementById("eliminar");  
divMid.removeChild(divMid.childNodes[0]);
```

Inserir abans

Amb `insertBefore` insertarem el primer parametre abans del segon parametre indicat.

```
divcr.insertBefore(pou, firstPara);
```

Copiar taula

Aquest es el codi per a copiar una taula que esta creada.

```
var tableCopy = document.getElementById("myTable");  
var newTable = tableCopy.cloneNode(true);  
newTable.id = "newtable_id";  
newDiv = document.getElementById("NewTableContainer");  
newDiv.appendChild(newTable);  
$("table").rowspanizer({  
    vertical_align: 'middle'  
});
```

WCAG2 Checklist

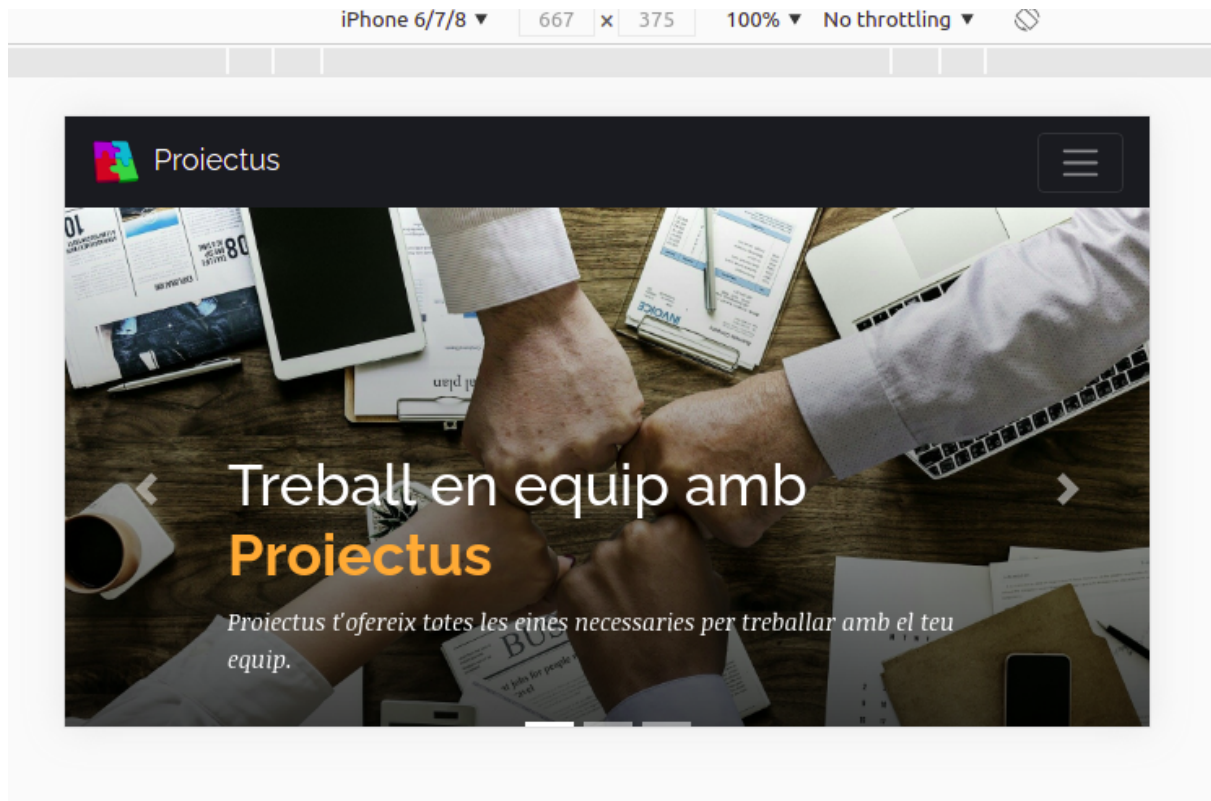
1.3.2 Meaningful Sequence (Level A)

Si es logic e intuitiu.



1.3.4 Orientation

El contingut es orienta automaticament.



1.3.5 Identify Purpose

Els inputs estan correctament indicats quin tipus es amb el atribut type.

```
<input type="text" ...
```

1.4.1 Use of Color

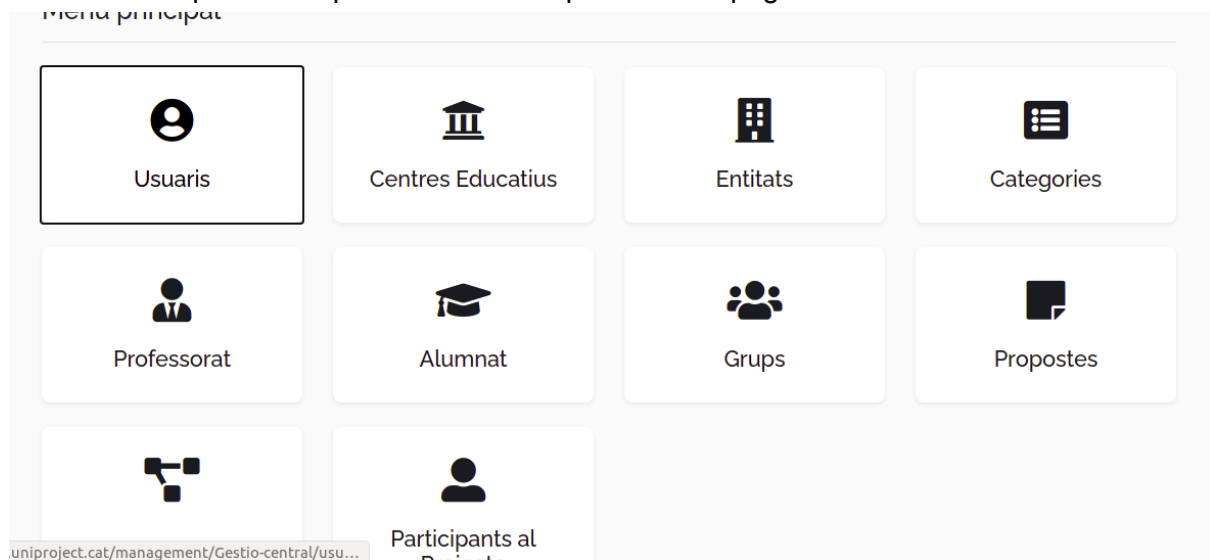
Als iconos i botons hi han etiquetes title per a la gent que no distinguen els colors be pugen sapiguer el color.

```
title="color groc">...
```

2.1.1 Keyboard

La navegacio del teclat la fa automaticament el chrome

Amb el tab es pot moure per les diferents opcions de la pagina.



2.1.2 No Keyboard Trap

El keyboard no es bloqueja i es pot utilitzar la pagina correctament amb nomes el teclat.

2.5.3 Label in Name

Si he posat un aria-label al boto de login per a que es pugue llegir per los programes especials

```
aria-label="Login">
```

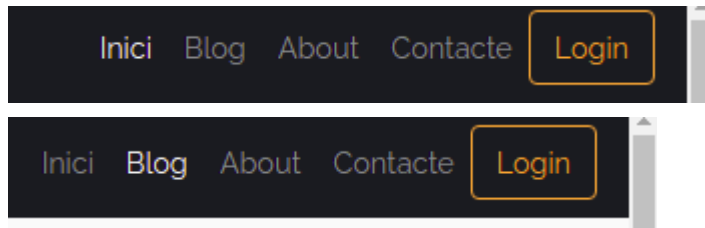

3.1.1 Language of Page

Si esta aplicat a la nostra pagina web.

```
<html lang="en">  
▶ <head>...</head>
```

3.2.3 Consistent Navegation

No es cambien els links de la nostra web.



3.3.4 Error Prevention

Tenim confirmacions a la nostra web.



Conclusió

Aquest treball hem estat un poquet despistats per la part de m6 i m9, hem anat molt justets pero bueno finalment ho hem aconseguit els objectius que voliem. Hem canviar de forma de treball i això ens ha anat bastant be i hem aconseguit els objectius buscats.

Webgrafía