



Escola Tècnica Superior d'Enginyeria
de Telecomunicació de Barcelona

UNIVERSITAT POLITÈCNICA DE CATALUNYA

PROJECTE FINAL DE CARRERA

WebGuide: Aplicació Web (HTML5) i Asterisk
utilitzant VoIP

(WebGuide: HTML5 and Asterisk application
using VoIP)

Estudis: Enginyeria de Telecomunicació

Autor: Albert Rodriguez Rodriguez

Director/a: Marcel Fernández Muñoz

Any: 2017



1. Índex general

Resum del Projecte	6
Resumen del Proyecto	7
Abstract	8
Introducció i objectius	12
Context del projecte	14
APPS vs HTML5	14
HTML5 o APPs?	18
Conceptes	21
VoIP	21
Funcionament del Protocol SIP	21
getUserMedia()	23
WebRTC	25
RTCWeb Breaker	27
Sipml5	28
Arquitectura de l'aplicació	30
Connexió i comunicació amb Asterisk	30
Lectura de QR	33
Funcionament de l'aplicació	37
Sessions i usuaris	34
Protocol SIP (usuaris)	39
Execució i problemes	41
Conclusions	46
Apèndix	49
Referències	58

2. Resum del Projecte

El projecte consisteix en crear una aplicació web que, en primera instància, podria actuar d'àudio-guia d'un museu tot i que pot tenir altres utilitats. Aquestes podrien ser, per exemple, un assistent a una tenda de roba o una aplicació d'ajuda per a cecs.

En aquest cas, l'usuari que visiti el museu, es podrà connectar a la xarxa Wifi des del seu terminal i amb el navegador web, es connectarà a la càmera del seu telèfon mòbil i escanejarà un codi QR situat al costat d'una obra d'art. Llavors, rebrà un àudio de veu amb l'explicació d'aquesta obra.

L'aplicació s'allotja en un servidor Linux Ubuntu Server. A dins d'aquest, hi ha un servidor Apache2, que junt amb llenguatge PHP és l'encarregat de proveir la interfície web i la gestió de sessions a l'usuari.

Per altra banda, també s'hi allotjarà un servidor Asterisk, que proporciona les funcionalitats d'una central telefònica. Asterisk gestionarà els usuaris SIP, rebrà les peticions SIP i enviarà els àudios als terminals mòbils. El protocol SIP és un estàndard per a la iniciació, manteniment i finalització de connexions multimèdia.

La connexió entre HTML5/JavaScript, és a dir, la plana web, i Asterisk, es produeix mitjançant la API WebRTC. Aquesta API té funcions per a connectar, enviar peticions i codificar/descodificar els àudios i vídeo que s'envien entre terminal mòbil i servidor.

Tant la informació de les sessions com la dels usuaris SIP, s'allotja en una base de dades PostgreSQL. La gestió d'aquesta base de dades, es realitza amb codi PHP inserit a dins del codi HTML5 o bé utilitzant el programa de gestió de PHPpgAdmin.

3. Resumen del Proyecto

El proyecto consiste en crear una aplicación web que, en primera instancia, podría actuar de audio-guía de un museo, a pesar de que puede tener otras utilidades. Estas podrían ser, por ejemplo, un asistente en una tienda de ropa o una aplicación de ayuda para ciegos.

En este caso, el usuario que visite el museo se podrá conectar a la red Wifi desde su terminal y, utilizando la cámara de su teléfono móvil, escaneará un código QR situado junto a una obra de arte. Entonces, recibirá un audio de voz con la explicación de esta obra.

La aplicación se aloja en un servidor Linux Ubuntu Server, dentro del cual hay un servidor Apache2, que junto con lenguaje PHP es el encargado de proveer la interfaz web y la gestión de sesiones al usuario.

Por otro lado, también se alojará un servidor Asterisk. Este programa proporciona las funcionalidades de una central telefónica. Asterisk gestionará los usuarios SIP, recibirá las peticiones SIP y enviará los audios a los terminales móviles. El protocolo SIP es un estándar para la iniciación, mantenimiento y finalización de conexiones multimedia.

La conexión entre HTML5/Javascript, es decir, la página web, y Asterisk, se produce mediante la API WebRTC. Esta API tiene funciones para conectar, enviar peticiones y codificar/descodificar los audios y vídeo que se envían entre terminal móvil y servidor.

Tanto la información de las sesiones como la de los usuarios SIP, se aloja en una base de datos PostgreSQL. La gestión de esta base de datos se realiza con código PHP insertado dentro del código HTML5 o bien utilizando el programa de gestión de PHPpgAdmin.

4. Abstract

The project consists in creating a web application that, in first instance, could act as an audio-guide of a museum, although this can have other utilities. These might be, for example, an assistant in a tent of clothes or an application for helping blind people.

In that case, the user that visits the museum will be able to connect to the Wifi network from his terminal and using his phone càmera, he will be able to scan a QR code situated next to an artwork. Then, he will receive an audio with the explanation of that artwork.

The application lays in a server Linux Ubuntu Server. Inside this server, there is an Apache2 server, than together with PHP language is the responsible to provide the web interface and the management of sessions to the user.

In the other hand, an Asterisk server will also be lodged. This program provides the functionalities of a telephonic head office. Asterisk will manage the SIP users, this will receive the SIP requests and this will also send the audios to the mobile terminal. The SIP protocol is an standard used for the initiation, maintenance and ending of multimedia connections.

The connection between HTML5/JavaScript, that means, the web page, and Asterisk, is performed by means of the API WebRTC. This API has functions to connect, send requests and encode/decode the audios and video sended between the mobile terminal and the server.

Information of the sessions as well as information of the SIP users, lays in a PostgreSQL database. Management of this database, is done with PHP code inserted in the HTML5 code or using the program of management of PHPpgAdmin.

5. Introducció i Objectius

En aquest projecte, s'ha fixat com a objectiu la realització d'una aplicació web funcional mitjançant la qual un usuari, llegint un codi QR, pugui accedir a un contingut en format d'àudio.

Aquest projecte ve motivat pel professor Marcel Fernández i departament de telemàtica (ENTEL) de l'ETSETB i les la possibilitat d'explorar noves tecnologies i adaptar-les a Asterisk.



L'aplicació, utilitzarà les noves funcionalitats que ofereix HTML5 en els camps de connexió i reproducció multimèdia. En concret, la connexió a un servidor Asterisk utilitzant una API de JavaScript anomenada WebRTC. S'utilitzarà el protocol SIP per a establir, mantenir i controlar la connexió entre navegador i servidor de VoIP.

Aquesta aplicació tindrà una base de dades relacional PostgreSQL. Utilitzarem codi PHP per a accedir-hi, ja sigui usant el client PHPpgAdmin o amb la inserció de comandes SQL a dintre del codi PHP. En aquesta base de dades s'hi guardarà la informació dels usuaris i un registre d'errors.

També es demostrarà que les aplicacions web poden tenir les mateixes funcionalitats que qualsevol aplicació nativa.

Procedir amb aquest desenvolupament permetrà familiaritzar-se amb un entorn Linux: comandes bàsiques/avançades de consola, instal·lació de noves apps a partir del codi font, gestió de xarxes, resoldre possibles problemes, etc.

A més a més, es guanyarà destresa en diferents llenguatges de programació com: HTML, JavaScript, CSS, PHP, SQL i llenguatge script intern d'Asterisk, que servirà com a complement al Pla d'estudis de la carrera.

D'aquesta manera, s'adquiriran coneixements nous, molt útils per al món laboral. L'experiència en el desenvolupament d'aplicacions (sobretot web) és un punt que les empreses destaquen a l'hora de buscar personal, ja sigui pels coneixements específics de programació o la visió global de tota l'estructura de l'aplicació.

6. Context del Projecte

El món de les aplicacions per a mòbils és un mercat a l'alça. Ja fa anys que les vendes de telèfons intel·ligents superen les vendes de tabletas, PCs i videojocs. Sembla ser que aquesta tendència continuarà i, per tant, el mercat de les apps continuarà creixent.

Tot i la gran saturació del mercat de les apps, aquest no deixa d'oferir oportunitats i nous nínxols de mercat per "atacar", ja que sempre ha sigut un mercat molt variant.

Al començament de la història de les apps, es veia Flash com a una cosa necessària, ja que, la idea de navegar per la web ens duia a pensar en la necessitat de tenir quasi la mateixa experiència que en navegador d'escriptori. Flash Lite definitivament no oferia l'experiència que necessitava un dispositiu mòbil. L'intent d'Adobe per adaptar aquest plugin en aquest mercat va ser una guerra que no va poder guanyar, almenys a la web.

Flash és un programa d'edició multimèdia distribuït per Adobe. A més a més, els fitxers Flash poden aparèixer a les pàgines web per ser vistos amb el reproductor de Adobe Flash Player.

Tot i això, i encara que flash tendeix a abandonar la web, el desenvolupament d'aplicacions natives pot ser una opció per a aquesta tecnologia (Flash ha sigut una bona eina per a desenvolupar jocs).

Hi ha hagut certa controvèrsia amb aquest programari degut a que no és de codi lliure i això pot limitar la llibertat dels usuaris.

APPS vs HTML5



Imatge 1: HTML vs Native

El món de les aplicacions per a mòbil sempre s'ha separat en dos grans grups:

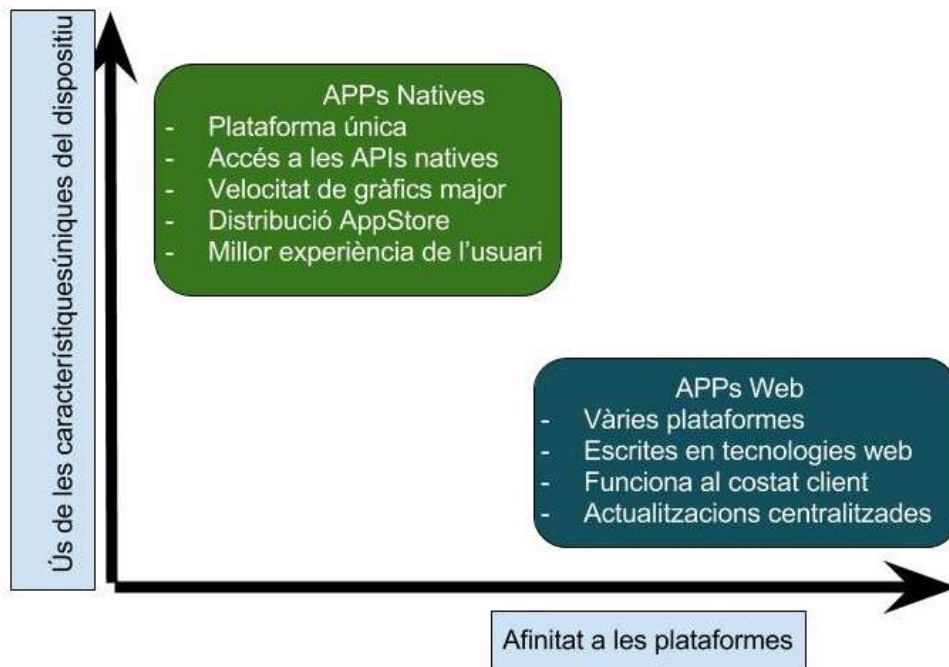
- Aplicacions natives: aquelles que estan íntegrament programades en l'entorn de desenvolupament específic per a cada sistema operatiu
- Aplicacions web: completament desenvolupades en llenguatge web.

La irrupció de HTML5, el 2008, oferia nous aires per a les aplicacions web i moltes noves possibilitats a explorar amb les seves millores respecte a HTML clàssic, sobretot les seves noves APIs:

- Per a treballar sense connexió
- De localització
- Per a reproduir vídeo i àudio amb les noves etiquetes.

Amb aquesta nova aparició es donava per començada l'anomenada “guerra de les apps”.

La falta de maduresa de HTML5, feia pensar en la mort de la web a mans de les aplicacions natives, és a dir, el final de l'ús general dels navegadors webs a canvi d'apps per a tasques específiques. Les raons: millor i més controlada experiència d'usuari, verticalitat dels continguts i major capacitat per a generar ingressos.



Imatge 2: Classificació entre aplicacions natives i web

Avui dia la revolució de les apps és indiscutible, però no totes les dades i prediccions apunten al fet que la web estigui agonitzant. Alguns experts, inclús, s'atreveixen a assegurar que estem a punt de veure el seu renaixement.

Invertir en aplicacions fa que només s'arribi a un determinat nombre del total d'usuaris potencials donat que només serveixen per a un sistema operatiu. A més a més, són més costoses i solen donar-li massa control a la plataforma a la qual estan destinades (p. Ex. Apple i el seu marge del 30% de guany de subscriptors, és a dir, Apple es queda amb el 30% del benefici de la seva tenda online).

Cadascuna d'elles té aspectes positius i negatius que influeixen a l'hora de fer una elecció. Amb una interfície d'usuari molt més amigable i personalitzada, les aplicacions natives tenen certament un avantatge a l'hora de vendre continguts.

En qüestió de disseny, les apps també tenen clar avantatge, qualsevol pot dissenyar, enriquir i exportar versions del seu contingut a les tendes en línia d'apps (Play-store, AppleStore, etc.).

També cal destacar que el principal impulsor de les apps, la revolució mòbil, seguirà en alça els pròxims anys i, com repercussió, continuaran creixent les possibilitats dels sistemes tancats d'aplicacions.

En canvi, una nova tendència de desenvolupament d'apps a dintre del navegador està prenent el control. De la mà de HTML5 i JavaScript, diverses funcionalitats específiques per a dispositius tableta o telèfons intel·ligents poden ser implementades (geo-localització, moviment, etc.) i utilitzades en qualsevol dispositiu portàtil, eliminant el problema de fer una aplicació diferent per a cada plataforma.

Un dels punts forts de les aplicacions web és el seu funcionament en totes les plataformes (només s'ha de tenir en compte la compatibilitat amb el navegador web). Una aplicació web funcionarà a tots els terminals mentre que les natives necessiten un desenvolupament diferent per a cadascun dels sistemes operatius (inclús per a versions diferents d'aquests). Donat que el seu funcionament es basa en recursos web, necessiten una bona capacitat de concurrència en els servidors on s'alberguin i una bona connectivitat per a funcionar de forma òptima.

Òbviament, això és també un dels seus punts crítics: algunes d'aquestes aplicacions no funcionen sense connexió, inclús si a priori sembla que no la necessiten. Altrament, el gran avantatge està en el fet que no són necessaris uns grans requisits de hardware en la part del dispositiu mòbil per a executar les aplicacions.

Si passem a analitzar la integració de les aplicacions amb les funcionalitats que ofereixen els mòbils, ens trobem amb que els llenguatges web no tenen encara compatibilitat amb totes les funcionalitats de les APIs natives i això fa la seva integració complexa, normalment es desenvolupen frameworks per utilitzar-les. Un framework inclou programari de suport, llibreries de codi i llenguatges de programació que poden ajudar a organitzar, desenvolupar i a integrar components diversos.

Com ja s'ha comentat anteriorment, la falta de maduresa de HTML5 pot jugar en la seva contra però, per altra banda, podem considerar l'hipertext com el gran bastió de poder de la web ja que ens ofereix un munt de possibilitats no explorades.

El gran avantatge de la web és permetre que un sense fi de llocs/usuaris es connectin els uns amb els altres a través d'enllaços, permetent el tràfic orgànic entre un i altre. Les aplicacions, per la seva estructura tancada, no permeten això i la seva capacitat d'interconnectar-se és limitada, inclús algunes apps, han d'obrir un navegador quan l'usuari clica en algun enllaç.

Davant la gran quantitat d'informació que es genera cada dia, l'hipertext segueix sent la millor arma de la web (fins i tot Google qualifica els continguts depenent dels enllaços entrants i sortints d'un lloc web).

Per tant, tot i que és difícil saber quina de les dues opcions és la millor quan es vol iniciar un desenvolupament d'una nova aplicació per a telèfons intel·ligents, s'ha escollit HTML5 pel ventall de noves possibilitats que aporta al món de les aplicacions per a mòbils. A continuació es fa un anàlisi més exhaustiu de les característiques d'ambdues solucions.

HTML5 o APPs?

Avui dia hi ha diferents factors a considerar a l'hora d'escollir entre desenvolupar web apps o apps natives: complexitat de desenvolupament, interfície de l'usuari, canal de distribució, etc.

Les aplicacions web requereixen una menor inversió inicial, pel fet que la major part del desenvolupament no s'ha de repetir en cada sistema operatiu. És important ressaltar que, quan es decideix fer múltiples versions d'una aplicació, no només s'han de desenvolupar, sinó que també s'han de testejar i mantenir. Aquest fet podria arribar a disparar els costos.

Un altre avantatge de les aplicacions web és que solen necessitar un menor temps de desenvolupament, el que pot ser crucial en el cas que existeixi una finestra d'oportunitat en el mercat.

Un aspecte a tenir en compte és la consistència de l'aplicació amb la interfície d'usuari de cada sistema operatiu. Si es requereix una aplicació única per a tots els sistemes operatius, el que es guanya en homogeneïtat es perd en personalització proporcionada des de la interfície nativa. S'ha de valorar si compensa tenir la mateixa aplicació web en un mòbil amb diferent sistema operatiu a costa de què aquesta aplicació suposi una ruptura amb l'estil de la interfície nativa del dispositiu i es perdin les opcions específiques de cada plataforma.

A més a més de la perspectiva del desenvolupament, hi ha una altra premissa important: la visió dels usuaris. Encara que no es tracti d'un aspecte tècnic, també és molt important considerar com és el canal de distribució per als diferents tipus d'aplicacions, és a dir, la tenda. Les aplicacions web són les més perjudicades en aquest aspecte perquè la seva presència en les botigues oficials dels sistemes operatius majoritaris no està admesa. Això genera la necessitat de major esforç de màrqueting per a aconseguir visibilitat.

Les enquestes revelen que l'experiència mòbil té un alt impacte en els clients, en com veuen una marca i com interactuen amb ella. La majoria de clients prefereixen les aplicacions natives.

	WEB	NATIVES
AVANTATGES	Opció ràpida	Millor experiència d'usuari
	Només necessita un desenvolupament	Temps de càrrega millor
	Portabilitat	Robustes, online i offline
		Millor seguretat
		Aprofitament del hardware

Taula 1: Avantatges aplicacions

	WEB	NATIVES
DESAVANTATGES	Experiència d'usuari pitjor	Varis desenvolupaments
	Velocitat de resposta inferior	Habilitats cares al món laboral

Taula 2: Desavantatges aplicacions

En resum, si hem d'escollir entre les dues opcions cal tenir ben clar els avantatges i els desavantatges. En aquest projecte s'ha escollit desenvolupar una aplicació web per el seu potencial i per la possibilitat d'integrar més fàcilment diferents softwares que estan en fase de creixement. [1][2][3][4][5][6][7][8][9][10][11][12]

7. Conceptes

VoIP

La VoIP és un conjunt de recursos que fan possible que la senyal de veu viatgi per una xarxa utilitzant el protocol IP (la senyal de veu s'empaqueta en paquets de dades). És més eficient que el servei tradicional de telefonia.

El protocol SIP s'utilitza per a la iniciació, modificació y finalització de sessions interactives d'usuari on intervenen elements multimèdia com el vídeo, veu, missatgeria instantània, etc. Va ser pensat per a que la telefonia es converteixi en un servei més d'internet. Aquest, permet l'establiment de sessions entre dos o més usuaris. És el protocol utilitzat en el desenvolupament del projecte.

Els agents són els usuaris, que poden ser humans o aplicacions software. Per a establir sessions utilitzen el que el protocol SIP denomina "Agents d'usuari". Aquests no són més que els punts extrems del protocol, és a dir, són els que emeten i consumeixen missatges. El protocol no es preocupa de la interfície d'aquests dispositius amb l'usuari final, només s'interessa pels missatges que aquests generen i com es comporten en rebre determinats missatges.

Els agents es comporten com a clients quan realitzen peticions (UAC: *User Agent Clients*) o com a servidors quan les reben (UAS: *User Agent Servers*).

El protocol SIP permet establir la ubicació d'un determinat usuari. Cada usuari té una direcció lògica que és invariable respecte a la ubicació física de l'usuari. Una direcció lògica del protocol és de la forma usuari@domini, és a dir, té la mateixa forma que una direcció de correu electrònic. La direcció física, depèn del lloc on l'usuari estigui connectat (direcció IP).

Funcionament del protocol SIP:

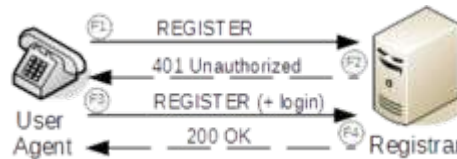
Quan un usuari, per exemple, inicialitza el seu terminal, l'agent usuari SIP que resideix en aquest terminal envia una petició amb el mètode REGISTER a un Servidor, informant de a quina direcció física ha d'associar-se la direcció lògica de l'usuari. El servidor, doncs, realitza un registre d'aquesta associació. Aquesta, té un període de vigència i, si no es renova, caduca. La comunicació també pot acabar-se mitjançant un des-registre.

Els missatges que s'intercanvien poden ser peticions o respostes. Les peticions i les respostes tenen una línia de petició, una sèrie d'encapçalats i un cos. A la línia de petició s'indica el propòsit de la petició i el destinatari.

Les peticions tenen diferents funcions. El propòsit està determinat pel que es denomina Mètode d'aquesta petició. Aquest no és més que un identificador de propòsit de la petició. A la línia de resposta s'indica el codi d'estat de la resposta, que és un número que indica el resultat del processament de la petició.

Els encapçalats de les peticions i respostes s'utilitzen per a diverses funcions relacionades amb l'encaminament de missatges, autenticació, etc.

El cos dels missatges és opcional i s'utilitza, entre d'altres coses, per a transportar les descripcions de les sessions que es volen establir. [13][14]



Imatge 3: Funcionament protocol SIP

El servidor Asterisk, gestiona connexions SIP amb els seus usuaris.

getUserMedia()

Com ja s'ha explicat a la secció “*Introducció i objectius*”, es va voler realitzar una aplicació web per explorar les possibilitats que ofereix HTML5.

La captura d'àudio i vídeo ha sigut una de les grans aspiracions del desenvolupament web durant molt de temps. Durant molts anys, s'ha tingut dependència dels complements dels navegadors.

Durant els últims anys, hi ha hagut una evolució en les APIs de captura multimèdia. Aquesta va ser tant ràpida que la comunitat internacional W3SC va decidir formar un grup de treball.

Per la captura multimèdia HTML aquest va ser el primer intent:

```
<input type="file" accept="image/*;capture=camera">  
<input type="file" accept="video/*;capture=camcorder">  
<input type="file" accept="audio/*;capture=microphone">
```

Així es reaprofitava la etiqueta input (utilitzada en entrada d'arxius). Per altra banda, es quedava curt a l'hora d'aplicar efectes o filtres a temps real. Només està permès gravar un arxiu multimèdia o una prendre una instantània a temps real.

Molts van pensar que la captura multimèdia estava massa limitada i es va inventar una nova etiqueta, <device>.

Poc després, aquesta etiqueta es va eliminar a favor d'una funció javascript anomenada navigator.getUserMedia().

getUserMedia() esta relacionat amb WebRTC perquè és la porta d'entrada a aquesta API. Proporciona la forma d'accedir al flux de dades de la càmera i el micròfon de l'usuari sense haver d'instal·lar res. Les dades es processen i s'envien directament al navegador.

Així seria un exemple bàsic de l'ús de navigator.getUserMedia():

```
<video autoplay></video>

<script>

  var onFailSoHard = function(e) {

    console.log('Reeeejected!', e);

  };

  // Not showing vendor prefixes.

  navigator.getUserMedia('video, audio', function(localMediaStream) {

    var video = document.querySelector('video');

    video.src = window.URL.createObjectURL(localMediaStream);

    video.onloadedmetadata = function(e) {

      // Aquí es pot treballar amb el element video.

    };

  }, onFailSoHard);

</script>
```

Per utilitzar la càmera web el micròfon, necessitem sol·licitar permís al navegador i que aquesta pagina estigui sota https. El primer paràmetre `getUserMedia()` és per especificar el tipus de medi al que s'hi vol accedir: àudio o vídeo. Per exemple, si es vol accedir a la càmera, hauria de ser vídeo. Si es vol accedir simultàniament a la càmera i al micròfon, s'haurà d'utilitzar vídeo, àudio (com es veu en el exemple anterior).

La captura multimèdia es un exemple perfecte de les noves API HTML5. S'ha de tenir en compte que no hi ha cap atribut "src" dins de l'etiqueta vídeo. Enlloc de donar al vídeo una direcció, se li dona una URL Blob obtinguda a partir de un objecte `LocalMediaStream` que representa la càmera (o el micròfon en el cas d'àudio).

En resum, amb aquestes instruccions, podem obtenir de manera fàcil i ràpida informació multimèdia per ser tractada amb els nostres objectius. En el nostre cas, seran dos: obtenir vídeo o captures de la càmera del dispositiu per processar-los reconèixer i llegir el codi QR i reproduir els àudios que el servidor Asterisk envii. [19]

WebRTC



Web Real-Time Communication és una API que està sent elaborada per la World Wide Web Consortium (W3C) per a permetre a les aplicacions del navegador realitzar trucades de veu, xat de vídeo i ús compartit de P2P sense plugins. En poques paraules, és un intermediari de programari que fa possible que els programes d'aplicació interactuïn entre si i comparteixin dades.

WebRTC

WebRTC és una tecnologia emergent que s'accedeix amb les API de JavaScript. Actualment està en desenvolupament.

WebRTC fa que sigui factible per als desenvolupadors web habilitar VoIP en les seves aplicacions basades en web.

Des de que WebRTC es trobava en les seves primeres etapes de desenvolupament, que no inclou cap protocol de senyalització. Deixa aquesta elecció, el desenvolupament i la integració, al desenvolupador.

Mitjançant la integració d'un protocol de senyalització en WebRTC , un desenvolupador pot crear un client de programari complet de VoIP en un navegador.



Imatge 4: Comunicació navegador web i Asterisk

Els principals components inclouen:

- **getUserMedia**, que permet a un navegador web accedir a la càmera o el micròfon
- **PeerConnection**, que estableix trucades d'àudio/vídeo
- **DataChannels**, que permeten als navegadors a compartir dades a través de peer-to-peer

Hi ha una mica de confusió en la indústria de les telecomunicacions pel que fa a si WebRTC va en contra de VoIP. WebRTC és un sistema de comunicacions de Protocol d'Internet viable (IP) funciona en paral·lel del VoIP. Serveis de VoIP i solucions de WebRTC tenen una forta promoció en els camps comercials. Pel que la confusió que s'estableix aquí és: són els proveïdors de VoIP i WebRTC amics o enemics? Poden coexistir els dos sistemes? Com es beneficia el client d'aquests?

- **Les similituds**

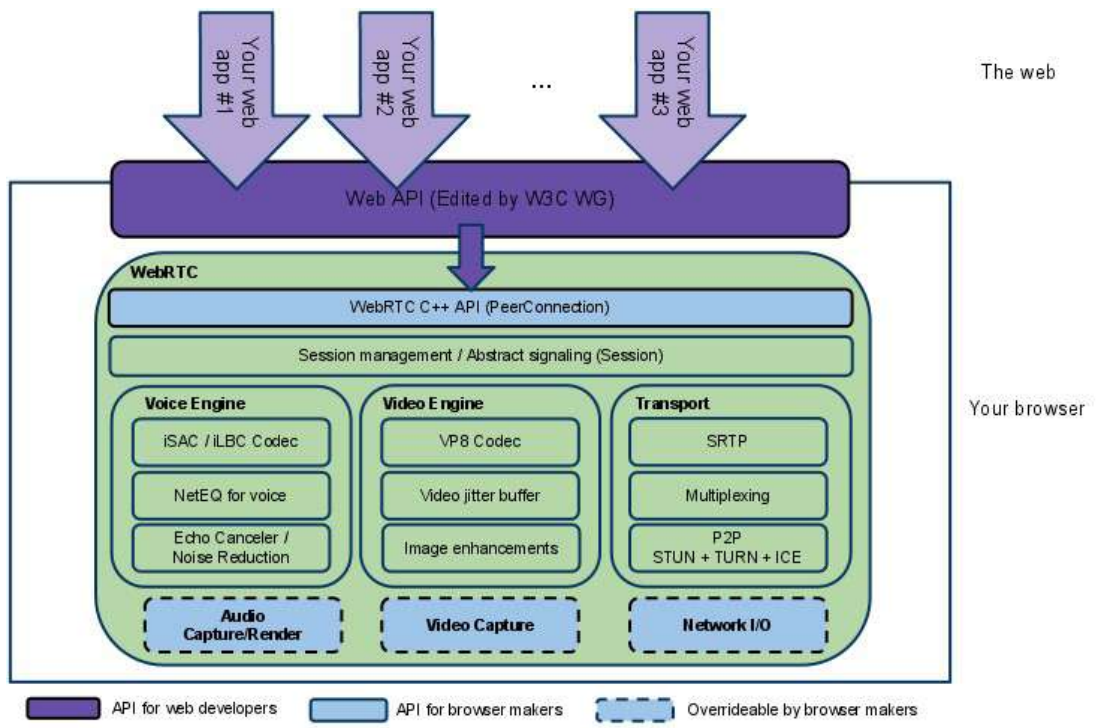
- WebRTC i VoIP són similars en que tant un com l'altre tenen l'objectiu de millorar l'experiència de l'usuari i permetre a qualsevol dispositiu de consum (ja sigui telèfon mòbil, fax, Internet, etc.) connectar sense esforç des de qualsevol lloc i en qualsevol xarxa internacional.

- **Les diferències**

- La principal diferència entre els dos serveis és que VoIP utilitza una multitud de variants com ara VoIP a través d'un mòdem DSL / cable , veu sobre Wi - Fi / 3G (vovifi / 3G) , veu sobre LTE (Volte) , i Rich Communication Suite (RCS) , mentre que WebRTC es centra exclusivament en les comunicacions basades en el navegador.

En aquesta aplicació, VoIP i WebRTC coexisteixen. El servidor Asterisk envia els àudios encriptats (VoIP) i WebRTC extreu i reproduïx la informació en el navegador de l'usuari.

L'arquitectura global de la API té aquesta forma:



Imatge 5: Esquema WebRTC API

RTCWeb Breaker

Les especificacions de WebRTC són compatibles amb la majoria de clients SIP. El problema és que molts endpoints SIP - Legacy (per exemple, la xarxa PSTN) no suporten aquestes característiques. És feina del RTCWeb Breaker negociar i convertir el flux dels mitjans de comunicació per a permetre que aquests dos mons interactuïn.

Per exemple, si el servidor no és compatible amb l'ICE això vol dir que es requereix RTCWeb Breaker per tal de poder connectar el navegador a un punt final SIP - legacy.

Per aquesta raó, dins de l'API de WebRTC, hi ha la opció d'utilitzar aquesta funcionalitat.



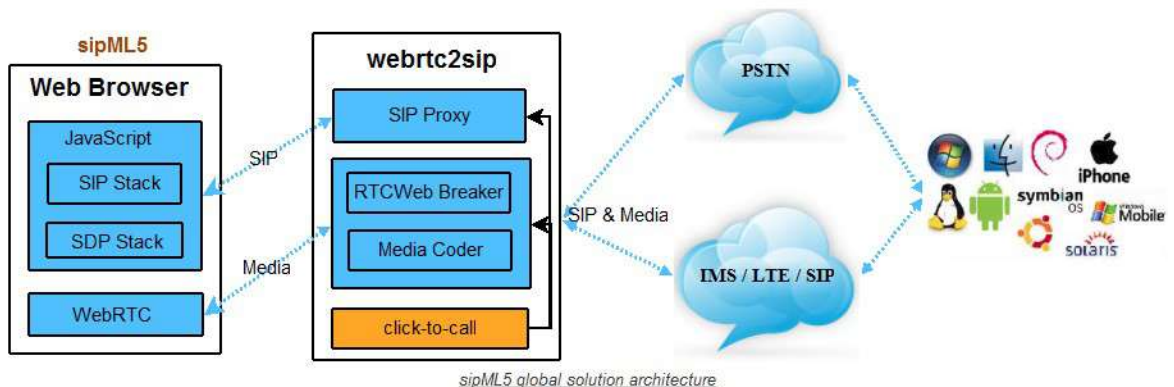
Imatge 6: Esquema RTCWebBreaker

SIP - legacy és una empresa que proveeix serveis de VoIP. Per exemple crear una xarxa de telefonia interna a dins d'una companyia. Encara que en aquest SIP – legacy es refereix a les tecnologies antigues que usen SIP. [13][15]

Simpl5



Aquest és el primer client SIP HTML5 de codi obert del món íntegrament escrit en JavaScript per a la integració en les xarxes socials (Facebook, Twitter, Google+) , jocs en línia, llocs web de comerç electrònic , signatures de correu electrònic ... No és necessària cap extensió, connector o porta d'enllaç. El media stack es basa en WebRTC . El client es pot utilitzar per a connectar-se a qualsevol xarxa IMS o SIP des del seu navegador preferit per a realitzar i rebre trucades d'àudio / vídeo i missatges instantanis.



Imatge 7: Esquema complet Sipml5

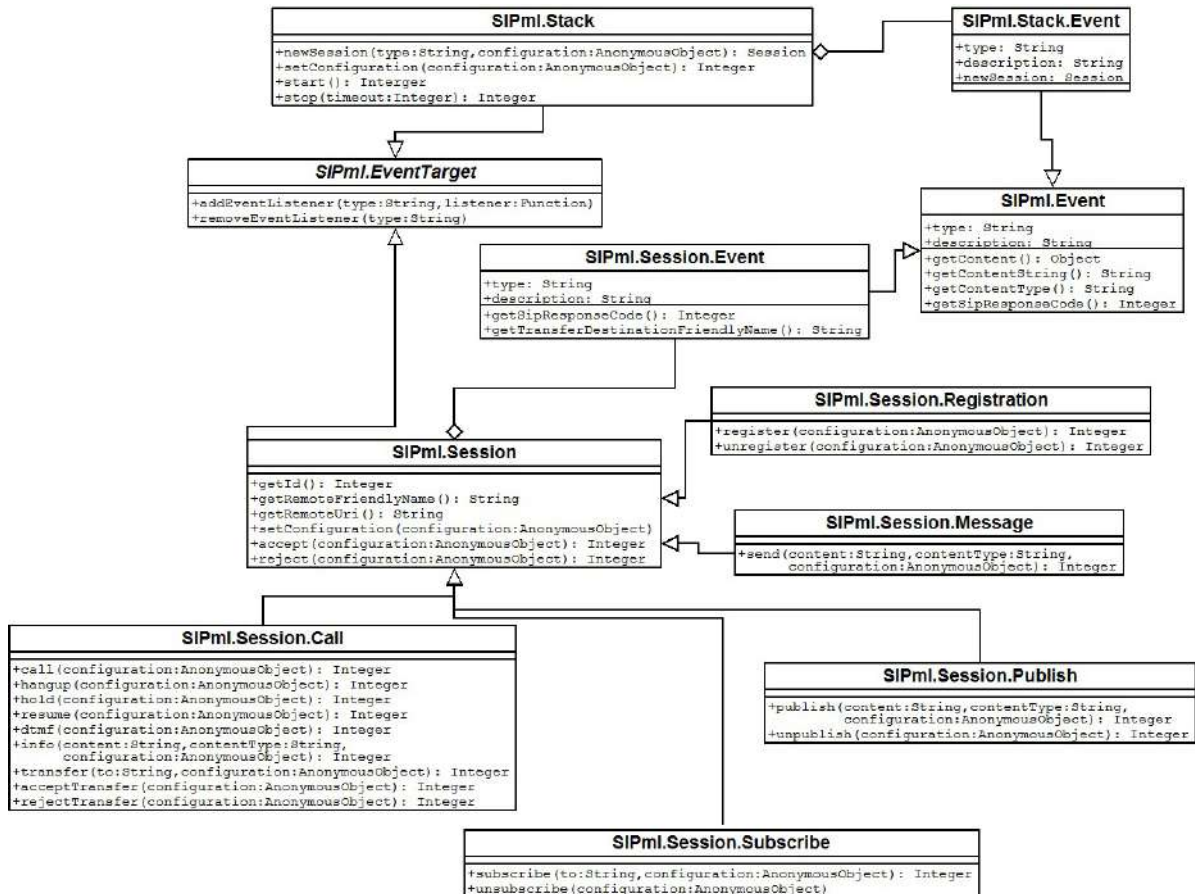
El projecte encara està en desenvolupament, però ofereix moltes possibilitats que fins ara semblaven impossibles. Per exemple: el xat de Google (HangOuts), tant de veu com de vídeo, necessita de la instal·lació d'un plugin al navegador per a funcionar. Aquests plugins, normalment no funcionen als dispositius mòbils. Amb aquesta tecnologia, s'ha desenvolupat un xat de veu i vídeo, que funciona a qualsevol terminal (amb un navegador compatible) i sense haver d'instal·lar cap software extra.

En l'aplicació desenvolupada en aquest projecte, s'han utilitzat les funcions del client SIP i les funcionalitats de WebRTC, que estan influïdes a dintre d'aquest software. [16]

8. Arquitectura de l'aplicació

Connexió i comunicació amb Asterisk

L'aplicació utilitza el model de classes JavaScript de WebRTC:



Les classes més importants són:

- Stack: “pila” de xarxa per “Real Time Protocol”.
- Session: una capa abstracta que permet la configuració de trucades i la capa d’administrador.

En el codi de l’aplicació, primer s’ha de configurar stack, per passar-li com a paràmetre per iniciar sessió. En aquest punt, i utilitzant websockets i el protocol SIP, s’inicia, si la validació és correcte, una sessió amb el servidor Asterisk.

```

oSipStack = new SIPml.Stack({
    realm: ...,
    impi: ...,
    impu: ...,
    password: ...,
    display_name: ...,
    websocket_proxy_url: ...,
    outbound_proxy_url: ...,
    ice_servers:...,
    enable_rtcweb_breaker:...,
    events_listener: { events: '*', listener: onSipEventStack },
    enable_early_ims:...,
    enable_media_stream_cache:...,
    bandwidth:...,
    video_size:...,
    sip_headers: [
        { name: 'User-Agent', value: 'IM-client/OMA1.0 sipML5-v1.2016.03.04' },
        { name: 'Organization', value: 'Doubango Telecom' }
    ]
})

oSipStack.start();

```

Com es pot observar, el stack porta tota la informació que un usuari SIP necessita per connectar-se al servidor.

Amb la última comana s'inicia la connexió amb el servidor Asterisk.

Per realitzar una trucada, el funcionament és similar. S'ha de crear un objecte de configuració oConfigCall i passar-lo com a paràmetre.

```

oConfigCall = {
    audio_remote: audioRemote,
    screencast_window_id: 0x00000000, // entire desktop
    bandwidth: { audio: undefined, video: undefined },
    video_size: { minWidth: undefined, minHeight: undefined, maxWidth:
undefined, maxHeight: undefined },
    events_listener: { events: '*', listener: onSipEventSession },
    sip_caps: [
        { name: '+g.oma.sip-im' },
        { name: 'language', value: '\"en,fr\"' }
    ]
};
oSipSessionCall = oSipStack.newSession('call-audio', oConfigCall);
oSipSessionCall.call(txtPhoneNumber.value)

```

Una vegada configurat ja es pot iniciar la trucada. En aquest cas, s'envia el nombre de telèfon que correspon a la extensió d'Asterisk que volem que s'executi.

En el nostre cas, aquesta extensió serà molt senzilla ja que només consta en despenjar, enviar l'àudio i penjar.

Aquí es pot veure un exemple d'una extensió d'Asterisk configurada en el fitxer `extensions.conf` d'Asterisk:

```
exten => 55555,1,Answer()  
same => n,Playback(guernica)  
same => n,Hangup()
```

Aquesta extensió correspon al número 55555. Aquest nombre, serà proveït per la lectura del codi QR.

Lectura de QR

La lectura QR funciona de la següent manera: Utilitzant la càmera, es realitzen varies captures per segons. Cadascuna d'elles, és escanejada per l'algoritme capaç de reconèixer el codi QR.

Aprofitant la nova funcionalitat de HTML5, el navegador pren el control de la càmera del dispositiu mòbil. Les captures, d'aquesta són guardades en un objecte "CANVAS" de HTML.

Aquest objecte és analitzat per un procés de diferents funcions en JavaScript que extreuen la informació guardada en el codi.

Quan aquesta informació és llegida, es guarda en una variable i s'envia una excepció. El valor de la variable és enviat a Asterisk com una extensió marcada. Asterisk executarà l'ordre que aquesta extensió tingui configurada en el "dialplan".

Errors

S'ha creat un sistema de maneig d'excepcions i registre d'errors. S'ha utilitzat la possibilitat de connectar-se a la base de dades PostgreSQL fent servir codi php o javascript.

A la taula següent hi ha la relació d'errors i el seu codi:

Codi	Error
10	Error al connectar-se amb la base de dades
20	Usuari existent
30	Error al crear stack
40	Error al connectar-se a Asterisk
50	Extensió Asterisk desconeguda
60	Error al rebre/reproduir àudio

Aquests codis han sigut de molta utilitat durant el desenvolupament ja que gracies a ells ha sigut fàcil depurar i corregir els errors que han anat sorgint durant aquest procés.

9. Funcionament de l'aplicació

Sessions i usuaris

En aquesta secció s'explicaran les diferents planes web de l'aplicació. Es reproduirà un flux d'un usuari a través dels diferents passos que haurà de seguir per a utilitzar-la. Es mostraran també, les seccions de notícies i informació general.

L'aplicació s'ha dissenyat de manera que sigui bàsica i senzilla d'entendre.

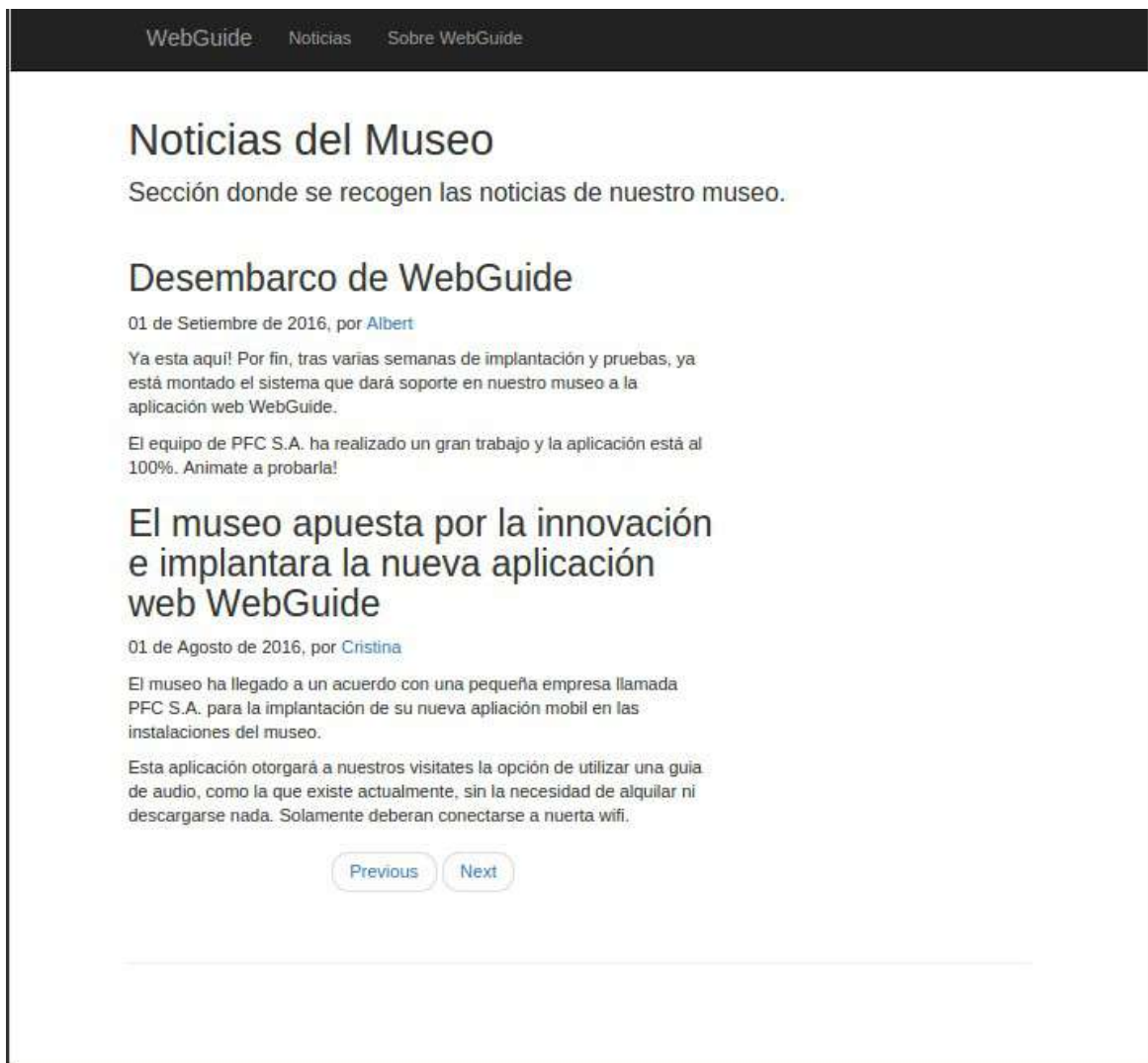
Imaginem que un usuari es connecta a la xarxa utilitzant el seu terminal. En obrir el navegador, per defecte apareixerà la pàgina “home” de l'aplicació.



Imatge 8: Pàgina principal de l'aplicació

Com es pot veure a la imatge, la pàgina consta de tres grans seccions: “Prova gratis”, “Iniciar sessió/crea’t compte” i la secció de notícies.

Aquesta última, contindrà els titulars de les últimes notícies del centre i els enllaços a les notícies completes.



Imatge 9: Secció notícies

Si es prem el botó de “Prova gratis” es tindrà un accés parcial al contingut de l’aplicació. En canvi si et registres o iniciés sessió amb un compte ja registrat, es mostrarà una plana on s’informa de a quin tipus de compte s’ha accedit i possibles millores de tipus de compte (Bàsic -> Premium).

Els comptes “prova gratis” són d’un sol ús, eliminats quan l’usuari es desconnecta de l’aplicació. Per altra banda, els comptes Bàsic o Premium s’allotgen a la base de dades encriptats, com marquen els requisits legals.



Imatge 10: Pàgina de benvinguda

Tant si l'usuari és gratuït o de pagament, es mostrarà una pàgina d'inici de lectura.



Imatge 11: Pàgina d'inici d'escaneig

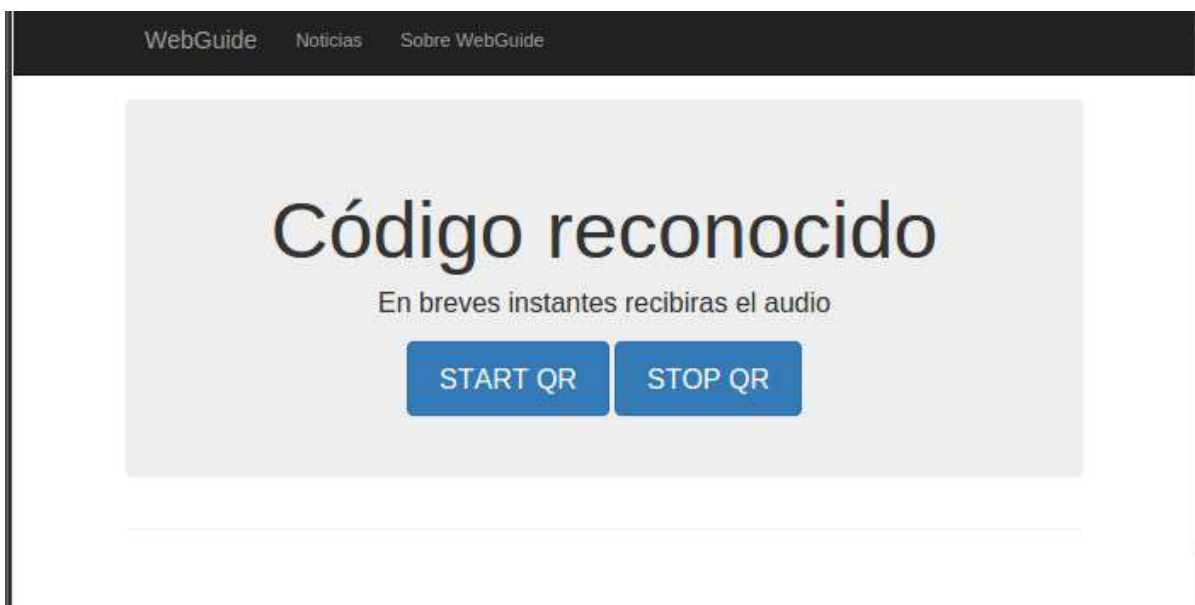
En prémer el boto "START QR" i donar permís al navegador per a accedir a la càmera del terminal, apareix a la pantalla la visió de la càmera. L'usuari ha d'apropar el mòbil al codi QR situat al costat del quadre/obra d'art de la qual en vol saber informació.

El desplegable superior et deixa escollir la càmera que es vol utilitzar.



Imatge 12: Pàgina d'escaneig

Un cop el codi QR ha estat llegit, apareix la plana web:



Imatge 13: Pàgina d'èxit en el escaneig

El valor d'aquest codi serà enviat al servidor Asterisk. Aquest enviarà l'àudio que explica l'obra.

En acabar la transmissió apareix una pàgina on es pregunta si es desitja més informació o per altra banda es vol escanejar un nou codi QR.



Imatge 14: Pàgina final

Si l'usuari desitja saber més d'aquesta obra, s'envia una petició a Asterisk amb una altra extensió que reproduceix més informació d'aquesta. Per altra banda, si es clica el botó "Un altre QR", la plana anterior serà carregada.

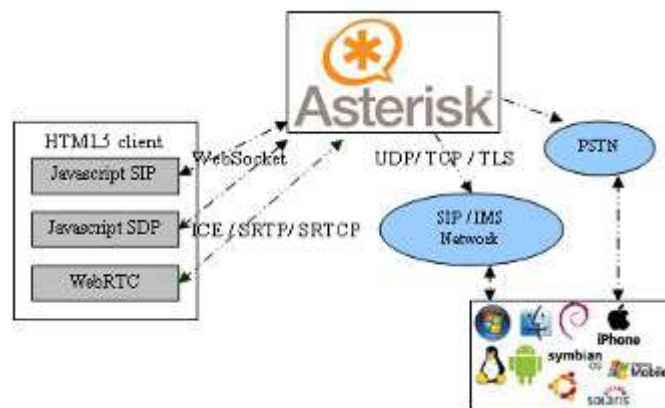
Per a tancar l'aplicació i desconnectar-se d'Asterisk, l'usuari només haurà de tancar el navegador o desconnectar-se de la xarxa Wifi del museu.

Protocol SIP (usuaris)

Un cop s'ha explicat l'aplicació des de la visió de l'usuari, s'explicarà la visió tècnica, a alt nivell, del funcionament d'aquesta.

En el moment que l'usuari inicia sessió (ja sigui usuari gratuït o no), el terminal es connecta a Asterisk. Aquest servidor té una base de dades dinàmica PostgreSQL on s'emmagatzema la informació dels diferents usuaris, no importa el tipus.

Per una banda, el codi PHP gestiona la sessió al navegador i per l'altra, utilitzant WebRTC, s'envia aquesta informació a Asterisk, que la compara amb la seva base de dades. El protocol SIP és molt adient en aquest cas, degut a que no sobrecarrega la xarxa. Crea la relació entre usuari i servidor, lliurant al terminal una direcció lògica assignada a la seva IP.



Imatge 15: Esquema aplicació

Els diferents tipus d'usuaris estaran emmagatzemats en diferents taules i tindran diferents permisos dins d'Asterisk, és a dir, Asterisk gestionarà els permisos dels usuaris per a poder rebre, o no, els diferents àudios.

Asterisk pot tenir configurats diferents tipus d'usuari amb diferents permisos per marcar diferents extensions. Les extensions són les "ordres" configurades en el "dialplan". Aquestes poden ser despenjar, mantenir-te en espera, reproduir un àudio, connectar-te amb un altre usuari (prèviament connectat a Asterisk), penjar, etc. N'hi ha més de trenta.

En aquest projecte, només s'han utilitzat les extensions despenjar, reproduir àudio i penjar, tot i que en un futur es pot implementar el servei de valorar el servei (com, per exemple, es fa en les enquestes telefòniques).

En el moment en que l'usuari es desconnecti de la xarxa, es tancarà la sessió tant a PHP com a Asterisk. Si l'usuari no és gratuït, la informació es mantindrà a la base de dades per a futures connexions. Aquesta informació podria ser compartida en una xarxa de museus on,

per exemple, si l'usuari visités un altre museu on el mateix sistema estigués implantat, aquest client pogués utilitzar el compte creat al museu anterior. [18][19]

10. Execució i problemes

El desenvolupament de l'aplicació no ha sigut fàcil degut a que moltes de les APIs utilitzades es troben encara en desenvolupament. Hi ha moltes funcionalitats encara no explorades de HTML5 que ofereixen moltes possibilitats i diferents punts de vista a l'hora de desenvolupar una aplicació web.

Tot i així, els diferents problemes que han anat sorgint durant el desenvolupament s'han solucionat gràcies a tutorials i a les respostes ràpides en els fòrums per part dels desenvolupadors dels projectes i a la informació del codi de les aplicacions.

Primers passos

Un cop es va tenir la idea de l'aplicació, el primer pas va ser familiaritzar-me amb Asterisk. Per aquell temps, el repositori d'aquest encara no estava configurat dintre de Linux, per tant, s'havia d'instal·lar des del codi font. Aquí van aparèixer els primers entrebancs. Vaig aprendre a configurar, compilar, depurar, construir i instal·lar programari a Ubuntu.

També es va haver d'aprendre a compilar, debugar i configurar programes a Linux.

Un cop instal·lat, vaig estar un temps aprenent a configurar Asterisk de diferents formes: creant diferents “dialplans” per a diferents usuaris.

Es va començar per la configuració bàsica, és a dir, dos usuaris utilitzant Sipphones (telèfons de VoIP que utilitzen el protocol SIP) que intercanvien àudio. Aquests usuaris estan configurats estàticament.

A més a més, per suposat, es va provar també, que el servidor Asterisk enviés àudios prèviament gravats als usuaris.

Client web

Més tard, vaig buscar un client web per a Asterisk. La cerca no va ser fàcil donat que HTML5 encara estava i està en un grau de maduresa molt baix. Un cop trobat el codi de Sipml5, vaig haver d'aprendre HTML i JavaScript.

L'aprenentatge de HTML bàsic va ser ràpid, no així amb JavaScript. Aquest últim és un llenguatge en el que es dona molta llibertat al desenvolupador i això fa que a vegades sigui difícil de llegir sinó es té la experiència necessària o s'utilitza algun framework.

Vaig crear una pàgina des de zero. D'aquesta forma vaig anar guanyant coneixement i habilitat a l'hora de dissenyar planes. Quan l'estructura estava ja clara, va ser el moment d'inserir les funcions JavaScript per fer les pàgines dinàmiques.

Per sort, el projecte Sipml5 està molt ben documentat no va ser difícil adaptar el seu codi a les meves necessitats. Tot i així, va ser necessari un gran esforç i vàries consultes als fòrums. Vaig haver de reportar un parell de errors al fòrum oficial per aconseguir que la recepció i reproducció d'àudios per part dels navegadors funcionés a la perfecció.

Quan vaig aconseguir connectar un navegador amb Asterisk i que el primer reproduís els àudios enviats pel segon, faltava configurar la seguretat de la aplicació.

HTTPS - WSS

Les versions més noves dels navegadors web, neguen la connexió utilitzant websockets o l'ús de la càmera o el micròfon dels dispositius si aquesta no es fa utilitzant una connexió segura https://.

Es van haver de crear certificats d'autenticació i configurar el servidor apache perquè només oferís la possibilitat de connexions https://.

També es van haver de crear certificats digitals i configurar els clients sip (les connexions dels usuaris) perquè utilitzessin websockets segurs (ws enlloc de wss).

D'aquesta manera s'assegura l'enciptació de les dades que circulen entre el servidor web o el servidor Asterisk i el navegador del nostre terminal.

Lectura QR

Ara faltava el lector de codis QR. Aquest software va ser fàcil de trobar i d'adaptar per al projecte.

Els problemes en aquesta part van sorgir en la comprensió del codi, degut a que no hi havia cap tipus de documentació. Van ser necessàries moltes proves i equivocar-se moltes vegades per comprendre el codi i adaptar-lo a els requisits de l'aplicació.

Tot i així, aquest pas no va ser tant problemàtic com els passos anteriors degut a la experiència guanyada en HTML i JavaScript realitzant les primeres fases.

Usuaris (PHP)

Com calia una gestió de sessions i usuaris es va optar per utilitzar PHP per a la seva flexibilitat i facilitat d'aprenentatge. Es va haver de reescriure el codi varies vegades perquè funcionés correctament sense errors, és a dir, es va haver d'introduir el codi PHP entre les línies del codi HTML. Això va ser un nou repte: aprendre un nou llenguatge de programació.

El codi PHP, igual que HTML i JavaScript, conta amb molts tutorial a internet, però la barreja de PHP i HTML, que pot semblar senzilla, al principi no es trivial d'entendre. Van ser necessàries moltes iteracions i molts errors per tenir completament integrats els codis dels dos llenguatges.

A més a més, es va utilitzar per registrar els possibles errors de l'alicació a la base de dades i d'aquesta manera portar-ne un control, identificar-ne quins son els mes freqüents i corregir-los el més aviat i fàcil possible.

Usuaris (SQL)

A més a més, es va haver de configurar una base de dades SQL. Aquest fet va obligar a aprendre comanes SQL i l'arquitectura i estructura d'una base de dades. No hi va haver cap dificultat remarcable en aquest pas, ja que SQL és un estàndard intuïtiu i de fàcil aprenentatge.

Per altra banda, la connexió entre Asterisk i la base de dades va ser molt problemàtica. Es va haver de reinstalar Asterisk amb un nou mòdul que permet la connexió entre aquests dos elements. Aquest canvi produeix que la configuració d'Asterisk deixi de ser estàtica i canviï cap a dinàmica.

Molts errors i problemes van sorgir. Es van anar solucionant accedint a diferents fòrums on es donaven la solució a altres usuaris que tenien els mateixos problemes.

Aparença pàgina web

Per acabar, amb l'aplicació completament funcional, es va haver d'abordar el disseny visual d'aquesta. Per una persona tècnica sense gaires coneixements de disseny, ha sigut un repte aprendre CSS. Tot i així, gràcies a que Bootstrap té un gran avantatge en el disseny adaptat per terminals mòbils no han sorgit grans problemes a remarcar.

Aquest framework, dona moltes facilitats i plantilles per a desenvolupadors novells de planes webs i permet que amb pocs coneixements, es pugui crear un disseny estètic i funcional.

Testeig i procés de qualitat

Durant l'execució del projecte no s'ha pogut dur a terme una estratègia de testeig estàndard degut a que l'aprenentatge dels codis i dels softwares anava en paral·lel amb el desenvolupament d'aquest. Tot i així s'han anat realitzant proves durant l'escriptura del codi per a comprovar que l'aplicació compleix amb els requisits definits.

11. Conclusions

Com ja s'ha comentat anteriorment, el desenvolupament d'aquesta aplicació no sigut fàcil degut a l'etapa en la que es troba el desenvolupament de HTML5 i a la multitud de llenguatges que hi conviuen. Tot i que al principi era una mica aclaparador, el fet de que els progressos durant l'evolució de la aplicació siguin fàcilment visibles, fa més fàcil l'aprenentatge.

Per altra banda, m'agradaria remarcar que aquests coneixements adquirits em són molt útils en el meu dia a dia laboral, no només per a la comprensió de llenguatges de programació, amb els que hi treballa diàriament, sinó també en la visió global d'una aplicació: totes les seves parts, dependències entre elles i problemes d'integració.

Després finalitzar aquest projecte i d'analitzar les avantatges i desavantatges de HTML5 i APP, no es pot afirmar que les aplicacions web siguin millor que les natives ni viceversa. S'haurà de fer un estudi de mercat abans de decidir-se per una. Les aplicacions HTML5, però, tenen un futur més esperançador i ofereixen moltes més possibilitats degut a la seva facilitat per a adaptar-se a noves tecnologies i la seva fàcil connexió amb altres softwares.

Possibles millores de futur

Hi ha unes futures millores que no s'han implementat per la dificultat i el salt de coneixements necessaris. Algunes d'aquestes podrien ser:

- Fer que la aplicació funcioni en HTTPS. D'aquesta manera milloraria la seguretat de l'aplicació i les dades que viatgen per la xarxa.
- Establir la possibilitat de vincular el teu compte de les xarxes socials (Twitter, Facebook, Google, etc). D'aquesta forma, l'usuari no hauria d'introduir cap dada per a registrar-se i podria compartir les seves experiències amb els seus contactes de manera més ràpida i senzilla. Això s'hauria de fer utilitzant les APIs proveïdes per aquestes mateixes empreses i inserir-les a la pàgina principal.
- Possibilitat de comptes avançats o premium de pagament. Per a aquesta implementació s'hauria d'integrar la possibilitat de efectuar pagaments via PayPal. S'hauria d'afegir d'una manera semblant a la anterior

Aplicacions Híbrides

Des de que es va començar aquest projecte han arribat un nou tipus d'aplicacions que semblen destinades a finalitzar amb aquesta "guerra": les aplicacions híbrides.

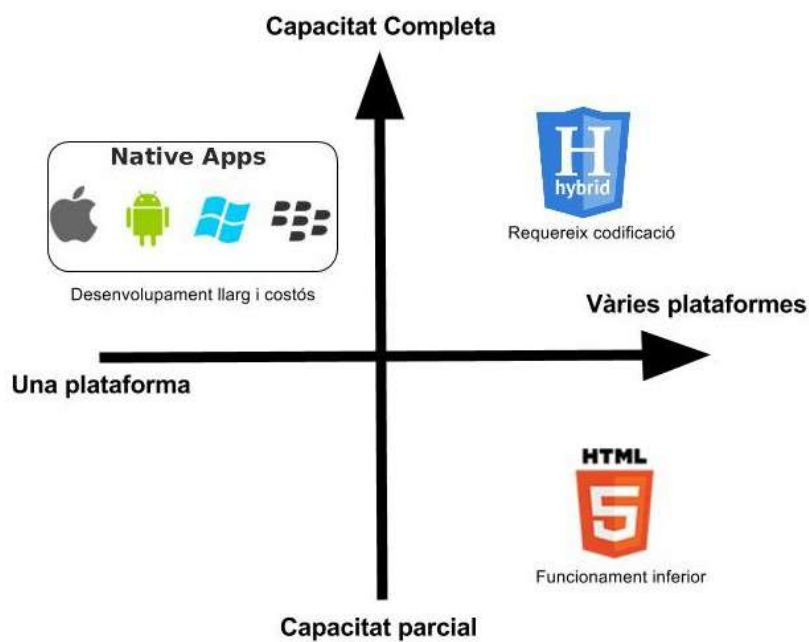
Les aplicacions híbrides estan desenvolupades en part amb l'entorn de desenvolupament natiu i en part en llenguatge web. Aquestes necessiten almenys el desenvolupament del “contenedor” natiu que allotgi les parts del llenguatge web. D'aquesta manera s'introdueixen ponts en algunes llacunes entre aplicacions natives i web.

Encara que el desenvolupament híbrid resulta més econòmic, un aspecte a tenir en compte en la creació d'una app que doni suport a diferents tasques, les aplicacions natives són més segures i reaccionen més ràpid que les apps en HTML5, pel que són més recomanables en casos en el que hi hagin moltes iteracions amb l'usuari.

Per la seva part, el desenvolupament híbrid o HTML5 tenen un cost més baix, un aspecte a tenir en compte. A més a més són més atractives per a equips interns que tenen coneixement de desenvolupament web i volen desenvolupar aplicacions addicionals per diversió.

Per contra el funcionament visual és HTML5, per tant requereixen connexió.

El temps dirà quin tipus d'aplicació és el guanyador d'aquesta batalla.



Imatge 16: Comparació aplicacions web, natives i híbrides

12. Apèndix

Software utilitzat

En aquesta secció s'explicaran els diferents softwares utilitzats en el desenvolupament d'aquesta aplicació, alguns ja esmentats anteriorment.

Tots els softwares són de codi obert, és a dir, programari distribuït i desenvolupat lliurement que es centra més en els beneficis pràctics i en la llibertat de poder modificar la font del programa sense restriccions de llicència.

Asterisk



Asterisk és una aplicació que proporciona les funcionalitats d'una central telefònica. Es pot connectar a un nombre determinat de telèfons per a trucades entre si i fins i tot pot ser o connectar-se a un proveïdor/servidor de VoIP.

Asterisk inclou moltes característiques i funcionalitats, inclús se'n poden crear de noves escrivint nous “dialplans”, en llenguatge script intern d'Asterisk, o afegint altres mòduls nous.

El “dialplan” o Pla de Marcació és la columna vertebral del sistema. És un llenguatge script, en el que funcions, aplicacions i recursos es van intercalant per formar procediments i accions.

Possiblement, la qüestió més important a l'hora d'utilitzar Asterisk és que reconeix molts protocols per a establir comunicacions de VoIP.

HTML5



HTML5 és l'actualització de HTML, el llenguatge en el que és creada una pàgina web. HTML5 també és un terme de màrqueting per a agrupar les noves tecnologies de desenvolupament d'aplicacions web: HTML5, CSS3 i noves capacitats de JavaScript. És la primera vegada que HTML i XHTML s'han desenvolupat en paral·lel. El desenvolupament d'aquest llenguatge és regulat pel Consorci World Wide Web Consortium (W3C)

La versió anterior, HTML4, no té les característiques necessàries per a la creació d'aplicacions modernes basades en un navegador. La utilització de JavaScript ha ajudat a millorar això, gràcies a frameworks.

Per altra banda, Flash ha sigut reemplaçat per HTML a l'hora de desenvolupar web apps amb "habilitats": àudio, vídeo, webcams, micròfons, animacions vectorials, etc. Ara HTML5 és capaç de fer això sense la necessitat de plugins i amb una gran compatibilitat entre navegadors.

En aquest projecte s'ha utilitzat el suport integrat per al contingut multimèdia gràcies als elements <àudio> i <vídeo>, oferint la possibilitat de afegir contingut multimèdia en documents HTML. [13]

JavaScript



JavaScript

Llenguatge de programació interpretat. Es defineix com a orientat a objectes, basat en prototips, imperatiu i dinàmic. S'utilitza normalment al client, implementat com a part d'un navegador web. Permet millores a la interfície d'usuari i ajuda a fer les pàgines web més dinàmiques.

Per altra banda, també existeix una forma al "server-side", costat del servidor. A més a més s'utilitza per a desenvolupar petits "widgets", petites aplicacions inserides a dins de les planes web.

Tots els navegadors moderns interpreten el codi JavaScript integrat a les planes web. Per a interactuar amb una web, es proveeix el llenguatge d'una implementació del Document Object Model (DOM). Aquest model és essencialment una interfície de plataforma que proporciona un conjunt estàndard d'objectes per a representar HTML, XHTML i XML, un model estàndard sobre com poden combinar-se aquests objectes i una interfície per a accedir a ells i manipular-los.

Donat que JavaScript pot ser executat localment en el navegador de l'usuari, el navegador pot respondre a les accions de l'usuari amb rapidesa, fent a l'aplicació més sensible. Per altra banda, el codi pot detectar accions dels usuaris de HTML, que per si sol no pot (p.ex. pulsacions de teclat).

En el desenvolupament d'aquest projecte s'ha usat per a dotar de dinamisme al flux de l'usuari a través dels diferents passos de l'aplicació.

A més a més, s'han usat dos APIs: WebRTC i WebQR. L'una conté les funcions per a connectar-se amb Asterisk i reproduir els àudios que aquest envia. La segona proporciona les

funcions necessàries per a connectar-se amb la càmera del dispositiu per a llegir el codi QR. Aquestes APIs estan explicades més endavant. [13]

CSS

CSS



Full d'estil en cascada (CSS) és un llenguatge usat per a definir i crear la presentació d'un document HTML o XML. La idea que hi ha darrera del desenvolupament de CSS és separar l'estructura d'un document de la seva presentació. Aquesta informació pot estar definida en un document separat o en el mateix HTML.

En aquest projecte s'ha utilitzat un framework de CSS anomenat **Bootstrap**.



Bootstrap és un programari lliure per a dissenyar pàgines web i aplicacions web. Conté les etiquetes de HTML i algunes extensions de JavaScript.

És compatible amb les últimes versions dels navegadors més populars. També ha adoptat la filosofia del disseny per a plataformes mòbils que el fa idoni per a aplicacions web utilitzades en dispositius portàtils.

A més a més, el fet de que sigui fàcil d'aprendre'n, intuïtiu i amb moltes plantilles fa més fàcil el disseny d'estil de les pàgines web. [13]

PHP



PHP (acrònim recursiu de PHP: Hypertext Preprocessor) és un llenguatge de codi obert molt popular i especialment adequat per al desenvolupament web i que pot ser incrustat en HTML. S'utilitza per a crear pàgines web de forma dinàmica. S'executa al cantó del servidor, per aquest motiu al navegador web ja li arriba la pàgina en format HTML. No es pot visualitzar-ne el codi php a client.

PHP es pot incloure a dins del codi HTML. Quan el llenguatge s'estigui interpretant, s'executarà el codi que està entre les etiquetes, generant un codi HTML com a sortida. El codi HTML generat pel codi PHP es barrejarà amb la pàgina HTML on hagués estat inclòs.

- Avantatges

- És un llenguatge multi plataforma.
- Completament orientat al desenvolupament d'aplicacions web dinàmiques amb accés a informació emmagatzemada en una base de dades.
- El codi font escrit en PHP és invisible al navegador i el client, ja que és el servidor el que s'encarrega d'executar el codi i enviar el seu resultat HTML al navegador. Això fa que la programació en PHP sigui segura i de confiança.
- La capacitat de connexió amb la majoria dels motors de base de dades que s'utilitzen en l'actualitat destaca la seva connectivitat amb MySQL i PostgreSQL.
- És lliure, pel que es presenta com a una alternativa de fàcil accés per a tothom.
- No requereix definició de tipus de variables encara que les seves variables es poden avaluar també pel tipus que estiguin fent servir en temps d'execució.
- Té maneig d'excepcions (des de PHP5).

PHP, en aquest projecte, s'ha utilitzat per a gestionar la creació i control dels usuaris de la aplicació. [13]

Apache2



El servidor HTTP Apache és un servidor web de codi obert que implementa el protocol HTTP/1.1 i la noció de lloc virtual. El codi, és desenvolupat i mantingut per una comunitat de usuaris sota la supervisió de la Apache Software Foundation dins del projecte HTTP Server (httpd).

Apache presenta, característiques altament configurables, però va ser criticat per la falta d'interfície gràfica que ajudi a la seva configuració.

La majoria de les vulnerabilitats de seguretat descobertes i resoltes, només poden ser aprofitades per usuaris locals i no remotament.

Avantatges:

- Modular
- Codi obert

- Multi-plataforma
- Extensible
- Popular (fàcil d'aconseguir ajuda, suport i actualitzacions)

Apache és utilitzat principalment per a enviar pàgines web estàtiques i dinàmiques en la World Wide Web. En aquest projecte, el servidor Apache és l'encarregat de proveir la informació HTML de les pàgines web als navegadors dels dispositius dels clients. [13]

PostgreSQL



Sistema de gestió de bases de dades relacionals orientat a objectes i lliure.

Com altres projectes de codi obert, el desenvolupament de PostgreSQL no és manejat per una empresa, sinó que és dirigit per una comunitat de desenvolupadors de forma altruista.

Aquest fet va ser determinant en l'elecció d'aquest davant MySQL, ja que aquest últim pertany a l'empresa multinacional de software ORACLE Corporation. [13]

A més a més, aquest sistema compta amb els següents avantatges:

- Seguretat
- Integritat en BD i integritat referencial
- “Assertions”
- “Triggers”

PHPPgAdmin



Aplicació web escrita en PHP per a administrar bases de dades PostgreSQL.

Originalment va començar utilitzant el codi d'una aplicació més popular, PHPMyAdmin, que proveeix una interfície web convenient per als usuaris per a gestionar bases de dades MySQL. En l'actualitat, el codi va ser reescrit i ara és completament diferent i ja no conté cap tros de codi de la aplicació anterior. [13]

Funcionalitats més destacades a l'hora de fer l'elecció d'aquest programa:

- Administrar diversos servidors.
- Gestionar tots els aspectes de : Usuaris i grups, bases de dades, esquemes, taules, índexs, restriccions, disparadors , regles i privilegis, vistes , seqüències i funcions, informes.
- Fàcil manipulació de dades : explorar taules , vistes i informes, executar SQL arbitrari.
- Scripts SQL d'importació , còpia de dades , XML, CSV i pestanyes.
- Fàcil d'instal·lar i configurar.

Linux Ubuntu



Ubuntu és una distribució del sistema operatiu GNU/Linux basada en Debian. El seu patrocinador és Canonical, una companyia britànica. A més de mantenir-lo lliure i gratuït, l'empresa és capaç d'aprofitar els desenvolupadors de la comunitat per millorar els components del seu sistema operatiu.

Aquesta empresa, a més a més de mantenir Ubuntu, també proveeix una versió orientada a servidors, Ubuntu server, utilitzada en aquest projecte.

Es va escollir Ubuntu, per ser un programari de codi obert, ser robust, tenir una versió server molt més lleugera i per la qualitat del suport tècnic d'aquest. [13]

Git



Software de control de versions pensat en l'eficiència del manteniment de versions d'aplicacions quan aquestes tenen un gran nombre d'arxius de codi font.

A l'inici es va pensar com un motor de baix nivell sobre el qual altres poguessin escriure la interfície d'usuari. Però s'ha convertit en un sistema de control de versions de funcionalitat plena. Hi ha molts projectes de molta rellevància que ja utilitzen Git.

Entre les característiques més rellevants:

- Suport al desenvolupament no lineal, per tant, rapidesa en la gestió de branques i barreja de diferents versions. Git inclou eines específiques per a navegar i visualitzar un historial de desenvolupament no lineal.
- Git li dona a cada desenvolupador una còpia local de l'historial del complet desenvolupament i els canvis es propaguen entre els repositoris locals. Els canvis s'importen com branques i poden ser fusionats en la mateixa manera que es fa amb la branca local.
- Els magatzems d'informació es poden publicar per HTTP o FTP (ja sigui a través d'una connexió TCP/IP simple o a través del xifrat SSH).
- Es poden utilitzar directament els repositoris de Subversion
- Gestió eficient de projectes grans, donada la rapidesa de gestió de diferències entre arxius.

En el desenvolupament d'aquesta aplicació, no s'ha utilitzat tot el potencial d'aquesta eina. Només les comandes més bàsiques han sigut usades. Tot i així, ha servit per a comprendre el desenvolupament en branques actual i a agilitzar el desenvolupament. [13]

QR



El codi QR (sigla provinent de l'anglès *Quick Response* o resposta ràpida) és un sistema per a emmagatzemar informació en una matriu quadrada de punts dissenyada per a ser llegida amb la càmera d'un telèfon intel·ligent o tauleta tàctil entre d'altres.

La informació codificada en una etiqueta QR conté caràcters alfanumèrics que poden contenir text simple, adreces web i targetes de presentació en format Vcard, entre d'altres formats.

L'avantatge dels codis QR és l'accés immediat a la informació a la qual fan referència, però, a pesar de que aquest és un aspecte molt positiu, pot resultar perillós ja que la forma immediata en la que ens facilita la informació o ens envia a un lloc web, pot vulnerar el dispositiu. Sense previ avís, es pot accedir a pàgines infectades o instal·lar software maliciós. Es recomana que només s'utilitzin els codis QR de llocs de confiança i saber escollir les aplicacions que permetin visualitzacions prèvies d'aquest codis QR. Així es coneix el contingut abans d'accedir-hi. [13]

Web QR

En aquest projecte s'ha utilitzat el codi de les llibreries de WebQR. Aquest software ofereix un lector de codis QR directament al navegador, implementat en JavaScript.

A diferència d'altres productes, aquesta solució proporciona una manera més fàcil de llegir els codis QR sense que cap càrrega d'imatges sigui necessària.

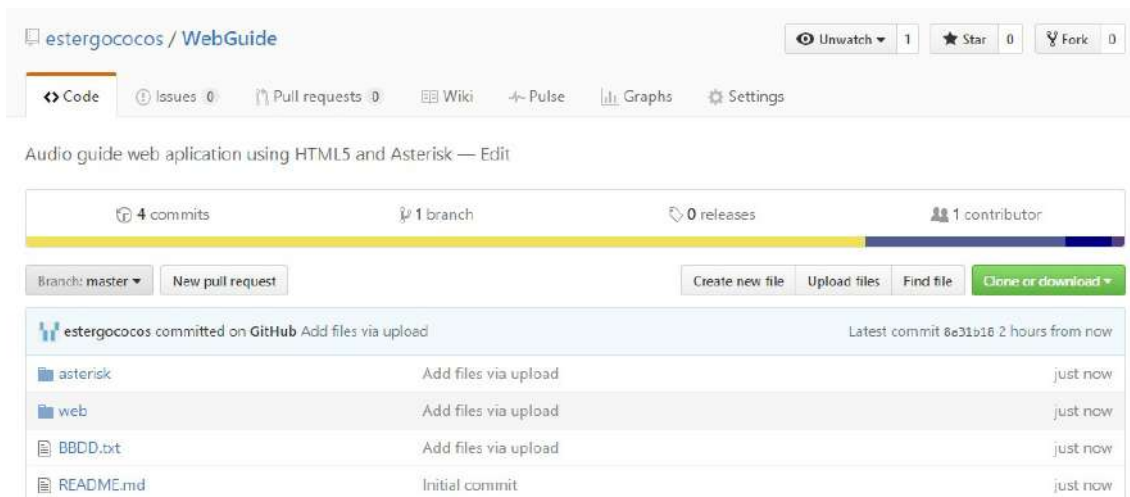
El software utilitza les funcions de JavaScript que controlen la captació de vídeo utilitzant la càmera del dispositiu. Aquestes imatges són “llegides” per una implementació de JavaScript que es basa en la biblioteca de processament d'imatges de codi de barres ZXing multiformat 1D / 2D, implementat en Java. [17]

Projecte a GitHub

El projecte consta de tres grans seccions:

- Configuració i posada a punt del servidor web amb els arxius de l'aplicació
- Asterisk i la seva configuració d'usuaris
- Configuració de base de dades

Al repositori hi podem trobar:



The screenshot shows the GitHub repository page for 'estergococos / WebGuide'. The repository is in the 'master' branch and has 4 commits, 1 branch, 0 releases, and 1 contributor. The commit history shows four commits: 'asterisk', 'web', 'BBDD.txt', and 'README.md', all added via upload 'just now'.

Branch	Commits	Branches	Releases	Contributors
master	4	1	0	1

File	Action	Time
asterisk	Add files via upload	just now
web	Add files via upload	just now
BBDD.txt	Add files via upload	just now
README.md	Initial commit	just now

Instruccions d'instal·lació

Si es vol reproduir o instal·lar la aplicació en un servidor s'han de seguir les instruccions següents on s'utilitzaran els arxius i carpetes mostrats a la imatge anterior.

1. Instal·lar Linux server en qualsevol servidor. Quasi totes les distribucions són compatibles. En aquest cas s'ha usat Ubuntu.
2. Configurar la xarxa d'entrada a */etc/network/interfaces*, així com obrir i comprovar que estiguin disponibles els ports 8088 i 5060.
3. Instal·lar i configurar Apache2 (la configuració per defecte ja és adequada). Es pot utilitzar la comana:

```
sudo apt-get install apache2
```

4. Copiar la carpeta web a la ruta */var/www/html/*
5. Iniciar el servei del servidor web:

```
sudo apache start
```

6. Instal·lar PHP

```
sudo apt-get install php
```

7. Instal·lar PostgreSQL

```
sudo apt-get install postgresql
```

8. Configurar PostgreSQL. Crear una base de dades i les taules com està descrit a dintre de l'arxiu *BBDD.txt*
9. Instal·lar Asterisk

```
sudo apt-get install asterisk
```

10. Configurar Asterisk copiant la carpeta asterisk substituint la carpeta de la ruta */etc/asterisk/*

11. Iniciar el servei d'Asterisk:

```
sudo asterisk start
```

Ja tindrem un servidor operatiu, accessible des de qualsevol terminal connectat a la xarxa configurada al pas dos.

Glossari

Recull dels acrònims utilitzats en aquesta memòria:

- API: Application Programming Interface
- App: Mobile application
- BD: Base de dades
- CSS: Cascading Style Sheets
- FTP: File Transfer Protocol
- HTML: Hyper Text Markup Language
- HTTP: HyperText Transfer Protocol
- ICE: Information and Content Exchange
- IMS: Information Management Services
- IP: Internet Protocol
- LTE: Long-Term Evolution
- PHP: PHP: Hypertext Preprocessor
- PSTN: public switched telephone network
- QR: Quick Response
- SIP: Session Initiation Protocol
- SQL: Structured Query Language
- SSH: Secure SHell
- TCP/IP: Transmission Control Protocol
- VoIP: Voice over IP
- WebRTC: Web Real-Time Communication

13. Referències

Planes web

[1] “Flash muere en dispositivos móviles y HTML5 es el objetivo de Adobe”, *Htm15facil*, <http://html5facil.com/>, setembre 2012, 12 de juny de 2016

[2] “Análisis: El futuro de las aplicaciones... ¿La muerte de la web?”, *Mercadotecnia*, <https://mercadotecnia.portada-online.com>, novembre de 2012, 12 de juny de 2016

[3] Pedro Gutiérrez “HTML5 VS apps nativas”, *Genbeta:dev*, <http://www.genbetadev.com/>, juliol de 2013, 12 de juny de 2016

[4] Obaro Ogbo, “HTML5 VS NATIVE ANDROID APP”, *Androidauthority*, <http://www.androidauthority.com/>, maig de 2015, 12 de juny de 2016

[5] Curtis Franklin Jr., “HTML 5 Vs. Native Apps: What's Best For Developers?”, *InformationWeek*, <http://www.informationweek.com/>, 26 d'octubre de 2015, 12 de juny de 2016

[6] Janna Badalian, “HTML5 vs Native: The Debate is Over ”, *MobileSmith*, <https://www.mobilesmith.com>, 20 de maig de 2015, 12 de juny de 2016

[7] Mario Korf i Eugene Oksman, “Native, HTML5, or Hybrid: Understanding Your Mobile Application Development Options”, *Salesforce Developer*, <https://developer.salesforce.com>, juny de 2016, 12 de juny de 2016

[8] Jacob Cass, “HTML5 vs Native Apps: What’s best for 2016?”, *JustCreative*, <http://justcreative.com/>, 17 març 2016, 12 de juny de 2016

[9] Alfonso Fernández Gutierrez, “Aplicaciones web vs. aplicaciones nativas vs. aplicaciones híbridas”, *Binkbig.com*, <http://blogthinkbig.com/>, 20 de febrer 2013, 12 de juny de 2016

[10] Ángel González, “En 2015, el 60% de las apps serán híbridas y combinarán desarrollo nativo, HTML5 y CSS3”, *techWEEK*, <http://www.techweek.es/>, 9 de gener de 2014, 12 de juny de 2016

[11] <https://www.mobilesmith.com/>

[12] <https://s3.amazonaws.com>

[13] <https://www.wikipedia.org/>

[14] <https://upload.wikimedia.org>

[15] <http://www.voip-info.org>

[16] <https://www.doubango.org/sipml5/>

[17] <https://webqr.com>

[18] <http://www.tizenexperts.com>

[19] Eric Bidelman, “Capturing Audio & Video in HTML5 – HTML5 Rocks”, <https://www.html5rocks.com/es/tutorials/getusermedia/intro/>, 12 d’octubre de 2016

