
This is the **published version** of the bachelor thesis:

Rescalvo Rius, David; Robles Martínez, Sergi, dir. Qiosk : desenvolupament d'un dispositiu de captació de l'opinió per a esdeveniments acadèmics. 2024. (Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/298986>

under the terms of the  license

Qiosk: Desenvolupament d'un dispositiu de captació de l'opinió per a esdeveniments acadèmics

David Rescalvo Rius

2 de juliol de 2024

Resum– La baixa participació és problemàtica perquè impedeix obtenir una mostra representativa de les opinions dels assistents, la qual cosa dificulta la presa de decisions informades per millorar futurs esdeveniments. El projecte "Qiosk" desenvolupa un dispositiu físic per captar l'opinió en esdeveniments acadèmics mitjançant una pantalla tàctil i una botonera, recollint dades in situ que permetrien millorar la qualitat dels esdeveniments gràcies a una alta participació. Aquest sistema inclou una API REST Node.js i una interfície web Vue.js per a la gestió i anàlisi de les respostes. Basat en una arquitectura client-servidor, el dispositiu envia dades al servidor mitjançant HTTP Requests, i aquestes es guarden en una base de dades MySQL, garantint una gestió segura i eficient de la informació.

Paraules clau– retroalimentació, enquestes, participació, quiosc, API REST, middleware, base de dades, interfície web, Python, Node.js, Vue.js, MySQL, JSON

Abstract– Low participation is problematic because it prevents obtaining a representative sample of attendees' opinions, making it difficult to make informed decisions to improve future events. The "Qiosk" project develops a physical device to capture feedback at academic events using a touchscreen and button panel, collecting on-site data that would improve event quality through high participation. This system includes a Node.js REST API and a Vue.js web interface for managing and analyzing responses. Based on a client-server architecture, the device sends data to the server via HTTP requests, which are stored in a MySQL database, ensuring secure and efficient information management.

Keywords– feedback, surveys, participation, kiosk, REST API, middleware, database, web interface, Python, Node.js, Vue.js, MySQL, JSON

1 INTRODUCCIÓ - CONTEXT DEL TREBALL

AQUESTA secció exposa la situació actual referent a la retroalimentació a través d'enquestes d'esdeveniments, com aquesta retroalimentació ens ajuda a prendre decisions i millorar la qualitat envers futurs esdeveniments. Però, a causa de la baixa participació, aquestes dades no ens acaben de servir del tot i no podem veure una imatge completa de l'opinió dels assistents. És per això

- E-mail de contacte: 1528514@uab.cat
- Menció realitzada: Tecnologies de la Informació
- Treball tutoritzat per: Sergi Robles Martínez (Departament d'Enginyeria de la Informació i de les Comunicacions)
- Curs 2023/24

que la nostra proposta es basa en un sistema de captació de l'opinió en persona, que permeti respondre enquestes en el mateix lloc de l'esdeveniment, amb una alta escalabilitat i que pugui ser utilitzat per diferents esdeveniments al mateix temps. Aquest sistema podrà captar més dades que seran usades per a fer mineria de dades.

Un bon feedback és important per millorar la qualitat dels esdeveniments acadèmics, mitjançant aquesta retroalimentació podem tenir una imatge més real del que cerca la comunitat acadèmica i podem aportar millors propostes que s'ajustin a aquests paràmetres, això podria millorar la taxa d'assistència i l'interès general.

Un problema important, i que sembla que està creixent, és la baixa taxa de resposta a les enquestes en línia i telefòniques, especialment, aquests tipus d'enquestes creen

un cert rebuig per part dels enquestats.[1]

La nostra proposta consisteix en un sistema amb més d'un dispositiu físic que permetran als assistents avaluar una activitat acadèmica de manera similar al que es veu als lavabos dels aeroports i altres llocs, on es pot valorar l'experiència. La pantalla tàctil mostrarà una pregunta, i amb uns botons físics amb il·luminació els assistents podran respondre a les preguntes. Opcionalment, aquests dispositius podran incloure una càmera per poder identificar el gènere i l'edat de l'enquestat, amb la finalitat de realitzar mineria de dades que podria obrir una línia futura d'investigació.

1.1 Objectius

Tot seguit s'especificaran els objectius d'aquest projecte, en ordre prioritari d'execució, per complir l'objectiu global de desenvolupar un dispositiu de captació de l'opinió per a esdeveniments acadèmics proposat anteriorment.

- Dissenyar i desenvolupar un quiosc que consti d'una botonera, una pantalla tàctil i un ordinador, aquest dispositiu ha de ser capaç de mostrar les preguntes per la pantalla tàctil, recollir les respostes segons el botó premut i enviar les dades a l'API REST.
- Desenvolupar una API REST capaç d'assignar enquestes als quioscos i de rebre les dades d'aquest per emmagatzemar-les en una base de dades. Haurà de poder gestionar un nombre indeterminat de quioscos.
- Desenvolupar una interfície web per a l'usuari administrador que permeti gestionar les enquestes actives i els quioscos assignats a les enquestes.
- Dissenyar un sistema de reconeixement facial per aportar més dades i així obrir una nova línia d'estudi utilitzant la mineria de dades.

2 ESTAT DE L'ART

El projecte cerca captar l'opinió a través de quioscos similar a la forma que ho fan en alguns aeroports, no és un sistema exclusiu dels aeroports sinó que hi ha altres sistemes i implementacions, els quals poden ser rellevants per al desenvolupament del sistema proposat.

SurveyStance

Mitjançant l'ús de terminals iPads, coneguts com a "Smiley Surveys", situats en diferents punts com ara banys d'aeroports o prop dels controls de seguretat, aquests terminals permeten als passatgers donar opinió de forma ràpida i fàcil, simplement prement un botó amb una cara feliç o trista per indicar la seva experiència. Aquesta informació ajuda els aeroports a millorar els seus serveis, ja que poden mesurar el rendiment del personal i la qualitat del servei. El sistema no es limita al seu ús en aeroports, també pot ser utilitzat per altres empreses.[2]

És un sistema privatiu que utilitza tauletes en els quioscos, té implementacions interessants en les quals ens podem fixar.

FeedbackNow

És un sistema de captació de l'opinió en aeroports, a través de les seves caixes amb emoticones, proporciona resultats

en temps real i s'adapta perfectament a àrees clau del servei ofert com ara seguretat, lavabos, facturació, comerços, hostaleria i recollida d'equipatges. El seu disseny intuïtiu assegura altes taxes de resposta i ofereix una visió màxima de la satisfacció dels passatgers per millorar la qualitat del servei. Amb només una pregunta i tres botons, són dispositius innovadors i intuïtius que no presenten barreres per als passatgers, permetent una resposta ràpida i eficaç.[3] És un sistema privatiu versàtil que només es pot respondre a una pregunta perquè manca de pantalla.

neXenio/Feedback-Kiosk

És una eina senzilla per recollir l'opinió en temps real, aquesta consta d'una aplicació React, mostra una enquesta generada a partir d'un fitxer JSON, que es pot configurar segons es necessiti, i un servidor Node que agrupa el feedback enviat i analitza les respostes utilitzant Google Analytics (o un altre proveïdor d'anàlisi diferent).[4] La llicència d'aquest repositori concedeix el permís a qualsevol persona que obtingui una còpia d'aquest programari per utilitzar-lo, per aquest motiu i que està desenvolupat en JavaScript, ens pot ser molt útil a l'hora de desenvolupar l'API REST.

smahesh29/Gender-and-Age-Detection

En aquest projecte de Python, s'utilitza OpenCV per identificar de forma aproximada el gènere i l'edat d'una persona a partir d'una única imatge del rostre.[5] La llicència d'aquest repositori concedeix el permís a qualsevol persona que obtingui una còpia d'aquest programari per utilitzar-lo i està desenvolupat en Python, ens garanteix que ho podem implementar en els nostres dispositius.

3 ANÀLISI DE REQUISITS

En aquesta secció veurem l'anàlisi del sistema i els requisits necessaris per al desenvolupament que ajudaran a garantir que el sistema final s'ajusti als objectius.

3.1 Anàlisi del sistema

Els dispositius físics constaran d'una botonera, una pantalla tàctil i un ordinador. La pantalla tàctil mostrarà la pregunta i permetrà canviar l'idioma d'aquesta. Podran incorporar també un sensor d'ultrasons que permetrà saber quan una persona està davant del dispositiu, quan aquest ha marxat, per poder refrescar l'enquesta, i quantes persones passen per davant. Opcionalment, es podria incloure una càmera per poder identificar el sexe i l'edat de l'enquestat.

S'utilitzarà Python per al desenvolupament del programari dels quioscos i es connectaran a una API REST desenvolupada amb Node.js. Aquest servidor tindrà una interfície d'usuari desenvolupada en Vue.js per poder gestionar preguntes, quioscos operatius, enquestes actives, i altres funcionalitats necessàries. A més, aquest servidor tindrà una base de dades SQL per emmagatzemar les dades recollides, les preguntes, les respostes, les enquestes, i altres informacions rellevants per al funcionament del sistema, tal com mostra la figura 1.

Quan els quioscos s'engeguin es connectaran a l'API en cerca d'enquestes obertes associades al quiosc. Aquests quioscos hauran de poder guardar la informació de forma

local en cas de fallada de la connexió i poder-la enviar en cas de reconexió amb el servidor.

El sistema ha de ser robust, segur i fiable, això significa que s'ha d'implementar mètodes de seguretat tant en la connexió entre els quioscos i el servidor com amb la connexió de l'usuari amb el servidor o la base de dades. També s'haurà de contemplar tots els casos possibles de fallada per evitar els possibles errors i que sigui robust, i la capacitat d'assegurar les dades de manera fiable i íntegra.

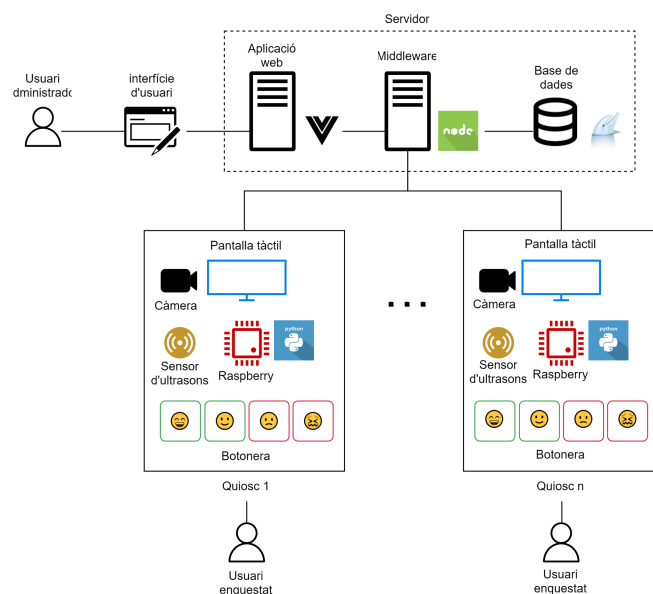


Fig. 1: Esquema del sistema

3.2 Requisits funcionals

Aquests requisits descriuen les accions que el sistema ha de poder realitzar per assegurar que s'hagin complert els objectius. Aquestes funcionalitats es classifiquen segons la prioritat en el desenvolupament: 1. Obligatori per a la primera implementació. 2. Obligatori per a la implementació final. 3. Funció opcional.

RF1 - El sistema Qiosk ha de ser capaç de mostrar per pantalla les preguntes de l'enquesta activa. Prioritat: 1.

RF2 - El sistema Qiosk ha de proveir a l'usuari enquestat la capacitat de canviar l'idioma de les preguntes en qualsevol moment. Prioritat: 3.

RF3 - El sistema Qiosk hauria de ser capaç de detectar si hi ha un usuari a enquestar, a través d'uns sensors d'ultrasons, per tal d'activar l'enquesta. Prioritat: 3.

RF4 - El sistema Qiosk podrà saber el nombre de persones que passen per davant del quiosc per tal de saber quin percentatge aproximat d'assistents responen l'enquesta. Prioritat: 3.

RF5 - El sistema Qiosk podrà identificar l'edat i el sexe dels usuaris enquestats, a través d'una camera, per tal de poder recopilar dades valuoses per a la mineria de dades. Prioritat: 3.

RF6 - El sistema Qiosk ha de proveir a l'usuari administrador la capacitat de gestionar les enquestes i les preguntes d'aquestes. Prioritat: 1.

RF7 - El sistema Qiosk ha de proveir a l'usuari administrador la capacitat de gestionar en quins quioscos es mostraran

les enquestes i quines enquestes. Prioritat: 1.

RF8 - El sistema Qiosk ha de ser capaç d'emmagatzemar les dades recollides en una base de dades. Prioritat: 1.

RF9 - En cas de no poder establir connexió amb el servidor, el sistema Qiosk hauria de proveir a l'usuari administrador la capacitat de carregar les preguntes de forma manual als quioscos. Prioritat: 2.

RF10 - En cas de no poder establir connexió amb el servidor, el sistema Qiosk hauria de proveir a l'usuari administrador la capacitat d'extreure les dades de forma manual dels quioscos. Prioritat: 2.

RF11 - El sistema Qiosk hauria de proveir a l'usuari administrador la capacitat de carregar dades de forma manual a la base de dades. Prioritat: 3.

RF12 - El sistema Qiosk hauria de proveir a l'usuari administrador la capacitat de canviar els quioscos de mode per tal que es puguin utilitzar de forma offline. Prioritat: 3.

RF13 - El sistema Qiosk podrà narrar les preguntes de les enquestes a través d'un altaveu que incorporaran els quioscos. Prioritat: 3.

RF14 - El sistema Qiosk podrà emetre una alerta sonora per avisar de la presència del quiosc. Prioritat: 3.

3.3 Requisits no funcionals

Aquests descriuen els atributs del sistema que no estan relacionats amb funcionalitats específiques, sinó amb les característiques que defineixen la seva qualitat i seguretat.

RNF1 - Si és necessari, el sistema Qiosk ha de ser capaç d'incorporar més quioscos.

RNF2 - El sistema Qiosk ha de garantir la integritat de les dades.

RNF3 - El sistema Qiosk ha de garantir la seguretat en la connexió amb l'usuari administrador.

RNF4 - El sistema Qiosk ha de garantir la seguretat en la connexió amb els quioscos.

RNF5 - El sistema Qiosk hauria de garantir només accés a l'usuari administrador en els quioscos a través d'una contrasenya o un pin.

RNF6 - Si és necessari, el sistema Qiosk ha de garantir l'escalabilitat de la base de dades.

RNF7 - El sistema Qiosk podrà garantir un nombre aproximat fiable de persones que passen per davant dels quioscos.

RNF8 - El sistema Qiosk ha de contemplar qualsevol escenari de fallada.

RNF9 - El sistema Qiosk podrà garantir l'accessibilitat a persones amb diversitat funcional.

RNF10 - El sistema Qiosk ha de garantir l'anonimat de les dades.

RNF11 - El sistema Qiosk hauria de ser capaç de recollir dades pròximes a la realitat.

4 METODOLOGIA

En aquest apartat, s'especifica la metodologia emprada per al desenvolupament del sistema Qiosk. S'ha decidit utilitzar la metodologia *Agile Scrumban*, que fusiona l'estructura i les rutines previsibles de *Scrum* amb la flexibilitat de *Kanban* per fer que el desenvolupament siguin més àgil, eficient i productiu.

4.1 Scrumban

Scrumban, és una metodologia de gestió de projectes que combina característiques importants de dues metodologies Agile: Scrum i Kanban. Aquesta metodologia pot ajudar a centrar-se en les tasques estratègiques al mateix temps que millora els seus processos. Així doncs, aquesta pràctica permet planificar el desenvolupament de les tasques crucials del projecte i deixa certa flexibilitat per millorar aspectes o afegir noves funcionalitats. Encara que es pot utilitzar iteracions si es desitja, el més important és que les tasques es puguin afegir o eliminar durant cada iteració del producte, assegurant una eficiència màxima.

4.2 Implementació

La implementació de l'Scrumban es realitzarà mitjançant Trello, una eina visual que permet als equips gestionar qual-sevol tipus de projecte i flux de treball, així com supervisar tasques.

4.3 Fases Scrumban

En la taula 1 es mostra les cinc fases per les quals passarà el projecte i la descripció de cada fase.

TAULA 1: TAULA FASES SCRUMBAN

Fase	Descripció
Iniciació	Primera reunió amb el client on s'aborda el problema.
Planificació	Es detalla els objectius dels projecte juntament amb els requisits, d'aquests s'extreu el product backlog. Es planifica les iteracions amb el seu iteration backlog corresponent. Es crea la taula Scrumban i s'estableix els límits del Work In Progress (WIP).
Execució	Execució de les iteracions planificades en la fase anterior.
Retroalimentació i lliurament	Es prova el sistema i s'extreu conclusions del seu funcionament. Es lliura el projecte al client.

4.4 Taulers Scrumban

S'aprofita l'aspecte visual dels taulers Kanban, que els permeten prioritzar i seguir el progrés de les tasques en temps real. Això els permet avaluar ràpidament el que s'ha de fer i quan. El seu objectiu és definir quines tasques s'han de completar durant la iteració. Això implica seleccionar les tasques del product backlog i afegir-les al iteration backlog.

El procés Scrumban comença amb la creació de la taula, una representació visual de les tasques futures. Utilitzada en la metodologia Kanban, aquesta eina permet als equips visualitzar i gestionar el seu flux de treball. És un aspecte fonamental de l'enfocament Scrumban.

El tauler Scrumban per aquest projecte consistirà en quatre columnes: "Product backlog", "Iteration backlog", "Doing" i "Done":

Una altra característica del Scrumban és el que es coneix com a "Work In Progress". Això determina el nombre de tasques que es poden dur a terme en un moment determinat. Per exemple, podria estar limitat a un WIP a la vegada, aquesta s'ha de completar i moure a la columna "Done" de la taula abans que es pugui avançar a la següent tasca. L'objectiu general d'aquest enfocament és centrar-se i ser productiu, assegurant no sentir-se desbordat per la càrrega de treball o distret per la feina imminent.

Tot i que els límits de treball en curs ajuden a simplificar les tasques, el Scrumban encara requereix prioritització de tasques, que es produeix durant la planificació i es pot anar modificant durant l'execució.

4.5 On-demand planning

Algunes metodologies, com el Scrum, inclouen reunions diàries on es discuteixen i es decideixen les tasques del dia, l'Scrumban no té aquest requisit. En comptes d'això, un dels seus principis fonamentals és conegut com a "On-demand planning".

Quan les tasques a la columna "Iteration backlog" de la taula Scrumban s'esgoten, això activa una planificació addicional i s'acorden noves tasques per afegir a la llista.

4.6 Iteració

Les iteracions del Scrumban estan molt menys estructurades que els sprints típics. En aquest sentit, s'apropen més al principi de flux continu associat amb Kanban. No obstant, es pot optar per utilitzar sprints més formals si es desitja.

Les iteracions per aquest projecte tindran una durada de dues setmanes, això garantirà que el projecte sigui àgil i adaptable, característiques fonamentals del Scrumban.

5 PLANIFICACIÓ

A continuació, s'explica la planificació del seguiment i el desenvolupament del sistema de captació d'opinió Qiosk, dissenyant un calendari de les fases mencionades en l'apartat 4.3 especificant les dates, obtenint les tasques necessàries segons els objectius i els requisits dels apartats 1.1 i 3, i detallant cada iteració del procés.

5.1 Taula de planificació

En la taula 2 es mostra les dates d'inici i finalització de cada fase, a més també es mostra les dates de cada iteració que es realitzarà durant la fase d'execució.

5.2 Product backlog

A continuació, es mostra una llista prioritzada de totes les tasques que s'han de realitzar en el projecte de desenvolupament del sistema Qiosk.

T01 - Dissenyar el quiosc que consisteixi en una botonera, una pantalla tàctil i un ordinador. Estimació: 1 dia.

T02 - Desenvolupar el programari del quiosc que funcioni en local. Estimació: 2 dies.

T03 - Incorporar els botons i la pantalla tàctil al quiosc. Estimació: 1 dia.

TAULA 2: TAULA DE PLANIFICACIÓ

Fase	Data d'inici	Data de finalització
Inici	18/02/2024	18/02/2024
Planificació	18/02/2024	10/03/2024
Execució	10/03/2024	16/06/2024
Iteració 1	24/03/2024	07/04/2024
Iteració 2	07/04/2024	28/04/2024
Iteració 3	28/04/2024	12/05/2024
Iteració 4	12/05/2024	02/06/2024
Iteració 5	02/06/2024	16/06/2024
Retroalimentació i lliurament	16/06/2024	29/06/2024

T04 - Desenvolupar la interfície de la pantalla tàctil que mostri les preguntes de les enquestes. Estimació: 1 dia.

T05 - Implementar la base de dades MySQL. Estimació: 5 dies.

T06 - Desenvolupar l'API REST amb Node.js. Estimació: 17 dies.

T07 - Implementar al programari dels quioscos una connexió amb l'API. Estimació: 10 dies.

T08 - Desenvolupar la interfície web utilitzant Vue.js. Estimació: 17 dies.

T09 - Implementar la funcionalitat als quioscos que permeti als usuaris canviar l'idioma. Estimació: 1 dia.

T10 - Implementar un mètode alternatiu de càrrega i descàrrega d'informació dels quioscos. Estimació: 3 dies.

T11 - Implementar un mètode per a que els quioscos puguin funcionar de forma autònoma en cas de fallada de connexió. Estimació: 4 dies.

T12 - Implementar mesures de seguretat per protegir les connexions. Estimació: 4 dies.

T13 - Integrar un sensor d'ultrasons per detectar la presència de persones. Estimació: 1 dia.

T14 - Implementar la funcionalitat per comptar les persones. Estimació: 2 dies.

T15 - Incorporar la càmera als quioscos. Estimació: 1 dia.

T16 - Implementar la funció de detecció de l'edat i el gènere dels enquestats amb OpenCV. Estimació: 10 dies.

T17 - Implementar la funció de narració de les preguntes als quioscos. Estimació: 1 dia.

T18 - Implementar alertes sonores als quioscos. Estimació: 1 dia.

T19 - Dissenyar el prototip dels quioscos. Estimació: 2 dies.

T20 - Fabricació i muntatge del prototip dels quioscos. Estimació: 5 dies.

T21 - Proves d'integració del sistema. Estimació: 5 dies.

En el desenvolupament d'aquest projecte, s'ha decidit descartar les tasques T14, T16, T17 i T18, a causa d'un canvi en la planificació inicial. No obstant això, aquestes tasques no seran oblidades, sinó que es mantindran en consideració per al disseny del sistema, amb la finalitat de poder implementar-les en el futur com a part d'una línia de recerca addicional.

5.3 Iteracions

En aquesta secció es detalla cada iteració, les dates en les quals s'executarà cadascuna, els objectius que es vol obtenir, l'iteration backlog de tasques a realitzar, els recursos dels quals es disposa i els entregables que s'han d'obtenir al final de cada iteració.

Iteració 1

Data d'inici: 24/03/2024

Data de finalització: 07/04/2024

Objectius: desenvolupar una primera versió dels quioscos que implementi el programari de forma local que després s'utilitzarà conjuntament amb l'API.

Iteration backlog:

T01 - Dissenyar el quiosc que consisteixi en una botonera, una pantalla tàctil i un ordinador.

T02 - Desenvolupar el programari del quiosc que funcioni en local.

T03 - Incorporar els botons i la pantalla tàctil al quiosc.

T04 - Desenvolupar la interfície de la pantalla tàctil que mostri les preguntes de les enquestes.

T05 - Implementar la base de dades MySQL.

Recursos: Una Raspberry Pi 4 Model B i quatre botons retroil·luminats. Allotjament en un servidor

Entregables: Prototip d'un dispositiu quiosc funcional capaç de treballar de forma local que mostri les preguntes i emmagatzemi les respostes. Prototip d'una base de dades amb les taules requerides.

Iteración 2

Data d'inici: 07/04/2024

Data de finalització: 28/04/2024

Objectius: Desenvolupar l'API REST amb les funcionalitats necessàries per a gestionar els quioscos i les enquestes.

Iteration backlog:

T06 - Desenvolupar l'API REST amb Node.js.

Recursos: Allotjament en un servidor.

Entregables: Prototip d'una API REST amb les funcions pertinents requerides.

Informe de progrés I.

Iteración 3

Data d'inici: 28/04/2024

Data de finalització: 12/05/2024

Objectius: Obtenir un prototip funcional entre un dispositiu i l'API REST, de tal forma que hàgim aconseguit els dos primers objectius del projecte (subsecció 1.1).

Iteration backlog:

T07 - Implementar al programari dels quioscos una connexió amb l'API.

Recursos: Allotjament en un servidor.

Iteració 4

Data d'inici: 12/05/2024

Data de finalització: 02/06/2024

Objectius: Obtenir un prototip funcional capaç de gestionar les enquestes actives i els quioscos a través d'una interfície web tal com marca el tercer objectiu del projecte (subsecció 1.1).

Iteration backlog:

T08 - Desenvolupar la interfície web utilitzant Vue.js

Entregables: Prototip d'interfície web capaç de comunicar-se amb l'API REST, gestionar quioscos i enquestes.

Iteració 5

Data d'inici: 02/06/2024

Data de finalització: 16/06/20

Objectius: Assegurar la robustesa del sistema i la integritat de les dades, dissenyar el prototip dels quioscos.

Iteration backlog:

T19 - Dissenyar el prototip dels quioscos.

T21 - proves d'integració del sistema.

Entregables: Informe de progrés II.

REST i la interfície web funcionant correctament. Proposta Informe final.

5.4 Diagrama de Gantt

Tal com es mostra en la figura 2, el diagrama de Gantt és una eina molt útil per visualitzar les relacions i dependències entre les tasques durant el desenvolupament d'un projecte. Aquest diagrama permet veure clarament quines tasques han de completar-se abans que comencin altres activitats, i quines tasques poden realitzar-se en paral·lel.

6 DESENVOLUPAMENT

En aquesta secció, es detallarà el desenvolupament que s'ha dut a terme en el projecte, des del disseny dels diferents components del sistema necessaris per al projecte fins a la implementació d'aquests. Per a la implementació del projecte, s'ha utilitzat GitHub com a plataforma per al control de versions. Aquesta eina ha estat clau per emmagatzemar el codi i mantenir un històric de les modificacions realitzades en el codi font.

6.1 Dispositius

6.1.1 Anàlisi

Es va desenvolupar un prototip inicial del dispositiu quiosc, com es mostra a la figura 3, que consisteix en una Raspberry Pi 4 amb el sistema Raspbian, una pantalla tàctil connectada a aquesta, i un únic botó controlat per la GPIO de la Raspberry. A més, es va incloure una font d'alimentació per proporcionar el voltatge necessari als LEDs dels botons i altres components, com un ventilador.

Pel que fa al desenvolupament del programari del quiosc, s'ha utilitzat Python amb les llibreries TKinter i CustomTKinter per a les interfícies gràfiques.

Les eines emprades per a la connexió remota amb la Raspberry han estat PuTTY i VNC.

6.1.2 Funcionament

El programa del dispositiu realitza crides HTTP a l'API on es troben les enquestes associades a cada quiosc, enviant l'adreça MAC del dispositiu anonimizada, rep per part

de l'API un JWT i la pròxima enquesta, si n'hi ha, per a aquell quiosc. Un cop te l'enquesta, el programa s'ocupa de mostrar-la en bucle a través de la pantalla. Quan rep el senyal del botó a través del pin GPIO, envia la resposta a l'API i avança a la següent pregunta fins a completar-les totes, reiniciant-se després.

6.2 Base de dades

6.2.1 Anàlisi

Es va dissenyar i implementar una base de dades composta per 6 taules, tal com es mostra en la figura 4, utilitzant SQL. Aquesta base de dades té la capacitat d'emmagatzemar la informació dels usuaris administradors del sistema amb les contrasenyes encriptades, les enquestes amb les preguntes en tres idiomes, els quioscos assignats, així com les respostes a les enquestes, juntament amb altres dades com l'edat i el gènere.

Per a la implementació del projecte, es va utilitzar un servidor local de MySQL.

6.3 Middleware

6.3.1 Anàlisi

El disseny de l'API REST va iniciar amb el diagrama de classes corresponent, com es pot veure a la figura 9. Aquest diagrama va ser fonamental per definir les funcions i crides requerides per a l'API. A mesura que el desenvolupament va anar avançant, van sorgir nous recursos necessaris que es van anar afegint al diagrama.

La implementació es va realitzar amb Node.js, inicialment en un entorn local, escoltant pel port 3000. S'han desenvolupat les funcions i crides amb les consultes pertinents, i aquestes crides van ser provades amb l'eina Insomnia.

Per garantir una comunicació segura i eficient, la comunicació amb l'API per a rebre i enviar dades, tant d'enquestes com d'usuari, es realitza mitjançant l'encriptació amb JWT (JSON Web Tokens)[6], assegurant així la confidencialitat i integritat de la informació transmesa. Aquesta implementació proporciona una capa addicional de seguretat, protegint les dades sensibles i assegurant que només els usuaris autoritzats tinguin accés a la informació i funcionalitats de l'administració.

6.3.2 Funcionament

En primer lloc, el middleware intercepta les peticions HTTP. Aquest component verifica la presència i la validesa dels tokens d'autorització continguts en els headers de les peticions. Les peticions, després de ser autenticades i autoritzades, són dirigides a través de les rutes, que actuen com a conductes. Les rutes, a través de la informació proporcionada en la petició, determinen quin servei o funció s'ha de cridar per manejar la tasca sol·licitada. Abans de transferir el control als serveis, les rutes poden realitzar comprovacions addicionals, com validar paràmetres de la petició o dur a terme altres tasques de preprocessament.

Un cop les rutes decideixen el destinatari de la petició, el control és passat als serveis. Aquests serveis executen la

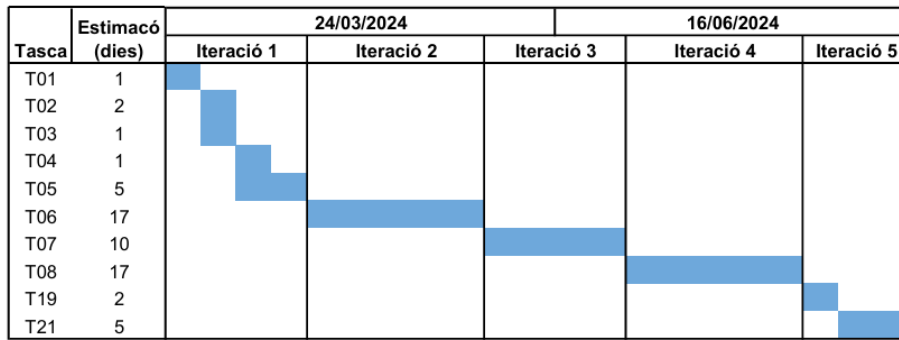


Fig. 2: Diagrama de Gantt



Fig. 3: Prototip quiosc



Fig. 4: Esquema base de dades

lògica del sistema, efectuant operacions com consultes SQL o altres tasques específiques de l'aplicació.

Quan s'hagin completat les tasques, els serveis retornen els resultats o les respostes corresponents a les peticions HTTP. Aquestes respostes poden incloure dades o missatges d'estat per indicar l'èxit o el fracàs de l'operació.

6.4 Interfície Web

6.4.1 Anàlisi

Es va dissenyar i implementar una interfície web utilitzant Vue.js per als administradors amb diverses funcionalitats. Aquesta interfície permet als administradors veure tant les enquestes obertes com les tancades, així com les respostes de les enquestes tancades, que es presenten en diagrames circulars o de barres per a una millor comprensió visual.

A més, la interfície ofereix la possibilitat d'afegir noves enquestes, amb preguntes de tipus booleanes o multivalors, segons les necessitats. Les noves enquestes es poden associar als quioscos que es desitgi, facilitant així la gestió i distribució de les mateixes.

6.4.2 Funcionament

La interfície web comença carregant el component d'inici de sessió, quan l'usuari intenta iniciar sessió, el programa envia una petició HTTP a l'API amb les credencials utilitzant Axios. Si les credencials són vàlides, el servidor respon amb un token d'autorització, que s'emmagatzema localment a la sessió amb el mòdul authService.

Les rutes de l'aplicació, gestionades des d'un fitxer encaïnador, dirigeixen l'usuari al primer component, una vista que mostra les enquestes actives i les últimes enquestes tancades. Aquí, l'usuari pot crear noves enquestes o revisar les tancades, amb la possibilitat de veure les respostes en un quadre de control senzill.

Totes les peticions HTTP passen a través d'una mateixa classe que ofereix serveis, utilitzant el mòdul Axios per comunicar-se amb l'API, això centralitza el maneig de les peticions i les respostes HTTP a tota l'aplicació.

7 RESULTATS

En aquesta secció es redactaran els resultats obtinguts durant el desenvolupament del projecte, s'inclouran anàlisis dels objectius aconseguits i una avaluació de l'efectivitat de les metodologies aplicades.

S'han realitzat diverses pràctiques per assegurar que la base de dades i l'API puguin gestionar adequadament la càrrega esperada i oferir un rendiment òptim. Primer, s'han utilitzat dades realistes que imiten l'ús real, incloent-hi activitats com inici de sessió, connexió amb els quioscos, descarregar enquestes i pujar respostes. S'han definit objectius de rendiment com temps de resposta i taxa d'error, permetent

mesurar i optimitzar aquests aspectes. També s'han provat diferents operacions de la base de dades (lectura i escriptura) i s'han simulat períodes de màxima i baixa demanda per obtenir una comprensió exhaustiva del rendiment en diverses condicions.

A més, s'ha testejat l'API amb el programa JMeter i s'ha registrat l'estat de la base de dades amb MySQL Workbench. En aquestes proves, s'ha simulat la connexió de 10 quioscos a l'API per rebre el JWT, rebre l'enquesta oberta i pujar les respostes, tots al mateix temps cada 10 segons durant 10 iteracions, tal com es mostra en la gràfica de la figura 5. Tot això s'ha realitzat en un entorn local amb un maquinari insuficient, però tenint en compte un entorn realista, la base de dades ha reaccionat bastant bé. En un primer testeig, es va detectar un error en el codi que impedia que es tanquessin les connexions, el qual va ser corregit per evitar fugues de memòria i millorar l'eficiència. Aquestes pràctiques han estat essencials per identificar problemes de rendiment, ajustar configuracions i assegurar l'eficiència de la base de dades en un entorn de producció real.

S'ha assolit el desenvolupament de la interfície web funcional que permet una sèrie d'operacions clau per a la gestió d'enquestes. Aquesta interfície permet autenticar els usuaris registrats a la base de dades, garantint així la seguretat i la integritat de l'accés al sistema, com es mostra en la figura 6. A més, facilita la creació de noves enquestes, on els administradors poden afegir quioscos i definir les preguntes necessàries de manera senzilla i eficient, figura 7. Un altre aspecte és la capacitat per visualitzar els resultats de les enquestes tancades, oferint una visió clara i detallada de les dades recollides, figura 8. Aquestes funcionalitats conjuntes asseguren una gestió integral i eficient de les enquestes, contribuint a una millor captació i anàlisi de l'opinió dels assistents.

8 CONCLUSIONS

Qiosk té com a objectiu ser una plataforma per a rebre una retroalimentació de qualitat que permeti millorar els esdeveniments acadèmics oferts. A través de Qiosk, es busca obtenir opinions i suggeriments valuosos dels participants, amb la finalitat de perfeccionar i adaptar les activitats acadèmiques a les necessitats i preferències dels usuaris.

La metodologia Scrumban ha aportat flexibilitat i adaptabilitat, permetent ajustos ràpids als canvis de requisits, i ha facilitat la gestió visual del treball amb taulers Kanban.

Per adversitats no contemplades en la planificació, no s'ha pogut assolir tots els requisits, com la detecció de gènere i edat. Aquests requisits s'han tingut en compte en el desenvolupament i podrien ser afegits posteriorment per tal d'obrir una nova línia de recerca. La integració futura d'aquestes funcionalitats permetria enriquir encara més el projecte, oferint una anàlisi més detallada i personalitzada que podria aportar nous coneixements i perspectives en el camp d'estudi.

Un cop finalitzat el projecte, hi ha algunes coses que es canviarien o es farien de manera diferent. En la part del servidor, tant Node.js com Vue.js s'adapten molt bé a les necessitats del projecte, però MySQL no tant. Les dades s'han d'introduir a través de consultes, que poden ser excessivament llargues i empitjorar el rendiment del sistema. Amb MongoDB, es podrien rebre les dades en format JSON

dels quioscos i introduir-les directament a la base de dades, oferint també una certa flexibilitat. MongoDB permet emmagatzemar documents en un format semblant a JSON, cosa que facilita la gestió de dades i l'escalabilitat d'aquestes, millorant el rendiment global del sistema.

Pel que fa als dispositius, Python no sembla el més adequat per a les interfícies gràfiques. Amb el coneixement adquirit durant la realització del projecte, el programari dels dispositius, es faria amb Dart. Dart, juntament amb el framework Flutter, permet desenvolupar aplicacions amb interfícies gràfiques modernes i responsives de manera eficient.[7]

Per finalitzar, és crucial implementar mesures de seguretat per a les peticions cap al servidor. Utilitzar TLS (Transport Layer Security) és essencial per encriptar les dades en trànsit, assegurant que la comunicació entre els dispositius clients i el servidor sigui segura i protegida contra interceptacions no autoritzades.

REFERÈNCIES

- [1] Wiersma, W. (2013) 'The Validity of Surveys: Online and Offline', <http://www.papers.wybowiersma.net/>. Oxford Internet Institute. Available at: http://www.papers.wybowiersma.net/abstracts/Wiersma_Wybo_The_validity_of_surveys_online_and_offline.pdf (Accessed: 2024).
- [2] Smiley button feedback - airport bathroom survey kiosk 2024 (2023) SurveyStance. Available at: <https://www.surveystance.com/airport-feedback-smiley-survey/> (Accessed: 09 March 2024).
- [3] FeedbackNow (2015) Real-time Passenger Feedback, Customer Satisfaction, Passenger Experience & Excellence in Service Quality at every Passenger Touch Point, Airport Suppliers. Available at: <https://www.airport-suppliers.com/supplier/feedbacknow/> (Accessed: 09 March 2024).
- [4] neXenio (2019) NeXenio/feedback-kiosk: A simple tool for realtime feedback gathering, GitHub. Available at: <https://github.com/neXenio/Feedback-Kiosk> (Accessed: 09 March 2024).
- [5] Sawant, M. (2019) SMAHESH29/gender-and-age-detection: A python project which can detect gender and age using opencv of the person (face) in a picture or through webcam., GitHub. Available at: <https://github.com/smahesh29/Gender-and-Age-Detection> (Accessed: 09 March 2024).
- [6] M. Jones, J. Bradley, and N. Sakimura. 2015. RFC 7519: JSON Web Token (JWT). RFC Editor, USA.
- [7] CloudDevs. (2023, October 13). DART for Desktop Applications: Creating Powerful GUIs. <https://cloudd devs.com/dart/desktop-applications/>

APÈNDIX

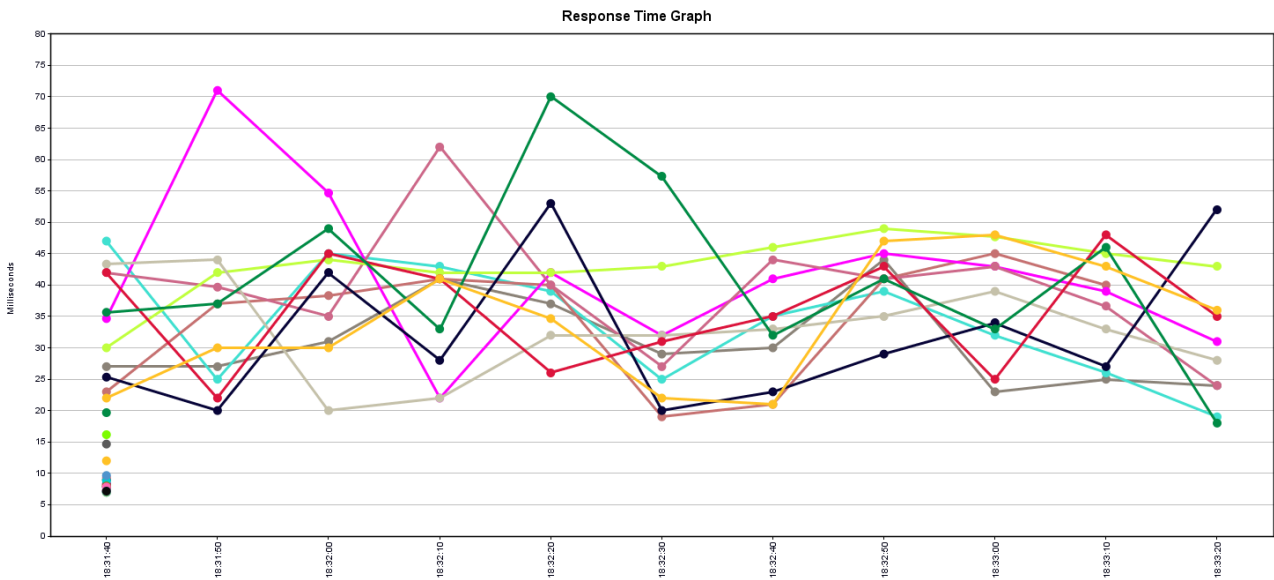


Fig. 5: Temps de resposta de la base de dades, 10 quioscos simulats amb JMeter demanant primer l'enquesta i després enviant 10 respostes

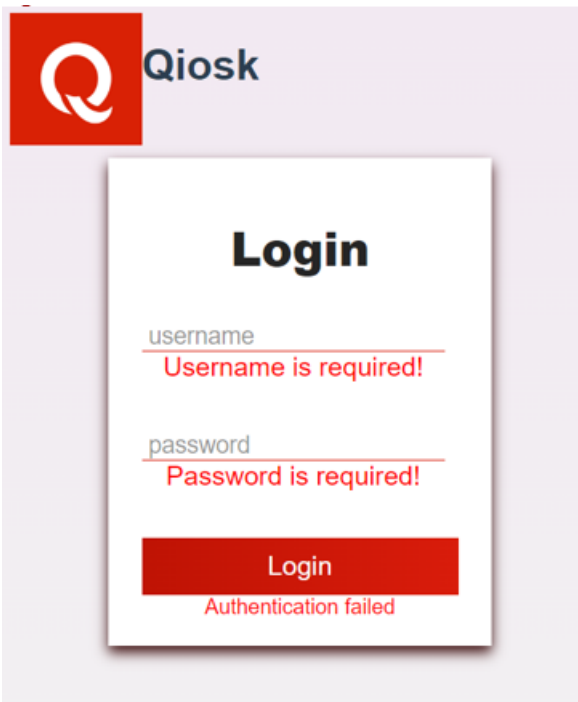


Fig. 6: Inici de sessió interfície web

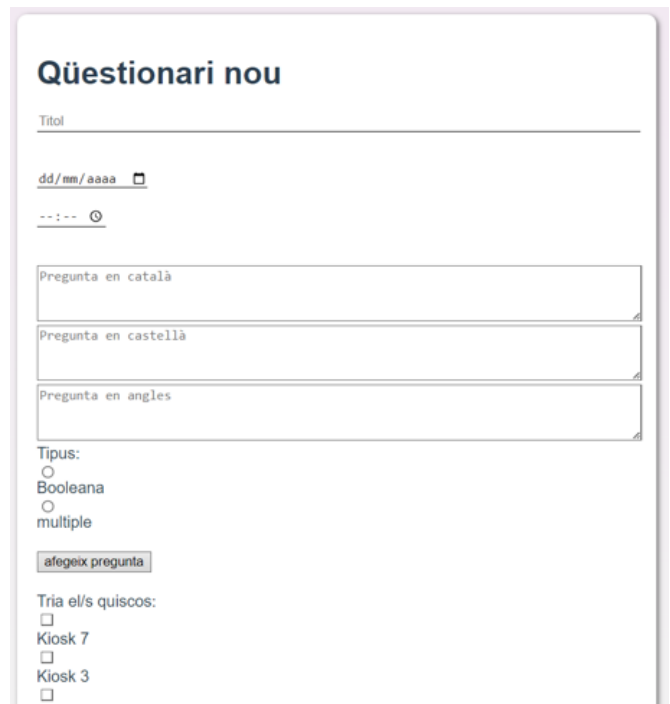


Fig. 7: Formulari de creació d'enquesta nova



Fig. 8: Visualització dels resultats d'una enquesta

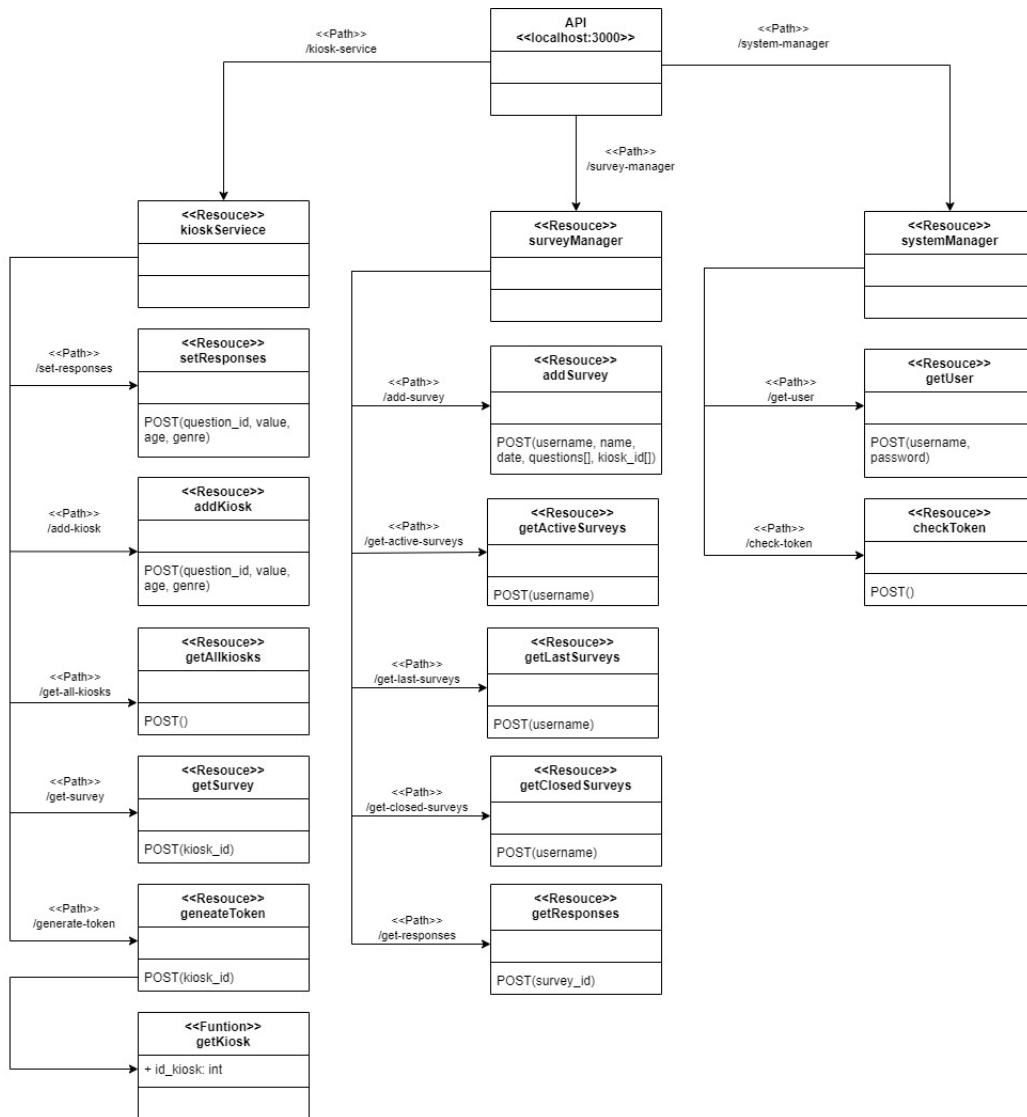


Fig. 9: Diagrama de classes middleware