

Aplicació per a l'optimització de la profilaxi antibiòtica a l'Hospital Trueta

Joan Paneque Domingo Emma Cardosa Martínez
Aniol Moreno Batlle Adrià Moya Duran
Alex Escribano Reina Luis Arauz Ortiz

Maig de 2024

Resum

Aquest document descriu el desenvolupament d'una aplicació destinada a optimitzar la profilaxi antibiòtica¹ a l'Hospital Trueta. Aquesta optimització permet estalviar temps de treball dels anestesistes, valorat en aproximadament 35.700€ anuals. L'objectiu de l'aplicació és proporcionar una ajuda als anestesidòlegs en l'elecció correcta dels antibiòtics, i assegurarà una millor eficiència i efectivitat en la presa de decisions clíniques amb profilaxi antibiòtica, reduint així els errors en la selecció d'antibiòtics. Aquesta ha estat desenvolupada per un equip d'estudiants de desenvolupament web sota la metodologia Agile/Scrum² i consisteix en una interfície intuïtiva pels usuaris, compatible amb els sistemes hospitalaris, i que garanteixi la seguretat i escalabilitat del sistema. Aquest document detalla el procés de desenvolupament, l'arquitectura del sistema, els requisits del sistema, i els resultats de les proves i la validació. A més, es parla sobre el desplegament de l'aplicació i es proposen futures millores.

¹Profilaxi antibiòtica: Administració d'antibiòtics preventius per protegir contra els gèrmens que més sovint causen infeccions en els diferents tipus de cirurgia.

²Agile/Scrum: Marc de treball per al desenvolupament de projectes que s'enfoca en la iteració contínua de petits increments de treball.

Taula de continguts

1	Desenvolupadors del Projecte	4
2	Introducció	5
3	Objectius del Projecte	6
4	Requisits del Sistema	7
4.1	Requisits d'Usuari	7
4.2	Requisits Tècnics	7
5	Arquitectura del Sistema	8
5.1	Visió General	8
5.2	Diagrama de l'Arquitectura del Sistema	8
5.3	Components Principals	9
5.4	Interacció dels Components	11
5.5	Tecnologies Utilitzades	11
6	Desenvolupament de l'Aplicació	12
6.1	Metodologia de Desenvolupament	12
6.1.1	Planificació	12
6.1.2	Disseny	13
6.1.3	Implementació	14
6.2	Captures de Pantalla i Funcionalitats	15
6.2.1	Pantalla d'Inici de Sessió	15
6.2.2	Pantalla Principal	16
6.2.3	Selecció de tipus de cirurgia	18
6.2.4	Selecció de condició de salut	19
6.2.5	Pantalles de creació i modificació de continguts	22
6.2.6	Pantalles de funcionalitats d'administrador	25
6.3	Fases del Desenvolupament	28
6.3.1	Fase de Proves	28
6.3.2	Fase de Desplegament de pre-producció	29
7	Prospectiva	32
8	Conclusió	32
A	Apèndix	34
A.1	Glossari de Termes	34
A.2	Estructura de Directoris	34
A.3	Configuració de l'Entorn de Desenvolupament	35
A.4	Scripts de Comandes Utilitzats	36

Índex de figures

1	Diagrama de l'arquitectura del sistema	8
2	Logotip Vue.js	9
3	Logotip Inertia.js	9
4	Logotip Laravel	9
5	Logotip MySQL	10
6	Contenedor docker de l'aplicació	10
7	Logotip nginx	11
8	20 de febrer de 2024 - Reunió d'equip	12
9	21 de febrer de 2024 - Planificant el projecte	13
10	Esbós de l'estructura de dades inicial del sistema	14
11	Estructura de dades final del sistema	14
12	Pantalla d'inici de sessió en MacBook Air	15
13	iPhone 13	15
14	Samsung Galaxy S21	15
15	Pantalla Principal MacBook Air	16
16	Pantalla Principal iPhone 13	17
17	Pantalla de selecció de tipus de cirurgia	18
18	Selecció de condició de salut (Al·lèrgia a Penicil·lina)	19
19	Resultats de la dosificació d'un recanvi valvular en cas d'al·lèrgia a penicil·lina	20
20	Informació d'un antibiòtic	21
21	Creació cirurgia	22
22	Edició cirurgia	22
23	Creació tipus de cirurgia	23
24	Edició tipus de cirurgia	23
25	Creació de condicions	24
26	Edició de condicions	24
27	Control d'altres	25
28	Control de baixes	25
29	Llista de registres	26
30	Registre individual	26
31	Edició d'un perfil com a administrador	27
32	Mostra de proves unitàries	28
33	Mostra de proves d'extrem a extrem amb Selenium IDE	29
34	Tros de script de desplegament	30
35	Claus secretes de repositori	31

1 Desenvolupadors del Projecte

Aquest projecte ha estat desenvolupat per un equip d'estudiants de desenvolupament web, cadascun dels quals ha contribuït a diverses àrees del desenvolupament de l'aplicació. Els membres de l'equip són:

- **Joan Paneque Domingo**
Scrum Master³ i organitzador del projecte. Ha planificat una estructura òptima per a la base de dades, i ha perfeccionat el disseny i polit l'aplicació tant en frontend⁴ com en backendBackend: . A més, s'ha assegurat que el producte es pugui entregar a temps i ha planificat reunions setmanals amb l'Hospital Trueta.
- **Emma Cardosa Martinez**
Disseny d'una interfície d'usuari intuïtiva i la millora de l'experiència d'usuari. Enfocament en la creació d'un disseny funcional i atractiu per optimitzar la usabilitat i la satisfacció dels usuaris sanitaris.
- **Luis Arauz Ortiz**
Contribucions al desenvolupament del frontend i backend.
- **Adrià Moya Duran**
Administració del backend, especialment en la gestió i administració d'usuaris. Ha desenvolupat solucions per al control d'accés, gestió de permisos i manteniment de bases de dades d'usuaris, assegurant una experiència segura i eficient.
- **Alex Escribano Reina**
Maneig de la base de dades, incloent-hi la migració de dades de la documentació de profilaxi de l'Hospital Trueta a la base de dades.
- **Aniol Moreno Batlle**
Contribucions al desenvolupament del frontend i backend.

Cada membre de l'equip ha aportat el seu coneixement i habilitats en diverses àrees del projecte, garantint un enfocament col·laboratiu i integral per al desenvolupament de l'aplicació. Aquest treball en equip assegura una solució robusta i ben integrada que compleix la necessitat de l'hospital.

³Scrum Master: És un facilitador en equips Agile/Scrum, que ajuda a l'equip a seguir les pràctiques de Scrum, elimina obstacles, facilita les reunions i promou la comunicació efectiva per garantir l'eficiència i la qualitat del projecte.

⁴Frontend: Part visible d'una aplicació web o mòbil amb la qual l'usuari interactua, utilitzant HTML, CSS i JavaScript per crear dissenys i elements interactius.

2 Introducció

La profilaxi amb antibiòtics és una pràctica que prevé infeccions en l'entorn hospitalari durant el període postoperatori, contribuint a una millor recuperació dels serveis. En aquest context, sorgeix la necessitat dels anestesiològics de decidir quins antibiòtics s'han d'utilitzar d'una llista molt àmplia i sovint complexa de procediments quirúrgics, la qual cosa tendeix a convertir-se en un procés propens a errors. Aquest projecte neix de la necessitat de millorar el procés de profilaxi antibiòtica amb una eina tecnològica que ajudi a la presa de decisions clíniques.

En aquest document descrivim el desenvolupament d'una aplicació per a l'optimització d'antibiòtics en pacients de l'Hospital Trueta. Principalment, l'aplicació es desenvolupa per servir com a guia d'ajuda per als anestesiològics per prendre la decisió correcta d'antibiòtics i ser més eficients i eficaços en la presa de decisions clíniques. A més, una selecció adequada d'antibiòtics no només ajuda a reduir errors mèdics, sinó que també millora els resultats dels pacients reduint el risc d'infeccions i complicacions postoperatòries.

L'aplicació ha estat desenvolupada per un equip d'estudiants de desenvolupament web seguint la metodologia Agile/Scrum. Amb aquesta metodologia, la iteració contínua i l'adaptació ràpida als canvis de requisits asseguren que el producte final compleixi les necessitats reals dels usuaris. La interfície d'usuari dissenyada per a l'aplicació és essencialment intuïtiva i, per tant, fàcil d'operar pels usuaris. És compatible amb els sistemes hospitalaris existents i s'asseguren garanties de seguretat i escalabilitat.

Aquest document descriu tot el procés de desenvolupament de l'aplicació, des de la definició de l'arquitectura del sistema i els seus requisits tècnics i d'usuari fins a les proves i la validació. A més, es proporciona una descripció detallada de com es desplega l'aplicació, i es proposen algunes millores futures per continuar optimitzant la profilaxi antibiòtica a l'Hospital Trueta.

3 Objectius del Projecte

El projecte té com a objectiu principal desenvolupar una aplicació per millorar la profilaxi antibiòtica a l'Hospital Trueta, que es pot desglossar en els següents objectius específics:

- Assistir en l'elecció d'antibiòtics: Dissenyar una eina que pugui ajudar un anestesiòleg, basant-se en dades específiques i recomanacions de l'Hospital Trueta, a triar els antibiòtics més adequats per a la profilaxi.
- Augmentar l'eficiència i l'efectivitat: Millorar l'eficiència i l'efectivitat en el procés de presa de decisions clíniques sobre la profilaxi antibiòtica, ja que redueix el temps i la possibilitat d'error en el procés de selecció d'antibiòtics.
- Reducció d'errors: Reduir errors en la prescripció d'antibiòtics, de manera que un pacient rebi la medicació més adequada segons les recomanacions dels protocols desenvolupats, reduint el nombre d'infeccions postoperatòries i, per tant, assolint un bon resultat clínic.
- Interfície intuïtiva: Dissenyar una interfície que sigui intuïtiva i fàcil d'utilitzar per simplificar-ne l'ús als sanitaris.
- Seguretat i escalabilitat: La informació dels antibiòtics ha d'estar protegida d'avant d'una intrusió no autoritzada, i el sistema d'informació desenvolupat ha de ser escalable a futures necessitats i expansions de l'hospital.

Amb aquests objectius, l'aplicació no només optimitzarà el procés de profilaxi antibiòtica, sinó que també contribuirà a millorar la qualitat de l'atenció mèdica proporcionada als pacients de l'Hospital Trueta.

4 Requisits del Sistema

4.1 Requisits d'Usuari

Els requisits d'usuari es centren en proporcionar una interfície intuïtiva, accessible i fàcil d'utilitzar per als professionals sanitaris. A continuació, es detallen els requisits específics d'usuari:

- **Interfície intuïtiva:** L'aplicació ha de tenir una interfície fàcil de comprendre i d'utilitzar, fins i tot per als usuaris amb poca experiència tècnica.
- **Facilitat d'ús:** Les funcionalitats han de ser accessibles amb un mínim nombre de clics i passos.
- **Accessibilitat:** L'aplicació ha de complir amb els estàndards d'accessibilitat web per assegurar que tots els usuaris, incloent aquells amb discapacitats, puguin utilitzar-la sense problemes.

4.2 Requisits Tècnics

Els requisits tècnics garanteixen que l'aplicació funcioni correctament en l'entorn hospitalari, assegurant compatibilitat, seguretat i escalabilitat. A continuació, es presenten els requisits tècnics específics:

- **Seguretat de les dades:**
 - Xifrat de dades tant en trànsit com en repòs.
 - Autenticació i autorització robusta per a l'accés als diferents nivells d'informació.
 - Rols per gestionar l'accés a la informació.
- **Escalabilitat:**
 - Ha de permetre fàcilment l'addició de noves funcionalitats en el futur.
- **Entorn de Desenvolupament i Producció:**
 - L'aplicació està desenvolupada utilitzant Laravel per al backend i Vue.js amb Inertia.js per al frontend.
 - La base de dades utilitzada és MySQL.
 - Docker s'utilitza per a la contenedorització de l'aplicació, permetent una gestió més eficient dels entorns de desenvolupament, prova i producció.
- **Requisits del servidor:**
 - Sistema operatiu: Linux (preferiblement distribucions com Ubuntu o CentOS).

- Servidor web: Nginx o Apache.
- PHP: Versió 8.2 o superior.
- MySQL: Versió 8.0 o superior.
- Docker: Versió 26.00 o superior.
- Composer: Per a la gestió de dependències PHP.
- Node.js i npm: Per a la gestió de paquets i compilació del frontend.

5 Arquitectura del Sistema

5.1 Visió General

L'arquitectura de l'aplicació per a l'optimització de la profilaxi antibiòtica a l'Hospital Trueta està dissenyada per garantir una alta disponibilitat, seguretat, escalabilitat i facilitat de manteniment. El sistema té diversos components interconnectats que treballen per proporcionar una solució robusta i fiable.

5.2 Diagrama de l'Arquitectura del Sistema

El següent diagrama mostra la vista general de l'arquitectura del sistema:

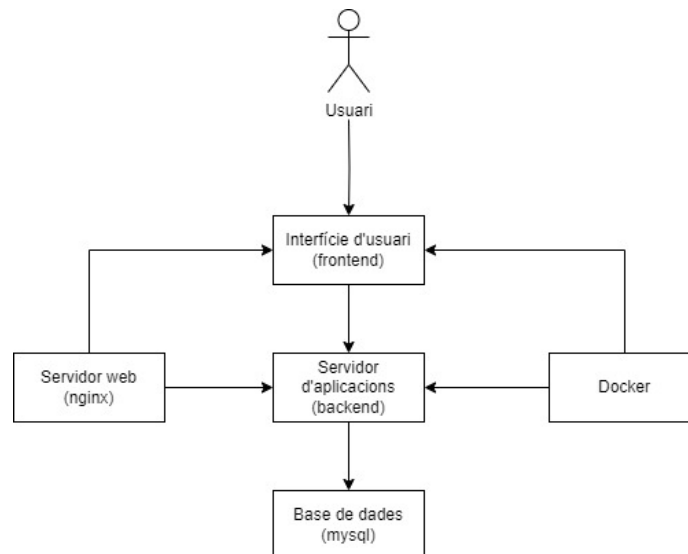


Figura 1: Diagrama de l'arquitectura del sistema

5.3 Components Principals

- **Interfície d'Usuari (Frontend):** Desenvolupada amb Vue.js i Inertia.js, proporciona una interfície intuïtiva i responsiva per als usuaris. Vue.js s'utilitza per gestionar la lògica de la interfície, mentre que Inertia.js facilita la comunicació amb el backend sense necessitat d'utilitzar APIs REST tradicionals.



Figura 2: Logotip Vue.js



Figura 3: Logotip Inertia.js

- **Servidor d'Aplicacions (Backend):** Construït amb Laravel, un framework PHP que permet el desenvolupament ràpid i segur d'aplicacions web. Laravel s'encarrega de la lògica de negoci, la gestió de bases de dades i la comunicació amb el frontend.

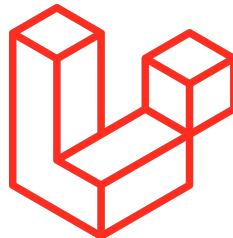


Figura 4: Logotip Laravel

- **Base de Dades:** MySQL és la base de dades relacional utilitzada per emmagatzemar totes les dades necessàries, incloent-hi informació sobre usuaris, protocols antibiòtics i registres d'activitat. Es garanteix la integritat i la seguretat de les dades mitjançant pràctiques de xifrat i gestió de permisos.



Figura 5: Logotip MySQL

- **Sistema de Contenidors (Docker):** Docker s'utilitza per a la contenedorització de l'aplicació, permetent una gestió eficient i consistent dels entorns de desenvolupament, prova i producció. Cada component de l'aplicació s'executa en un contenidor separat, facilitant l'escalabilitat i el desplegament.

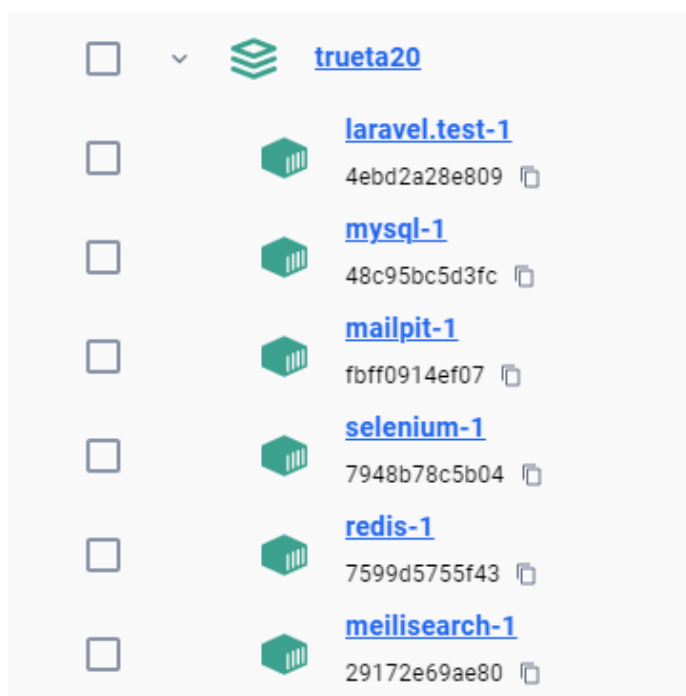


Figura 6: Contenidor docker de l'aplicació

- **Servidor Web:** Nginx s'utilitza com a servidor web per gestionar les sol·licituds dels usuaris i dirigir-les al backend adequat. Nginx és conegut pel seu alt rendiment i eficiència en la gestió de connexions simultànies.



Figura 7: Logotip nginx

5.4 Interacció dels Components

Els components interactuen entre ells de la següent manera:

- **Usuari → Interfície d'Usuari:** Els usuaris accedeixen a l'aplicació a través d'un navegador web, interactuant amb la interfície d'usuari desenvolupada amb Vue.js i Inertia.js.
- **Interfície d'Usuari → Servidor d'Aplicacions:** Les accions dels usuaris a la interfície d'usuari generen sol·licituds HTTP que són enviades al servidor d'aplicacions Laravel.
- **Servidor d'Aplicacions → Base de Dades:** Laravel processa les sol·licituds i, si cal, accedeix a la base de dades MySQL per obtenir o emmagatzemar informació.
- **Servidor d'Aplicacions → Interfície d'Usuari:** Les respostes del servidor d'aplicacions són enviades de tornada a la interfície d'usuari per actualitzar la vista de l'usuari.
- **Contenidors Docker:** Tots els components estan empaquetats en contenidors Docker, la qual cosa permet desplegar l'aplicació de manera consistent en diferents entorns.

5.5 Tecnologies Utilitzades

- **Llenguatges de Programació:** PHP (per a Laravel), JavaScript (per a Vue.js i Inertia.js).
- **Bases de Dades:** MySQL.
- **Frameworks i Llibreries:** Laravel, Vue.js, Inertia.js.
- **Contenidors:** Docker.

- **Altres Eines:** Composer (per a la gestió de dependències PHP), npm (per a la gestió de paquets JavaScript), Node.js (per a la compilació del frontend).

6 Desenvolupament de l'Aplicació

6.1 Metodologia de Desenvolupament

L'aplicació ha estat desenvolupada seguint la metodologia Agile/Scrum. Aquesta metodologia es basa en iteracions curtes i retroacció contínua per assegurar que el producte final compleixi les necessitats del client. El procés de desenvolupament es va dividir en Sprints⁵ setmanals, amb reunions diàries (Daily Standups) per avaluar el progrés i identificar obstacles.

A més, es va realitzar una reunió setmanal amb els representants de l'Hospital Trueta per garantir que el projecte seguia el camí correcte. Aquestes reunions van permetre ajustar les prioritats, rebre retroacció valuosa i assegurar que les necessitats i expectatives de l'hospital estiguessin alineades amb el desenvolupament de l'aplicació.



Figura 8: 20 de febrer de 2024 - Reunió d'equip

6.1.1 Planificació

Durant la fase de planificació es van definir els objectius del projecte, els requisits d'usuari, i les funcionalitats essencials de l'aplicació. Es va crear un product backlog⁶ amb totes les tasques a realitzar, prioritzades segons la seva

⁵Sprints: Períodes curts i fixos de temps, típicament d'una a quatre setmanes, durant els quals un equip Agile/Scrum treballa per completar un conjunt específic de treballs predefinitos.

⁶Product backlog: Llistat de tasques que es volen realitzar en el desenvolupament d'un projecte.

importància i complexitat.

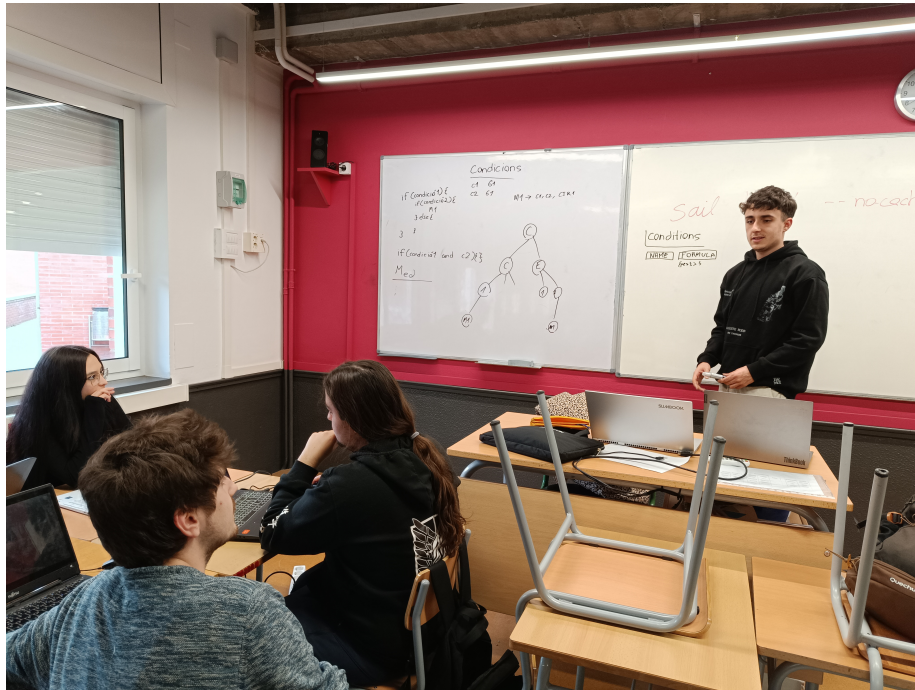


Figura 9: 21 de febrer de 2024 - Planificant el projecte

6.1.2 Disseny

En aquesta fase es va crear l'arquitectura del sistema i es van dissenyar els components principals de l'aplicació. Es van utilitzar eines de prototipat per dissenyar la interfície d'usuari i es van crear diagrames UML per representar la lògica del backend.

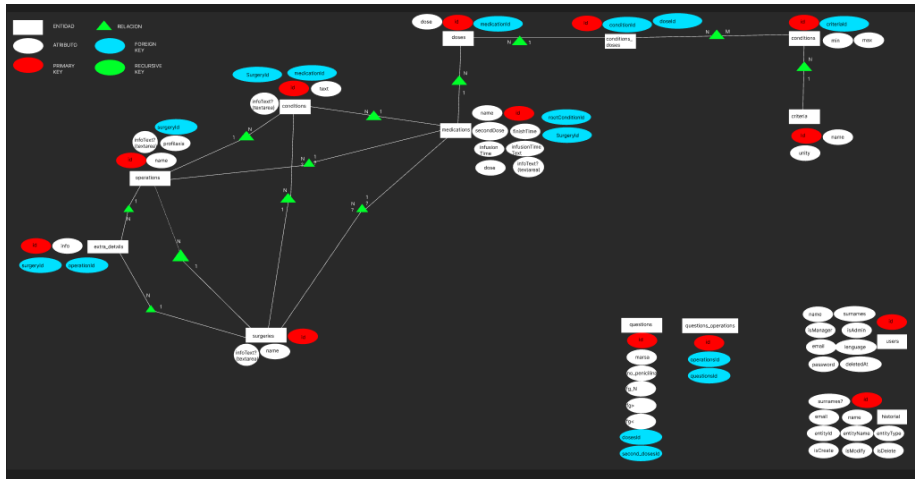


Figura 10: Esbós de l'estructura de dades inicial del sistema

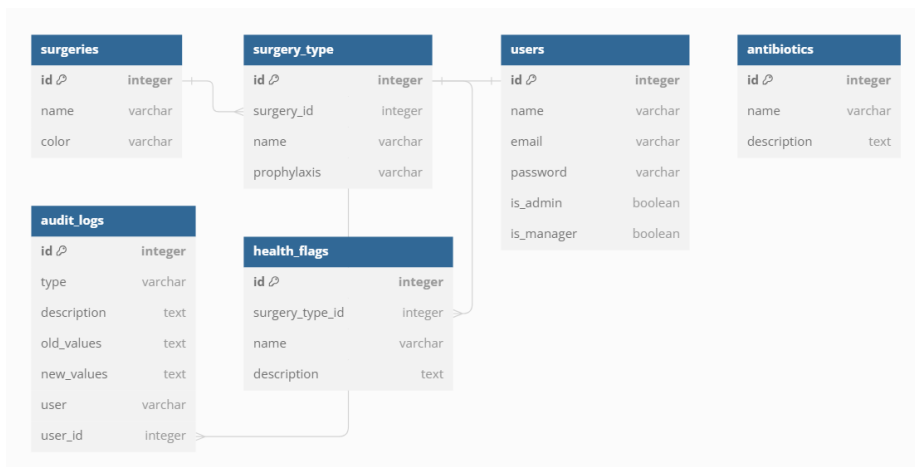


Figura 11: Estructura de dades final del sistema

6.1.3 Implementació

La implementació es va dur a terme seguint les bones pràctiques de desenvolupament de programari, amb un enfocament modular i reutilitzable. El frontend es va desenvolupar utilitzant Vue.js i Inertia.js, mentre que el backend es va implementar amb Laravel. Es va assegurar que el codi fos documentat adequadament per facilitar el manteniment futur.

6.2 Captures de Pantalla i Funcionalitats

A continuació es presenten algunes captures de pantalla de l'aplicació, acompanyades d'una descripció de les funcionalitats corresponents.

6.2.1 Pantalla d'Inici de Sessió

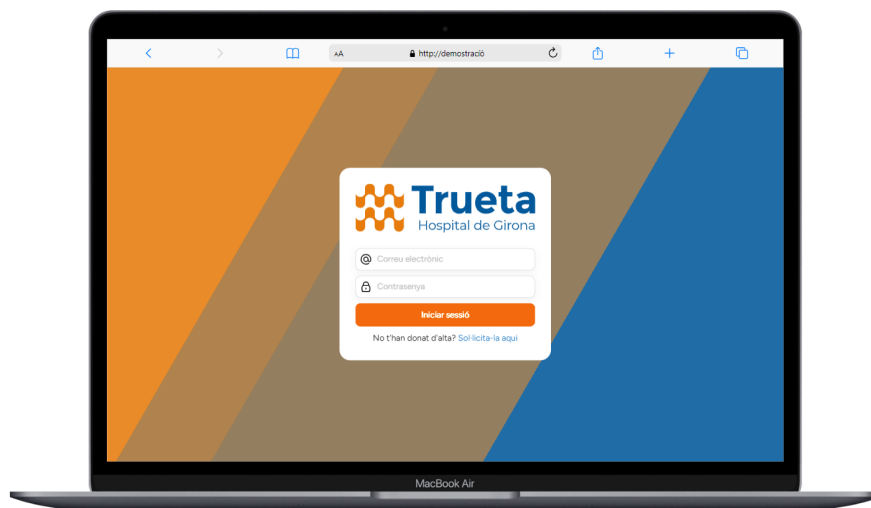


Figura 12: Pantalla d'inici de sessió en MacBook Air



Figura 13: iPhone 13

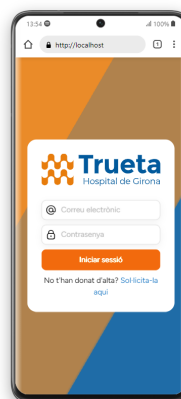


Figura 14: Samsung Galaxy S21

La pantalla d'inici de sessió permet als usuaris autenticar-se mitjançant les seves credencials. S'ha implementat un sistema d'autenticació segur amb gestió de sessions.

6.2.2 Pantalla Principal

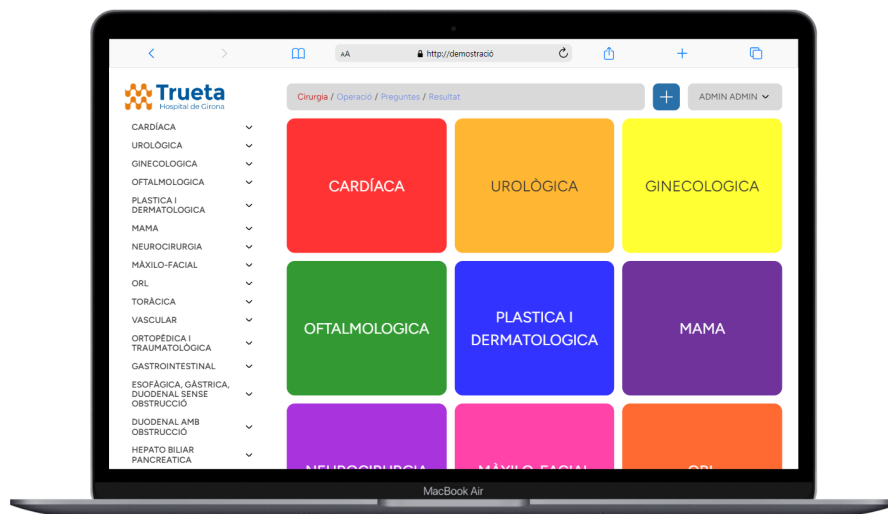


Figura 15: Pantalla Principal MacBook Air



Figura 16: Pantalla Principal iPhone 13

La pantalla principal ofereix accés ràpid a les cirurgies que es realitzen a l'Hospital Trueta, cadascuna amb un color diferent per fer-la més intuïtiva per als anestesistes.

6.2.3 Selecció de tipus de cirurgia



Figura 17: Pantalla de selecció de tipus de cirurgia

Aquesta pantalla permet als anestesiològs seleccionar el tipus de cirurgia que s'ha de realitzar, és a dir, el procediment quirúrgic. Els tipus de cirurgia segueixen el mateix color que la seva cirurgia pare per a que sigui molt més intuïtiu visualment.

6.2.4 Selecció de condició de salut



Figura 18: Selecció de condició de salut (Al·lèrgia a Penicil·lina)

En aquesta pantalla els anestesistes hauràn d'indicar quina o quines condicions s'ajusten més a la situació del pacient.



Figura 19: Resultats de la dosificació d'un recanvi valvular en cas d'al·lèrgia a penicil·lina

Si l'anestesta fa clic sobre un antibiòtic, s'obrirà una pantalla flotant amb la informació sobre aquest.

Tancar

VANCOMICINA

- Taula segons pes:
 - < 50 kg: 750 mg
 - 50-70 Kg: 1000 mg.
 - 70-85 Kg: 1250 mg.
 - 85-100 Kg: 1500 mg.
 - >100 kg: 1750 mg.
 - >120 kg: 2000 mg.
- Temps infusió: ≤ 1g: 60 min.
 - > 1g: 120 min.
- La infusió ha d'haver finalitzat dins els 60 minuts previs a la incisió de la pell, per lo que serà necessari iniciar-la almenys 2 hores abans de la incisió de la pell.
- No cal re dosificació intra operatòria.

Figura 20: Informació d'un antibiòtic

6.2.5 Pantalles de creació i modificació de continguts



Figura 21: Creació cirurgia



Figura 22: Edició cirurgia

Com es pot observar, es poden crear i editar cirurgies i l'aplicació ens permet canviar el nom i color d'aquesta.



Figura 23: Creació tipus de cirurgia Figura 24: Edició tipus de cirurgia

La creació i edició de tipus de cirurgies és molt semblant a la creació de cirurgies, amb la diferència que aquestes adapten el color del node pare i demanen si aquest tipus de cirurgia requereix de profilaxi antibiòtica.



Figura 25: Creació de condicions

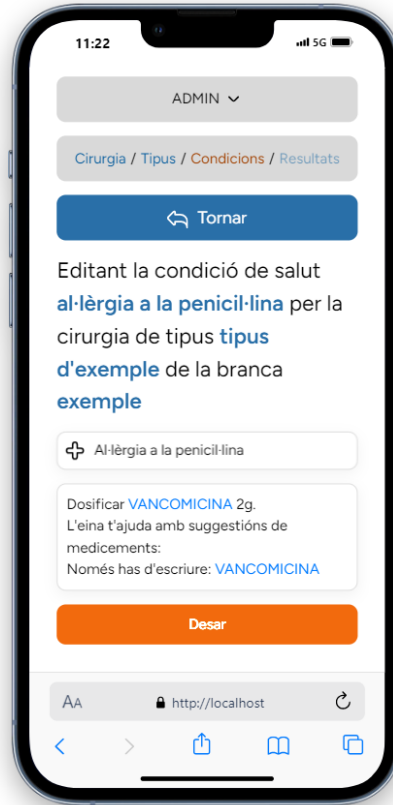


Figura 26: Edició de condicions

Aquestes pàgines permeten crear i editar les condicions d'un tipus de cirurgia. En aquestes pàgines, es defineixen el nom i una descripció amb els antibiòtics i les dosis. Aquests apartats utilitzen un sistema avançat de reconeixement automàtic d'antibiòtics, i a mesura que es va escrivint, el programa autocompleta el text amb suggeriments. Després de fer diverses proves, hem comprovat que encerta la major part del temps, reduint així el temps d'escriptura.

6.2.6 Pantalles de funcionalitats d'administrador

Control d'altres i baixes d'usuaris:

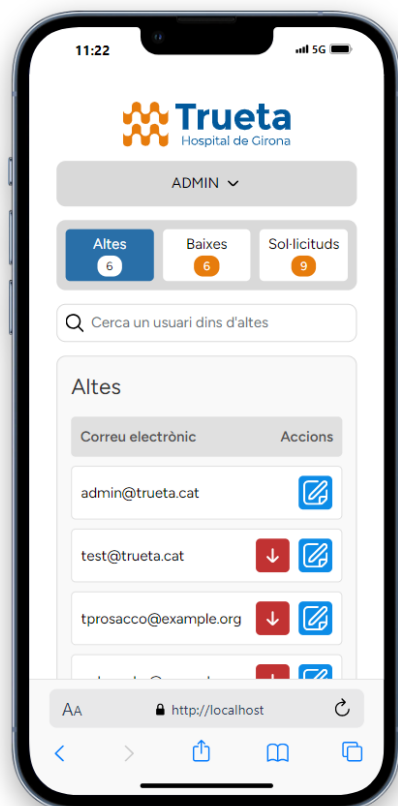


Figura 27: Control d'altres

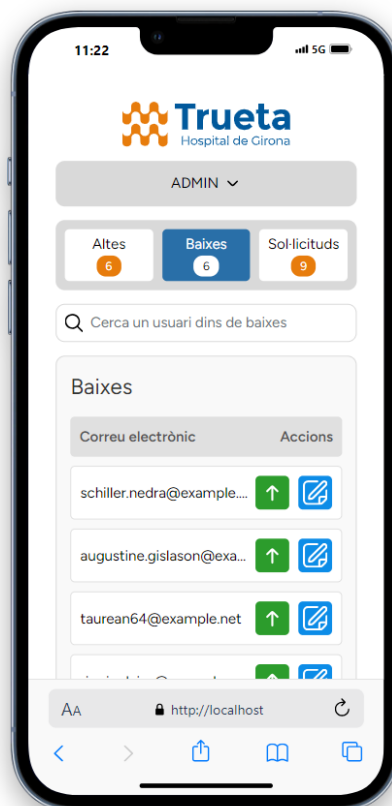


Figura 28: Control de baixes

Registres d'auditoria:

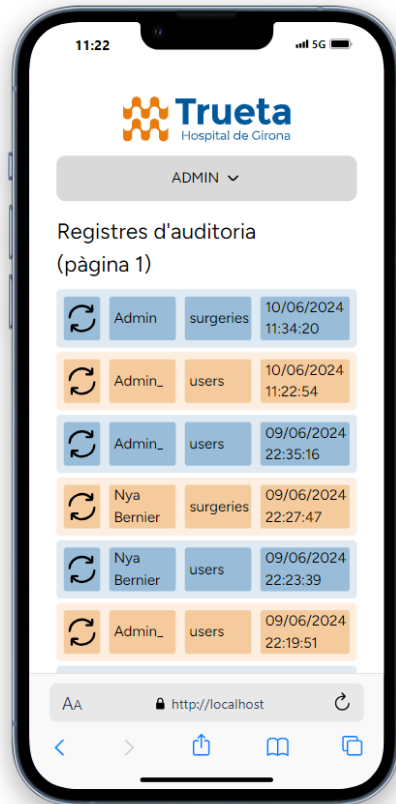


Figura 29: Llista de registres



Figura 30: Registre individual

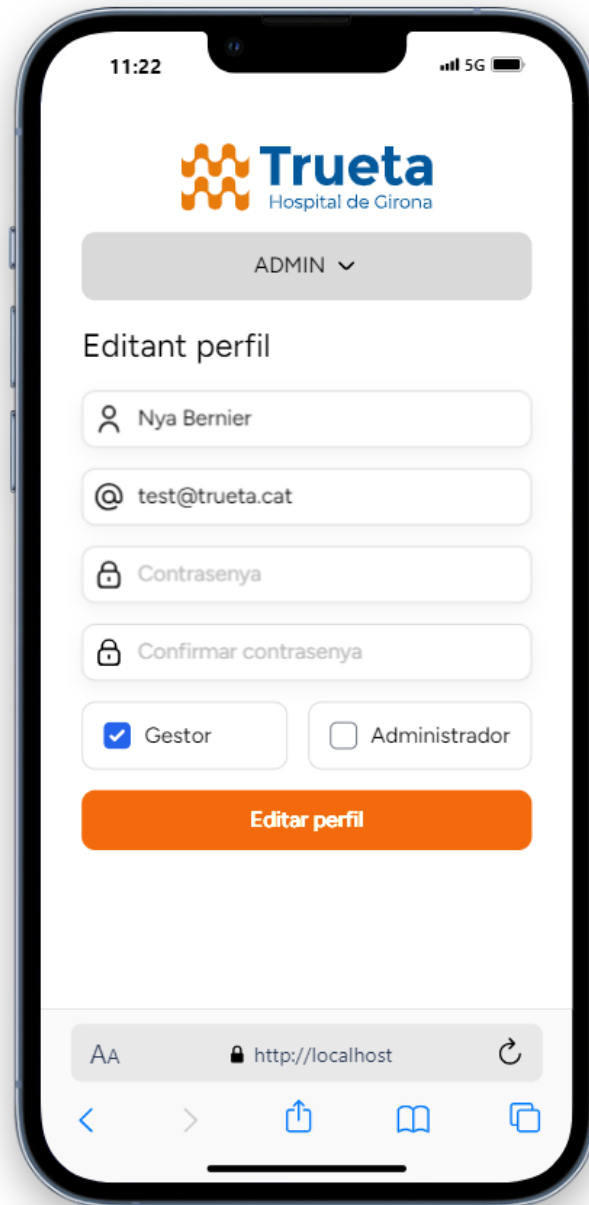


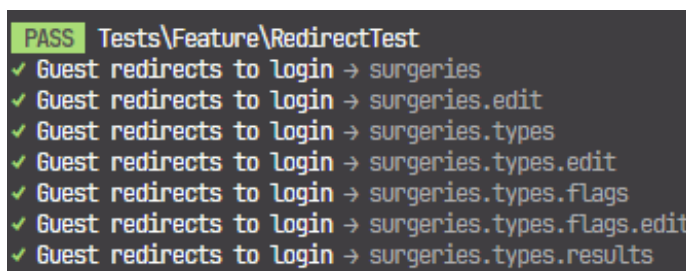
Figura 31: Edició d'un perfil com a administrador

6.3 Fases del Desenvolupament

6.3.1 Fase de Proves

Les proves es van realitzar en diverses etapes per garantir la qualitat del producte:

- **Proves Unitàries:** S'han escrit proves unitàries que comproven cada component en funcionament per separat. Aquestes solen centrar-se més a assegurar que les parts funcionin efectivament de manera individual en un sistema aïllat. Això garantirà que cada mòdul compleixi amb les especificacions, donant els resultats esperats. Les proves unitàries permeten la detecció precoç d'errors en l'etapa de desenvolupament i, per tant, es poden rectificar ràpidament i de manera eficient.



```
PASS Tests\Feature\RedirectTest
✓ Guest redirects to login → surgeries
✓ Guest redirects to login → surgeries.edit
✓ Guest redirects to login → surgeries.types
✓ Guest redirects to login → surgeries.types.edit
✓ Guest redirects to login → surgeries.types.flags
✓ Guest redirects to login → surgeries.types.flags.edit
✓ Guest redirects to login → surgeries.types.results
```

Figura 32: Mostra de proves unitàries

- **Proves d'extrem a extrem:** Les proves d'extrem a extrem les vam realitzar utilitzant Selenium IDE per assegurar-nos que els components del sistema funcionaven correctament de manera automàtica. Tots aquests tipus de proves simulen el comportament real de l'usuari i, després d'això, garantim que cada part del sistema interactui bé des del principi fins al final del procés, incloent-hi la verificació de fluxos de treball complexos i la detecció de qualsevol problema que pugui sorgir a causa de la interconnexió de diversos components.

	Command	Target
1	✓ open	/login
2	✓ set window size	974x1040
3	✓ click	id=email
4	✓ type	id=email
5	✓ type	id=password
6	✓ click	css=submit-button
7	✓ click	css=wizard-square-container: nth-child(1)
8	✓ click	css=wizard-square-container: nth-child(1)
9	✓ click	css=health-flags-form-checkb ox:nth-child(1) > .input-contain er
10	✓ click	css=health-flags-form-checkb ox:nth-child(2) > .input-contain er
11	✓ click	css=health-flags-form-checkb ox:nth-child(3) > .input-contain er
12	✓ click	css=health-flags-form-checkb ox:nth-child(4) > .input-contain er

Figura 33: Mostra de proves d'extrem a extrem amb Selenium IDE

6.3.2 Fase de Desplegament de pre-producció

S'ha fet un desplegament de pre-producció utilitzant GitHub Actions⁷ a l'Hospital Trueta, permetent una instal·lació ràpida i consistent. Es van seguir els següents passos:

- **Preparació de l'entorn de producció:** Vam crear un VPS⁸ amb Contabo, una empresa coneguda per oferir serveis de servidors virtuals privats d'alt rendiment a preus assequibles. Contabo proporciona una infraestructura fiable i escalable que és ideal per a desplegaments de pre-producció i producció. El pla que hem agafat és el més assequible, que costa 4€ al mes, i ens ofereix 4 nuclis de CPU, 6GB de RAM i 400GB SSD d'espai en

⁷GitHub Actions: Entorn que permet la integració contínua de l'aplicació

⁸VPS: (Virtual Private Server) És un servidor al núvol que ens permet pujar la web online

disc. Aquesta configuració ens permet tenir suficient capacitat per provar el sistema en condicions properes a la producció.

- **Script de desplegament:** Vam codificar un script de desplegament del servidor amb un pipeline de SSH.⁹ Aquest script automatitza el procés de desplegament, reduint el risc d'errors humans i assegurant que totes les actualitzacions es facin de manera consistent.⁴

```
last_ssh: |
cd ${ secrets.SSH_DIR }
tar -xzf app.tgz
composer install --optimize-autoloader --no-dev
npm install --production
cp env.example .env
sed -i "s/DB_CONNECTION=.*DB_CONNECTION=${ secrets.DB_CONNECTION }/" .env
sed -i "s/DB_HOST=.*DB_HOST=${ secrets.DB_HOST }/" .env
sed -i "s/DB_PORT=.*DB_PORT=${ secrets.DB_PORT }/" .env
sed -i "s/DB_DATABASE=.*DB_DATABASE=${ secrets.DB_DATABASE }/" .env
sed -i "s/DB_USERNAME=.*DB_USERNAME=${ secrets.DB_USERNAME }/" .env
sed -i "s/DB_PASSWORD=.*DB_PASSWORD=${ secrets.DB_PASSWORD }/" .env
mysql -u ${ secrets.DB_USERNAME } -p${ secrets.DB_PASSWORD } -e "DROP DATABASE IF EXISTS ${ secrets.DB_DATABASE }; CREATE DATABASE ${ secrets.DB_DATABASE };"
php artisan config:cache
php artisan key:generate --force
php artisan config:cache
mkdir -p storage/framework/views
chmod -R 775 storage
chown -R www-data:www-data storage
mkdir -p bootstrap/cache
chmod -R 775 bootstrap/cache
chown -R www-data:www-data bootstrap/cache
php artisan optimize:clear
php artisan migrate:refresh --force
php artisan db:seed --force
```

Figura 34: Tros de script de desplegament

Aquest script està situat a la ruta `.github/workflows/build.yml`, i el que aconseguim amb això, és que GitHub l'executi automàticament cada cop que pugem canvis a la branca 'develop' del repositori del projecte. Això ens permet tenir una integració contínua i desplegament continu (CI/CD), garantint que les noves funcionalitats i correccions de bugs es despleguin ràpidament i sense interrupcions.

- **Configuració de GitHub:** Per assegurar la seguretat i la integritat del procés de desplegament, hem utilitzat claus secretes de GitHub. Aquestes claus secretes són necessàries per establir connexions segures amb el servidor de producció mitjançant SSH. Les claus secretes s'emmagatzemen al repositori de GitHub de manera segura i es fan servir durant el pipeline de desplegament per autenticar-se en el servidor i executar els scripts necessaris.

⁹Pipeline: Sèrie d'instruccions o processos que s'executen de manera seqüencial i automàtica, sovint utilitzada en el desenvolupament de programari per a integrar i desplegar codi.

The screenshot shows a dark-themed interface for managing repository secrets. At the top left is the title 'Repository secrets' and at the top right is a blue button labeled 'New repository secret'. Below the header is a table with two columns: 'Name' and 'Last updated'. Each row represents a secret and includes a lock icon, the secret name, the time since it was last updated, and edit/delete icons.

Name	Last updated
DB_CONNECTION	17 hours ago
DB_DATABASE	17 hours ago
DB_HOST	17 hours ago
DB_PASSWORD	17 hours ago
DB_PORT	17 hours ago
DB_USERNAME	17 hours ago
SSH_DIR	18 hours ago
SSH_HOST	18 hours ago
SSH_PASSWORD	18 hours ago
SSH_PORT	18 hours ago
SSH_USER	18 hours ago

Figura 35: Claus secretes de repositori

L'ús de claus secretes ajuda a prevenir accessos no autoritzats i a mantenir la seguretat del sistema, assegurant que només les persones autoritzades puguin fer canvis al servidor de producció.

7 Prospectiva

Tot i que la solució desenvolupada ja està en fase de desplegament, hi ha diverses àrees on es pot treballar i desenvolupar més per millorar i augmentar la funcionalitat. Hem plantejat les següents idees:

- **Intel·ligència artificial:** Els algorismes d'intel·ligència artificial i l'aprenentatge automàtic poden derivar millors recomanacions antibiòtiques a partir de dades històriques i detectar patrons en els registres mèdics. Això permet fer prediccions sobre les respostes dels pacients a diferents antibiòtics, així com la possibilitat d'al·lèrgies o resistències.
- **Característiques de formació i educació:** Desenvolupar mòduls per a la formació basada en aplicacions sobre pràctiques adequades de profilaxi antibiòtica entre els treballadors de la salut. Aquests mòduls, dissenyats per a incloure tutorials interactius i vídeos educatius, podrien presentar casos pràctics per desenvolupar habilitats pràctiques mitjançant els coneixements impartits.
- **Millora de la interfície d'usuari:** Continuar iterant el disseny de la interfície d'usuari segons la retroacció per fer l'aplicació més intuïtiva i fàcil d'usar. Això pot incloure canvis en el disseny visual, l'optimització del flux de processos o noves funcions per servir l'usuari de la millor manera possible.
- **Desplegament a Altres Hospitals:** Un cop l'aplicació demostrí ser valuosa i funcional a l'Hospital Trueta, es pot considerar implementar-la en altres hospitals i centres de salut.

Les propostes de desenvolupament futures tenen com a objectiu la continuïtat de l'aplicació, afegint nous valors per facilitar la vida dels professionals de la salut, augmentar la qualitat de l'atenció mèdica i, per tant, millorar els resultats dels pacients.

8 Conclusió

En conclusió, el desenvolupament de l'aplicació per a l'Hospital Trueta representa un pas significatiu cap a la millora de l'eficiència i l'efectivitat en la presa de decisions clíniques.

A més, la gran qualitat d'aquest projecte queda demostrada pel fet que l'Hospital Trueta ha mostrat un gran interès en el projecte i ha decidit implementar l'aplicació a les seves instal·lacions.

Els efectes positius que es podran observar d'ara endavant a l'Hospital són els següents:

- Ajudar els anestesiològics a seleccionar els antibiòtics adequats de manera més precisa i ràpida.

- Reduir els errors mèdics, minimitzant la possibilitat d'error humà, cosa que disminueix el risc d'infeccions postoperatòries i millora els resultats clínics dels pacients.
- Millorar la precisió de la dosificació per evitar la resistència als antibiòtics a llarg termini, que és una de les majors amenaces per a la salut mundial.
- Reduir el temps per prendre decisions de 7,5 minuts a 45 segons, millorant l'eficiència aproximadament en un 90%.
- L'estalvi de temps gràcies a aquesta aplicació té un valor equivalent a 35.700€ anuals, mentre que el cost estimat per desenvolupar l'aplicació és de 15.750€. No obstant això, aprofitant que és un projecte de final de cicle, ha tingut un cost de 0€.

A Apèndix

A.1 Glossari de Termes

- **Profilaxi antibiòtica:** Administració d'antibiòtics preventius per protegir contra els gèrmens que més sovint causen infeccions en els diferents tipus de cirurgia.
- **Agile/Scrum:** Marc de treball per al desenvolupament de projectes que s'enfoca en la iteració contínua de petits increments de treball.
- **Scrum Master:** És un facilitador en equips Agile/Scrum, que ajuda a l'equip a seguir les pràctiques de Scrum, elimina obstacles, facilita les reunions i promou la comunicació efectiva per garantir l'eficiència i la qualitat del projecte.
- **Product backlog:** Llistat de tasques que es volen realitzar en el desenvolupament d'un projecte.
- **Sprint:** Període curt i fix de temps, típicament d'una a quatre setmanes, durant el qual un equip Agile/Scrum treballa per completar un conjunt específic de treballs predefinitos.
- **CI/CD:** Integració contínua i desplegament continu, un mètode de desenvolupament que permet realitzar desplegaments automàtics de codi de manera contínua.
- **VPS:** (Virtual Private Server) És un servidor al núvol que permet pujar la web online.

A.2 Estructura de Directoris

A continuació es mostra l'estructura de directoris del projecte:

```
project-root/  
  app/  
    Http/  
      Controllers/  
      Models/  
  bootstrap/  
  config/  
  database/  
    migrations/  
    seeders/  
  public/  
  resources/  
    js/  
      Pages/  
      Components/
```

```
views/  
routes/  
  api.php  
  web.php  
storage/  
tests/  
.github/  
  workflows/  
    build.yml
```

A.3 Configuració de l'Entorn de Desenvolupament

Per configurar l'entorn de desenvolupament, seguiu els següents passos:

- **Requisits previs:** Assegureu-vos de tenir instal·lats els següents programes:

- PHP (versió 8.2 o superior)
- Composer
- Node.js i npm
- Docker

- **Instal·lació de dependències:**

```
composer install  
npm install
```

- **Configuració de l'entorn:**

```
cp .env.example .env  
php artisan key:generate
```

Editeu el fitxer '.env' per configurar la base de dades i altres paràmetres necessaris.

- **Llançament de l'aplicació amb Docker:**

```
docker-compose up -d
```

A.4 Scripts de Comandes Utilitzats

A continuació es mostren alguns scripts de comandes utilitzats durant el desenvolupament:

- **Desplegament de pre-producció:**

```
name: laravel-build
on:
  push:
    branches:
      - develop
jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout code
        uses: actions/checkout@v3

      - name: Setup Node.js
        uses: actions/setup-node@v3
        with:
          node-version: '20'

      - name: Setup PHP
        uses: shivammathur/setup-php@master
        with:
          php-version: 8.3
          extensions: mbstring, ctype, fileinfo, openssl, PDO, bcmath, json,
            tokenizer, xml

      - name: Install Composer dependencies
        run: composer install --no-dev --no-interaction --prefer-dist

      - name: Install npm dependencies
        run: npm install

      - name: Build assets
        run: npm run build

      - name: Clean up
        run: |
          rm -rf node_modules
          rm -rf storage

      - name: Archive application
        run: tar -czf app.tgz ./*
```

```

- name: SSH SCP and deployment
  uses: cross-the-world/ssh-scp-ssh-pipelines@latest
  env:
    WELCOME: "ssh scp ssh pipelines"
    LASTSSH: "Doing something after copying"
  with:
    host: ${ secrets.SSH_HOST }
    user: ${ secrets.SSH_USER }
    pass: ${ secrets.SSH_PASSWORD }
    port: ${ secrets.SSH_PORT }
    connect_timeout: 10s
    scp: |
      './app.tgz' => ${ secrets.SSH_DIR }
    last_ssh: |
      cd ${ secrets.SSH_DIR }
      tar -xzf app.tgz
      composer install --optimize-autoloader --no-dev
      npm install --production
      cp env.example .env
      sed -i "s/DB_CONNECTION=.* /DB_CONNECTION=${ secrets.DB_CONNECTION
      }/" .env

      sed -i "s/DB_HOST=.* /DB_HOST=${ secrets.DB_HOST }/" .env
      sed -i "s/DB_PORT=.* /DB_PORT=${ secrets.DB_PORT }/" .env
      sed -i "s/DB_DATABASE=.* /DB_DATABASE=${ secrets.DB_DATABASE }
      }/" .env
      sed -i "s/DB_USERNAME=.* /DB_USERNAME=${ secrets.DB_USERNAME }
      }/" .env
      sed -i "s/DB_PASSWORD=.* /DB_PASSWORD=${ secrets.DB_PASSWORD }
      }/" .env
      mysql -u ${ secrets.DB_USERNAME } -p${ secrets.DB_PASSWORD }
      -e "DROP DATABASE IF EXISTS ${ secrets.DB_DATABASE };
      CREATE DATABASE ${ secrets.DB_DATABASE };"
      php artisan config:cache
      php artisan key:generate --force
      php artisan config:cache
      mkdir -p storage/framework/views
      chmod -R 775 storage
      chown -R www-data:www-data storage
      mkdir -p bootstrap/cache
      chmod -R 775 bootstrap/cache
      chown -R www-data:www-data bootstrap/cache
      php artisan optimize:clear
      php artisan migrate:refresh --force
      php artisan db:seed --force

```

- Execució de proves unitàries:

```
./vendor/bin/sail test
```