

Roger Massana

CONECTADRIVE

TREBALL DE FI DE GRAU

dirigit per Marc Sánchez

Grau d'Enginyeria Informàtica/Telemàtica



UNIVERSITAT ROVIRA I VIRGILI

Tarragona

2024

Resum.

Aquest TFG descriu el desenvolupament de ConectaDrive, una aplicació per gestionar reserves en empreses de VTC a Mallorca. Amb l'objectiu de modernitzar el sistema existent, reduir costos i millorar el servei al client, el projecte ha integrat tecnologies com AWS i Vue.js, utilitzant una arquitectura de microserveis. Aquest procés ha permès adquirir coneixements tècnics avançats, desenvolupar competències en gestió de projectes, practicar la integració de sistemes i millorar el pensament analític i resolutiu. ConectaDrive representa una millora significativa respecte a la plataforma anterior, aportant una solució robusta, escalable i segura.

Resumen.

Este TFG describe el desarrollo de ConectaDrive, una aplicación para gestionar reservas en empresas de VTC en Mallorca. Con el objetivo de modernizar el sistema existente, reducir costos y mejorar el servicio al cliente, el proyecto ha integrado tecnologías como AWS y Vue.js, utilizando una arquitectura de microservicios. Este proceso ha permitido adquirir conocimientos técnicos avanzados, desarrollar competencias en gestión de proyectos, practicar la integración de sistemas y mejorar el pensamiento analítico y resolutive. ConectaDrive representa una mejora significativa respecto a la plataforma anterior, aportando una solución robusta, escalable y segura.

Abstract.

This final project describes the development of ConectaDrive, an application to manage bookings for VTC companies in Mallorca. Aimed at modernizing the existing system, reducing costs, and improving customer service, the project has integrated technologies such as AWS and Vue.js, using a microservices architecture. This process has enabled the acquisition of advanced technical knowledge, the development of project management skills, practice in system integration, and improved analytical and problem-solving thinking. ConectaDrive represents a significant improvement over the previous platform, providing a robust, scalable, and secure solution.

Índex

1	INTRODUCCIÓ	V
1.1	MOTIVACIONS DEL PROJECTE EN ÀMBIT FORMATIU	V
1.1.1	<i>Adquisició de Coneixements Tècnics Avançats</i>	v
1.1.2	<i>Desenvolupament de Competències en Gestió de Projectes</i>	v
1.1.3	<i>Pràctica en Integració de Sistemes</i>	v
1.1.4	<i>Millorar i Practicar el Pensament Analític i Resolutiu</i>	v
1.2	MOTIVACIONS DEL PROJECTE A NIVELL EMPRESARIAL	VI
1.2.1	<i>Necessitat de Millora Tecnològica</i>	vi
1.2.2	<i>Optimització de la Gestió de Reserves</i>	vi
1.2.3	<i>Millora de la Competitivitat</i>	vi
2	REQUISITS	VII
2.1	REQUISITS FUNCIONALS	VII
2.2	REQUISITS NO FUNCIONALS	VIII
2.3	DESCRIPCIÓ DELS CASOS D'ÚS	VIII
2.3.1	<i>Visualització de Reserves</i>	viii
2.3.2	<i>Creació de Reserves Manualment</i>	x
2.3.3	<i>Integració amb Agències de Reserva Externes</i>	x
2.3.4	<i>Exportació de Dades a CSV</i>	xi
2.3.5	<i>Filtrat de Reserves</i>	xi
2.3.6	<i>Edició de Reserves</i>	xii
2.3.7	<i>Visualització de l'Historial d'una Reserva</i>	xii
2.3.8	<i>Visualització de Vehicles</i>	xiii
2.3.9	<i>Creació de Vehicles</i>	xiii
2.3.10	<i>Eliminació de Vehicles</i>	xiv
2.3.11	<i>Edició de Vehicles</i>	xiv
2.3.12	<i>Visualització de Mapa i Polígons</i>	xv
2.3.13	<i>Creació de Polígons per Localitzacions</i>	xv
2.3.14	<i>Visualització de Taula de Localitzacions</i>	xvii
2.3.15	<i>Edició de Localitzacions</i>	xvii
2.3.16	<i>Eliminació de Localitzacions</i>	xviii
2.3.17	<i>Càrrega de Fitxers</i>	xviii
2.3.18	<i>Visualització de Dades de Reserva</i>	xix
2.3.19	<i>Penjament de Dades de Reserva Importades</i>	xx
2.4	DIAGRAMES DE CASOS D'ÚS	XXI
3	DISSENY	XXIII
3.1	ARQUITECTURA DEL NÚVOL	XXIII
3.2	ROL DE CADA COMPONENT	XXIII
3.2.2	<i>Descripció Detallada del Flux</i>	xxvii
3.3	ARQUITECTURA I MODEL MVC	XXVIII
3.4	DESCRIPCIÓ I UTILITAT DE CADA PÀGINA	XXIX
3.4.1	<i>Gestor de Reserves</i>	xxix
3.4.2	<i>Gestor de Vehicles</i>	31
3.4.3	<i>Importador de Reserves</i>	32
3.4.4	<i>Pàgina de Localitzacions</i>	33
3.5	IMPLEMENTACIÓ DE LA BASE DE DADES EN POSTGRES	34
3.5.1	<i>Diagrama de la Base de Dades</i>	35
3.5.2	<i>Descripció dels Camps de la Base de Dades</i>	36
3.5.3	<i>Resum de la Relació entre les Taules</i>	37
4	IMPLEMENTACIÓ	38
4.1	DESCRIPCIÓ DE DIVERSES PÀGINES	38
4.1.1	<i>Pàgina de Login</i>	38

4.1.2	<i>Recuperar Contrasenya</i>	40
4.1.3	<i>Gestor de Reserves</i>	42
4.1.4	<i>Nova Reserva / Editar Reserva</i>	51
4.1.5	<i>Pàgina de Gestió de Vehicles</i>	55
4.1.6	<i>Pàgina de Gestió de Localitats</i>	57
4.1.7	<i>Pàgina per Importar Reserves</i>	58
5	AVALUACIÓ	61
5.1	TESTING.....	61
5.1.1	<i>Mocks</i>	61
5.2	IMPLEMENTACIÓ DELS TESTS	61
5.2.1	<i>Mock de l'API</i>	61
5.2.2	<i>Esperar a que Carregui</i>	61
5.2.3	<i>Verificació del Text</i>	62
5.2.4	<i>Verificació de Dades de l'Origen</i>	62
5.2.5	<i>Canvi d'estat</i>	62
5.3	COMPLIMENT DELS REQUISITS.....	62
5.3.1	<i>Requisits funcionals</i>	62
5.3.2	<i>Requisits no Funcionals</i>	65
6	CONCLUSIONS	67
7	REFERÈNCIES	68
8	ANNEX	69

Índex de figures

FIGURA 1. DIAGRAMA DE SEQÜÈNCIA PER LA VISUALITZACIÓ DE RESERVES.....	IX
FIGURA 2. DIAGRAMA DE SEQÜÈNCIA PER A LA INTEGRACIÓ AMB AGÈNCIES DE RESERVA EXTERNES.....	XI
FIGURA 3. DIAGRAMA DE SEQÜÈNCIA PER A LA CREACIÓ DE POLÍGONS PER LOCALITZACIONS	XVI
FIGURA 4. DIAGRAMA DE SEQÜÈNCIA PER A LA CÀRREGA DE FITXERS.....	XIX
FIGURA 5. DIAGRAMA GENERAL DE L'APLICACIÓ.....	XXI
FIGURA 6. DIAGRAMA DE CASOS D'ÚS RESERVES	XXII
FIGURA 7. DIAGRAMA DE CASOS D'ÚS VEHICLES	XXII
FIGURA 8. DIAGRAMA DE CASOS D'ÚS MAPA LOCALITZACIONS.....	XXII
FIGURA 9. DIAGRAMA DE L'ARQUITECTURA AL NÚVOL.....	XXIV
FIGURA 10. PÀGINA DE VISUALITZACIÓ DE RESERVES.....	XXX
FIGURA 11. PÀGINA PER EDITAR UNA RESERVA EXISTENT	XXX
FIGURA 12. PÀGINA PER CREAR UNA NOVA RESERVA	XXX
FIGURA 13. PÀGINA DE CREACIÓ I EDICIÓ D'UN VEHICLE	31
FIGURA 14. PÀGINA DE VISUALITZACIÓ DE VEHICLES.....	31
FIGURA 15. PÀGINA PER IMPORTAR NOVES RESERVES.....	32
FIGURA 16. PÀGINA PER VEURE I CREAR NOVES LOCALITATS	33
FIGURA 17. BASE DE DADES DE CONECTADRIVE	35
FIGURA 18. PÀGINA D'INICI DE SESSIÓ	38
FIGURA 19. PÀGINA CONTRASENYA PERDUDA.....	41
FIGURA 20. PÀGINA CONTRASENYA PERDUDA DESPRÉS DE FICAR EL CORREU.....	41
FIGURA 21. EMAIL PER RECUPERAR LA CONTRASENYA.....	41
FIGURA 22. PÀGINA PER FICAR LA NOVA CONTRASENYA	42
FIGURA 23. PÀGINA DEL GESTOR DE RESERVES.....	43
FIGURA 24. FILTRE DE DATA AL GESTOR DE RESERVES.....	43
FIGURA 25. HORA FILTRADA I ORDENADA AL GESTOR DE RESERVES	44
FIGURA 26. PÀGINA DE RESERVES FILTRE DE DESTINACIONS	47
FIGURA 27. GESTOR DE RESERVES EXPORTAR A CSV.....	50
FIGURA 28. CSV EXPORTAT DEL GESTOR DE RESERVES.....	50
FIGURA 29. PÀGINA PER CREAR O EDITAR UNA RESERVA	51
FIGURA 30. AGRUPAR RESERVES.....	54
FIGURA 31. HISTÒRIC DE LES RESERVES.....	55
FIGURA 32. GESTOR DE VEHICLES.....	55
FIGURA 33. CREAR UN VEHICLE.....	56
FIGURA 34. EDITAR UN VEHICLE.....	56
FIGURA 35. GESTIÓ DE LOCALITATS.....	57
FIGURA 36. PÀGINA D'IMPORTADOR DE RESERVES.....	59
FIGURA 37. VISUALITZACIÓ DE LES DADES DE RESERVA IMPORTADES	59

1 Introducció

Aquest Treball de Fi de Grau (TFG) es centra en el desenvolupament d'una aplicació innovadora per a Conectadrive, una empresa de VTC¹ que opera a Mallorca sota la marca Avantcab. L'objectiu principal d'aquest projecte és dissenyar i implementar una solució tecnològica que permeti gestionar eficientment les reserves programades, a diferència de les plataformes com ara Uber o Cabify que operen en temps real. Aquesta aplicació busca modernitzar el sistema actual, reduir costos operatius, i millorar l'experiència de servei al client, tot oferint una plataforma robusta, escalable i segura.

El projecte inclou la integració de tecnologies avançades com serveis d'AWS² incloent EC2³, VPC⁴, entre d'altres. A més, s'utilitza una arquitectura basada en microserveis per facilitar la mantenibilitat i escalabilitat del sistema.

1.1 Motivacions del Projecte en Àmbit Formatiu

1.1.1 Adquisició de Coneixements Tècnics Avançats

A través del desenvolupament d'aquest projecte, l'objectiu és adquirir un coneixement profund sobre les arquitectures de microserveis, així com aprendre a utilitzar tecnologies actuals de frontend. Això inclou entendre el desplegament d'infraestructures com EC2 i VPC, i la utilització d'eines de desenvolupament frontend com Vue.js i PrimeVue.

1.1.2 Desenvolupament de Competències en Gestió de Projectes

El projecte pretén també desenvolupar habilitats de planificació i execució de projectes en un entorn real. Això comporta la gestió del temps, recursos, i riscos, així com la coordinació amb diferents companys de feina per assegurar que els requisits del negoci són adequadament coberts.

1.1.3 Pràctica en Integració de Sistemes

Un altre objectiu formatiu clau és la capacitat d'integrar diferents components de software i hardware per crear una solució completa que interacció amb sistemes de tercers, com plataformes de reserva externes. Això requereix una comprensió detallada de les API⁵ i la gestió de la interacció entre diferents tecnologies.

1.1.4 Millorar i Practicar el Pensament Analític i Resolutiu

Enfrontar-se a problemes reals i trobar solucions innovadores i adequades, això inclou analitzar i millorar el pensament analític i resolutiu.

¹ Vehicle de Transport amb Conductor

² Amazon Web Services

³ Elastic Compute Cloud

⁴ Virtual Private Cloud

⁵ Application Programming Interface

1.2 Motivacions del Projecte a Nivell Empresarial

1.2.1 Necessitat de Millora Tecnològica

Un dels principals motius pel qual es va voler fer aquest projecte és la necessitat de substituir una aplicació antiga i costosa que utilitzava Avantcab anomenada Gestrans que era on tenien totes les reserves guardades. La nova aplicació desenvolupada per a Conectadrive no només modernitza el sistema existent sinó que també redueix els costos operatius a llarg termini per a l'empresa.

1.2.2 Optimització de la Gestió de Reserves

L'aplicació permet una gestió més eficaç de les reserves programades, essencial per a un model de negoci que opera amb planificació anticipada. Això inclou la coordinació de múltiples viatges, la gestió de canvis d'última hora, i la interacció amb plataformes de reserva de tercers, com Hoppa o Booking.com.

1.2.3 Millora de la Competitivitat

Integrar l'aplicació amb altres plataformes de reserva permet a Conectadrive accedir a un major volum de clients sense necessitat de gestionar totes les interaccions amb el client final. Això millora significativament la seva posició en el mercat competitiu del transport de passatgers.

2 Requisites

Els requisits defineixen clarament què ha de fer el sistema (funcionals) i com ha de ser el sistema en termes de qualitat o restriccions (no funcionals).

Els requisits funcionals descriuen les funcions específiques que el software o sistema ha de ser capaç de realitzar. Aquests requisits són accions concretes que el sistema ha de poder executar per a satisfer les necessitats del negoci, l'usuari o altres sistemes.

Els requisits no funcionals, d'altra banda, són criteris que descriuen com el sistema ha de ser. Aquests no estan relacionats directament amb les funcions específiques del sistema, sinó amb aspectes com el rendiment, la seguretat, la usabilitat, la compatibilitat, i altres qualitats del sistema.

2.1 Requisites Funcionals

- **RF1:** Mostrar una taula amb llistat de totes les reserves. Aquesta taula inclourà columnes com el número de reserva, data de reserva, origen, destinació, estat de la reserva, i altres detalls pertinents.
- **RF2:** Proveir una funcionalitat per a crear noves reserves manualment. Això inclourà un formulari amb els camps necessaris com data, hora, origen, destinació, nombre de passatgers, tipus de vehicle, entre altres.
- **RF3:** Proporcionar la capacitat d'exportar les dades mostrades a la taula a un fitxer CSV⁶. Això pot incloure el filtre aplicat, per exemple, mostrar només les 100 últimes reserves d'un dia específic.
- **RF4:** Permetre als usuaris filtrar la taula de reserves per diversos criteris, incloent-hi amb número de reserva, data, origen, destinació, estat, entre altres.
- **RF5:** Facilitar una pàgina que permeti editar els detalls d'una reserva específica seleccionant una fila de la taula. Això inclourà funcionalitats per modificar camps com ara l'hora, la data, el nombre de passatgers, entre d'altres.
- **RF6:** Proporcionar una pàgina on els usuaris puguin veure un registre de les modificacions que s'han fet anteriorment a una reserva, això inclourà qui va fer el canvi, quan es va fer, i què va ser modificat.
- **RF7:** Visualitzar una llista completa de vehicles en el sistema amb opcions de filtrat per característiques com marca, model, any, etc., mostrant detalls com matrícula, estat i quilometratge.
- **RF8:** Proporcionar una interfície per introduir les dades d'un nou vehicle, incloent-hi la matrícula, marca, model, any, estat i quilometratge, i confirmar la seva creació.
- **RF9:** Habilitar la funcionalitat per eliminar un vehicle del sistema mitjançant un identificador únic, proporcionant confirmació d'eliminació.
- **RF10:** Permetre l'edició de dades d'un vehicle existent amb validació d'entrada per assegurar la correcció dels canvis i confirmació de l'actualització.
- **RF11:** Mostrar un mapa interactiu amb polígons que representen diferents localitzacions, carregant totes les localitzacions existents en iniciar.
- **RF12:** Permetre als usuaris dibuixar polígons en el mapa per definir noves localitzacions, guardant les dades de la nova localització amb confirmació de creació.

⁶ Comma-Separated Values

- **RF13:** Visualitzar una taula amb totes les localitzacions definides, permetent interacció amb el mapa per fer zoom a localitzacions específiques.
- **RF14:** Ofereix la capacitat d'editar detalls o la forma dels polígons de localitzacions existents, actualitzant les dades en la base de dades i en el mapa amb confirmació d'edició.
- **RF15:** Habilitar l'eliminació de polígons i les localitzacions associades del sistema, proporcionant confirmació d'eliminació.
- **RF16:** Permetre als usuaris autenticats pujar fitxers XML⁷ o JSON⁸ que contenen informació de reserves, amb confirmació de càrrega i gestió d'errors de format o càrrega.
- **RF17:** Mostrar les dades de reserves processades des dels fitxers carregats, permetent als usuaris visualitzar les informacions detallades com data, origen, i destinació.
- **RF18:** Facilitar el penjament de les dades de reserva importades a la base de dades central, amb confirmació d'actualització i maneig d'errors en la connexió o procés de penjament.

2.2 Requisits no Funcionals

- **RNF1:** La interfície d'usuari ha de ser intuïtiva i fàcil d'utilitzar per a tot tipus d'usuaris, sense necessitat de formació extensiva.
- **RNF2:** La càrrega de la pàgina de gestió de reserves i les accions subsegüents com la cerca, filtrat, i exportació de dades a CSV han de completar-se amb un temps de resposta acceptable (menys de 2 segons per a interaccions comunes).
- **RNF3:** Protecció de dades personals i de negoci amb estàndards d'indústria per a la seguretat, com ara l'ús de HTTPS⁹, xifrat en trànsit i en repòs, i autenticació robusta.
- **RNF4:** L'aplicació ha de tenir una alta disponibilitat, amb objectius de nivell de servei (SLAs) establerts, un temps de funcionament del 99,9%.
- **RNF5:** La pàgina ha de ser compatible amb els navegadors web més comuns i versions recents.
- **RNF6:** El codi de l'aplicació ha de ser ben documentat, estructurat i fàcil de mantenir.

2.3 Descripció dels Casos d'Ús

2.3.1 Visualització de Reserves

- **Resum de la funcionalitat:** Mostrar una llista de totes les reserves en una taula.
- **Paràmetres d'entrada:** Cap, però pot incloure filtres si s'apliquen.

⁷ Extensible Markup Language

⁸ JavaScript Object Notation

⁹ Hypertext Transfer Protocol Secure

- **Paràmetres de sortida:** Llista de reserves amb detalls com ara el número de reserva, la data, l'origen, la destinació, el seu estat, etc.
 - **Actor:** Usuari autenticat.
 - **Precondició:** L'usuari ha d'estar autenticat i autoritzat per accedir a la informació de reserves.
 - **Post condició:** La taula de reserves s'actualitza amb les dades més recents disponibles.
- **Procés normal principal:**
 1. L'usuari inicia sessió.
 2. L'usuari accedeix a la secció de reserves.
 3. Es mostren les reserves en una taula.
 4. L'usuari pot aplicar filtres per refinar la visualització.
 5. La taula s'actualitza amb les dades més recents.
 - **Excepcions:**
 - Error de connexió amb la base de dades.
 - No hi ha dades disponibles per mostrar.

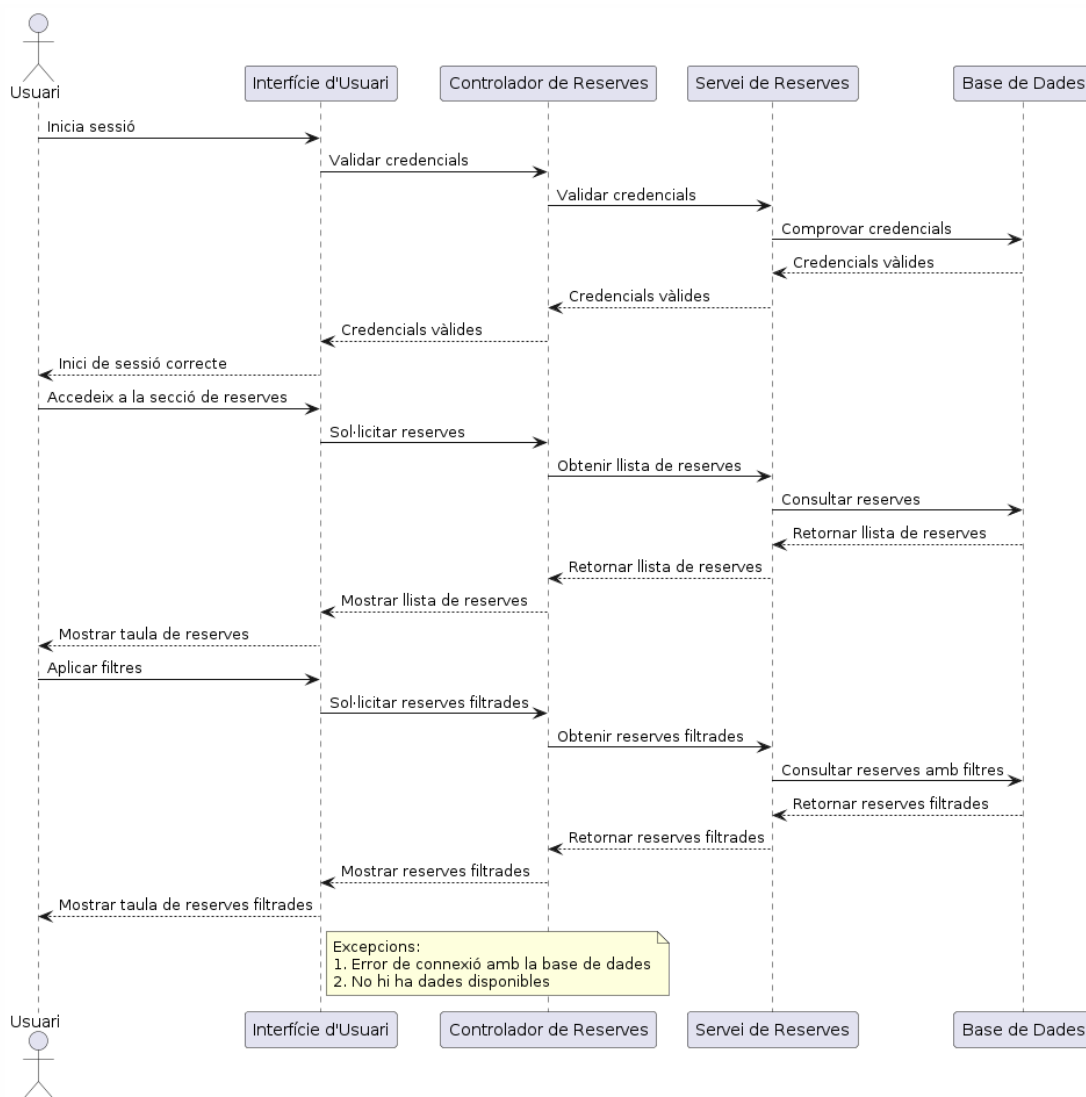


Figura 1. Diagrama de seqüència per la visualització de reserves

2.3.2 Creació de Reserves Manualment

- **Resum de la funcionalitat:** Proporcionar una interfície per introduir les dades d'una nova reserva manualment.
- **Paràmetres d'entrada:** Dades de la reserva com data, hora, origen, destinació, nombre de passatgers, tipus de vehicle.
- **Paràmetres de sortida:** Confirmació de la reserva creada.
- **Actor:** Usuari autenticat
- **Precondició:** L'usuari ha d'estar autenticat i tenir permisos per crear reserves.
- **Post condició:** Una nova reserva és afegida a la base de dades i es mostra una confirmació a l'usuari.
- **Procés normal principal:**
 1. L'usuari inicia sessió.
 2. L'usuari accedeix a la secció de creació de reserves.
 3. L'usuari introdueix les dades de la nova reserva.
 4. El sistema valida les dades introduïdes.
 5. Si les dades són vàlides, el sistema guarda la reserva a la base de dades.
 6. El sistema mostra una confirmació de la reserva creada.
- **Excepcions:**
 - Dades de reserva incompletes o invàlides.
 - Fallada al guardar la informació a la base de dades

2.3.3 Integració amb Agències de Reserva Externes

- **Resum de la funcionalitat:** Automatitzar la importació de reserves des de plataformes externes.
- **Paràmetres d'entrada:** Credencials/API Keys necessàries per a l'accés a les plataformes externes.
- **Paràmetres de sortida:** Reserves importades mostrades en la taula de reserves.
- **Actor:** Sistema
- **Precondició:** El sistema ha de tenir accés vàlid a les API de les agències externes amb les credencials correctes.
- **Post condició:** Les reserves importades són afegides a la taula de gestió de reserves.
- **Procés normal principal:**
 1. El sistema accedeix a les API de les agències externes utilitzant les credencials proporcionades.
 2. El sistema envia una sol·licitud d'importació de reserves a les API externes.
 3. Les API externes retornen les dades de les reserves.
 4. El sistema processa i valida les dades de les reserves rebudes.
 5. Les reserves vàlides són afegides a la taula de gestió de reserves.
 6. El sistema mostra les noves reserves importades en la taula de reserves.
- **Excepcions:**
 - Credencials invàlides o expirades.
 - Error de connexió amb les plataformes externes.
 - No es reben les dades de les plataformes externes.

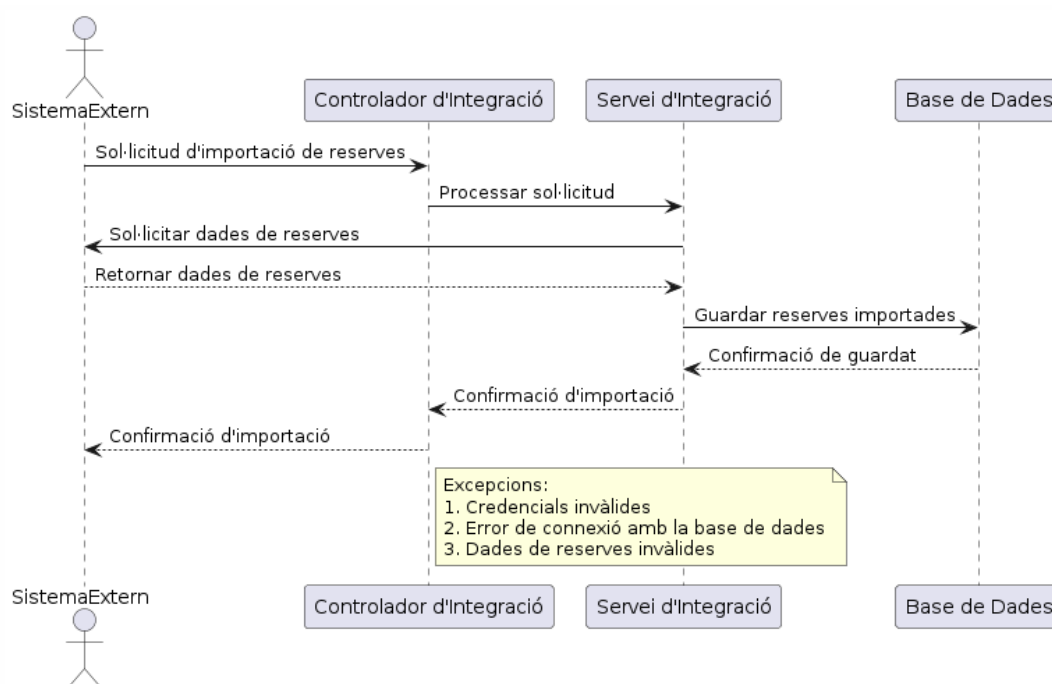


Figura 2. Diagrama de seqüència per a la integració amb agències de reserva externes

2.3.4 Exportació de Dades a CSV

- **Resum de la funcionalitat:** Permetre als usuaris exportar la llista de reserves a un arxiu CSV.
- **Paràmetres d'entrada:** Filtres aplicats a la llista de reserves.
- **Paràmetres de sortida:** Arxiu CSV amb les dades de les reserves.
- **Actor:** Usuari autènticat
- **Precondició:** L'usuari ha d'estar autènticat i tenir permisos per exportar dades.
- **Post condició:** Un arxiu CSV és generat i descarregat per l'usuari.
- **Procés normal principal:**
 1. L'usuari inicia sessió.
 2. L'usuari aplica els filtres desitjats a la llista de reserves.
 3. L'usuari selecciona l'opció per exportar les dades a CSV.
 4. El sistema processa les dades de les reserves segons els filtres aplicats.
 5. L'usuari descarrega l'arxiu CSV generat.
- **Excepcions:**
 - Error en la generació del fitxer CSV.
 - No hi ha dades per exportar segons els filtres aplicats.

2.3.5 Filtrat de Reserves

- **Resum de la funcionalitat:** Aplicar filtres a la llista de reserves per facilitar la cerca d'informació específica.
- **Paràmetres d'entrada:** Criteris de filtre com número de reserva, data, origen, etc.
- **Paràmetres de sortida:** Llista filtrada de reserves.
- **Actor:** Usuari autènticat

- **Precondició:** L'usuari ha d'estar autenticat i tenir accés a la taula de reserves.
- **Post condició:** La llista de reserves es mostra actualitzada segons els criteris de filtre aplicats.
- **Procés normal principal:**
 1. L'usuari inicia sessió.
 2. L'usuari accedeix a la taula de reserves.
 3. L'usuari aplica els criteris de filtre desitjats (número de reserva, data, origen, etc.).
 4. El sistema processa els criteris de filtre aplicats.
 5. El sistema actualitza i mostra la llista de reserves segons els criteris de filtre.
 6. L'usuari visualitza la llista filtrada de reserves.
- **Excepcions:**
 - No hi ha reserves que coincideixin amb els criteris de filtre.
 - Error en processar la sol·licitud de filtratge.

2.3.6 Edició de Reserves

- **Resum de la funcionalitat:** Permetre l'edició de les dades d'una reserva existent.
- **Paràmetres d'entrada:** Dades actualitzades de la reserva seleccionada.
- **Paràmetres de sortida:** Confirmació de les dades actualitzades.
- **Actor:** Usuari autenticat
- **Precondició:** L'usuari ha d'estar autenticat i tenir permisos per editar reserves.
- **Post condició:** Les modificacions en la reserva són guardades a la base de dades i es confirma a l'usuari.
- **Procés normal principal:**
 1. L'usuari inicia sessió.
 2. L'usuari accedeix a la taula de reserves.
 3. L'usuari selecciona la reserva que vol editar.
 4. L'usuari introdueix les dades actualitzades de la reserva.
 5. El sistema valida les dades introduïdes.
 6. El sistema guarda les modificacions a la base de dades.
 7. El sistema mostra una confirmació de les dades actualitzades a l'usuari.
- **Excepcions:**
 - Dades invàlides introduïdes per l'usuari.
 - La reserva que s'intenta editar ja no existeix.
 - Error de connexió a la base de dades durant l'actualització.

2.3.7 Visualització de l'Historial d'una Reserva

- **Resum de la funcionalitat:** Mostrar una llista de l'historial d'una reserva en una taula.
- **Paràmetres d'entrada:** Cap.
- **Paràmetres de sortida:** Llista de les modificacions amb detalls com ara el número de reserva, la data de la modificació, el tipus de modificació, i l'usuari que ha fet la modificació.
- **Actor:** Usuari autenticat
- **Precondició:** L'usuari ha d'estar autenticat i tenir permisos per accedir a l'historial de reserves.
- **Post condició:** Es mostren les modificacions històriques de la reserva seleccionada.
- **Procés normal principal:**

1. L'usuari inicia sessió.
 2. L'usuari accedeix a la secció de reserves.
 3. L'usuari selecciona una reserva específica.
 4. L'usuari accedeix a l'opció per visualitzar l'historial de la reserva.
 5. El sistema recupera les dades de l'historial de la reserva.
 6. El sistema mostra una taula amb l'historial de la reserva, incloent-hi el número de reserva, la data de la modificació, el tipus de modificació, i l'usuari que ha fet la modificació.
- **Excepcions:**
 - Error en recuperar les dades dels històrics.

2.3.8 Visualització de Vehicles

- **Resum de la funcionalitat:** Mostrar una llista de tots els vehicles registrats en el sistema.
- **Paràmetres d'entrada:** Cap, però pot incloure filtres si s'apliquen per buscar vehicles específics segons marca, model, any, etc.
- **Paràmetres de sortida:** Llista de vehicles amb detalls com matrícula, marca, model, any, estat (actiu/inactiu), i nombre de quilòmetres.
- **Actor:** Usuari autenticat
- **Precondició:** L'usuari ha d'estar autenticat i tenir permisos per veure els vehicles.
- **Post condició:** Es mostren tots els vehicles que hi ha actualment al sistema.
- **Procés normal principal:**
 1. L'usuari inicia sessió.
 2. L'usuari accedeix a la secció de vehicles.
 3. L'usuari aplica filtres (opcional) per buscar vehicles específics segons criteris com marca, model, any, etc.
 4. El sistema processa els criteris de filtre aplicats (si n'hi ha).
 5. El sistema recupera i mostra la llista de vehicles amb detalls com matrícula, marca, model, any, estat (actiu/inactiu), i nombre de quilòmetres.
 6. L'usuari visualitza la llista de vehicles actualment registrats en el sistema.
- **Excepcions:**
 - No hi ha vehicles que coincideixin amb els criteris dels vehicles.
 - Error en processar la sol·licitud de filtratge.
 - No hi ha vehicles registrats per mostrar.

2.3.9 Creació de Vehicles

- **Resum de la funcionalitat:** Proporcionar una interfície per a introduir les dades d'un nou vehicle i afegir-lo al sistema.
- **Paràmetres d'entrada:** Dades del vehicle com matrícula, marca, model, any, estat i quilometratge.
- **Paràmetres de sortida:** Confirmació de la creació del vehicle.
- **Actor:** Usuari autenticat
- **Precondició:** L'usuari ha d'estar autenticat i tenir permisos per afegir un nou vehicle.
- **Post condició:** S'afegeix el vehicle i es dona una confirmació a l'usuari.
- **Procés normal principal:**
 1. L'usuari inicia sessió.
 2. L'usuari accedeix a la secció d'afegir vehicle.

3. L'usuari introdueix les dades del nou vehicle (matrícula, marca, model, any, estat, quilometratge).
 4. El sistema valida les dades introduïdes.
 5. Si les dades són vàlides, el sistema guarda el nou vehicle a la base de dades.
 6. El sistema mostra una confirmació de la creació del vehicle a l'usuari.
- **Excepcions:**
 - Error al crear el vehicle.
 - Error de connexió amb la base de dades.

2.3.10 Eliminació de Vehicles

- **Resum de la funcionalitat:** Proporcionar una interfície per a seleccionar i eliminar un o diversos vehicles del sistema.
- **Paràmetres d'entrada:** Identificadors dels vehicles a eliminar seleccionats mitjançant checkboxes.
- **Paràmetres de sortida:** Confirmació de l'eliminació dels vehicles.
- **Actor:** Usuari autènticat
- **Precondició:** L'usuari ha d'estar autènticat i tenir permisos per eliminar vehicles.
- **Post condició:** Els vehicles seleccionats s'eliminen del sistema i es mostra una confirmació a l'usuari.
- **Procés normal principal:**
 - L'usuari inicia sessió.
 - L'usuari accedeix a la secció de veure vehicles.
 - L'usuari selecciona els vehicles que vol eliminar utilitzant checkboxes.
 - L'usuari fa clic al botó d'eliminar.
 - El sistema demana confirmació de l'eliminació.
 - L'usuari confirma l'eliminació.
 - El sistema verifica els permisos i procedeix a eliminar els vehicles seleccionats.
 - El sistema mostra una confirmació de l'eliminació dels vehicles a l'usuari.
- **Excepcions:**
 - Error al eliminar els vehicles.
 - Error de connexió amb la base de dades.
 - L'usuari no té permisos suficients per eliminar vehicles.

2.3.11 Edició de Vehicles

- **Resum de la funcionalitat:** Permetre l'edició de les dades d'un vehicle existent.
- **Paràmetres d'entrada:** Identificador del vehicle i les dades actualitzades que es volen modificar (marca, model, any, estat, quilometratge).
- **Paràmetres de sortida:** Confirmació de les modificacions realitzades en les dades del vehicle.
- **Actor:** Usuari autènticat
- **Precondició:** L'usuari ha d'estar autènticat i tenir permisos per editar les dades dels vehicles.
- **Post condició:** Les dades del vehicle són actualitzades a la base de dades i una confirmació és mostrada a l'usuari.
- **Procés normal principal:**
 1. L'usuari inicia sessió.
 2. L'usuari accedeix a la secció d'edició de vehicles.
 3. L'usuari selecciona el vehicle que vol editar mitjançant el seu identificador.

4. L'usuari introdueix les dades actualitzades del vehicle (marca, model, any, estat, quilometratge).
 5. El sistema valida les dades introduïdes.
 6. Si les dades són vàlides, el sistema actualitza les dades del vehicle a la base de dades.
 7. El sistema mostra una confirmació de les modificacions realitzades a l'usuari.
- **Excepcions:**
 - Entrada de dades invàlida (e.g., any del vehicle no realista, format incorrecte de matrícula).
 - El vehicle que s'intenta editar no existeix.
 - Error en actualitzar les dades a la base de dades.

2.3.12 Visualització de Mapa i Polígons

- **Resum de la funcionalitat:** Mostrar un mapa interactiu que permet als usuaris veure i dibuixar polígons que representen diferents localitats.
- **Paràmetres d'entrada:** Cap, però la pàgina carregarà totes les localitzacions existents en iniciar.
- **Paràmetres de sortida:** Mapa amb polígons dibuixats que representen les localitzacions.
- **Actor:** Usuari autenticat
- **Precondició:** L'usuari ha d'estar autenticat i autoritzat per accedir a la pàgina de gestió de localitzacions.
- **Post condició:** El mapa es carrega amb tots els polígons corresponents a les localitzacions actuals mostrades.
- **Procés normal principal:**
 1. L'usuari inicia sessió.
 2. L'usuari accedeix a la pàgina de gestió de localitzacions.
 3. El sistema carrega el mapa interactiu.
 4. El sistema recupera les dades de totes les localitzacions existents des de la base de dades.
 5. El sistema dibuixa polígons al mapa per representar les localitzacions recuperades.
 6. L'usuari pot veure i interactuar amb el mapa i els polígons representats.
- **Excepcions:**
 - Error de càrrega del mapa a causa de problemes amb el servei de mapes.
 - No es poden carregar les dades de localitzacions des de la base de dades.

2.3.13 Creació de Polígons per Localitzacions

- **Resum de la funcionalitat:** Permetre als usuaris dibuixar polígons en el mapa per definir noves localitzacions.
- **Paràmetres d'entrada:** Punts seleccionats en el mapa per formar el polígon.
- **Paràmetres de sortida:** Una nova localització representada per un polígon és creada i guardada.
- **Actor:** Usuari autenticat
- **Precondició:** L'usuari ha d'estar autenticat i tenir permisos per crear noves localitzacions.
- **Post condició:** Un nou polígon és creat en el mapa i les dades de la nova localització són guardades a la base de dades.

- **Procés normal principal:**
 1. L'usuari inicia sessió.
 2. L'usuari accedeix a la pàgina de gestió de localitzacions.
 3. L'usuari selecciona punts en el mapa per dibuixar un nou polígon.
 4. El sistema valida els punts seleccionats.
 5. Si els punts són vàlids, el sistema crea el polígon al mapa.
 6. L'usuari introdueix les dades de la nova localització (nom, descripció, etc.).
 7. El sistema guarda les dades de la nova localització a la base de dades.
 8. El sistema mostra una confirmació de la creació de la nova localització a l'usuari.
- **Excepcions:**
 - Dades de localització invàlides o incompletes.
 - Error en guardar les dades de la localització a la base de dades.

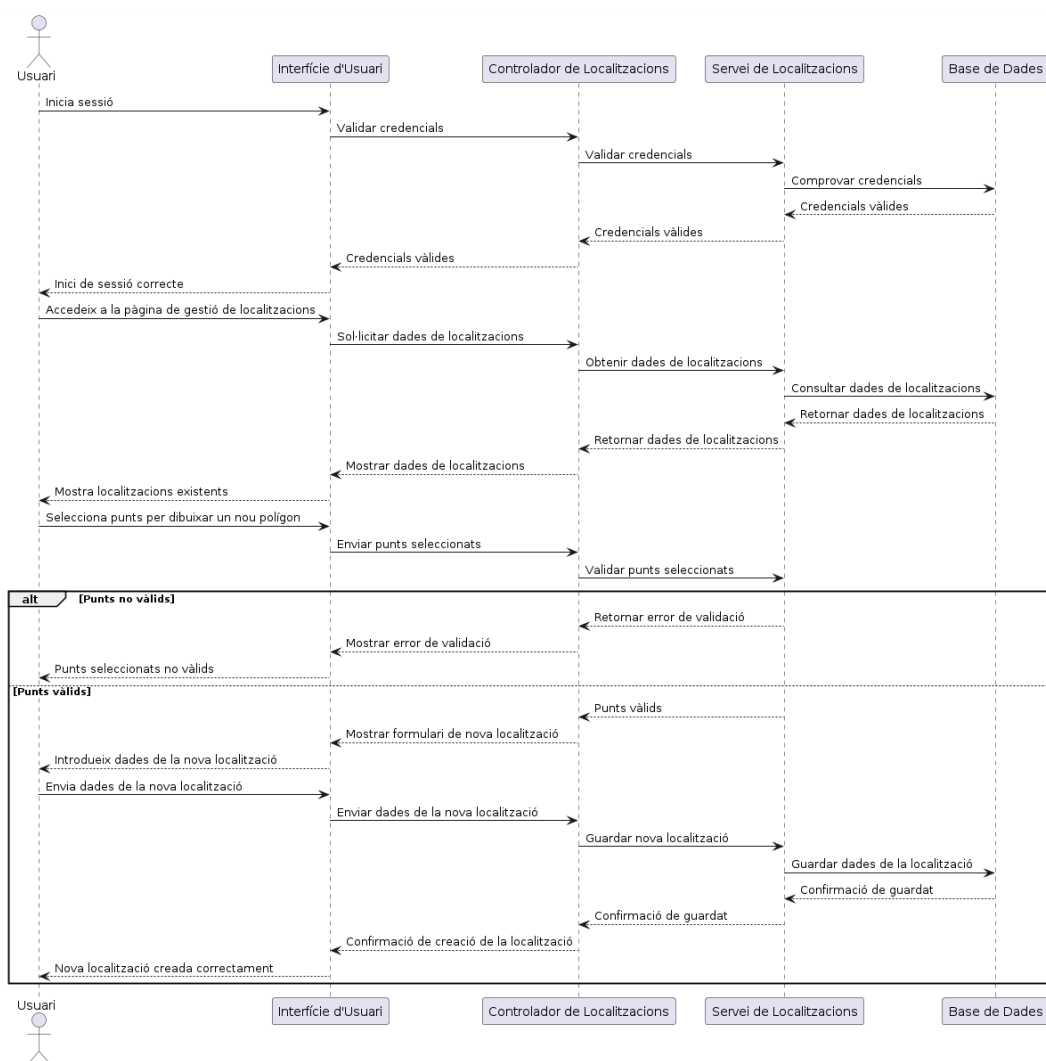


Figura 3. Diagrama de seqüència per a la creació de polígons per localitzacions

2.3.14 *Visualització de Taula de Localitzacions*

- **Resum de la funcionalitat:** Mostrar una taula amb la llista de totes les localitzacions definides, amb opció de fer zoom en la localitat en el mapa quan es selecciona una entrada de la taula.
- **Paràmetres d'entrada:** Cap.
- **Paràmetres de sortida:** Llista de localitzacions amb detalls bàsics i funcionalitat per interactuar amb el mapa.
- **Actor:** Usuari autenticat
- **Precondició:** L'usuari ha d'estar autenticat i autoritzat per accedir a les dades de localització.
- **Post condició:** La taula es mostra amb totes les localitzacions disponibles i permet la interacció amb el mapa.
- **Procés normal principal:**
 1. L'usuari inicia sessió.
 2. L'usuari accedeix a la pàgina de visualització de localitzacions.
 3. El sistema recupera les dades de totes les localitzacions des de la base de dades.
 4. El sistema mostra una taula amb la llista de totes les localitzacions disponibles.
 5. L'usuari selecciona una localització de la taula.
 6. El sistema fa zoom en la localització seleccionada en el mapa interactiu.
- **Excepcions:**
 - No hi ha localitzacions disponibles per mostrar.
 - Error en la recuperació de dades de la base de dades.

2.3.15 *Edició de Localitzacions*

- **Resum de la funcionalitat:** Permetre als usuaris editar els detalls o la forma dels polígons de les localitzacions existents.
- **Paràmetres d'entrada:** Dades actualitzades o nous punts del polígon en el mapa.
- **Paràmetres de sortida:** Les dades actualitzades de la localització són guardades.
- **Actor:** Usuari autenticat
- **Precondició:** L'usuari ha d'estar autenticat i tenir permisos per editar localitzacions.
- **Post condició:** Els canvis en la localització seleccionada són actualitzats a la base de dades i reflectits en el mapa.
- **Procés normal principal:**
 1. L'usuari inicia sessió.
 2. L'usuari accedeix a la pàgina de gestió de localitzacions.
 3. L'usuari selecciona la localització que vol editar.
 4. L'usuari edita els detalls de la localització o modifica la forma del polígon en el mapa.
 5. El sistema valida les dades actualitzades.
 6. Si les dades són vàlides, el sistema actualitza la localització a la base de dades.
 7. El sistema reflecteix els canvis en el mapa.
 8. El sistema mostra una confirmació de les modificacions a l'usuari.
- **Excepcions:**
 - Dades invàlides o format incorrecte de la informació modificada.
 - La localització que s'intenta editar no existeix o no es pot trobar.

- Error en actualitzar les dades a la base de dades.

2.3.16 *Eliminació de Localitzacions*

- **Resum de la funcionalitat:** Permetre als usuaris eliminar polígons i les localitzacions associades del sistema.
- **Paràmetres d'entrada:** Identificador de la localització a eliminar.
- **Paràmetres de sortida:** Confirmació de l'eliminació de la localització.
- **Actor:** Usuari autènticat
- **Precondició:** L'usuari ha d'estar autènticat i tenir permisos per editar localitzacions.
- **Post condició:** Els canvis en la localització seleccionada són actualitzats a la base de dades i reflectits en el mapa.
- **Procés normal principal:**
 1. L'usuari inicia sessió.
 2. L'usuari accedeix a la pàgina de gestió de localitzacions.
 3. L'usuari selecciona la localització que vol eliminar.
 4. L'usuari confirma l'eliminació de la localització.
 5. El sistema valida la sol·licitud d'eliminació.
 6. Si la sol·licitud és vàlida, el sistema elimina la localització de la base de dades.
 7. El sistema actualitza el mapa per reflectir l'eliminació.
 8. El sistema mostra una confirmació de l'eliminació a l'usuari.
- **Excepcions:**
 - Dades invàlides o format incorrecte de la informació modificada.
 - La localització que s'intenta editar no existeix o no es pot trobar.
 - Error en actualitzar les dades a la base de dades.

2.3.17 *Càrrega de Fitxers*

- **Resum de la funcionalitat:** Permetre als usuaris pujar fitxers XML, JSON o CSV que contenen informació de reserves.
- **Paràmetres d'entrada:** Fitxers XML, JSON o CSV seleccionats per l'usuari.
- **Paràmetres de sortida:** Confirmació de la càrrega reeixida dels fitxers.
- **Actor:** Usuari autènticat
- **Precondició:** L'usuari ha d'estar autènticat i tenir permisos per pujar fitxers.
- **Post condició:** Els fitxers es guarden temporalment per a processament ulterior.
- **Procés normal principal:**
 1. L'usuari inicia sessió.
 2. L'usuari accedeix a la pàgina de càrrega de fitxers.
 3. L'usuari selecciona el fitxer XML, JSON o CSV per pujar.
 4. El sistema valida el format del fitxer.
 5. Si el fitxer és vàlid, el sistema carrega el fitxer i el guarda temporalment per a processament ulterior.
 6. El sistema mostra una confirmació de la càrrega reeixida a l'usuari.
- **Excepcions:**
 - El fitxer no està en un format compatible (XML, JSON o CSV).
 - Error en la càrrega del fitxer.

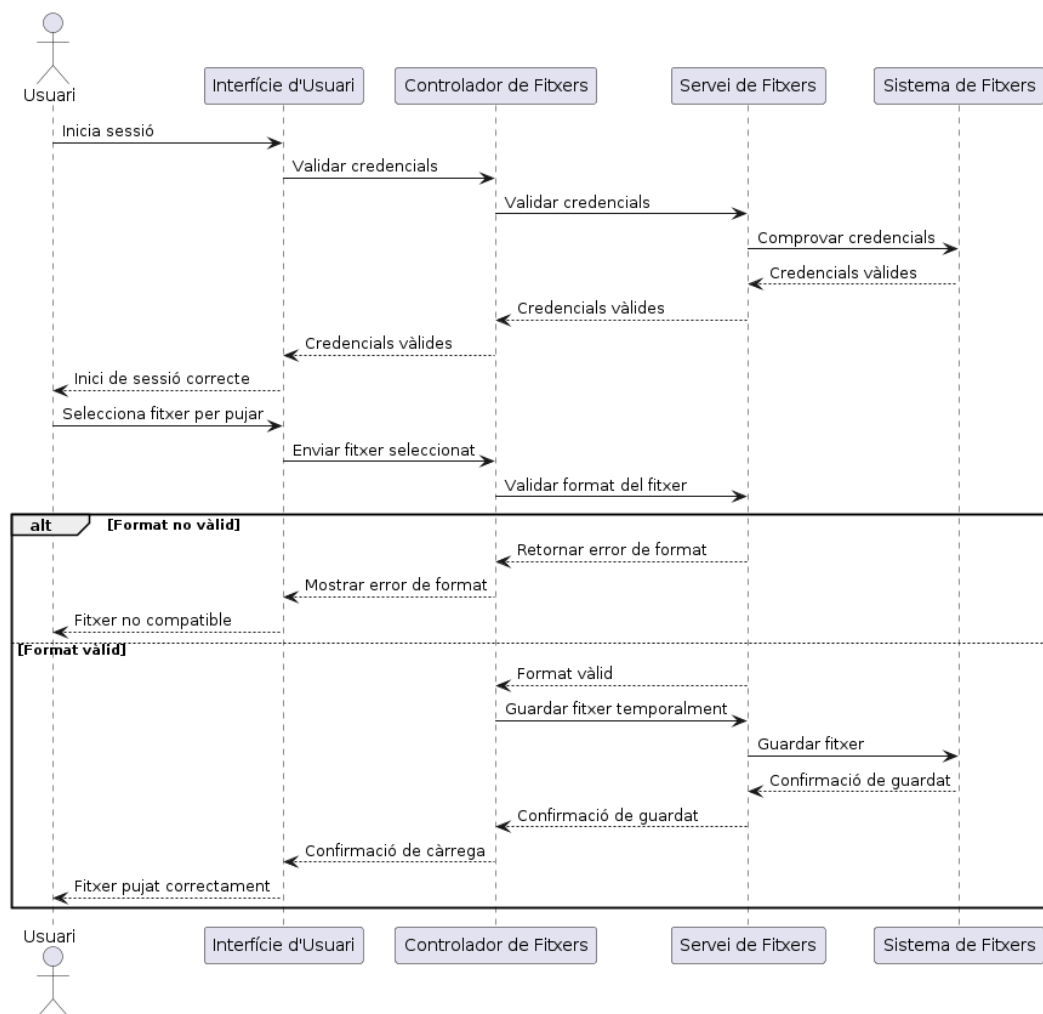


Figura 4. Diagrama de seqüència per a la càrrega de fitxers

2.3.18 Visualització de Dades de Reserva

- **Resum de la funcionalitat:** Visualitzar les dades de les reserves extretes dels fitxers pujats.
- **Paràmetres d'entrada:** Cap, utilitza els fitxers ja carregats.
- **Paràmetres de sortida:** Mostra una llista o taula de les reserves amb detalls com data, origen, destinació, etc.
- **Actor:** Usuari autènticat
- **Precondició:** Fitxers ja han estat carregats i processats correctament.
- **Post condició:** L'usuari pot visualitzar les dades de reserva processades.
- **Procés normal principal:**
 1. L'usuari inicia sessió.
 2. L'usuari accedeix a la pàgina de visualització de dades de reserva.
 3. El sistema recupera les dades de reserva dels fitxers pujats i processats.
 4. El sistema mostra una llista o taula amb les dades de reserva, incloent-hi detalls com data, origen, destinació, etc.
 5. L'usuari visualitza les dades de reserva processades.
- **Excepcions:**

- No hi ha dades disponibles per mostrar (p. ex., fitxers buits o no processats).

2.3.19 *Penjament de Dades de Reserva Importades*

- **Resum de la funcionalitat:** Penjar les dades de reserva importades a la base de dades central.
- **Paràmetres d'entrada:** Dades de reserva modificades.
- **Paràmetres de sortida:** Confirmació del penjament exitós de les dades.
- **Actor:** Usuari autenticat
- **Precondició:** Reserves importades guardades temporalment.
- **Post condició:** Les dades són actualitzades en la base de dades.
- **Procés normal principal:**
 1. L'usuari inicia sessió.
 2. L'usuari accedeix a la pàgina d'importar reserves.
 3. L'usuari importa les reserves i clica l'opció per penjar les dades a la base de dades central.
 4. El sistema valida les reserves guardades temporalment.
 5. El sistema intenta connectar-se a la base de dades central.
 6. Si la connexió és exitosa, el sistema penja les dades modificades a la base de dades.
 7. El sistema mostra una confirmació del penjament exitós de les dades a l'usuari.
- **Excepcions:**
 - Error en la connexió amb la base de dades.
 - Errors en el procés de penjament de dades.

2.4 Diagrames de Casos d'Ús

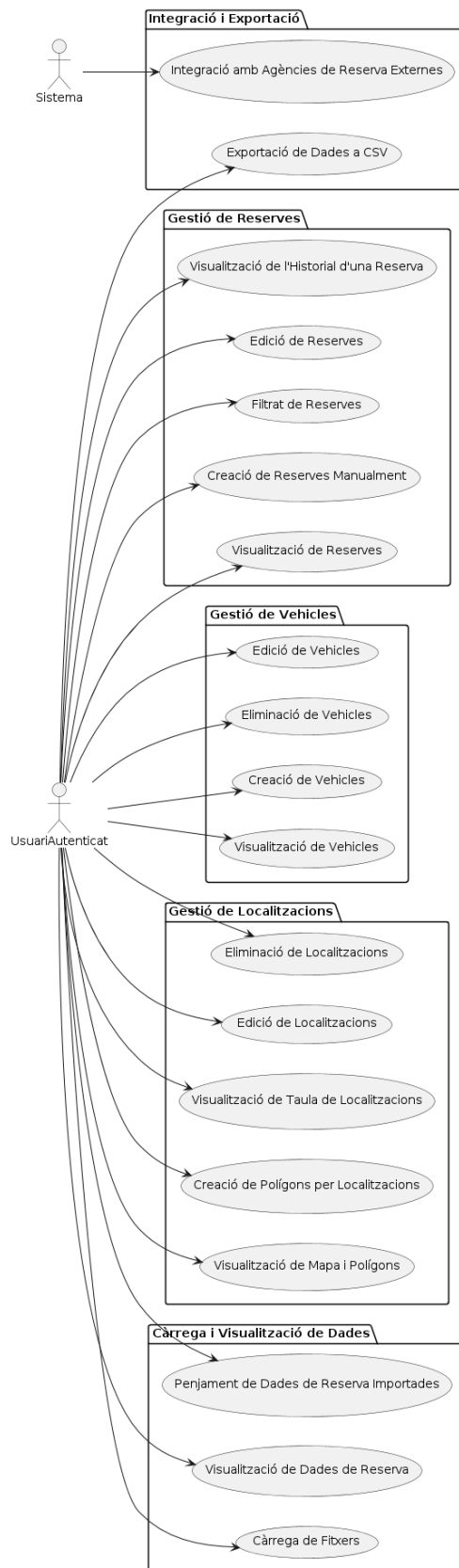


Figura 5. Diagrama general de l'aplicació

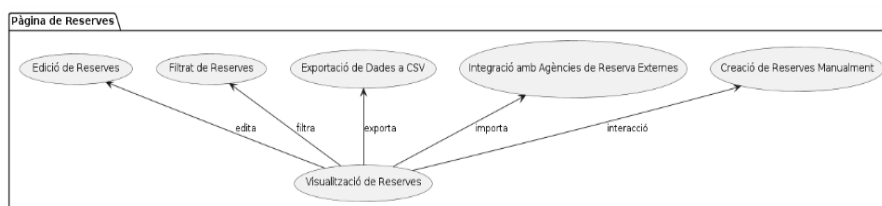


Figura 6. Diagrama de casos d'ús reserves

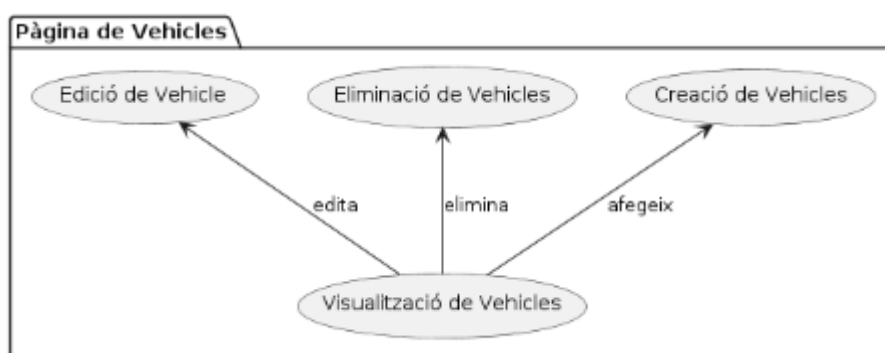


Figura 7. Diagrama de casos d'ús vehicles

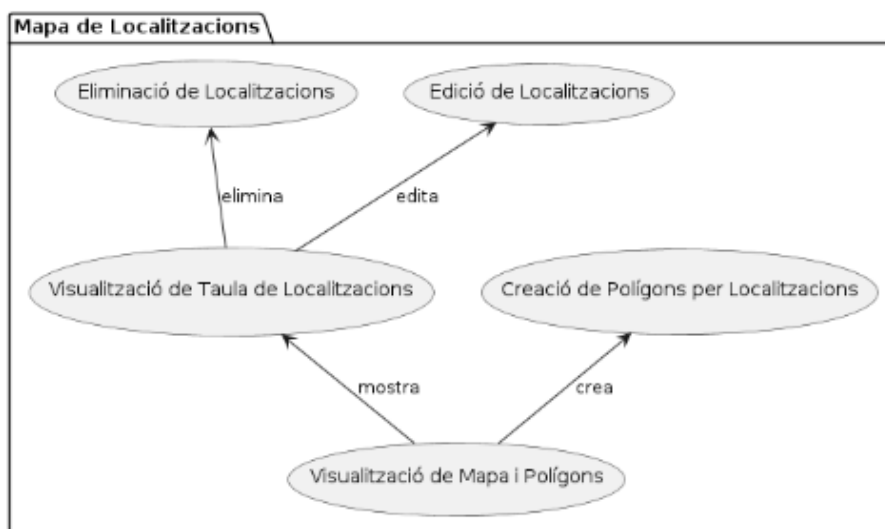


Figura 8. Diagrama de casos d'ús mapa localitzacions

3 Disseny

En aquest apartat de disseny, es descriu el rol de cada component del sistema, l'arquitectura utilitzada i les funcionalitats de les diverses pàgines de l'aplicació, així com la implementació de la base de dades. El sistema està estructurat seguint el patró de disseny MVC (Model-View-Controller) per garantir una separació clara entre la lògica de negoci, la presentació i el control, facilitant així el manteniment, l'escalabilitat i l'eficiència del desenvolupament.

Els components inclouen el frontend desenvolupat amb Vue.js per a una interfície d'usuari interactiva, el backend amb Spring Boot per gestionar la lògica de negoci, i una base de dades relacional Amazon RDS per a l'emmagatzematge segur de dades persistents. A més, s'utilitza Redis per a l'emmagatzematge temporal que millora el temps de resposta i AWS per proporcionar la infraestructura de desplegament.

3.1 Arquitectura del Núvol

Amazon Web Services (AWS) és una plataforma de serveis en núvol àmpliament adoptada que ofereix una infraestructura escalable, segura i de confiança. Hem seleccionat AWS per diverses raons:

- **Escalabilitat:** AWS permet escalar els recursos de manera dinàmica segons les necessitats de l'aplicació, assegurant que podem gestionar increments de demanda sense interrupcions del servei.
- **Disponibilitat:** Ens garanteix una alta disponibilitat mitjançant la seva infraestructura global, assegurant que l'aplicació estigui disponible per als usuaris en qualsevol moment.
- **Seguretat:** AWS proporciona eines avançades de seguretat i compliment normatiu, assegurant que les dades i les transaccions es mantenen segures.
- **Costos Efectius:** Amb el model de pagament per ús d'AWS, només paguem pels recursos que utilitzem, optimitzant els costos operatius.
- **Integració Senzilla:** Ofereix una àmplia gamma de serveis fàcils d'integrar amb altres tecnologies que fem servir, com Docker, Redis, i Spring Security.

3.2 Rol de Cada Component

En aquest apartat es tracten aquests aspectes previs i s'explica l'arquitectura que hem seguit a AWS i es pot veure a la Figura 9. En el meu cas, m'he centrat particularment en el desenvolupament del frontend, l'arquitectura que es pot apreciar al requadre de la part superior dreta de la següent imatge.

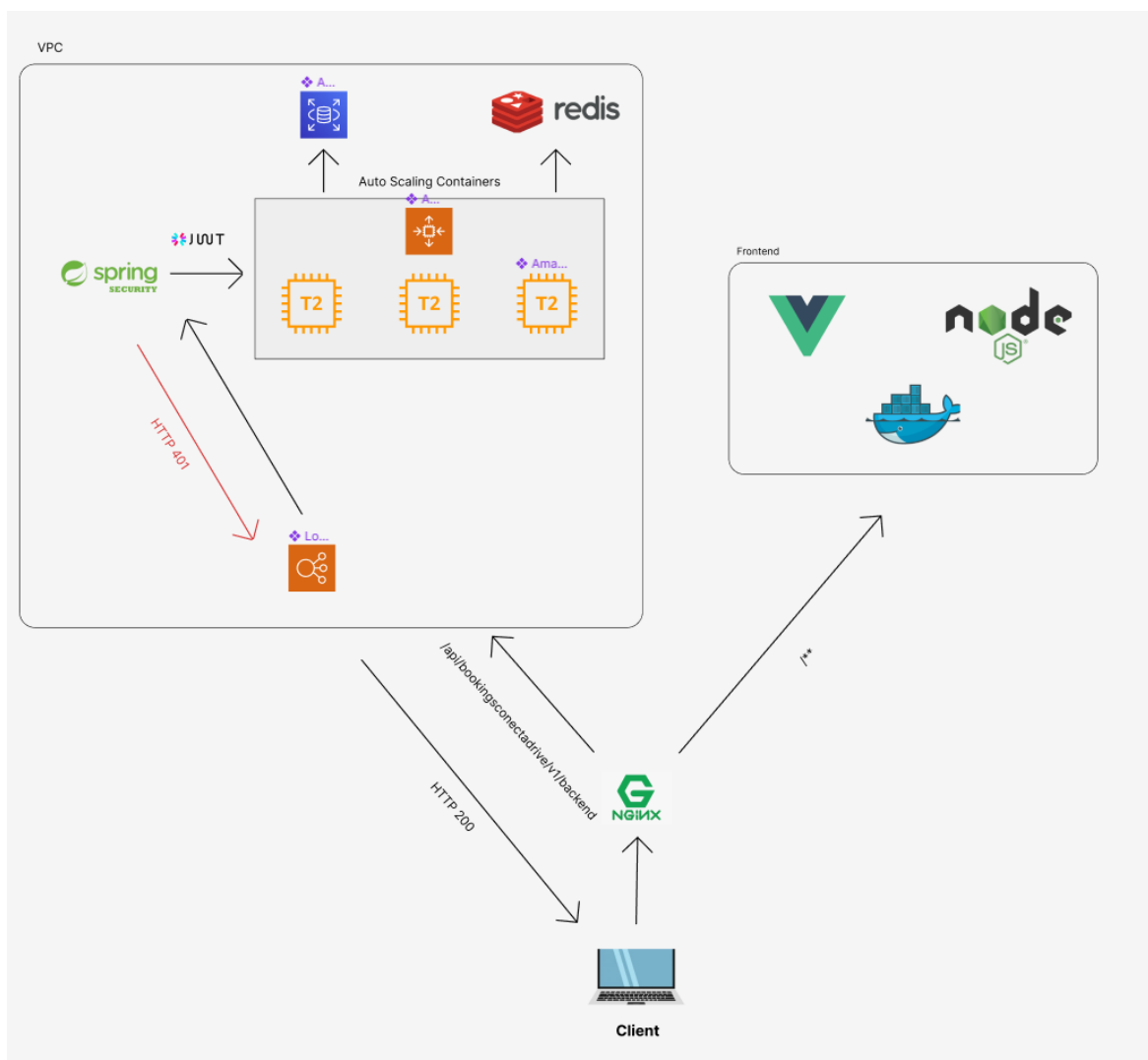


Figura 9. Diagrama de l'arquitectura al núvol

3.2.1.1 Frontend

- **Tecnologies Utilitzades:** Vue.js, Node.js, Docker, Typescript, TailwindCSS, Primevue, Pinia.
- **Funció:**
 - **Vue.js:** És un framework JavaScript per construir interfícies d'usuari. Facilita la creació de components reutilitzables i la gestió de l'estat de l'aplicació, permetent el desenvolupament de SPA¹⁰ millorant l'experiència de l'usuari amb càrregues de pàgina més ràpides.
 - **Docker:** Encapsula l'aplicació frontend en un contenidor, garantint que es pot desplegar de manera consistent en qualsevol entorn. Docker encapsula tot el que necessita l'aplicació (codi, dependències, configuracions) en un contenidor, assegurant que el desenvolupament, les proves i la producció siguin coherents.

¹⁰ Single Page Applications

- **Typescript:** És un superconjunt tipat de JavaScript que afegeix tipus estàtics al llenguatge. Typescript ajuda a detectar errors durant el temps de compilació en lloc del temps d'execució, la qual cosa millora la qualitat del codi i redueix els errors. També facilita el manteniment del codi en projectes grans gràcies a la seva capacitat per definir interfícies i tipus personalitzats, proporcionant una millor documentació i autocompletat durant el desenvolupament.
- **TailwindCSS:** És un framework CSS utilitari que permet aplicar estils directament a l'HTML mitjançant classes. TailwindCSS facilita el disseny d'interfícies perquè proporciona una àmplia gamma de classes predefinides per a estilitzar els elements de manera modular i coherent, reduint així la necessitat d'escriure CSS personalitzat, i accelerant el procés de desenvolupament.
- **Primevue:** És una biblioteca de components d'interfície d'usuari per a Vue.js. Primevue ofereix una col·lecció de components personalitzables i preparats per a la producció, com ara taules, diàlegs, calendaris i molts altres.
- **Pinia:** És una biblioteca de gestió de l'estat per a Vue.js, considerada una alternativa moderna a Vuex. Pinia proporciona una manera senzilla i intuïtiva de gestionar l'estat compartit de l'aplicació. Permet definir magatzems (stores) que contenen l'estat, mutacions i accions.

3.2.1.2 NGINX

- **Funció:**
 - Actua com a **servidor proxy invers**, rebent totes les sol·licituds dels clients i redirigint-les a la destinació adequada (frontend o backend).
 - **Gestió SSL:** Proporciona terminació SSL¹¹, encriptant i desencriptant el trànsit per protegir les dades mentre viatgen per la xarxa.
 - **Equilibri de càrrega:** Pot distribuir sol·licituds entre diversos servidors del backend per assegurar-se que cap servidor es sobre càrrega, millorant la disponibilitat i el rendiment.

3.2.1.3 Backend

- **Tecnologia Utilitzada:** Spring Security, Redis, Auto Scaling Containers.
- **Funció:**
 - **Spring Security:** Proporciona autenticació i autorització per a l'aplicació. Assegura que només els usuaris amb les credencials correctes puguin accedir a les funcionalitats protegides, gestionant diferents esquemes d'autenticació (per exemple, JWT, OAuth).
 - **Redis:** Actua com a **magatzem de memòria cau**. Emmagatzema dades que es consulten sovint en memòria, permetent accedir-hi més ràpidament que si es consultessin directament a la base de dades principal. Això redueix la latència i millora el rendiment general de l'aplicació. En el cas de la nostra aplicació ens serveix per a guardar els diferents vehicles que es demanen des del frontend, i així si l'usuari recarrega la pàgina, veu els vehicles en un temps més reduït.

¹¹ Secure Sockets Layer

- **Auto Scaling Containers:** Utilitza contenidors que s'escala automàticament per gestionar variacions en la càrrega de treball. Fent que l'aplicació pugui gestionar pics de trànsit sense degradació del servei, ajustant dinàmicament el nombre de contenidors actius segons la demanda.

3.2.1.4 Núvol (AWS)

- **Tecnologia Utilitzada:** EC2 Auto Scaling, Amplify, RDS, CodeCommit, VPC, Route 53, EKS, Cloudwatch, IAM
- **Funció:**
 - **CodeCommit:**

AWS CodeCommit s'ha fet servir com a repositori de control de versions, on es realitzen tots els commits del codi de l'aplicació. Aquest servei permet la col·laboració eficient entre els desenvolupadors i integra un pipeline de CI/CD. Això significa que cada vegada que es puja un canvi a la branca principal (master), es desencadena automàticament el pipeline de CI/CD, assegurant que el codi es compili, es testii i es desplegui sense problemes.
 - **Amplify:**

S'ha utilitzat per desplegar el frontend de l'aplicació de manera senzilla i eficient. Aquest servei d'AWS facilita la integració contínua i el desplegament automàtic de les actualitzacions del frontend, permetent una gestió àgil dels canvis en el codi. Gràcies a Amplify, qualsevol actualització en el frontend es desplega automàticament quan es fa un push a la branca master.
 - **VPC:**
 - **Aïllament de Xarxa:** Proporciona un entorn de xarxa aïllat dins del núvol, on es despleguen tots els recursos del núvol de l'aplicació. Això millora la seguretat en permetre que només els serveis dins de la VPC puguin comunicar-se entre si directament.
 - **Subxarxes:** Es poden definir subxarxes privades i públiques per controlar l'accés als recursos. Els recursos que necessiten accés públic, com els servidors web, es col·loquen en subxarxes públiques, mentre que els recursos interns es mantenen en subxarxes privades.
 - **Control de Trànsit:** Utilitza ACL¹², que defineixen els permisos que té cada usuari o grup per gestionar les regles de trànsit d'entrada i sortida, garantint que només el trànsit autoritzat pugui accedir als recursos.
 - **Gateway d'Internet:** Permet als recursos dins de la VPC accedir a internet i viceversa, sempre que les regles de seguretat ho permetin.
 - **RDS¹³:**

Amazon RDS s'ha utilitzat per gestionar la base de dades PostgreSQL de l'aplicació. Aquest servei proporciona una base de dades gestionada que és tolerant a fallades i altament disponible. RDS simplifica tasques administratives com les còpies de seguretat, les actualitzacions de programari

¹² Access Control Lists

¹³ Relational Database Service

i la recuperació davant fallades, assegurant que les dades dels usuaris, reserves i vehicles estiguin sempre accessibles i segures.

- **Route 53:**

AWS Route 53 s'ha utilitzat per gestionar el DNS de l'aplicació. Aquest servei permet dirigir el trànsit de xarxa a diferents recursos d'AWS. Gràcies a Route 53, es pot assegurar que les sol·licituds dels usuaris s'encaminen correctament als serveis corresponents.

- **EKS¹⁴:**

Elastic Kubernetes Service s'ha utilitzat per orquestrar els contenidors de l'aplicació. EKS facilita el desplegament, la gestió i l'escalabilitat de l'aplicació, permetent ajustar els recursos segons la demanda. Tot i que actualment l'aplicació s'utilitza internament, EKS permetrà escalar-la fàcilment quan estigui llesta per al públic. A més, s'han utilitzat scripts de Bash per permetre la flexibilitat de migrar l'aplicació a una altra plataforma si fos necessari.

- **CloudWatch:**

Amazon CloudWatch s'ha utilitzat per monitorar els recursos i l'aplicació en temps real. Aquest servei recull i fa el seguiment de mètriques, registre de logs i configuració d'alarmes per assegurar que el rendiment de l'aplicació es mantingui òptim i qualsevol problema es detecti i es resolgui ràpidament.

- **IAM¹⁵:**

AWS Identity and Access Management s'ha utilitzat per gestionar els usuaris i grups, proporcionant un control detallat d'accés als serveis i recursos d'AWS. IAM assegura que només els usuaris autoritzats tinguin accés als recursos necessaris, millorant la seguretat de l'aplicació.

3.2.2 Descripció Detallada del Flux

3.2.2.1 Enviament de Sol·licitud pel Client

El client, mitjançant un navegador web o una aplicació, envia una sol·licitud HTTP a la URL de l'aplicació.

3.2.2.2 Recepció i Processament Inicial per NGINX

NGINX rep la sol·licitud a través del seu punt d'entrada públic. NGINX determina si la sol·licitud és per al frontend o per al backend. Per a sol·licituds d'API, la redirigeix al backend.

3.2.2.3 Autenticació i Autorització amb Spring Security

La sol·licitud arriba al backend, on Spring Security s'encarrega de l'autenticació. Si l'usuari no està autenticat, es retorna un error HTTP 401, i en cas d'estar-ho, es procedeix

¹⁴ Elastic Kubernetes Service

¹⁵ Identity and Access Management

amb l'autorització per assegurar-se que té els permisos adequats per accedir al recurs sol·licitat.

3.2.2.4 Processament en EKS

Els contenidors backend en EKS gestionen la lògica de la sol·licitud. Això pot incloure la lectura i escriptura de dades, la realització de càlculs, etc. Si es requereixen dades ràpides, el backend consulta Redis.

3.2.2.5 Accés Ràpid a Dades amb Redis

Redis proporciona dades emmagatzemades en memòria cau per reduir el temps de resposta i millorar el rendiment de l'aplicació.

3.2.2.6 Retorn de Resposta

Després de processar la sol·licitud, el backend envia la resposta de tornada a NGINX. NGINX rep la resposta del backend i la transmet de tornada al client, completant així el cicle de la sol·licitud.

3.3 Arquitectura i Model MVC

El patró MVC¹⁶ és un patró de disseny de software que separa les dades d'una aplicació (el model), la interfície d'usuari (la vista) i la lògica de control (el controlador). Aquesta separació ajuda a gestionar la complexitat en aplicacions de software, facilitant el manteniment i l'escalabilitat del codi.

En el nostre projecte, amb els requisits funcionals i no funcionals explicats anteriorment, l'ús del patró MVC ens dona els següents avantatges:

- **Separació de responsabilitats:** Cada component del MVC gestiona una part específica de la funcionalitat de l'aplicació. El model gestiona les dades i la lògica de negoci, la vista s'encarrega de mostrar aquestes dades, i el controlador actua com a intermediari entre el model i la vista, gestionant la lògica d'interacció. Això fa que el codi sigui més modular, fàcil de modificar i de mantenir.
- **Facilitat en el testatge:** Cada component pot ser testejat independentment. Això és particularment útil en els requisits no funcionals on es requereix que el sistema sigui fiable i fàcil de mantenir (R6).
- **Flexibilitat en el desenvolupament:** Els desenvolupadors poden treballar en components diferents sense interferir els uns amb els altres, millorant l'eficiència en l'equip.
- **Fàcil d'adaptar a noves necessitats:** A mesura que els requisits del projecte evolucionen, com poden ser l'addició de noves funcionalitats (com les definides en els requisits funcionals RF1-RF19), MVC permet una incorporació més senzilla sense alterar tot el sistema.
- **Millora de l'experiència d'usuari:** Amb una vista clarament separada, es pot garantir una millor experiència d'usuari (UI¹⁷), complint amb els requisits d'usabilitat i rendiment (RNF1 i RNF2).

¹⁶ Model View Controller

¹⁷ User Interface

3.4 Descripció i Utilitat de cada Pàgina

3.4.1 Gestor de Reserves

Aquesta pàgina mostra una taula amb totes les reserves registrades en el sistema. La taula inclou columnes com el número de reserva, data de reserva, origen, destinació, estat de la reserva i altres detalls pertinents. També té una pestanya per afegir una nova reserva manualment i una altra per editar una reserva existent.

La necessitat d'aquesta pàgina va néixer, ja que Avantcab, que és l'empresa de taxis, treballa amb reserves programades amb antelació, pel fet que funcionen amb agències. Aquest sistema els permet conèixer les reserves que tindran uns dies abans, facilitant la planificació i la gestió eficient del servei.

Aquesta pàgina serveix per a visualitzar i gestionar les reserves, així com per afegir una reserva manualment. Tot i que normalment no és necessari afegir reserves manualment perquè el sistema té una pàgina per importar reserves automàticament de diversos importadors, aquesta funcionalitat és útil per a casos puntuals. També permet editar una reserva en cas de canvis d'última hora que requereixin intervenció manual, assegurant així la flexibilitat i la capacitat de resposta a situacions imprevistes. A part d'això també serveix per editar alguna reserva en cas que hi hagi un cas puntual a última hora que requereixi d'intervenció manual.

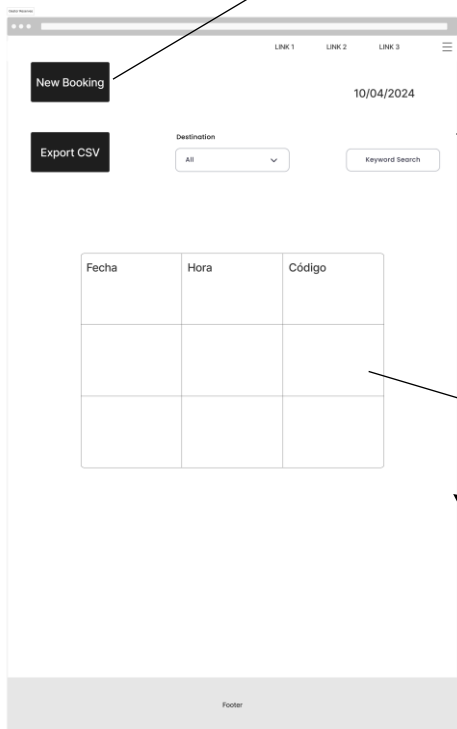


Figura 10. Pàgina de visualització de reserves

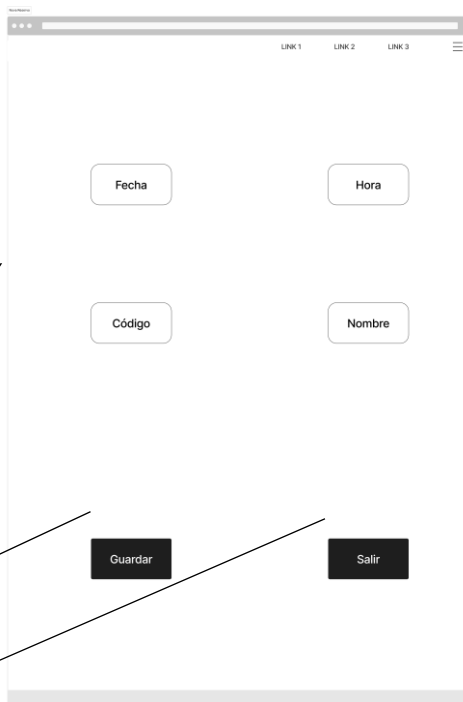


Figura 12. Pàgina per crear una nova reserva

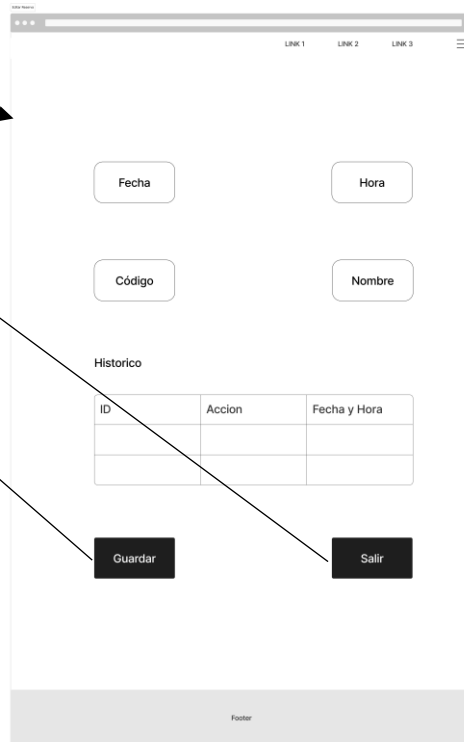


Figura 11. Pàgina per editar una reserva existent

3.4.2 Gestor de Vehicles

Aquesta pàgina permet visualitzar i gestionar els vehicles de la flota. Mostra una llista de vehicles amb detalls com matrícula, marca, model, any, estat, i quilometratge. També proporciona funcionalitats per afegir nous vehicles, editar els existents i eliminar aquells que ja no són necessaris.

Aquesta pàgina és important perquè Avantcab té diferents vehicles de diferents mides, i depenent del nombre de passatgers d'una reserva es necessitarà un tipus de vehicle o un altre. A més, alguns vehicles disposen de cadiretes per a bebès, cosa que és essencial per a determinades reserves. Així, la pàgina facilita la gestió eficient de la flota, assegurant que es poden assignar els vehicles adequats segons les necessitats específiques de cada reserva.

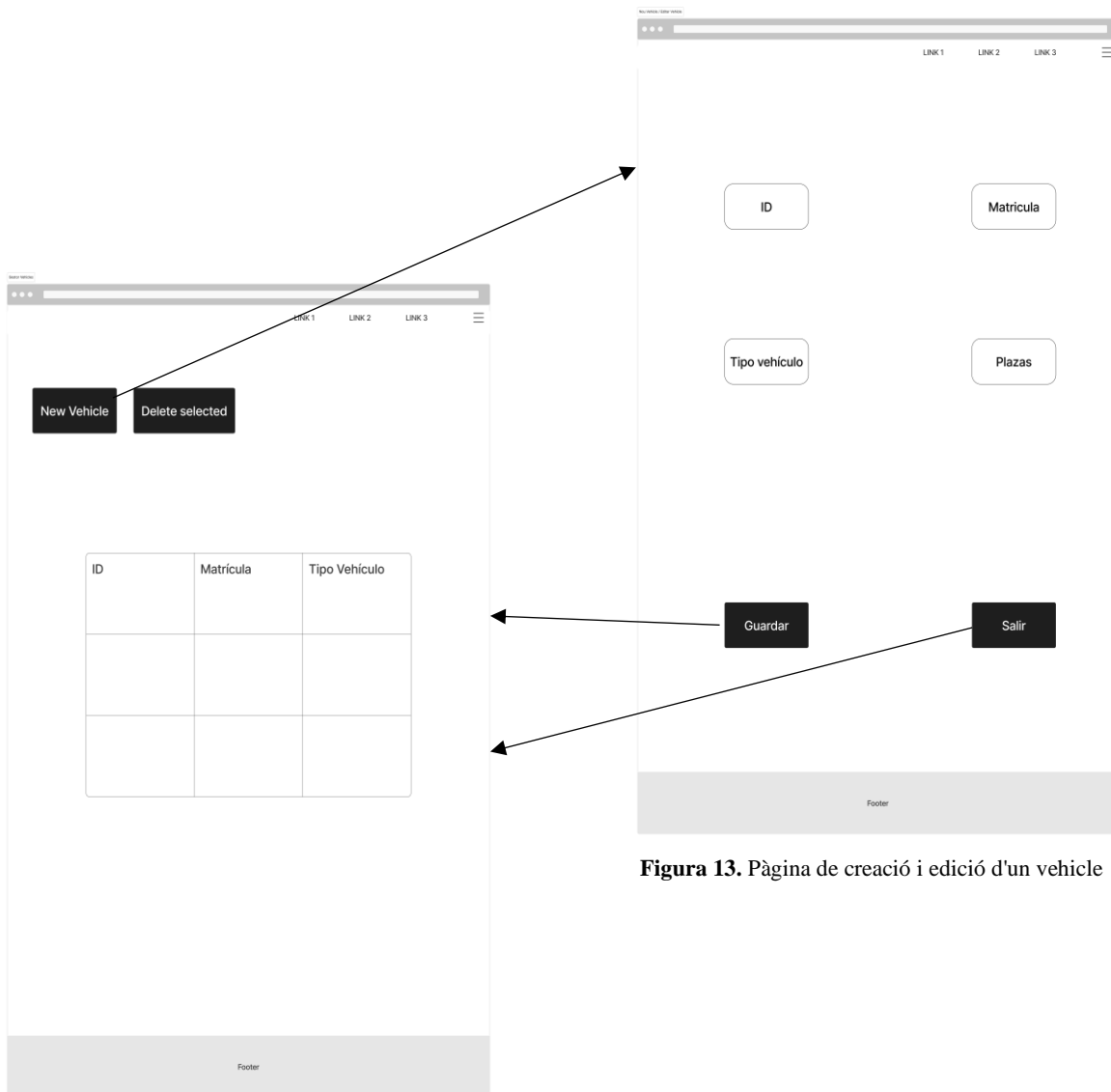


Figura 14. Pàgina de visualització de vehicles

Figura 13. Pàgina de creació i edició d'un vehicle

3.4.3 Importador de Reserves

La necessitat d'aquesta pàgina va aparèixer perquè era necessària una eina per poder importar les reserves de diferents proveïdors i agències al nostre sistema. La pàgina permet importar un fitxer de reserves en formats CSV, JSON o XML, i mostra les reserves que s'importaran. Si tot està correcte, permet importar-les al sistema un cop revisades i clicat el botó d'importar.

Aquesta funcionalitat ajuda a reduir el temps i l'esforç que requeriria importar tots els serveis manualment, que poden arribar a ser centenars per dia. Automatitzar aquest procés minimitza els errors humans, assegurant que totes les reserves es gestionen de manera precisa.

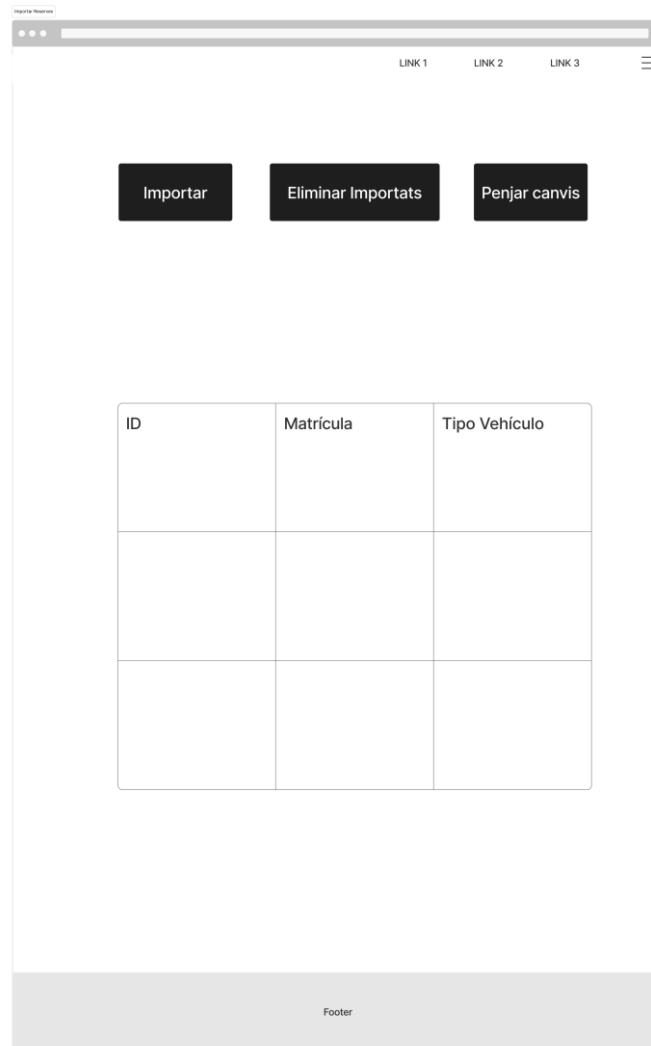


Figura 15. Pàgina per importar noves reserves

3.4.4 Pàgina de Localitzacions

Avantcab treballa amb diferents zones, i cada zona té un preu de servei. Per tant, la ubicació d'un servei determinarà el seu cost, ja que hi ha serveis que són a 10 minuts de l'aeroport i d'altres que són a l'altra punta de l'illa, a una hora de distància en cotxe.

Amb aquesta pàgina, es crea un sistema que permet dibuixar diferents polígons per definir zones i guardar-los a una base de dades. També permet editar, eliminar i veure tots els polígons, així com una fàcil navegació pel mapa de polígons. Això facilita la gestió de les tarifes segons la zona, assegurant una facturació precisa basada en la distància i la ubicació del servei. També ens serveix per a definir aquelles zones on l'empresa opera, i si tenim una zona la qual no coincideix cap polígon, aquesta zona representarà que l'empresa no hi opera, i per tant no tindrà un preu definit.

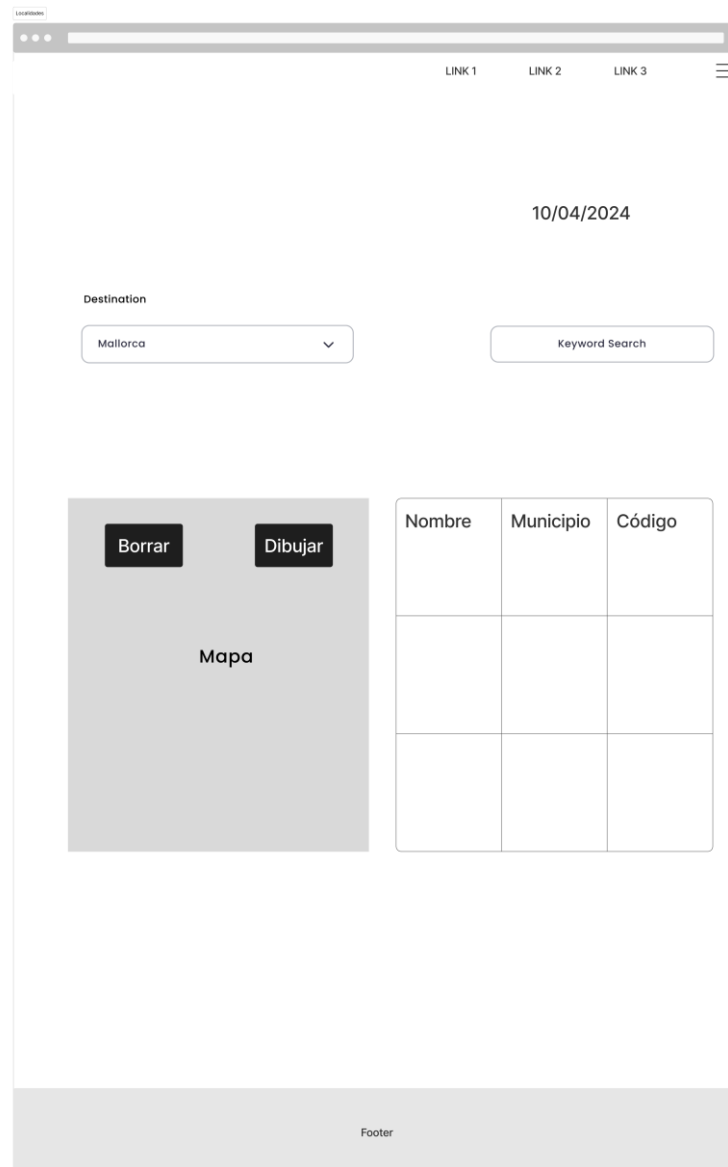


Figura 16. Pàgina per veure i crear noves localitats

3.5 Implementació de la Base de Dades en Postgres

A continuació, es descriuen les taules i els camps rellevants de la base de dades per implementar les pàgines de reserves, vehicles, importadors i localitzacions en l'aplicació ConectaDrive. Han estat omeses altres taules que apareixen al diagrama, però que no eren rellevants per a les pàgines mostrades en aquest TFG.

3.5.1 Diagrama de la Base de Datos

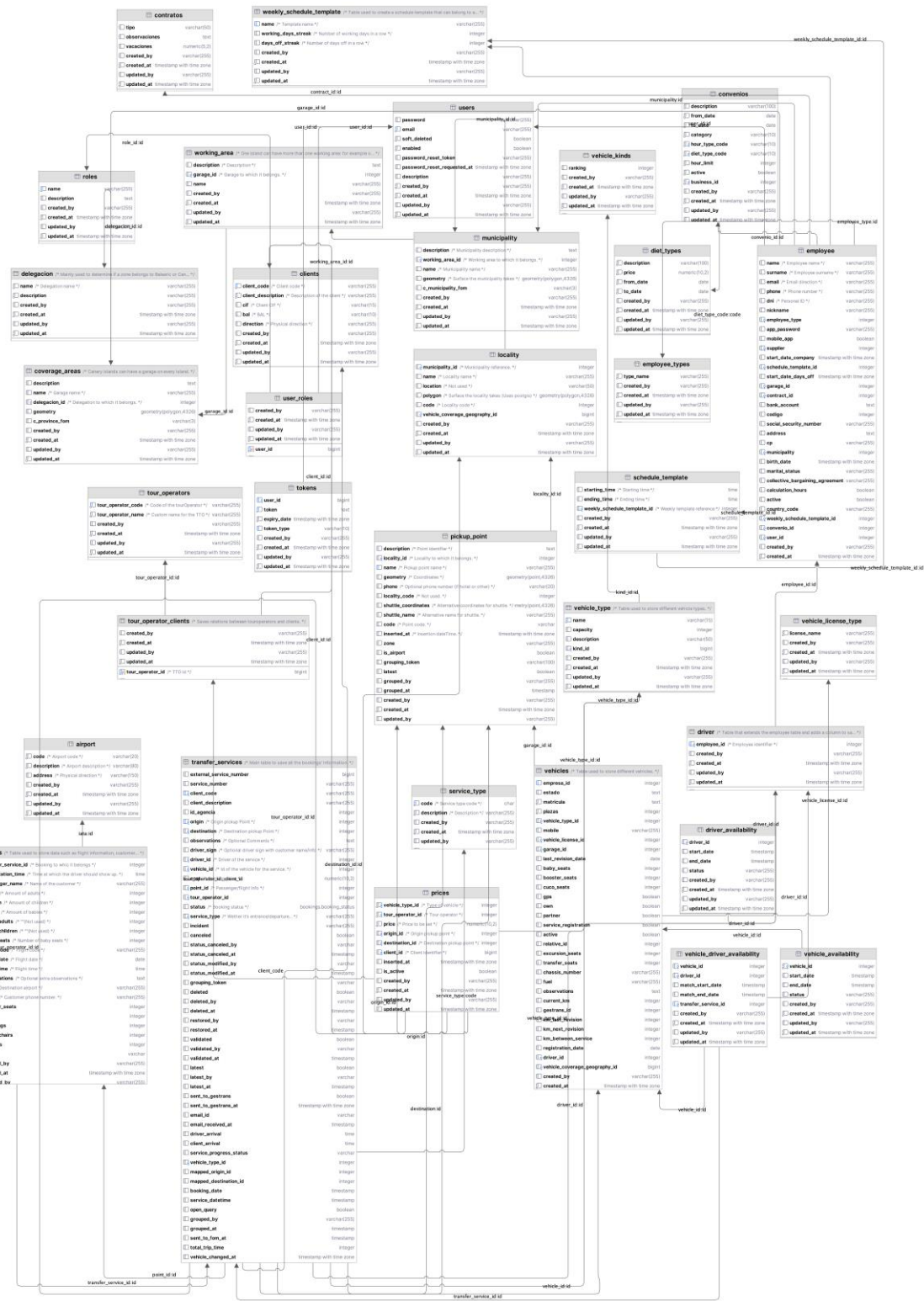


Figura 17. Base de datos de Conectadrive

3.5.2 Descripció dels Camps de la Base de Dades

3.5.2.1 Taules per la Pàgina de Reserves

3.5.2.1.1 Taula *transfer_services*

- **id (PK¹⁸):** Identificador únic de la reserva.
- **created_by (FK):** Referència a l'usuari que ha creat la reserva. És una columna d'auditoria que tenen totes les taules per saber quin usuari ha fet quina acció.
- **driver_id (FK):** Referència al conductor assignat.
- **vehicle_id (FK):** Referència al vehicle assignat.
- **origin_id (FK):** Referència al punt de recollida.
- **destination_id (FK):** Referència al punt de deixada.
- **service_date_time:** Hora i data de recollida.
- **status:** Estat de la reserva (per exemple, pendent, confirmada, completada, cancel·lada).

3.5.2.1.2 Taula *users*

- **id (PK):** Identificador únic de l'usuari.
- **name:** Nom de l'usuari.
- **email:** Correu electrònic de l'usuari.

3.5.2.1.3 Taula *drivers*

- **id (PK):** Identificador únic del conductor.
- **name:** Nom del conductor.
- **license_number:** Número de llicència del conductor.

3.5.2.1.4 Taula *vehicles*

- **id (PK):** Identificador únic del vehicle.
- **license_plate:** Matrícula del vehicle.
- **vehicle_type_id (FK):** Referència al tipus de vehicle.

3.5.2.1.5 Taula *points*

- **id (PK):** Identificador únic del punt (pot ser recollida o deixada).
- **name:** Nom del punt de recollida, és un nom més curt que la descripció i fàcilment identificable.
- **description:** Direcció física sencera del punt de recollida.
- **is_airport:** Serveix per indicar si el punt és un aeroport o no.

3.5.2.2 Taules per la Pàgina de Vehicles

3.5.2.2.1 Taula *vehicles*

- **id (PK):** Identificador únic del vehicle.
- **license_plate:** Matrícula del vehicle.
- **vehicle_type_id (FK):** Referència al tipus de vehicle.

¹⁸ Foreign Key (Clau Forana)

- **model:** Model del vehicle.
- **brand:** Marca del vehicle.
- **year:** Any de fabricació del vehicle.
- **status:** Estat del vehicle (per exemple, disponible, en manteniment).

3.5.2.2.2 *Taula vehicle_types*

- **id (PK):** Identificador únic del tipus de vehicle.
- **type_name:** Nom del tipus de vehicle (per exemple, sedan, SUV, furgoneta).

3.5.2.3 Taules per la Pàgina de Localitzacions

3.5.2.3.1 *Taula localities*

- **id (PK):** Identificador únic de la localitat.
- **name:** Nom de la localitat.
- **municipality_id (FK):** Referència al municipi al qual pertany la localitat.

3.5.2.3.2 *Taula municipalities*

- **id (PK):** Identificador únic del municipi.
- **name:** Nom del municipi.
- **working_area_id (FK):** Referència a la taula de zones de treball a la qual pertany el municipi.

3.5.2.3.3 *Taula coverage_areas*

- **id (PK):** Identificador únic del garatge.
- **name:** Nom del garatge.

3.5.3 *Resum de la Relació entre les Taules*

- **Reserves:** Les reserves estan relacionades amb els usuaris, conductors, vehicles i punts (recollida i deixada).
- **Vehicles:** Cada vehicle té un tipus associat i pot estar en diversos estats operacionals.
- **Localitzacions:** Les localitats estan organitzades en municipis, que al seu torn pertanyen a províncies.

4 Implementació

4.1 Descripció de diverses pàgines

A continuació es detallarà la implementació de diverses pàgines fetes amb Vue, Typescript i Primevue. Per mantenir la claredat i la concisió d'aquest document, els fragments de codi presentats en aquesta secció són una selecció que representa les funcionalitats clau de l'aplicació. S'han omès alguns tipus específics de TypeScript i altres detalls de codificació per fer els exemples més accessibles i breus.

4.1.1 Pàgina de Login

De moment no tenim pàgina de registre, ja que és una aplicació que només s'utilitza internament, i per tant, creem els usuaris a mesura que ens arriben per així no deixar que es registrin usuaris no desitjats

Com podem veure, podem ficar l'usuari i la contrasenya com una pàgina de login normal i a part tenim el botó de Recordar i el de recuperar contrasenya.

Quan l'usuari fa el login amb l'opció de recordar, el pròxim cop que es faci login l'usuari tindrà ja l'usuari i la contrasenya escrites per a poder entrar.

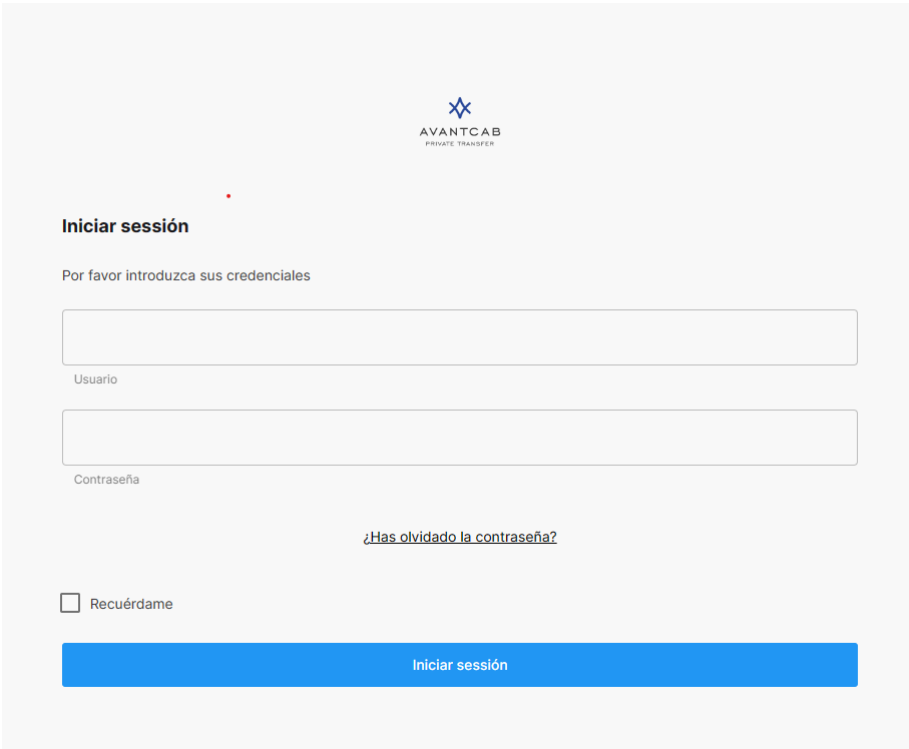


Figura 18. Pàgina d'inici de sessió

4.1.1.1 Estructura de dades i inicialització

Per a gestionar les credencials de l'usuari de manera reactiva, s'utilitzen referències de Vue (`ref`), que permeten actualitzar la UI automàticament quan les dades canvien. Els següents camps són inicialitzats:

```
const inputs = ref({
```

```

    email: '',
    password: '',
    remember: false
  })

```

4.1.1.2 Carregament i assignació de dades

Dins la funció `onMounted`, que s'executa un cop el component és carregat, les dades s'obtenen de `authStore`, un magatzem centralitzat amb Pinia que gestiona l'estat de l'aplicació:

```

onMounted(() => {
  const roles = authStore.roles;

  inputs.value.email = authStore.email;
  inputs.value.password = authStore.password;
  inputs.value.remember = !!inputs.value.email;

  checkRole(roles);
});

```

Les credencials s'assignen a les variables reactives si l'usuari anteriorment havia seleccionat l'opció de "recordar-me". A més, es verifica el rol de l'usuari per determinar l'accés permès.

4.1.1.3 Funció d'Autenticació

La funció `onSubmit` gestiona el procés d'inici de sessió:

```

const onSubmit = async () => {
  loading.value = true;
  try {
    const response = await authStore.login({
      email: inputs.value.email,
      password: inputs.value.password
    });
    checkRole(response.roles);
    rememberCredentials();
  } catch (err) {
    httpErrorHandler(err);
  }
  loading.value = false;
};

```

Aquí, es realitza una crida asincrònica a `authStore.login` per autenticar l'usuari. Si l'autenticació és exitosa, es comproven els rols de l'usuari i, si correspon, es recorden les seves credencials.

4.1.1.4 Gestió de Credencials Recordades

Finalment, la funció `rememberCredentials` decideix si guardar o no les credencials de l'usuari:

```
const rememberCredentials = () => {
  if (inputs.value.remember) {
    authStore.email = inputs.value.email;
    authStore.password = inputs.value.password;
  } else {
    authStore.email = '';
    authStore.password = '';
  }
};
```

Aquesta part del codi es responsabilitza de mantenir o esborrar les credencials de l'usuari al magatzem, basant-se en si l'usuari ha activat l'opció "recordar-me".

4.1.2 Recuperar Contrasenya

Aquesta pàgina serveix per recuperar la contrasenya en cas que l'usuari no la recordi, el procés és el següent:

- L'usuari fica el seu correu on vol que li arribi l'enllaç per modificar la contrasenya.
- Se li notifica a l'usuari que s'ha enviat un correu per recuperar la contrasenya.
- A l'usuari li arriba un correu amb l'enllaç per recuperar la contrasenya.
- L'usuari clica a l'enllaç i el porta a una pàgina on pot modificar la contrasenya per una nova.

CONECTADRIVE
DRIVING THE FUTURE

¿Ha olvidado la contraseña?

Por favor introduzca la dirección de correo electrónico que usó para crear su cuenta y le enviaremos un enlace para restablecer su contraseña

Usuario

Enviar

Figura 19. Pàgina contrasenya perduda

¿Ha olvidado la contraseña?

Gracias, le hemos enviado un correo electrónico. Por favor revisar su carpeta de correo no deseado.


Si no recibe ningún correo electrónico, es posible que no tenga una cuenta registrada con esta dirección. Si es el caso por favor contacte con la oficina.

Figura 20. Pàgina contrasenya perduda després de ficar el correu


N noreply@conectadrive.com
Password reset request
Para:


To reset your password, click the link below:
<https://conectadrive.com/reset-password?token=f1b74cc2-b918-4470-8b86-8ef6527f85fa>

Figura 21. Email per recuperar la contrasenya



Reseteo de contraseña

Contraseña 

Repita la Contraseña 

Enviar

Figura 22. Pàgina per ficar la nova contrasenya

4.1.3 Gestor de Reserves

El gestor de reserves és la funcionalitat principal de l'aplicació. És una pàgina on podem veure les reserves que s'han fet últimament, i on les podem filtrar per diferents paràmetres.

Reservas Día: 2024-05-15 (Hoy) + New Booking 15/05/2024

Export to CSV Destination: All Keyword Search

Fecha	Hora	Codigo	Referencia	Garaje	Origen	Destino	Vuelo	Nombre	Pax	Extras	Conductor	Vehiculo	Proveedor	Tipo Vehiculo
2024-05-18	13:50	S	-	-	Parque Santiago IV	Tenerife South Airport (Reina Sofía) (TFS)			4					4PRIVADO
2024-05-12	01:05	E	-	-	Tenerife South Airport (Reina Sofía) (TFS)	Parque Santiago IV	FR1346		4					4PRIVADO
2024-05-27	17:25	S	-	-	Mela Jardines del Teide - Adults only, Calle Agando, Adeje, Spain	Tenerife South Airport (Reina Sofía) (TFS)			2					4PRIVADO
2024-07-18	17:45	S	-	-	Coral Teide Mar	Tenerife South Airport (Reina Sofía) (TFS)			5					8VAN
2024-05-17	10:20	S	-	-	Barcelo Tenerife, Urbanizacion San Blas, Oasis del Sur, Spain	Tenerife South Airport (Reina Sofía) (TFS)			2					4PRIVADO
2024-07-16	16:20	S	-	-	Hotel Best Tenerife, Calle Antonio Domínguez Alfonso, Playa de las Americas, Spain	Tenerife South Airport (Reina Sofía) (TFS)			2					4PRIVADO
2024-10-03	02:00	E	-	-	Tenerife South Airport (Reina Sofía) (TFS)	HOWMA Santa María	FR8693		8					8VAN
2024-05-17	11:35	E	-	-	Tenerife South Airport (Reina Sofía) (TFS)	Mela Jardines del Teide - Adults only, Calle Agando, Adeje, Spain			2					4PRIVADO
					Tenerife South Airport	Barcelo Tenerife, Urbanizacion San Blas								

Figura 23. Pàgina del gestor de reserves

4.1.3.1 Filtres

Dins del sistema de gestió de reserves, els filtres juguen un paper crucial permetent als usuaris visualitzar i gestionar les reserves de manera eficient. A continuació, detallarem dos dels filtres més rellevants i tècnicament interessants implementats en l'aplicació. Aquests filtres han estat seleccionats per la seva complexitat i el seu impacte directe en la funcionalitat de l'aplicació.

4.1.3.1.1 Filtre de Data

El filtre de data, serveix per filtrar les reserves en un específic rang de dates. Per exemple, si seleccionem una data entre el 01/04/2024 i el 01/05/2024 veurem una taula només amb dates amb aquest rang.

Fecha	Hora	Codigo	Referencia	Garaje	Origen	Destino
2024-05-1			H1869948/2	-		
2024-05-1			H1869381/1	-		
2024-05-13	16:35	S	H1869381/2	-		
2024-05-15	21:50	E	H1869900/1	-		
2024-05-13	09:15	E	H1881315/1	-		
2024-05-11	20:30	E	H1870505/1	Palma Central		Hotel Hoposa Daina

Figura 24. Filtre de data al gestor de reserves

Fecha	Hora	Codigo	Referencia	Garaje	Origen	Destino	Vuelo
2024-05-07	22:00	E	23519	-	FUE	GLOBALES COSTA TROPICAL (CALETA FUSTE)	I2 3902
2024-05-07	22:25	E	H1889853/1	-	Majorca (Palma de Mallorca Airport) (PMI)	Protur Alicia Hotel, Carrer na Llambies, Cala Bona, Spain	JTD691
2024-05-07	23:35	S	H1889639/2	-	BLUESEA Cala Millor, Carrer Caravel·les, Cala Millor, Spain	Majorca (Palma de Mallorca Airport) (PMI)	
2024-05-07	23:55	E	H1862964/1	-	Majorca (Palma de Mallorca Airport) (PMI)	Sol Barbados, Avinguda Notari Alemany, Calvia, Spain	EZY7262
2024-05-08	00:25	E	H1865276/1	-	Majorca (Palma de Mallorca Airport) (PMI)	Hotel Cala d'Or, Avinguda de Belgica, 49, 07660 Cala d'Or, Spain	3445

Figura 25. Hora filtrada i ordenada al gestor de reserves

Inicialment, els filtres de reserves es realitzaven directament al frontend utilitzant JavaScript. Aquesta aproximació permetia filtrar únicament les dades ja presentades a l'usuari. No obstant això, amb l'augment del nombre de reserves, vam constatar la necessitat de realitzar un filtrat més exhaustiu que abastés tot el conjunt de dades. En resposta a aquesta necessitat, vam desenvolupar una API al backend específicament dissenyada per filtrar les reserves de manera global. Així, cada vegada que un filtre és aplicat des del frontend, aquest recupera les dades ja filtrades des del backend, millorant significativament la gestió i l'eficiència de l'aplicació.

El fragment de codi a continuació mostra la configuració del component `<Column>` dins d'una taula que utilitza `<PrimeVue>` per gestionar la visualització de les dates de servei de les reserves.

```
<Column
  field="serviceDate"
  header="Fecha"
  style="min-width: 10rem"
  :show-filter-menu="true"
  sortable
  data-type="date"
  :show-filter-match-modes="false"
  :show-filter-operator="false"
  :show-add-button="false">
```

- **field:** Aquest atribut identifica el camp de dades que la columna ha de representar, en aquest cas, `serviceDate`.
- **header:** Defineix el títol de la columna, "Fecha", que es mostra als usuaris.
- **sortable:** Permet als usuaris ordenar les dades basant-se en aquesta columna.
- **data-type:** Especifica que el tipus de dades d'aquesta columna és una data, el que pot influir en com es gestionen les operacions de filtrat i ordenació.
- **show-filter-menu, show-filter-match-modes, show-filter-operator, show-add-button:** Aquestes propietats controlen elements de la UI com menús de filtrat i botons, adaptant la interfície a les necessitats específiques de filtrat.

```
<template #filter="{ }">
```

```

<Calendar
  v-model="dateRangeForm"
  selection-mode="range"
  placeholder="Selecciona un rango de fechas"
  date-format="yy-mm-dd"
  show-icon
  :show-on-focus="false"
  :manual-input="false"
/>
</template>

```

4.1.3.1.2 Calendar

Aquest component de PrimeVue permet als usuaris seleccionar un rang de dates utilitzant un calendari interactiu. La propietat `v-model` està enllaçada amb `dateRangeForm`, que és gestionat en l'estat central de Vue, permetent així que les seleccions de l'usuari actualitzin instantàniament el model de dades.

selection-mode: Configurat a "range" per permetre la selecció de rangs de dates, facilitant així els usuaris filtrar per períodes específics.

date-format: Especifica el format de la data, assegurant-se que les dades introduïdes o visualitzades segueixen una coherència formativa.

```

<template #filterapply="{ field, filterModel, filterCallback }">
  <Button
    type="button"
    @click="handleFilterButtonClick(field, filterModel,
filterCallback)"
  >Apply
</Button>
</template>

```

4.1.3.1.3 Template #filterapply

Conté un botó "Apply" que, quan es prem, executa la funció `handleFilterButtonClick`. Aquesta funció és responsable de processar i aplicar el filtre basat en els criteris especificats per l'usuari.

```

<template #filterclear="{ field, filterCallback }">
  <Button
    type="button"
    @click="handleClearFilterButtonClick(field, filterCallback)"
  >Clear
</Button>
</template>

```

4.1.3.1.4 Template #filterclear

Similarment, aquest template conté un botó "Clear" que executa la funció `handleClearFilterButtonClick`. Aquesta funció neteja el filtre aplicat al camp específic.

```
async function handleFilterButtonClick(field, filterModel,
filterCallback) {
  if (isColumnFilterModel(filterModel)) {
    buildfilterData(field, filterModel)
  } else if (isDataTableOperatorFilterMetaData(filterModel)) {
    buildfilterData(field, filterModel.constraints[0])
  }
  console.log(filterCriteria.value)
  fetchBookings(filterCriteria.value, currentPage.value,
totalElems.value)
}
```

Aquesta funció avalua el model de filtre (`filterModel`) i, depenent del tipus, construeix els criteris de filtre utilitzant `buildfilterData`.

Una vegada construïts els criteris, es fa una crida a `fetchBookings` per actualitzar les reserves mostrades segons els filtres aplicats.

```
function buildfilterData(field, filterModel) {
  if (!field || !filterModel) return null
  const mapping = filterMappings.value[field as keyof typeof
filterMapping]
  const modelValue = normalizedModelValue(field, filterModel)
  if (filterCriteria.value && mapping) {
    filterCriteria.value[mapping as keyof
TransferServiceColumnFilters] = modelValue
  }
}
```

Construeix els criteris de filtre basant-se en el camp i el model de filtre proporcionats. Aquesta funció ajuda a normalitzar els valors del model de filtre per a ser utilitzats eficaçment en consultes o altres operacions.

```
async function handleClearFilterButtonClick(field, filterCallback) {
  filterCallback()
  const mapping = filterMappings.value[field as keyof typeof
filterMapping]
  if (filterCriteria.value[mapping as keyof
TransferServiceColumnFilters] === undefined) return
  delete filterCriteria.value[mapping as keyof
TransferServiceColumnFilters]
  fetchBookings(filterCriteria.value, currentPage.value,
totalElems.value)
}
```

Neteja el filtre aplicat cridant `filterCallback` i eliminant el criteri de filtre de l'estat global. A continuació, actualitza les dades mostrades.

4.1.3.1.5 Filtre de Destinacions

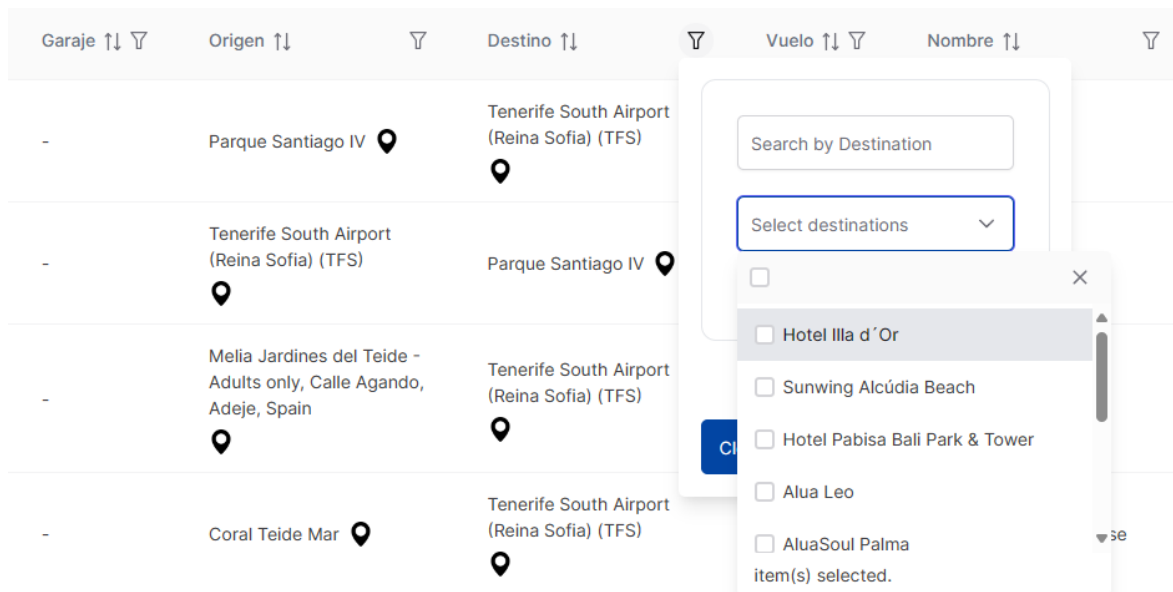


Figura 26. Pàgina de reserves filtre de destinacions

```
<Column
  field="destination.name"
  header="Destino"
  :show-filter-menu="true"
  style="min-width: 15rem"
  sortable
  :show-filter-operator="false"
  :show-add-button="false"
  :show-filter-match-modes="false"
>
```

`field="destination.name"`: Aquesta columna està vinculada al camp destination.name de les dades.

`header="Destino"`: El capçal de la columna es mostra com "Destino".

`sortable`: Permet que la columna sigui ordenable.

```
<input
  type="text" placeholder="Search by Destination"
  @input="onQueryChange(field, $event)">
<MultiSelect
  v-model="filterModel.value" :options="pointOptions"
```

```

        placeholder="Select destinations" display="chip" :max-
selected-labels="3"
        @input="onQueryChange(field, $event)">
    </MultiSelect>

```

```

function onQueryChange(field: string, event: Event) {
    const query = (event.target as HTMLInputElement).value
    console.log(field, query)
    debouncedSearch(field, query)
}

const debouncedSearch = debounce(async (field: string, query:
string) => {
    try {
        switch (filterMappings.value[field as keyof typeof
filterMapping]) {
            case 'origin':
            case 'destination':
                pointOptions.value = await
getPickupPointsByName(query).then((res) => res.filter((p) =>
p.groupingToken != null).map((p) => p.groupingToken!))
                break
            case 'serviceNumber':
                serviceNumbers.value = await
getMatchingServiceNumbersData(query).then((res) => res?.map((s) => s) ||
[])
                break
            case 'flightCode':
                flightCodeOptions.value = await
getMatchingFlightcodesData(query).then((res) => res?.map((s) => s) || [])
                break
        }
    } catch (error) {
        console.error(error)
    }
}, 300)

```

4.1.3.1.6 MultiSelect

Aquest component permet als usuaris seleccionar múltiples destinacions de la llista que s'actualitza dinàmicament basada en la consulta de l'usuari. Utilitza diverses propietats:

- **v-model="filterModel.value"**: Vincula el valor del component amb una variable de Vue que manté les destinacions seleccionades.

- **:options="pointOptions"**: Les opcions disponibles que l'usuari pot seleccionar, que s'actualitzen en resposta a la consulta de l'usuari.
- **display="chip"**: Mostra les opcions seleccionades en forma de "xips", que són etiquetes compactes.
- **:max-selected-labels="3"**: Limita el nombre de xips mostrats a tres, amb els addicionals resumits.
- **@input="onQueryChange(field, \$event)"**: Un gestor d'events que s'activa quan l'usuari interactua amb el component, passant l'event al mètode onQueryChange.
- **onQueryChange**: Aquest mètode s'activa quan l'usuari escriu en el camp de text o quan selecciona/deselecciona opcions en el MultiSelect. Captura l'entrada de l'usuari i la passa junt amb el camp associat ('destination' en aquest cas) al mètode **debouncedSearch**.
- **debouncedSearch**: Utilitza la tècnica de debouncing per limitar el nombre de crides a la funció, evitant així una sobrecàrrega de sol·licituds mentre l'usuari està escrivint.
 - Espera fins que l'usuari ha deixat d'escriure durant 300 mil·lisegons abans de processar la consulta.
 - Depenent del camp de dades, es fan crides a funcions específiques que obtenen les dades del servidor, com ara **getPickupPointsByName**, que cerca punts de recollida basats en el nom introduït.
 - Les dades obtingudes són després emmagatzemades en variables corresponents (com **pointOptions**), que després s'utilitzen per a actualitzar les opcions del MultiSelect.

4.1.3.2 Exportar a CSV

La funcionalitat d'exportar a CSV és una funcionalitat bastant simple, que el que fa és agafar totes les reserves mostrades actualment i exportar-les a un CSV amb els camps de la taula.

Si li donem clic al botó d'exportar a CSV, en aquest cas, ens genera el CSV que veiem a sota, amb el codi, la referència, l'origen, etc.

Export to CSV								Destination
Fecha	Hora	Codigo	Referencia	Garaje	Origen	Destino	Vuelo	
2024-05-18	13:50	S	H1864137/2	-	Parque Santiago IV	Tenerife South Airport (Reina Sofia) (TFS)		
2024-05-12	01:05	E	H1864137/1	-	Tenerife South Airport (Reina Sofia) (TFS)	Parque Santiago IV	FR1346	
2024-05-27	17:25	S	H1895491/2	-	Melia Jardines del Teide - Adults only, Calle Agando, Adeje, Spain	Tenerife South Airport (Reina Sofia) (TFS)		
2024-07-18	17:45	S	G2417384/2	-	Coral Teide Mar	Tenerife South Airport (Reina Sofia) (TFS)		

Figura 27. Gestor de reserves exportar a CSV

Codigo	Referencia	Origen	Destino	Vuelo
S	H1864137/2	Parque Santiago IV	Tenerife South Airport (Reina Sofia) (TFS)	
E	H1864137/1	Tenerife South Airport (Reina Sofia) (TFS)	Parque Santiago IV	FR1346
S	H1895491/2	Melia Jardines del Teide - Adults only, Calle Agando, Adeje, Spain	Tenerife South Airport (Reina Sofia) (TFS)	
S	G2417384/2	Coral Teide Mar	Tenerife South Airport (Reina Sofia) (TFS)	
S	H1895491/2	Barcelo Tenerife, Urbanizacion San Blas, Oasis del Sur, Spain	Tenerife South Airport (Reina Sofia) (TFS)	
S	G2395188/2	Hotel Best Tenerife, Calle Antonio Dominguez Alfonso, Playa de las Americas, Spain	Tenerife South Airport (Reina Sofia) (TFS)	
E	M1778774/1	Tenerife South Airport (Reina Sofia) (TFS)	HOVMA Santa María	FR8093
E	H1895491/1	Tenerife South Airport (Reina Sofia) (TFS)	Melia Jardines del Teide - Adults only, Calle Agando, Adeje, Spain	8033
E	H1895491/1	Tenerife South Airport (Reina Sofia) (TFS)	Barcelo Tenerife, Urbanizacion San Blas, Oasis del Sur, Spain	8039
E	M1778773/1	Tenerife South Airport (Reina Sofia) (TFS)	HOVMA Santa María	FR8093
E	G2417384/1	Tenerife South Airport (Reina Sofia) (TFS)	Coral Teide Mar	FR1121
T	N1416122/1	Port of Santa Cruz de Tenerife, Spain	Tenerife South Airport (Reina Sofia) (TFS)	
E	G2395188/1	Tenerife South Airport (Reina Sofia) (TFS)	Hotel Best Tenerife, Calle Antonio Dominguez Alfonso, Playa de las Americas, Spain	TOM8286
S	H1867551/2	Veil Mari Hotel & Resort, Carretera Arta-Alcudia, Can Picalof, Spain	Majorca (Palma de Mallorca Airport) (PMI)	
S	H1876311/2	Sol Lunamar Palmanova Apartamentos, Avinuda de Cas Saboners, Calvia, Spain	Majorca (Palma de Mallorca Airport) (PMI)	

Figura 28. CSV exportat del gestor de reserves

```

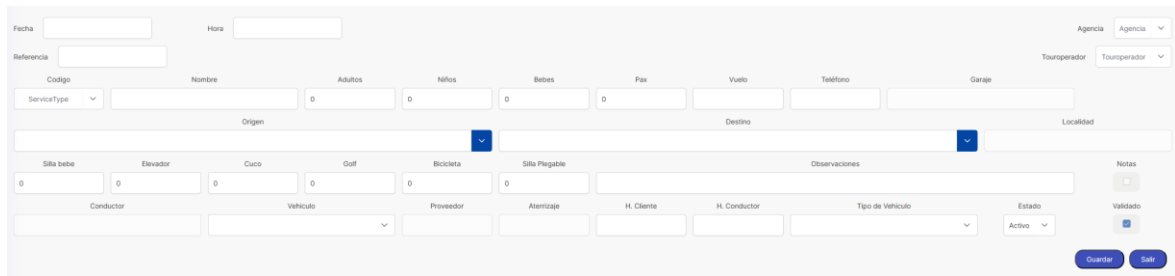
<Button
  icon="pi pi-external-link"
  label="Export to CSV"
  class="barElement"
  @click="exportCSV"
/>

const exportCSV = () => {
  dt.value.exportCSV()
}

```

En aquest cas, el codi és molt senzill, ja que PrimeVue ens ofereix aquesta funcionalitat amb el DataTable per defecte, i per tant, només necessitem un botó que quan el cliquis cridi a la funció `dt.value.exportCSV()` per exportar a CSV la informació que hi ha mostrant-se a la taula.

4.1.4 Nova Reserva / Editar Reserva



The screenshot shows a complex reservation form. At the top, there are fields for 'Fecha' (Date) and 'Hora' (Time). Below these are 'Referencia' (Reference) and 'Agencia' (Agency) dropdown menus. The main section contains several rows of input fields: 'Codigo' (Code), 'Nombre' (Name), 'Adultos' (Adults), 'Niños' (Children), 'Bebes' (Infants), 'Pax' (Passengers), 'Vuelo' (Flight), 'Telefono' (Phone), and 'Garaje' (Garage). There are also dropdown menus for 'ServiceType', 'Origen' (Origin), 'Destino' (Destination), and 'Localidad' (Location). A second row includes 'Silla bebe' (Baby seat), 'Elevador' (Lift), 'Cucio' (Cushion), 'Golf', 'Bicicleta' (Bicycle), 'Silla Pegable' (Foldable chair), 'Observaciones' (Observations), and 'Notas' (Notes). The bottom row features 'Conductor' (Driver), 'Vehiculo' (Vehicle), 'Proveedor' (Provider), 'Atarraje' (Attachment), 'H. Cliente' (Client H.), 'H. Conductor' (Driver H.), 'Tipo de Vehiculo' (Vehicle Type), 'Estado' (Status), and 'Validado' (Validated). At the bottom right, there are 'Guardar' (Save) and 'Salir' (Exit) buttons.

Figura 29. Pàgina per crear o editar una reserva

4.1.4.1 Comportament en carregar la pàgina

La pàgina per afegir una nova reserva o editar una existent és principalment un formulari amb components de PrimeVue, com podrien ser components per escollir la data, la hora, per seleccionar una o múltiples opcions, etc.

```
onMounted(async () => {  
  bookingId.value = Number(router.currentRoute.value.params.id)  
  if (bookingId.value) {  
    isLoading.value = true  
    const bookingData = await getBookingById(bookingId.value)  
    if (bookingData.serviceDateTime) {  
      const dateTime = new Date(bookingData.serviceDateTime)  
      form.serviceDate = dateTime.toISOString().split('T')[0] //  
Toma sólo la parte de la fecha (yyyy-mm-dd)  
      form.serviceTime = dateTime.toString().slice(0, 5) //  
Toma sólo la hora y minutos (HH:mm)  
    }  
    const statusHash = {  
      MODIFIED: 'Modificado',  
      ACTIVE: 'Activo',  
      CANCELLED: 'Cancelado'  
    }  
    bookingData.status = bookingData.status  
    ? { label: statusHash[bookingData?.status], value:  
bookingData?.status }  
    : { label: 'Activo', value: 'ACTIVE' }  
    isLoading.value = false  
    if (bookingData) {  
      Object.assign(form, bookingData)  
    }  
  }  
}
```

})

La funció ``onMounted`` s'utilitza per carregar les dades necessàries quan un component Vue es carrega inicialment. Aquesta funció és especialment important per a la pàgina de detall o edició de reserves, ja que assegura que les dades de la reserva seleccionada es carreguin correctament i es mostrin a l'usuari. A continuació es descriu el funcionament de la funció ``onMounted`` pas a pas.

1. **Extracció de l'ID de la Reserva:** Quan el component es carrega, la funció ``onMounted`` obté l'ID de la reserva dels paràmetres de la ruta actual. Si no hi ha id, vol dir que hem de crear una nova reserva i, per tant, no cal carregar res més.
2. **Indicació de Càrrega:** Es configura una variable d'estat per indicar que les dades estan en procés de càrrega. Això permet mostrar un indicador de càrrega a l'usuari mentre es recuperen les dades del servidor.
3. **Obtenció de Dades de la Reserva:** Es fa una crida a l'API per obtenir les dades de la reserva utilitzant l'ID obtingut anteriorment. Aquesta crida és asíncrona i espera a rebre la resposta del servidor.
4. **Processament de la Data i l'Hora:** Si les dades de la reserva inclouen informació sobre la data i hora del servei, aquesta informació es processa per separar la data i l'hora en formats adequats per al formulari. La data es formata com yyyy-mm-dd i l'hora com HH:mm.
5. **Mapatge de l'Estat de la Reserva:** La funció tradueix l'estat de la reserva a un format més comprensible per a l'usuari. Per exemple, estats interns com "MODIFIED" o "ACTIVE" es converteixen en etiquetes amigables com "Modificado" o "Activo".
6. **Finalització de la Càrrega:** Un cop s'han processat totes les dades, es configura la variable d'estat per indicar que la càrrega ha finalitzat. Això permet ocultar l'indicador de càrrega i mostrar les dades al formulari.
7. **Assignació de les Dades al Formulari:** Finalment, les dades de la reserva obtingudes es copien al formulari perquè l'usuari pugui veure-les i, si cal, modificar-les.

4.1.4.2 Descripció dels camps

```
<label class="mr-4" for="fecha">Fecha</label>
<Calendar
  id="vehicle"
  v-model="serviceDate"
  :error="errors.serviceDate"
  type="text"
  date-format="yy-mm-dd"
>
const serviceDate = computed({
  get: () => (form?.serviceDate ? String(form?.serviceDate) : ''),
  set: (value) => {
    const date1 = new Date(value)
    const formattedDate1 =
      date1.getFullYear() +
      1 +
```

```

        '-' +
        ('0' + (date1.getMonth() + 2)).slice(-2) +
        '-' +
        ('0' + date1.getDate()).slice(-2)
        form.serviceDate = formattedDate1
    }
})

```

Aquest fragment defineix una propietat computada `serviceDate` que actua com a interfície reactiva entre el formulari i el component `Calendar`. La propietat `computed` té dos mètodes:

- **get:** Recupera el valor de `form.serviceDate` com una cadena de text.
- **set:** Agafa una data (`value`), la converteix en un objecte `Date`, la formata a l'estil `yyyy-mm-dd` i actualitza `form.serviceDate`.

Funcionament General del Codi:

- **Seleccionar Data:** Quan l'usuari selecciona una data al `Calendar`, aquesta es converteix a una cadena de text amb el format `yyyy-mm-dd` i s'actualitza la propietat `form.serviceDate`.
- **Mostrar Errors:** Si hi ha errors de validació en la data, es mostren automàticament prop del component gràcies a la vinculació `:error="errors.serviceDate"`.

4.1.4.3 Enviament del formulari

```

const handleSubmit = async () => {
  errors.value = validate()
  if (Object.keys(errors.value).length > 0) {
    const [firstError] = Object.keys(errors.value)
    return showError(`${firstError} is required`, toast)
  }
  if (!isValidTimeFormat(form.serviceTime)) {
    form.serviceTime = formatTime(form.serviceTime)
  }
  if (!isValidTimeFormat(form.clientArrival)) {
    form.clientArrival = formatTime(form.clientArrival)
  }
  if (!isValidTimeFormat(form.driverArrival)) {
    form.driverArrival = formatTime(form.driverArrival)
  }
  if (form.status?.value) form.status = form.status.value

  if (form.id) {
    return updateBooking(form as TransferService)
  }
}

```

```

    createBooking(form as TransferService)
  }

```

Aquesta funció es crida quan l'usuari envia el formulari de reserva i s'encarrega de validar les dades, assegurar-se que estan en el format correcte i després realitzar la crida API corresponent. A continuació es detalla el funcionament de la funció:

1. Es crida a la funció `validate` per validar totes les dades del formulari. Aquesta funció retorna un objecte que conté qualsevol error de validació. Si hi ha errors (és a dir, si l'objecte `errors.value` no està buit), es mostra un missatge d'error a l'usuari amb la funció `showError` i la funció `handleSubmit` es deté aquí.
2. Assegurem que els camps de temps (`serviceTime`, `clientArrival` i `driverArrival`) estan en un format vàlid. Si no ho estan, és formaten els camps utilitzant la funció `formatTime`.
3. Finalment, es comprova si el formulari té un id. Si en té, vol dir que és una actualització d'una reserva existent i es crida la funció `updateBooking`. Si no en té, es tracta d'una nova reserva i es crida la funció `createBooking`.

4.1.4.4 Agrupació de Reserves

Per treballar amb diferents proveïdors, es pot donar el cas que tinguem la mateixa reserva duplicada en dos d'ells. Per exemple, podríem tenir la mateixa reserva tant a Booking com a Hoppa, de manera que, quan les importem, s'importen les dues alhora. Per solucionar aquest problema, hem creat una pàgina de seguiment que agrupa les reserves duplicades com si fossin una sola. Aquesta pàgina permet marcar una reserva com l'última i, per tant, la més actualitzada. També permet cancel·lar, validar i eliminar reserves.

Aquesta finestra de seguiment apareix a la pàgina d'editar reserves, just a sota del formulari principal. Quan es clica en una reserva, la informació detallada apareix a sota del formulari, mostrant qui ha realitzat l'acció i quan s'ha fet. Això facilita la gestió de les reserves i assegura que sempre es treballi amb la informació més actualitzada.

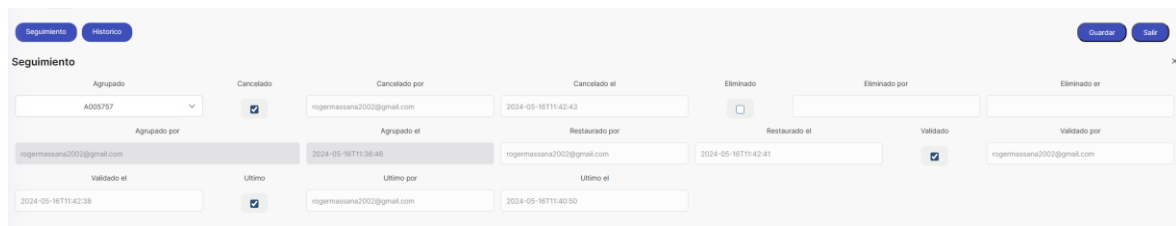


Figura 30. Agrupar reserves

4.1.4.5 Històric d'una Reserva

La finestra d'històric també apareix a la pàgina d'editar reserves. Si la finestra de seguiment està oberta, es tanca automàticament quan s'obre la finestra d'històric. Aquesta finestra serveix per visualitzar l'històric complet de la reserva, incloent-hi quan es va crear, totes les edicions que s'han fet i qui les ha realitzat durant la duració de la reserva.

Les modificacions es mostren una a una en una taula. Si hi ha més d'una modificació en el mateix moment, es mostren un sota l'altre amb la mateixa data. Això permet als usuaris tenir una visió clara i detallada de totes les accions realitzades sobre la reserva al llarg del temps.

ID Historico	Referencia	Acción	Fecha y Hora	Usuario	Campo Editado	Antiguo	Nuevo
14085	H1864137/2	INSERT	2024-05-10T09:29:12				
14086	H1864137/2	UPDATE	2024-05-16T11:39:58	rogemassana2002@gmail.com	points.passengerInfo.ad	4	5
14087	H1864137/2	UPDATE	2024-05-16T11:40:50	rogemassana2002@gmail.com	points.passengerInfo.ad	5	3
14088	H1864137/2	UPDATE	2024-05-16T11:40:50	rogemassana2002@gmail.com	points.passengerInfo.ch	0	1
14089	H1864137/2	UPDATE	2024-05-16T11:42:05	rogemassana2002@gmail.com	Cancelled		
14090	H1864137/2	DELETE	2024-05-16T11:42:21	rogemassana2002@gmail.com			
14091	H1864137/2	RESTORE	2024-05-16T11:42:22	rogemassana2002@gmail.com			
14092	H1864137/2	DELETE	2024-05-16T11:42:39	rogemassana2002@gmail.com			
14093	H1864137/2	RESTORE	2024-05-16T11:42:41	rogemassana2002@gmail.com			
14094	H1864137/2	UPDATE	2024-05-16T11:42:43	rogemassana2002@gmail.com	Cancelled		

Figura 31. Històric de les reserves

4.1.5 Pàgina de Gestió de Vehicles

Aquesta pàgina de gestió de vehicles proporciona una interfície centralitzada on els administradors poden visualitzar, afegir, editar i eliminar vehicles de manera senzilla i ràpida.

ConectaDrive gestiona una flota de vehicles que pertanyen a diferents empreses. Això implica que cal tenir una vista general i detallada de cada vehicle per assegurar-ne el correcte manteniment, ús i disponibilitat. La pàgina de gestió de vehicles facilita aquesta tasca permetent als administradors tenir un control complet sobre els vehicles, independentment de l'empresa a la qual pertanyin.

ID	Matrícula	Movil	Tipo Vehiculo	Licencia	Garaje	Plazas	Revisión	
<input type="checkbox"/>	ID.	Matrícula	Movil	Seleccione	Seleccione	Seleccione	Plazas	Fecha de re
<input type="checkbox"/>	226	IZ15MTW	+34 123 456 789	3PRIVADO	TAXI	Gran Canaria	4	2024-05-06
<input type="checkbox"/>	228	IZ15MTW	+34 123 456 789	3PRIVADO	TAXI	Gran Canaria	4	2024-05-06
<input type="checkbox"/>	231	TS18FNA	+34 123 456 789	3PRIVADO	TAXI	Gran Canaria	4	2024-05-06
<input type="checkbox"/>	232	TS18FNA	+34 123 456 789	3PRIVADO	TAXI	Gran Canaria	4	2024-05-06
<input type="checkbox"/>	242	DG46PKR	+34 123 456 789	3PRIVADO	TAXI	Gran Canaria	4	2024-05-07
<input type="checkbox"/>	175	YX81DGM	+34 123 456 789	3PRIVADO	TAXI	Fuerteventura	4	2024-05-06
<input type="checkbox"/>	49	ES001	559-768-5290 x09897	3PRIVADO	TAXI	Gran Canaria	7	2024-04-15
<input type="checkbox"/>	256	EQ68CTO	+34 123 456 789	3PRIVADO	TAXI	Gran Canaria	4	2024-05-07
<input type="checkbox"/>	147	KP82USN	+34 123 456 789	3PRIVADO	TAXI	Gran Canaria	4	2024-05-05
<input type="checkbox"/>	161	AU61IHM	+34 123 456 789	3PRIVADO	TAXI	Gran Canaria	4	2024-05-05

Figura 32. Gestor de vehicles

4.1.5.1 Vista general de vehicles

- **Llista de vehicles:** A la pàgina principal, es mostra una taula amb informació sobre els vehicles registrats. Els camps inclouen ID, matrícula, mòbil, tipus de vehicle, llicència, garatge, places, i data de revisió.
- **Accions:** A la part superior de la taula, hi ha opcions per afegir un nou vehicle o eliminar vehicles seleccionats. És possible seleccionar múltiples vehicles marcant les caselles de selecció corresponents.
- **Filtres:** Es poden utilitzar els filtres situats a la part superior de la taula per cercar vehicles específics per matrícula, mòbil, tipus de vehicle, llicència, garatge, o data de revisió.

Formulari "Ficha del vehículo" per crear un nou vehicle. Els camps són:

ID	Matrícula	Plazas	Excursion	Trasladados	Tipo vehículo	Licencia	Garaje
					Tipo vehículo	Tipo licencia	Tipo garaje
GPS	Activo	Propio	Asociado	Fomento			
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
Empresa propietaria	Matriculación	No Chasis	Combustible	Móvil coche	Observaciones	Km	
Proveedor							
Gestrans	Silla	Elevador	Cuoco	Ultima revisión	Km ultima revisión	Km entre revisión	Km prox. revisión

Botons: Guardar, Salir.

Figura 33. Crear un vehicle

4.1.5.2 Nou vehicle

- **Accés al formulari:** Quan es clica el botó "Nou", es redirigeix a aquesta pàgina on es pot omplir un formulari per afegir un nou vehicle.
- **Camps del formulari:** Els camps inclouen ID, matrícula, places, excursions, trasllats, tipus de vehicle, llicència, garatge, GPS, actiu, propi, associat, foment, empresa propietària, matriculació, número de xassís, combustible, mòbil del cotxe, observacions, quilometratge, última revisió, quilòmetres entre revisions, i altres detalls com seients per a nadons, elevadors, i cadires de rodes.
- **Guardar o cancel·lar:** Hi ha botons per guardar el nou vehicle o sortir del formulari sense guardar els canvis.

Formulari "Ficha del vehículo" per editar un vehicle. Els camps són:

ID	Matrícula	Plazas	Excursion	Trasladados	Tipo vehículo	Licencia	Garaje
228	1275MTW	4	4	4	PRIVADO	TAXI	Gran Canaria
GPS	Activo	Propio	Asociado	Fomento			
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
Empresa propietaria	Matriculación	No Chasis	Combustible	Móvil coche	Observaciones	Km	
Transfer Bissal, S.L.	2024-05-06	ES001	Gasolina	+34 123 456 789	Observaciones	10	
Gestrans	Silla	Elevador	Cuoco	Ultima revisión	Km ultima revisión	Km entre revisión	Km prox. revisión
4	4	4	4	2024-05-06	10	10	10

Botons: Calendario, Guardar, Salir.

Figura 34. Editar un vehicle

4.1.5.3 Editar vehicle existent

- **Accés a l'edició:** Quan es clica sobre una fila específica de la taula de la pàgina principal, es redirigeix a aquesta pàgina on es poden veure i editar els detalls del vehicle seleccionat. Per editar un vehicle s'ha de primer clicar al botó editar de dalt a la dreta, si no, els camps són només de lectura.
- **Camps prèviament emplenats:** Els camps del formulari estan prèviament emplenats amb la informació actual del vehicle seleccionat, permetent modificar les dades necessàries.
- **Guardar o cancel·lar:** Igual que amb el formulari de creació, hi ha opcions per guardar els canvis o sortir sense guardar-los.

4.1.6 Pàgina de Gestió de Localitats

La Figura 35 que hi ha a continuació ofereix una interfície intuïtiva i funcional que permet visualitzar, crear, editar i eliminar localitzacions, facilitant així l'administració de les zones de servei per a les empreses de VTC.

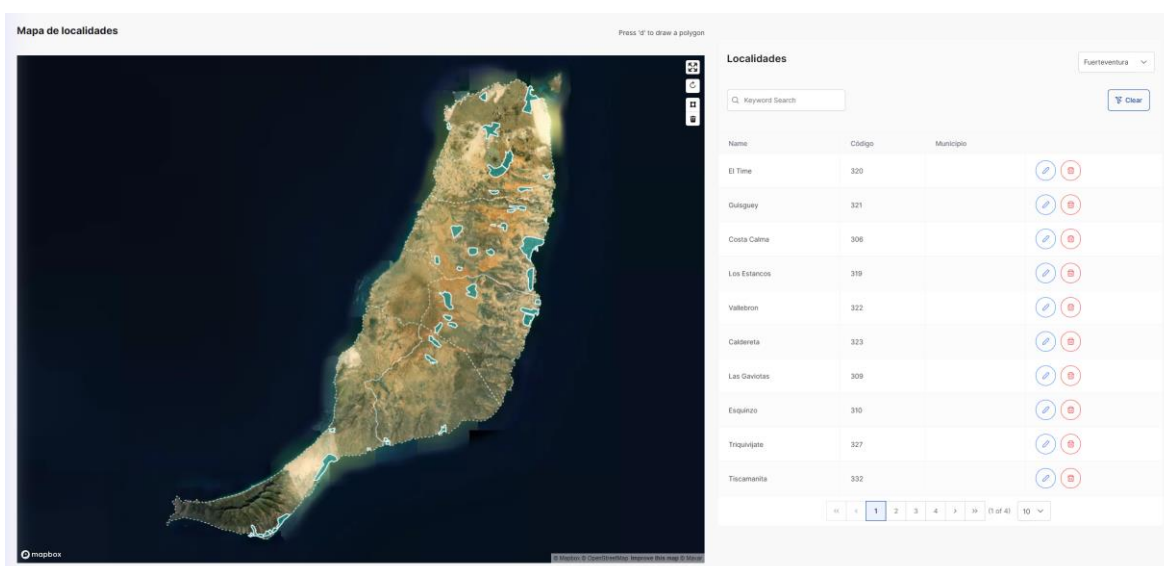


Figura 35. Gestió de Localitats

4.1.6.1 Visualització del Mapa

En accedir a la pàgina de gestió de localitzacions, els usuaris veuran un mapa interactiu de Fuerteventura, proporcionat per la plataforma Mapbox. Aquest mapa és molt detallat i permet una interacció completa. Les diferents localitzacions estan marcades amb polígons que en delimiten els límits geogràfics. Aquests polígons són visibles a simple vista, proporcionant una representació clara de les zones cobertes pel servei.

4.1.6.2 Taula de Localitzacions

A la dreta del mapa, hi ha una taula que llista totes les localitzacions disponibles. Aquesta taula mostra informació clau sobre cada localització, incloent-hi el nom, el codi i el municipi. Cada entrada de la taula té dues icones a la dreta: una per editar i una per eliminar. Aquestes opcions permeten als usuaris gestionar fàcilment les localitzacions directament des de la taula.

4.1.6.3 Interacció amb el Mapa

La integració entre la taula i el mapa és molt fluida. Quan un usuari selecciona una localització a la taula, el mapa fa zoom automàticament en aquesta àrea, facilitant la visualització detallada i l'edició de la localització. A més, hi ha una funció de cerca que permet als usuaris trobar ràpidament localitzacions específiques, millorant així l'eficiència de la gestió.

4.1.6.4 Gestió de Localitzacions

Els usuaris autenticats tenen diverses opcions per gestionar les localitzacions:

- **Edició de Localitzacions:** Fent clic a la icona de llapis al costat de qualsevol localització a la taula, els usuaris poden editar els detalls de la localització, com el nom, el codi i el municipi. Un cop s'han fet els canvis necessaris, aquests es guarden i es reflecteixen immediatament al mapa.
- **Eliminació de Localitzacions:** Fent clic a la icona de paperera, els usuaris poden eliminar una localització que ja no sigui necessària. El sistema demanarà una confirmació abans d'eliminar la localització per evitar errors accidentals.
- **Creació de Noves Localitzacions:** Els usuaris poden crear noves localitzacions dibuixant polígons directament al mapa. Per fer això, simplement seleccionen punts al mapa que formaran el contorn de la nova localització o poden clicar la lletra 'd'. Un cop dibuixat el polígon, l'usuari pot introduir els detalls de la nova localització (com el nom i la descripció) i guardar-los. El nou polígon es mostrarà immediatament al mapa, i la nova localització s'afegirà a la taula.

4.1.6.5 Excepcions i Gestió d'Errors

Tot i que el sistema està dissenyat per ser robust i fiable, poden ocórrer errors. Per exemple, si hi ha un problema amb el servei de mapes, el mapa pot no carregar-se correctament. També pot haver-hi problemes en la recuperació de dades de la base de dades, implicant que es mostrin les localitzacions. En aquests casos, el sistema mostrarà missatges d'error clars per informar l'usuari del problema i oferir possibles solucions.

4.1.7 Pàgina per Importar Reserves

Les empreses de VTC treballen amb una varietat de proveïdors que ofereixen serveis de reserves, com ara Hoppa, Booking i altres plataformes similars. Cadascun d'aquests proveïdors pot utilitzar formats de dades diferents per transmetre la informació de les reserves. Això pot incloure formats XML, JSON i CSV, entre d'altres. La diversitat de formats presenta un gran repte a l'hora d'integrar i gestionar aquestes dades de manera eficient.

Sense una eina específica per a la importació de reserves, les empreses es veurien obligades a realitzar processos manuals i repetitius per convertir i integrar aquestes dades, cosa que podria comportar errors humans, ineficiència i una pèrdua de temps considerable. Per tant, la pàgina d'importació de reserves automatitza aquest procés, assegurant que totes les dades de reserva es puguin importar fàcilment independentment del seu format original.

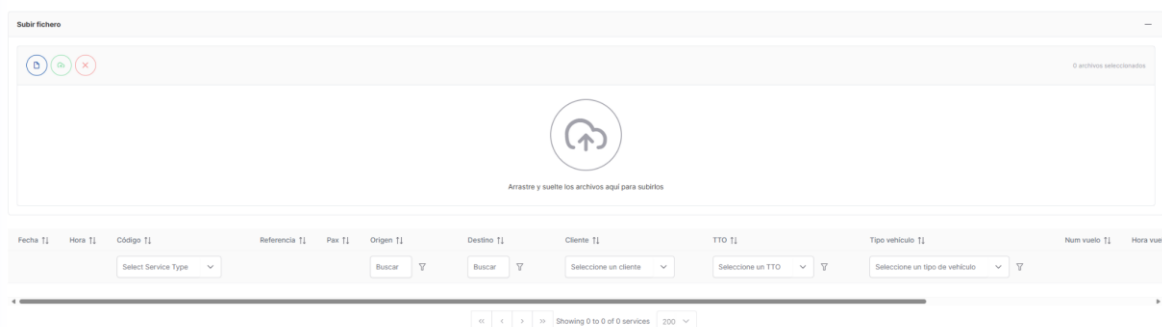


Figura 36. Pàgina d'importador de reserves

4.1.7.1 Càrrega de Fitxers

A la part superior de la pàgina, hi ha una àrea designada per a la càrrega de fitxers. Els usuaris poden seleccionar els fitxers que desitgen pujar, ja sigui arrossegant-los i deixant-los anar a l'àrea designada o seleccionant-los directament des de l'explorador de fitxers del seu dispositiu. Els formats acceptats són XML, JSON i CSV, els quals són comuns per a l'intercanvi de dades estructurades.

Un cop seleccionat el fitxer, el sistema procedeix a validar-lo automàticament. Aquesta validació és crucial per assegurar-se que el fitxer compleix amb els requisits de format. Si el fitxer no és compatible, el sistema mostrarà un missatge d'error, informant l'usuari que el format del fitxer no és vàlid. Si el fitxer és vàlid, el sistema el carrega i el guarda temporalment per a un processament posterior.

Després de completar la càrrega, el sistema mostra una confirmació a l'usuari, indicant que el fitxer s'ha carregat correctament i està llest per ser processat.

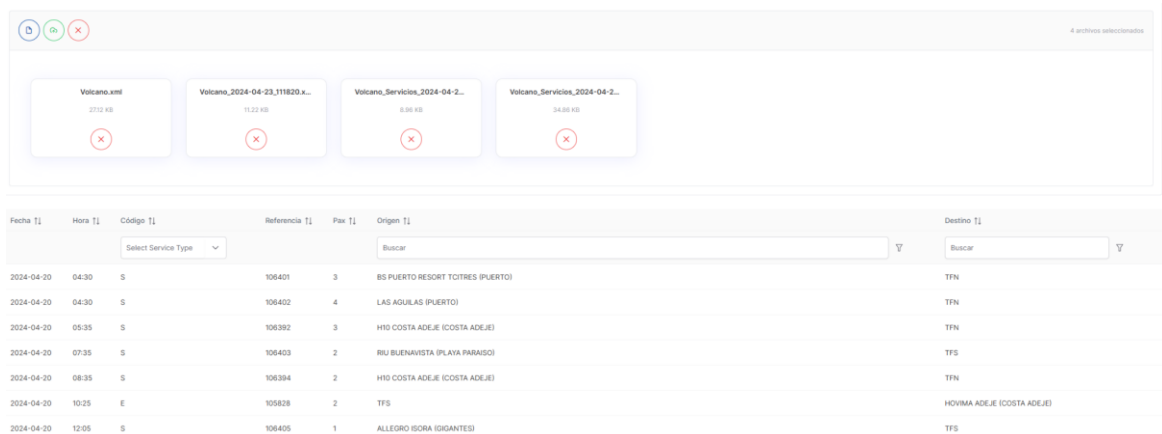


Figura 37. Visualització de les dades de reserva importades

4.1.7.2 Visualització de Dades de Reserva Importades

Un cop els fitxers han estat carregats, l'usuari pot accedir a la pàgina de visualització de dades de reserva. Aquesta pàgina permet veure les dades extretes dels fitxers carregats, presentades en una taula. Això és especialment útil per revisar i validar la informació abans de penjar-la definitivament a la base de dades central.

La taula de visualització mostra diversos detalls sobre les reserves, com ara la data, l'hora, el codi del servei, el número de referència, el nombre de passatgers (Pax), l'origen, la destinació, el client, el tipus de servei (TTO), el tipus de vehicle, el número de vol i l'hora de vol. Aquesta disposició permet als usuaris veure tota la informació rellevant de manera clara i ordenada, facilitant la revisió i la identificació de qualsevol error o inconsistència.

4.1.7.3 Penjament de Dades de Reserva Importades

Després de revisar les dades de les reserves, l'usuari pot procedir a penjar aquestes dades a la base de dades central clicant el botó verd d'un núvol a la part superior esquerra. Aquest procés garanteix que la informació estigui disponible per a tots els components del sistema que la necessitin.

Per penjar les dades, el sistema primer valida les reserves guardades temporalment per assegurar-se que són correctes i completes. Després, el sistema intenta penjar les dades modificades, actualitzant la informació existent amb les noves dades de les reserves.

Un cop el procés de penjament és complet, el sistema mostra una confirmació a l'usuari, assegurant que les dades han estat penjades correctament i que ara formen part de la base de dades.

5 Avaluació

5.1 Testing

Per a les proves, hem decidit utilitzar test E2E¹⁹ amb Playwright. Playwright ens permet testear el comportament de l'aplicació en diferents casos, testejant tant el frontend com el backend en un sol test. Això és possible perquè el comportament del frontend depèn de les respostes del backend. Tot i que aquest enfocament té inconvenients, com ara no poder testear components de manera aïllada i la dificultat de determinar si l'error prové del backend o del frontend, vam optar per aquesta metodologia per tal d'assegurar un desenvolupament àgil, donada la naturalesa de startup del projecte.

A més de les proves E2E, algunes pàgines, com la de localitzacions, s'han testejat manualment per agilitzar el desenvolupament.

5.1.1 Mocks

Pel que fa a les dades, hem utilitzat una resposta d'una reserva antiga de l'API de reserves per fer les proves. Això ens permet provar amb dades fictícies de reserves que ja no estan actives.

5.2 Implementació dels tests

El següent codi és un exemple d'un test E2E utilitzant Playwright. A continuació, s'expliquen les parts més importants:

5.2.1 Mock de l'API

Aquesta part del codi redirigeix les peticions a l'API a una resposta fictícia per provar el comportament del sistema amb dades controlades.

```
test('mobile/servicios-in-curso | should change progress status to En
Origen', async ({ page }) => {
  const { transferService } = transferServices[0];

  await page.route(API_URL, async (route) => {
    await route.fulfill({
      body: JSON.stringify(transferServices)
    });
  });

  await page.goto(URL_MOBILE_IN_PROGREESS_SERVICE!);
  await page.waitForResponse(API_URL);
```

5.2.2 Esperar a que Carregui

Aquí es verifica que l'indicador de càrrega desaparegui abans de continuar amb altres comprovacions, assegurant-se que la pàgina ha carregat completament.

```
const loader = await page.locator('.loader');
```

¹⁹ End-to-End

```
await expect(loader).toBeHidden();
```

5.2.3 Verificació del Text

Aquesta línia comprova que el botó de tipus de servei tingui el text correcte, assegurant-se que la informació es mostra com s'espera.

```
const btnServiceType = await page.locator('#btn-service-type');
await expect(btnServiceType).toHaveText('ENTRADA');
```

5.2.4 Verificació de Dades de l'Origen

Es verifica que el nom de l'origen mostrat a la pàgina coincideixi amb les dades del servei de transferència utilitzat en les proves.

```
const originName = await page.locator("#origin-name");
await expect(originName).toContainText(transferService?.origin?.name
?? 'invalid');
```

5.2.5 Canvi d'estat

Aquesta secció del codi simula el canvi d'estat d'un servei, des d'"En Camino" a "En Origen", comprovant que el sistema respon correctament a aquesta interacció de l'usuari.

```
await btnProgressStatus.click();

const dropdownProgressStatus = await page.locator("#dropdown-
progress-status");

await expect(dropdownProgressStatus).toBeVisible();

const itemEnOrigen = await page.locator('[data-item-
status="EN_ORIGEN"]');
await itemEnOrigen.click();

const spinner = await btnProgressStatus.locator('.q-spinner');
await expect(spinner).toBeVisible();

await expect(spinner).toBeHidden();
await expect(btnProgressStatusLabel).toContainText("En Origen");
});
```

5.3 Compliment dels Requisits

5.3.1 Requisits funcionals

- RF1: Mostrar una taula amb el llistat de totes les reserves.
 - **Compliment:** S'ha implementat una taula que mostra totes les reserves amb columnes com el número de reserva, data de reserva, origen, destinació, estat de la reserva, entre altres detalls pertinents.
 - **Proves Realitzades:** S'ha verificat que la taula carrega correctament totes les reserves des de la base de dades i que les dades es mostren de manera coherent utilitzant tests automatitzats amb Playwright.
- RF2: Proveir una funcionalitat per crear noves reserves manualment.

- **Compliment:** S'ha creat un formulari per a la creació de noves reserves amb tots els camps necessaris com data, hora, origen, destinació, nombre de passatgers, tipus de vehicle, entre altres.
- **Proves Realitzades:** S'han dut a terme proves d'usuari per assegurar que el formulari permet crear reserves noves i que aquestes s'emmagatzemen correctament a la base de dades utilitzant tests automatitzats amb Playwright.
- RF3: Proporcionar la capacitat d'exportar les dades mostrades a la taula a un fitxer CSV.
 - **Compliment:** S'ha afegit una funcionalitat que permet exportar les dades mostrades a la taula a un fitxer CSV, incloent-hi els filtres aplicats.
 - **Proves Realitzades:** S'ha provat l'exportació de dades en diferents escenaris per assegurar que el fitxer CSV generat conté les dades correctes utilitzant tests automatitzats amb Playwright.
- RF4: Permetre als usuaris filtrar la taula de reserves per diversos criteris.
 - **Compliment:** S'ha implementat una funcionalitat de filtratge per permetre als usuaris cercar per número de reserva, data, origen, destinació, estat, entre altres criteris.
 - **Proves Realitzades:** S'han fet proves de filtratge per assegurar que els resultats retornats són correctes segons els criteris especificats utilitzant tests automatitzats amb Playwright.
- RF5: Facilitar una pàgina que permeti editar els detalls d'una reserva específica.
 - **Compliment:** S'ha creat una pàgina d'edició de reserves que permet modificar camps com ara l'hora, la data, el número de passatgers, entre altres.
 - **Proves Realitzades:** S'han dut a terme proves d'edició per assegurar que els canvis es guarden correctament a la base de dades utilitzant tests automatitzats amb Playwright.
- RF6: Proporcionar una pàgina on els usuaris puguin veure un registre de les modificacions fetes a una reserva.
 - **Compliment:** S'ha implementat una pàgina d'històric que mostra les modificacions fetes a les reserves, incloent qui va fer el canvi, quan es va fer i què va ser modificat.
 - **Proves Realitzades:** S'han revisat els registres per assegurar que totes les modificacions es mostren correctament utilitzant tests automatitzats amb Playwright.
- RF7: Visualitzar una llista completa de vehicles amb opcions de filtratge.
 - **Compliment:** S'ha creat una pàgina que mostra una llista completa de vehicles amb opcions de filtrat per característiques com marca, model, any, entre altres.
 - **Proves Realitzades:** S'han fet proves per assegurar que els filtres funcionen correctament i que es mostren els vehicles adequats utilitzant tests automatitzats amb Playwright.
- RF8: Proporcionar una interfície per introduir dades d'un nou vehicle.
 - **Compliment:** S'ha creat un formulari que permet introduir dades de nous vehicles, incloent-hi matrícula, marca, model, any, estat i quilometratge.
 - **Proves Realitzades:** S'han fet proves de creació de vehicles per assegurar que les dades es guarden correctament utilitzant tests automatitzats amb Playwright.

- RF9: Habilitar la funcionalitat per eliminar un vehicle del sistema.
 - **Compliment:** S'ha afegit la funcionalitat per eliminar vehicles del sistema mitjançant un identificador únic, amb confirmació d'eliminació.
 - **Proves Realitzades:** S'han dut a terme proves d'eliminació per assegurar que els vehicles es poden eliminar correctament i que es mostra una confirmació utilitzant tests automatitzats amb Playwright.
- RF10: Permetre l'edició de dades d'un vehicle existent.
 - **Compliment:** S'ha creat una funcionalitat que permet editar les dades dels vehicles existents amb validació d'entrada per assegurar la correcció dels canvis.
 - **Proves Realitzades:** S'han realitzat proves d'edició per assegurar que els canvis es guarden i es validen correctament utilitzant tests automatitzats amb Playwright.
- RF11: Mostrar un mapa interactiu amb polígons que representen localitzacions.
 - **Compliment:** S'ha implementat un mapa interactiu que carrega totes les localitzacions existents amb polígons representatius.
 - **Proves Realitzades:** S'han fet proves per assegurar que el mapa es carrega correctament i que les localitzacions es mostren adequadament testejant-ho manualment.
- RF12: Permetre als usuaris dibuixar polígons en el mapa per definir noves localitzacions.
 - **Compliment:** S'ha afegit una funcionalitat que permet dibuixar nous polígons en el mapa i guardar les dades de la nova localització.
 - **Proves Realitzades:** S'han dut a terme proves de creació de localitzacions per assegurar que els polígons es dibuixen i es guarden correctament testejant-ho manualment.
- RF13: Visualitzar una taula amb totes les localitzacions definides.
 - **Compliment:** S'ha creat una taula que mostra totes les localitzacions definides, permetent als usuaris fer zoom a localitzacions específiques al mapa.
 - **Proves Realitzades:** S'han fet proves per assegurar que la taula es carrega correctament i que les interaccions amb el mapa funcionen testejant-ho manualment.
- RF14: Ofereix la capacitat d'editar detalls o la forma dels polígons de localitzacions existents.
 - **Compliment:** S'ha implementat la funcionalitat per editar els detalls o la forma dels polígons de les localitzacions existents.
 - **Proves Realitzades:** S'han fet proves d'edició per assegurar que els canvis es reflecteixen correctament al mapa i a la base de dades testejant-ho manualment.
- RF15: Habilitar l'eliminació de polígons i localitzacions associades.
 - **Compliment:** S'ha afegit la funcionalitat per eliminar polígons i localitzacions associades del sistema.
 - **Proves Realitzades:** S'han fet proves d'eliminació per assegurar que els polígons i les localitzacions es poden eliminar correctament testejant-ho manualment.
- RF16: Permetre als usuaris pujar fitxers XML o JSON amb informació de reserves.

- **Compliment:** S'ha creat una funcionalitat que permet pujar fitxers XML, JSON o CSV amb informació de reserves, amb confirmació de càrrega reeixida.
- **Proves Realitzades:** S'han dut a terme proves de càrrega de fitxers per assegurar que es validen correctament i que es gestionen errors de format o càrrega utilitzant tests automatitzats amb Playwright.
- RF17: Mostrar les dades de reserves processades des dels fitxers carregats.
 - **Compliment:** S'ha implementat una funcionalitat que mostra les dades de les reserves processades des dels fitxers carregats en una taula detallada.
 - **Proves Realitzades:** S'han fet proves per assegurar que les dades es mostren correctament i que es poden visualitzar tots els detalls rellevants utilitzant tests automatitzats amb Playwright.
- RF18: Facilitar el penjament de les dades de reserva importades a la base de dades central.
 - **Compliment:** S'ha creat una funcionalitat que permet penjar les dades de reserva importades a la base de dades central amb confirmació d'actualització.
 - **Proves Realitzades:** S'han fet proves de penjament per assegurar que les dades es penjen correctament i que es gestionen errors en la connexió o el procés de penjament utilitzant tests automatitzats amb Playwright.

5.3.2 Requisits no Funcionals

- Interfície d'usuari intuïtiva i fàcil d'utilitzar (RNF1):
 - **Metodologia:** S'han fet proves amb 5 treballadors de l'empresa, representant el perfil d'usuari final. Cada treballador ha completat un conjunt de tasques predefinides sense rebre formació prèvia.
 - **Resultats:** De mitjana, els usuaris han puntuat la interfície amb un 8.5 sobre 10 en termes de facilitat d'ús.
- Temps de resposta acceptable (RNF2):
 - **Metodologia:** S'han dut a terme proves manuals cronometrant el temps de càrrega de les diferents pàgines i accions específiques.
 - **Resultats:** En la gran majoria d'interaccions, el temps de càrrega ha estat d'1 segon com a màxim, excepte en sol·licituds on es requereixen moltes files d'alguna taula.
- Protecció de dades personals i de negoci (RNF3):
 - **Metodologia:** S'ha implementat HTTPS per a totes les comunicacions entre el client i el servidor.
 - **Resultats:** Les proves han demostrat que HTTPS funciona correctament, garantint la seguretat del trànsit de dades.
- Alta disponibilitat (RNF4):
 - **Metodologia:** S'ha utilitzat AWS CloudWatch per monitorar i detectar el temps de funcionament de l'aplicació durant un període de tres mesos.
 - **Resultats:** L'aplicació ha mantingut una disponibilitat del 99,95% durant el període de prova, superant l'objectiu establert.
- Compatibilitat amb navegadors comuns (RNF5):
 - **Metodologia:** S'han realitzat proves de compatibilitat amb diferents versions de Safari, Chrome i Edge.
 - **Resultats:** No s'han detectat problemes de compatibilitat amb cap de les versions provades.

- Codi ben documentat i fàcil de mantenir (RNF6):
 - **Metodologia:** S'han afegit comentaris a la majoria de funcions i s'han utilitzat noms de variables auto explicatius.
 - **Resultats:** El codi és fàcil de seguir i mantenir gràcies a la documentació i la nomenclatura clara.

6 Conclusions

Aquest Treball de Fi de Grau ha estat una experiència molt enriquidora tant en l'àmbit tècnic com personal. Aquest projecte m'ha permès aprendre i consolidar una sèrie de coneixements i competències essencials en l'àmbit de l'enginyeria informàtica i ha tingut un impacte gran en la meva formació.

Durant el desenvolupament de ConectaDrive, he tingut l'oportunitat d'aprendre i aprofundir en tecnologies avançades com Vue.js, Spring Boot, AWS, i Docker, que són eines fonamentals en el desenvolupament modern d'aplicacions web, i la seva integració en aquest projecte m'ha permès entendre millor el seu funcionament i aplicació pràctica. He adquirit coneixements sòlids sobre com construir aplicacions escalables i segures, utilitzant arquitectures basades en microserveis, i desplegar-les en entorns de núvol.

El TFG m'ha proporcionat una experiència valuosa en la gestió de projectes, ja que he après a planificar i organitzar el treball, establir objectius clars i dates límit, i coordinar tasques amb altres membres de l'equip. La gestió del temps i dels recursos ha estat un aspecte important, així com la identificació i mitigació de riscos potencials.

Un altre aspecte a destacar ha sigut la integració de l'aplicació amb sistemes de tercers. He après a treballar amb API externes per importar reserves de plataformes com Booking i Hoppa, i a gestionar aquestes dades de manera eficient. Això m'ha donat una visió completa de com construir solucions integrades que responen a necessitats reals del negoci.

Enfrontar-me a problemes reals i trobar solucions efectives ha estat una part crucial del TFG. He desenvolupat un pensament analític i resolutiu, capaç d'abordar problemes complexos i trobar solucions innovadores. He après a analitzar els requisits del projecte, a dissenyar solucions tècniques adequades i a implementar-les de manera eficient.

En resum, el desenvolupament de ConectaDrive ha estat una experiència molt enriquidora que ha contribuït positivament a la meva formació personal i professional. He adquirit coneixements tècnics avançats, desenvolupades competències en gestió de projectes, millorat en la integració de sistemes i reforçat el meu pensament analític i resolutiu. En definitiva, aquest TFG m'ha preparat per afrontar futurs reptes professionals amb una base sòlida i amb més confiança.

7 Referències

- [1] Vue.js. Disponible a: <https://vuejs.org/>. [Consulta: 8 Abril 2024].
- [2] Node.js. Disponible a: <https://nodejs.org/>. [Consulta: 10 Maig 2024].
- [3] Docker. Disponible a: <https://docs.docker.com/>. [Consulta: 13 Maig 2024].
- [4] NGINX. Disponible a: <https://nginx.org/>. [Consulta: 5 Maig 2024].
- [5] AWS Lambda. Disponible a: <https://aws.amazon.com/lambda/>. [Consulta: 21 Abril 2024].
- [6] AWS Auto Scaling. Disponible a: <https://aws.amazon.com/autoscaling/>. [Consulta: 8 Abril 2024].
- [7] Amazon VPC. Disponible a: <https://aws.amazon.com/vpc/>. [Consulta: 10 Maig 2024].
- [8] Redis. Disponible a: <https://redis.io/>. [Consulta: 15 Maig 2024].
- [9] Spring Security. Disponible a: <https://spring.io/projects/spring-security/>. [Consulta: 4 Juny 2024].
- [10] Amazon EC2. Disponible a: <https://aws.amazon.com/ec2/>. [Consulta: 12 Maig 2024].
- [11] Amazon S3. Disponible a: <https://aws.amazon.com/s3/>. [Consulta: 10 Maig 2024].
- [12] Amazon RDS. Disponible a: <https://aws.amazon.com/rds/>. [Consulta: 5 Maig 2024].
- [13] Visual Studio Code. Disponible a: <https://code.visualstudio.com/>.
- [14] IntelliJ IDEA. Disponible a: <https://www.jetbrains.com/idea/>.

8 Annex

L'entorn de testing desenvolupat es pot trobar a la URL indicada a continuació. En cas que no sigui funcional o necessiteu accés, podeu contactar a través del meu correu electrònic o del meu perfil de LinkedIn:

- URL d'accés a la web: <https://conectadrive.software/>
- Correu: roger.massana@estudiants.urv.cat
- LinkedIn: [Roger Massana](#)
- Github: <https://github.com/massanaRoger>