

Estudi i implementació del procés digital d'una reserva i compra en el sector automobilístic.

Núria Andión Sesé

Resum– Avui en dia les compres en línia són un aspecte imprescindible tant per les empreses com per particulars. Gràcies a l'evolució constant de les TIC, aquest sistema de venda ha crescut enormement, arribant inclús al sector de l'automòbil. L'objectiu d'aquest projecte, creat per a Volkswagen-Audi España, S.A. (VAESA), és proporcionar al client final la possibilitat de signar una oferta i fer el corresponent pagament i senyal des de qualsevol plataforma, per a la reserva en ferm d'un vehicle. Prèviament, a través d'una aplicació parcialment realitzada en aquest Treball de Fi de Grau, el client haurà pogut concertar una cita per provar el vehicle en el qual està interessat.

Paraules clau– E-commerce, automòbil, concessionari, aplicació, multiplataforma, reserva, Angular2, JavaEE, plataforma de pagament, signatura digital, enginyeria del software.

Abstract– Nowadays, online shopping is an essential aspect for both companies and individuals. Due to the constant evolution of ICT, this system of selling has grown enormously, reaching even the automobile sector. The aim of this project, created for Volkswagen-Audi España, S.A. (VAESA), is to provide to the end customer the possibility of signing an offer and making the corresponding deposit payment through any platform, to confirm the booking of a vehicle. Previously, through an application partially made in this project, the customer will be able to make an appointment to test the vehicle in which is interested.

Keywords– E-commerce, automobile, dealer, cross-platform, application, booking, Angular2, JavaEE, payment platform, digital signature, software engineering.



1 INTRODUCCIÓ

AKTIOS [1] és una consultoria que ofereix als seus clients solucions, tant a l'àmbit de l'estratègia i els negocis com l'externalització de processos de negoci, la gestió de les infraestructures informàtiques i el desenvolupament i manteniment d'aplicacions. L'objectiu principal que té l'empresa és acompanyar al client en tot el procés. Un dels serveis existents és l'e-commerce, que ofereix als seus clients disposar d'una tenda en línia en un temps rècord i mantenir-la amb el mínim esforç propi possible, encarregant-se l'empresa de tot el procés. Dins dels projectes que corresponen a aquest servei, és on es troba el projecte en el que es basa aquest TFG.

- E-mail de contacte: nuria.andion@e-campus.uab.cat
- Menció realitzada: Enginyeria del Software
- Treball tutoritzat per: Lluís Gesa Boté, UAB (DCC)
- Curs 2016/17

Cal esmentar que aquesta empresa és on l'alumne ha estat realitzant les pràctiques en paral·lel. El projecte gestionat per Aktios, anomenat VAESA, és d'on es van separar certs elements que formen part de l'enginyeria del software perquè fossin realitzats en aquest TFG.

1.1 Motivació

Volkswagen-Audi España, S.A. (VAESA) [2] és l'empresa distribuïdora dels vehicles, peces de recanvi i accessoris de les marques Volkswagen, Audi, Sköda i Volkswagen Vehicles Comercials a la Península i Balears. L'objectiu que tenen és oferir les marques i productes més atractius i sostenibles, amb el millor nivell de satisfacció del client, per a ser la millor companyia en vendes i servei del sector d'automoció a Espanya.

En els últims anys l'experiència de la compra en línia cada vegada està més estesa en tots els àmbits. És per això que el grup VAESA vol apropar el seu mercat automobilístic a les noves tecnologies de comerç electrònic, el que permetrà ampliar les funcionalitats i capacitats de negoci amb

els seus clients.

A Aktios, dins del projecte d'Experiència Digital en Compra i amb l'objectiu de millorar l'experiència de compra del client, es precisa una nova solució construïda basant-se en els sistemes de gestió comercial que s'utilitzen actualment per les concessions (IMAWEB) i en coordinació amb el proveïdor de la solució de Mobilitat d'Assessors Comercials. El TFG proposat, per tant, s'engloba com una part dins d'aquesta solució.

2 PROJECTE VAESA

El projecte complet gestionat per Aktios constava en trobar una solució que inclogués l'estudi, disseny, implementació i testing de:

- Una aplicació en la qual s'habiliti un sistema de signatura digital i una passarel·la de pagament que permetin al client fer un pagament i senyal en línia del seu vehicle, després de fer-se-li una oferta comercial (**Digital Ordering**).
- Una segona aplicació amb la qual es possibiliti al client concretar una cita per una prova dinàmica, reservant el vehicle de la seva elecció, en la data i hora que més li convinguin (**Test Drive**)

Totes dues solucions s'havien d'implementar inicialment amb la marca Audi com a base. A partir d'aquesta s'aniria adaptant la solució a les altres marques.

3 EN QUE CONSTA AQUEST TFG?

Quan em vaig integrar en l'equip d'aquest projecte, el mòdul de *Test Drive* es trobava en una fase avançada i el mòdul de *Digital Ordering* encara estava per començar. Partint d'aquesta base, es van separar diversos aspectes d'aquest projecte perquè passessin a formar part d'aquest TFG. Després de revisar l'estat en què es trobava el projecte i de valorar el temps de què es disposava, es va acordar que es realitzaria:

- L'estudi, disseny, implementació i testing del mòdul de **Digital Ordering**.
- Afegir funcionalitat que permeti concertar una cita per la prova d'un vehicle especial o un R8 i realització de testing a **Test Drive**.
- Adaptació tant de **Test Drive**, com de **Digital Ordering**, a les marques de Skoda, Volkswagen i Volkswagen Vehicles Comercials.

4 OBJECTIUS DEL TFG

Per tal d'aconseguir complir els objectius que es plantejaven per aquest treball, després d'algunes iteracions, es van detallar en un document inicial als quals se'ls va donar el formalisme d'assignar a cadascun una nomenclatura, de manera que durant tot el procés sigues més fàcil fer-hi referència, podem veure el detall dels objectius a la Fig.11 de l'apèndix. A més a més, va aportar qualitat al treball que ja s'estava realitzant dins de l'empresa, ja que en certs punts hi havia carència de formalismes.

L'objectiu principal d'aquest TFG és aplicar l'**Enginyeria del Software**, ja que és la menció que vaig realitzar. Dins d'aquest trobem una subdivisió de dues fases: L'estudi i disseny, i la implementació i testing.

Per a la fase d'**estudi i disseny** primerament s'havien de realitzar les preses de requisits de tots dos mòduls a partir de reunions amb el client o de documents on s'especificaven les funcionalitats a realitzar. A més de realitzar el document pertinent de requisits, es va marcar com a objectiu realitzar documentació tant funcional com tècnica d'ambdues parts on es realitzarien diagrames de casos d'ús, diagrames seqüencials, d'activitats, de components, de classes i d'arquitectura. Per últim, en aquesta fase, s'haurà de realitzar la documentació oficial indicada per la universitat i entregar-ho a les dates indicades.

En relació a la fase d'**implementació i testing**, en primer lloc s'havia de finalitzar el mòdul de *Test Drive* de la marca Audi, ja que és la fase que es trobava en un procés avançat i la part restant incloïa finalitzar la implementació tant del front-end com del back-end i realitzar el testing corresponent. En segon lloc s'havia de començar a desenvolupar la part de *Digital Ordering*, en la qual també es començarà amb la marca base, Audi. A grans trets, s'havia de realitzar la part de front-end, la de back-end, integrar-ho amb la plataforma de signatura digital, implementar la passarel·la de pagament i realitzar el testing.

Per últim, tant per *Test Drive* com per *Digital Ordering*, un cop implementada la marca base, s'hauria d'adaptar l'aplicació a les altres marques; Volkswagen, Skoda i Volkswagen Vehicles Comercials.

5 ESTAT DE L'ART

Com ja s'ha mencionat anteriorment, aquest TFG partia d'un projecte ja iniciat, on el mòdul de *Test Drive* d'Audi es trobava en una fase avançada de desenvolupament la qual va ser necessari continuar, realitzar les funcions restants fins a acabar el projecte. *Digital Ordering* per altra banda, era un mòdul a començar des de zero i, per tant, em va ser possible visualitzar el flux complet d'un projecte des del principi.

Des del primer moment que vaig entrar al projecte, em va ser necessari realitzar un curs d'Angular2 [3], ja que no tenia coneixements sobre aquest framework. Un cop finalitzat i tenint una bona base, em van incorporar al projecte de *Test Drive* començant pel desenvolupament de petites funcionalitats perquè m'anés adaptant al projecte i tingués una visió global de com s'havia implementat la solució. A part d'això, se'm va assignar la tasca d'aplicar l'enginyeria del software, ja que pel que fa a la documentació, mancaven alguns formalismes i informació que és necessària tenir ben documentada per a un bon seguiment i realització del projecte.

A partir d'aquí va néixer el projecte del qual consta aquest TFG.

6 METODOLOGIA I TECNOLOGIES APLICADES

La metodologia que s'emprarà en aquest projecte serà SCRUM [4], ja que és la mateixa que s'ha utilitzat fins ara

a l'empresa.

6.1 Que és SCRUM?

SCRUM és una metodologia àgil aplicada per al desenvolupament de software en el qual s'apliquen un conjunt de bones pràctiques per treballar en equip, on cada persona participant assumeix un rol i així obtenir el millor resultat possible d'un projecte.

Aquest mètode permet realitzar entregues parcials i regulars del producte final i està especialment indicat per projectes en entorns complexos, on s'han d'obtenir resultats ràpidament i els requisits poden anar canviant durant el procés o des d'un principi estar poc definits, d'aquesta manera l'equip es pot anar adaptant a les necessitats del client i estar en constant contacte amb ell.

SCRUM funciona a través de la definició tasques i la prioritització d'aquestes segons el temps estimat i la dificultat que poden comportar. El temps es mesura en sprints (blocs de temps curts i fixes) que es revisen en reunions diàries i a principi i final de cada sprint.

6.2 Aplicació al TFG

Aquesta metodologia ha donat bons resultats en aquest TFG, ja que l'empresa ja l'aplicava anteriorment i per tant no ha sorgit cap problema. SCRUM està dividit en una sèrie de rols on a cada membre li corresponen certes responsabilitats [5], com per exemple el *Product Owner* que es qui decideix les tasques a fer i les prioritza, aquest rol per aquest projecte el durà a terme el meu tutor dins de l'empresa, que també realitza la funció de *Scrum Master*.

Al ser una metodologia que requereix reunions diàries, entregues parcials i una bona detecció i definició de tasques, entre altres característiques, s'ha valorat que s'adequa molt bé ja que permet un seguiment i una comunicació constant entre tot l'equip. A més, com es tracta d'un projecte en el qual els requisits poden anar canviant a mesura que s'avança, SCRUM s'adapta de manera eficient, ja que els períodes de temps generalment són blocs de dues setmanes (Sprints), i al final de cada bloc es pot veure el que s'ha realitzat i els clients poden opinar i canviar els aspectes que desitgin.

6.2.1 Reunions de seguiment

Aquesta metodologia implica una sèrie de reunions específiques al llarg del projecte. A l'inici de cada sprint es realitza una reunió per definir i planificar la feina a realitzar en aquest període. Les reunions diàries no s'han fet rigorosament, ja que no s'ha vist necessari, però sí que s'han realitzat amb una certa freqüència per poder anar veient com va avançant el projecte i les dificultats que van apareixent. Al final de cada sprint es duia a terme una altra reunió, per veure la feina que s'ha realitzat, els problemes que han sorgit i si les estimacions de temps han estat encertades.

A més de les reunions pertinents a l'empresa on es realitza el projecte, també hi ha una sèrie de reunions de seguiment amb el tutor del TFG. Tot i que no estan definides, segueixen un ritme similar per a cada entrega. Una reunió prèvia a una entrega i una posterior per comentar la feina a fer i la feina ja realitzada, tenint sempre a l'abast el contacte a través de correu electrònic.

6.2.2 Eines utilitzades

Per a una correcta aplicació d'aquesta metodologia, respecte a definició, organització i assignació de tasques, s'han utilitzat les eines Confluence i Jira, que es detallen en següent apartat de planificació.

7 PLANIFICACIÓ

7.1 Eines de planificació i documentació

Per portar una bona organització i seguiment d'equip i tenir tot ben centralitzat i documentat s'utilitzaran les següents eines.

7.1.1 Confluence

Confluence [6] és un software enfocat a la col·laboració en equip, utilitzat en entorns corporatius.

L'objectiu de Confluence és que es dediqui menys temps a detectar errors i més en complir els teus objectius i projectes. Ajuda a organitzar la feina, a crear documents gràcies a plantilles disponibles i et permet tenir-ho tot centralitzat, a l'abast de tots els membres del teu equip, perquè pugui, tant visualitzar-lo en qualsevol moment, com contribuir amb canvis i/o millores.

7.1.2 Jira

JIRA [7] ajuda a gestionar els projectes, a administrar les tasques relacionades i proporciona la capacitat de fer un seguiment d'errors i d'incidents.

Ofereix eines per a la gestió de requisits i la planificació dels sprints, la creació de tasques i la distribució d'aquestes als diversos membres de l'equip. També permet la creació d'incidències i poder supervisar-les, prioritzar-les i analitzar la feina realitzada per l'equip amb total visibilitat.

Es pot integrar JIRA fàcilment amb Confluence, fet que permetrà oferir a tothom una transparència i relació entre les incidències supervisades al JIRA i el contingut relacionat amb aquestes al Confluence. També podem transformar els requisits de projectes existents al Confluence en tasques pendents al JIRA.

7.2 Organització

Per planificar les tasques en el temps que disposem, s'ha realitzat un diagrama de Gantt on s'ha assignat a cada objectiu i subobjectiu un temps determinat que s'ha considerat adient per completar cada tasca. A partir del diagrama inicial que es va realitzar, s'ha anat actualitzant amb els canvis corresponents i replanificant les tasques necessàries si no es finalitzaven a temps. Gràcies a això s'ha pogut tenir sempre una visió global de la feina a realitzar, el temps que implica cada tasca i les dates en què s'haurien de finalitzar.

Es pot visualitzar un exemple de l'últim diagrama de Gantt que es va crear a la Fig. 12 de l'apèndix.

8 REQUISITS

Tant pel mòdul de *Test Drive* com pel de *Digital Ordering*, s'han realitzat uns documents amb la captura dels requisits

extrets d'una sèrie de reunions realitzades amb l'equip i els clients.

Tot i que el mòdul de *Test Drive* ja estava iniciat, és una bona pràctica per l'Enginyeria del Software tenir tots els conceptes ben documentats, així que també s'ha realitzat el document d'aquesta part.

Una bona captura dels requisits permet obtenir una millor visió del conjunt i compondre una especificació completa, correcta i consistent. És clau detectar-los bé, ja que en aquesta etapa és quan el cost de modificació i correcció és més baix.

Aquest document consta d'informació sobre el problema, especificació de propietats i comportament del sistema i restriccions que puguin existir.

Per realitzar-lo s'ha seguit la següent formalitat (Fig. 1) on s'assignava un codi, un nom, el tipus de validació que tindrà aquest requisit (A través del disseny, amb una revisió o realitzant un test), una breu explicació d'aquest i si té algun requisit pare.

CODI_REQUISIT	NOM_REQUISIT	TIPUS DE VERIFICACIÓ (Disseny, Revisió, Test)
Breu explicació del requisit		
		Requisit pare

Fig. 1: Estructura requisit

9 DISSENY

9.1 Documentació funcional i tècnica

La documentació funcional consta d'una declaració d'intencions, és un pas previ a la implementació que expressa amb un gran nivell de detall tècnic les funcionalitats que s'han d'implementar posteriorment. La documentació tècnica per altra banda consisteix en tota la informació que ens explica com funciona un sistema, com està dissenyat i amb quin objectiu. Per fer-ho s'ha seguit l'estàndard Llenguatge Unificat de Modelat (UML) [8].

Per aquest projecte s'han realitzat tant diagrames estructurals com de comportament i d'interacció per ambdós mòduls. De la part estructural, que mostra l'estructura estàtica dels objectes en un sistema, s'han creat diagrames de classes i de components. Per altra banda, com a diagrames de comportament, els quals mostren el comportament dinàmic dels objectes en el sistema, tenim diagrames d'activitats i de casos d'ús. Per últim, com a diagrames d'interacció, s'han realitzat diagrames de seqüència i diagrames de flux.

Per realitzar els diagrames dels documents de disseny s'ha utilitzat l'eina Gliffy [9], que és un software utilitzat per la creació de diagrames de tot tipus que funciona arrossegant i deixant anar objectes, a través d'una aplicació HTML5 cloud-based.

Gràcies al fet que està integrat amb JIRA i Confluence, permet afegir directament qualsevol tipus de diagrama a les pàgines per poder documentar més visualment els conceptes, fent que siguin més fàcil d'entendre i més ràpid de compartir i col·laborar amb l'equip, ja que tothom pot veure i editar els diagrames en temps real.

10 IMPLEMENTACIÓ

Com s'ha mencionat anteriorment, en aquest TFG es troben dos mòduls, *Test Drive* i *Digital Ordering*. En aquesta secció es parlarà sobre la implementació realitzada al front-end tant pel mòdul de *Digital Ordering* com al de *Test Drive*, no obstant, es detallaran conceptes comuns que s'han aplicat en ambdós mòduls.

10.1 Solució tecnològica

La solució que es va proposar va ser una aplicació cloud híbrida. Es tractaria d'un frontal web executant-se al cloud i el backoffice (IMAWEB) executant-se als servidors centrals de VAESA. Els valors que tindria aquesta solució són la seva escalabilitat, agilitat i la seguretat de la informació (dades del client a VAESA).

La infraestructura consisteix en aplicacions executant-se en contenidors dintre d'instàncies cloud (Amazon, Blue-mix, Openshift).

Es tracta d'una aplicació Java EE 7 [10] executant-se sobre WildFly 10 [11]. Les aplicacions es troben emmagatzemades a contenidors Docker [12].



Fig. 2: Solució tecnològica

10.2 Integració Continua

La integració continua és una pràctica que implica que els desenvolupadors integrin el codi en un repositori compartit diverses vegades al dia.

Cada entrada es verificada per un build automatitzat, permetent als equips detectar errors d'integració tan aviat com es pugui i localitzar-los més fàcilment.

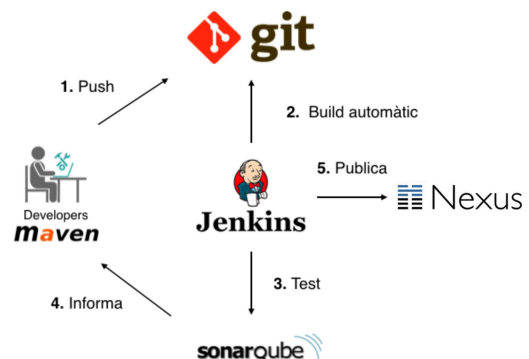


Fig. 3: Integració continua

L'entorn està format per (Observar procés a la Fig. 3):

- Maven [13]: Eina de software per la gestió i construcció de projectes Java.
- GIT [14]: Sistema de control de versions on estarà allotjada l'aplicació

- Jenkins [15]: Servidor d'integració contínua, s'utilitza per a l'automatització del procés.
- SonarQube [16]: Eina per l'anàlisi del codi
- Nexus [17]: Repositori d'artefactes Maven

10.3 Entorn de desenvolupament i tecnologies

Per a aquest projecte utilitzem un sistema de control de versions GIT que permet, d'una forma còmoda, el treball en equip. El repositori estarà allotjat en un servidor Bitbucket [18].

Per mantenir un control al flux de treball de desenvolupament, JIRA i Confluence estan integrats amb Bitbucket, això comportarà que cada vegada que fem un commit s'actualitzin les incidències al JIRA automàticament.

Pel back-end s'utilitzarà Red Hat Jboss Developer Studio [19] com a entorn de desenvolupament integrat, que està desenvolupat per Oracle Corporation [20] i simplifica el desenvolupament d'aplicacions basades en Java que aborden cada pas del cicle de vida de l'aplicació.

Com a editor de codi pel front-end, s'utilitzarà Visual Studio Code [21], que és gratuït i de codi obert. Inclou suport per a la depuració, control per GIT i completat de codi intel·ligent.

Les tecnologies que s'utilitzaran pel desenvolupament d'aquest TFG seran Angular2 [3] per la part de front-end i JavaEE [10] per la part de back-end.

Angular2

Angular2 és un framework basat en JavaScript [22] de codi obert mantingut per Google, que s'utilitza per a la creació d'aplicacions web.

Està basat en components (Els quals s'encarreguen de gestionar les vistes), mòduls, templates, el concepte de data binding, directives, serveis i dependency injection.

Per aquest projecte s'utilitzarà TypeScript [23], ja que el propi framework està desenvolupat en aquest llenguatge i tota la documentació i els generadors de codi estan pensats en TypeScript. Tot i així, tot el codi que s'executa en el navegador és al final JavaScript, ja que es realitza una transformació prèvia.

11 FRONT-END DIGITAL ORDERING

En aquesta secció podem visualitzar tots els detalls de la implementació del front-end d'aquest mòdul que, com s'ha mencionat anteriorment, consisteix en una aplicació en la qual s'habilita un sistema de signatura digital i una passarel·la de pagament, que permet al client fer un pagament i senyal en línia del seu vehicle, després d'haver-li fet una oferta comercial.

Com a la majoria d'aplicacions web, els projectes de VAESA es divideixen en dues grans parts, Front-end i Back-end. La part de front-end és la part del software que interactua amb els usuaris, aquesta part consta d'un projecte Angular2 amb les vistes, que són interpretades i mostrades a l'usuari final a través d'un navegador.

11.1 Flux de l'aplicació

Com podem visualitzar a la Fig. 4, el flux de l'aplicació està format per tres grans passes: La comprovació de l'oferta, la signatura des del dispositiu i la realització del pagament i senyal.

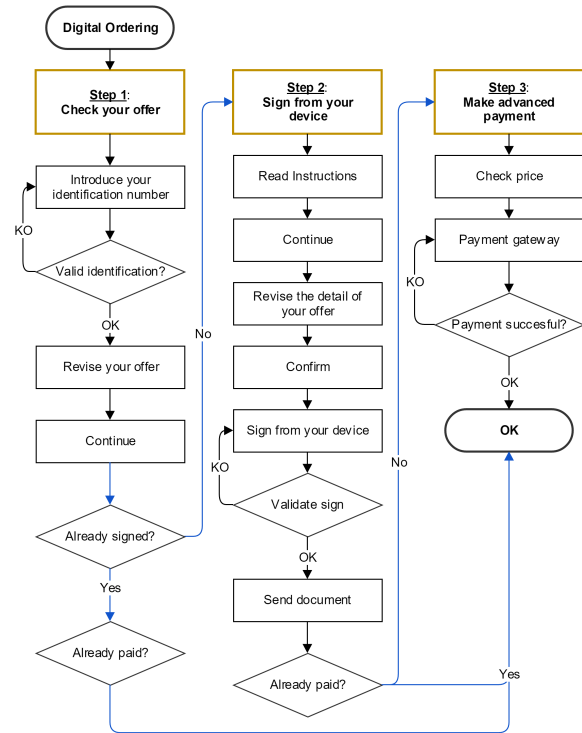


Fig. 4: Mapa de navegació *Digital Ordering*

Abans d'entrar a la nostra aplicació, a l'usuari li arriba un e-mail amb l'oferta que se li ha realitzat pel vehicle que vol comprar i un link on podrà realitzar-la. Una vegada l'usuari accedeix a aquest link haurà d'introduir el seu DNI perquè el sistema comprovi que correspon al DNI associat a l'oferta.

A continuació, en cas d'introduir-lo correctament, l'usuari podrà visualitzar l'oferta que se li ha realitzat i comprovar que totes les dades són correctes. Un cop les confirmi, navegarà al següent pas, la realització de la signatura. L'usuari haurà de llegir les instruccions del procediment a seguir i, a continuació, passarà a mostrar-se-li el document de la compra, que haurà de revisar, i si tot està correcte, signar a través de *Signaturit* [25] als llocs indicats. Una vegada validada la signatura, s'enviarà el document i es procedirà al pagament i senyal.

Per realitzar el pagament i senyal, es comprovarà el preu i es farà el pagament introduint les dades necessàries a través del TPV de *Cecabank* [26]. A continuació es mostrarà el missatge pertinent segons la resposta al pagament.

Finalment es mostrarà un resum de la compra i la confirmació de què els procediments han tingut èxit.

11.2 Estructura

Aquestes accions explicades anteriorment tenen certs components Angular associats. L'estructura està composta per passes, que son gestionades pel component Step a través del servei *SessionService* que serà utilitzat per cada un dels

steps per realitzar la navegació entre ells. La resolució de la firma i el pagament es realitzarà als components: Sign i Pay, la validació serà donada pel component Validation i dependent del resultat portarà a Signature-OK, Payment-KO, Payment-OK. Dependent de l'estat de l'oferta (Vàlida, no trobada o caducada) es dirigeix als components Display, Expired o NotFound. L'oferta es pot trobar ja firmada o pagada anteriorment, aspecte que redirigirà a Already-Paid o Already-Signed.

11.3 Serveis

Els serveis de què disposa aquest front-end Angular són els següents:

Primer de tot podem trobar l'**API Caller**. És el que s'encarrega, a través de l'arxiu 'configuration.json', de configurar l'aplicació amb els paràmetres necessaris. Aquest arxiu està compost pel paràmetre d'entorn, la marca i els URL de *Cecabank* i de l'oferta. També aquest és el que realitzarà de fer les peticions GET a l'API i d'aquesta manera totes les entitats que estan escoltant a l'API-caller rebran la resposta corresponent.

El següent servei que podem trobar és el **Data Manager**. Aquest servei gestiona les dades de l'usuari i de l'aplicació. Escolta a l'API-caller i s'encarrega de gestionar les accions a realitzar segons la resposta que rebí de la verificació del DNI. En cas de resposta afirmativa (Codi 200) recuperarà les dades privades o públiques, en cas d'error, mostrarà el missatge pertinent. Els estats en què es pot trobar una oferta en cas d'error són *no existent* o *caducat*. En aquest servei és on es guarda el formulari que posteriorment s'enviarà a *Cecabank*, s'uneix tot en un string i s'encrypta.

Per últim el servei **Session** s'encarrega de desar les dades de l'estat de l'oferta i l'usuari per poder disposar d'elles en qualsevol moment que siguin necessàries.

Per aquest mòdul el procés de firma està definit que s'implementa amb *Signaturit* [16] i la passarel·la de pagament ha de ser realitzada amb *Cecabank*[17].

11.4 Signatura digital

Un dels requisits del projecte de *Digital Ordering* consistia en afegir un proveïdor de firma digital de documents, en aquest cas *Signaturit* [25]. *Signaturit* crea un entorn de confiança entre les empreses i els clients per millorar la productivitat i seguretat en els processos de contractació, facilita la finalització de qualsevol procés de firma proporcionant una solució per sol·licitar i realitzar firmes electròniques legalment vinculants. Per utilitzar-la, els clients no han d'accedir a plataformes externes ni descarregar-se cap aplicació.

Pel procés de firma es realitzen les següents passes. En quant l'usuari cliqui per realitzar la firma, s'obrirà un iframe de *Signaturit* (Fig. 5) on l'usuari haurà de revisar el document de l'oferta i realitzar la seva signatura. L'aplicació es quedarà esperant i una vegada s'hagi signat i es cliqui a acceptar, s'enviarà un event que tancarà l'iframe i es donarà per vàlida la signatura.

Per realitzar el procés de signatura, IMAWEB s'integrarà amb *Signaturit* per enviar el document de la comanda i obtenir el token i l'URL que *Digital Ordering* necessita per

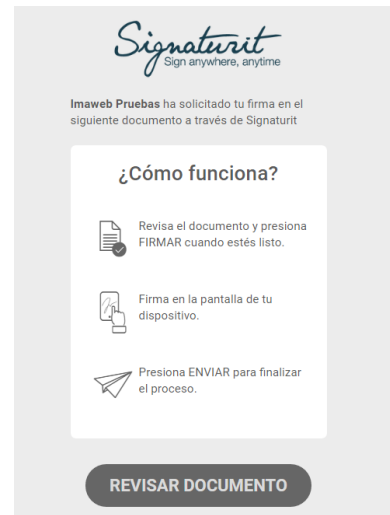


Fig. 5: iframe Signaturit

mostrar-lo i que el client pugui signar el document. Quan el client signi la comanda, serà IMAWEB qui obtingui el contracte i les metadades de *Signaturit*.

11.5 Passarel·la de pagament

Per a aquest procés, es va definir que el pagament es faria a través del TPV virtual ofert per *Cecabank* [26], que permet realitzar pagaments mitjançant una targeta de crèdit/dèbit a internet. Ofereix integritat de les dades enviades i seguretat en la comunicació entre el client/comerç i el TPV virtual, ja que el diàleg és a través del protocol HTTPS.

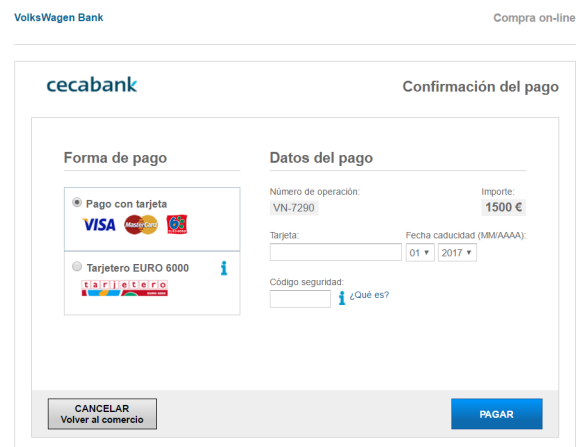


Fig. 6: Cecabank

Una vegada el client procedeix al pagament (Fig. 6), per evitar que un tercer pugui manipular qualsevol dada entre l'enviament de les dades des del comerç fins al TPV virtual, s'ha de generar una signatura a partir d'una sèrie de camps ocults i enviar-los al TPV de *Cecabank* perquè s'encarregui de processar-la. La signatura electrònica és recalculada pel TPV virtual i comparada amb la signatura electrònica rebuda abans de procedir a acceptar qualsevol pagament. En quant es finalitza el pagament, es retorna el control a l'URL corresponent, dependent del resultat.

12 FRONT-END TEST DRIVE

Un cop vist el mòdul de *Digital Ordering*, que és el que s'ha implementat per complet, parlarem sobre el mòdul de *Test Drive*. Com recordarem que s'ha explicat anteriorment, *Test Drive* consta d'una aplicació amb la qual es possibilita al client concretar una cita per una prova dinàmica, reservant el vehicle de la seva elecció, en la data i hora que més li convinguin. D'aquest mòdul la part que s'ha realitzat per aquest TFG és, com veurem a continuació, el flux dels cotxes especials i dels R8.

Aquest mòdul està realitzat d'una manera molt similar a *Digital Ordering*, consta d'un projecte front-end implementat amb Angular2 que conté les vistes, les quals són interpretades i mostrades a l'usuari final a través d'un navegador.

12.1 Flux de l'aplicació

Per a aquest mòdul, es pot veure el flux complet de l'aplicació a la Fig. 10 de l'apèndix, i en relació amb aquest, a la Fig. 7 es pot veure una reducció de la funcionalitat realitzada per aquest TFG. Com podem visualitzar en aquesta última, el flux de l'aplicació per a la concertació d'una cita d'una prova dinàmica d'un vehicle especial o un R8, està format per tres passes; la selecció del vehicle, l'ompliment del formulari i la confirmació de la reserva.

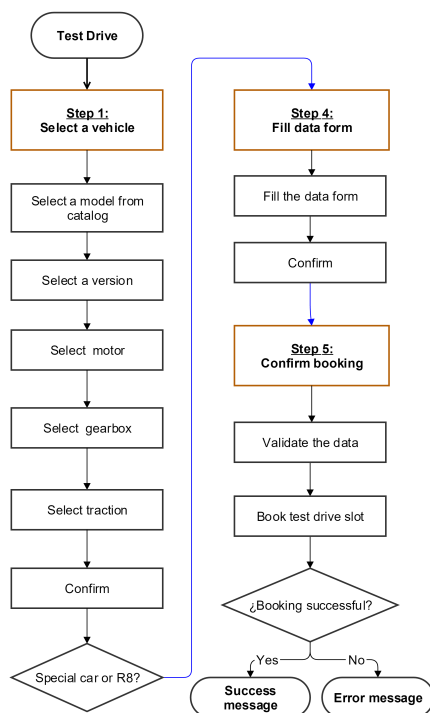


Fig. 7: Mapa de navegació *Test Drive* (Especial)

El procés que es realitza és el següent: L'usuari selecciona un vehicle identificat com especial o un R8. A continuació, realitza la selecció de la versió, el motor, el canvi i la tracció. Seguidament, al següent pas, haurà d'omplir les dades corresponents al formulari i confirmar la reserva. Un cop es validin les dades introduïdes i la disponibilitat, es mostrarà el missatge corresponent i es reservarà el vehicle desitjat.

12.2 Estructura i Serveis

Fent referència a l'estructura i els serveis de *Test Drive*, es va seguir la mateixa lògica que al mòdul de *Digital Ordering* a l'hora de crear tant els components com els serveis.

Pel fet que no ha sigut feina d'aquest TFG crear-los (Degut a que ja es trobava implementat quan es va començar el projecte) i el que s'ha realitzat és afegir funcionalitats, es mencionarà de manera breu l'estructura involucrada.

Els components amb els quals s'ha treballat són Step-Component, Car-Selector, Dealer-Selector, UserForm-Data i Confirmation. Els serveis de dels que disposa segueixen la mateixa lògica que a *Digital Ordering*, però realitzant les funcionalitats específiques relacionades amb *Test Drive*.

13 BACK-END

La part de back-end és la que conté tota la lògica de negoci, i la que processa les dades d'entrada del front-end. Aquesta part serà desenvolupada amb Java i JDeveloper. A part d'això, en aquest projecte disposem dels serveis de IMAWEB i DPS.

13.1 Definició

El back-end d'aquesta aplicació haurà de realitzar una sèrie de crides a l'API REST de IMAWEB.

En el cas de *Digital Ordering* per obtenir les dades de l'oferta, el client rep un e-mail amb un enllaç que inclou l'URL al servei amb la ID de l'oferta (un token) amb el qual accedeix al servei. El servidor necessita les dades de l'oferta i fa una petició a IMAWEB per obtenir-les.

Per marcar l'oferta com a pagada, el client paga la reserva a través del servei i servidor fa una petició a IMAWEB indicant que l'oferta ha estat pagada.

En canvi, pel mòdul de *Test Drive* el que s'ha implementat al back-end és l'enviament de les dades a IMAWEB i DPS. A continuació es detallaran els conceptes tant de DPS com de IMAWEB.

13.1.1 DPS Services

DPS Services [27] és una solució existent a VAESA per recollir, centralitzar i unificar els formularis de les campanyes que es publiquen a diversos medis (portals, web, accions de màrqueting, etc.). DPS unifica la manera d'enviar les dades a VAESA i s'encarrega de derivar les dades a les eines corresponents (Com IMAWEB).

La comunicació es realitza mitjançant peticions HTTP, documents XML i codificació UTF-8, per tant aquesta operació és realitzada al back-end.

En aquest projecte s'ha establert que només s'utilitza DPS per informar, no per obtenir dades.

13.1.2 IMAWEB

IMAWEB [28] sorgeix amb l'objectiu d'oferir solucions coherents i àmpliament demandades dintre del sector de l'automòbil per col·lectius de venda i màrqueting de les marques, així com dels seus concessionaris.

L'aplicació CRM [29] de IMAWEB és l'únic CRM desenvolupat específicament pel sector de l'automòbil, que aporta les últimes tendències en aquestes àrees, facilitant

l'aplicació a la pràctica del mètode de seguiment comercial recomanat per la marca.

En aquest projecte s'utilitza com a servei, proporciona totes les dades necessàries, no cal cap tipus d'instal·lació, actualització o manteniment.

13.1.3 Estructura

Per aquest projecte, tots dos mòduls disposen de la mateixa arquitectura i segueixen el mateix flux entre mòduls, es pot visualitzar el diagrama corresponent a la Fig. 13 de l'apèndix (Figura realitzada per l'arquitecte d'Aktios).

Com podem observar, l'arquitectura del back-end està formada per diversos mòduls, aquest consten en els següents: **Business Service** és la interfície utilitzada pel front per obtenir la comanda i injecta el següent mòdul; **Business Delegate** que s'encarrega de delegar. Això es realitza per desacoblar l'end-point REST amb la lògica de l'aplicació. És útil perquè si en un futur si l'API deixés de ser REST i es volgués canviar a SOAP es podria realitzar el canvi fàcilment, però de moment, és només una operació de delegació.

El mòdul **Service Facade** és injectat pel Business Delegate. Aquest és l'encarregat d'aconseguir les dades necessàries de la capa lògica (Injecta Application Service) i crear el missatge de resposta per a les peticions REST.

D'altra banda, **Application Service** és el responsable d'obtenir les dades de IMAWEB i les processa perquè el front de l'aplicació les consumeixi, aquest injecta a l'últim mòdul, **Service Gateway**, que es troba a la interfície per separar la comunicació específica de l'end-point de la lògica de l'aplicació. Aquest mòdul és el que decideix el servei de l'API a cridar amb els paràmetres necessaris i fa les conversions de dades corresponents. Seguint al mateix mòdul trobem els **Utils** d'aquesta classe, que consten d'una sèrie de mappers que converteixen els objectes que retorna IMAWEB a objectes d'aquest mòdul.

14 TESTING

El testeig del software és una part de molta importància en el cicle de vida d'un projecte, ja que ajuda a millorar la qualitat, la fiabilitat i el rendiment del sistema. Consisteix en el procés de trobar errors al software i corregir-los. És necessari perquè, encara que sembli que tot funciona correctament, pot ser que s'hagin comès errors no percebuts, per tant és necessari realitzar comprovacions.

Alguns errors poden no tenir importància o ser petits detalls, però a la llarga, poden aparèixer errors cars, crítics o perillosos, per això s'ha d'assegurar la qualitat del producte que s'entrega als clients a través del testeig de l'aplicació.

Per aquest projecte s'ha decidit implementar testing end-to-end per el front-end i testing unitari per el back-end.

14.1 Testing Unitari

El testing unitari consta en testejar unitats individuals del codi d'un o més components, juntament amb dades associades, per demostrar si es realitza la funcionalitat esperada. Proven una funcionalitat concreta d'un mètode, independentment de la resta.

Per realitzar el testing unitari al Back-End s'ha utilitzat JUnit [30], que és un framework creat per realitzar proves unitaries en aplicacions Java.

14.2 End to end

El testing end-to-end és una metodologia utilitzada per comprovar si el flux de l'aplicació s'està comportant com s'espera, des del principi fins al final. L'objectiu d'aquest tipus de test és identificar les dependències existents per assegurar que la informació que es passa entre els diversos components és la correcta.

És important ja que fa un testing complet de l'aplicació des de la perspectiva de l'usuari, per evitar els errors més comuns que es donen en escenaris del món real.

14.2.1 Selenium

Per realitzar el testing end-to-end al Front-End s'ha triat Selenium [31].

Selenium és un framework d'Apache open-source per fer testing a aplicacions web. Et proporciona un llenguatge específic (Selenese) per escriure test en els llenguatges de programació més populars. Els tests es poden executar a través de navegadors com Windows, Linux i OS X.

Per automatitzar el testing a navegadors, Selenium WebDriver conté un servidor que obra un navegador i estableix una comunicació amb aquest. El servidor espera que el client enviï comandes a ser executades, llavors transmet aquestes comandes al navegador i mostra resultats. El WebDriver és el client per aquest servidor, permet enviar comandes des d'un llenguatge de programació.

Procés

Les passes que segueix el testing end-to-end simulen les accions que realitzaria l'usuari, per testejar totes les passes de l'aplicació i comprovar que la dependència de dades està sota control i que tot el flux de l'aplicació s'executa correctament.

Un exemple del procés de testing end-to-end amb Selenium, per el mòdul de *Test Drive* de vehicles especials, consistiria en implementar les comandes necessàries perquè es realitzin els següents passos; selecció d'un vehicle especial, selecció del motor, el canvi i la tracció i, al pas següent, la selecció del concessionari i província i ompliment del formulari. Per últim caldrà comprovar la confirmació de la cita o el pas enrere.

L'últim pas després de programar els tests i executar-los, és analitzar els resultats, per saber si coincideix amb l'output esperat o no i analitzar els possibles problemes que s'han observat. Si el test ha fallat, es pot localitzar el problema que té la nostra aplicació veient en quin punt del test ha fallat.

15 DOCUMENTACIÓ DEL CODI

Documentar el codi és essencial per ajudar a fer que qualsevol persona l'entengui, però, lamentablement és molt fàcil que la documentació externa quedi obsoleta a mesura que es van fent canvis. Per això és útil anar documentant el codi mentre es va realitzant, així el resultat és més fàcilment modificable i entenedor.

15.1 JavaDoc

Javadoc [32] és una eina de documentació que defineix un format estàndard pels comentaris i et permet generar un fitxer HTML a partir de codi Java per veure la documentació des d'un navegador.

Per generar APIs amb Javadoc s'han d'utilitzar etiquetes (tags) de HTML o certes paraules reservades precedides pel caràcter "@" . Aquestes etiquetes s'escriuen al principi de cada classe, variable o mètode, començant el comentari amb `/**` i acabat amb `*/`.

15.2 Swagger

L'objectiu de Swagger [33] és poder definir una interfície per les API REST que sigui estàndard, independent de l'idioma i que permeti descobrir i comprendre les capacitats que té un servei sense haver d'accedir al codi font del projecte, a la documentació o inspeccionar el trànsit de la xarxa. Si es defineix adequadament amb Swagger, es pot entendre i interactuar amb el servei remot sense tenir molta idea de la lògica de l'aplicació.

En aquest projecte, a través d'una sèrie d'anotacions al codi, Swagger et genera un entorn on poder veure i provar tots els mètodes realitzats al servei, els paràmetres que necessita i les possibles respostes que pots rebre.

16 ADAPTACIÓ A ALTRES MARQUES

Com s'ha mencionat anteriorment, aquesta solució s'havia d'adaptar a les diverses marques que formen VAESA, un cop es trobarà implementada la marca base Audi, les marques que restarien serien Skoda, Volkswagen i Volkswagen Vehicles Comercials.

Al realitzar la solució tenint present aquest aspecte i comptant amb configuració per entorn (per cada marca es crearia un `configuration.json` amb els paràmetres personalitzats), els canvis serien mínims. En quant es disposa dels dissenys per part dels clients, l'equip de maquetació, que és la part on hi haurà més canvi, maquetarà el resultat per poder-lo integrar a la solució. També es realitzaran petites implementacions personalitzades per marca sol·licitades pels clients, però, en general, els canvis s'haurien de realitzar amb certa facilitat.

17 RESULTATS

En aquesta secció es parlarà sobre els resultats obtinguts tant en l'àmbit d'aplicació, com de documentació i sobre els coneixements adquirits al llarg de la realització de la carrera que s'han posat en pràctica en aquest projecte.

17.1 Documents realitzats

Durant el procés d'aquest TFG, s'han realitzat una sèrie de documents. Primerament s'han entregat els documents acadèmics corresponents a les dates d'entrega que estaven prèviament definides, a més d'aquest es van realitzar d'altres que es creien adients com els documents de requisits i els documents de disseny explicats anteriorment.

Per a la realització d'aquests documents s'ha seguit un estàndard de nomenclatura per a aconseguir més formalitat, organització i traçabilitat.

Per exemple: **AUTBOOK-INF-03-v0.2**

- **AUTBOOK**: Diminutiu del nom del projecte
- **INF**: En aquest cas INF es per "Informe", però també trobem "DR" per els documents de requeriments i "DD" per els documents de disseny.
- **03**: Número d'informe. 01 per l'informe inicial, 02 per l'informe de progrés I i 03 per l'informe de progrés II. Als documents de disseny i de requisits, 01 per *Test Drive* i 02 per *Digital Ordering*.
- **v0.2**: Número de versió

A la següent taula (Taula 1) podem visualitzar tots els documents generats, la nomenclatura d'aquests i les dates en què s'han entregat.

TAULA 1: DOCUMENTS GENERATS (TD: TEST DRIVE, DO: DIGITAL ORDERING)

Document	Nom	Data
Informe Inicial	AUTBOOK-INF-01-v0.2	02/10/2016
Informe de progrés I	AUTBOOK-INF-02-v0.2	06/11/2016
Informe de progrés II	AUTBOOK-INF-03-v0.2	18/12/2016
Document de requisits TD	AUTBOOK-DR-01-v0.2	18/12/2016
Document de requisits DO	AUTBOOK-DR-02-v0.2	18/12/2016
Document de disseny TD	AUTBOOK-DD-01-v0.2	06/11/2016
Document de disseny DO	AUTBOOK-DD-02-v0.2	06/11/2016

17.2 Coneixements aplicats

En aquesta secció es podrà veure una relació sobre els coneixements aplicats adquirits a la universitat en relació a les fases del projecte.

Durant la primera fase del projecte, la captura de requisits, es van aplicar coneixements com la detecció d'actors, les relacions entre ells, la importància de la captura de requisits i posterior documentació d'aquest, com realitzar una bona documentació i el llenguatge apropiat a utilitzar.

Per a la fase de disseny, va ser de gran ajuda saber abstraure el problema i tenir coneixements sobre la creació de diagrames UML de diversos tipus per tal de poder documentar correctament i de manera comprensible per tothom.

A l'hora de realitzar la implementació, és on més s'ha denotat els coneixements previs que es tenien. Per a la part de front-end va ser molt útil conèixer les bases del concepte de Data Binding per comunicar les dades de la vista de l'aplicació amb la lògica. Per altra banda, va ser essencial pel desenvolupament del back-end els coneixements previs que

es tenien sobre Java Avançat i JavaEE, ja que això va estalviar temps d'aprenentatge previ, cosa que va ser necessària amb Angular2.

Per últim, s'ha de ser conscient de la importància que té el procés de testing en el cicle de vida d'un projecte, ja que ajuda a millorar la qualitat, la fiabilitat i el rendiment del sistema. Per aquest TFG s'ha realitzat testing end-to-end i testing unitari la qual cosa ha ajudat a detectar, i per tant solucionar, molts errors que d'altra manera potser no s'haguessin trobat.

17.3 Vistes de l'aplicació

A la Fig. 8 podem visualitzar un exemple d'una vista del resultat de la solució creada per *Digital Ordering* d'Audi, el pas inicial de la revisió de l'oferta. A l'apèndix es pot visualitzar el flux complet (Fig. 15) i exemples de vistes de la solució realitzada per *Test Drive* d'Audi juntament amb un exemple de l'adaptació per Skoda i Volkswagen (Fig. 14).

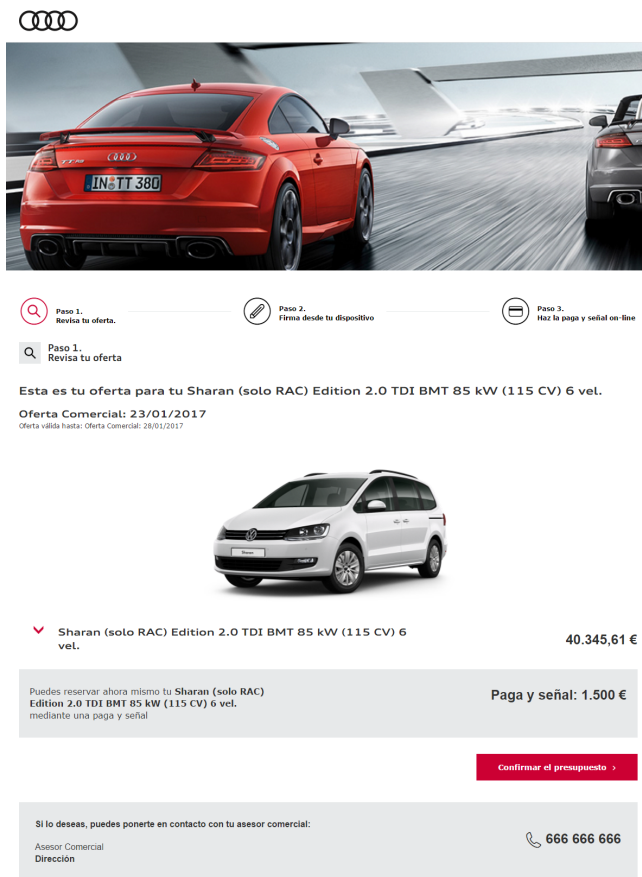


Fig. 8: Vista de l'oferta *Digital Ordering*

17.4 Beneficis

Les noves funcionalitats que s'han implementat per aquest TFG milloraran l'experiència de compra del client, el que comportarà, en primer lloc, un augment del nombre de comandes signades a través del canal en línia, la reducció del nombre de visites als concessionaris, un augment del nombre de reserves de proves dinàmiques respecte a les cerques totals i, per últim l'atracció de més persones al concessionari a través del canal en línia.

18 CONCLUSIONS I LINIES FUTURES

Un cop finalitzat el TFG, podem extreure com a conclusions que, tot i que la feina ha estat ben organitzada i els temps de realització de tasques encertats, per problemes interns de l'empresa no s'han pogut completar (ni se sap quan es podrà) tots els mòduls dels objectius.

Tot i així s'ha pogut veure que s'han arribat a realitzar tots els documents corresponents amb èxit, aspecte que a la llarga serà d'ajuda, ja que tot projecte ha de tenir una bona documentació a la que poder acudir en cas de dubte i tenir-ho tot ben definit, és una bona pràctica que sempre s'hauria de complir.

Fent referència a la metodologia, podem concloure que ha estat encertada, ja que SCRUM ha permès fer reunions regulades i continuades en el temps i així s'ha pogut anar veient dia rere dia el progrés del projecte i detectar possibles errors o bloquejos a temps per resoldre'ls. També ha sigut de gran utilitat a l'hora d'organitzar-se gràcies a la seva estructura de basada en sprints i assignació i puntuació de tasques.

Fent una visió global als objectius (es poden visualitzar amb més detall a l'apartat 2.4), els que s'han complert fins ara són:

- ✓ [OBJ1] Presa de requisits de *Digital Ordering*
- ✓ [OBJ1-2] Presa de requisits de *Test Drive*
- ✓ [OBJ2] Crear la documentació funcional de *Digital Ordering*
- ✓ [OBJ3] Crear documentació tècnica de *Digital Ordering*
- ✓ [OBJ2-2] Crear la documentació funcional de *Test Drive*
- ✓ [OBJ3-2] Crear documentació tècnica de *Test Drive*
- ✓ [OBJ4] Crear documentació oficial TFG
- ✓ [OBJ5] Finalització *Test Drive* d'Audi
- ✓ [OBJ6] *Digital Ordering* Audi
- ✓ [OBJ7] Skoda: Adaptar *Test Drive*
- ✗ [OBJ7] Skoda: Adaptar *Digital Ordering*
- ✗ [OBJ7] Volkswagen: Adaptar *Test Drive*
- ✗ [OBJ7] Volkswagen: Adaptar *Digital Ordering*
- ✗ [OBJ7] Volkswagen vehicles comercials: Adaptar *Test Drive* i *Digital Ordering*

Fig. 9: Estat dels objectius

Com podem observar a la Fig. 11 no s'ha pogut realitzar l'adaptació de la part de *Digital Ordering* per a les diverses marques, però en ser temes externs a l'empresa i relacionats amb les marques per a les quals es treballa, no s'ha pogut influir al respecte.

Encara que per aquest TFG no s'hagin complert tots els mòduls, aquest projecte segueix endavant, es segueixen resolent errors que apareixen a la feina realitzada i realitzant testings, proves i millores per a posar-ho en producció. Gràcies al treball realitzat, en quant se solucionin els aspectes que bloquegen el projecte, les tasques que queden per finalitzar seran un problema menor, ja que l'estructura del projecte ja està feta i és adaptable fàcilment, només caldrà fer canvis mínims per complir els objectius previstos.

AGRAÏMENTS

En primer lloc, agrair al meu tutor, Lluís Gesa, per la preocupació i ajuda constant durant tot el projecte, per estar disponible en qualsevol moment per resoldre els dubtes que pogués tenir.

També agrair als companys de feina, als amics i sobretot al meu pare i a la meua parella per tot el suport emocional i ajuda durant aquesta època.

REFERÈNCIES

- [1] Aktios - https://www.aktios.com/home_es/
- [2] VAESA - <https://www.volkswagen-audi-espana.es>
- [3] Angular2 - <https://angular.io>
- [4] SCRUM - <https://proyectosagiles.org/que-es-scrum>
- [5] Rols SCRUM - <https://www.atlassian.com/agile/scrum>
- [6] Confluence - <https://es.atlassian.com/software/confluence>
- [7] JIRA - <https://es.atlassian.com/software/jira>
- [8] UML - <https://www.uml.org>
- [9] Gliffy - <https://www.gliffy.com>
- [10] JavaEE - <https://www.oracle.com/technetwork/java/javaee/overview/index.html>
- [11] Wildfly - <https://wildfly.org>
- [12] Docker - <https://www.docker.com>
- [13] Maven - <https://maven.apache.org>
- [14] GIT - <https://git-scm.com>
- [15] Jenkins - <https://jenkins.io>
- [16] SonarQube - <https://www.sonarqube.org>
- [17] Nexus - https://www.google.es/intl/es_es/nexus/
- [18] Bitbucket - <https://bitbucket.org>
- [19] JBoss Developer - <https://developers.redhat.com/products/devstudio/overview>
- [20] Oracle - <https://www.oracle.com>
- [21] Visual Studio Code - <https://code.visualstudio.com>
- [22] JavaScript - <https://es.wikipedia.org/wiki/JavaScript>
- [23] TypeScript - <https://www.typescriptlang.org>
- [24] Cordova - <https://cordova.apache.org>
- [25] Signaturit - <https://www.signaturit.com/es>
- [26] Cecabank - <https://www.cecabank.es>
- [27] DPS - <https://www.dps-services-demenagement.com/en>
- [28] IMAWEB - <https://www.imaweb.es>
- [29] CRM - <https://www.webandmacros.com/crm.htm>
- [30] JUnit - <https://junit.org/junit4>
- [31] Selenium - <https://www.seleniumhq.org>
- [32] JavaDoc - <https://www.oracle.com/technetwork/articles/java/index-137868.html>
- [33] Swagger - <https://swagger.io>

APÈNDIX

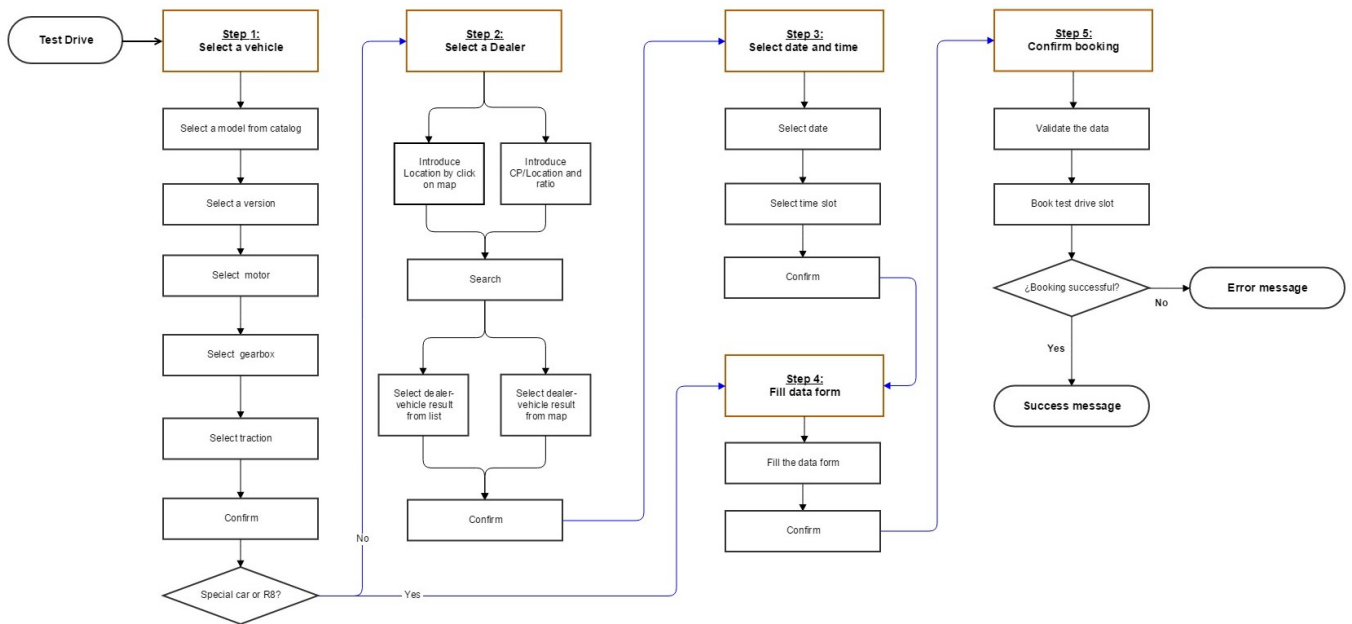


Fig. 10: Mapa de navegació Test Drive

ESTUDI I DISENY:

[OBJ1] Fer una presa de requisits de *Digital Ordering* a partir de actes de reunions amb el client

[OBJ1-2] Fer una presa de requisits de *Test Drive* a partir de actes de reunions amb el client

[OBJ2] Crear la documentació funcional de *Digital Ordering*:

- [OBJ2.1] Diagrama de casos d'ús
- [OBJ2.2] Diagrama seqüencial
- [OBJ2.3] Diagrama d'activitats

[OBJ3] Crear documentació tècnica de *Digital Ordering*:

- [OBJ3.1] Diagrama de classes
- [OBJ3.2] Diagrama de l'arquitectura

[OBJ2-2] Crear la documentació funcional de *Test Drive*:

- [OBJ2-2.1] Diagrama de casos d'ús
- [OBJ2-2.2] Diagrama seqüencial
- [OBJ2-2.3] Diagrama d'activitats

[OBJ3-2] Crear documentació tècnica de *Test Drive*:

- [OBJ3-2.1] Diagrama de classes
- [OBJ3-2.2] Diagrama de l'arquitectura

[OBJ4] Crear documentació oficial TFG:

- [OBJ4.1] Informe inicial
- [OBJ4.2] Informes de progrés
- [OBJ4.3] Informe Final
- [OBJ4.4] Presentació
- [OBJ4.5] Pòster

IMPLEMENTACIÓ:

[OBJ5] Per la part de *Test Drive* d'Audi (Que es la marca escollida com a base del projecte, les demés s'adaptaran a l'implementació d'aquesta), la part restant que faltaria realitzar són:

- [OBJ5.1] Proves d'integració amb la API d'IMAWEB
- [OBJ5.1-2] Implementació front-end
- [OBJ5.2-2] Enviament del formulari a DPS (Back-end)
- [OBJ5.3] Testing i validació del client
- [OBJ5.4] Correcció d'errors

[OBJ6] Per la part de *Digital Ordering*, a la qual també s'utilitzarà com a base la marca Audi, es comença el projecte des de zero, per tant els objectius seran els següents:

- [OBJ6.1-2] Implementació front-end
- [OBJ6.2] Implementar la API del back-end:
 - Login
 - Recuperació dades de l'oferta
- [OBJ6.3] Integració amb la plataforma de firma digital (Signaturit)
- [OBJ6.4] Implementar formulari per la passarel·la de pagament
- [OBJ6.5-2] Proves d'integració amb la API d'IMAWEB
- [OBJ6.5] Realitzar proves UAT (Prova d'acceptació de l'usuari) i testing end-to-end
- [OBJ6.6] Testing i validació del client
- [OBJ6.7] Correcció d'errors

[OBJ7] Tant per la part de *Test Drive* com per *Digital Ordering*, com hem agafat de base la marca Audi, s'haurà d'adaptar la pàgina web a les demés marques (Volkswaguen, Skoda i Volkswaguen vehicles comercials), això inclourà:

- [OBJ7.1] Anàlisi de la proposta de la UI
- [OBJ7.2] Adaptar la API i la funcionalitat a la marca
- [OBJ7.4] Proves UAT i test end-to-end
- [OBJ7.5] Testing i validació del client
- [OBJ7.6] Correcció d'errors

Fig. 11: Llistat d'objectius

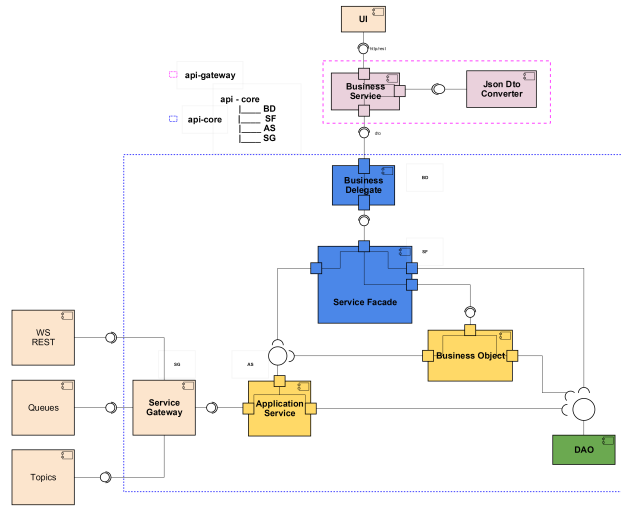


Fig. 13: Arquitectura Back-End (Realitzada per l'arquitecte d'Aktios)

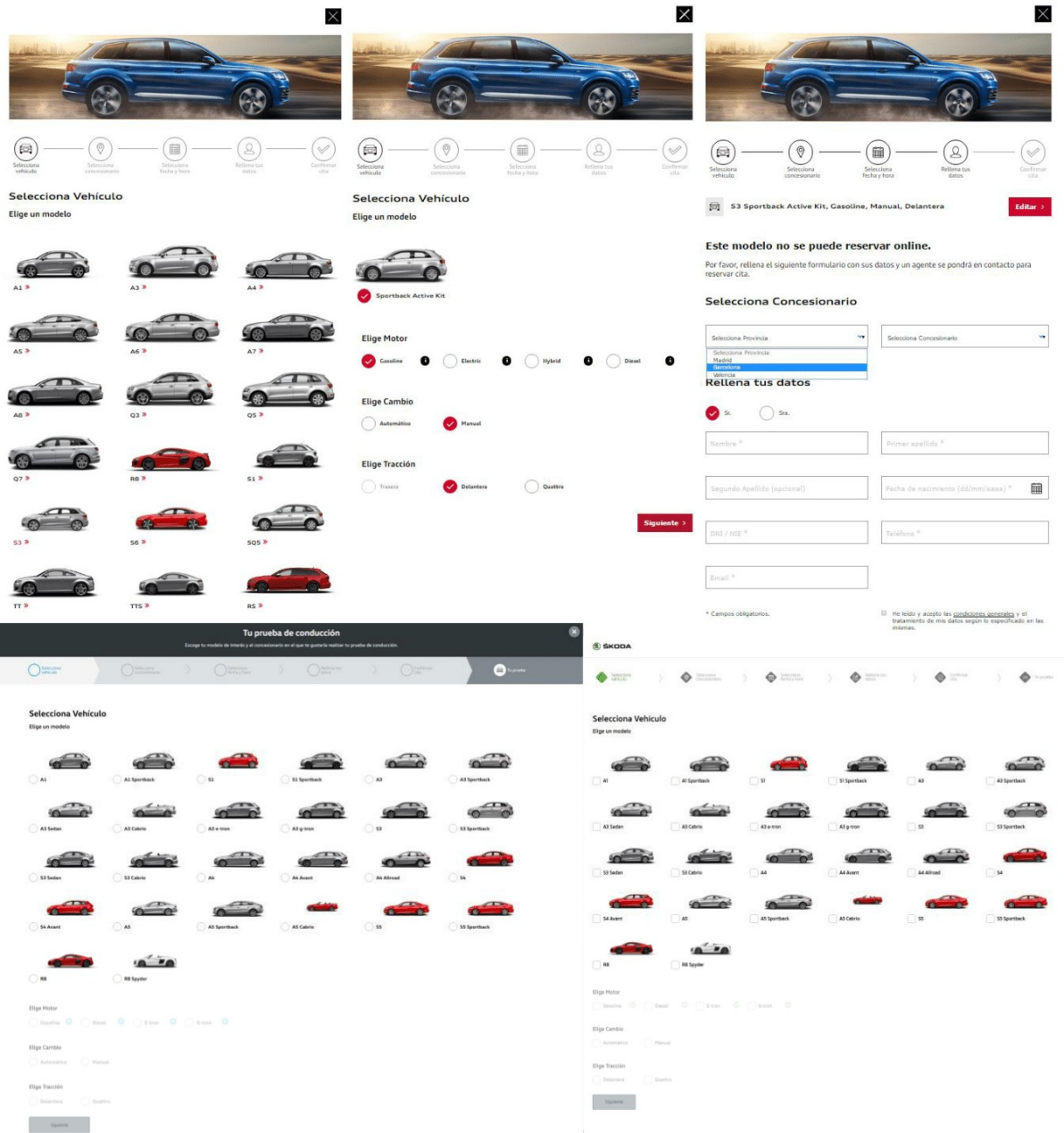


Fig. 14: Vistes Test Drive (Audi, Skoda i Volkswagen)

Hola Aktios,

Te presentamos el presupuesto de tu Sharan (solo RAC) Edition 2.0 TDI BMT 85 kW (115 CV) 6 vel. realizado en el concesionario IMA Pruebas QA, para confirmar el pedido sólo debes seguir los siguientes tres pasos.

Paso 1. Revisa tu oferta. **Paso 2. Firma desde tu dispositivo.** **Paso 3. Haz la paga y señal on-line.**

Para poder ver la oferta, primero deberás identificarte

Introduce tu DNI / NIE

[Entrar >](#)

Si lo deseas, puedes ponerte en contacto con tu asesor comercial:

Asesor Comercial Dirección **666 666 666**

Sharan (solo RAC) Edition 2.0 TDI BMT 85 kW (115 CV) 6 vel. **40.345,61 €**

Paga y señal: 1.500 €

[Confirmar el presupuesto >](#)

Es necesario firmar en la página 1

Exposición SIM AUTO ADBI
 Paseo de la Castellana, 105, 28050 Madrid, Madrid
 Cif: N2050701
 Tel: 916420001 Fax: 916420002
 E-mail: SIMAutoV@sim.es
 Web: www.sim.es/SIMAutoV www.sim.es

Das WellAuto.
 Vehículos de Ocasión de calidad. Garantizados.

Pedido vehículo de ocasión

Fecha del pedido:	23 de Enero de 2017	Asesor comercial:	Dirección Auto 1
Número del pedido:	1847	Fecha estimada de entrega:	

Datos del Comprador:

Nombre:	Alfons Aktios Aktios	Calle:	Arceas, 9
Apellidos:	92264620F	Madrid:	28033
D.N.I. / C.I.F.:	460147500	Provincia:	Madrid
Teléfono de contacto:		E-mail:	emartin@imaweb.net

Datos del Vehículo:

VOLKSWAGEN Polo 1.4 TDI BMT Sport 105 (2015)

Matrícula:	5585PU
Fecha matriculación:	1/02/2015
Kilómetros:	48752
Botones:	96325484421262
Garantía:	12 meses
Servicio anterior:	Autosid
Combustible:	Diesel
Potencia:	78 Kw (105 CV)
Color:	Rajo Atanador metalizado
Consumo (l/100km):	Ciudad: 5,9 / Carretera: 3,2 / Combinado: 3,4
Emissiones:	90 g/km

Precio del vehículo: 12.000,00 €
 Base imponible: 9.912,64 €
 IVA Normal (21%): 2.087,34 €
Precio final impuestos incluidos: 12.250,00 €

FIRMAR

REVISAR DOCUMENTO

CONFIRMAR EL PRESUPUESTO

¿Cómo funciona?

- Revisa el documento y presiona **FIRMAR** cuando estés listo.
- Firma en la pantalla de tu dispositivo.
- Presiona **ENVIAR** para finalizar el proceso.

VolksWagen Bank **Compra on-line**

cecbank **Confirmación del pago**

Forma de pago: Pago con tarjeta (VISA, Mastercard, etc.), Tarjetero EURO 6000

Datos del pago: Número de operación: 1VN-7290, Importe: 1500 €, Fecha caducidad (MM/AAAA): 01 / 2017

PAGAR

El documento se ha firmado correctamente. Ya puedes hacer la paga y señal.

Paso 3. Haz la paga y señal.

Ahora debes realizar el pago de la paga y señal.

Paga y señal: 1500 €

[Hacer la paga y señal >](#)

Ver la oferta > **Ver contrato >**

Si lo deseas, puedes ponerte en contacto con tu asesor comercial: **666 666 666**

El pago se ha realizado correctamente.

Gracias por realizar la paga y señal. Tu asesor comercial se pondrá en contacto contigo. Si lo deseas, puedes descargar el contrato que has firmado y revisar la oferta.

Si lo deseas, puedes ponerte en contacto con tu asesor comercial: **666 666 666**

Fig. 15: Vistes Digital Ordering