
Enginyeria del Software

Anàlisi i Disseny *Airhopping*

Versió 2.3

Membres del grup

NIA	Nom	Cognoms
1535791	Arnau	Closa Manteca
1420576	David	Méndez Ribera
1492658	Lucas	Gonzalo Priego
1494644	Victor	Castro Garcia
1423759	Joan	Rosell Rodríguez

Historial de Revisions

Data	Versió	Descripció	Autor
17/02/2020	0.1	Proposta inicial del document i pluja d'idees de requeriments.	Professors de la assignatura.
24/02/2020	0.2	Estructuració dels requisits en mòduls i primers requeriments sense classificació.	Grup.
26/02/2020	0.3	Creació parcial de requeriments per mòdul i començament de la seva classificació.	Lucas Gonzalo, Victor Castro
26/02/2020	0.4	Finalització de selecció de requeriments i inici de la formalització d'aquests per a l'entrega.	Joan Rosell, Arnau Closa, David Méndez
26/02/2020	0.5	Retocs i maquetat final dels requisits a l'informe.	Lucas Gonzalo, Joan Rosell, Arnau Closa
27/02/2020	0.6	Implementació requeriments d'última hora, revisió del projecte en la seva totalitat i entrega.	Grup.
28/02/2020	0.7	Petits canvis en els requeriments post-entrega Practica 1.	Grup.
10/04/2020	1.0	Primera planificació i pluja d'idees sobre el diagrama de casos d'ús.	Grup.
15/04/2020	1.1	Primera implementació del diagrama de casos d'ús i plantejament del diagrama de classes.	Joan Rosell, Arnau Closa, Lucas Gonzalo
16/04/2020	1.2	Creació de l'esbós de diagrama de classes i idees sobre els diagrames de seqüència.	Grup.
17/04/2020	1.3	Primers diagrames de seqüència i retocs en el diagrama de classes.	David Méndez, Victor Castro, Joan Rosell
18/04/2020	1.4	Reestructuració diagrama de classes, petits retocs diagrama de casos d'ús, implementació diagrames de seqüència restants.	Grup
19/04/2020	1.5	Darreres revisions en els diferents apartats de la Pràctica 2.	Grup
21/05/2020	2.0	Estructuració en apartats de la tercera pràctica.	Lucas Gonzalo
21/05/2020	2.1	Implementació de l'apartat de Scrum.	Lucas Gonzalo
23/05/2020	2.2	Implementació apartat test unitaris.	Grup
23/05/2020	2.3	Correcció de faltes d'ortografia i revisió final.	Grup

Taula de Contingut

1. Requisites	5
Introducció	5
Requisits funcionals	5
Requisits no funcionals	8
2. Casos d'ús	16
Introducció	16
Actors	16
Casos d'Ús	16
Vista	17
3. Especificació dels casos d'ús	18
4. Diagrama de classes	27
Introducció	27
Classes	28
Vista	31
5. Diagrames de seqüència	32
Introducció	32
Vistes	32
6. Test unitaris	36
7. Scrum	40

1. Requisits

Introducció

En aquest apartat es defineixen els diferents requeriments que es faran servir per desenvolupar la pagina web Airhopping, classificats a partir de diferents criteris com, la importància front al producte final o les verificacions a establir, i la jerarquia formada entre aquests.

Així mateix, destaquem clarament dos blocs principals de requeriments, essent els funcionals, aquells que fan referencia a les funcions pràctiques o programables del software, i els no funcionals, aquells relacionats amb l'entorn del software i les seves característiques.

A més, l'identificador (ID) de cada requisit es mostrarà segons el tipus de requisit (F - NF), el mòdul al que pertany (INT, USR, etc.) i la subdivisió a la qual pertany.

En les següents seccions es fan servir els següents assumpcions: En relació a les Prioritats:

- A : Prioritat Alta.
- M : Prioritat Mitja.
- B : Prioritat Baixa.

En relació a les Verificacions:

- R : Verificació per Revisió.
- T : Verificació per Test.
- D : Verificació per Disseny.

Requisits funcionals

ID	R-F-ALG-01
Títol	Algorisme (Ordre)
Descripció	Algoritme que en rebre una sèrie de preus, els estructuri de manera ascendent mostrant els preus mes baixos els primers.
Prioritat	B
Verificació	R, T
Pares	

ID	R-F-ALG-02
Títol	Algorisme (Neteja de dades sessió)
Descripció	Algoritme pel qual un cop finalitzada la reserva tota la informació de la sessió serà netejada.
Prioritat	A
Verificació	R, T
Pares	

ID	R-F-INT-01-01
Títol	Interfície (Menú Principal)
Descripció	S'ha de mostrar un botó que ens permeti iniciar la recerca del nostre viatge únicament tenint en compte el nombre de viatgers per, a continuació, demanar la ciutat d'origen, i les dates d'anada i tornada.
Prioritat	A
Verificació	R, T
Pares	

ID	R-F-INT-01-02
Títol	Interfície (Destinacions)
Descripció	Tenint en compte l'origen i les dates es mostraran les diferents destinacions disponibles per preu.
Prioritat	A
Verificació	R, T
Pares	R-F-INT-01-01, R-F-ALG-01-01

ID	R-F-INT-01-03
Títol	Interfície (Informació Vols)
Descripció	A partir del destí escollit, es mostrarà la informació dels vols suggerits per la pàgina i la possibilitat d'afegir més destinacions a la reserva i modificar els vols, sempre mostrant el preu per persona.
Prioritat	A
Verificació	R, T
Pares	R-F-ALG-01-01, R-F-INT-01-02

ID	R-F-INT-01-04
Títol	Interfície (Allotjaments)
Descripció	A partir dels vols seleccionats, es mostraran els allotjaments disponibles en cada una de les destinacions prèviament seleccionades.
Prioritat	A
Verificació	R, T
Pares	R-F-ALG-01-01, R-F-INT-01-03

ID	R-F-INT-01-05
Títol	Interfície (Vehicles)
Descripció	Tenint en compte els vols seleccionats, es donarà l'opció de llogar un vehicle en qualsevol dels destins que el client ha seleccionat, tenint en compte les seves restriccions en quant a nombres de passatgers o característiques del vehicle que desitgi.
Prioritat	M
Verificació	R, T
Pares	R-F-ALG-01-01, R-F-INT-01-03

ID	R-F-INT-01-06
Títol	Interfície (Menú Client)
Descripció	A partir de tota la informació del viatge, es demanarà diferents dades sobre la informació personal dels usuaris i el equipatge.
Prioritat	A
Verificació	R, T
Pares	R-F-INT-01-03, R-F-INT-01-04, R-F-INT-01-05

ID	R-F-INT-01-07
Títol	Interfície (Menú pagament)
Descripció	Finalment, els usuaris hauran de emplenar les dades bancaries per tal de rebre un localitzador de la seva reserva..
Prioritat	A
Verificació	R, T
Pares	R-F-INT-01-06

ID	R-F-BD-01
Títol	Estructura de la Base de dades
Descripció	Les taules que guardaran la informació de cada reserva seran Client, Vols, Allotjament, Cotxe.
Prioritat	A
Verificació	R, T, D
Pares	

Requisits no funcionals

ID	R-NF-INT-01-01
Títol	Interfície (Preu Actualitzat)
Descripció	En una finestra lateral, sempre se'ns mostrarà el preu per persona actualitzat del nostre viatge.
Prioritat	M
Verificació	R, T
Pares	R-F-INT-01

ID	R-NF-INT-01-02
Títol	Interfície (Suggeriments diaris)
Descripció	En la pagina principal, s'ha de mostrar els viatges més barats del dia, tenint en compte restriccions pel que fa nombre de persones, origen i dates.
Prioritat	A
Verificació	R, T
Pares	R-F-INT-01-01

ID	R-NF-INT-01-03
Títol	Interfície (Contacte amb els administradors)
Descripció	S'ha d'incloure una pestanya de contacte amb els administradors de la pàgina.
Prioritat	M
Verificació	R, T
Pares	R-F-INT-01-01

ID	R-NF-INT-01-04
Títol	Interfície (Preguntes freqüents)
Descripció	En una pestanya es mostraran un conjunt de preguntes y dubtes freqüents dels clients.
Prioritat	M
Verificació	R, T
Pares	R-F-INT-01-01

ID	R-NF-INT-01-05
Títol	Interfície / Mòdul Software extern (Connexió amb lloguer d'allotjament)
Descripció	Per mostrar l'allotjament interactuarem directament amb la pagina web de reserva d'allotjament "TripAdvisor", mostrant a l'usuari una finestra amb el mateix format que la pàgina d'origen.
Prioritat	A
Verificació	R, T
Pares	R-F-ALG-01, R-F-INT-01-04

ID	R-NF-INT-01-06
Títol	Interfície / Mòdul Software extern (Connexió lloguer de vehicles)
Descripció	Per mostrar les opcions de lloguer de vehicles interactuarem directament amb la pàgina web de lloguer de vehicles "Rentalcars", mostrant a l'usuari una finestra amb el mateix format que la pàgina d'origen.
Prioritat	A
Verificació	R, T
Pares	R-F-ALG-01, R-F-INT-01-05

ID	R-NF-INT-01-07
Títol	Interfície / Mòdul Software extern (Connexió comparador de vols)
Descripció	Per mostrar la informació del vols interactuarem directament amb el mòdul extern de reserva de vols "Lanzadera", mostrant a l'usuari una finestra amb el mateix format que la pàgina d'origen.
Prioritat	A
Verificació	R, T
Pares	R-F-ALG-01, R-RF-INT-01-03

ID	R-NF-USR-01
Títol	Persistència de dades
Descripció	Les dades de l'usuari romandran en la web fins que no finalitzi el procés de reserva.
Prioritat	A
Verificació	R, T
Pares	R-F-ALG-02

ID	R-NF-USR-02
Títol	Modificació de dades
Descripció	L'usuari ha de poder modificar totes les dades introduïdes fins que no finalitzi el procés de reserva.
Prioritat	A
Verificació	R, T
Pares	

ID	R-NF-USR-03
Títol	Rols i permisos d'usuari
Descripció	Existiran diversos rols amb diferents permisos per usuaris (DBA, Webmaster, etc...)
Prioritat	A
Verificació	R, T, D
Pares	

ID	R-NF-BD-01
Títol	Emmagatzematge de reserves
Descripció	Les reserves s'hauran d'emmagatzemar en una base de dades pròpia per tal de garantir la traçabilitat en tot el procés de compra.
Prioritat	A
Verificació	R, T, D
Pares	R-F-BD-01

ID	R-NF-ALG-01
Títol	Eficiència dels algorismes
Descripció	Els algorismes emprats en el processament d'informació hauran de ser el més eficients possible.
Prioritat	A
Verificació	R, T
Pares	

ID	R-NF-ALG-02-01
Títol	Algorisme principal
Descripció	S'haurà d'implementar un algorisme per optimitzar rutes de vol amb l'objectiu d'abaratir costos.
Prioritat	A
Verificació	R, T, D
Pares	R-F-INT-01-03, R-NF-ALG-01-01

ID	R-NF-ALG-02-02
Títol	Algorismes secundaris
Descripció	Es crearan i ampliaran diverses famílies d'algorismes que serveixin per ajudar l'algorisme principal a decidir destinacions a partir de les opcions d'allotjament i vehicles.
Prioritat	A
Verificació	R, T
Pares	R-F-INT-01-03, R-NF-ALG-02-01

ID	R-NF-HW-01
Títol	Aplicació multi-plataforma
Descripció	L'aplicació ha de ser compatible per a diferents sistemes operatius i plataformes com Windows, Android, IOS, etc.
Prioritat	A
Verificació	R, T
Pares	R-F-ALG-01

ID	R-NF-SEC-01-01
Títol	Validació dels formularis
Descripció	Tots els formularis es validaran tant al Front-End com al Back-End utilitzant els mecanismes més adients en funció de la plataforma destí de l'aplicació.
Prioritat	A
Verificació	R, T
Pares	R-F-INT-01-07

ID	R-NF-SEC-01-02
Títol	Validacions HTML al Front-End
Descripció	Els formularis faran servir validació HTML5 en la mesura del possible.
Prioritat	A
Verificació	R, T
Pares	R-NF-SEC-01-01

ID	R-NF-SEC-01-03
Títol	Validacions JavaScript al Front-End
Descripció	Els formularis mes importants implementaran mecanismes de validació interactius per millorar la seguretat i l'experiència d'usuari.
Prioritat	M
Verificació	R, T
Pares	R-NF-SEC-01-01

ID	R-NF-SEC-01-04
Títol	Validacions al Back-End
Descripció	Tota la informació provinent d'un formulari serà validada y filtrada abans de ser processada.
Prioritat	A
Verificació	R, T
Pares	R-NF-SEC-01-01

ID	R-NF-SEC-02
Títol	Prevenició d'atacs XSS (Cross-Site Scripting)
Descripció	Es prendran mesures a l'hora de mostrar la informació a l'usuari per evitar atacs XSS
Prioritat	A
Verificació	R, T
Pares	R-F-INT-01-01

ID	R-NF-RD-01
Títol	Rendiment (Capacitat d'usuaris)
Descripció	El sistema ha de tenir capacitat per donar servei a un mínim de 500 usuaris simultanis, que naveguin per la web, facin consultes, etc.
Prioritat	M
Verificació	R, T
Pares	R-F-INT-01-01

ID	R-NF-RD-02
Títol	Rendiment (Temps de resposta)
Descripció	El cercador ha de retornar els resultats de la cerca desitjada en menys de 3 segons.
Prioritat	A
Verificació	R, T
Pares	R-F-INT-01-01

ID	R-NF-RD-03
Títol	Rendiment (Temps de connexió)
Descripció	El sistema ha de ser capaç de donar a l'usuari connexió a la pàgina web en menys de 5 segons.
Prioritat	M
Verificació	R, T
Pares	R-F-INT-01-01

ID	R-NF-RE-01
Títol	Rendiment (Emmagatzematge de la informació)
Descripció	Tota la informació dels usuaris i les seves reserves es desarà en un disc dur de 10TB.
Prioritat	M
Verificació	R, T
Pares	R-F-BD-01

ID	R-NF-FIR-01-01
Títol	Recuperació davant d'errors (Reinici automàtic)
Descripció	Els serveis web han de ser operatius durant tots els dies de l'any, i en cas d'atur involuntari, han de reiniciar-se automàticament en menys de 5 minuts.
Prioritat	A
Verificació	R, T
Pares	

ID	R-NF-FIR-01-02
Títol	Recuperació davant d'errors (Avis atur dels serveis)
Descripció	En el cas que el sistema intenti reiniciar-se tres vegades sense èxit, ha d'enviar un missatge d'error als administradors de manera immediata.
Prioritat	M
Verificació	R, T
Pares	R-NF-FIR-01-01

ID	R-NF-FIR-02
Títol	Fiabilitat davant d'errors (Mètodes anti-pèrdua d'informació)
Descripció	El servidor ha de tenir protocols i estructures per evitar la pèrdua d'informació durant tot el procés de selecció del viatge.
Prioritat	A
Verificació	R, T
Pares	R-F-FIR-01

ID	R-NF-OD-01
Títol	Aplicació User Friendly
Descripció	Tot el sistema de cerca ha de ser User Friendly.
Prioritat	A
Verificació	R, T
Pares	R-F-INT-01-01

ID	R-NF-RIE-01
Títol	Interacció amb usuari (Botó inici)
Descripció	El cercador de viatges personalitzat ha de tenir un botó per començar on digui: 'Empezar Viaje'.
Prioritat	B
Verificació	R, T
Pares	R-F-INT-01-01

ID	R-NF-EST-01
Títol	Estàndards d'enciptació
Descripció	La comunicació entre el Back-End i el Front-End serà xifrada mitjançant l'estàndard d'enciptació AES-256.
Prioritat	A
Verificació	R, T
Pares	R-NF-SEC-01-01, R-NF-USR-01, R-NF-USR-02

ID	R-NF-EST-02
Títol	Estàndard d'arquitectura REST
Descripció	S'utilitzarà l'estàndard REST per l'arquitectura del web.
Prioritat	B
Verificació	R, T
Pares	R-NF-HW-01,

2. Casos d'ús

Introducció

El propòsit d'aquest apartat és definir les funcionalitats del sistema (casos d'ús) i el context d'aquest (interacció amb entitats externes o actors). A continuació es presenten els diferents diagrames de casos d'ús per als paquets en què es divideix el sistema i les descripcions resumides de cada actor i cas d'ús.

El model de casos d'ús que es presenta consta d'un paquet i el respectiu diagrama de casos d'ús.

En el següent diagrama es mostra l'arquitectura inicial del software en termes dels paquets en què es divideix.

Actors

Usuari

Aquest actor representa a l'usuari, la seva funció és representar les accions que faria un Usuari qualsevol que accedeix a la pàgina com a client. Més concretament, en el nostre problema d'AirHopping representaria un client que vol planificar el seu viatge.

SkyScanner

SkyScanner representa la plataforma que conté tota la informació sobre els vols. Per tant, el seu objectiu dins del nostre projecte serà donar la informació sobre els vols al sistema sempre que aquest ho requereixi.

Booking

Booking representa la plataforma que conté tota la informació sobre els allotjaments. Per tant, el seu objectiu dins del nostre projecte serà donar la informació sobre els allotjaments al sistema sempre que aquest ho requereixi.

RentalCars

RentalCars representa la plataforma que conté tota la informació sobre els vehicles. Per tant, el seu objectiu dins del nostre projecte serà donar la informació sobre els vehicles al sistema sempre que aquest ho requereixi.

Plataforma de pagaments

Aquest actor representa la plataforma que s'encarrega de gestionar les funcionalitats tant d'orientació a l'API desitjada (Visa o Mastercard), com del pagament i devolució de diners, en cas que el sistema ho sol·liciti.

Visa

Visa representa l'API d'aquesta companyia. L'objectiu d'aquest actor és gestionar els pagaments que es realitzin utilitzant targetes gestionades per aquesta empresa.

Mastercard

Mastercard representa l'API d'aquesta companyia. L'objectiu d'aquest actor és gestionar els pagaments que es realitzin utilitzant targetes gestionades per aquesta empresa.

Casos d'Ús

Llistat Vols

El sistema disposarà d'un mòdul per buscar els vols més barats entre unes dates seleccionades i un destí. El mòdul seleccionarà els vols més econòmics per sortir i arribar a la ciutat origen.

Selecció Vols Intermitjos

El sistema buscarà vols intermitjos entre dues dates a diferents ciutats, on la primera i l'última són la ciutat d'origen de l'usuari.

Canvi vol

L'usuari canvia un dels vols seleccionats anteriorment pel d'una llista de vols alternatius.

Pagament de Viatge

El sistema calcula el cost total del viatge i realitza el cobrament mitjançant les diferents metodologies de pagament.

Buscador vols

El sistema busca un o més vols segons l'origen i les dades seleccionades per l'usuari.

Llistat Cotxes

El sistema ha de disposar d'un mòdul per buscar els cotxes de lloguer més barats. El mòdul selecciona els cotxes més econòmics a la ciutat destí.

Selecció cotxe

L'usuari pot seleccionar un dels cotxes de lloguer que s'han passat des del llistat de cotxes.

Cancel·lació Reserva

L'usuari pot cancel·lar una reserva a partir de l'identificador. Un cop cancel·lada, el sistema ha d'esborrar aquesta reserva en les plataformes involucrades i retornar l'import segons la forma de pagament.

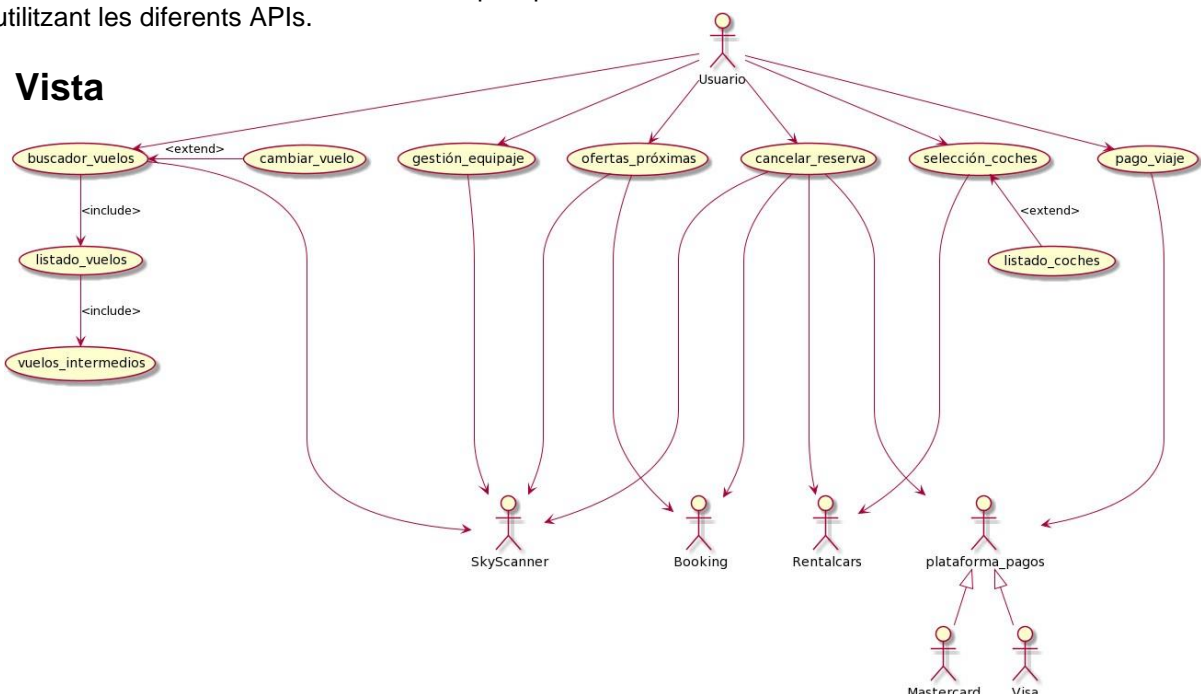
Gestió Equipatge

L'usuari decideix gestionar el seu equipatge, tenint en compte els vols seleccionats prèviament.

Ofertes Pròximes

El sistema mostrarà les ofertes trobades pels pròxims dies. Ho farà consultant els vols i els hotels utilitzant les diferents APIs.

Vista



3. Especificació dels casos d'ús

Cas d'Ús	Llistat Vols		
Versió	1.2	Data	16/04/2020
Autors	Joan, Víctor		
Descripció	El sistema haurà de disposar d'un mòdul per buscar vols més barats entre unes dates seleccionades i un destí. El mòdul seleccionarà els vols més econòmics per sortir i arribar a la ciutat origen.		
Actors	Usuari		
Precondició	Per fer la cerca l'usuari haurà introduït les dates inicial, final i la ciutat d'origen i s'haurà generat un llistat amb tots els vols possibles.		
Flux Principal	1. El mòdul rep un llistat de vols que compleixin els paràmetres de cerca. 2. Es crida a un submòdul que selecciona els vols més barats que uneixen les ciutats, buscant un camí que minimitzi el cost mitjà del trajecte.		
Subfluxos			
Fluxos alternatius	Si el llistat està buit, el procés s'atura prematurament i es retorna un llistat buit amb un identificador d'error.		
Postcondició	S'ha filtrat el llistat de vols inicials i s'han seleccionat un conjunt de vols que minimitzen el cost del trajecte.		
Requeriments no funcionals	REG-NF-1-01: Selecció de vols - paràmetres REG-NF-1-04: Ordenació		
Prioritat	Alta		
Comentaris			

Cas d'Ús		Selecció Vols Intermitjos	
Versió	1.1	Data	18/04/2020
Autors	Joan, Victor		
Descripció	El sistema busca vols intermitjos entre dues dates a diferents ciutats, on la primera i l'última són la ciutat d'origen de l'usuari.		
Actors	Usuari		
Precondició	Per fer la tria de vols es necessita un llistat de vols amb totes les combinacions de data/ciutat.		
Flux Principal	<ol style="list-style-type: none"> 1. Es processa el llistat de vols en una matriu d'adjacències amb els diferents costos dels vols. 2. Sobre aquesta matriu aplicarem l'algorisme de Floyd-Warshall per computar els camins de cost mínim entre totes les parelles de ciutats. 3. Un cop processada la matriu escollirem el camí que compleixi les següents propietats: <ol style="list-style-type: none"> 3.1. La longitud màxima del camí no ha de superar 4 destinacions. 3.2. El camí ha de ser el més barat possible. 4. Retornem el camí amb el cost total. 		
Subfluxos			
Fluxos alternatius			
Postcondició			
Requeriments no funcionals	REG-NF-1-02: Selecció vols intermitjos cost 0 REG-NF-1-03: Màxim nombre de vols REG-NF-1-04: Ordenació		
Prioritat	Alta		
Comentaris	Un altre algorisme eficient per computar el problema ASP (All-pairs Shortest Path) és l'algorisme de Bellman-Ford.		

Cas d'Ús	Canvi Vol		
Versió	1.2	Data	16/04/2020
Autors	Arnau, Lucas		
Descripció	L'usuari canvia un dels vols seleccionats anteriorment pel d'una llista de vols alternatius.		
Actors	Usuari		
Precondició	Existència d'un o més vols en la reserva.		
Flux Principal	1. L'usuari selecciona el vol a canviar amb el cursor. 2. Es mostra un llistat d'alternatives econòmiques a l'usuari. 3. L'usuari selecciona el nou vol desitjat. 4. S'actualitzen les dades de reserva de l'usuari.		
Subfluxos			
Fluxos alternatius			
Postcondició	1. Canvis en els preus de reserva. 2. Canvis en la cerca d'hotels i cotxes respecte al nou destí.		
Requeriments no funcionals			
Prioritat	Alta		
Comentaris			

Cas d'Ús	Pagament Viatge		
Versió	1.2	Data	17/04/2020
Autors	Arnau, David		
Descripció	El sistema calcula el cost total del viatge i realitza el cobrament mitjançant les diferents metodologies de pagament.		
Actors	Usuari, Plataforma pagos		
Precondició	Passar per la finestra de resum del viatge i pressionar el botó per anar a la finestra de pagament.		
Flux Principal	<ol style="list-style-type: none"> 1. El sistema calcula el preu total de totes les seleccions de l'usuari i el mostra. 2. L'usuari anota les seves dades personals. 3. L'usuari selecciona una metodologia de pagament amb targeta (Visa o Mastercard). 4. L'usuari dóna les dades de la targeta necessàries per procedir al pagament. 5. L'usuari prem el botó de pagar. 6. El sistema es posa en contacte amb la plataforma de pagaments i envia les dades de l'usuari. 7. El sistema envia un correu electrònic a l'usuari amb el resultat de l'operació que ha retornat la plataforma de pagaments. 8. Es dóna un identificador de la reserva a l'usuari. 		
Subfluxos	En qualsevol moment l'usuari pot tornar enrere i sortir de la finestra de pagament, esborrant totes les dades generades o escrites prèviament.		
Fluxos alternatius	<i>Pagament no realitzat:</i> En el cas que la plataforma de pagament impedeixi l'operació de pagament, el sistema ha de mostrar a l'usuari el missatge d'error corresponent.		
Postcondició			
Requeriments no funcionals	REG-NF-1-05: Plataforma de Pagament (APIs Visa i Mastercard)		
Prioritat	Alta		
Comentaris			

Cas d'Ús	Buscador Vols		
Versió	1.3	Data	16/04/2020
Autors	Victor, David		
Descripció	El sistema busca un o més vols segons l'origen i les dades seleccionades per l'usuari.		
Actors	Usuari, SkyScanner		
Precondició			
Flux Principal	1. L'usuari introdueix el nombre de persones i el seu origen. 2. El sistema es posa amb contacte amb l'API de vols SkyScanner i rep una llista de vols amb les característiques demanades. 3. Es crida al cas d'ús "Llistat Vols". 4. El sistema imprimeix per pantalla els vols amb els preus corresponents.		
Subfluxos	En qualsevol moment de la cerca, l'usuari pot editar les persones que viatgen, la data i l'origen.		
Fluxos alternatius			
Postcondició			
Requeriments no funcionals	REG-NF-1-06: Meta-buscador de vols		
Prioritat	Alta		
Comentaris			

Cas d'Ús	Llistat cotxes		
Versió	1.2	Data	17/04/2020
Autors	David, Arnau		
Descripció	El sistema ha de disposar d'un mòdul per buscar els cotxes de lloguer més barats. El mòdul selecciona els cotxes més econòmics a la ciutat destí.		
Actors	Usuari		
Precondició	És necessari saber el destí del vol, i la durada de l'estada.		
Flux Principal	1. El mòdul rep un llistat de cotxes que compleixin els paràmetres de cerca (ciutat destí i durada igual). 2. El sistema selecciona els cotxes més barats i organitza la llista.		
Subfluxos	Si es modifica la duració del lloguer de cotxe, es tornarà a fer la cerca amb aquests nous paràmetres.		
Fluxos alternatius	Si el llistat està buit, el procés s'atura prematurament i es retorna un llistat buit amb un identificador d'error.		
Postcondició	S'ha filtrat el llistat de lloguer de cotxes inicial i s'han seleccionat un conjunt de cotxes amb els paràmetres establerts i amb els costos més baixos.		
Requeriments no funcionals	REG-NF-1-04: Ordenació REG-NF-1-07: API de cotxes		
Prioritat	Alta		
Comentaris			

Cas d'Ús	Selecció Cotxe		
Versió	1.2	Data	17/04/2020
Autors	David, Lucas		
Descripció	L'usuari pot seleccionar un dels cotxes de lloguer que s'han passat des del llistat de cotxes.		
Actors	Usuari, Rentalcars.		
Precondició	Per fer la selecció de cotxe, s'ha d'haver creat el llistat de cotxes disponible.		
Flux Principal	1. El sistema es posa en contacte amb l'API de cotxes Rentalcars i envia les dades de preferència del client d'acord amb les seleccions anteriors de vols i hotels. 2. El sistema rep un llistat de cotxes. 3. Es crida al cas d'ús "Llistat cotxes". 4. L'usuari selecciona l'opció que desitgi i el nombre de dies.		
Subfluxos	L'usuari pot canviar el cotxe escollit i continuar des d'on ho havia deixat. Si ho fa, cal recalcular el preu del viatge en funció del cost del cotxe seleccionat.		
Fluxos alternatius	L'usuari pot no seleccionar cap cotxe, el que resulta en evitar fer cap modificació al preu del viatge.		
Postcondició			
Requeriments no funcionals	REG-NF-1-04: Ordenació REG-NF-1-07: API de cotxes REG-NF-1-09: Vista de quadrícula REG-NF-1-10: Ofertes recents		
Prioritat	Alta		
Comentaris			

Cas d'Ús	Cancel·lació Reserva		
Versió	1.1	Data	18/04/2020
Autors	Victor, Arnau		
Descripció	L'usuari pot cancel·lar una reserva a partir de l'identificador. Un cop cancel·lada, el sistema ha d'esborrar aquesta reserva en les plataformes involucrades i retornar l'import segons la forma de pagament.		
Actors	Usuari, SkyScanner, Booking, Rentalcars, Plataforma pagos		
Precondició	L'usuari ha d'haver completat i pagat una reserva.		
Flux Principal	<ol style="list-style-type: none"> 1. El sistema ofereix l'opció de cancel·lació de reserva. 2. L'usuari selecciona aquesta opció i insereix l'identificador de la reserva. 3. Si l'usuari prem el botó de 'Cancel·lar', el sistema anul·la la cancel·lació. Si decideix seguir amb aquesta i confirma, el sistema es posa en contacte amb les diferents APIs de reserva i pagament per cancel·lar les reserves i fer la devolució de l'import. 4. El sistema interpreta els diferents resultats provinents de les APIs i mostra un missatge de devolució si totes les cancel·lacions s'han efectuat. 		
Subfluxos	Si l'usuari decideix no cancel·lar la reserva, el sistema no esborrarà res ni retornarà l'import.		
Fluxos alternatius	<p>En el cas que l'identificador de la reserva sigui erroni, el sistema imprimirà per pantalla un missatge amb l'error corresponent.</p> <p>En el cas que les APIs no realitzin la cancel·lació, es mostrarà un missatge d'error.</p>		
Postcondició			
Requeriments no funcionals	REG-NF-1-06: Meta-buscador de vols REG-NF-1-07: API de cotxes REG-NF-1-08: Meta-buscador d'hotels		
Prioritat	Alta		
Comentaris			

Cas d'Ús	Gestió Equipatge		
Versió	1.1	Data	17/04/2020
Autors	Lucas, Joan		
Descripció	L'usuari decideix gestionar el seu equipatge, tenint en compte els vols seleccionats prèviament.		
Actors	Usuari, SkyScanner		
Precondició	L'usuari ha d'haver seleccionat prèviament algun vol.		
Flux Principal	1. L'Usuari selecciona l'opció gestionar equipatge. 2. Es mostren els vols seleccionats i l'opció de gestionar l'equipatge per cadascun dels vols. 3. S'actualitza el preu total del viatge tenint en compte l'augment que hagi suposat la gestió de l'equipatge.		
Subfluxos	Una vegada es mostren els vols seleccionats, es dona l'opció d'afegir l'equipatge directament a tots els vols.		
Fluxos alternatius			
Postcondició	S'actualitza el preu total de la reserva.		
Requeriments no funcionals	REG-NF-1-06 : Meta-buscador de vols		
Prioritat	Alta		
Comentaris			

Cas d'Ús	Ofertes Pròximes		
Versió	1.2	Data	18/04/2020
Autors	Lucas, Joan		
Descripció	El sistema mostrarà les ofertes trobades pels pròxims dies. Ho farà consultant els vols i els hotels utilitzant les diferents APIs.		
Actors	Usuari, SkyScanner, Booking		
Precondició	En el cas que l'usuari hagi realitzat alguna cerca d'algun viatge prèviament, tindrem en compte les seves dades per tal de mostrar-li unes ofertes adaptades.		
Flux Principal	1. El sistema es posa en contacte amb les APIs de vols i hotels. 2. Es mostren a l'Usuari les diferents ofertes de manera ascendent per preu. 3. L'usuari pot seleccionar una oferta i s'afegeix a la reserva del viatge.		
Subfluxos			
Fluxos alternatius	En el cas que l'Usuari no seleccioni cap oferta no caldrà realitzar cap acció.		
Postcondició			
Requeriments no funcionals	REG-NF-1-04: Ordenació REG-NF-1-06: Meta-buscador de vols REG-NF-1-08: Meta-buscador d'hotels REG-NF-1-09: Vista de quadricula REG-NF-1-10: Ofertes recents		
Prioritat	Alta		
Comentaris			

4. Diagrama de classes

Introducció

Els diagrames de classes són aquells que s'encarreguen de representar un conjunt de classes, interfícies i col·laboracions, així com les seves relacions. Per tant d'entendre-ho hem de saber que una classe és la descripció d'un grup d'objectes amb propietats comunes (atributs), comportament comú (operacions), relacions comunes amb altres objectes i semàntica comuna.

Centrant-nos en el nostre diagrama podem veure que l'hem agrupat en tres mòduls, artefactesAirHopping, interfícies i APIs.

En el primer d'ells trobem les classes Reserva, Trajecte, Control, Algoritmes, Vol, Hotel, Cotxe, Equipatge que gestionaran i guardaran tota la informació relacionada amb elles.

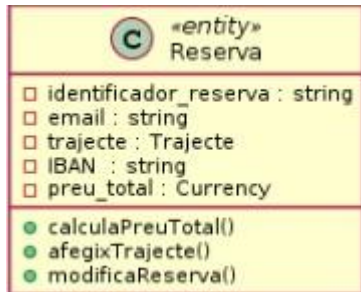
En el segon hem trobat adient afegir dues classes de tipus interfície, menuWeb i InterfReserva.

Per últim, per tenir diferents mòduls que gestionen el tràfic amb APIs externes, hem decidit crear quatre classes, PlataformaPagament, API_RentalCars, API_Booking, API_SkyScanner, i finalment una classe generalitzadora de les tres últimes, anomenada GestorAPIs.

A continuació explicarem amb detall les diferents classes anomenades, amb el corresponent diagrama.

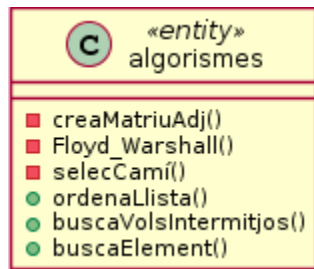
Classes

artefactesAirHopping



Reserva

La classe Reserva conté el conjunt d'atributs dels quals ha de disposar una reserva. Consta d'un identificador, el correu del client, el trajecte que aquest ha creat anteriorment, el preu total del viatge i finalment un IBAN. Pel que fa als mètodes, disposa de diferents funcions per calcular el preu total de la reserva a partir del trajecte "calculaPreuTotal()", així com per afegir el trajecte a la reserva final "afegeixTrajecte()" o modificar aquesta directament, "modificaReserva()".



Algorismes

Algorismes conté un conjunt de mètodes per tal de gestionar l'ordenació de les dades que arribin al sistema. Trobem tres mètodes interns de la classe, creaMatriuAdj(), FloydWarshall() i selecCami(), i tres mètodes ordenaLlista(), buscaVolsIntermitjos(), buscaElement().



Trajecte

La classe Trajecte s'encarrega d'emmagatzemar les dades dels vols, hotels i cotxes seleccionats per l'usuari així com d'afegir-ne o eliminar-ne algun. Consta de tres atributs (vols[], hotels[], cotxes[]), i les funcions afegeix(), i elimina() per cadascun dels camps (vols, hotels i cotxes). Finalment una funció modificaTrajecte() per tal de modificar internament la reserva.



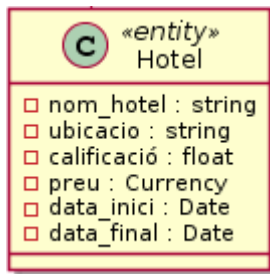
GestorReserva

GestorReserva és la classe encarregada de coordinar tots els artefactes d'AirHopping. És l'intermediari entre les interfícies, les APIs i les entitats. Conte diferents funcions per cobrir les necessitats de generar noves reserves amb els corresponents vols, cotxes i hotels, aplicar ofertes, pagar la reserva o cancel·lar-la, així com iniciar processos del sistema o enviar informació que pugui ser requerida per l'usuari.



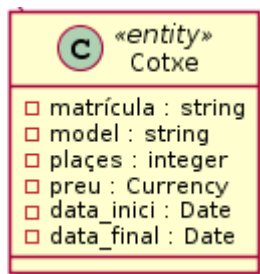
Vol

La classe Vol conté tots els atributs necessaris per gestionar les dades dels vols, on trobem una ID de vol, un origen i el seu destí, una data de sortida, el tipus d'avió, la companyia, el preu i l'equipatge. També disposa d'una funció `afegeixEquipatge()` per tal d'afegir l'equipatge i `modificaEquipatge()` per tal de modificar-ho.



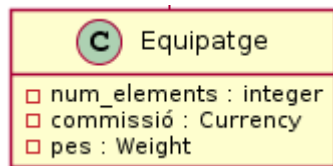
Hotel

Hotel és un compendi d'atributs necessaris per gestionar les dades dels hotels, on trobem el nom de l'hotel, la ubicació d'aquest, així com la qualificació, el preu per nit i les dates d'inici i final de l'estança.



Cotxe

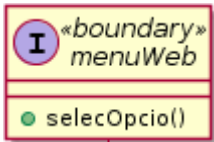
L'entitat Cotxe conté els atributs necessaris per gestionar les dades dels cotxes llogats pel client, on trobem la matrícula, el model, el nombre de places del vehicle, el seu preu i les dates d'inici i final del lloguer.



Equipatge

Equipatge és la classe encarregada de gestionar les dades de l'equipatge. Aquí trobem el nombre de paquets del client, la comissió a pagar per ells i el pes total de l'equipatge.

Interfícies



MenuWeb

MenuWeb s'encarrega de gestionar la interfície del menú de la web, és a dir, tots els menús als quals pot accedir l'usuari. Conte la funció `selecOpcio()` que actua com a senyal per iniciar diferents processos en tot el sistema.

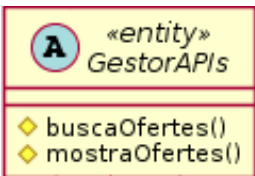


InterfReserva

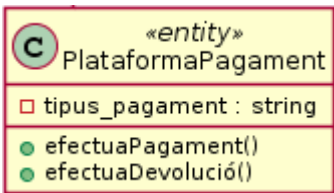
InterfReserva conté aquells mètodes que fan referència a les interaccions amb la interfície de reserves i el mostratge de missatges i dades del sistema. Aquí trobem diferents funcions per a la impressió per pantalla de diferents atributs i dades de tota la reserva, així com mètodes per entrar dades i seleccionar dades específiques mostrades prèviament a l'usuari, i finalment la mostra de missatges d'error o resultat satisfactori per les operacions del sistema.

APIS

GestorAPIs

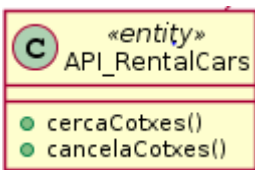


La classe Gestor APIs és una generalització de les APIs de SkyScanner, Booking i Rentalcars, que genera flexibilitat en el sistema agrupant-les en una entitat que les gestiona. Conté dues funcions que permetran la busca d'ofertes en les diferents APIs "buscaOfertes()" i la mostra d'aquestes ofertes a l'usuari "mostraOfertes()".



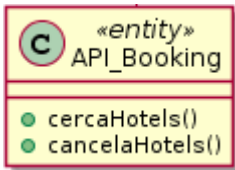
PlataformaPagament

La plataforma de pagament conté les funcions necessàries per efectuar els pagaments i les devolucions de les reserves. Aquesta classe es posarà en contacte amb les APIs de pagament (Visa o Mastercard) en funció del tipus de pagament seleccionat pel client i farà l'operació demanada pel client.

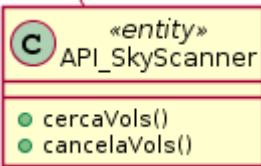


API_RentalCars

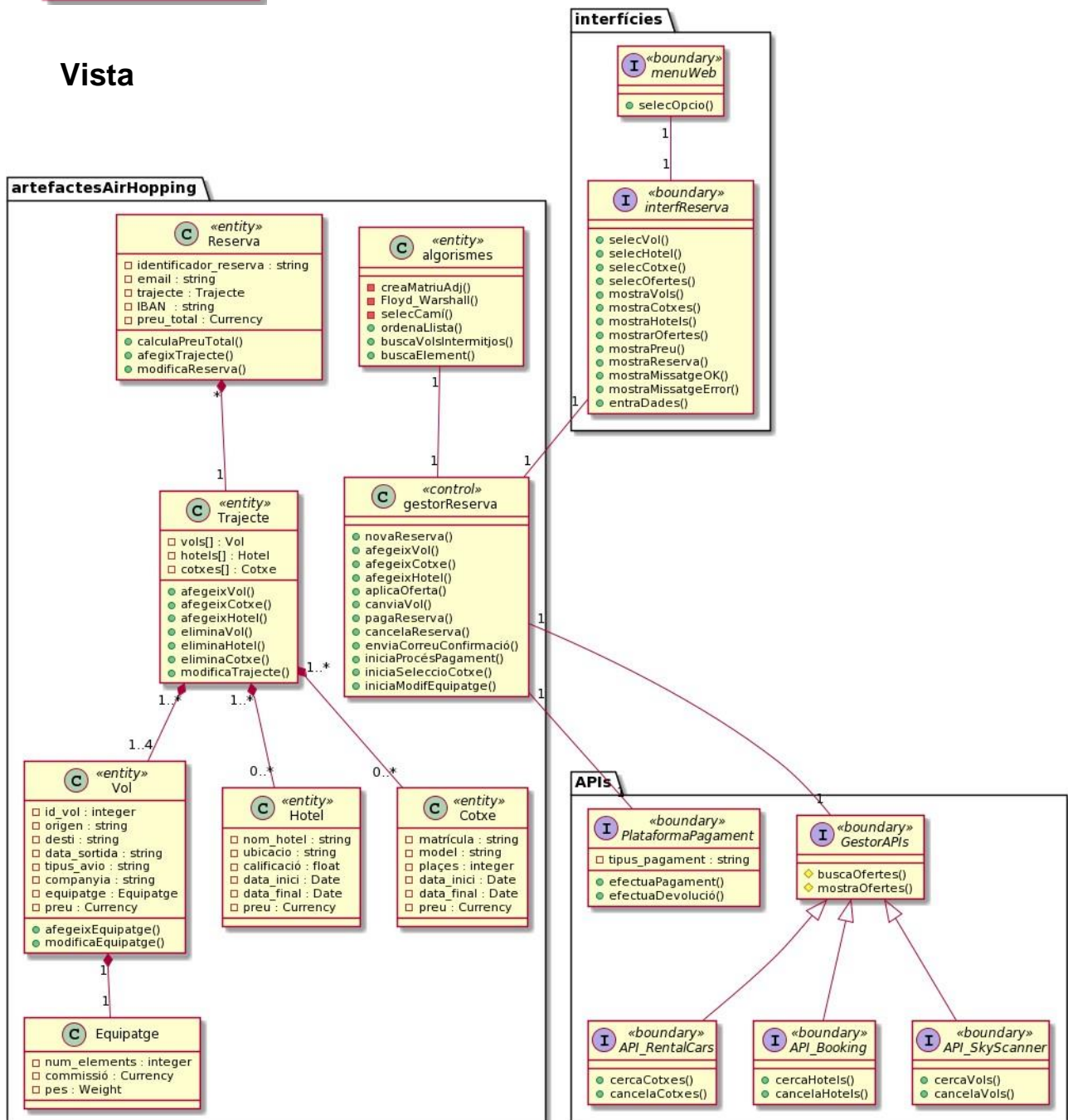
API_RentalCars conté dues funcions per a la cerca dels vehicles disponibles i la cancel·lació de la reserva del vehicle.

**API_Booking**

API_Booking conté dues funcions, la primera, “cercaHotels()” que s’encarrega de la cerca dels hotels disponibles i “cancelaHotels()” que permet cancel·lar la selecció d’un hotel en cas de cancel·lar la reserva.

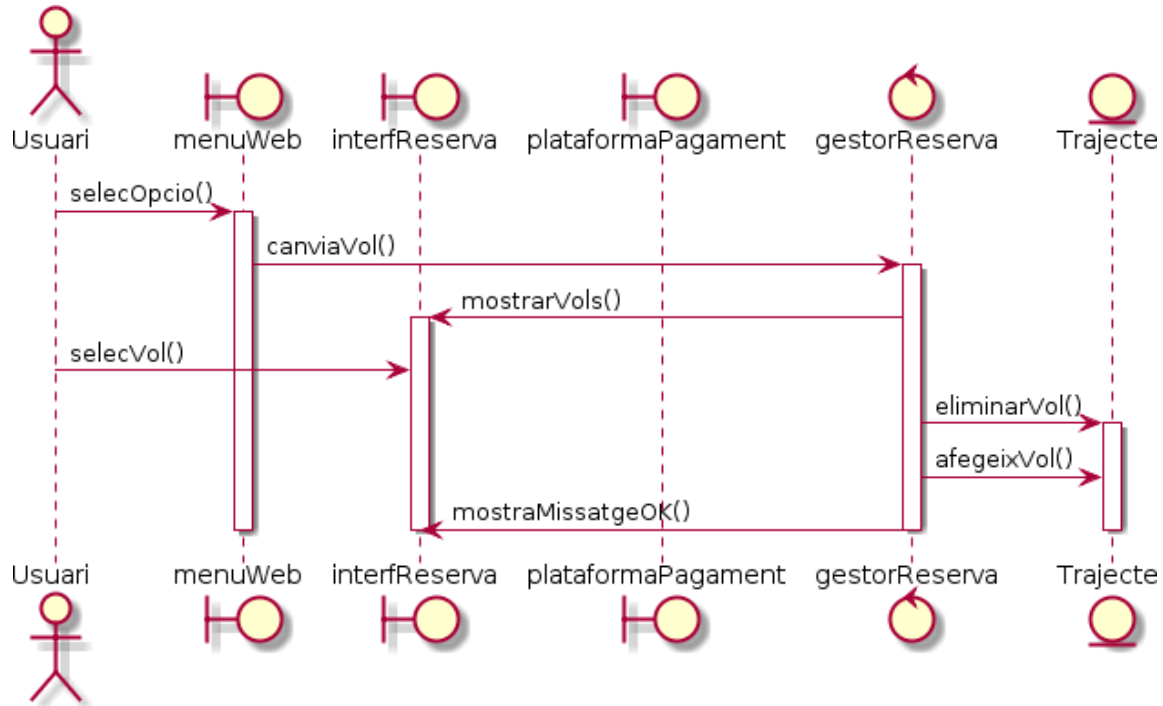
**API_SkyScanner**

API_SkyScanner s’encarrega de les funcions per cercar vols disponibles i cancel·lar vols ja seleccionats i reservats.

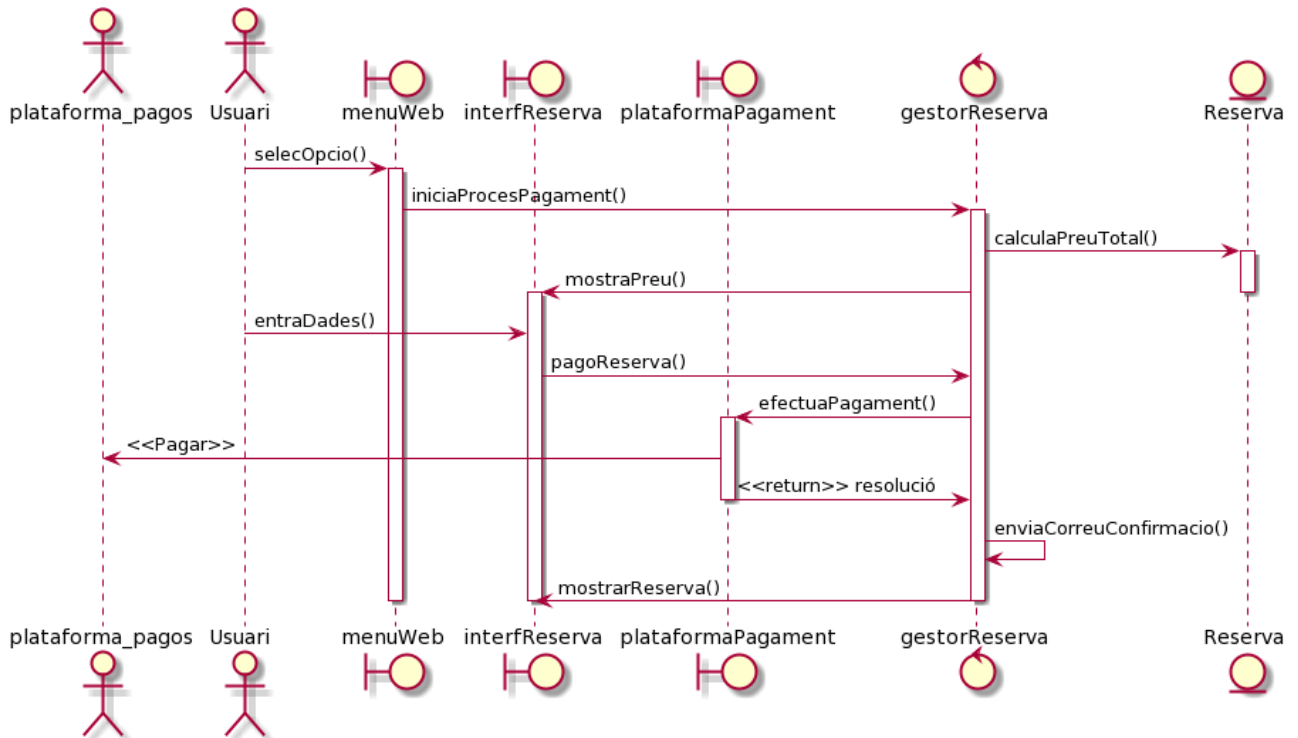
Vista

32

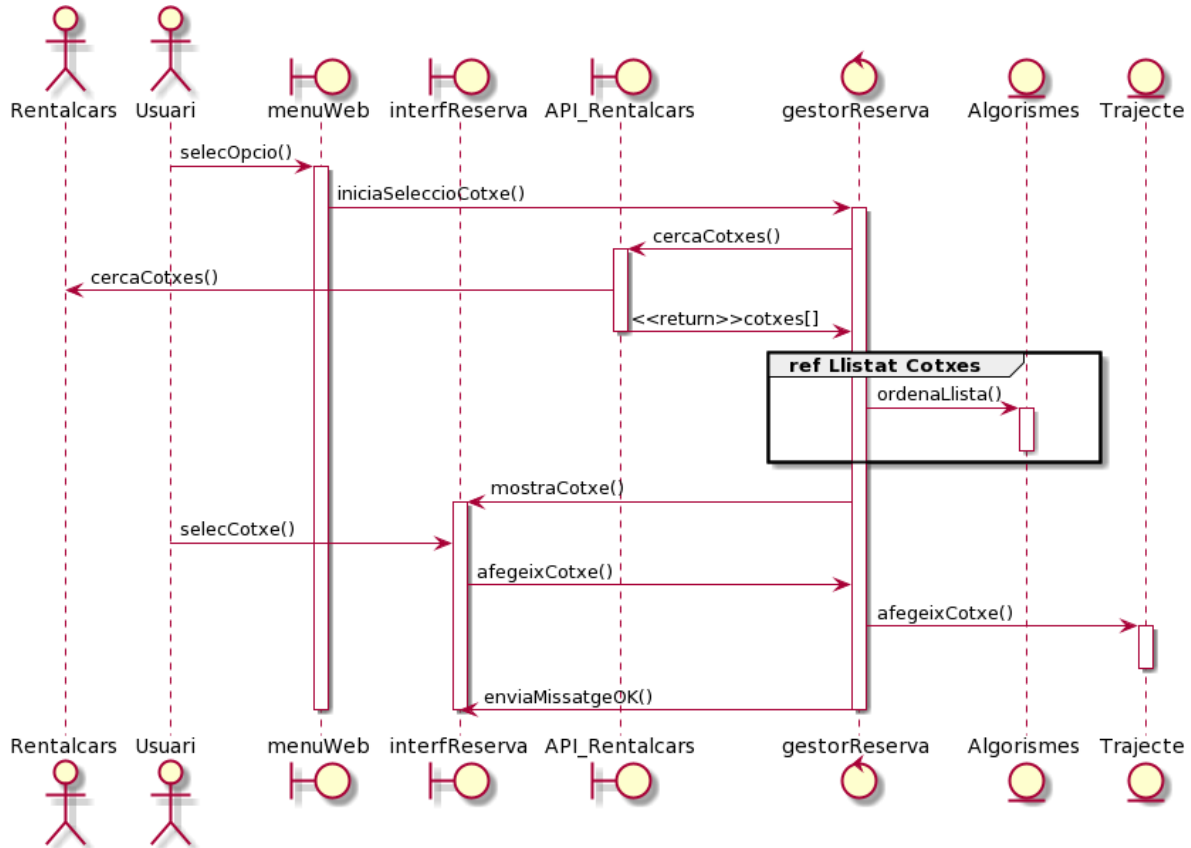
- Canvi Vol



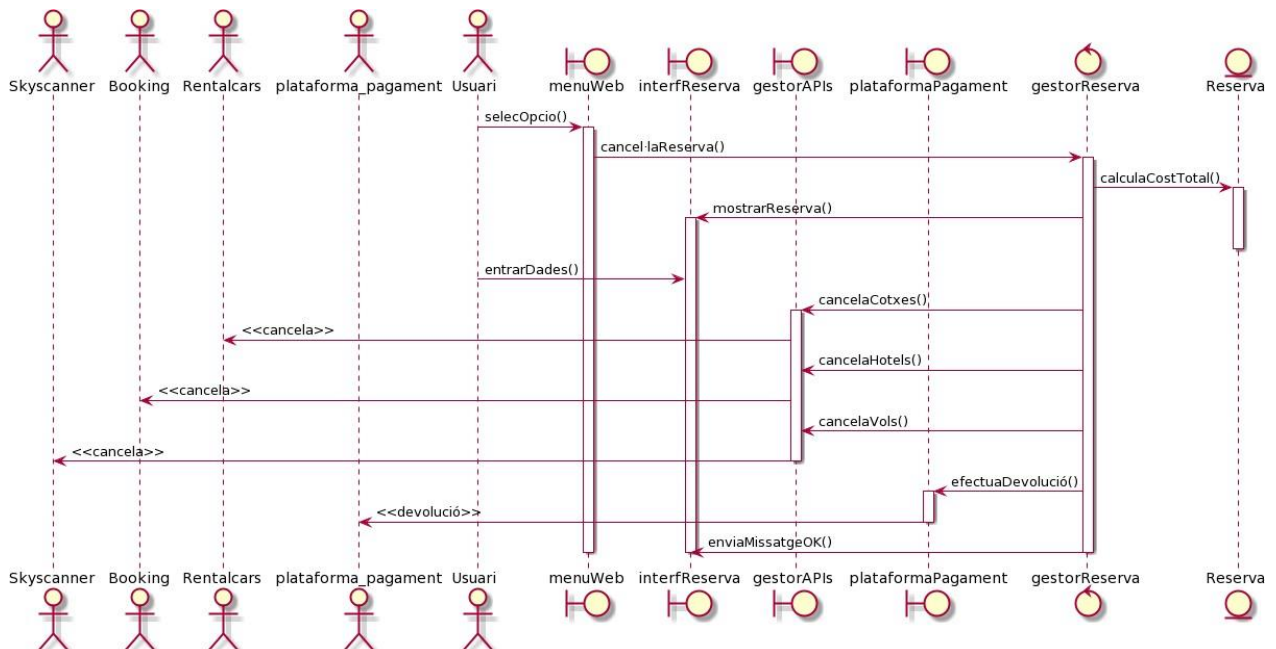
- Pagament de Viatge



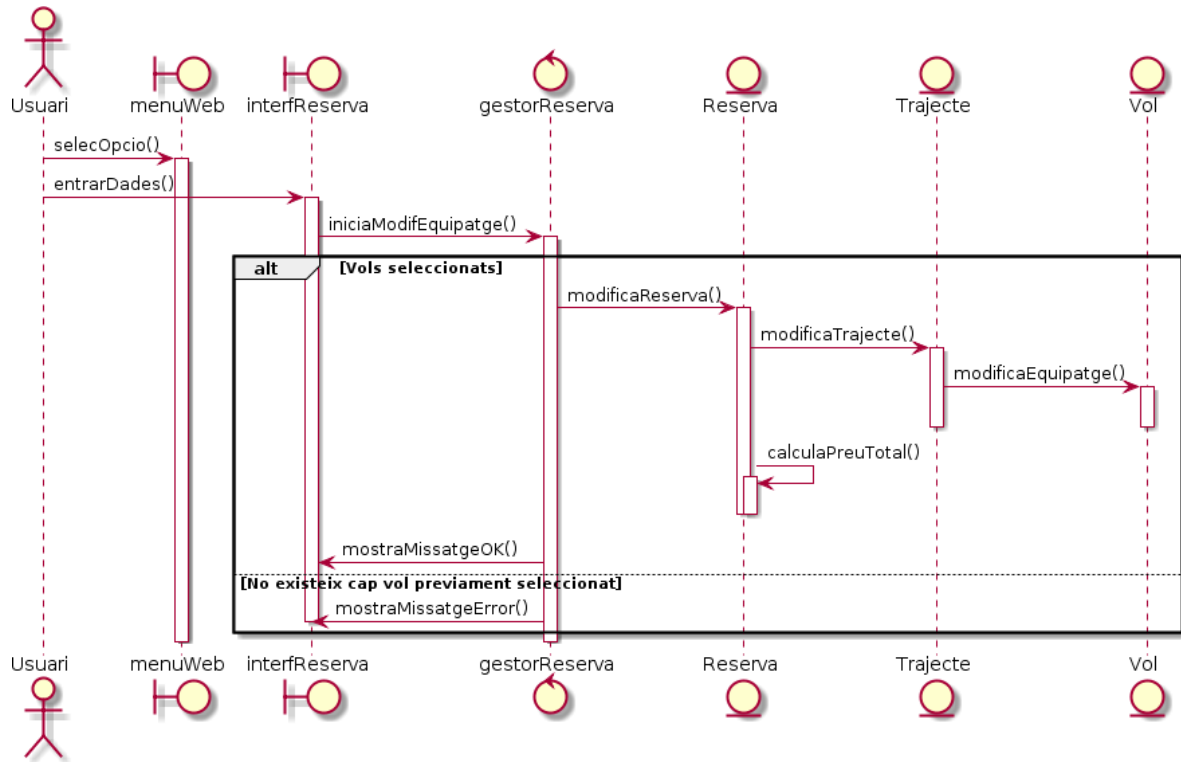
• Llistat Cotxes i Selecció Cotxes



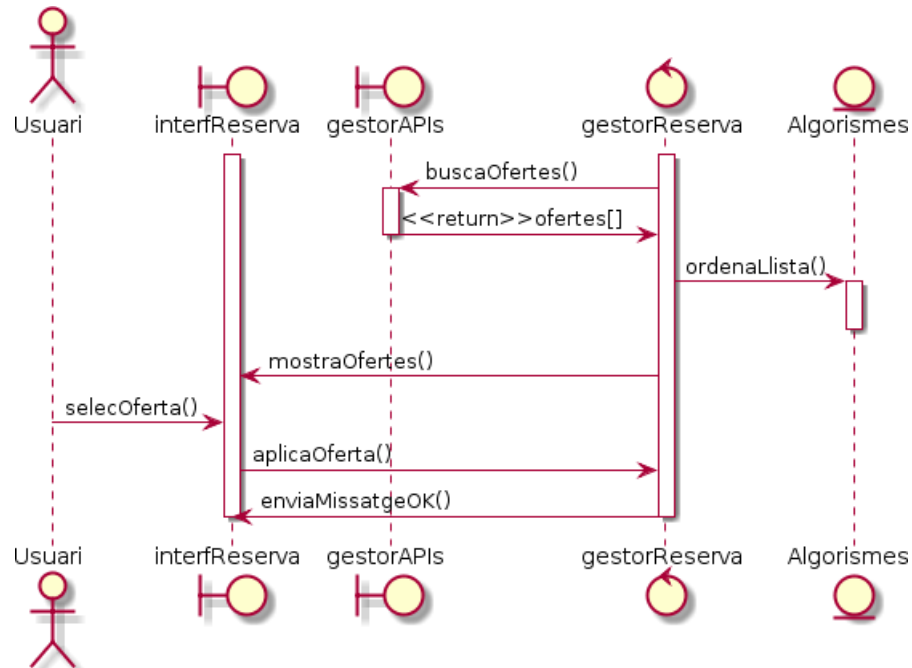
• Cancelació reserva



- Gestió equipatge



- Busca Ofertes



6. Test unitaris

Repositori

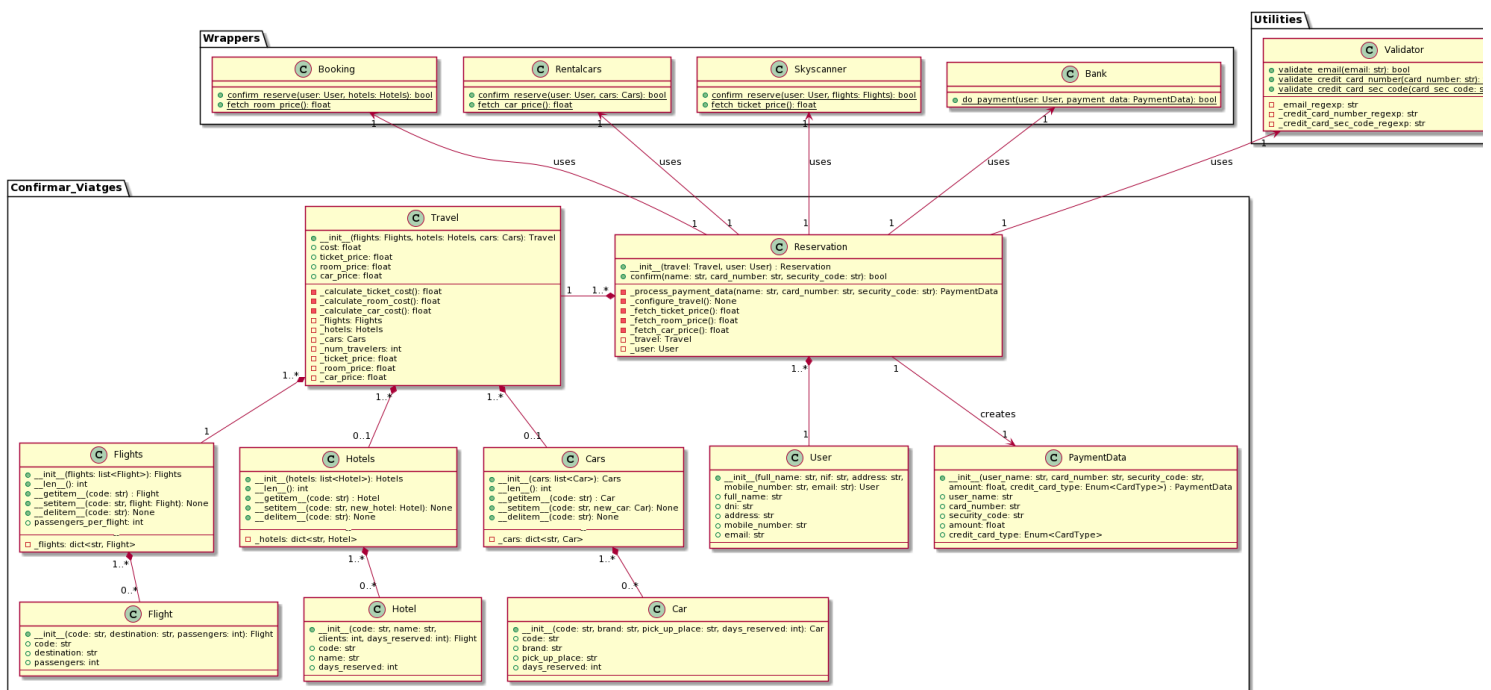
Per tal de facilitar el treball en grup hem creat un repositori a la plataforma GitHub. Aquesta plataforma ens ha permès treballar alhora als diferents membres del grup. A través de “commits” i “push” hem anat actualitzant-lo a mesura que anàvem treballant.

Així mateix hem utilitzat el programari Git per manipular els diferents arxius de manera còmoda i poder realitzar els canvis a GitHub.

Enllaç al repositori: <https://github.com/JoanRosell/ES-2020-412-06>

Diagrama de classes

A partir dels coneixements assolits a l'assignatura sobre disseny i el llenguatge de modelat UML hem realitzat el següent diagrama de classes:



(al repositori es pot veure limatge amb millor qualitat)

Al diagrama de classes hem llistat les classes implementades per nosaltres a més a més de les que ja ens venien fixades. Tal com es pot apreciar, aquestes últimes classes les hem estès per fer el treball de testeig més senzill. El nostre disseny ha intentat seguir els principis SOLID i la filosofia KISS (Keep It Simple, Stupid), utilitzant les mínimes classes possibles que garanteixin la modularitat i flexibilitat que caracteritzen un bon disseny orientat a objectes.

Els objectes que componen Travel són abstraccions dels llistats de vols, hotels i cotxes mencionats a l'anunciat. Hem decidit emprar diccionaris, ja que cada objecte concret té un codi únic, fent evident l'idoneïtat d'aquest tipus d'estructures de dades. Reservation empra Travel per accedir al cost de tot el viatge, configurant el viatge segons els paràmetres que rep de les diferents APIs. La lògica del càlcul del preu es troba a Travel, ja que trobem més semàntic definir el cost d'un viatge com una propietat del mateix, i per aquest cas d'ús Python ens ofereix el decorator property. Aquest constructe també és utilitzat per definir els diferents preus com a propietats, afegint més seguretat a les interaccions entre Reservation i Travel.

Reservation és la classe més important i és amb qui es comunica la Interfície d'usuari. Hem de reconèixer que aquesta classe ha acabat fent massa coses i que el disseny és més que millorable, podent refactoritzar part de la funcionalitat de connexió amb les APIs en una interfície especialitzada. Tanmateix, hem optat per fer una aproximació més centralitzada, ja que aquesta última part de la pràctica tenia un conjunt finit de funcionalitats i no era necessari fer un disseny 'perfecte', que maximitzes l'encapsulació i minimitzés l'acoblament.

En última instància tenim la classe Validator, que al principi anava a ser un singleton i al final vàrem decidir que podríem aprofitar la funcionalitat de les classes estàtiques de Python, convertint-lo en un conjunt de mètodes estàtics agrupats sota un únic 'namespace'. Aquesta classe és emprada per Reservation per validar de forma dèbil les dades de l'usuari. Diem validació dèbil, ja que per validar correctament un correu electrònic hauríem d'utilitzar algun servidor SMTP per verificar l'existència del correu, però aquesta funcionalitat quedava fora dels objectius de la pràctica.

Implementació en codi

Phyton 3.7.x

S'ha establert un conveni entre els membres de l'equip per treballar en la versió de python 3.7 amb entorns virtuals que mantenen una estabilitat fiable a l'hora de treballar entre diferents integrants.

IDE PyCharm

Per tal d'implementar el codi hem utilitzat l'IDE PyCharm, un compilador principalment preparat pel llenguatge Python que a més inclou compatibilitat amb Git i Github, facilitant els canvis en el repositori i permetent portar un historial dels diferents "comits" realitzats.

Metodologia

La metodologia que hem utilitzat per a la implementació del codi ha estat la TDD (Test Driven

Development). Aquesta metodologia de disseny de software és una pràctica que consisteix a realitzar primerament els test d'una funcionalitat, i una vegada fallits, s'ha d'implementar codi per tal d'assolir aquesta funcionalitat de manera correcta.

Centrant-nos en el nostre projecte, al seguir la metodologia TDD per fer el desenvolupament ens hem centrat a fer objectes molt modulars amb gran quantitat de petits mètodes que millorin la testabilitat. Hem aconseguit un test coverage del 94% segons l'utilitat pytest-cov, fet que demostra que el nostre disseny és fàcilment testable. Al tenir petits mètodes els hem pogut testar per separat i en combinació amb d'altres, simplificant l'escriptura dels test cases.

7. Scrum

Introducció

Scrum és un mètode per a treballar en equip a partir d'iteracions o Sprints. Així doncs, Scrum és una metodologia àgil que tindrà com objectiu controlar i planificar projectes. En el nostre cas utilitzarem Jira per realitzar les cinc versions del cas d'us "pagar_viatge" del programa "Air hopping", on haurem de gestionar el pagament tenint en compte les reserves de vols, allotjaments y vehicles (veure flux del programa en l'enunciat de la pràctica).

Product Backlog

El product backlog és una llista ordenada de tot el que podria ser necessari per la realització del projecte. En aquesta llista s'han d'enumerar totes les característiques, funcionalitats, requisits i millores que ha d'assolir el producte final. Per fer-ho cada element d'aquesta llista conté atributs com la descripció, l'estimació de temps i la versió a la que afecta.

Pel que fa el nostre projecte hem organitzat el nostre "Product Backlog" en diferents "Stories", on cadascuna conté les diferents subtasques necessàries per realitzar-les. A continuació podem veure la llista:

Versió 1

Gestionar el nombre de viatgers :

Descripció

Amb aquesta funcionalitat, l'usuari podrà introduir les dades demanades pel sistema. Per tant s'haurà d'implementar la classe "Flight" i "Flights" i els test necessaris per a la classe "User". També per considerar el nombre de viatgers haurem d'afegir un mètode que ens permeti obtenir el número de clients en la classe "Travel".

Sub-tasks

Implementar classe "Flight"

Crear classe "Flights"

Refactoritzar unit tests classe "Flights"

Afegir tests per la classe "User"

Afegir mètode "get_num_clients" a "Travel"

Gestionar diversos destins :

Descripció

Aquesta funcionalitat serà l'encarregada de rebre, analitzar i validar les diferents destinacions escollides per l'usuari. S'hauran d'afegir els mètodes necessaris per afegir o eliminar vols tant a la classe "Flights" com "Travel" i "Reservation".

Sub-tasks

Afegir el mètode add_flight per la classe "Flights"
Afegir mètode delete_flight per la classe "Flights"
Afegir delete_flight per la classe "Travel"
Afegir mètode delete_flight per la classe "Reservation".

Calcular el preu total del viatge tenint en compte els vols seleccionats i el nombre de viatgers :

Descripció

Es calcularà el preu final en funció de diversos paràmetres i configuracions. S'haurà d'implementar la classe "PaymentData" i els mètodes necessaris per modificar els preus dels vols.

Sub-tasks

Acumular els preus en funció del numero de passatgers
Implementar la classe "PaymentData"
Afegir tests per la classe "Reservation"
Acumular els preus en funció del nombre de vols
Afegir mètode "add_flight" per la classe Travel
Ampliar els test de "calculate_flight_price" quan s'afegeixen els vols
Ampliar els test de "calculate_flight_price" quan s'eliminen vols

Confirmar la reserva dels vols seleccionats :

Descripció

Aquesta funció realitza la confirmació de la reserva dels vols seleccionats sense tenir en compte cap altre paràmetre. S'haurà de crear un mètode "confirm" i un altre per processar el pagament.

Sub-tasks

Crear mètode "confirm" per la classe "Reservation"
Crear mètode "process_payment"

Realitzar el pagament d'un viatge:

Descripció

S'hauran d'afegir els test necessaris per simular el pagament d'un viatge.

Sub-tasks

Afegir test per simular un pagament correcte.

Versió 2

Gestionar el mètode de pagament seleccionat per realitzar el pagament:

Descripció

Una vegada seleccionada la metodologia de pagament, la funcionalitat a desenvolupar és contactar amb les APIs de pagament i enviar la informació que l'usuari ha proporcionat per tal de realitzar el pagament. S'haurà de crear un test per comprovar que s'ha seleccionat el pagament correcte, també s'haurà d'afegir la variable "credit_card_type" en la classe "PaymentData".

Sub-tasks

Crear Test pagament seleccionat correcte.

Implementar variable "credit_card_type" en la classe "PaymentData".

Gestionar errors en la realització del pagament del viatge

Descripció

S'hauran de crear els test necessaris per comprovar el missatge d'error al pagament i les pertinents modificacions a la classe "Reservation".

Sub-tasks

Crear test comprovació missatge d'error de pagament

Modificar codi classe "Reservation" per implementar missatge d'error de pagament

Gestionar errors en la confirmació de la reserva dels vols

Descripció

S'hauran de crear els test necessaris per comprovar el missatge d'error de la reserva dels vols i les pertinents modificacions a la classe "Reservation".

Sub-tasks

Crear test comprovació de missatge d'error de reserva.

Modificar codi de la classe "Reservation" per implementar el missatge d'error a la reserva.

L'aplicació haurà de validar les dades de l'usuari

Descripció

Per tal de validar les dades de l'usuari haurem d'afegir la classe "Validator" i el conjunt de mètodes necessaris per validar les dades que considerem necessàries.

Sub-tasks

Crear classe "Validator"

Crear mètode `validate_email` de la classe "Validator".

Crear mètode `validate_dni` de la classe "Validator".

Crear mètode `validate_card_number` de la classe "Validator".

Crear mètode `validate_card_security_code` de la classe "Validator".

Validar dades de facturació de l'usuari.

Validar mètode de pagament.

Validar dades de pagament.

Refactoritzar el càlcul del preu total

Descripció

S'hauran de modificar els mètodes necessaris per tal de no utilitzar les funcions de `calculate_*_price` de la classe "Reservation".

Sub-tasks

Eliminar funcions de `calculate_*_price` de "Reservation".

Modificar el mètode `_process_payment_data` de la classe "Reservation".

Implementar mètodes `_fetch_*_price` en "Reservation".

Implementar mètodes `_fetch_*_price` en els diferents "Wrappers".

Implementar propietat `Travel.cost`

Afegir propietats `*_price` a "Travel"

Versió 3

Gestionar vehicles

Descripció

S'han de gestionar el conjunt de funcions necessàries per afegir o eliminar vehicles de la reserva. Haurem de crear els test necessaris per comprovar la informació dels cotxes i crear la classe "Car" i traslladar les variables de "Cars" a "Car"

Sub-tasks

Crear test de comprovació de la informació dels cotxes.
Crear Classe Car / Traslladar variables "Cars" a "Car".

Gestionar allotjaments

Descripció

S'han de gestionar el conjunt de funcions necessàries per afegir o eliminar allotjaments de la reserva. Haurem de crear els test necessaris per comprovar la informació dels allotjaments i crear la classe "Hotel" i traslladar les variables de "Hotels" a "Hotel"

Sub-tasks

Crear test de comprovació de la informació dels Hotels.
Crear Classe Car / Traslladar variables "Hotels" a "Hotel".

Confirmar la reserva dels vehicles seleccionats considerant possibles errors

Descripció

Considerar possibles errors en la selecció dels vehicles. S'hauran de crear els test de comprovació de missatge d'error dels vehicles, també haurem de realitzar els canvis necessaris a la classe "Reservation".

Sub-tasks

Crear Tests de comprovació de missatge d'error dels vehicles.
Modificar codi de la classe "Reservation" per implementar el missatge d'error dels vehicles.

Confirmar la reserva dels allotjaments seleccionats considerant possibles errors

Descripció

Considerar possibles errors en la selecció dels allotjaments. S'hauran de crear els test de comprovació de missatge d'error dels allotjaments, també haurem de realitzar els canvis necessaris a la classe "Reservation".

Sub-tasks

Crear Tests de comprovació de missatge d'error dels allotjaments.

Modificar codi de la classe "Reservation" per implementar el missatge d'error dels allotjaments.

Calcular el preu total del viatge tenint en compte els vols, els vehicles, els allotjaments y el número de viatgers

Descripció

Calcular el preu total del viatge tenint en compte els vols, els vehicles, els allotjaments i el nombre de viatgers.

Sub-tasks

Implementar càlcul del preu de les habitacions del hotel.

Implementar càlcul del preu del lloguer de cotxes.

Versió 4

Gestionar reintents quan es produeix error al realitzar el pagament

Descripció

En el moment que es produeix un error en el pagament haurem de gestionar els reintents. S'hauran de crear els test necessaris per comprovar els reintents del pagament.

Sub-tasks

Implementació de reintents de la funció "confirm" pel pagament

Crear test reintent del pagament.

Gestionar reintents quan es produeix error al confirmar la reserva dels vols

Descripció

En el moment que es produeix un error en la confirmació de la reserva dels vols haurem de gestionar els reintents. . S'hauran de crear els test necessaris per comprovar els reintents de la reserva dels vols.

Sub-tasks

Implementació de reintents de la funció “confirm” pels vols.

Crear test reintents dels vols .

Versió 5

Gestionar dades de facturació de l'usuari que realitza la reserva considerant possibles errors

Descripció

Gestionar dades de facturació de l'usuari que realitza la reserva considerant possibles errors. S'hauran de crear els test de comprovació de missatge d'error de les dades de la facturació i de les dades de l'usuari.

Sub-tasks

Crear test de comprovació de missatge d'error de les dades de la facturació

Test de comprovació de les dades de l'usuari de la facturació.

Gestionar reintents quan es produeix error al confirmar la reserva de vehicles y allotjaments

Descripció

Gestionar reintents quan es produeix error al confirmar la reserva de vehicles y allotjaments. S'hauran de crear els test de reintents i la implementació d'aquests a la classe confirm.

Sub-tasks

Implementació de reintents en la funció “confirm” pels vehicles i l'allotjament.

Crear test de reintents de confirmació de reserva de vehicles i allotjament.

Generals

Aplicar convencions de noms a la part privada de les classes

Descripció

Generalitzar els noms de les parts privades de les classes per tal de mantenir l'estil de programació.

Redissenyar les interaccions entre les classes

Descripció

Afegir mètodes per gestionar els ítems de la classe Flight (`__getitem__`, `__setitem__`, `__delitem__`,
i `__len__`)

Sub-tasks

Afegir mètodes `__*item__` a la classe “Flights”

Documentació

Descripció

Treball de documentació del codi, referent a les llibreries utilitzades i la descripció de cadascuna de les classes i tests implementats.

Sub-tasks

Buscar una llibreria per generar la documentació del projecte.

Afegir docstrings a la classe “Reservation”

Afegir docstrings a la classe “Travel”

Afegir docstrings als test de la classe “Reservation”

Millorar la descripció dels test.

Afegir la documentació als test fixtures.

Sprint Backlog

Introducció

L'sprint backlog és el subconjunt d'objectius seleccionats del product backlog per la iteració actual (Sprint). Per tant l'equip s'ha de reunir per planificar-ho i estimar totes les tasques que s'han de realitzar en aquest sprint.

Sprint 1 Backlog

Per la realització del nostre primer Sprint vam decidir seleccionar les tasques corresponents a la versió 1 del product backlog. Vam considerar de fer aquest Sprint d'un volum de treball més petit respecte al segon.

Summary

Gestionar el número de viajeros

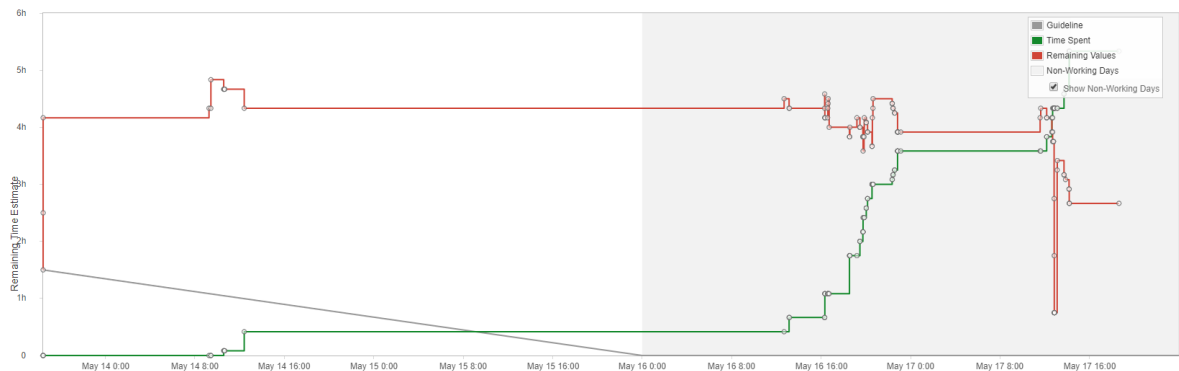
Gestionar varios destinos

Calcular el precio total del viaje teniendo en cuenta los vuelos seleccionados y el número de viajeros

Confirmar la reserva de los vuelos seleccionados

Realizar el pago de un viaje (sin considerar errores y sin necesidad de seleccionar el método de pago)

Burdown Chart



Interpretació : Inicialment vam començar el sprint sense afegir totes les tasques necessàries. Tampoc era correcte el temps estimat per cadascuna d'elles. Això provoca la pujada inicial de la línia vermella al gràfic.

Seguidament fixant-nos en la línia verda del gràfic podem comprovar que el temps empleat per part del grup durant el sprint ha estat desigual, es a dir de cara al final de l'sprint s'ha incrementat el treball realitzat considerablement.

Per tant el principal problema que hem tingut durant aquest primer sprint ha estat que inicialment no vam considerar que el temps estimat per fer cada sub-tasca es sumava amb el de les stories. En el proper sprint, aquest error es té en compte i no es repeteix.

Sprint 2 Backlog

Per la realització del segon Sprint hem decidit seleccionar les tasques corresponents a la segona, tercera, quarta i cinquena versió. Per tant el volum de treball corresponent va ser més gran que del primer sprint, com es pot comprovar al “Burndown Chart”.

Per tant l'objectiu d'aquest Sprint va ser acabar el total de tasques del product backlog i per tant finalitzar el projecte.

Gestionar el método de pago seleccionado para realizar el pago

Documentación

Gestionar errores en la realización del pago del viaje

Gestionar errores en la confirmación de la reserva de los vuelos

La aplicación deberá validar los datos del usuario

Refactorizar el cálculo del precio total

Gestionar vehículos

Gestionar alojamientos.

Confirmar la reserva de los vehículos seleccionados considerando posibles errores

Confirmar la reserva de los alojamientos seleccionados considerando posibles errores.

Calcular el precio total del viaje teniendo en cuenta los vuelos, los coches, los alojamientos y el número de viajeros.

Gestionar reintentos cuando se produce error al realizar el pago

Gestionar reintentos cuando se produce error al confirmar la reserva de vuelos

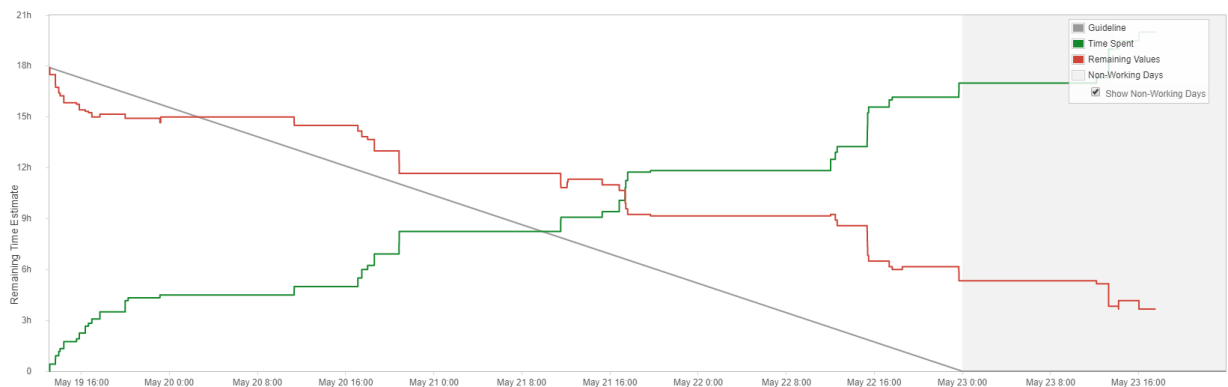
3 Aplicar convenciones de nombres a la parte privada de las clases

Gestionar datos de facturación del usuario que realiza la reserva considerando posibles errores

Gestionar reintentos cuando se produce error al confirmar la reserva de vehículos y alojamientos

2 Rediseñar interacciones entre clases

Burndown Chart



Interpretació : Com podem veure el volum de treball d'aquest segon sprint ha estat molt més gran que en l'anterior sprint. El temps estimat inicialment ha estat d'unes 18h. Aquesta previsió ha estat bastant ben encaminada, encara que em empleat unes dues hores més de les estimades (línea verda del gràfic, tapada per la llegenda). Com podem veure el treball s'ha realitzat d'una manera progressiva, tot i això ens ha faltat completar una tasca referent a buscar i utilitzar una llibreria per generar la documentació del projecte.

Com podem veure inicialment vam seguir bastant bé la “guideline”, encara que a mesura que passen els dies no hem aconseguit mantenir aquesta tendència. Un dels motius pot ser que en la “guideline” del Burndown Chart no es tenen en compte els caps de setmana. Si hagués estat així, el nostre treball hauria estat molt semblant a la línia marcada per la guideline.

Per tant com a conclusió podem dir que hauríem d'haver repartit millor el treball en els sprints per tal de que aquest segon hagués tingut un volum de treball menor. Pel que fa al treball realitzat en aquest segon sprint podem afirmar que estem satisfets, ja que hem pogut realitzar pràcticament tot el treball que havíem planejat a l'inici del mateix.