



Culture finder

MANAGERIAL REPORT

[Github](#) - [Taiga](#) - [Drive](#) - [Project Record Track](#)

Cognom	Nom	Responsable	UPC e-mail	Taiga	GDrive	Github
Morón	Daniel	Management services of	daniel.moron.roces@estudiantat.upc.edu	danielmr_6	daniel.moron.roces@estudiantat.upc.edu	danielmr6
Risueño	Iván	Inception	ivan.risueno@estudiantat.upc.edu	ivan.risueno	ivan.risueno@estudiantat.upc.edu	ivan-risueno
Rodriguez	Marc	Sprint 2	marc.rodriguez.martin@estudiantat.upc.edu	marcrd11	marc.rodriguez.martin@estudiantat.upc.edu	MarcRd11
Duch	Marc	Sprint 1	marc.duch@estudiantat.upc.edu	marc.duch	marc.duch@estudiantat.upc.edu	Marcarrones
Moreno	Miguel	Final demo and closing documentation	miguel.moreno.alcaraz@estudiantat.upc.edu	MiguelMorenoAlcaraz	miguel.moreno.alcaraz@estudiantat.upc.edu	MiguelMorenoAlcaraz
Delgado	Òscar	Sprint 3	oscar.delgado.gomez@estudiantat.upc.edu	oscard14	oscar.delgado.gomez@estudiantat.upc.edu	oscard147

Índex

1.1. Problemes a resoldre	3
1.2. Solució proposada	3
1.3. Proposta de valor	3
3.1. Conclusions generals	7
3.2. Conceptes aplicats adquirits en altres assignatures	7
4.1. Óscar Delgado:	8
4.2. Marc Duch:	8
4.3. Miguel Moreno:	10
4.4. Daniel Morón:	11
4.5. Iván Risueño:	12
4.6. Marc Rodriguez:	14

1. Resum executiu

Quan va arribar el moment d'escollir el tema de la nostra aplicació, el qual va ser l'Agenda Cultural, vam haver d'investigar i entrar en matèria sobre l'estat actual de sistemes existents i analitzar quins punts forts i febles tenien cadascun d'ells. A partir d'aquest estudi vam proposar un conjunt de solucions als problemes existents al pensar que faria la nostra aplicació.

1.1. Problemes a resoldre

El problema general que considerem és la dificultat per informar-se sobre events culturals a Catalunya, ja que la informació està escampada i poc organitzada.

Entrant més en detall, dividim els problemes en els següents:

- No hi ha molts llocs on veure el conjunt d'esdeveniments a Catalunya i informar-se.
- La informació acostuma a estar incompleta o desorganitzada.
- No poder personalitzar els events que et recomanen segons els teus interessos.
- No poder veure els events llistats d'una manera més visual, com a un mapa o calendari.
- No poder guardar informació d'esdeveniments que et resulten interessants.

1.2. Solució proposada

- Sistema per poder mostrar de manera clara i concisa els esdeveniments disponibles de Catalunya a partir de l'ubicació de l'usuari.
- Sistema recomanador d'esdeveniments a partir dels interessos de l'usuari.
- Sistema per visualitzar els esdeveniments més populars de manera clara a partir d'un rang de dates.
- Poder visualitzar els esdeveniments en diferents formats com llista, mapa o calendari a partir dels filtres indicats.
- Tenir la possibilitat de guardar un conjunt d'esdeveniments i la seva informació creant llistes pròpies. Un bon exemple seria la llista de favorits.

1.3. Proposta de valor

- L'abast del nostre sistema es pot ampliar fàcilment per a altres localitzacions apart de Catalunya sempre que tinguem accés a la base de dades del lloc corresponent.

- Com l'usuari client pot veure esdeveniments de manera personalitzada, la qual cosa ens dona un valor extra com a producte, ja que el client estarà més interessat en les activitats de l'aplicació.
- Un altre punt important seria la possibilitat de reportar incidències relacionades amb problemes sobre l'aplicació i els esdeveniments d'aquesta.

2. Dificultats trobades

La primera dificultat que vam tenir va ser la corba d'aprenentatge i el temps que aquesta ens va consumir. Tant els membres de l'equip encarregats del frontend com els encarregats del backend vam tenir que aprendre les tecnologies utilitzades (flutter per front, springboot per back) gairebé des de zero. Al primer sprint, vam subestimar el temps que trigariem en familiaritzar-nos amb els entorns i vam acabar fent bastantes menys tasques de les que en un principi estava planificat. Aquest problema el vam arrastrar durant la resta del projecte en forma de tasques addicionals a cada sprint, les que no s'havien pogut completar a l'anterior, i ens va forçar a desviar-nos del pla inicial de totes les funcionalitats per la nostra aplicació.

De similar manera, coordinar un equip de sis persones, on tots estem fent diverses assignatures simultàniament i uns pocs tenen pràctiques, ha estat difícil en alguns moments, sobre tot quan coincidien exàmens o entregues d'altres assignatures. Tot i aixó, la comunicació ha estat bona i aquest no ha sigut un dels problemes principals.

Durant la realització del projecte, ens vam adonar de que el servidor de Virtech, de la FIB, deixa de funcionar a les 12 de la nit i torna a activar-se a les 8 de la matinada. Això va suposar un problema per a l'equip de frontend, ja que Virtech es el que permet que les crides a la API del backend funcionin i durant aquestes hores no es podien provar les funcionalitats que fessin crides. Tot i a no ser un problema gran ja que la majoria de treball es realitzava durant el dia, en moments puntuals on es va voler aprofitar temps per la nit no es va poder avançar gaire. Aquests casos puntuals es devien sovint a la dificultat d'horaris deguts a altres assignatures o a pràctiques.

Una de les desviacions importants que vam decidir entre el final del primer sprint i el principi del segon va ser la de no fer testing amb el model TDD. Degut a la falta de temps que vam anar acumulant per les diverses raons explicades a dalt, vam decidir fer el testing manualment. En l'equip de frontend, el nostre testing consistia en comprovar que certes dades s'obtenien imprimint-les en consola i provant a l'emulador d'Android Studio, l'entorn que hem utilitzat, que les vistes i funcionalitats sortissin correctament a l'aplicació.

En l'equip de backend, el testing consistia en provar les funcionalitats de manera aïllada al acabar d'implementar-les. Si aquestes proves anaven bé, es provava de manera integrada aquesta funcionalitat amb la resta del clúster al que pertany(per exemple, si la funcionalitat era implementar l'addició d'un esdeveniment a una llista, es prova també a obtenir tots els esdeveniments d'una llista, crear una nova, esborrar-la, etc). Per últim, es fan proves on s'executa una gran part de les crides a la API simulant un ús normal de l'aplicació per part d'un usuari.

Una petita dificultat que ens vam trobar va ser al final de l'sprint 1 i principi de l'sprint 2, quan vam intentar fer la crida periòdica a l'agenda cultural. No ens funcionava de cap manera que es comprovés si un esdeveniment “nou” ja hi era a la nostra base de dades. Per últim, no aconseguíem acabar de fer funcionar els mappers en alguns casos, ja que hi havia certa informació derivada o calculada que ens va costar generar(per temes de dependències entre capes).

Altra desviació ha sigut les funcionalitats que hem hagut de deixar sense fer, ja que vam posar-lis menys prioritat i no va donar temps a fer-les. Les més importants d'aquestes són:

- **Compartir en xarxes:** Inicialment, volíem implementar la capacitat de compartir un link a la vista d'un event de la nostre aplicació a diverses xarxes socials, com ara WhatsApp, Instagram i Twitter.
- **Valoració:** Estava previst poder valorar un event al que l'usuari havia assistit, amb una puntuació basada en estrelles, però a última hora la funcionalitat va deixar de funcionar en el backend.
- **Editar perfil:** Quan es crea un usuari, s'afegeix el nom, la imatge de perfil i els interessos. La idea era posar un botó per poder editar aquestes dades.

Durant el desenvolupament de la pàgina web vam trobar diferents dificultats referents a la sincronització amb altres serveis i la integració amb l'API de backend.

Primerament la integració de Firebase. Ho vam idear per sincronitzar l'inici de sessió amb les mateixes credencials que a l'aplicació de frontend. Vam estar seguint tutorials i buscant solucions pel problema que teníem, però no ho vam aconseguir integrar. El problema que teníem era que al fer servir "type = module" als scripts de JavaScript ens deixava de reconèixer les classes.

Per la part de la integració de l'API vam trobar problemes de la política de CORS. Vam trobar maneres de deshabilitar CORS per fer les crides d'API, però no va ser una solució factible, ja que ens vam adonar que era necessari tenir CORS activat per com teníem configurada l'API.

Aquests errors van alentir molt el desenvolupament de la pàgina web que ignorant aquests dos problemes, funciona correctament i és una eina útil per gestionar les incidències.

3.Autoavaluació

En aquest apartat realitzem una autoavaluació de tot el que hem anat fent i aprenent durant tot el procés de desenvolupament de CultureFinder, així com conceptes aplicats d'altres assignatures que hem posat en pràctica al implementar el sistema.

3.1. Conclusions generals

Com una visió general del grup, pensem que tots els membres hem treballat bastant bé i ens hem coordinat de la millor forma possible aplicant la metodologia Scrum. Revisant l'excel on hem anat registrant totes les hores, hem pogut observar que tot l'equip s'ha implicat i hem dedicat una quantitat d'hores considerable, la qual cosa ha sigut principalment perquè aquesta assignatura ens ha motivat bastant a tots els membres i estem d'acord que és, amb diferència, l'assignatura de la carrera en la qual més hem après. Fonamentalment, pel fet d'aplicar tots els conceptes que hem adquirit en un projecte bastant complet.

Un dels aspectes més positius que podem destacar és la llibertat que hem tingut des d'un principi per afegir funcionalitats i com organitzar-nos a l'hora de desenvolupar el sistema CultureFinder, tenint en compte la disponibilitat de cada membre de l'equip.

Per acabar, com a inconvenient que modificariem des de l'inici és la forma en la qual vam estimar les històries d'usuari i la manera en la qual vam fer la granularització de les tasques. Tanmateix, tal i com s'ha comentat a l'anterior punt aquesta dificultat, a mesura que vam avançar en els següents sprints vam anar millorant aquest aspecte i vam corregir diferents errors de planificació que havíem fet al sprint anterior. Com a conclusió, hem vist que amb el pas de temps es poden aprendre conceptes d'organització i gestió que al principi ni pensàvem que suposarien un problema.

3.2. Conceptes aplicats adquirits en altres assignatures

Al ser un projecte d'enginyeria del *software*, hem hagut d'utilitzar diversos conceptes de diferents assignatures, especialment d'assignatures de l'especialitat d'enginyeria del *software*.

Primerament, el que fa a tots els aspectes de planificació i gestió de tasques, a Gestió de Projectes del Programari (**GPS**) vam aprendre algunes de les metodologies àgils per dur a terme alguns dels nostres projectes. Aquestes van ser Scrum, una metodologia que hem utilitzat per a tota la planificació del projecte, a partir d'històries, tasques i treballar per sprints.

Tot el concepte d'històries d'usuari, èpiques i tasques, ho vam aprendre a Enginyeria de Requisits (**ER**). Aquests termes els hem utilitzat per a dividir les funcionalitats de la nostra aplicació en parts més petites i manejables per poder desenvolupar. També vam veure a ER tot el tema de requisits no funcionals, els quals hem tingut en compte en desenvolupar l'aplicació del nostre projecte.

Pel que fa als patrons de disseny que hem implementat en el nostre projecte, com el patró adaptador, singleton o el model-vista-controlador, els vam aprendre a l'assignatura d'Arquitectura del Programari (**AS**).

I pel que fa a la gestió dels repositoris, sobretot el procés de fer commits, push, gestió d'issues, pull requests, etc., tots aquests conceptes els vam aprendre a Projectes de Programació (**PROP**), en utilitzar GitLab com a eina per pujar el nostre codi, que és una versió molt semblant a GitHub, la qual hem utilitzat per fer l'aplicació CultureFinder.

4. Feina individual al llarg del projecte

4.1. Óscar Delgado:

El meu treball s'ha centrat en el desenvolupament del frontend de l'aplicació. Durant el primer sprint m'he enfocat en la implementació les vistes i funcionalitats relacionades amb el mapa. També m'he encarregat de les vistes per crear usuaris i del model d'usuari.

Per crear un mapa funcional, he hagut d'utilitzar els plugins de Google Maps i permission handler. El primer m'ha permès mostrar un mapa interactiu en la vista, el qual es pot centrar a la ubicació actual del dispositiu i mostrar els diferents esdeveniments al mapa, permetent una vista prèvia d'aquests si es seleccionen. El segon plugin m'ha ajudat amb tot el relacionat amb els permisos d'ubicació d'Android.

Pel que fa a la creació d'usuaris, he construït una vista amb un formulari i una altra per seleccionar els interessos entre les categories de tots els esdeveniments del sistema. També he creat el model d'usuari i he establert la navegació entre algunes vistes.

Durant el segon sprint, he començat modificar les credencials de l'API de Google Maps perquè els meus companys poguessin visualitzar correctament el mapa. També he modificat el mapa perquè comenci amb un zoom centrat a la ubicació actual del dispositiu, utilitzant l'API de Geolocator.

He creat les diferents vistes per crear un compte i realitzar l'inici de sessió a l'aplicació. Per facilitar l'autenticació d'usuaris tant amb correu electrònic i contrasenya com amb Google, he utilitzat Firebase. També he implementat la verificació de correu electrònic i

la recuperació de contrasenya amb l'API de Firebase. Per realitzar el registre, autenticació i tancament de sessió d'usuaris, he implementat crides a l'API del nostre backend. Per mantenir la persistència dels usuaris connectats i facilitar als meus companys l'accés al token de l'API per fer crides al nostre backend, he utilitzat Shared Preferences, implementant tots els mètodes per gestionar el token de l'API i la informació de l'usuari connectat.

En el tercer i últim sprint, he començat finalitzant la implementació de les vistes necessàries per crear un nou compte d'usuari. Primer, he fet un refactor del qüestionari inicial d'interessos del nou usuari, millorant la interfície d'usuari i creant finestres emergents per mostrar les excepcions. També he implementat el selector d'imatges per triar la imatge de perfil, configurant prèviament els permisos d'Android per accedir a la memòria del dispositiu. En aquesta mateixa vista, he implementat el formatge necessari per introduir la data de naixement i les finestres emergents adequades per a les diferents excepcions.

D'altra banda, he implementat un repositori per utilitzar el servei ofert pel grup PlugFinder, que consistia en obtenir tots els carregadors que compleixen cert filtre. Per a la nostra aplicació, he fet que en la vista detallada de cada esdeveniment es mostressin els carregadors més propers a l'esdeveniment, mostrant-ne l'adreça i els tipus de carregadors disponibles.

Després, he configurat el servei de Firebase Storage i he implementat el sistema de guardat i actualització d'imatges de perfil. He fet un refactor del codi per millorar l'eficiència i la independència del mateix, minimitzant les crides a l'API amb la classe "user_updater", convertint el repositori d'usuaris en Singleton i creant un "currentUser" únic.

He configurat i implementat les notificacions push amb FCM de manera que també arribin quan l'aplicació estigui en segon pla. He fet un refactor d'Authview i de certes crides a endpoints tenint en compte els tokens del dispositiu.

Finalment, he fet que el perfil de l'usuari connectat es vegi en la barra lateral, la vista que mostra la informació de l'usuari, i he creat la pàgina principal on es mostren carrusels amb tots els esdeveniments, els més populars i els recomanats per a l'usuari.

4.2. Marc Duch:

Pel que fa el desenvolupament del front end, durant el primer Sprint vaig fer de *Scrum master* i com a tal em vaig encarregar de dirigir les primeres reunions per tal de dividir-nos la feina, decidir quines històries incloure al sprint amb l'equip i d'escriure el Sprint planning report.

També em vaig encarregar de generar el codi inicial que es basava en la implementació del primer model del sistema, l'*EventModel* i una implementació mock de la futura classe *EventRepository*, que com el nom indica, és una classe que aplica el patró repository per tal de crear una abstracció a la hora de aconseguir els *EventModels*. Tot i que aquest codi inicial el vaig escriure amb la intenció de que fos temporal, degut a la meva falta de coneixement, el vaig haver de rescriure varies vegades al llarg del primer sprint, fins que finalment vaig implementar la primera crida de la API.

Durant aquest sprint vaig implementar la vista de la llista d'esdeveniments amb un sistema de paginació utilitzant Riverpod i la seva abstracció del patró observer per tal d'actualitzar la llista.

Durant el segon sprint el vaig començar implementant una primera versió de la "homepage", una vista que permet accedir a les pàgines principals.

Vaig aprofitar la setmana santa per reescriure la vista en detall d'un esdeveniment per que s'apropés més al disseny mostrat al mockup. Amb això acabat, vaig implementar el popup dels filtres, el qual vaig reutilitzar més endavant al implementar els filtres en el mapa.

Els filtres que vaig implementar son, filtrar per nom, categories, distància i per dates d'inici i de fi. Tot i que al final, el filtre per distancia no apareix a la UI.

També vaig fer petits ajustos a parts del codi que vam fer al primer sprint, com treure mètodes del *EventRepository* que no feien falta o separar la lògica de la vista del Calendari en dos classes *View* i *Notifier* (controlador).

Finalment, al tercer sprint el vaig començar fent un petit refactor del sistema per demanar el permís de la ubicació, ara és un widget que determina si l'usuari ha donat el permís o no, i en cas de que no s'accepti el permís, es mostra un altre widget (un text).

Amb l'ajuda d'una llibreria vaig implementar el sistema de clustering per tal de mostrar els esdeveniments en el mapa.

I per últim vaig implementar el mapa de calor. Originalment volíem fer servir un package per tal de facilitar la mostra visual d'aquestes dades, però aquest portava 3 anys sense ser actualitzat. Després de mirar la implementació vaig descartar la idea de actualitzar-lo jo mateix ja que la implementació estava massa actualitzada. Finalment vaig decidir implementar-ho fent servir el sistema de Map Tiles que fa servir google maps per sobreposar imatges sobre un mapa. Bàsicament, tradueix les posicions dels esdeveniments al mapa, a posicions en un pla 2D on aquestes doncs es fan servir per construir imatges amb cercles amb un radi que depèn del nombre d'assistents.

4.3. Miguel Moreno:

Durant el projecte, he treballat a la part de frontend, implementant principalment vistes, i també alguns models i repositoris.

Al primer sprint, vaig centrar-me en fer vistes, tot i que la majoria del temps el vaig dedicar a formació en flutter i dart.

Primer vaig fer la vista per el calendari, incloent els events del dia seleccionat si hi havien, però no vaig implementar que fagi una crida a la API i per tant no mostrava events reals.

Després vaig implementar la vista d'events detallada. Aquesta sí li vaig fer la crida a la API per obtenir les dades correctes. Si l passes una id, et mostra la informació de l'event, llistant els diferents camps, inclosa la imatge de l'event, si tenia. Vaig deixar els camps que podien ser nuls buits sense missatge alternatiu, i no em va donar temps a millorar l'aspecte de la vista.

Al segon sprint, vaig començar fent que es poguessin fer incidències d'un event. Primer, fent el repositori d'incidències, amb la crida per postejar una incidència donat una descripció i un event. A continuació, afegint un botó a la vista d'events detallada que obre un pop-up on es pot introduir l'incidència, i quan es prem "Acceptar" es fa la crida a postejar incidència. Provisionalment, vaig haver de passar un userId manualment ja que la funció per obtenir l'userId no estava implementada.

També vaig començar a implementar les llistes de favorits. Vaig fer la vista on surten les llistes de favorits de l'usuari, que sempre té la llista "Favorits" com a predeterminada quan es crea un usuari (això es fa des del backend). Quan es prem una llista, vaig fer que es mostri la vista d'aquesta llista, amb els events corresponents. Per tal de que funcionin aquestes vistes, vaig implementar el repositori de llistes, amb crides a la API per que retorni totes les llistes d'un usuari, i per retornar els events d'una llista. També vaig fer que es poguessin postejar noves vistes, donat un nom, i ho vaig afegir a la vista de llistes amb un botó. Vaig implementar també el model de llista, per facilitar el funcionament de les vistes i les crides.

Al tercer sprint, vaig començar per el que em quedava de llistes de favorits i incidències.

Per les llistes, primer de tot, vaig implementar la crida per afegir un event a una llista. Aquesta funció la vaig afegir primer a la vista d'events detallats, en un botó que obre un pop-up on et surten totes les llistes ja existents, i la possibilitat de

crear una nova llista, per afegir aquest event a la llista seleccionada. Després vaig afegir aquest mateix pop-up al llistat d'events, en el qual surt si mantens premut un event. També vaig implementar la crida per eliminar una llista de favorits, el qual es fa anant a la vista de llistes i mantenint premut una llista, on surt un missatge, i si dones a "Acceptar" elimina la llista corresponent.

A les incidències, primer de tot vaig substituir l'userId posat manualment per una crida a l'API-token, ja que es va canviar el mètode per postejar incidències a la part del backend. Un cop deixat per definitiu el tema de postejar incidències, vaig fer la vista d'incidències de l'usuari. Vaig afegir una crida per obtenir les incidències que ha fet un usuari, i a la vista es mostren en un llistat. Vaig fer un pop-up que surt quan cliques una de les incidències, i et dona informació sobre la incidència corresponent.

Finalment vaig implementar el calendari d'usuari, aprofitant el calendari fet al primer sprint. Aquest calendari mostra els events al que l'usuari ha marcat que assistirà. Vaig implementar la crida que retorna el llistat d'events als que assisteix un usuari. Amb aquesta llista, vaig fer una funció on es crea una llista amb els events assistits on la data coincideix amb la del dia seleccionat al calendari, i mostra aquests events.

Vaig afegir a la barra lateral el botó que redirecciona a la vista del calendari d'usuari, i que el botó de la llista de favorits redireccioni a la llista de favorits de l'usuari, que es crea amb l'usuari.

4.4. Daniel Morón:

Respecte al meu treball pel backend comença inicialment al primer sprint, i es va basar principalment en la implementació de les crides API que utilitza el model *Event* i el model *Incident*. Al principi em vaig focalitzar en la configuració de la base de dades, la qual ens va retardar bastant tot el tema d'aprenentatge i formació. Després a l'hora de programar, les seccions en les quals em vaig centrar van ser: processar correctament el preu i les dates finals i inicials proporcionades per l'API de l'Agenda Cultural, permetent tots els tipus de filtratges que tindrà el nostre producte pels esdeveniments amb una mateixa crida a l'API, a més de les crides individuals per ambit, categoria i d'altres categories i per rang de dates. Addicionalment, he estat fent una primera versió del model *Incident* esmentat anteriorment, el qual ens facilita la creació d'incidències per reportar qualsevol tipus d'error. Per acabar, vaig estar implementant la tasca de crida periòdica a l'API de l'Agenda Cultural, ja que ens interessa fer la sincronització una vegada al dia per tenir les nostres dades actualitzades. Vaig estar fent *pair programming* amb Ivan per acabar de treure l'actualització periòdica de les dades, ja que ens hem trobat el problema de que si esborrem la BD i tornem a carregar-la, a més de que els esdeveniments antics es perden, els identificadors canvien i això implica que la informació personalitzada de cada usuari romandria inconsistent.

Pel que fa al segon sprint, s'ha basat principalment en la implementació del servei que oferim durant aquesta fase, juntament amb el refactor corresponent relacionat amb la crida relacionada amb els filtres de cerca. Tot i això, al principi del sprint em vaig centrar en acabar les tasques prèvies que s'havien quedat del primer sprint, la qual era la sincronització periòdica de les dades. Cal destacar que a mesura que anàvem implementant nou codi, anaven sorgint noves millores, per aquest mateix motiu ens va retardar una mica el refactor inicial que vam comentar.

Una vegada vaig acabar això amb l'ajuda d'Ivan fent diferents dies *Pair Programming*, em vaig focalitzar més, en el refactor que vam comentar Ivan i jo amb el Jordi sobre la separació entre capes i les millores possibles que podíem afegir a l'arquitectura del nostre projecte. Al principi vam tenir diferents problemes de configuració i dubtes sobre alguns conceptes, que vam anar esmentant i solucionant conjuntament a mesura que avançàvem en les activitats del sprint. A la part final, em vaig centrar fer diferents modificacions i millores pel que fa a Esdeveniments i Incidències per augmentar i precisar de millor

manera el funcionament del filtratge, i a partir d'aquí, vaig implementar el servei que oferim, el qual consisteix en donat un radi, una posició i un rang de data inicial i final, retorna tots els esdeveniments que estan localitzats dins d'aquell radi.

Per al darrer sprint, principalment m'he centrat en acabar les tasques que tenia assignades relacionades sobretot amb el refactor de totes les crides periòdiques amb l'Agenda Cultural per tal d'evitar conflictes entre capes i a més evitar esdeveniments repetits quan es fa la sincronització corresponent. Per acabar, he intentat donar suport a la resta de l'equip en les seves tasques i he estat preparant la presentació final del nostre projecte.

4.5. Iván Risueño:

La meva aportació al backend comença a l'sprint 1, on vaig fer tota la infraestructura inicial de l'aplicació, dividint-la per *packages*, *subpackages* i classes. Aquesta infraestructura la hem mantingut al llarg del projecte, i és la subdivisió entre capes amb els serveis, la lògica, els controladors i els repositoris. Vaig fer també la connexió a la base de dades, que des d'un primer moment ha sigut funcional. A més, vaig escriure la primera versió del que seria el codi inicial per a la classe User i la classe Event (també les funcions que parsejen de JSON a Event i viceversa, així com algunes de les que netegen les dades), i la seva lògica per a obtenir dades mitjançant la API, i poder editar o guardar noves a la base de dades.

Pel que fa al segon sprint, el backend ens vam trobar que calia fer un refactor enorme perquè no vam tenir en compte aspectes clau com podria ser la securització de la API, el tractament d'excepcions, o augmentar la independència entre capes afegint entitats, DTOs i els mappers. Tot el que he mencionat va ser desenvolupat per mi, a excepció dels DTOs i els mappers que ens ho vam dividir en Dani i jo. En quant a les funcionalitats principals, vaig desenvolupar el codi necessari per a afegir assistències i llistes d'esdeveniments favorits, les valoracions d'esdeveniments, categories d'interés per als usuaris (així com l'algorisme de recomanació d'esdeveniments) i usuaris administradors al sistema.

Per últim, les meves tasques al tercer sprint han sigut principalment afegir *liquibase* al projecte, implementar les notificacions push i la configuració de *Firebase* parlant amb Oscar, l'algorisme per a obtenir els esdeveniments més populars, i canviar la manera en la que feiem el deployment de l'aplicació a Virtech, a més de preparar amb en Dani la presentació *onboarding*.

En general, les meves tasques com a *backend leader*, a més de participar en el desenvolupament individual d'algunes funcionalitats, ha sigut mantenir la comunicació i rebre molt de feedback per part de l'equip de *frontend*, així com efectuar tasques que tenen a veure amb la qualitat del *backend* com a aplicació. A més, he estat assabentat en tot moment de quina era la situació del *backend* respecte als objectius que ens hem marcat, i els hem aconseguit complir-los quasi tots*.

*A última hora ens vam adonar de que la funcionalitat d'afegir valoracions a la base de dades havia deixat de funcionar. Després de buscar la causa de l'error

durant hores individualment, en Dani i jo vam estar una estona més mirant juntament i al final no ho hem pogut solucionar. A banda d'aquest contratemps, totes les funcionalitats que vam decidir en un principi per al backend han estat implementades i es poden testejar fent ús de la nostra API.

4.6. Marc Rodriguez:

Des del primer sprint vaig començar a l'equip de backend. La major part del treball a aquest primer sprint va ser la formació i la inicialització del projecte mitjançant el framework Spring Boot. A més vaig estar desenvolupant maneres de com realitzar la crida de dades periòdica de manera eficient i també vaig col·laborar en algunes tasques com el processament de dades. A més de col·laborar a les reunions setmanals de Backend sobre com implementar la base de dades i quin servidor utilitzar per pujar l'aplicació.

Més tard, al segon sprint vaig canviar l'enfocament del meu treball per treballar a la pàgina web d'administració per gestionar incidències. Vaig estar desenvolupant una web amb llenguatge javascript amb arquitectura model-view-controller. Aquesta web conté una pàgina inicial on pot iniciar sessió d'un usuari ja registrat a l'aplicació i a més, amb permisos d'administrador. Una vegada loguejat, l'usuari pot veure les incidències filtrant per diferents filtres i obrir un formulari on resoldre la incidència. Una gran part d'aquest sprint també la vaig dedicar la formació d'aquestes tecnologies. A part d'aquesta feina, també vaig desenvolupar la feina de sprint master. Intentant facilitar la feina i la comunicació de l'equip per treballar de manera eficient.

El meu treball al tercer sprint va ser continuar amb la pàgina web d'administració. Vaig millorar l'aspecte de la pàgina web i el seu funcionament. També vaig estar en comunicació amb l'equip de frontend i backend per veure com abordar el problema d'iniciar sessió amb les mateixes credencials que a l'aplicació de frontend. Vaig estar intentant sincronitzar Firebase amb la meua pàgina web el que em va donar molts problemes. També vaig provar altres maneres de fer l'autenticació mitjançant un usuari creat al swagger i loguejant-se amb l'api. Un altre problema que vaig trobar va ser un error de CORS a l'hora d'intentar connectar-me a l'api del backend. Per tant, una gran part d'aquest sprint ha estat solucionant bugs i errors.