



FreskitoBCN

Canaleta I Garcia, Sara

Cano Gil, Jordi

Casajoana Santandreu, Miquel

Cervera Moreno, Carles

Cilleruelo González, Samuel

Gómez Cobos, Javier





Gestió del Projecte amb Scrum



Metodologia Scrum

El projecte segueix la metodologia Scrum, un procés iteratiu i incremental, utilitzant Taiga per a la gestió de tasques.



Fase de Desenvolupament

Dividida en sprints, on l'equip organitza, desenvolupa i prova el codi, finalitzant amb una versió lliurable.



Fase d'Inception

Es defineix l'abast, s'identifiquen riscos, es crea el product backlog inicial i es prepara l'entorn de treball.

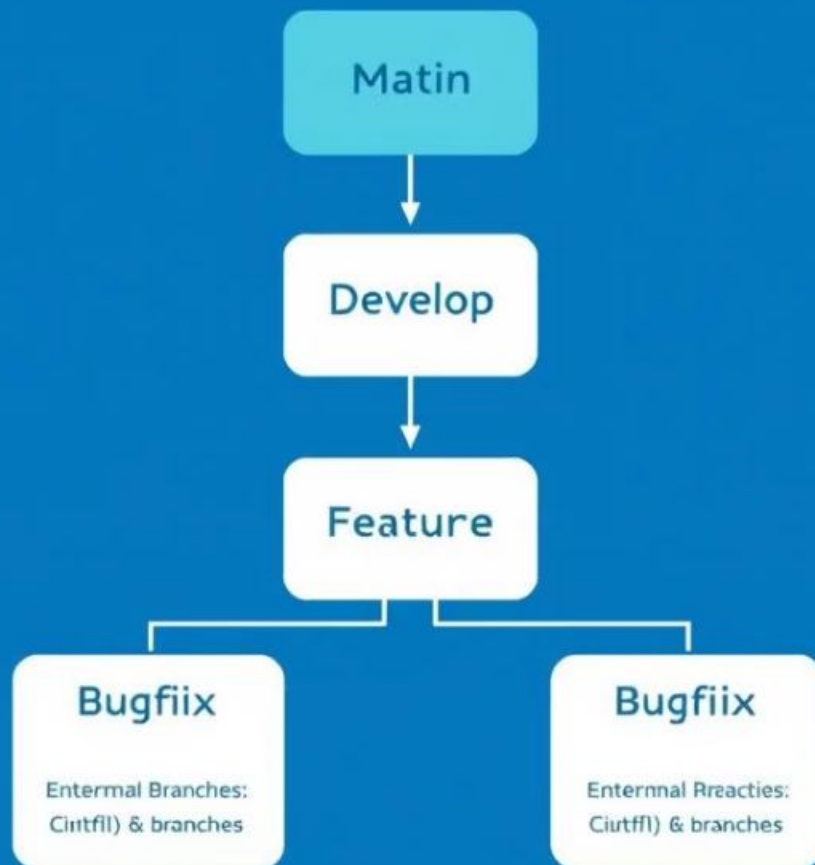


Rol de Sprint Master

A cada sprint, un membre de l'equip actua com a sprint master, facilitant el procés àgil i assegurant el compliment de les pràctiques Scrum.

GittFlow model

Git Flow



Gestió del Repositori amb GitFlow

Estratègia GitFlow

L'administració dels repositoris segueix l'estratègia GitFlow, un model de ramificació per gestionar el desenvolupament amb Git en metodologies àgils.

Branques Principals

La branca **main** conté el codi estable per a producció, actualitzant-se amb **merge** de **develop** al final de cada sprint.

Branques de Desenvolupament

La branca **develop** és la base per a les funcionalitats, que un cop validades, es fusionen aquí. Les branques **feature** implementen noves funcionalitats des de **develop**.

Branques de Bugfix

Quan es detecta un bug a **develop**, es creen branques **fix** per implementar la correcció de l'error.

Comunicació Interna de l'Equip

Comunicació amb Slack

La comunicació interna de l'equip FreskitoBCN s'ha gestionat principalment a través de Slack, utilitzant canals específics per a cada àrea del projecte (backend, frontend i agile).



Reunions Setmanals

A més, s'han fet servir reunions de seguiment setmanals per revisar l'estat de les funcionalitats, detectar bloquejos i repartir responsabilitats.



Gestió de la Qualitat amb SonarQube

Anàlisi Contínua del Codi

Per garantir la qualitat del codi, s'ha fet ús de la plataforma SonarQube, que analitza contínuament el codi font.

Detecció de Problemes

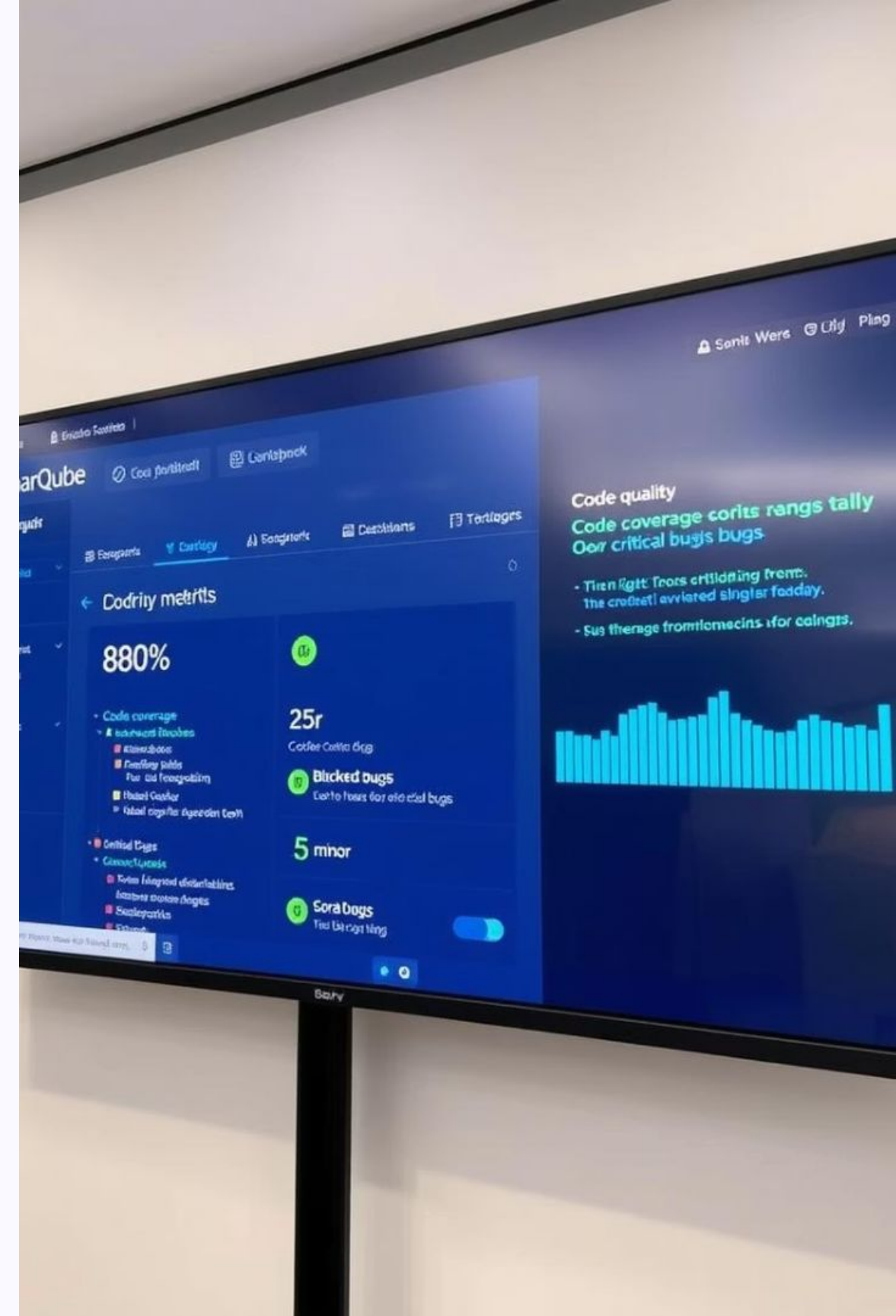
SonarQube detecta vulnerabilitats, errors i possibles mals hàbits de programació, assegurant un codi robust.

Regles de Qualitat

Les regles de qualitat cobreixen aspectes com la cobertura mínima de proves unitàries, l'absència de bugs crítics i la mantenibilitat del codi.

Estàndard Homogeni

D'aquesta manera, es manté un estàndard homogeni durant tot el desenvolupament, evitant la degradació del projecte amb el pas del temps.



Estratègia de Proves



Disseny de Tests

La metodologia de proves segueix un enfocament basat en TDD (Test-Driven Development), dissenyant i executant tests abans d'implementar funcionalitats.



Unit Tests

Cobreixen funcions i mètodes de la lògica de negoci, assegurant la correcció de components individuals.



Tests d'Integració

Asseguren la correcta interacció entre mòduls, com l'accés a la base de dades i la resposta dels endpoints.



Cobertura

SonarQube genera un informe amb la cobertura de codi per garantir la qualitat.



Gestió de Configuracions i Desplegament



Docker Compose

Automatitza la configuració i inicialització del backend i la base de dades en un entorn controlat.



Versionat amb Git

Tot el codi i la configuració de desplegament estan versionats en Git, assegurant que la infraestructura i les aplicacions són replicables i portables entre entorns.

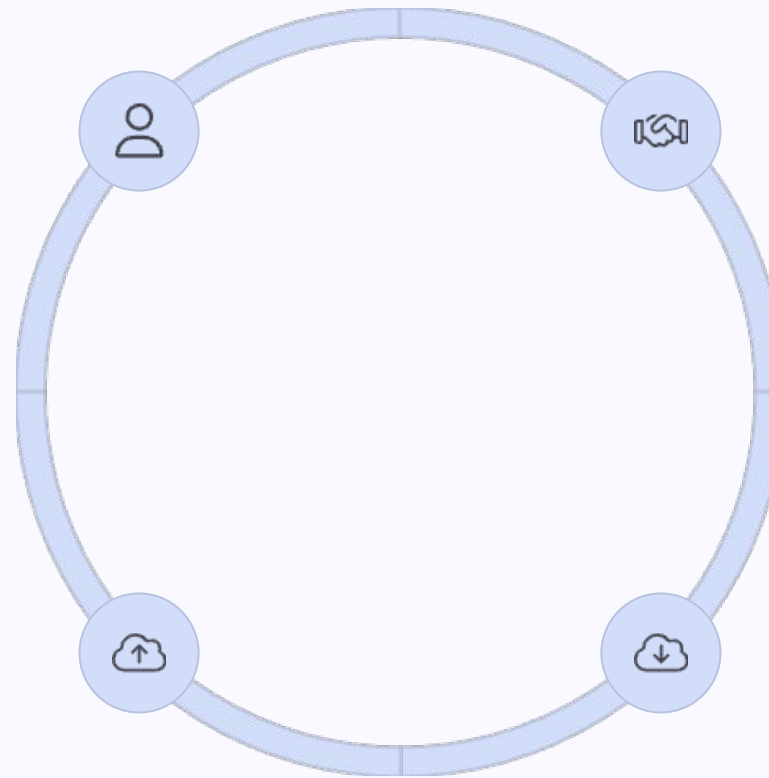
Interacció amb Altres Equips

Col·laboració Activa

S'ha establert una col·laboració activa amb altres equips mitjançant reunions de coordinació amb els portaveus de cada projecte.

Subministrament de Dades

FreskitoBCN subministra dades de refugis climàtics a altres aplicacions com eco-move, mitjançant l'endpoint `/api/refugios/cercanos/`.



Acords d'API

S'han debatut i acordat les crides d'API entre aplicacions, definint clarament els serveis consumits i subministrats.

Consum de Dades

FreskitoBCN consumeix dades del servei Culturunya, accedint a esdeveniments culturals dins d'un radi concret.



Gestió de Bugs

Comunicació Immediata

Quan es detecta un bug, es comunica immediatament a l'equip a través de Slack per a una ràpida resposta.

Registre a Taiga

A continuació, es registra com a tasca a Taiga, associant-li un número identificador i una descripció clara per al seguiment.

Branca Específica

El membre de l'equip encarregat de resoldre el problema crea una branca específica seguint el patró **fix/<número-tasca>-descripció**.

Pull Request i Revisió

Un cop aplicada la correcció, es fa una pull request cap a **develop**, on es revisa el codi abans d'integrar-lo, assegurant la qualitat.

Assistents de Codi per a Productivitat

Copilot

Suggeriments automàtics de codi i snippets per agilitzar el desenvolupament.

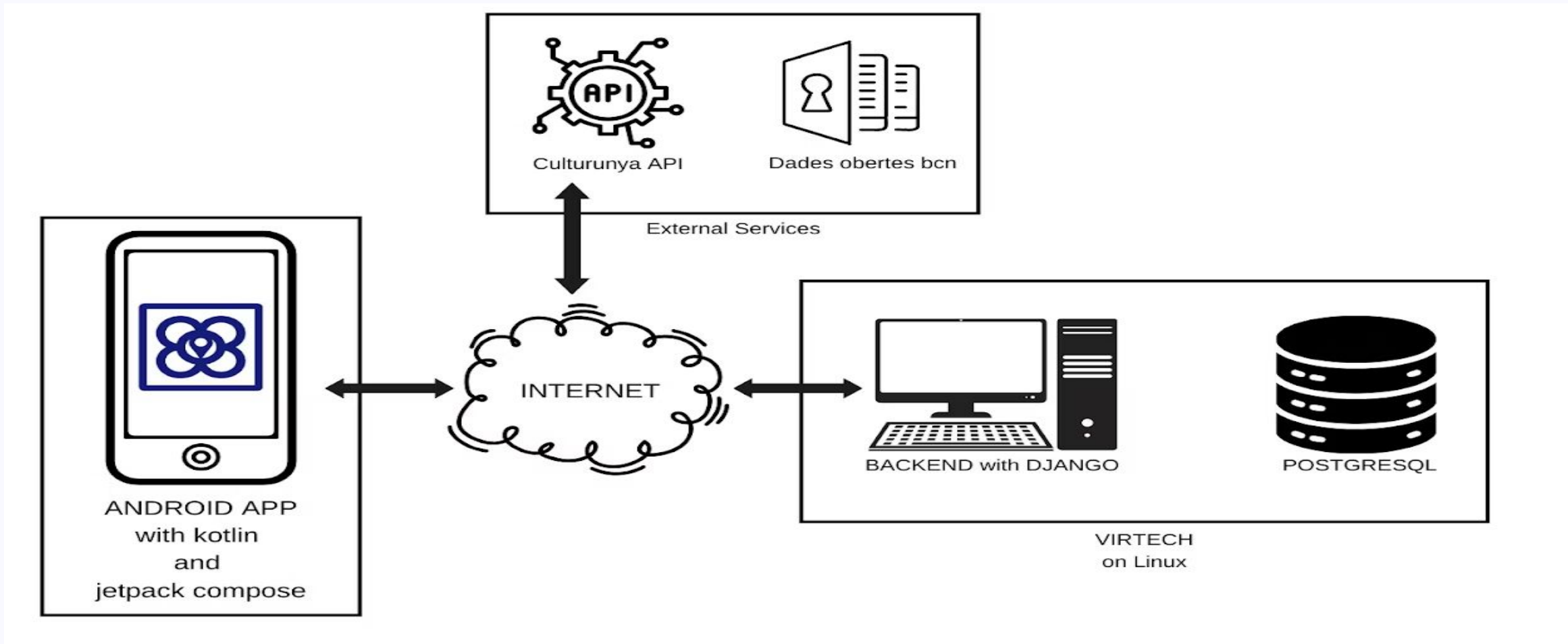
ChatGPT

Consultes puntuals i generació de fragments de codi amb suport immediat.

Validació Manual

Totes les propostes de codi es revisen i ajusten per garantir qualitat i seguretat.

Arquitectura del Servidor



Màquina Virtual i SO

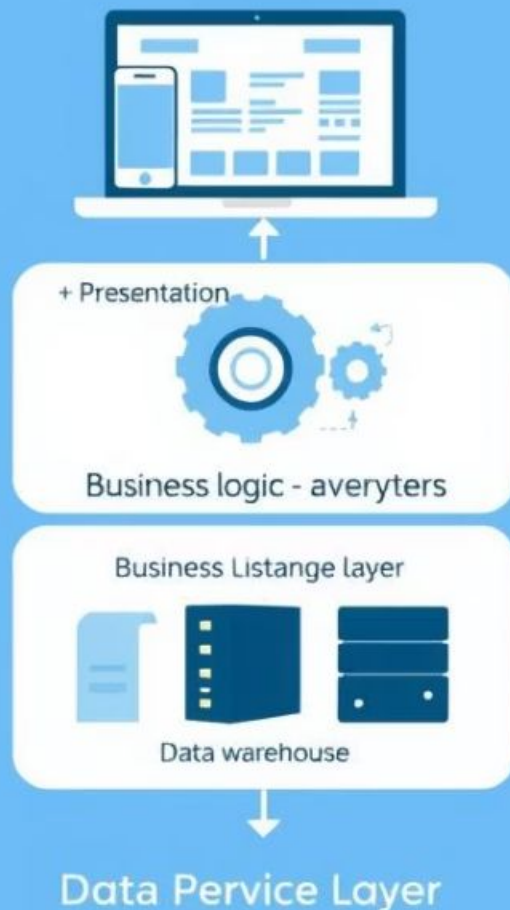
Utilitzem una màquina virtual (VIRTECH) proporcionada per la universitat, amb Linux com a sistema operatiu.

Contenidors Docker

El desplegament es realitza amb Docker i Docker Compose, amb dos contenidors principals: un backend Django (Python) i una base de dades PostgreSQL.

Comunicació Interna

Els contenidors es comuniquen mitjançant una xarxa privada de Docker, assegurant aïllament i control eficient.



Patró d'Arquitectura de Tres Capes

Capa de Presentació

Representada per l'aplicació Android, construïda amb Jetpack Compose, que gestiona la interfície d'usuari.

Capa de Lògica de Negoci

Implementada amb Django, que segueix el patró Model-Vista-Controlador per processar les dades i la lògica de l'aplicació.

Capa de Persistència

Gestionada per PostgreSQL, encarregada d'emmagatzemar i recuperar tota la informació de l'aplicació de manera eficient.

APIs Pròpies i Externes



API Pròpia

El backend exposa una API RESTful amb Django REST Framework per gestionar usuaris, refugis i valoracions.



Culturunya API

Permet obtenir informació sobre activitats culturals properes a la ubicació de l'usuari.



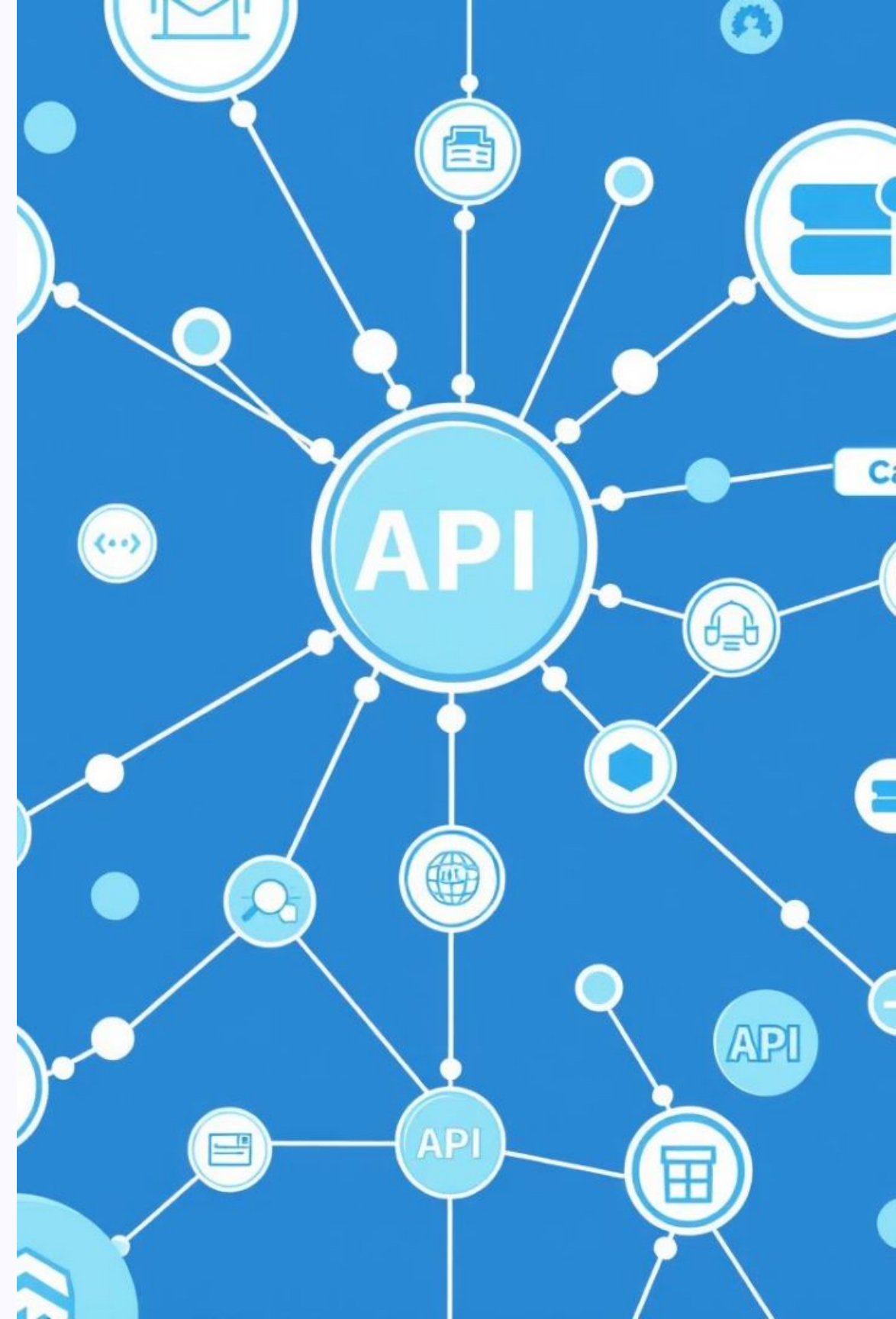
Google Maps API

Integrada per a la visualització i geolocalització precisa dels refugis climàtics.



Dades Obertes Barcelona

Utilitzada per obtenir informació actualitzada sobre els refugis climàtics de la ciutat.



Consum de Serveis Externs: Culturunya

1

Integració Cultural

FreskitoBCN integra funcionalitats culturals mitjançant el consum de serveis externs com el de Culturunya.

2

Endpoint Consumit

L'aplicació utilitza l'endpoint <http://nattech.fib.upc.edu:40369/api/events/filter/> per cercar esdeveniments.

3

Paràmetres de Cerca

Retorna esdeveniments dins d'un radi específic a partir de latitud, longitud i distància en quilòmetres.

4

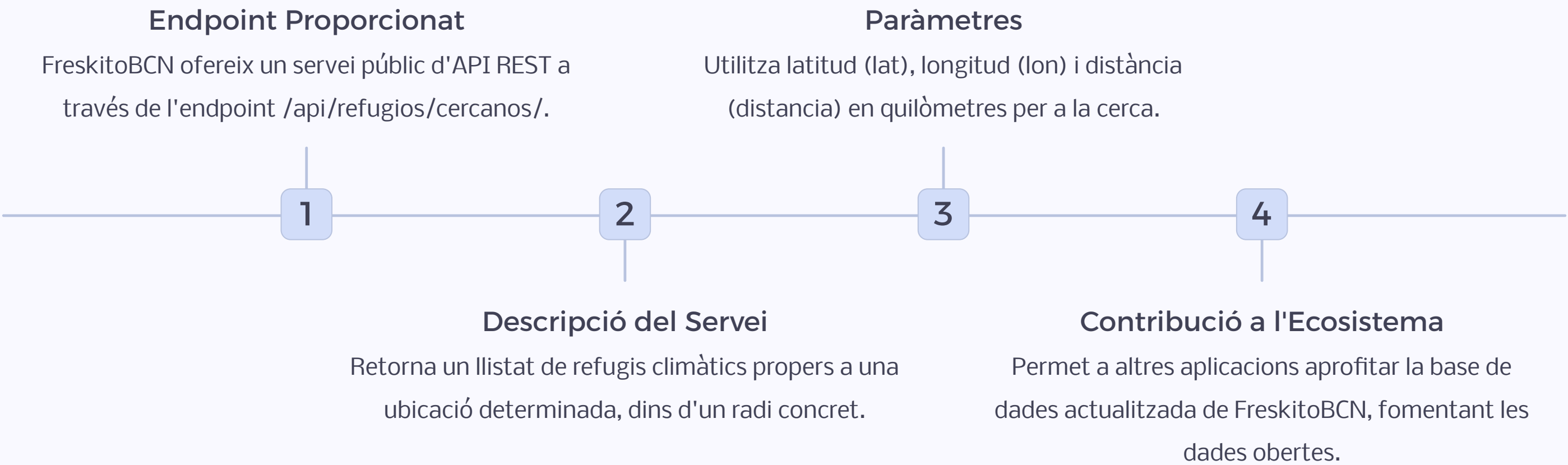
Valor Afegit

Aquest servei permet a FreskitoBCN oferir contingut cultural contextualitzat, reforçant el valor de l'app.





Subministrament de Serveis: API de Refugis



Consum de Dades Obertes de Barcelona

Font Oficial

FreskitoBCN utilitza dades obertes de la Xarxa de Refugis Climàtics de l'Ajuntament de Barcelona.

Integració de Dades

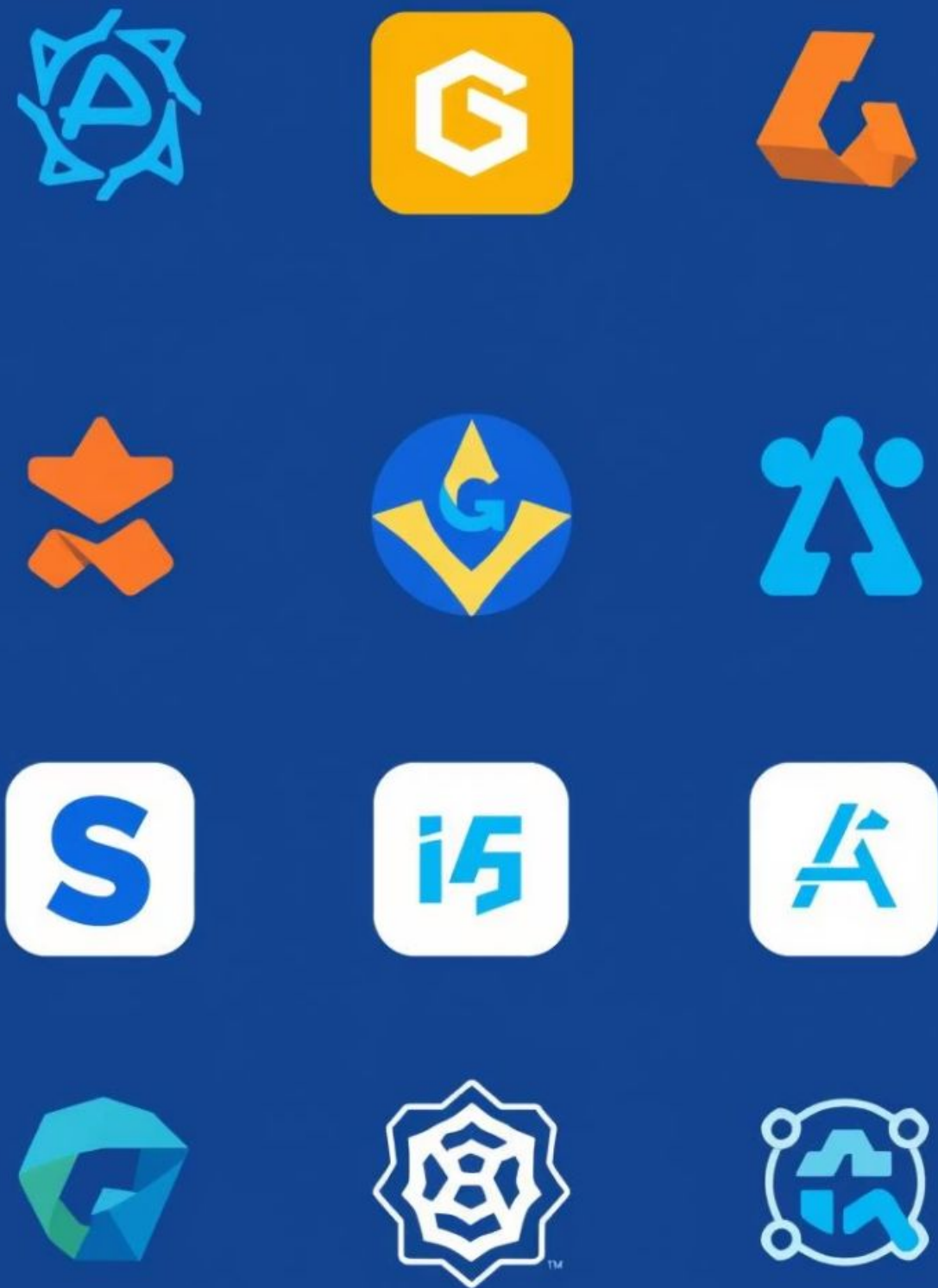
Les dades es consumeixen mitjançant la lectura i processament automàtic d'un fitxer JSON, actualitzat periòdicament.

Funcionalitats Clau

Permet mostrar refugis en un mapa, oferir fitxes detallades, filtrar per tipus i consultar els més propers.

Interacció Social

Inclou opinions, comentaris i la possibilitat de reportar problemes per part dels usuaris.



Ús de Frameworks de Desenvolupament

1

Backend Django

Django i Django REST Framework per a una API RESTful robusta i escalable.

2

Android Kotlin

Kotlin amb Jetpack Compose per a una interfície moderna i reactiva en Android.

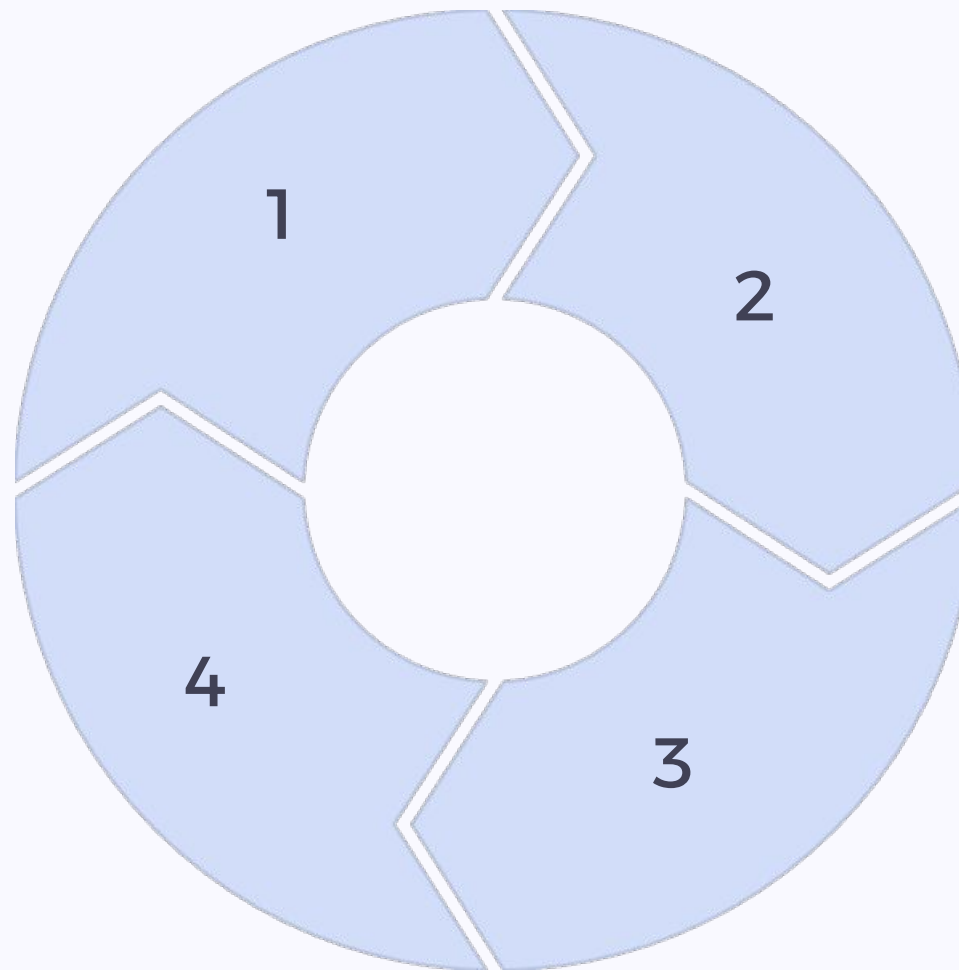
Integració Contínua i Desplegament

Gestió de Codi

GitHub amb model de ramificació basat en feature branches i pull requests per a la col·laboració.

Accés Públic

L'API de FreskitoBCN és accessible públicament, facilitant el desenvolupament i test de les aplicacions client.



Bones Pràctiques

Aplicació de bones pràctiques de desenvolupament col·laboratiu, amb revisió manual del codi.

Desplegament Docker

Servidor desplegat en màquina virtual amb Docker i Docker Compose per a un entorn aïllat i replicable.