

# **STRAVA-KO PROIEKTUAREN** **DOKUMENTAZIOA**

# STRAVA

2021-2022 ikasturtea

Informazio Sistemen Analisia eta Diseinua

Egileak: Andoni Olabarria, Gaizka Zuazo, Eneko Perez, Martin Amezola, Gorka del Río

GitHub-eko url-a: <https://github.com/spaidermart/STRAVA>



# AURKIBIDEA

|   |          |
|---|----------|
| <b>1. Proiektuaren helburuen dokumentua</b> |          |
| 1.1 Sarrera .....                           | 2        |
| 1.2 Proiektuaren deskribapena .....         | 2        |
| 1.3 Proiektuaren arkitektura .....          | 2        |
| 1.4 Plangintza .....                        | 3        |
| <b>2. Domeinuaren eredua .....</b>          | <b>3</b> |
| <b>3. Diseinua</b>                          |          |
| 3.1. Sekuentzia diagrama .....              | 4        |
| <b>4. Inplementazioa .....</b>              | <b>5</b> |
| <b>5. Bibliografia .....</b>                | <b>6</b> |

# 1. Proiektuaren helburuen dokumentua

## 1.1 Sarrera

Dokumentazio honen bidez, Informazio Sistemen Analisia eta Diseinua irakasgaiari 2021. urtean egindako Strava-ko proiektuaren azalpenak emango dira. Azalpen hauen artean inplementatutako kodea, interfazeak, grafikoak, baliabideak, etab. aurkitu daitezke, puntuz puntu idatzita orden logiko bat jarraituz.

Eskatu zaiguna Strava-tik erabiltzaile baten hainbat entrenamenduen informazioa lortzea API-en bidez izan da, ondoren datu horiek filtratzeko Python-eko programa baten bidez eta datu horien erregistroa datu base batean gordetzeko.

Proiektua egiteko 2 hilabete inguru izan ditugu, lehenik eta behin diseinuarekin lan egiteko eta ondoren egin beharreko programak kodetzeko eta txosten hau egiteko. Ondorengo ataletan egindako gauzen azalpenak sakonduko dira.

## 1.2 Proiektuaren deskribapena

Eskatzen ziren funtzionalitateen artean, gure proiektuak ondorengoak betetzen ditu: lehenik eta behin, Strava-ko erabiltzaile baten aktibitateen datuak lortuko ditu. Datu horiek datu base batean gordeko dira, aktibitate bakoitza identifikatzaile batekin erregistratuz aktibitate horren medizioetara joan nahi izateko kasuan hauetara ailegatzeko. Aktibitate bakoitzeko agertuko diren medizioak ondokoak dira: Moving, LatLng (latitudea eta luzera), Velocity\_Smooth, Grade\_Smooth, Cadence, Distance, Altitude, HeartRate eta Time (abiadura, pultsazioak... hauetan batzbestekoa agertuko da). Datu horiek sartu ondoren, beste aukera batzuk egongo dira.

Noski, aplikazio bat denez, dena erakutsiko da interfaze baten bidez, eta honen erabilerarekin erabiltzaileak datu hauek eguneratzeko edo bilaketa filtroak erabiltzeko aukera izango du. Aukera hauek ondorengoak izango dira: datuak freskatu (honen bidez datuak eguneratuko dira, erabiltzailearen azken 30. entrenamenduak hartuz), dataren bidez egindako entrenamenduak kontsultatu (hasierako eta bukaerako data bat jarritz denbora tarte horretan egindako entrenamenduak adieraziko dira), datuen eguneraketa (aldaketak ariketa batean egon badira datubasea eguneratu), datubasean gordeta dauden datuak lista batean bistaratu (alde batetik lista bistaratuko du eta bestetik bilaketa egiteko aukera) eta bueltak bistaratzea (buektak taula batean bistaratuko ditu).

## 1.3 Proiektuaren arkitektura

Arkitekturaren aldetik, MVC software arkitektura patroia erabili egin dugu, hau da, inplementazioa banatu egin dugu Model, View (interfazea hemen egongo da) eta Controller (datu base kudeatzailea eta Strava-ko Api kudeatzailea izango dituen) ataletan.

Honen bidez akatsak errezago zuzendu ahal dira eta txukuntasuna eta ordena bermatuko dira.

Beste aldetik, hainbat liburutegi erabili ditugu, eta hauen artean aurkitzen diren garrantzitsuenak hauek dira: datetime (ordua eta data formatu egokian adierazteko), sqlite3 (python-en SQL-ko kontsultak idazteko, irakurtzeko...), json (json formatuko datuak parseatzeko String edo fitxategietara irakurri ahal izateko), urllib3 (HTTP bezeroa), sys (interpreter-arekin lan egin ahal izateko oinarritzat), webbrowser (web orri baten edukia bistaratu ahal izateko), tkinter (python-en interfaze grafikoak sortzeko).

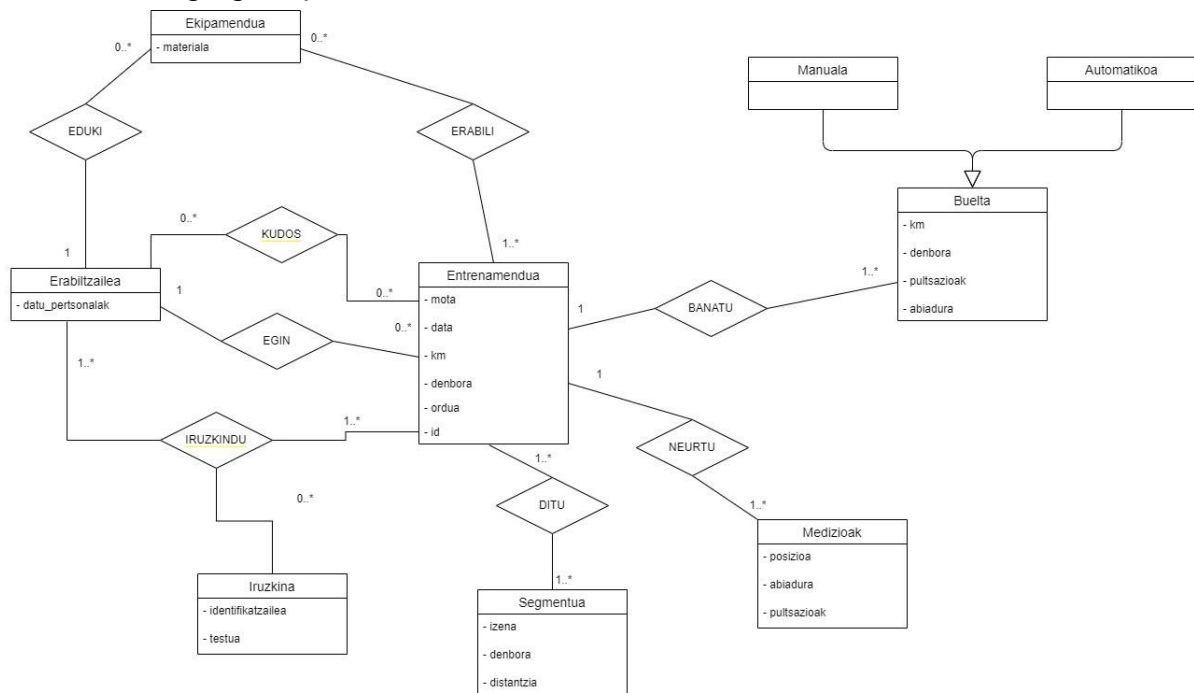
## 1.4. Plangintza

Lehenik eta behin domeinuaren eredua diseinatu genuen. Honen bidez, proiektuan zehar izango ditugun klase/entitate motak eta bakoitzaren atributu desberdinak ikusi ahalko ditugu eta honez gain haien arteko erlazioak ezarri. Erlazio horiek garrantzi handia izango dute datu basean datuak sartzerako orduan.

Behin hori diseinatuta, Strava-ko API-ak erabiliz erabiltzaile baten datuak lortzeko Python lengoaiari hainbat programa egin genituen (datu horiek tratatzeko eta filtratzeko). Hori lortu eta gero, datu basearen taulak sortu eta konexioa ezarri Python-eko programetatik lortutako datuak datu basera pasatzen joateko (bai datu berriak sartzeko, bai datuak eguneratzeko, bai datu-zerrenda zehatzak pantailaratzeko. Azken hauek lortzeko, interfaze bat eraiki beharko da ere).

## 2. Domeinuaren eredua

Hemen daukagu gure proiektuaren domeinuaren eredua:

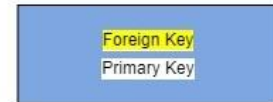


1. irudia: domeinu eredua

Eta hauek dira eredutik abiatuz sortutako taulak:

### Taulak:

Erabiltzailea(ErabiltzaileIz, Izena, Abizena, EkipMateriala)  
 Entrenamendua(ID\_Data\_Ordua, ErabiltzaileIz, Mota, Km, Denbora, SegmentuKod)  
 Buelta(Km, EntreData, EntreOrdua, Denbora, Pultsazioak, Abiadura)  
 Medizioak(Posizioa, EntreData, EntreOrdua, Abiadura, Pultsazioak, EntrenamenduID)  
 Kudos(ErabIz, EntreD, EntreO, EntrenamenduID)  
 Iruzkina(Identifikatzailea, Testua)  
 Iruzkindu(ErabIz, EntreData, EntreOrdua, Iruzkind, EntrenamenduID)  
 Ekipamendua(Materiala)  
 Erabili(MaterialaIzena, EntreD, EntreO, EntrenamenduID)  
 Segmentua(Izena, Denbora, Distantzia, EntrenamenduID)  
 Ditu(EntreData, EntreOrdua, SegmentuIzena, EntrenamenduID)



2. irudia: taulak

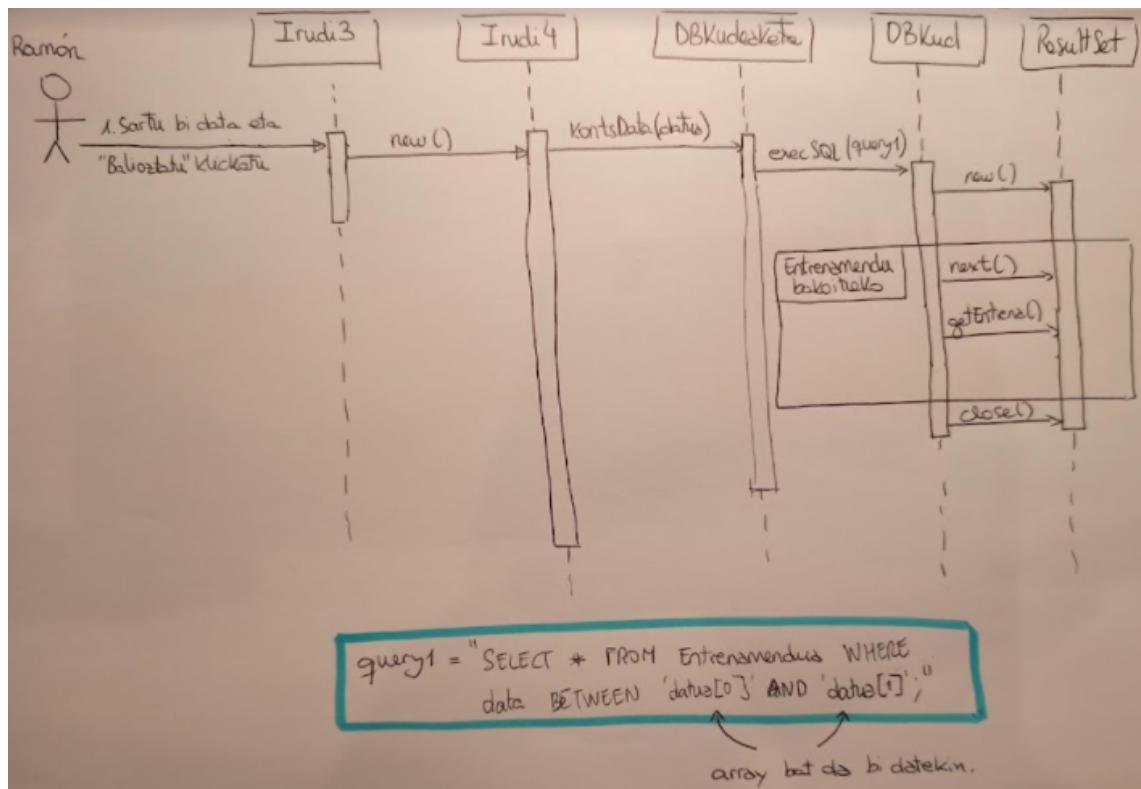
Ikusten denez, 7 entitate ditugu eta 8 erlazio. Entrenamendutik erlazioak beste entitate guztietara pasako dira. Izan ere, entrenamendu batean bueltak emango dira (hauek automatikoak edo manualak izan daitezke adierazten da goian txertatutako ereduan), erabiltzaile bat entrenamenduak egingo ditu eta ekipamendu bat izango du hau gauzatzeko. Hauetaz gain, entrenamenduek ere iruzkinak, segmentuak eta medizioak (heart rate, distantzia, denbora...) izango dituzte.

Taulei buruz hitz egiterakoan, benetan entitateak desberdinduko dituen datak dira. Izan ere, aktibitate bat hainbat aldiz errepikatu daiteke, baina ez du inoiz izango data berdina. Horrexegatik entitate eta erlazio askotan data izango da primary edo foreign key, edo biak (noski, ez dira egongo diren bakarrik, identifikatzaileak egongo baitira ere).

## 3. Diseinua

### 3.1. Sekuentzia diagrama

Aukeratu dugun funtzionalitatea sekuentzia diagrama egiteko erabiltzaileak hasierako eta bukaerako datak sartu ondoren data horien artean egindako entrenamenduen zerrenda itzultzea da. Hau izango litzateke gure sekuentzia diagrama kontsultarekin:



3. irudia: sekuentzia diagrama

## 4. Inplementazioa

Gure kasuan SQLite-ren eta DB Browser-en erabilera azpimarratu dezakegu. Izan ere, hauen bidez taulen sorkuntza askoz errazago egin dugu eta era zuzenean denbora galdu gabe. Honetaz aparte, esan behar da interfazearen erabilera erreza dela eta emaitzak argi ikusten direla era zuzenean, ez beste zabalgarri edo leiho batean.

### Tablas (9)

|                 |  |
|-----------------|--|
| > Buelta        | CREATE TABLE "Buelta" ( "km" INTEGER, "denbora" INTEGER, "pultsazioak" INTEGER, "abiadura" INTEGER, "entreDa   |
| > Ekipamendua   | CREATE TABLE "Ekipamendua" ( "materiala" TEXT NOT NULL, PRIMARY KEY("materiala") )                             |
| > Entrenamendua | CREATE TABLE "Entrenamendua" ( "ID" INTEGER, "mota" TEXT, "data" TEXT, "km" REAL, "denbora" TEXT, "ordua"      |
| > Erabiltzailea | CREATE TABLE "Erabiltzailea" ( "erabID" INTEGER, "izena" TEXT, "abizena" TEXT, "ekipamenduMat" TEXT, FOREIGN   |
| > Informazioa   | CREATE TABLE "Informazioa" ( "ActivityID" INTEGER, "Moving" INTEGER, PRIMARY KEY("ActivityID") )               |
| > Iruzkina      | CREATE TABLE "Iruzkina" ( "id" INTEGER NOT NULL, "testua" TEXT, PRIMARY KEY("id") )                            |
| > Kudos         | CREATE TABLE "Kudos" ( "ErabID" INTEGER NOT NULL, "EntrenaID" INTEGER NOT NULL, FOREIGN KEY("EntrenaID")       |
| > Medizioak     | CREATE TABLE "Medizioak" ( "posizioa" TEXT, "abiadura" REAL, "pultsazioak" REAL, "entreOrdua" TEXT, "entreData |
| > Segmentua     | CREATE TABLE "Segmentua" ( "izena" TEXT NOT NULL, "denbora" TEXT NOT NULL, "idEntrenamendua" INTEGER, "i       |

4 irudia: SQLite-ren erabilerak

## 5. Bibliografia

Gure baliabideak internet-etik atera ditugu web-orri desberdinetan. Ondorioz, hemen idatzi ditugunak ez dira ez liburuak ez artikuluak, eta horrexegatik ez dugu eGelan dagoen “proiektuaren dokumentazioa” pdf-an agertzen diren idazketa argibideak jarraitu, baizik eta zuzenean url-ak jarri eta zertarako erabili dugun bakoitza. Beraz, eGela-ko apunteetaz aparte, honako informazioa bilatu dugu:

Strava-ko api-aren dokumentazioa: <https://developers.strava.com/docs/reference/>  
SQLite-ko datu motak zeintzuk diren ikusteko: <https://www.sqlite.org/datatype3.html>  
SQLite-n DateTime nola jarri jakiteko: <https://www.sqlitetutorial.net/sqlite-date/>  
MVC nola erabili Python eta SQLite-rekin:  
<https://www.giacomodebidda.com/posts/mvc-pattern-in-python-sqlite/>