FOOTMATCH

Daniel Arellano Martínez
Bruno González Llaga
Carlos González Arenas
Diego Monsalves Vázquez
Víctor Manuel Vázquez García
Brian Calatrava Hurtado
Johnsiel Antonio Castaños Hernández
Jose María Cruz Lorite

Fundamentos de Ingeniería del Software para Sistemas Cloud

Índice

- Descripción de la aplicación
- Descomposición en microservicios
- Diseño de cada microservicio
- Aspectos relevantes
- Demo de la aplicación
- Conclusiones

Descripción de la aplicación

docker

okteto

django

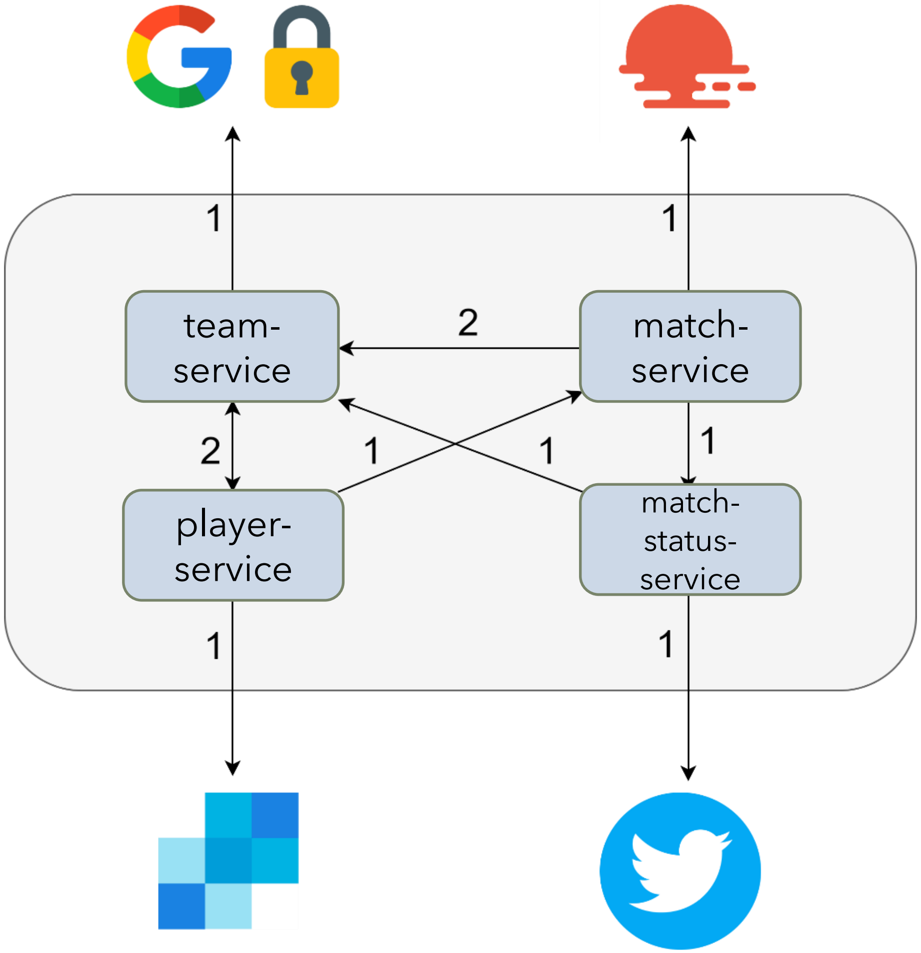
GitHub

Flask

web development,
one drop at a time

Descomposición en microservicios

footmatch -service



Diseño de cada microservicio

team-service

mongoDB®

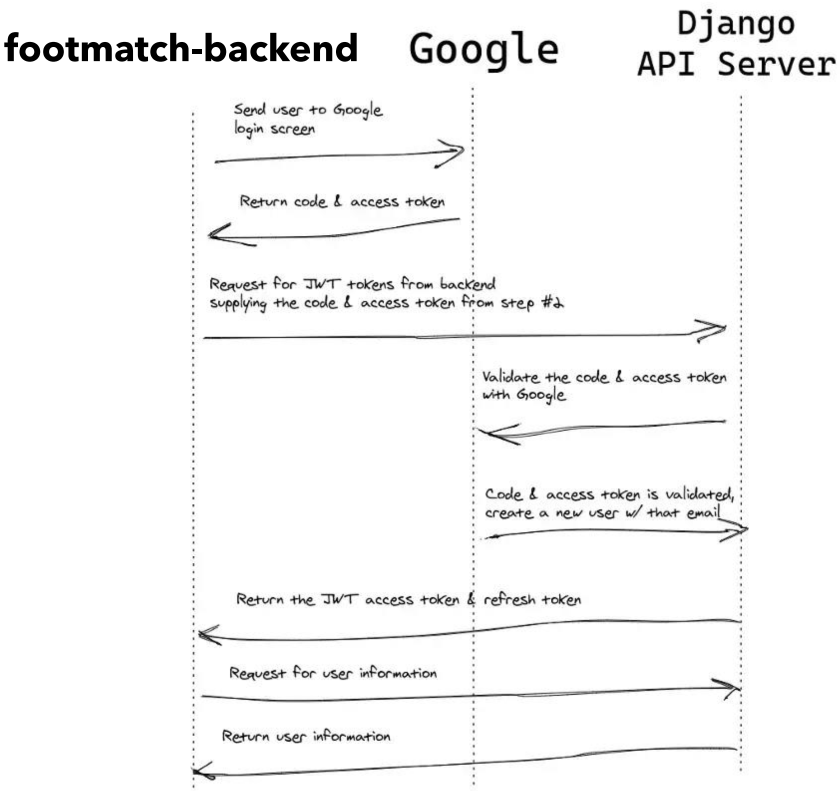
Google

django

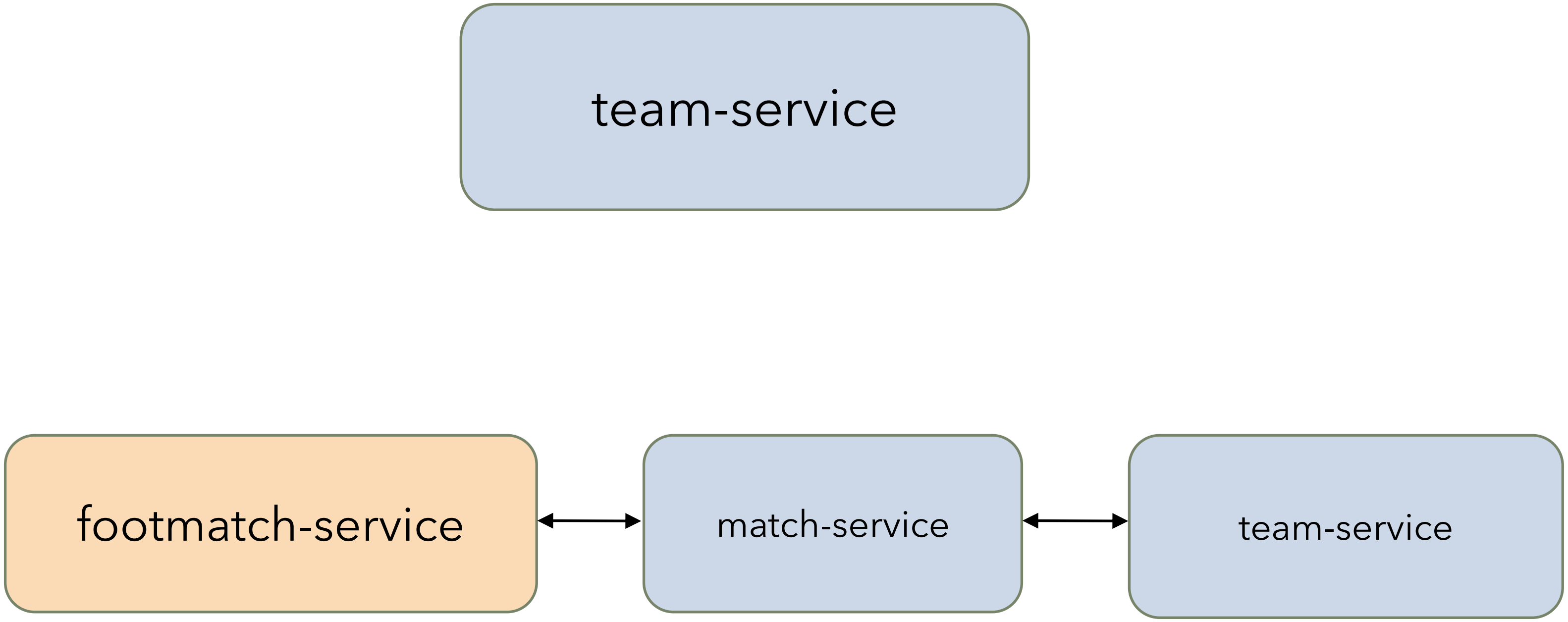
- Auth
- CRUD
- Feature toggles

Diseño de cada microservicio

team-service

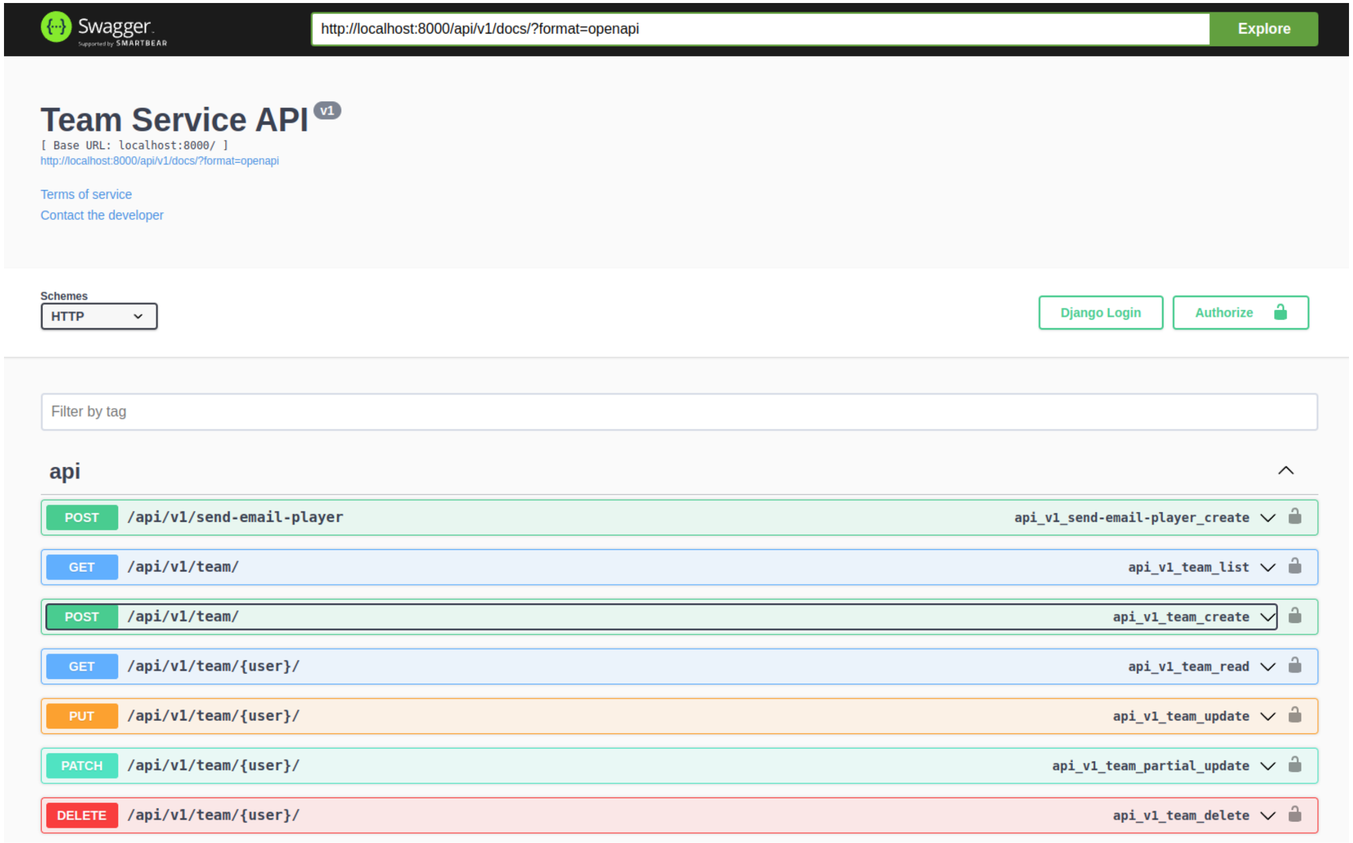

footmatch-backend
Google
Django API Server
Send user to Google login screen
Return code & access token
Request for JWT tokens from backend supplying the code & access token from step #2
Validate the code & access token with Google
Code & access token is validated, create a new user w/ that email
Return the JWT access token & refresh token
Request for user information
Return user information

Diseño de cada microservicio



Diseño de cada microservicio

team-service



http://localhost:8000/api/v1/docs/?format=openapi

Team Service API v1

[Base URL: localhost:8000/]
http://localhost:8000/api/v1/docs/?format=openapi

Terms of service
Contact the developer

api

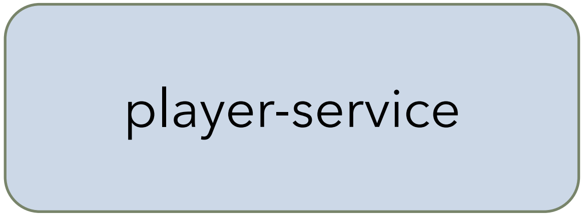
```
POST    /api/v1/send-email-player     api_v1_send-email-player_create
GET     /api/v1/team/                 api_v1_team_list
POST    /api/v1/team/                 api_v1_team_create
GET     /api/v1/team/{user}/          api_v1_team_read
PUT     /api/v1/team/{user}/          api_v1_team_update
PATCH   /api/v1/team/{user}/          api_v1_team_partial_update
DELETE  /api/v1/team/{user}/          api_v1_team_delete
```

Diseño de cada microservicio

team-service

redis

Diseño de cada microservicio



player-service

Diseño de cada microservicio

match-service

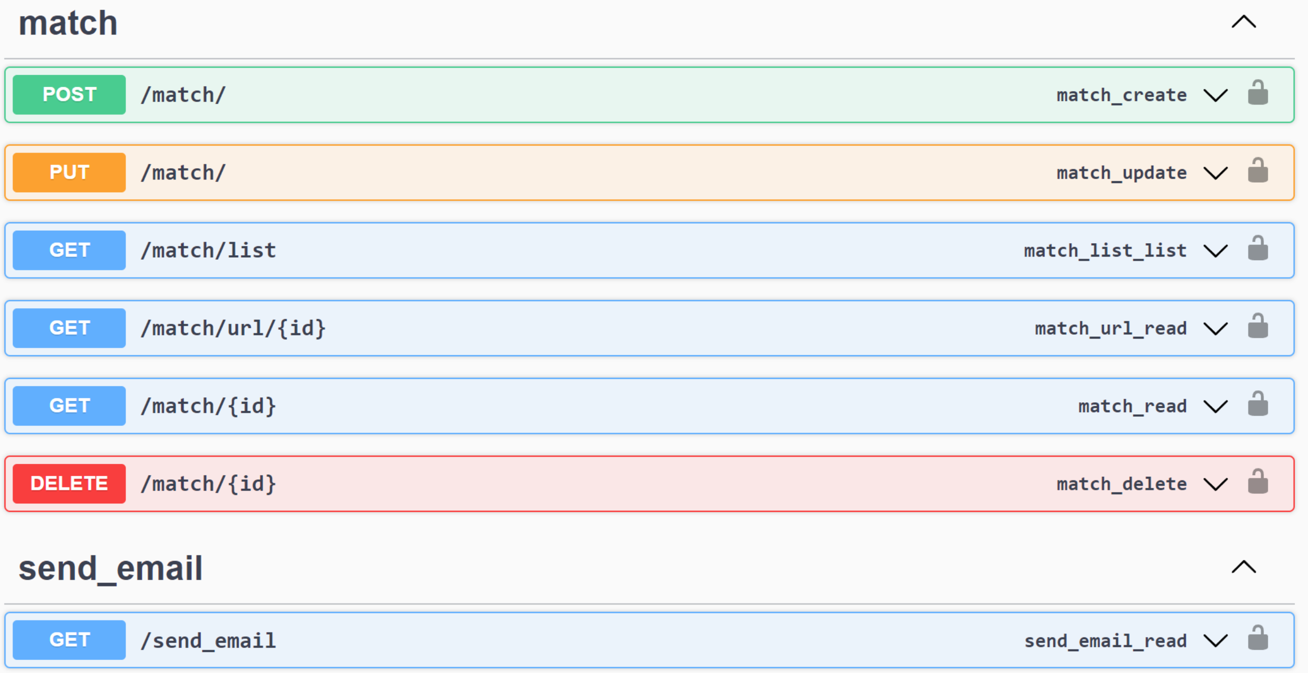
django

mongoDB®

- Caché local frontend
- CRUD
- Envío automatizado
- OpenWeather

Diseño de cada microservicio

match-service



match

POST /match/ match_create

PUT /match/ match_update

GET /match/list match_list_list

GET /match/url/{id} match_url_read

GET /match/{id} match_read

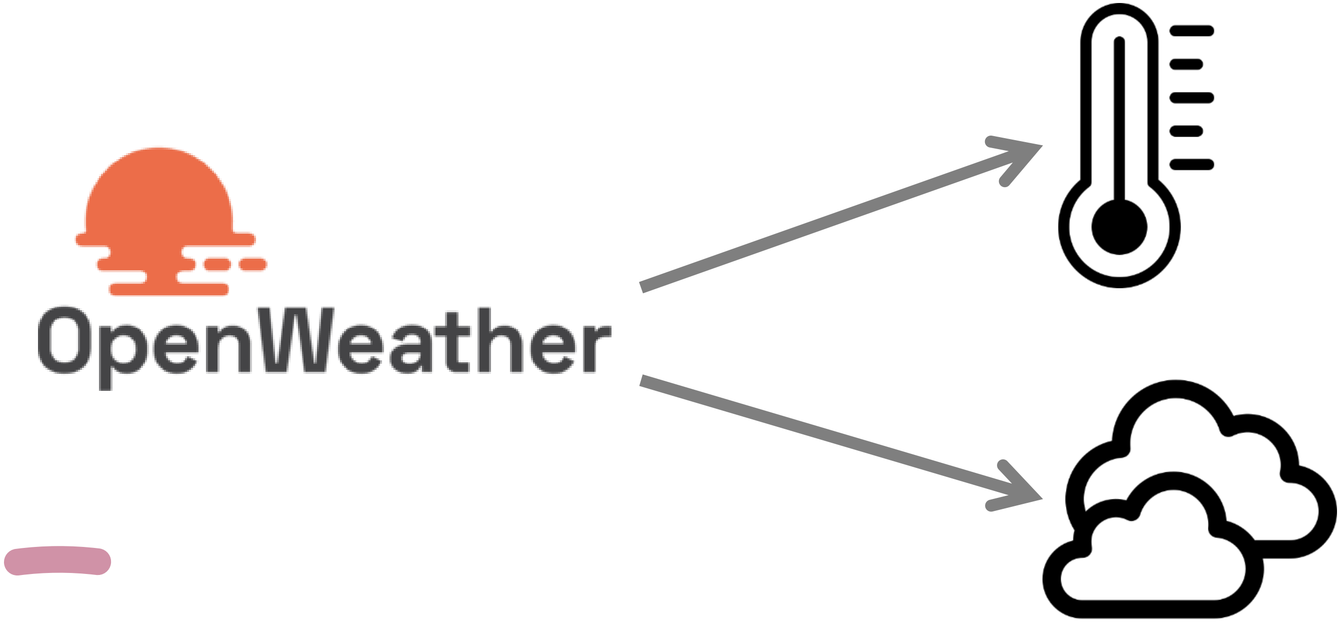
DELETE /match/{id} match_delete

send_email

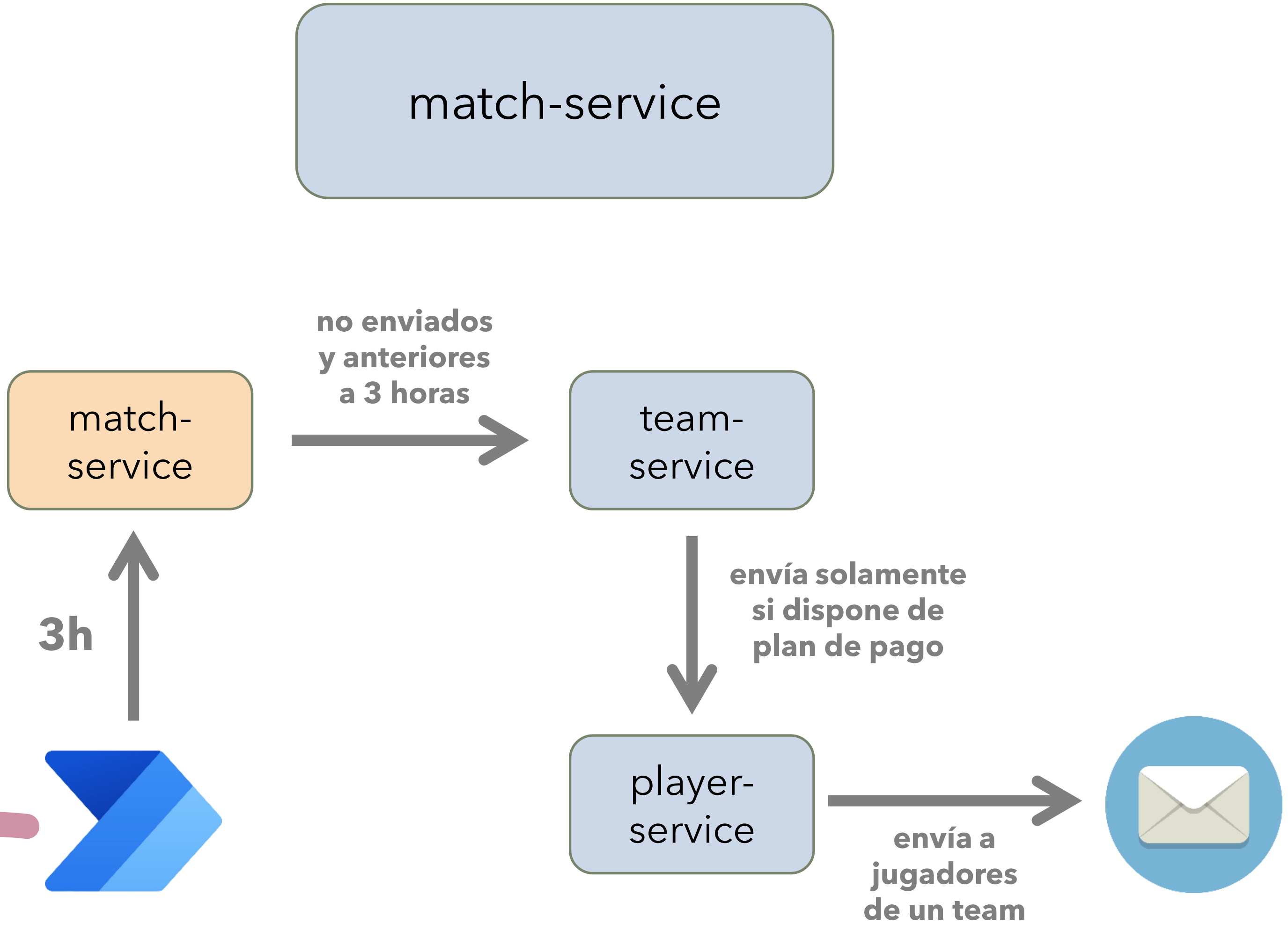
GET /send_email send_email_read

Diseño de cada microservicio

match-service



Diseño de cada microservicio



match-service

match-service

no enviados
y anteriores
a 3 horas

team-service

envía solamente
si dispone de
plan de pago

player-service

envía a
jugadores
de un team

3h

Diseño de cada microservicio

match-status-service

mongoDB®

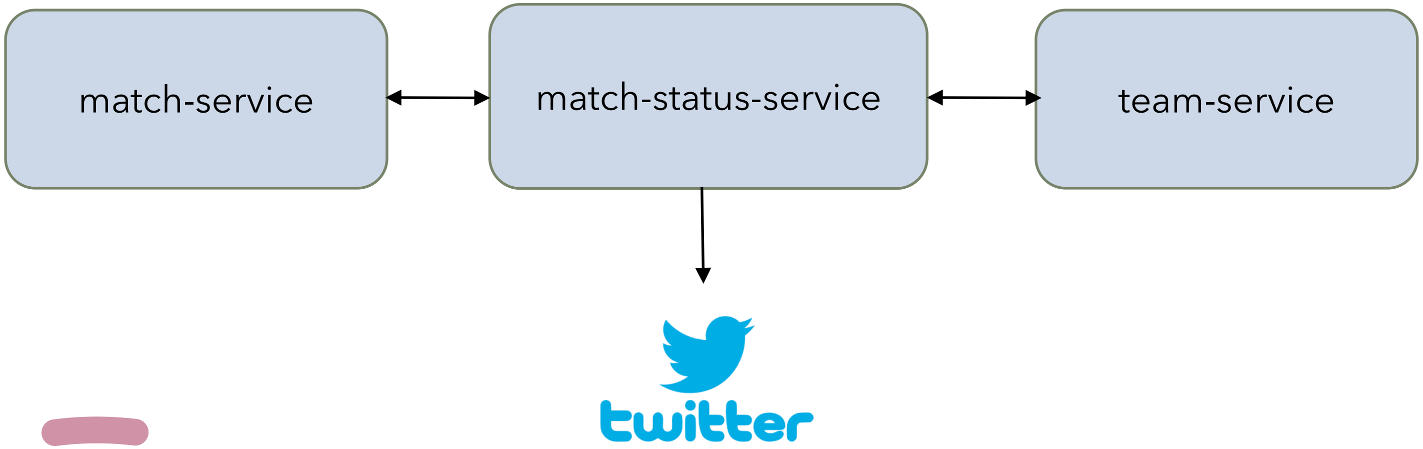
twitter

django

- Tweepy
  Posting tweets
- CRUD
- Feature toggles
- Auth

Diseño de cada microservicio

match-status-service



match-service

match-status-service

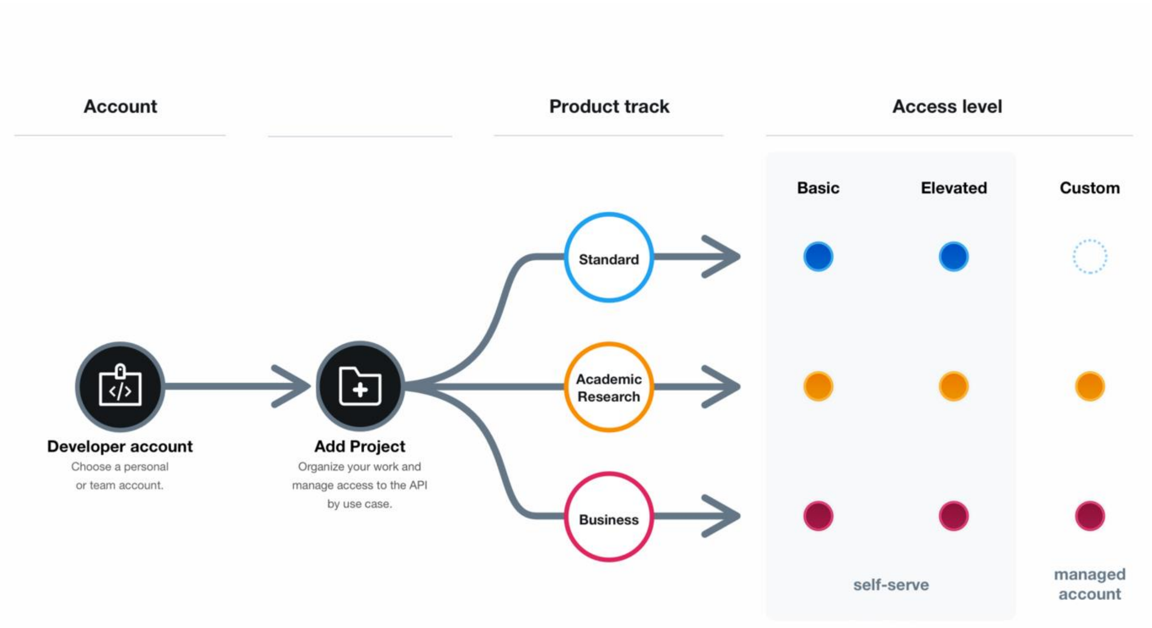
team-service

Diseño de cada microservicio

match-status-service

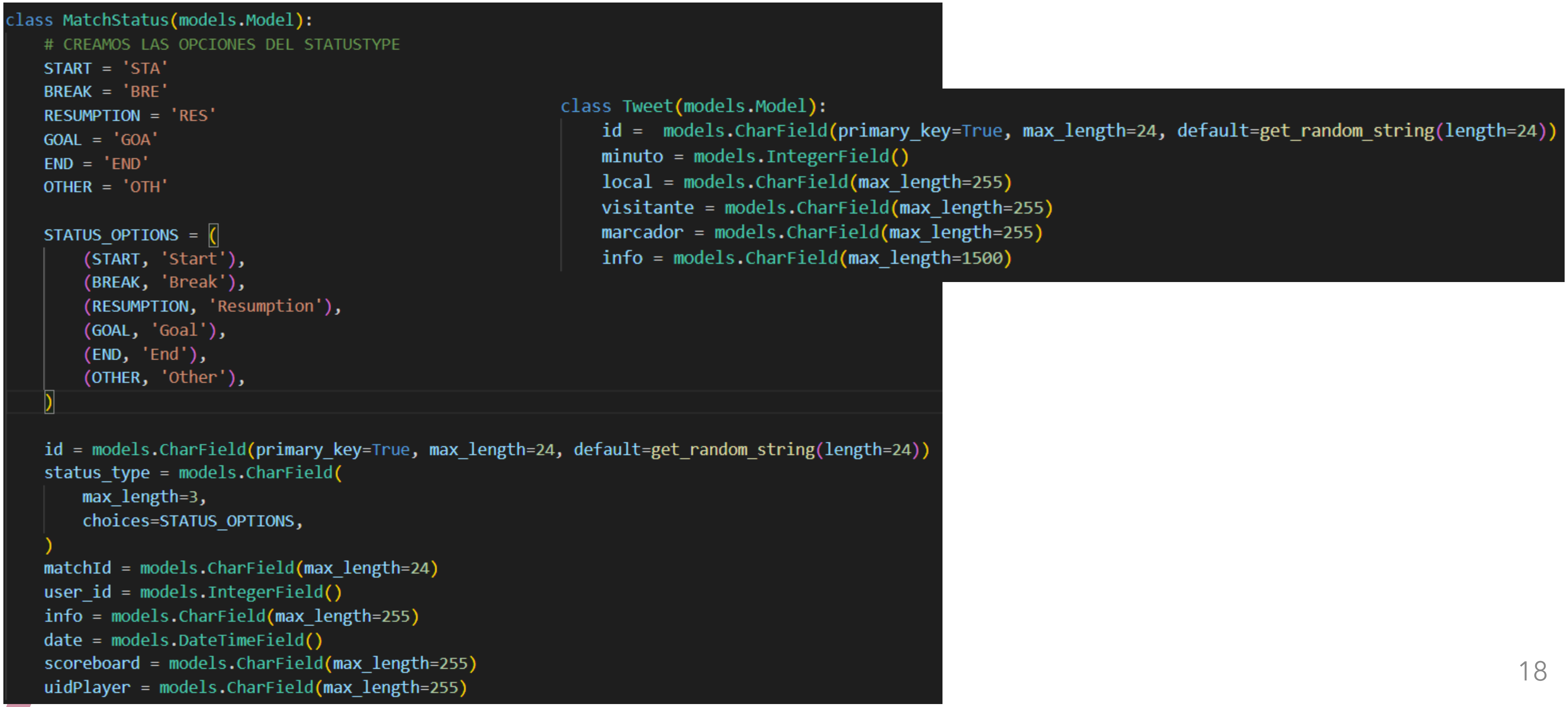
Tweepy

An easy-to-use Python library for accessing the Twitter API.



Diseño de cada microservicio

match-status-service

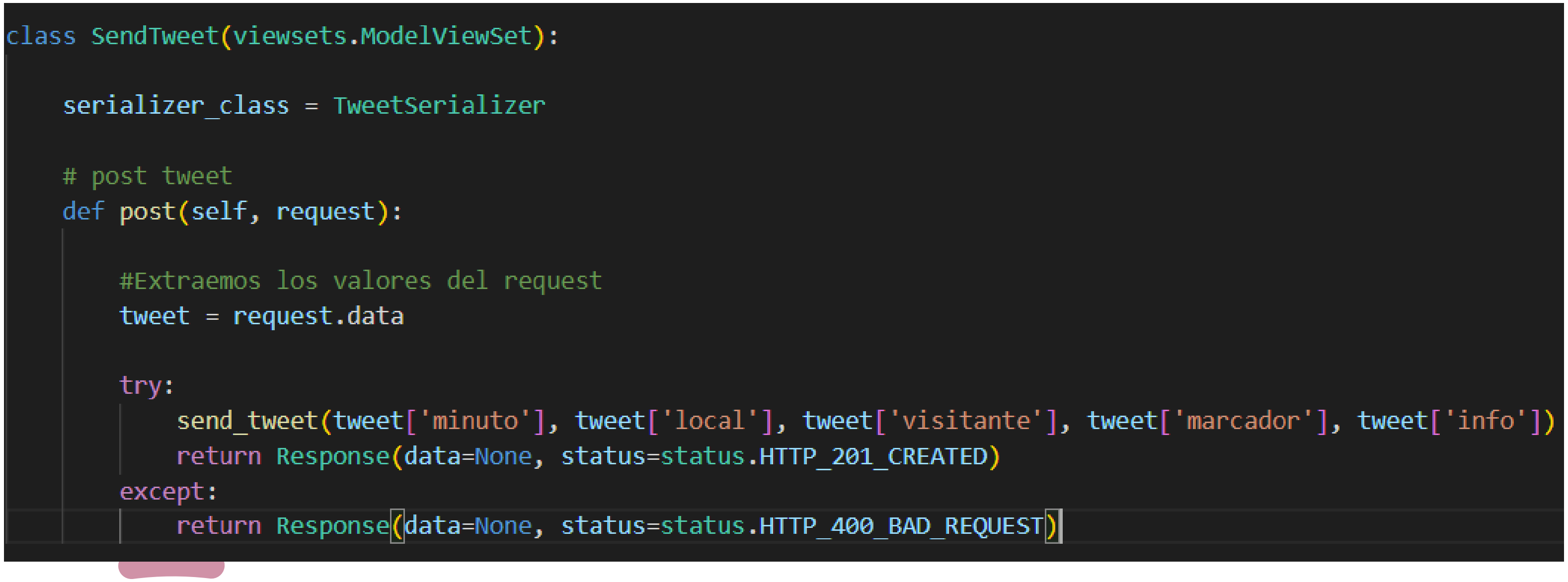


```
class MatchStatus(models.Model):
    # CREAMOS LAS OPCIONES DEL STATUSTYPE
    START = 'STA'
    BREAK = 'BRE'
    RESUMPTION = 'RES'
    GOAL = 'GOA'
    END = 'END'
    OTHER = 'OTH'

    STATUS_OPTIONS = (
        (START, 'Start'),
        (BREAK, 'Break'),
        (RESUMPTION, 'Resumption'),
        (GOAL, 'Goal'),
        (END, 'End'),
        (OTHER, 'Other'),
    )

    id = models.CharField(primary_key=True, max_length=24, default=get_random_string(length=24))
    status_type = models.CharField(
        max_length=3,
        choices=STATUS_OPTIONS,
    )
    matchId = models.CharField(max_length=24)
    user_id = models.IntegerField()
    info = models.CharField(max_length=255)
    date = models.DateTimeField()
    scoreboard = models.CharField(max_length=255)
    uidPlayer = models.CharField(max_length=255)
```

```
class Tweet(models.Model):
    id =  models.CharField(primary_key=True, max_length=24, default=get_random_string(length=24))
    minuto = models.IntegerField()
    local = models.CharField(max_length=255)
    visitante = models.CharField(max_length=255)
    marcador = models.CharField(max_length=255)
    info = models.CharField(max_length=1500)
```

Diseño de cada microservicio

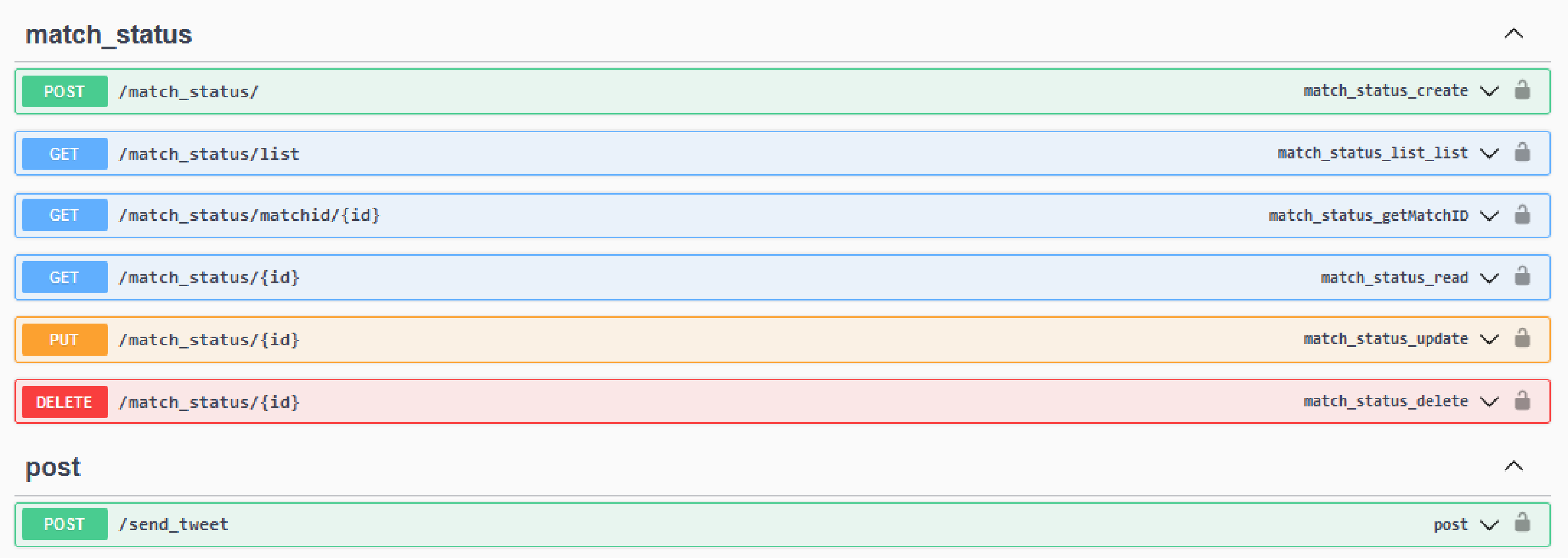
match-status-service



```
class SendTweet(viewsets.ModelViewSet):

    serializer_class = TweetSerializer

    # post tweet
    def post(self, request):

        #Extraemos los valores del request
        tweet = request.data

        try:
            send_tweet(tweet['minuto'], tweet['local'], tweet['visitante'], tweet['marcador'], tweet['info'])
            return Response(data=None, status=status.HTTP_201_CREATED)
        except:
            return Response(data=None, status=status.HTTP_400_BAD_REQUEST)
```

Diseño de cada microservicio

match-status-service



match_status

POST /match_status/ match_status_create

GET /match_status/list match_status_list_list

GET /match_status/matchid/{id} match_status_getMatchID

GET /match_status/{id} match_status_read

PUT /match_status/{id} match_status_update

DELETE /match_status/{id} match_status_delete

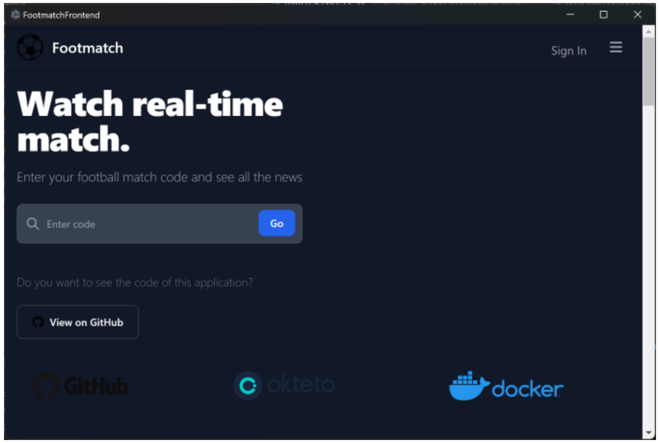
post

POST /send_tweet post

Aspectos relevantes

Aspectos relevantes

ELECTRON

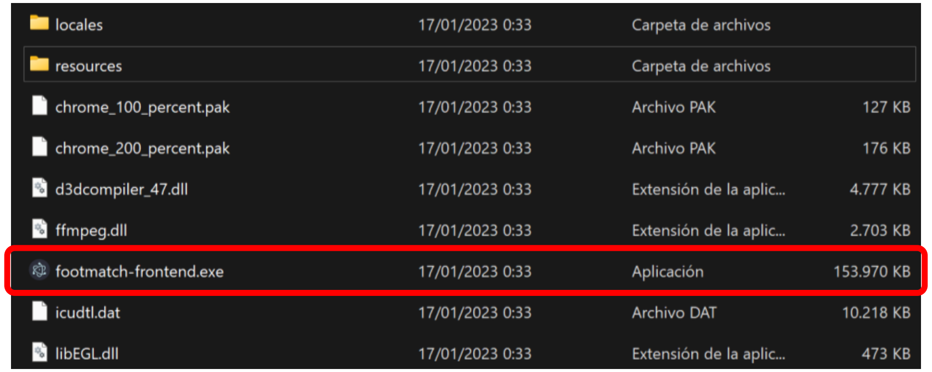


Watch real-time match.

Enter your football match code and see all the news

Do you want to see the code of this application?

npm run electron

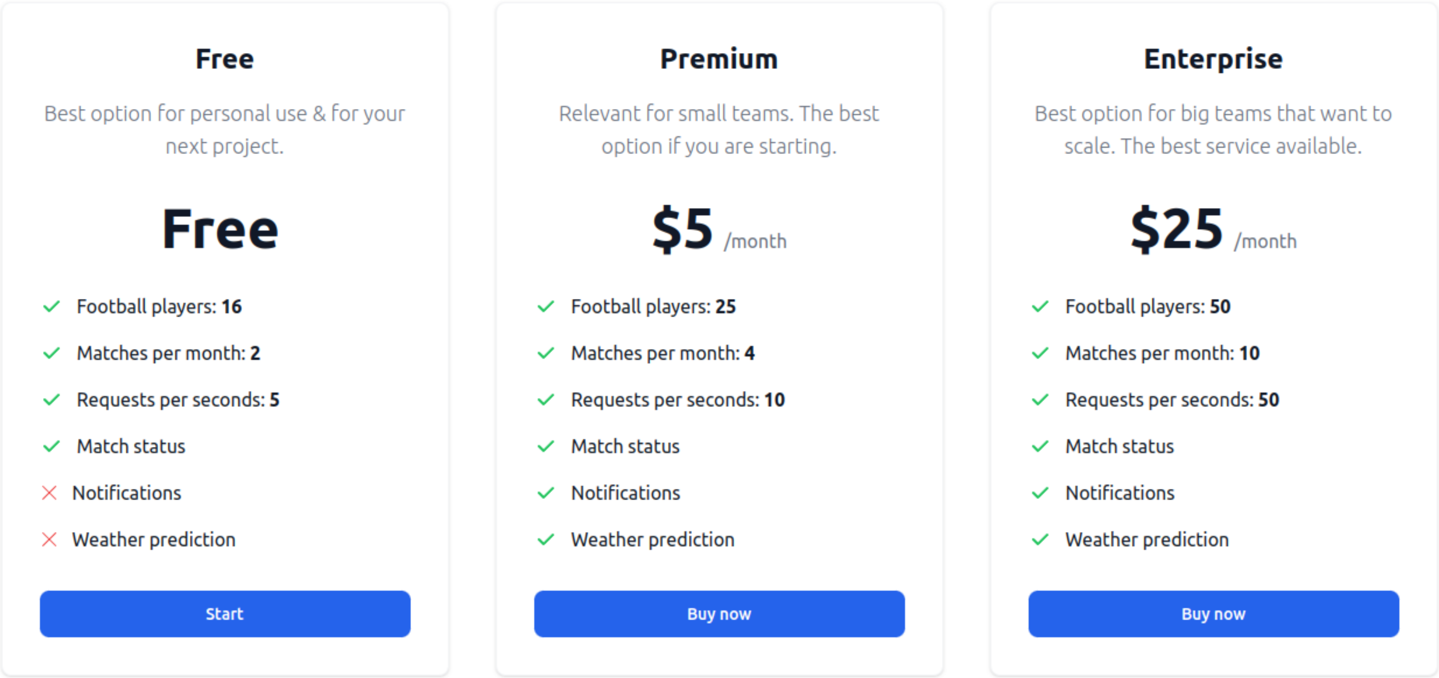


locales	17/01/2023 0:33	Carpeta de archivos	
resources	17/01/2023 0:33	Carpeta de archivos	
chrome_100_percent.pak	17/01/2023 0:33	Archivo PAK	127 KB
chrome_200_percent.pak	17/01/2023 0:33	Archivo PAK	176 KB
d3dcompiler_47.dll	17/01/2023 0:33	Extensión de la aplic...	4.777 KB
ffmpeg.dll	17/01/2023 0:33	Extensión de la aplic...	2.703 KB
footmatch-frontend.exe	17/01/2023 0:33	Aplicación	153.970 KB
icudtl.dat	17/01/2023 0:33	Archivo DAT	10.218 KB
libEGL.dll	17/01/2023 0:33	Extensión de la aplic...	473 KB

npm run electron-compile-win

Demo

Customer agreements



Free

Best option for personal use & for your next project.

Free

- ✓ Football players: **16**
- ✓ Matches per month: **2**
- ✓ Requests per seconds: **5**
- ✓ Match status
- ✗ Notifications
- ✗ Weather prediction

Premium

Relevant for small teams. The best option if you are starting.

$5 /month

- ✓ Football players: **25**
- ✓ Matches per month: **4**
- ✓ Requests per seconds: **10**
- ✓ Match status
- ✓ Notifications
- ✓ Weather prediction

Enterprise

Best option for big teams that want to scale. The best service available.

$25 /month

- ✓ Football players: **50**
- ✓ Matches per month: **10**
- ✓ Requests per seconds: **50**
- ✓ Match status
- ✓ Notifications
- ✓ Weather prediction

Customer agreements

Plan	Req/max	Precio	Nº usuarios
Free	5	0	150
Premium	10	5	60
Enterprise	50	25	20

Balance	355,05€
Ingresos	800€
Gastos	444€

API	Peticiones MS	Plan / €	Peticiones API / s	Riesgo	Peticiones / s
SSO	19	Free	10.000	85%	6697
OW	2	Pro / 410€	761	85%	705
TW	1	Free	1.000.000	0%	2350
SG	1	Essen. / 34€	896	65%	822

Conclusión

Preguntas