



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

FOOTMATCH

**Fundamentos de Ingeniería del
Software para Sistemas Cloud**

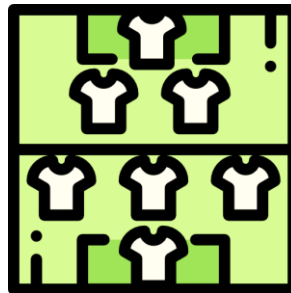
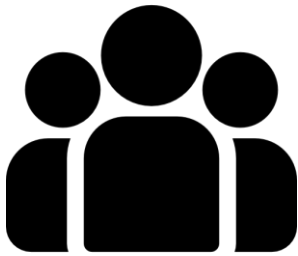


Índice

- Descripción de la aplicación
- Descomposición en microservicios
- Diseño de cada microservicio
- Aspectos relevantes
- Customer agreements
- Análisis de la capacidad
- Conclusiones

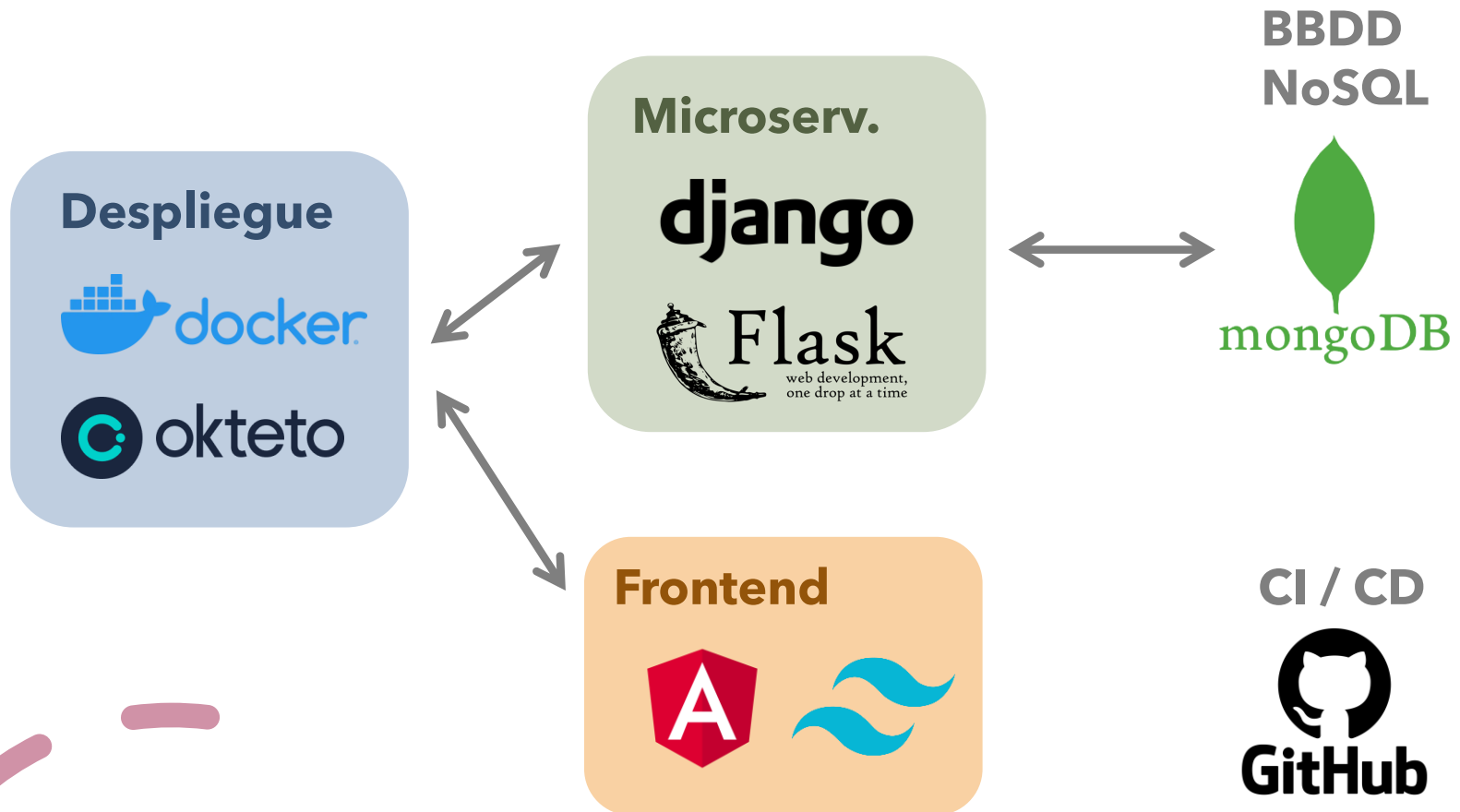
Descripción de la aplicación

Temática

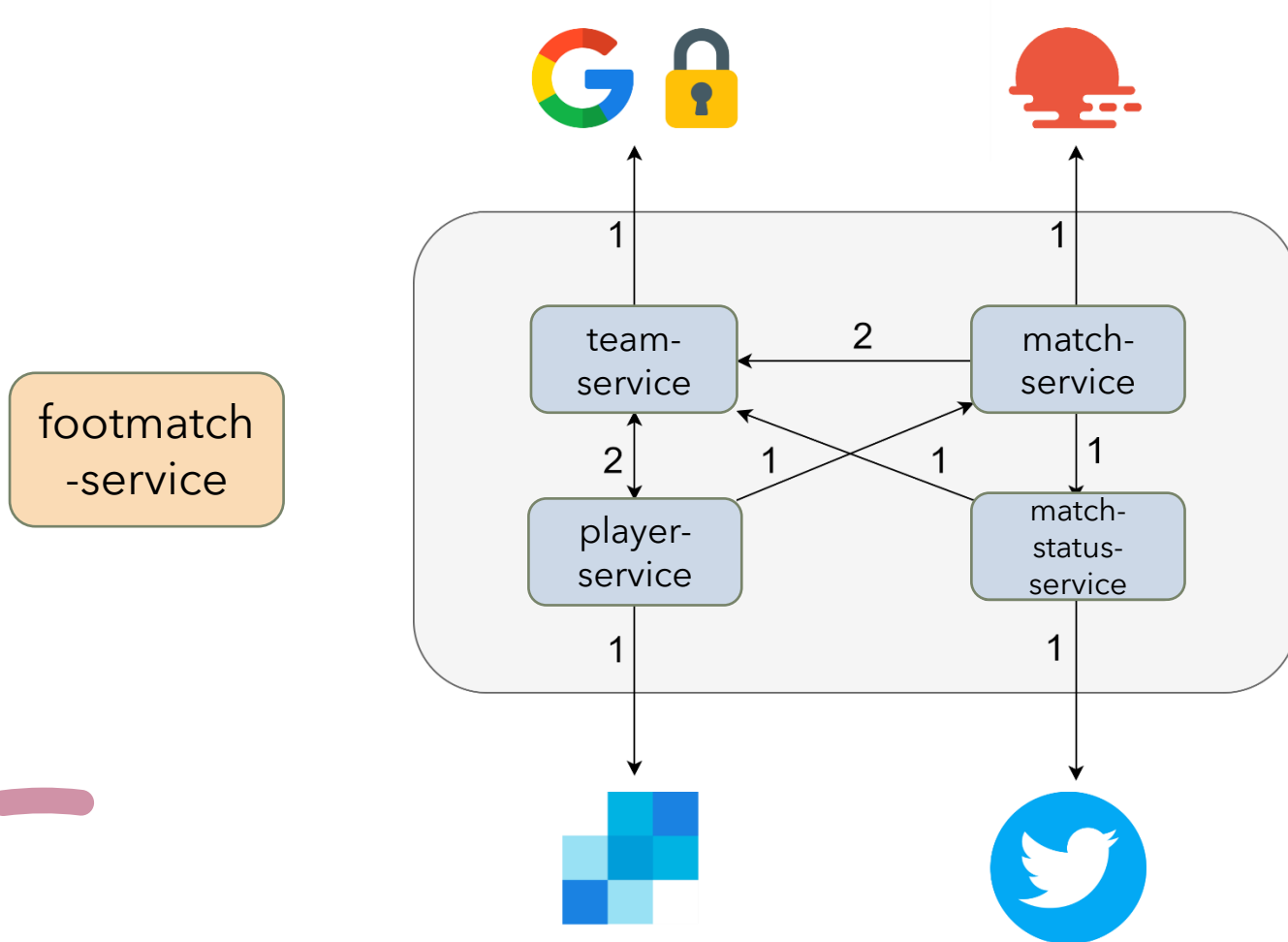


Descripción de la aplicación

Tecnologías



Descomposición en microservicios



Diseño de cada microservicio

team-service

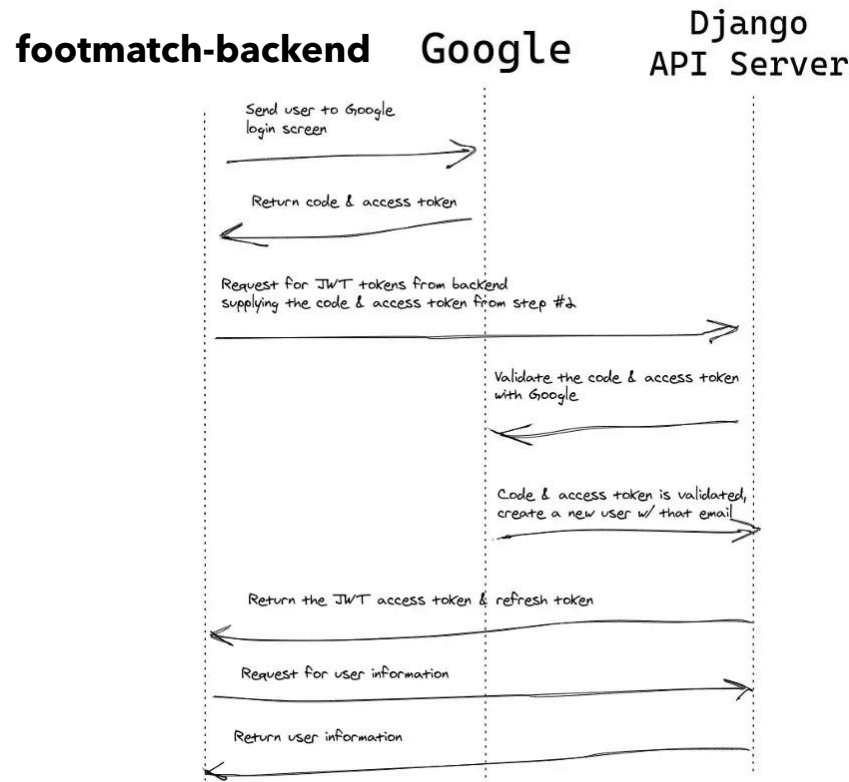


—
django

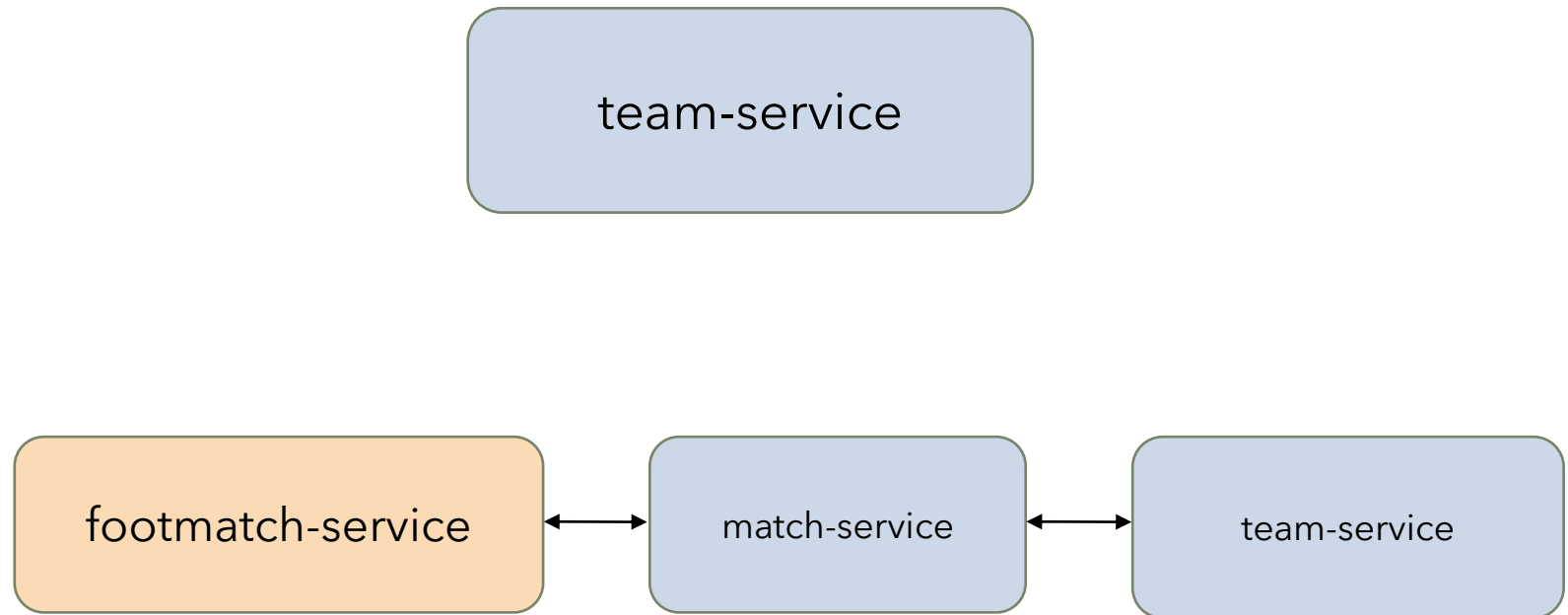
- Auth
- CRUD
- Feature toggles

Diseño de cada microservicio

team-service

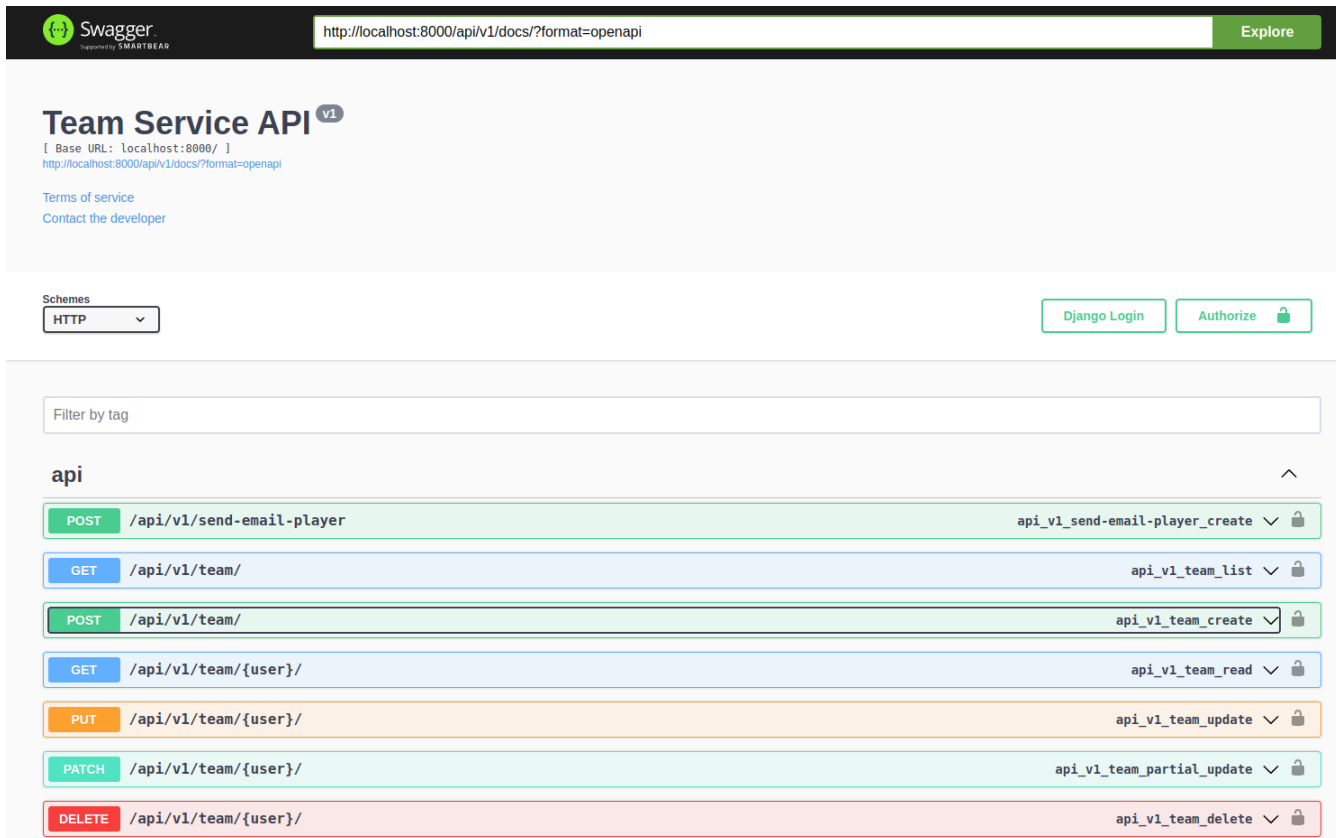


Diseño de cada microservicio



Diseño de cada microservicio

team-service



The image shows the Swagger UI for the Team Service API v1. The interface includes a Swagger logo, a search bar with the URL `http://localhost:8000/api/v1/docs/?format=openapi`, and an "Explore" button. Below the header, the title "Team Service API v1" is displayed, along with the base URL `localhost:8000/` and a link to the OpenAPI spec. There are links for "Terms of service" and "Contact the developer". A "Schemes" dropdown is set to "HTTP". On the right, there are buttons for "Django Login" and "Authorize". A "Filter by tag" input is present. The main content area lists several API endpoints under the "api" tag:

Method	Endpoint	Operation	Lock
POST	<code>/api/v1/send-email-player</code>	<code>api_v1_send-email-player_create</code>	Yes
GET	<code>/api/v1/team/</code>	<code>api_v1_team_list</code>	Yes
POST	<code>/api/v1/team/</code>	<code>api_v1_team_create</code>	Yes
GET	<code>/api/v1/team/{user}/</code>	<code>api_v1_team_read</code>	Yes
PUT	<code>/api/v1/team/{user}/</code>	<code>api_v1_team_update</code>	Yes
PATCH	<code>/api/v1/team/{user}/</code>	<code>api_v1_team_partial_update</code>	Yes
DELETE	<code>/api/v1/team/{user}/</code>	<code>api_v1_team_delete</code>	Yes

Diseño de cada microservicio

team-service



Diseño de cada microservicio

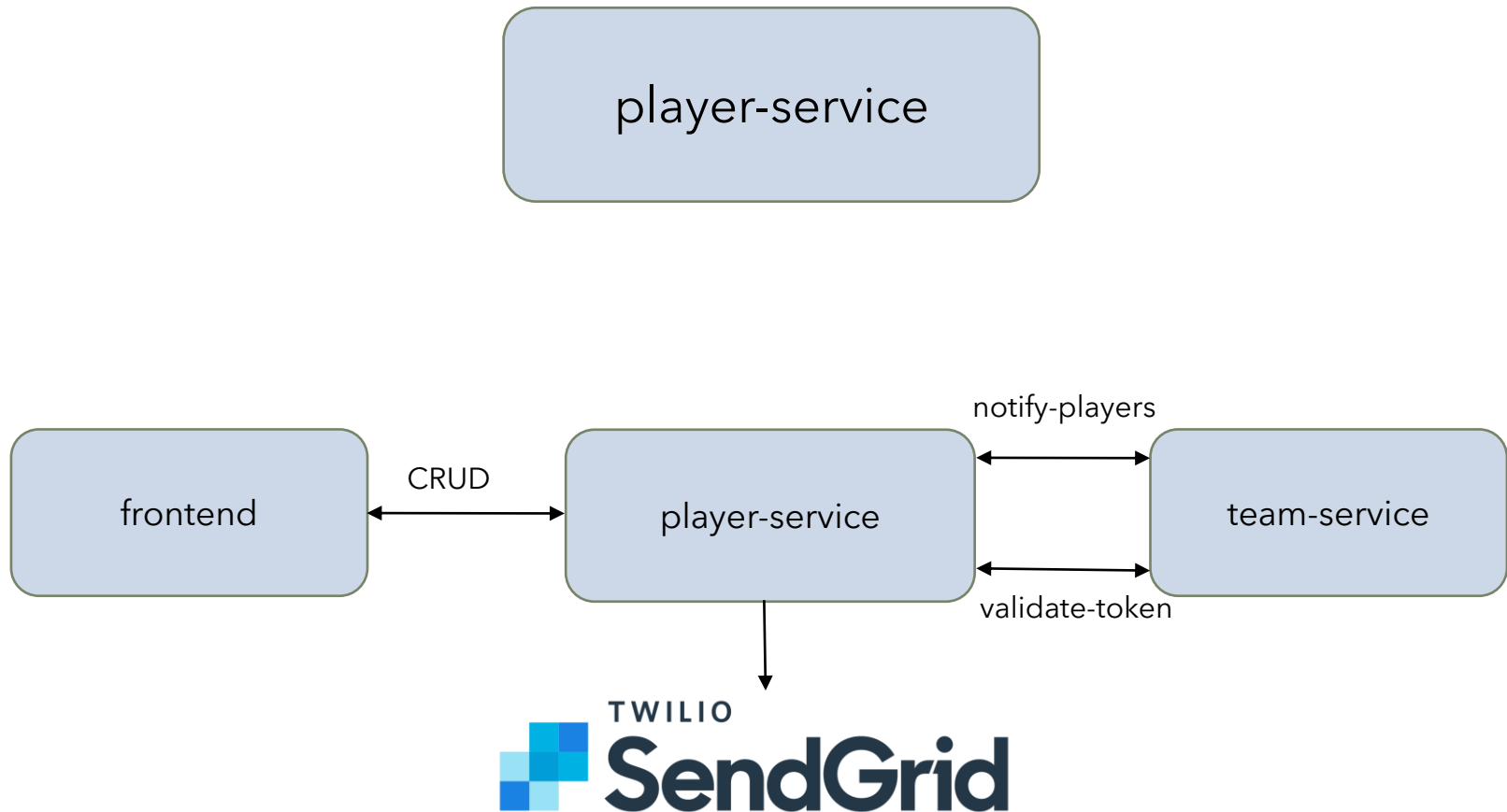
player-service



- Email sending
- **CRUD**



Diseño de cada microservicio



Diseño de cada microservicio

player-service

- PATCH** Update an alert
- GET** Retrieve all alerts
- ▼ Mail
 - POST** Send
- ▼ Blocks
 - GET** Retrieve a specific block
 - GET** Retrieve all blocks
 - DELETE** Delete a specific block
- > Bounces

▼ Request Body

Send email payload. For optional payload parameters visit https://sendgrid.com/docs/API_Reference/Web_API_v3/Mail/index.html#Request-Body-Parameters

Example_1 ▼

Body Schema

```
1 {
2   "personalizations": [
3     {
4       "to": [
5         {
6           "email": "john@example.com"
7         }
8       ],
9       "subject": "Hello, World!"
10    }
11  ],
12  "from": {
13    "email": "from_address@example.com"
14  },
15  "content": [
16    {
```

Diseño de cada microservicio

player-service

default

POST /api/v1/notify-players/{team_id} Send messages to players.

POST /api/v1/player Post a new player.

DELETE /api/v1/player/{id} Delete a player.

GET /api/v1/player/{id} Get a player from player id.

PUT /api/v1/player/{id} Modify a player.

GET /api/v1/players Get list of players. If team_id is not specified, caller team_id will be used.

GET /api/v1/players/{team_id} Get list of players. If team_id is not specified, caller team_id will be used.

Models

MailRequest >

Player >

Team >

Diseño de cada microservicio

match-service

django



- **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}

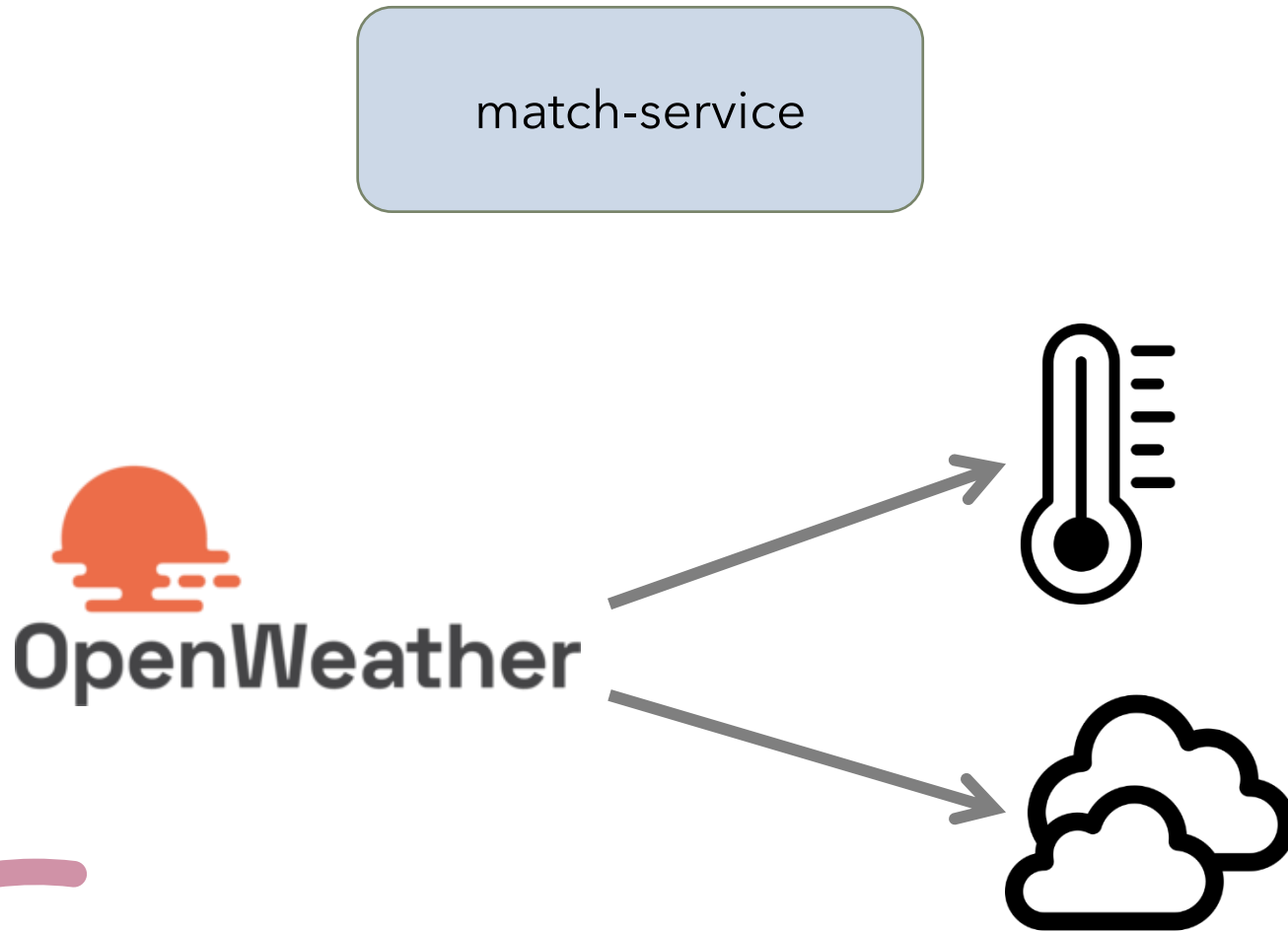
match_delete ▼ 🔒

send_email

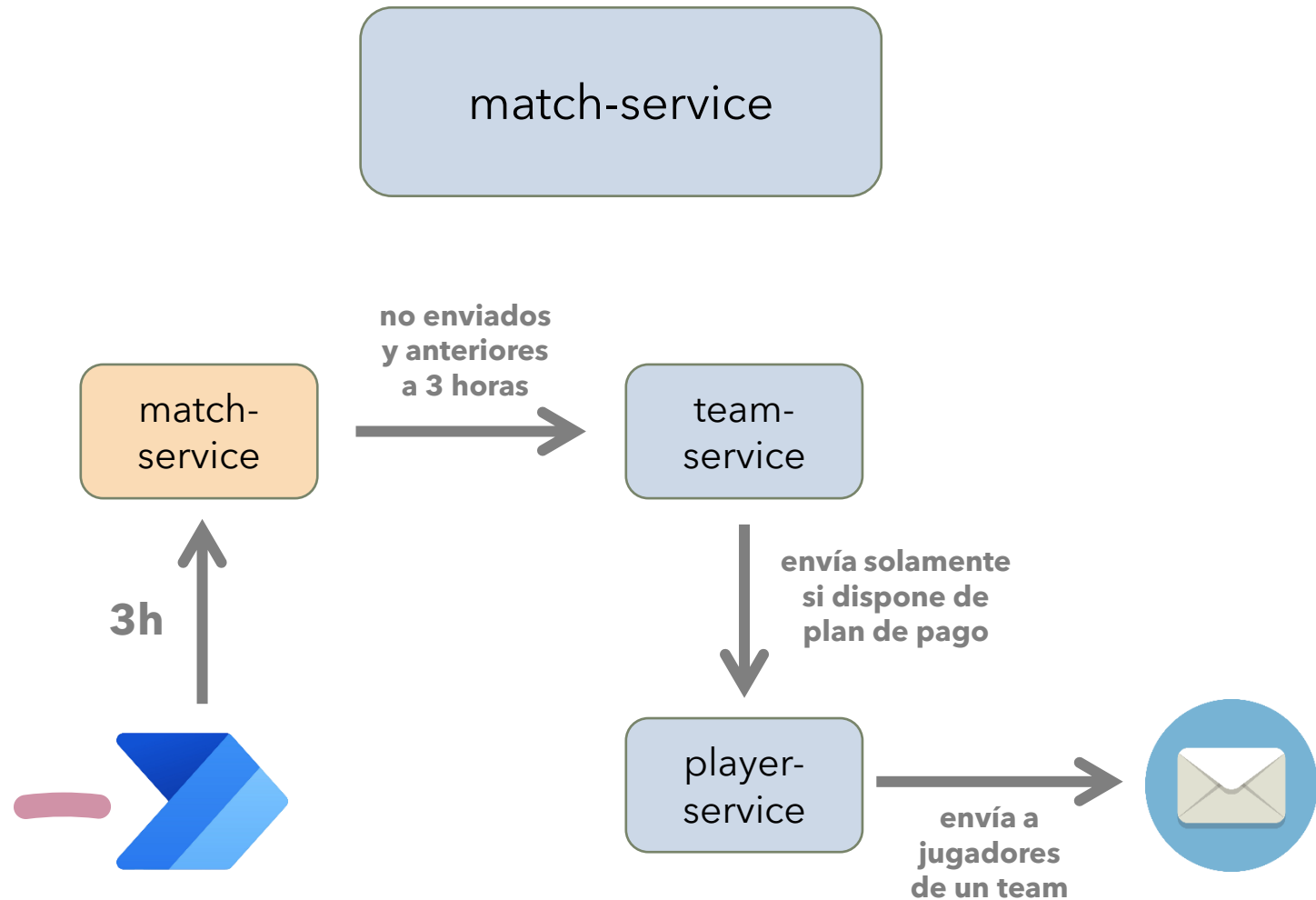
GET /send_email

send_email_read ▼ 🔒

Diseño de cada microservicio



Diseño de cada microservicio



Diseño de cada microservicio

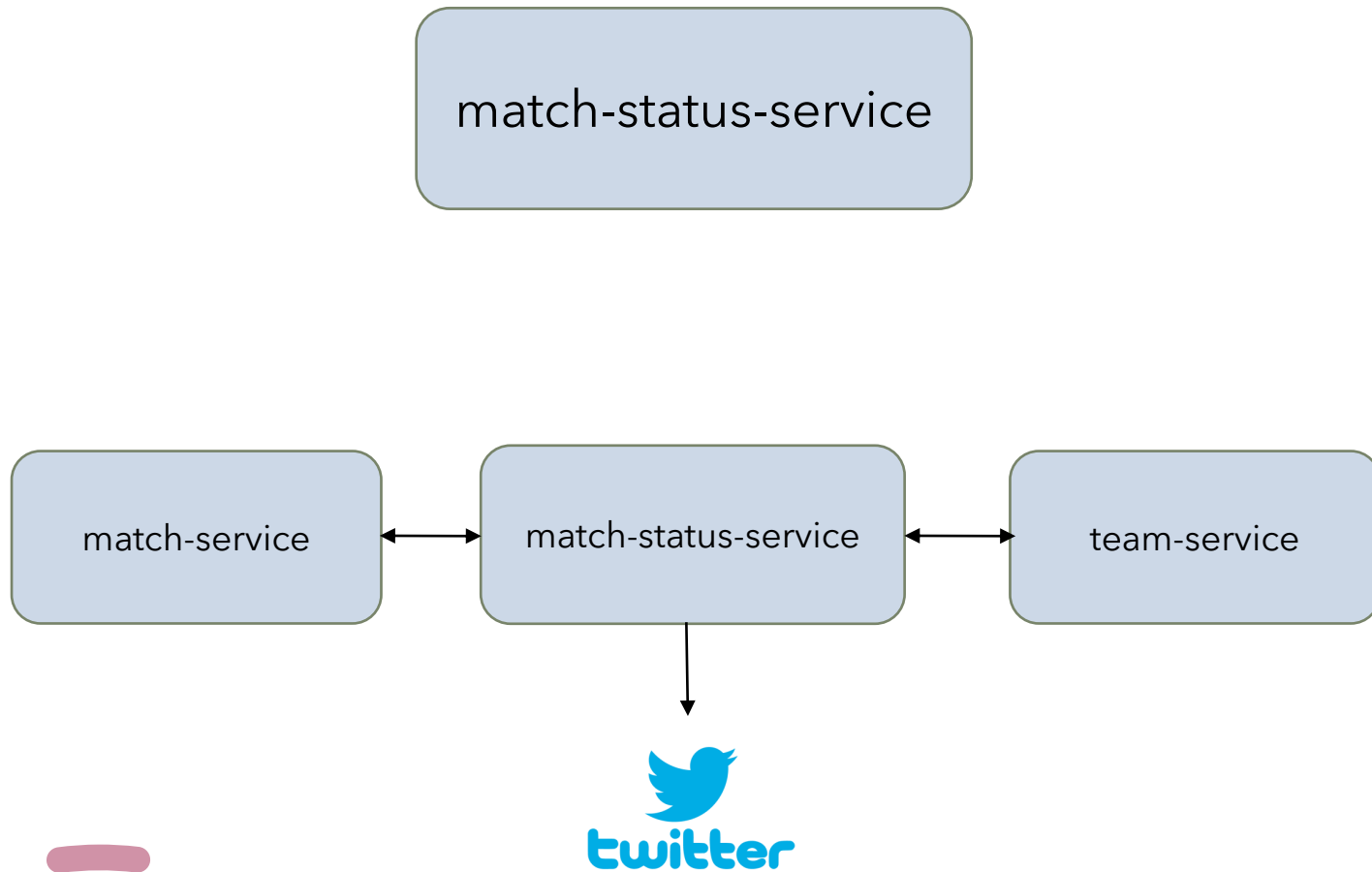
match-status-service



django

- Tweepy
Posting tweets
- CRUD
- Feature toggles
- Auth

Diseño de cada microservicio

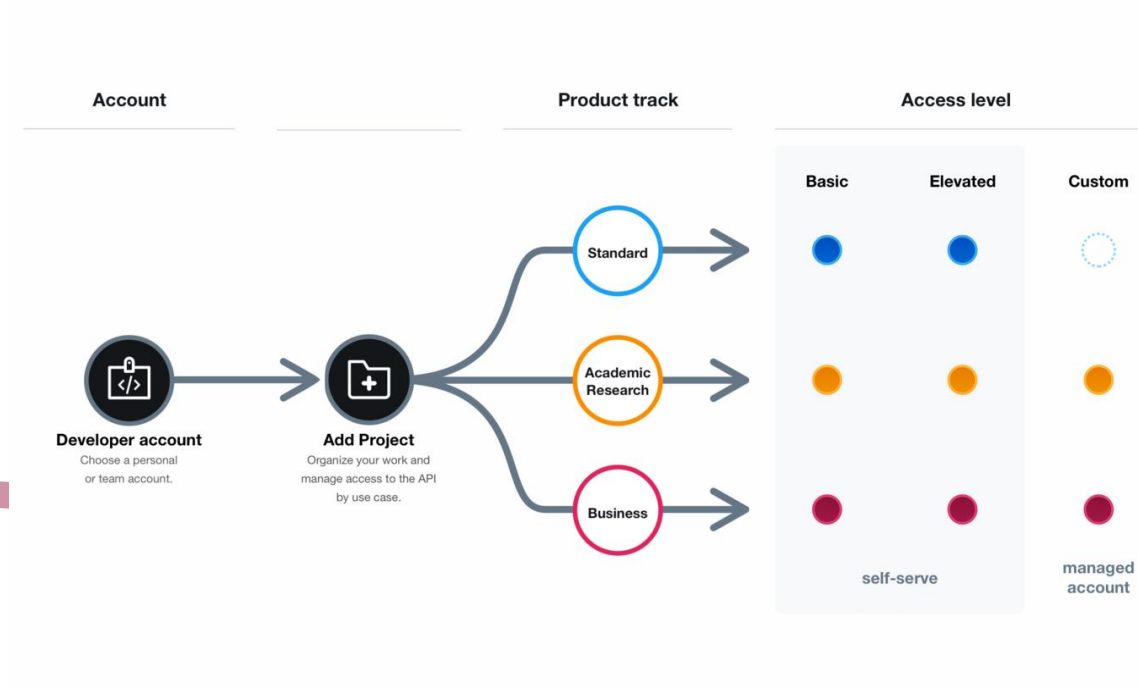


Diseño de cada microservicio

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

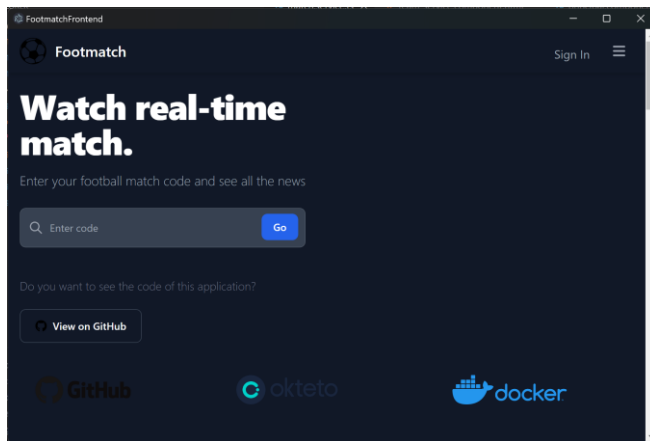
match-status-service

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



`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`

Customer agreements

Pricing

Free	Premium	Enterprise
Best option for personal use & for your next project.	Relevant for small teams. The best option if you are starting.	Best option for big teams that want to scale. The best service available.
Free	\$5 /month	\$25 /month
<ul style="list-style-type: none">✓ Football players: 16✓ Matches per month: 2✓ Requests per seconds: 5✓ Match status✗ Notifications✗ Weather prediction	<ul style="list-style-type: none">✓ Football players: 25✓ Matches per month: 4✓ Requests per seconds: 10✓ Match status✓ Notifications✓ Weather prediction	<ul style="list-style-type: none">✓ Football players: 50✓ Matches per month: 10✓ Requests per seconds: 50✓ Match status✓ Notifications✓ Weather prediction
Start	Buy now	Buy now

Customer agreements

Pricing

SLA

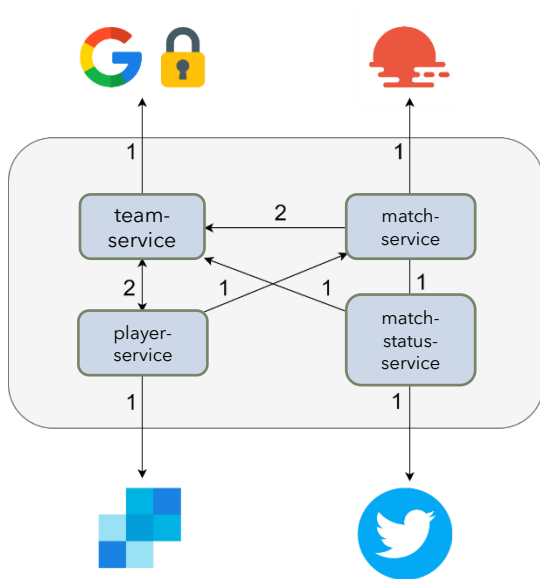
Free	Premium	Enterprise
Best option for personal use & for your next project.	Relevant for small teams. The best option if you are starting.	Best option for big teams that want to scale. The best service available.
Free	\$5 /month	\$25 /month
<ul style="list-style-type: none">✓ Football players: 16✓ Matches per month: 2✓ Requests per seconds: 5✓ Match status✗ Notifications✗ Weather prediction	<ul style="list-style-type: none">✓ Football players: 25✓ Matches per month: 4✓ Requests per seconds: 10✓ Match status✓ Notifications✓ Weather prediction	<ul style="list-style-type: none">✓ Football players: 50✓ Matches per month: 10✓ Requests per seconds: 50✓ Match status✓ Notifications✓ Weather prediction
Start	Buy now	Buy now

Customer agreements

SLA

Características	Free	Premium	Enterprise
Disponibilidad (tasa acierto por peticiones al mes)	N/A	95/100	99/100
Tipo contacto	N/A	email	sms / email
Tiempo resp.	N/A	3 días	1 día

Análisis de capacidad



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

Análisis de capacidad

Plan	Req/max Precio	N° 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