

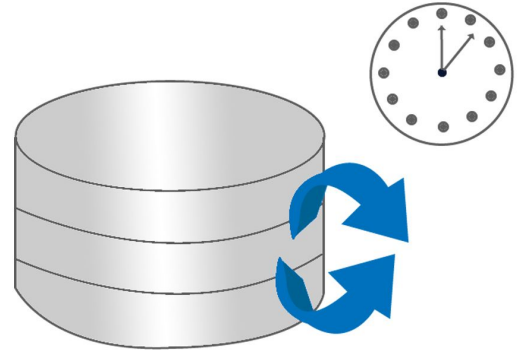
Persistencia y comunicaciones



Persistencia de Datos

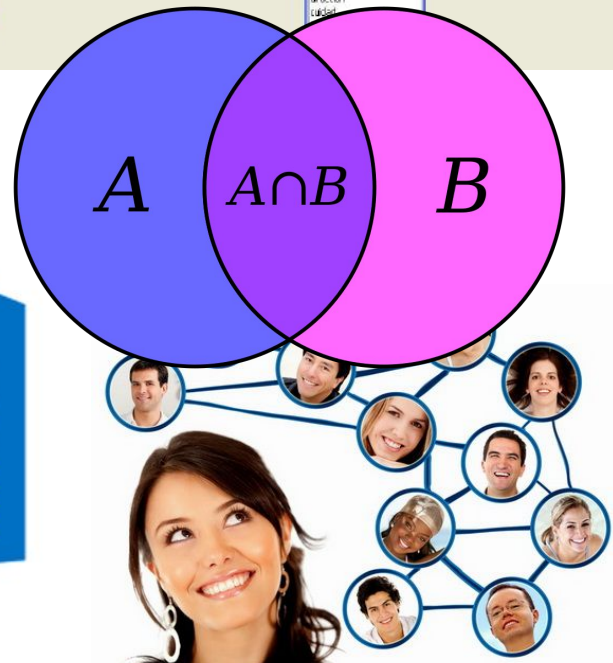
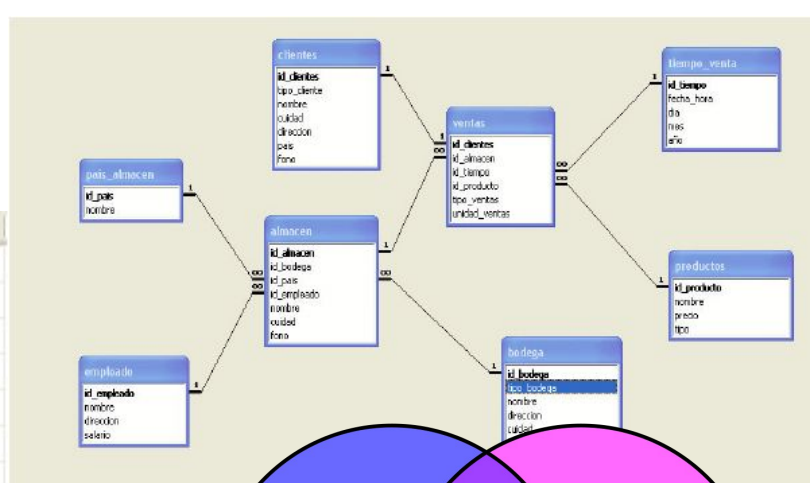
Mantener el valor de los datos procesados a través del tiempo.

- Registros físicos
- Estructuras de datos
- Archivos
- Bases de datos jerárquicas
- Bases de datos de red
- Bases de datos relacionales
- Bases de datos post-relacionales
 - Bases de datos orientadas a objetos



Bases de datos Relacionales

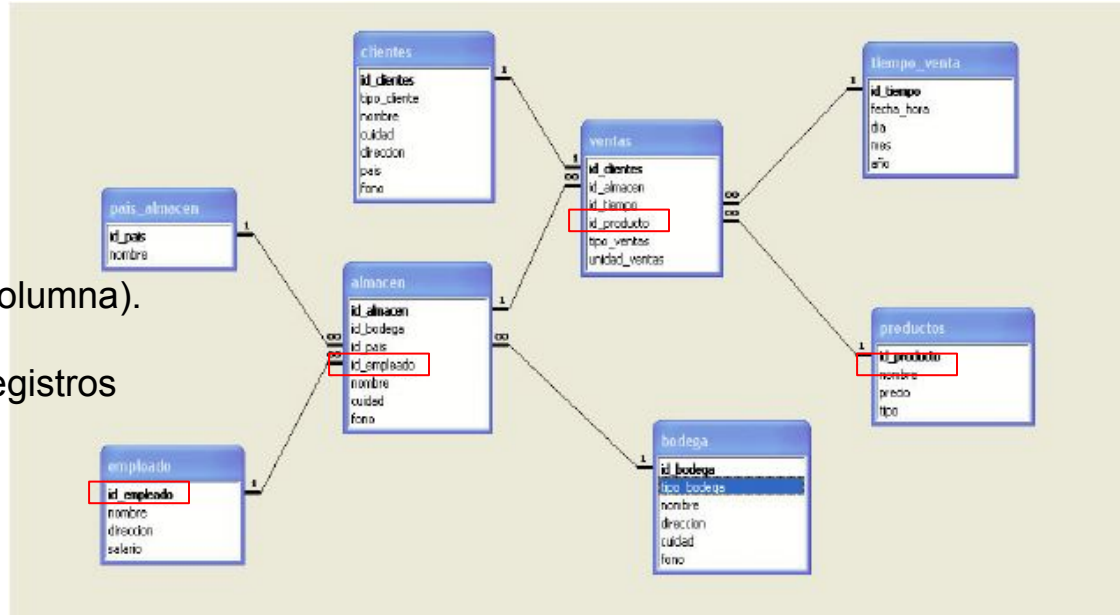
	FirstName	LastName	AddressLine1	City	StateProvinceCode	PostalCode
1	Ben	Miller	101 Candy Rd.	Redmond	WA	98052
2	Garrett	Vargas	10203 Acom Avenue	Calgary	AB	T2P 2G8
3	Gabe	Mares	1061 Buskrik Avenue	Edmonds	WA	98020
4	Reuben	D'sa	1064 Slow Creek Road	Seattle	WA	98104
5	Gordon	Hee	108 Lakeside Court	Bellevue	WA	98004
6	Karan	Khanna	1102 Ravenwood	Seattle	WA	98104
7	François	Ajenstat	1144 Paradise Ct.	Issaquah	WA	98027
8	Sariya	Harnpadoungsataya	1185 Dallas Drive	Everett	WA	98201
9	Kirk	Koenigsbauer	1220 Bradfr			
10	Kim	Ralls	1226 Shoe			
11	Michael	Raheem	1234 Seasi			
12	Mike	Seamans	1245 Clay F			
13	Reed	Koch	1275 West			
14	Fadi	Fakhouri	1285 Greer			
15	Paul	Singh	1343 Prosp			
16	Brenda	Diaz	1349 Steve			
17	Jack	Richins	1356 Grove			
18	John	Evans	136 Balboa			
19	Ken	Myer	1362 Some			
20	Barbara	Moreland	137 Mazatl			



Bases de datos Relacionales

Una Base de datos Relacional trabaja con:

- Tablas (Entidades)
 - Llave primaria
 - Propiedades (Atributos)
 - Cada atributo pasa a ser un campo (columna).
 - Las tablas contienen tuplas, son los registros
- Relaciones entre Entidades
 - Relaciones entre Tablas
 - Llaves foráneas
- Constraints





- Es una Biblioteca en C
- Funciona como manejador de Bases de Datos Relacionales
- La base de datos se crea y se resguarda en el dispositivo
- SQL

```
SELECT column_name(s)
FROM table_name
WHERE column_1 = value_1
    AND column_2 = value_2;
```

```
UPDATE table_name
SET some_column = some_value
WHERE some_column = some_value;
```

```
INSERT INTO table_name (column_1, column_2, column_3)
VALUES (value_1, 'value_2', value_3);
```

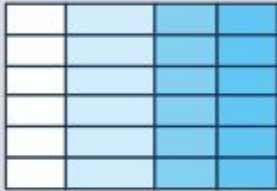
```
DELETE FROM table_name
WHERE some_column = some_value;
```

Bases de datos

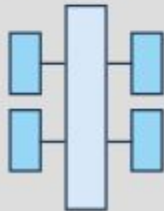
Post-Relacionales

SQL

Relational

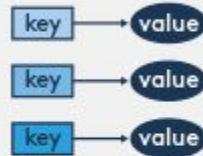


Analytical (OLAP)



NoSQL

Key-Value



Column-Family



Graph



Document



Core Data



- Es un framework basado en el modelo de bases de datos orientado a objetos
- Es un *Mecanismo de persistencia integrada*, no es una BD, NoSQL
- Va a trabajar dentro de la aplicación en un archivo *.xcdatamodeld* al que llamaremos **Modelo de Datos**
- Trabaja con Entidades - Objeto, a los que podremos asociar con Clases

Choose options for your new project:

Product Name:

Team:

Organization Name:

Organization Identifier:

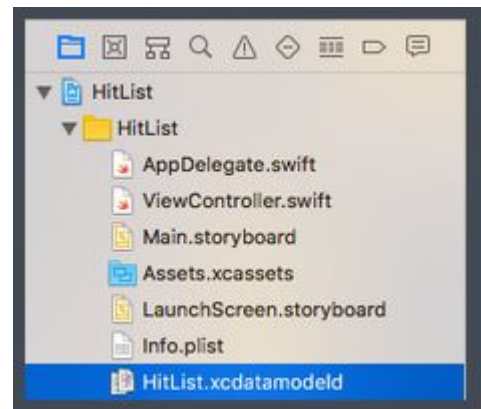
Bundle Identifier:

Language:

☒ Use Core Data

☐ Include Unit Tests

☐ Include UI Tests



ENTITIES

E Location

E Run

FETCH REQUESTS

CONFIGURATIONS

C Default

▼ Attributes

Attribute ^	Type	
N distance	Float	↕
N duration	Integer 16	↕
D timestamp	Date	↕
+ -		

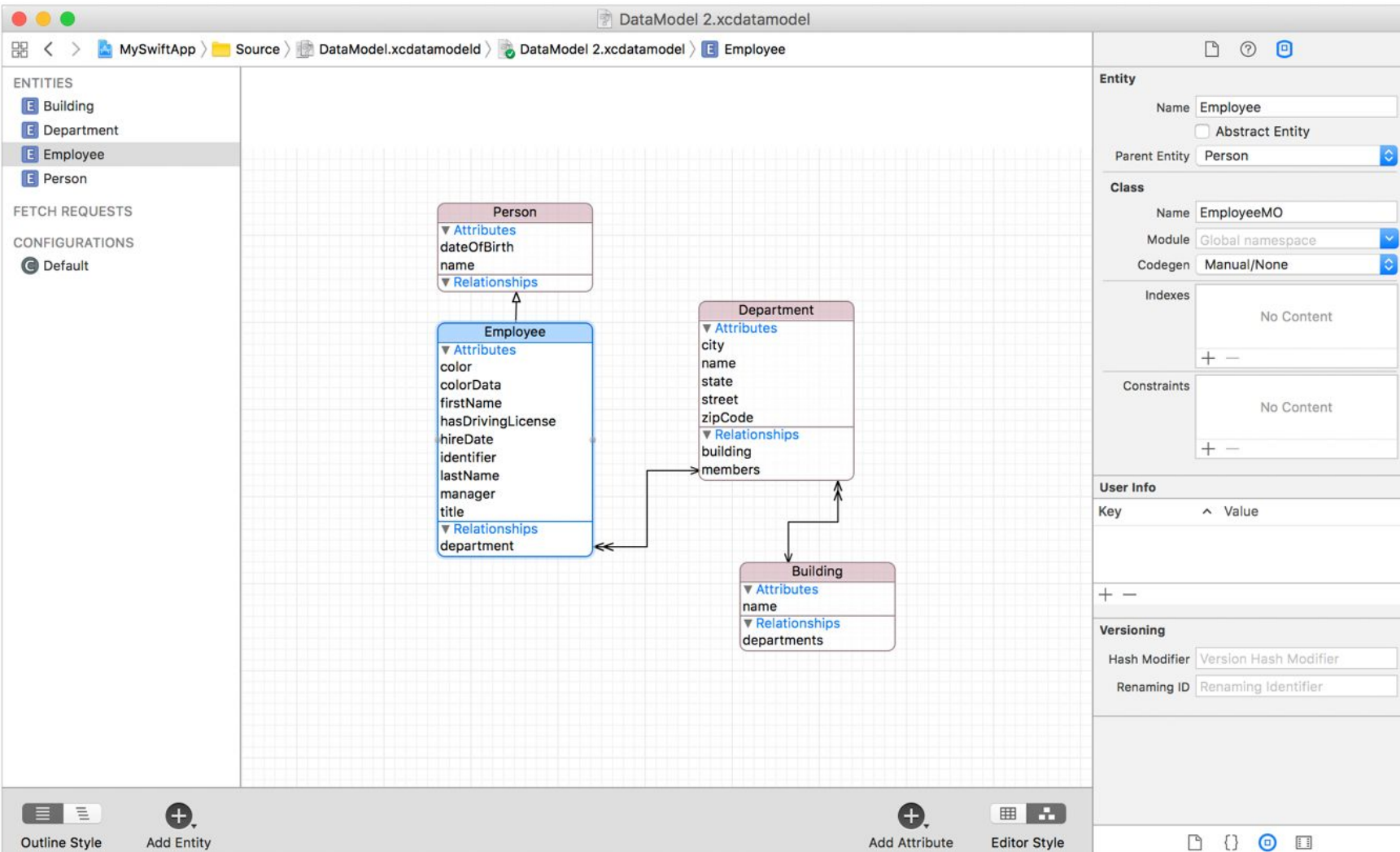
▼ Relationships

Relationship ^	Destination	Inverse
O locations	Location	↕ run ↕
+ -		

▼ Fetched Properties

Fetches Property ^	Predicate
+ -	

Not Applicable

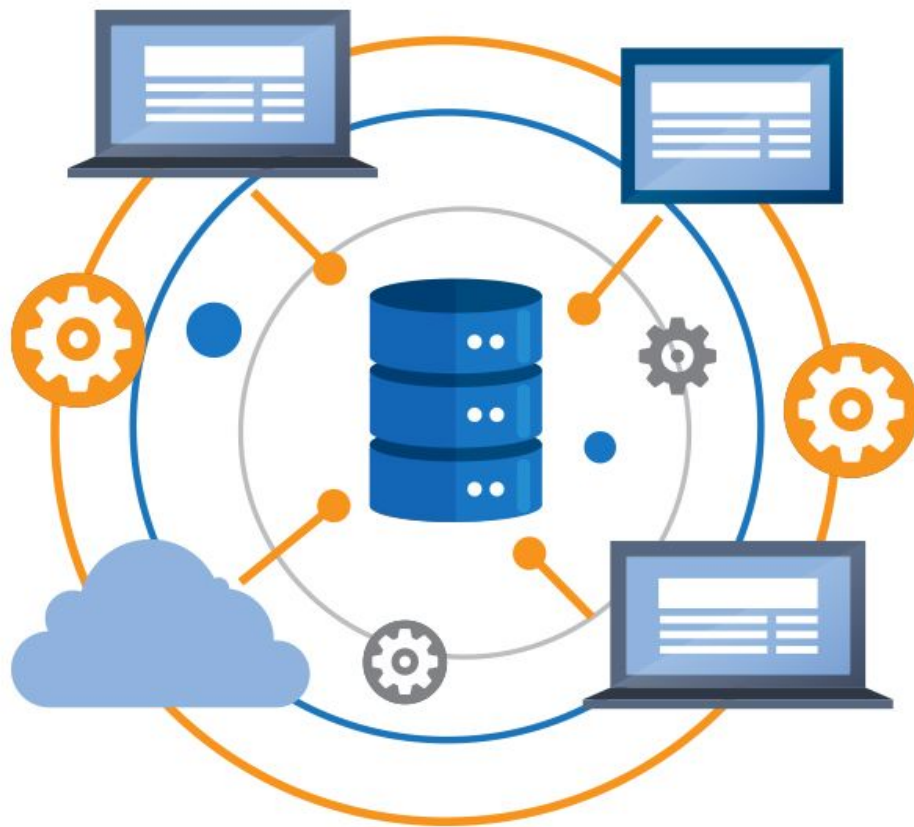


Ciclo de vida de un ViewController

viewDidLoad only executes once.

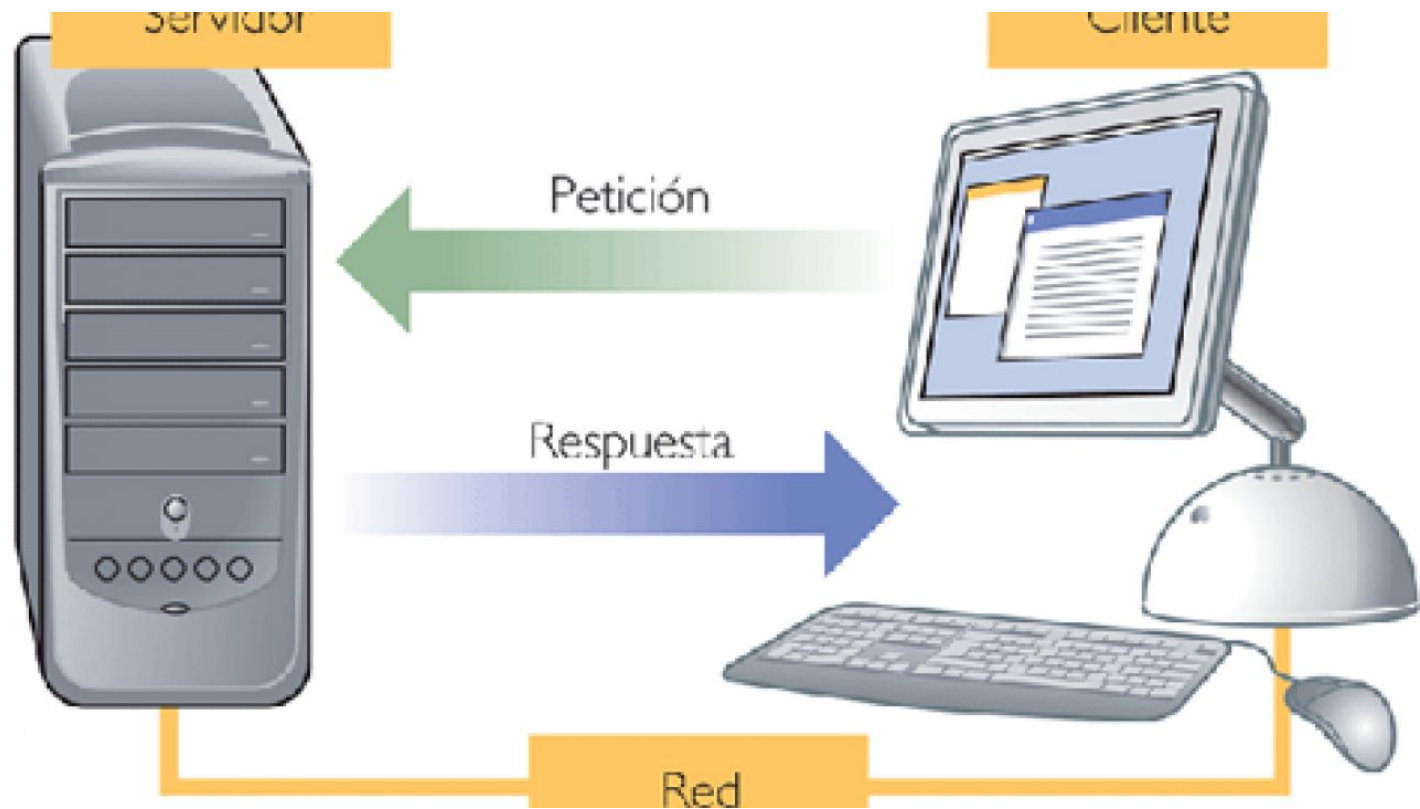


?



?

Arquitectura Cliente - Servidor



Comunicaciones

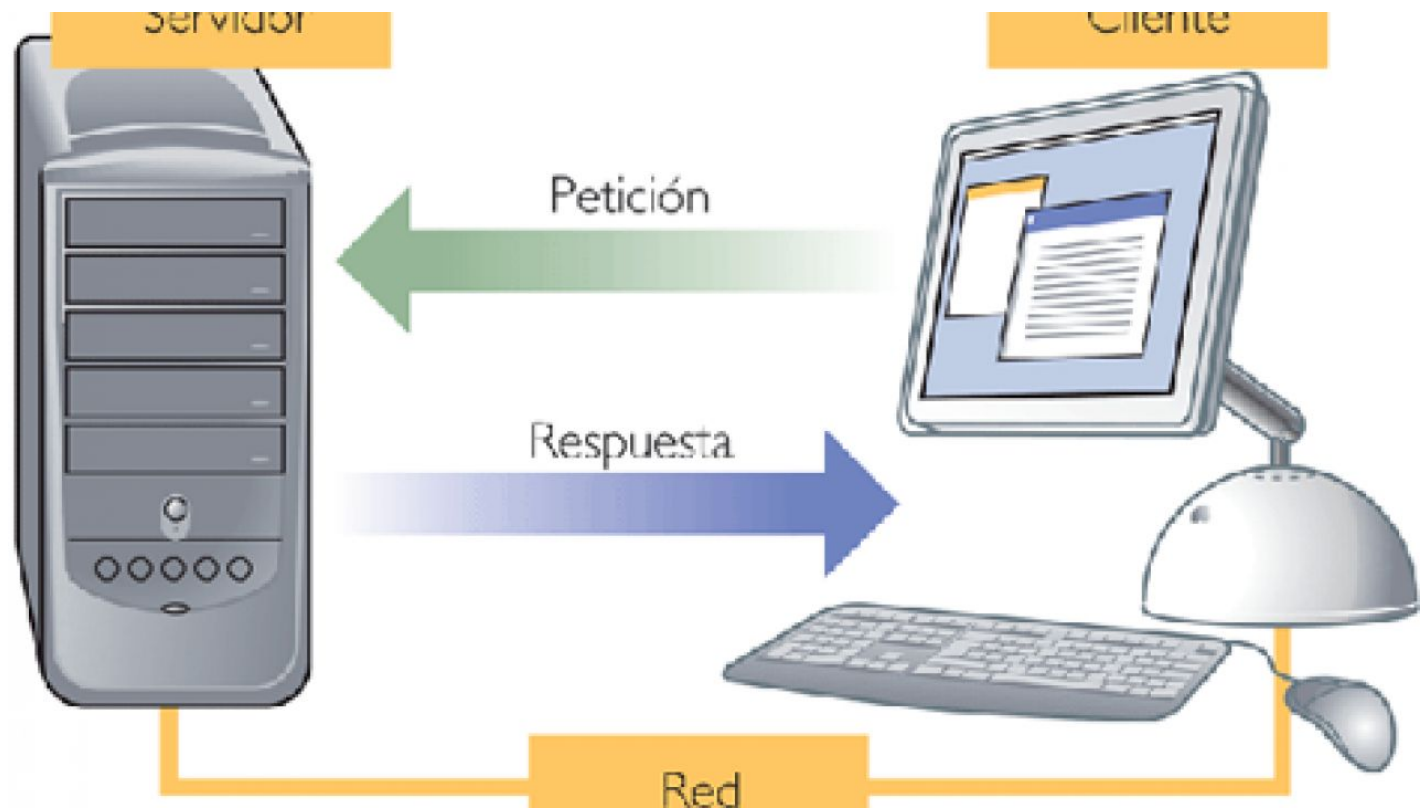
- **Internet**

- ...
- TCP/IP
 - IP *Internet Protocol*
 - FTP *File Transfer Protocol*
(Archivos)
 - SMTP *Simple Mail Transfer Protocol*
(Correos)
 - **HTTP *Hyper Text Transfer Protocol***
(Páginas web)
 - HTML *Hyper Text Markup Language*
 - ...
- ...

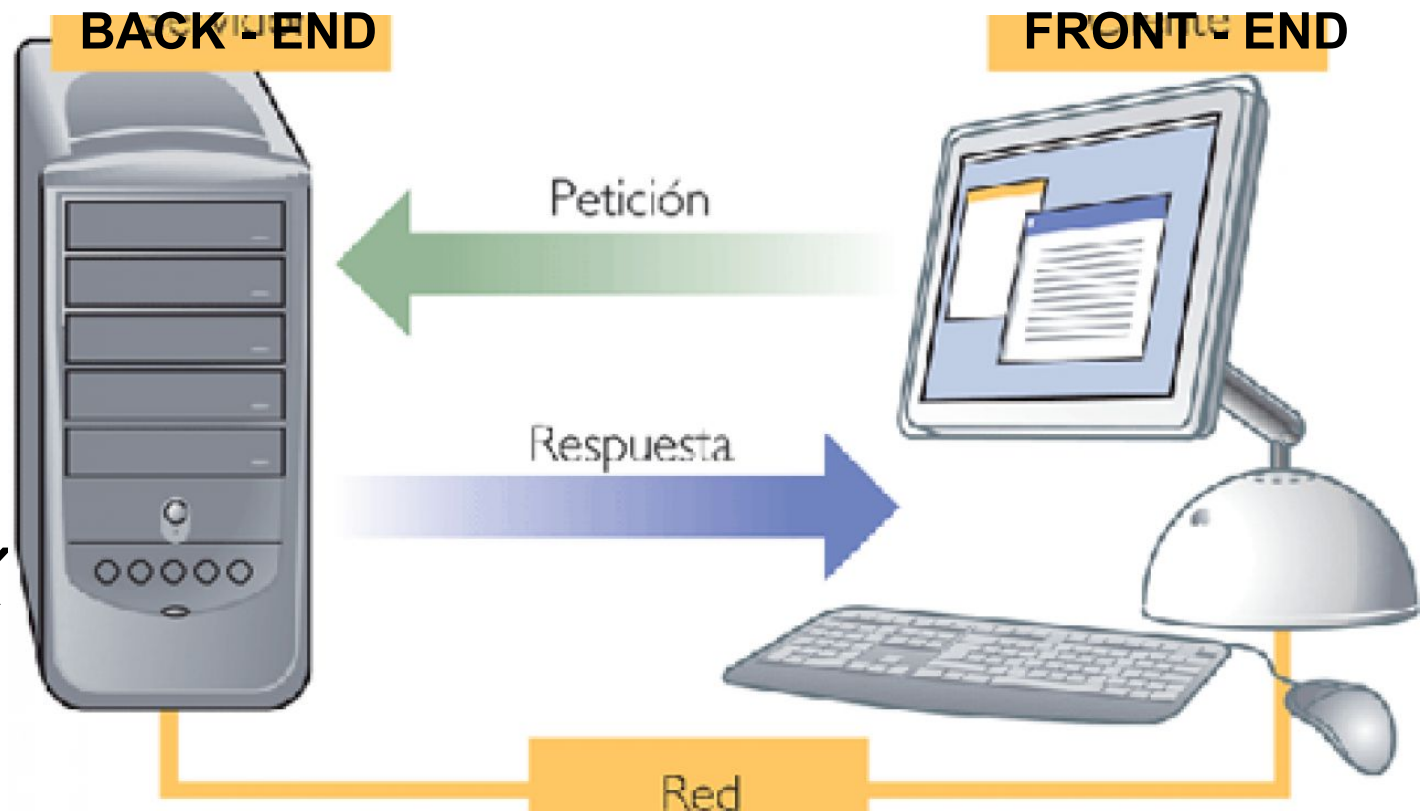
Móviles



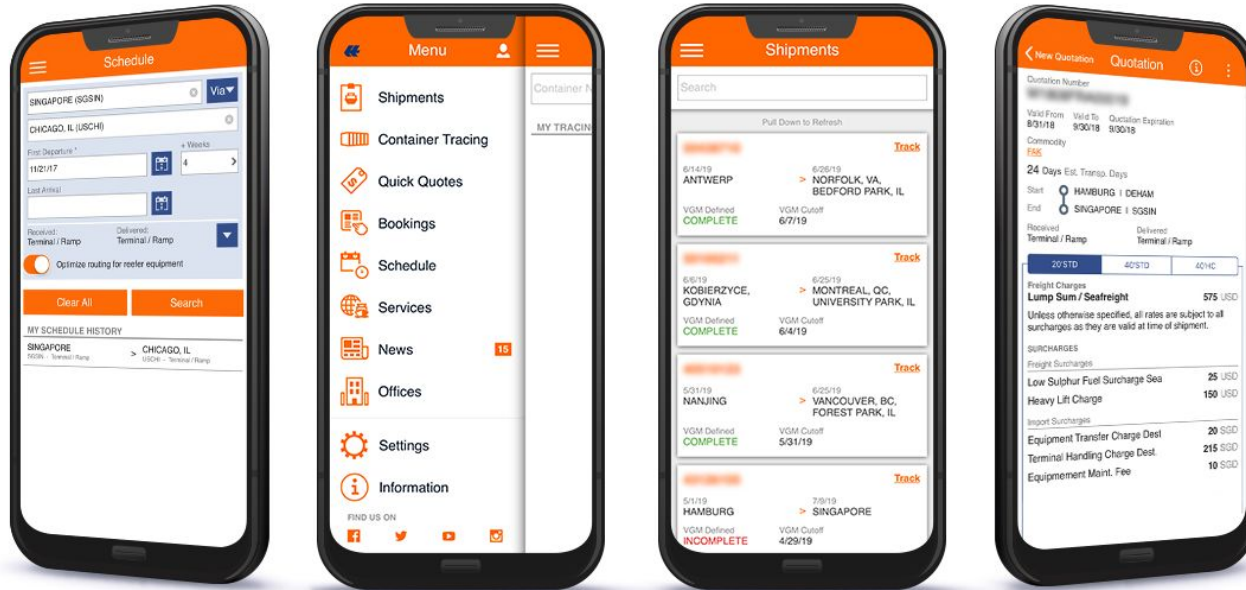
Arquitectura Cliente - Servidor

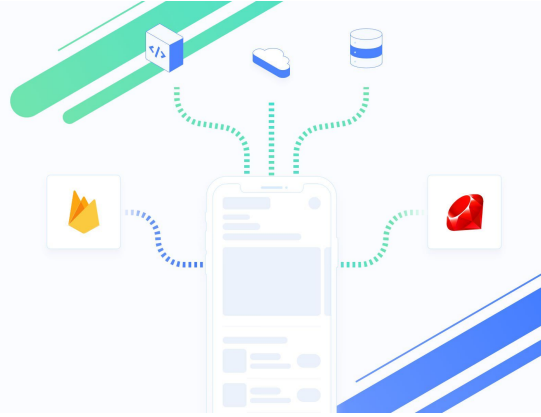


Arquitectura 'X'



Front-End





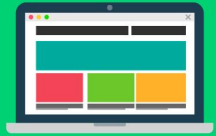
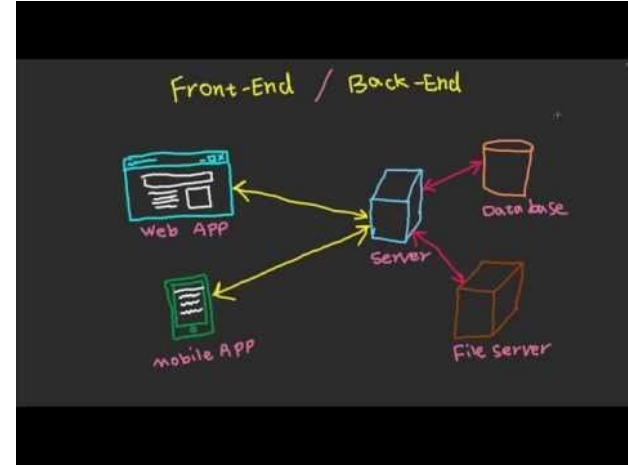
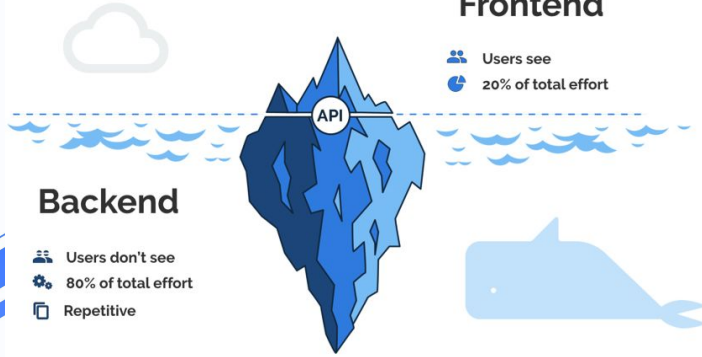
Back-End

Frontend

- Users see
- 20% of total effort

Backend

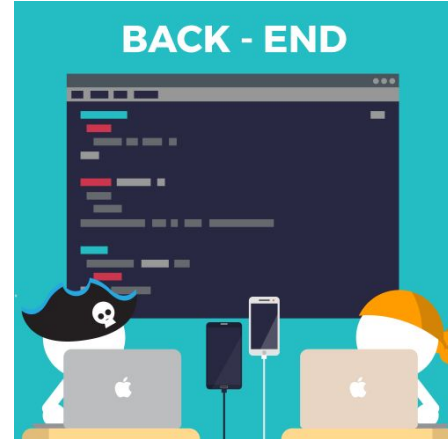
- Users don't see
- 80% of total effort
- Repetitive

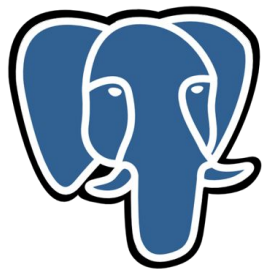


FRONTEND



BACKEND





PostgreSQL



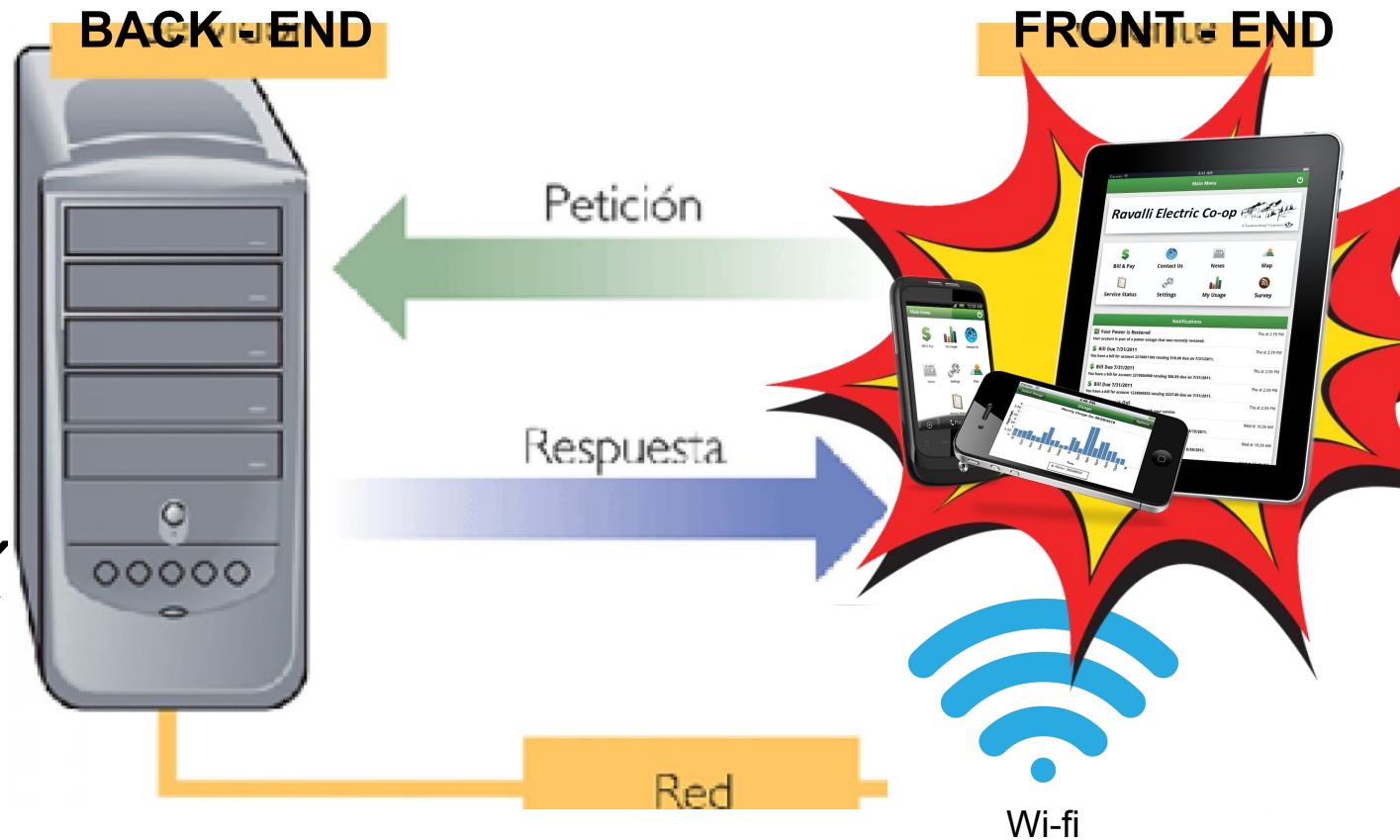
CloudKit



Firebase



Arquitectura 'X'



[illegible]



Android Google Maps



API

learn2crack.com

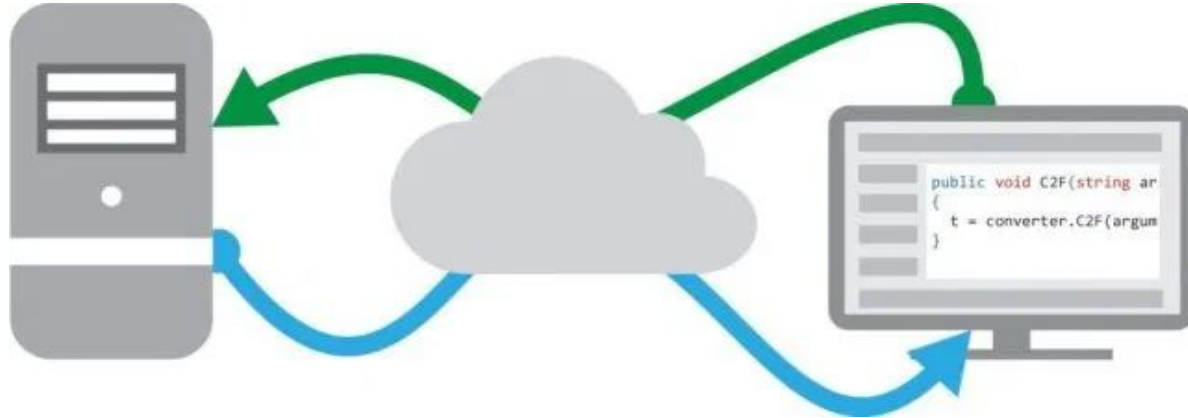


WEATHER API

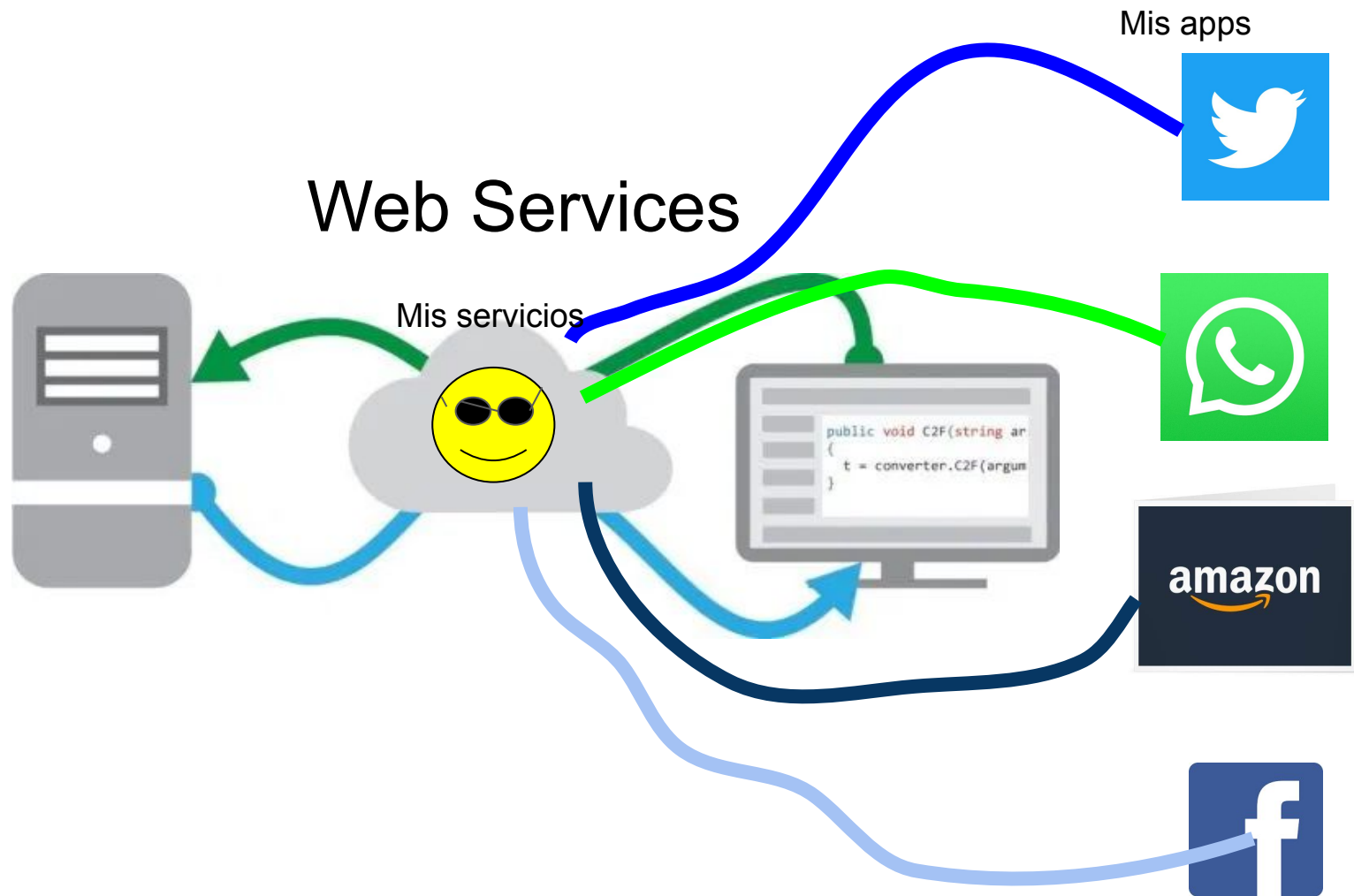


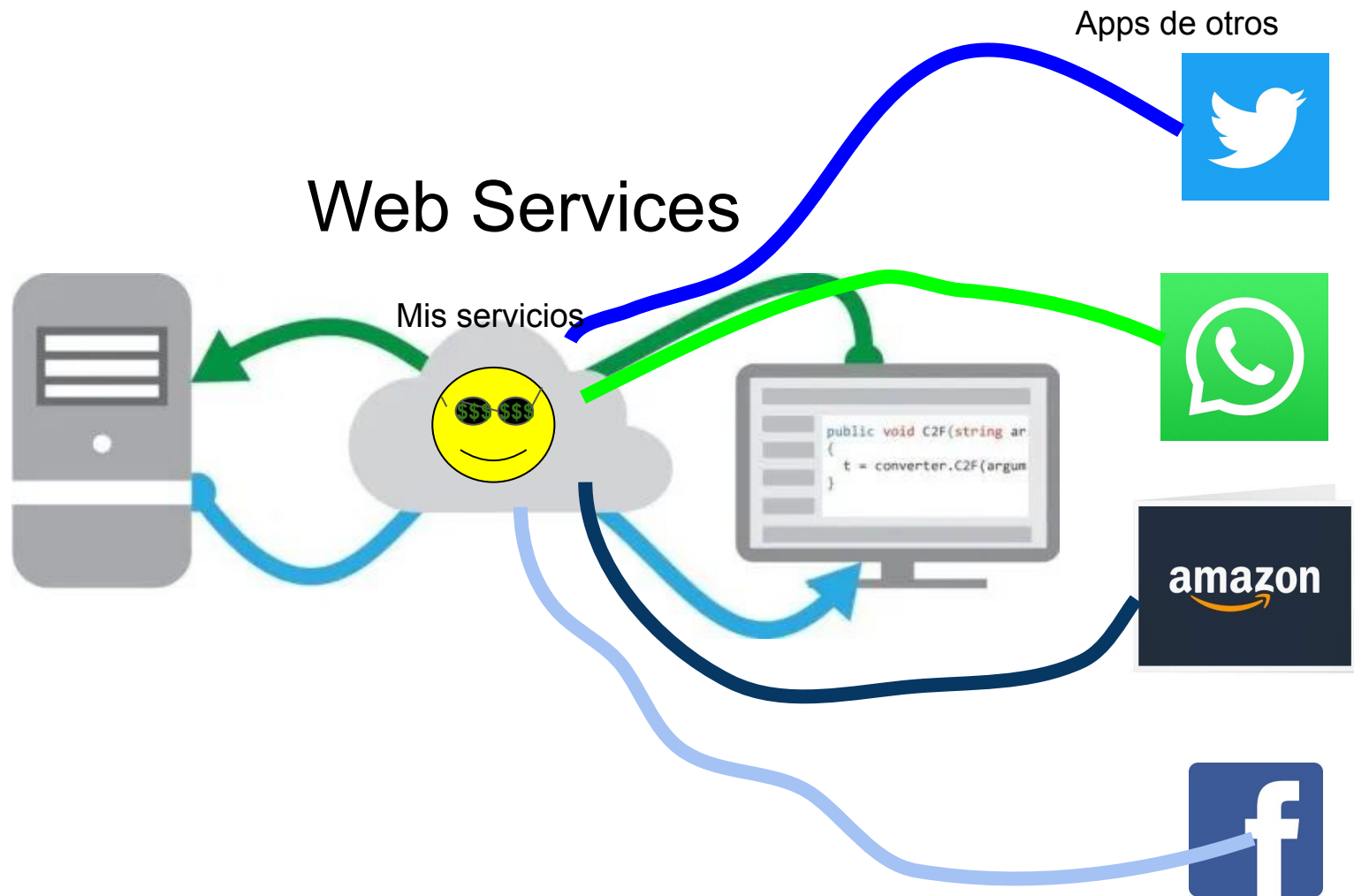


Web Services

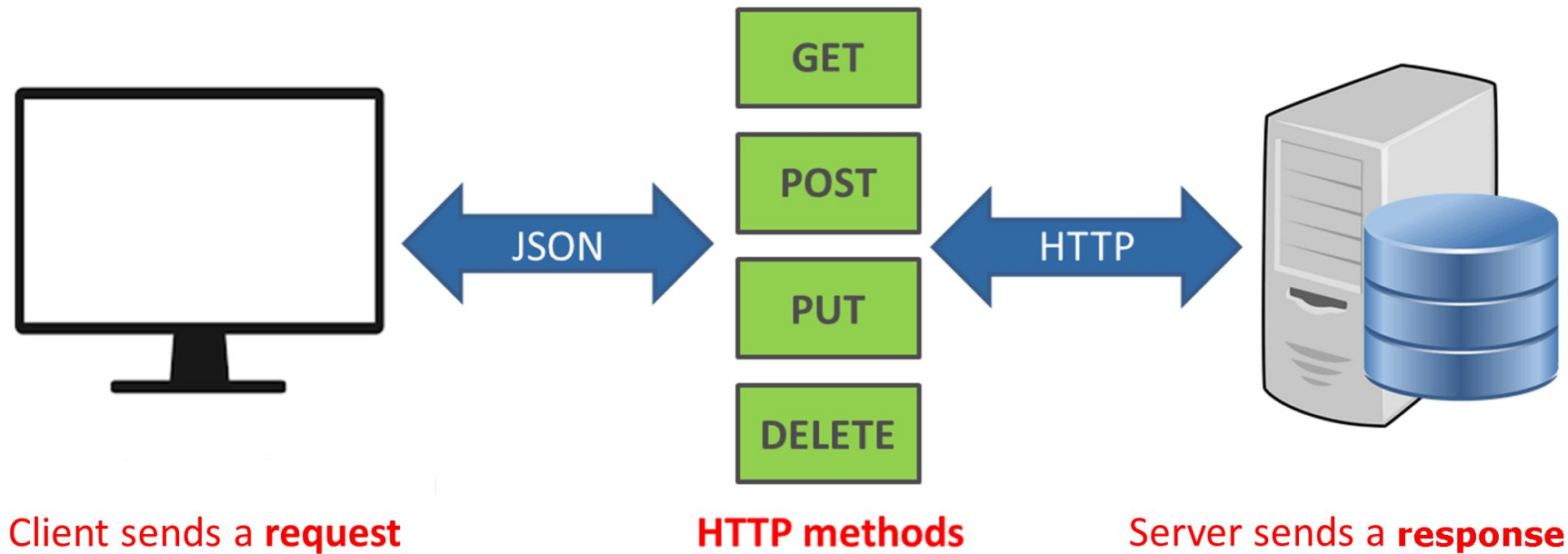


XML
JSON





API REST



Back-End de Swift

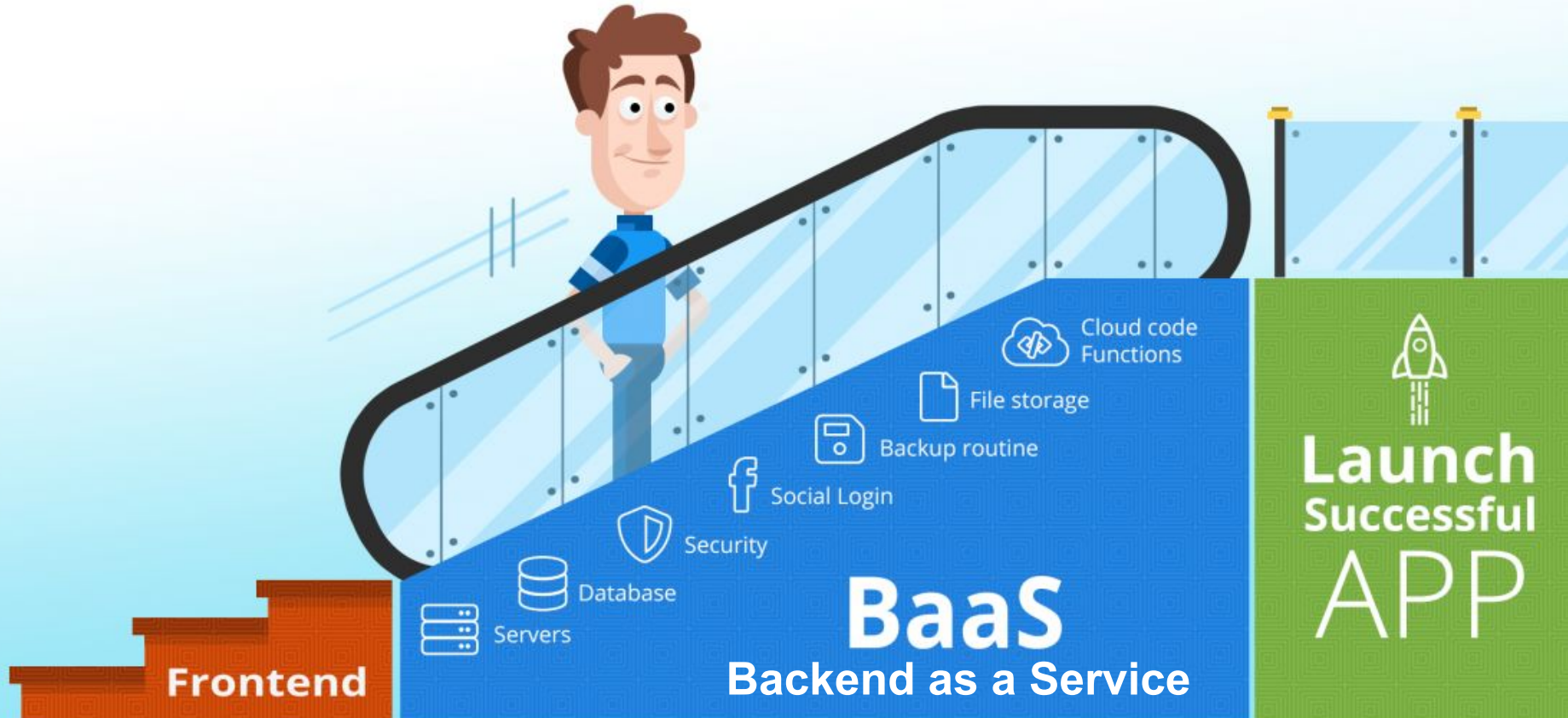


Packages:

- PostgreSQL
- MySQL
- SQLite
- Crypto
- Web Socket
- Auth
- Fluent
- Service
- Redis
- Leaf
- HTTP
- JWT



MBaaS *Mobile Backend as a Service*





- Realm Studio
- Base de datos
- Integración de datos síncrona
- Comunicaciones Offline
- Respaldos
- Mecanismos de Cifrado
- Escalabilidad
- Monitoreo y Alertas
- Global Clusters



Firebase

- Cloud Firestore
- Machine Learning Kit
- Authentication
- Cloud Storage
- Hosting
- Realtime Database
- Crashlytics
- Test Lab
- Performance Monitoring
- App Distribution
- In-App Messaging
- Google Analytics
- Predictions
- A/B Testing
- Cloud Messaging
- Remote Config
- Dynamic Links

aws mobile services

- **AWS Amplify**
 - Control de acceso
 - Machine Learning
 - Mecanismos de Seguridad
 - Escalabilidad
- **Amazon Cognito**
 - Control de acceso SAML
- **AWS Appsync**
 - Integración de datos síncrona
- **Amazon API Gateway**
 - Crear y usar API's RESTFUL
- **AWS Device Farm**
 - Herramientas y ambiente de pruebas
- **AWS Amplify Console**
 - Hosting
- **AWS Mobile Hub**
 - Monitoreo y análisis
 - Base de datos
- **AWS Pinpoint**
 - Herramientas para análisis y marketing