## Objetivo general y objetivos específicos, tecnológicos y de negocio.

Nuestro objetivo principal es la recopilación de la información para su tratamiento, obteniendo las métricas principales que nos permitan entender el comportamiento de los clientes. Una vez finalizado el proyecto podremos responder de forma clara y de un solo vistazo a cuestiones como:

* Tipos de compras realizadas.
* Origen de los clientes.
* Gasto de los clientes.
* Localización y tipo de gasto.

Para ello comenzaremos recopilando la información en formato JSON ofrecida por el api, al ser información ya recogida no necesitaremos un método de almacenamiento previo, por lo que enlazaremos el api con el tratamiento back en python. Recopilaremos toda la información y la filtraremos para generar conjuntos de datos que podamos representar y nos permitan contestar a preguntas como las anteriores. Una vez tratada la información ofreceremos una web sencilla en la que integraremos las gráficas y datos. Esta página web la publicaremos en una plataforma como Firebase para que sea accesible desde cualquier navegador web. De esta forma cualquier persona podrá visualizar las estadísticas de gasto de los usuarios, constituyendo una información muy valiosa para cualquier emprendedor que quiera abrir su negocio.

# Arquitectura inicial

Paystats nos ofrece toda la información por medio de consultas en formato json. Gracias a esta forma de recogida de datos, no necesitamos de un software específico como Kafka para recolectar los datos, ya que la aplicación nos ofrece los datos ya de forma ordenada. La información que nos ofrece es de un sandbox con todos los datos del año 2015, por lo que no necesitamos tratar información en tiempo real, y los datos no se van a actualizar una vez publicados.

Por todo lo anterior se va a realizar el procesamiento de toda la información y posteriormente una vez conseguidos los resultados se realizara la publicación en la web. Para ello se realizará una propuesta simple de arquitectura, primero se enlazará el entorno de trabajo en Python con la aplicación, en el que se realizará el procesamiento de los datos, y una vez generados los resultados se generarán archivos de datos, que se incrustarán en el desarrollo web para la generación de las diversas gráficas e informes que nos permitirán responder a las preguntas planteadas.

Jupyter

(Procesamiento de datos)

Paystats

(Origen de datos)

Web

(Visualizacion)

Ficheros de datos

Como podemos ver primero haremos uso de un entorno de desarrollo para trabajar los datos ofrecidos por paystats en python. Haciendo uso de un entorno de trabajo como Jupyter, se ha creado un proyecto de desarrollo en Python que permite realizar la conexión con la aplicación, desde el entorno de trabajo podemos conseguir todos los datos en formato json, se realiza el procesamiento y una vez obtenidos los datos necesarios para representar la información, estos son almacenados en archivos planos.

El segundo paso es la representación de la información procesada, para ello se ha desarrollado una web que permita visualizar de forma clara todos los datos. Primero se ha escogido una plantilla base con un estilo limpio y claro, a partir de ello se han realizado todas las modificaciones y se ha expandido la web para albergar todos los datos. Los datos se recogen de los ficheros de salida y se incrustan en la web, de forma que toda la ejecución se realiza desde el navegador. Para que esta web sea accesible se ha realizado el despliegue y hosting de la web gracias a Firebase.

# Arquitectura Online

Paystats también nos ofrece la opción de tener acceso a todos los datos posteriores al año 2015 hasta la actualidad. Si se realiza un acuerdo comercial con BBVA podemos tener acceso a estos nuevos datos, y la información se actualizaría de forma semanal. En este caso habría que tener en cuenta esta actualización de los datos por lo que habría que implementar una arquitectura más ambiciosa.

Aunque no se trata de una actualización en tiempo real, sí que habría que hacer una actualización de forma periódica por lo que sería adecuado usar una arquitectura de tipo Kappa. Como en el desarrollo realizado la información la ofrece directamente Paystats por lo que no sería necesario un recolector de información, pero el procesamiento sí que sería necesario realizarlo de forma automática con cada actualización de los datos, y también tendríamos que conectar de forma directa la web con los resultados obtenidos.

Cassandra

(Almacenamiento de datos)

Spark

(Procesamiento de datos)

Paystats

(Origen de datos)

Web

(Visualizacion)

Consultas

En esta arquitectura haríamos uso directamente de la tecnología Spark, como sistema de procesamiento, gracias a PySpark podemos directamente migrar el desarrollo en Python realizado, lo que nos permitirá procesar los datos de Paystats, los resultados en lugar de ser ofrecidos en ficheros, tendrán que ser almacenados en una base de datos como Cassandra. Para recoger los datos podemos hacer uso de servicios Rest, que puedan ser leídos desde la web. Igualmente habría que modificar la web para recepcionar los datos desde servicios rest. Además habría que realizar el hosting no solo de la web sino del procesamiento y almacenamiento.