**Data Engineering Challenge**

**About the Challenge...**

At Zé, we were facing the issue of not knowing where are our couriers, making the current status of deliveries invisible to consumers and to our internal backoffice team. Therefore, we decided to start the development of a new app, to be used by each courier, called Zé Entregador.

With this app, the couriers (its users) will not only manage data regarding orders that are being delivered but also will allow the retrieval of their geolocation information in real-time.

Thus, while tracking each courier current location, we will be able to proper inform the consumers where their order is, thus, reducing their anxiety and also bringing quality, confidence and value for the entire experience with Zé.

**What we need to define**

With the current volume of orders being sold at Zé, we expect to receive around 1M records of coordinates each day - we are considering that every courier will use the new app and this volume will grow each month following Zé's growth.

The app is almost finished, and the couriers are already able to manage the order's information while associating them with the delivery of it, inform that the delivery of the order is starting, and also assign that the order was just delivered - we expect they do this when actually delivering the beverages to the consumer. So, for every action that the courier does regarding each order, we are collecting the location using the device's GPS sensor. Besides these moments, we are also collecting location data while the courier is driving to the consumer's place - this way we expect to send notifications to the consumer that his/her order is arriving or just arrived, for instance.

Regarding the infrastructure on our cloud that will receive all this data, we are really delayed, and it is exactly where we need your help, to help us answering these questions:

* How are we going to receive all the location data from the couriers' app? What protocols, services, components we are going to use to proper receive the data, store it and be available to be used in other products.

**A: Is very important to create some components inside AWS to handle ‘moved’ data as location for each order requested. If need to handle datas that are changing and these infos are needed to design the business model effectively it will be necessary take a look to those component offered by AWS.**

**For sensitive data as in architecture Pub/Sub AWS offers the SNS (Simple Notification Service) and SQS (Simple Queue Service) that works properly with messages without losing any info and still ensuring the quality no need a 3rd part system.**

**key-words: scalable, confiável, transito entre aplicações distribuídas. desacoplamento de aplicativos, agrupa tarefas, conecta microservicos**

* + Should we create an API for it?

**Answer**: It is necessary to create a stable communication between the endpoints and an API can perform this job properly. API Gateway of AWS works fine with lambda functions

* + Should we use some specific managed component to receive this information?

**Answer :** To manager the informations the SNS service is the most recommended because it deliver trusted information and durable, auto scale large payload and provide a durable storage of messages.

* + Should we use queues, pub-sub mechanisms, serverless components?

**Answer:** To achieve the objective of delivery with quality and reliability, the SQS ‘organizes’ the messages flow coordinated by SNS mechanism to deliver message for endpoints once requested. The Lambda will enable or activate services once requested performing operational activities.

* While receiving this location data associated with order's information, imagine we need to ingest more information to it.
  + How would we do this?

**Answer :** You can create new queue(s) to store messages, to group tasks and store notification. So every time you receive a new message, or updates **about the order location the messages will be stored in queue and sent to the user.**

* + In which layer that you have proposed in the previous answer? **- responder...**
* We need to create an API to retrieve the last retrieved location from an order.
  + How do you propose us to do this?

**Answer:**  create an API Gateway to communicate with a queue SQS

* + While receiving location data, can you elaborate a solution to store the order's last location information? - DynamoDB is a good options to store the register once the database offers high availability and quality.

* If we need to notify the users about each order's status, how would you implement this while collecting the data?

**Answer:** lambda function listening or location update event and giving push push to the user.

* How to store the data and how to make it available to be queried by our Data Analytics team?

**Answer:** By the API Gateway you can create a lambda function to receive the datas and store into DynamoDB. It’s an easy and performatic way to handle. Dynamo offers high availability and quality.