

## Entregable 4

### Descripción de la Empresa:

Pagosonline S.A. es una empresa dedicada a ofrecer servicios de pago y financieros de forma digital. Su principal línea de productos es una aplicación que permite a los usuarios transferirse dinero entre sí y realizar pagos en distintos comercios adheridos. Actualmente, toda su infraestructura digital está montada on premise, y el objetivo de este documento es detallar el plan para realizar una migración hacia la nube, para reducir costos y hacer frente a la creciente demanda de expansión en otros mercados.

### Problemática:

La popularidad del servicio que ofrecen ha llevado a la empresa a experimentar fallos en sus sistemas por sobrecarga de peticiones en momentos claves, como por ejemplo CyberWeek y BlackFriday. La empresa está en búsqueda de una solución para poder escalar la potencia de sus equipos durante esas fechas críticas. Adicionalmente, dada la naturaleza sensible de los datos manejados por la empresa, contar con respaldos adecuados (persistentes, seguros y redundantes) resulta difícil ya que implica la construcción de un datacenter propio con distintos servicios especializados de alto coste (seguridad, red eléctrica especial, etc).

### Arquitectura de la solución:

La solución propuesta cuenta con varios servicios de AWS. Entre ellos:

- **EC2:** El backend de la aplicación escrito en Node.js y Python actualmente corre en servidores propios de la empresa, que no son fácilmente escalables. Este mismo backend se implementará en la familia m de instancias que ofrece EC2, dada la naturaleza genérica de la aplicación.
- **RDS:** Para realizar las transacciones es necesario contar con sistemas de bases de datos que soporten transacciones, para eso se migrarán las bases de datos SQL que la empresa utiliza actualmente a bases de datos, también de tipo SQL, pero alojadas en la nube mediante el sistema RDS de AWS.
- **DynamoDB:** Los datos de los usuarios, (identificadores, red de contactos, etc) tienen una naturaleza que favorece su guardado en base de datos noSQL. Actualmente, la empresa utiliza bases de datos en MongoDB para guardar estos datos, y el objetivo de esta migración será comenzar a utilizar el mismo servicio de bases de datos noSQL que ofrece DynamoDB.
- **S3:** Actualmente, PagosOnline.com tiene varios servidores dedicados a guardar información de todo tipo (un data lake propio) dentro de servidores on premise. La gestión de este data lake sin herramientas especializadas se vuelve tediosa y costosa. Adicionalmente dificulta el proceso de ETL. Es por esto que se optará por migrar el datalake de la empresa a S3, donde los accesos podrán ser gestionados con mayor seguridad. Por último, la migración a S3 también permitirá configurar en el futuro herramientas propias de AWS como por ejemplo **Athena** para realizar informes de business intelligence.

- **IAM:** La empresa cuenta con una creciente cantidad de empleados para atender sus servicios. El manejo de accesos a la información se está volviendo exponencialmente complejo con los servidores on premise. Se utilizará la herramienta IAM de AWS para solucionar este problema mediante la creación de una serie de políticas que ayudará a que solamente ciertos empleados tengan acceso a ciertos datos.

#### Cronograma de implementación:

Obviamente, la creación de todos estos servicios en la nube, junto con la migración de los datos a las diferentes bases de datos para consumo instantáneo (Dynamo y RDS) junto con la migración de los datos más permanentes a S3 lleva tiempo. Adicionalmente, es necesario calcular una cierta cantidad de tiempo para testing. Aquí un cronograma de seis semanas para la migración.

- **Semana 1:**
  - Creación de cuenta de AWS, usuario Root, usuarios IAM y políticas de permisos.
  - Creación de instancia EC2 y de imagen de sistema para correr el backend.
  - Prueba de funcionamiento del backend en la nueva instancia EC2.
- **Semana 2:**
  - Creación de bases de datos en RDS para modelar las transacciones.
  - Creación de bases de datos en DynamoDB para modelar los datos personales de los usuarios.
  - Testeo de funcionalidad del backend corriendo en EC2 haciendo llamadas a las bases de datos en RDS y DynamoDB.
- **Semana 3:**
  - Creación de distintos *buckets* de S3 para almacenar la información de tipo permanente.
  - Testeo de conexión entre los backend y los datos almacenados en S3.
  - Testeo general de accesos al sistema y del backend.
- **Semana 4:**
  - Migración parcial de datos a RDS, DynamoDB y S3
  - Testing intensivo y automatizado de todas las funciones del backend tras la migración final.
- **Semana 5:**
  - Migración total de los datos tras el testeo de la semana anterior.
- **Semana 6:**
  - Levantamiento de los nuevos sistemas en producción.
  - Revisión de los objetivos alcanzados y seteo de nuevas métricas.