

## Entregable 5

### **Proyecto de migración a la nube: PagosOnline S.A.**

#### **1- Introducción**

Para realizar la migración a la nube vamos a utilizar un *framework* de cinco pasos para ordenar las ideas y los pasos a seguir. En particular, vamos a describir las estrategias utilizadas en la migración en cinco secciones a lo largo de este trabajo. A saber:

- 2- Oportunidad
- 3- Análisis
- 4- Planificación de tiempos de implementación
- 5 - Migración y resultados
- 6- Optimizaciones Futuras

Por último, el trabajo cuenta con una séptima sección de mejores prácticas, una octava sección de conclusiones y una novena sección de bibliografía.

#### **2- Oportunidad**

##### 2.1 - Análisis de la compañía

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*.

##### 2.2 - Contexto

La compañía nacida en el año 2010 desarrolló sus operaciones inicialmente en Argentina, pero en los últimos años comenzó a expandirse a países de la región. Esta expansión, sumada a la creciente popularidad del producto ha llevado a la empresa a experimentar fallos en sus sistemas en momentos críticos, como por ejemplo cuando los comercios electrónicos realizan muchas peticiones en fechas como CyberWeek o Black Friday.

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 las

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).

### **3- Análisis**

#### **3.1 - Descripción de la solución**

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

- **EC2**

El backend de la aplicación principal de PagosOnline S.A. está 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.

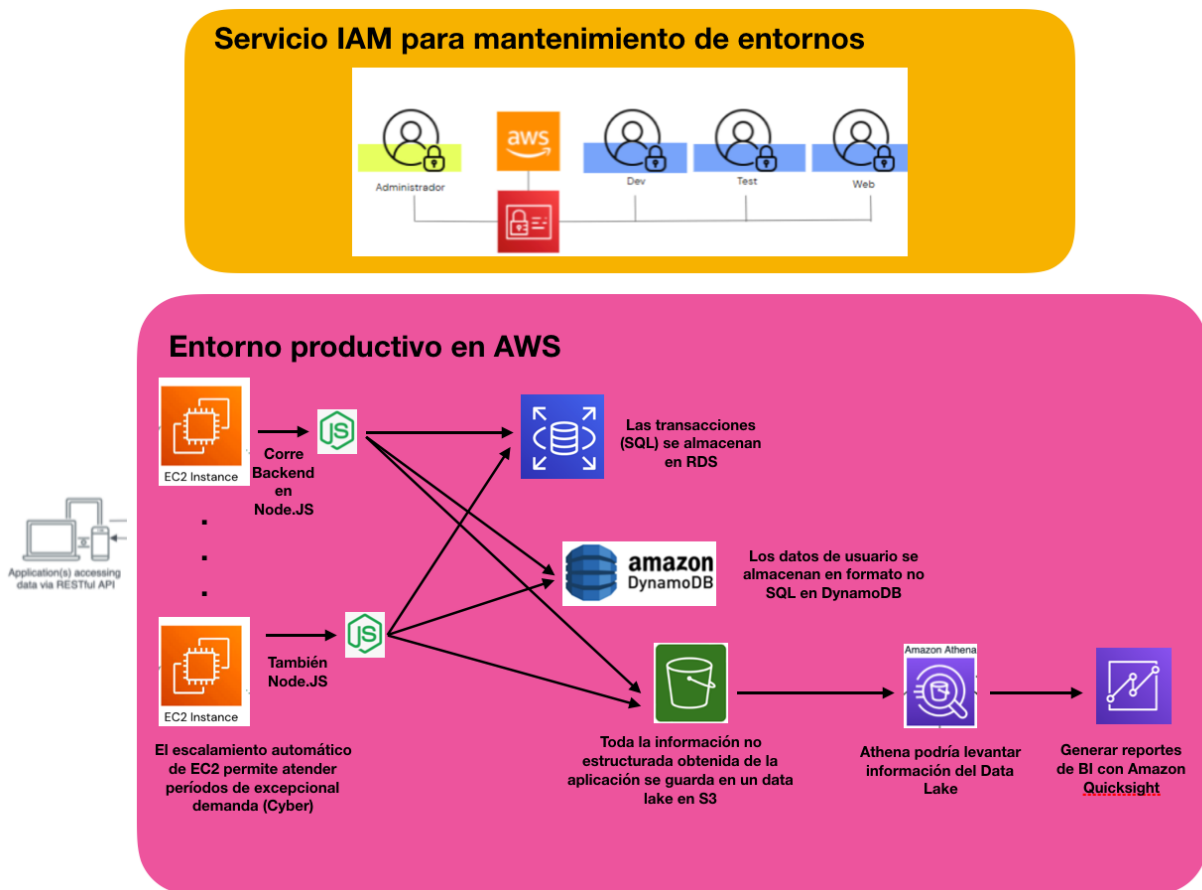
- **IAM**

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.

### 3.2 - Parámetros de la cuenta de AWS

La mayoría de los servicios de PagosOnline S.A son prestados a clientes basados en latinoamérica. En general, la latencia no es un factor determinante para los usuarios de este tipo de servicios. Dado lo anterior, **se optará por configurar la cuenta de AWS en la región de North Virginia**. De esta forma, se ahorrarán costos comparado con la alternativa de utilizar la región de Sao Paulo.

### 3.3 - Visualización de la arquitectura de la solución



## 4- Planificación de tiempos de implementación

### 4.1 - Descripción del 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.

#### 4.2 - Tabla Gráfica del Cronograma de Implementación

	Mes	1				2	
Actividad	Semana	1	2	3	4	5	6
Creación de cuenta AWS root + Usuarios IAM							
Creación de políticas de permisos para desarrolladores y empleados administrativos							
Creación de instancia EC2 con imagen Linux							
Subida preliminar y testing del backend Node.JS en EC2							
Creación de bases de Datos RDS para modelar transacciones con SQL							
Creación de Bases de Datos DynamoDB para modelar datos personales							
Testeo de Conexión del Backend con las bases							
Creación de Buckets S3							
Testeo de conexión Backend con Buckets S3							
Testeo general del funcionamiento del sistema							
Migración preliminar de datos desde on premise a Dynamo y RDS tras testeo exitoso							
Testeo general tras la migración							
Levantamiento en producción							
Revisión de objetivos alcanzados y definición de nuevas metas							

## **5- Migración y resultados**

### **5.1 Resultados Estratégicos**

Con la implementación de un sistema como el propuesto, se lograrían mejores resultados en términos de latencia durante períodos de alta intensidad en la Api. Adicionalmente, habría mejor control de accesos a los datos gracias al sistema IAM de AWS. En total, se estima una reducción de costos de alrededor del 30%, sumado a las mejoras intangibles en capacidad de escalamiento.

## **6- Optimizaciones Futuras**

Una última ventaja del sistema propuesto es la futura integración con Athena de AWS, que permitiría la creación de reportes de Business Intelligence especializados en función de los datos obtenidos mediante queries SQL del *data lake* almacenado en S3.

## **7- Mejores Prácticas Recomendadas**

- Guardar cuidadosamente el usuario y clave *root* del nuevo sistema. Este usuario posee permisos de administrador superiores a los de cualquier usuario IAM. De esta forma, un compromiso de esta cuenta representaría significativas vulnerabilidades para los sistemas informáticos de la empresa.
- Delegar un equipo de personas dentro de la empresa para trabajar, modificar y mantener el sistema de permisos IAM. Controlar la seguridad de este tipo de datos es fundamental, y AWS ofrece un sistema muy poderoso para manejarlos. Es indispensable, no obstante, que quienes manejen este sistema estén capacitados en su uso.

## **8- Conclusiones**

Actualmente, la empresa enfrenta grandes obstáculos en el escalamiento de sus sistemas informáticos. La migración a la nube de los sistemas ofrece en este caso grandes ventajas. En primer lugar, permitirá el escalamiento de los servidores con mayor facilidad. En segundo lugar, abaratará los costos de mantenimiento, ya que AWS es quien se ocupa del mantenimiento de los datacenters, la redundancia y la seguridad del hardware. En tercer lugar, la migración resolverá problemas de accesos y seguridad a los datos mediante el poderoso sistema de accesos ofrecido por Amazon IAM. En cuarto lugar, mediante S3 será posible construir un data lake propio del que la empresa podría obtener valiables *insights*.

En definitiva, la migración representará una reducción de costos y una ventaja competitiva para la empresa, que podrá ofrecer mejores servicios a sus clientes que el resto de los competidores en la industria.

## 9- Bibliografía

- AMAZON (2023). Six advantages of cloud Computing. Recuperado de:  
<https://docs.aws.amazon.com/whitepapers/latest/aws-overview/six-advantages-of-cloud-computing.html>
- AMAZON (2023). Cost optimization Pillar - AWS Well-Architected Framework. Recuperado de:  
<https://docs.aws.amazon.com/whitepapers/latest/aws-overview/six-advantages-of-cloud-computing.html>