

ARTÍCULO PROYECTO 2 AREP

First A. Author, CESAR EDUARDO LANOS CAMACHO,
PROYECTO #2

• INTRODUCCIÓN

Este documento presenta la arquitectura del aplicativo web “Convertidor de moneda”. Convertidor de moneda es un convertidor de moneda de USD a cualquier valor (ya que la API de conversión de moneda pública es limitada) pero fue perfectamente diseñada para manejar cualquier conversión. Se usaron tecnologías: HTML5/CSS con Javascript para el aplicativo web (página web) y los javascript que manejaba los valores de las diferentes monedas por medio de un JSON que lo daba un API Gateway de Amazon, Bootstrap para el manejo del diseño responsive, de AWS (Amazon Web Service) se usó función Lambda y API Gateway para el manejo y extracción de las tasas de conversión de la API de conversión de moneda pública y por último el Servicio S3 también de AWS para el alojamiento y visualización de nuestra página web para nuestro usuario final.

PASOS PARA LA PREPARACIÓN Y EJECUCIÓN DEL CÓDIGO

Para compilar el **PROYECTO #2 AREM** se deben seguir los siguientes pasos en orden:

A. Empezar

- Visite el repositorio del proyecto del siguiente enlace
<https://github.com/arem2019-1/Proyecto2Arem> copie el link **HTTPS** del proyecto y usando Linux haga una clonación en su carpeta preferida o copie la siguiente línea de código y péguela en su consola para alojar el proyecto:

```
git clone  
https://github.com/arem2019-1/Proyecto2Arem
```

- Una vez ubicado en la raíz del proyecto, dentro de la carpeta `public_html` están alojados todos los archivos para el funcionamiento de la página que se puede abrir por medio de un navegador el siguiente archivo: **index.html**

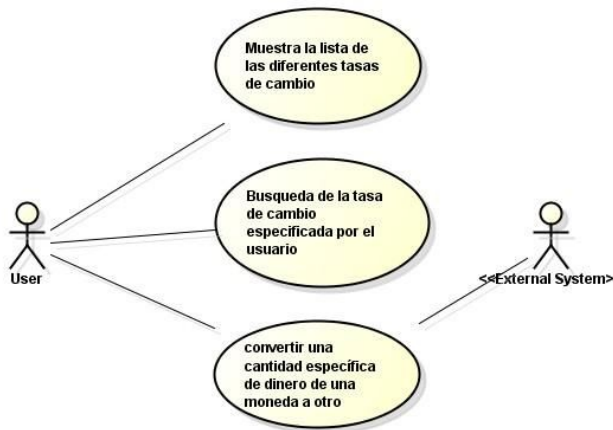
CONTEXTO DEL APLICATIVO

Para este proyecto se consideró diseñar e implementar una aplicación para la consulta y búsqueda de tasas de cambio y para la conversión de monedas. La aplicación debe construirse con:

- microservicios usando la infraestructura de AWS lambda, AWS API Gateway y Amazon S3 para el alojamiento de nuestra página web.
- El aplicativo debe tener una interfaz de usuario simple con Responsive Design, el servidor es un servidor de publicación de tasas de cambio y el canal de comunicación es TCP o HTTP.

3. DISEÑO DEL APLICATIVO

a. Requerimientos Funcionales



1 Figura 1 – Diagrama de casos de uso

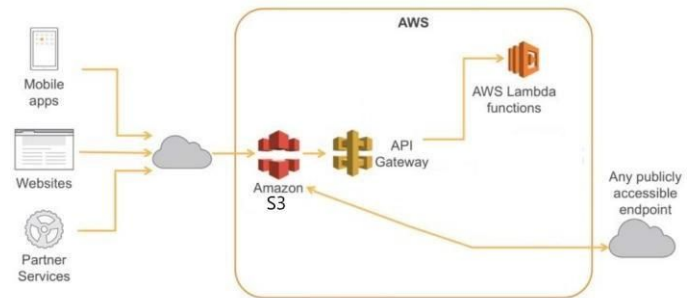
En la figura 1 se presenta el diagrama de casos de uso con los servicios funcionales de la aplicación que son:

- Lista de las diferentes tasas de cambio.
- Búsqueda de una tasa de cambio específica por el usuario.
- El debido cambio de una determinada tasa a otra.
- Las operaciones de la red deben ser operaciones sin bloqueo.

b. Atributos de Calidad

- Responsive: el cliente puede estar satisfecho de ver la aplicación en diferentes dispositivos sin causar algún error.
- Sin Bloqueo: Las operaciones que haga el usuario siempre darán respuesta.

3. DISEÑO DEL APLICATIVO INFRAESTRUCTURA



2 Figura 2 – Diseño de Infraestructura

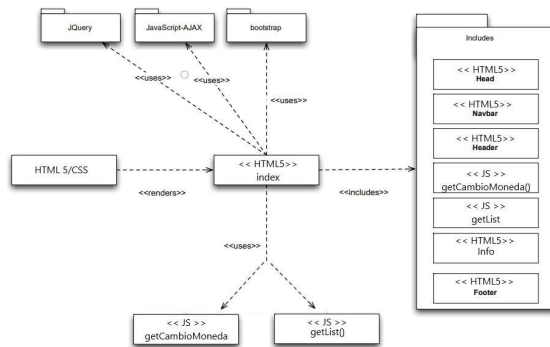
En la Figura 2 se presenta la aplicación de microservicios desplegada en la infraestructura de AWS. La aplicación está diseñada para soportar clientes web, clientes móviles y servicios externos, que se comunican usando la infraestructura web (e.j., web services REST). En la primera parte encontramos Amazon S3, que es un servicio de alojamiento web donde como su nombre lo dice alojamos todos los archivos en este caso la página web del aplicativo donde a su vez este nos crea un punto final o una comunicación final para el usuario generandonos un link mostrando el aplicativo en funcionamiento. Luego encontramos el gateway API, que expone la interfaz de los servicios de la aplicación, ocultando la tecnología de implementación. Finalmente, el microservicio que traía los valores de cada moneda disponible del API de conversión de moneda pública se implementó en una función Lambda de Amazon.

Proyecto 2

Arep

REFERENCIAS

4. Diseño capa WEB



3 Figura 3 – Diagrama Capa Web

Todo se trabajó con HTML5, CSS y JAVASCRIPT directamente, debido a que Amazon S3 ya ofrecía un alojamiento web. Todo se maneja desde el index que está usando JQuery, Javascript para la comunicación directa con el API Gateway que creamos para que por medio de la función Lambda nos retornará un JSON de los datos de las diferentes monedas, las clases Javascript getCambioMoneda y getList son las encargadas de trabajar con ese JSON y ahí creamos las diferentes funciones o métodos para trabajar directamente en el index para que muestre o trabaje la información. Se usa Bootstrap para el manejo del Responsive Design

5. CONCLUSIONES

- Amazon S3 ofrece una completa seguridad a nuestros archivos en este caso el aplicativo web para almacenarlo en su nube y brindando también un punto de comunicación final para el usuario por medio de un link donde se visualizaban los archivos de la página desde la nube.
- Los servicios que ofrece Amazon en la nube hacen una herramienta muy práctica y cómoda para utilizar, pero así mismo es complejo de implementar.

Luis Daniel Benavides Navarro,
“CurrencyC: Architecture Document”
Artículo. (Bogota - Colombia).

Nicolas Osorio, “Configuración API Gateway con función Lambda para consumo de un JSON de un API de ejemplo”, link del video:
https://www.youtube.com/watch?v=fUeVpSm_o_gk&list=LLBhJ_K9A01xzlnO7poBs93A&index=3