

# Proyecto cambios de divisa

**Abstract—** En este proyecto se busca realizar una aplicación que realice cambios en las tasas, usando apis de divisas capaces de entregar en tiempo real el valor de las tasas, además usaremos la API Gateway de AWS y Lambda para el funcionamiento de la aplicación.

**Index Terms—**Amazon Web Services (AWS), GitHub, JavaScript, bootstrap.

Aplicación web en <http://tasacambio2.s3-website-us-east-1.amazonaws.com/>

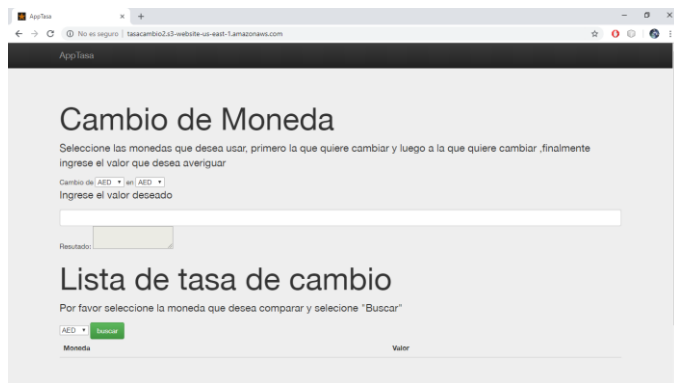
Repositorio del proyecto:  
<https://github.com/alejandrordto/TasadeCambio>

## I. INTRODUCCIÓN

En el siguiente documento voy a describir el trabajo que se ha realizado. Este trabajo consistió en conectar una aplicación financiera para mirar las tasas, con nuestra API web para tener una aplicación de cambio de moneda funcional.

## II. PASO A PASO

- Lo primero a realizar es una aplicación web simple en NetBeans usando Bootstrap.



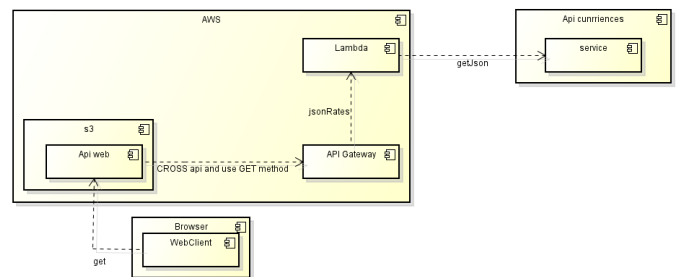
La aplicación está dividida en dos partes, primero el Cambio de moneda donde el usuario cuenta con dos menús desplegables para que el usuario pueda elegir qué tipo de moneda quiere y en cuál quiere convertirla, también cuenta con un cuadro de texto donde el usuario colocará la cantidad que desee, y un recuadro para el resultado, la segunda parte realizada fue la Lista de cambio la cual tiene un menú

desplegable con las divisas, y al elegir una se mostrará esta vs el resto en una lista.

- Ya hecha la página web, vamos a conectar la API financiera [openexchangerates.org](https://openexchangerates.org/) capaz de entregarnos con un JSON el valor de las divisas respecto al dólar de manera real, para ello primero debemos acceder a la consola de AWS y crearemos un IAM que tenga full acceso, luego dicho IAM se usará al crear una función Lambda, en la cual vamos a recoger el JSON de [openexchangerates.org](https://openexchangerates.org/) y finalmente, accedemos a API Gateway, donde crearemos un recurso y un método GET, conectado a nuestra función Lambda.
- Ya con el servicio AWS funcionando lo implementamos dentro de nuestra aplicación web y hacemos las pruebas finales, para finalmente subir el código a S3 y lograr que este se despliegue.

## III. ARQUITECTURA USADA

En este trabajo, se tiene un cliente web que accede desde un navegador a nuestra página web, por lo que mandará una solicitud de GET, la aplicación web está dentro de S3, el cual puede traducir la información y mandársela al navegador, este se vincula con la API Gateway haciendo uso del método GET y la API hace uso de la función Lambda, en este caso llamada `jsonRates`, la cual es la encargada de conectarse a la aplicación financiera y pedir por el servicio específico el JSON con las divisas.



## IV. INCONVENIENTES

El código para conseguir el JSON de [openexchangerates.org](https://openexchangerates.org/), no fue encontrado hasta después de la fecha de entrega, lo que causó un retraso con la finalización correcta de la aplicación.

## V. CONCLUSION

En este proyecto, aun así haya habido inconvenientes por el encabezado en la función Lambda haciendo uso de los servicios de API Gateway y Lambda podemos hacer una arquitectura más sencilla que usando EC2 y Spring o Spark, y facilita entonces la implementación de los servicios de AWS.