**HACKATHON BBVA 2021**

**Team: X-Women** *23/10/2021*

**Challenge:** ***Buró de contracargos***
**TÍTULO DEL PROYECTO: Contrabank**

Entregables
**OBJETIVO DEL CHALLENGE:**

Construir una herramienta que ayude a los bancos y comercios nacionales a protegerse de consumidores que posiblemente realicen fraude por medio de contracargos.

**RESUMEN DEL PROYECTO:**

[¿Por qué es importante?]
Nos resulta importante porque estadísticamente ha habido muchos fraudes por este medio y queremos ayudar a combatir este tipo de transacciones fraudulentas.

## DESCRIPCIÓN DETALLADA

Aplicación web de acceso abierto a información mediante visualización de datos que ayude a los bancos y comercios a identificar usuarios que hayan hecho contracargos para tomar las acciones pertinentes.
Se pretende hacer uso de HTML, CSS, JS y PHP (el ultimo tentativamente para posible desarrollo de API) así como el uso de layouts y Datasets en uso conjunto con herramientas como Power BI o R.
También se tiene previsto realizar nuestros avances a través de la plataforma GITHUB
Herramienta tentativamente usada (Cloud):AWS

## Explica la arquitectura y componentes principales de tu solución.

Como "Contrabank" será una aplicación web que alojará datos bancarios, hemos tomado la decisión de hacer de esta aplicación un sitio seguro que le brinde al usuario la oportunidad de manejar la información como mejor le convenga de forma que al interactuar con nuestra app no presente

dificultades en navegarla. Para cumplir con los anterior, nuestra aplicación es estructurada de la siguiente manera:

Power BI: Ofrecerá simultáneamente una vista atractiva e interactiva con los datos y/o filtros al usuario, mostrando de esta manera las estadísticas que maneja el buró de contracargos cumplen la función de filtrar la información de manera organizada a través del uso de gráficos y complementos visuales.

HTML, CSS: Los usamos como medios para darle vida y sentido a la aplicación web.

JS: Nos encargamos de resolver la parte lógica de la aplicación web.

PHP: Nos aseguramos de que la API esté lista para conectar a la base de datos con el Dashboard.

Postman: Prueba del servicio web.

Slim Framework: Construcción de la API.

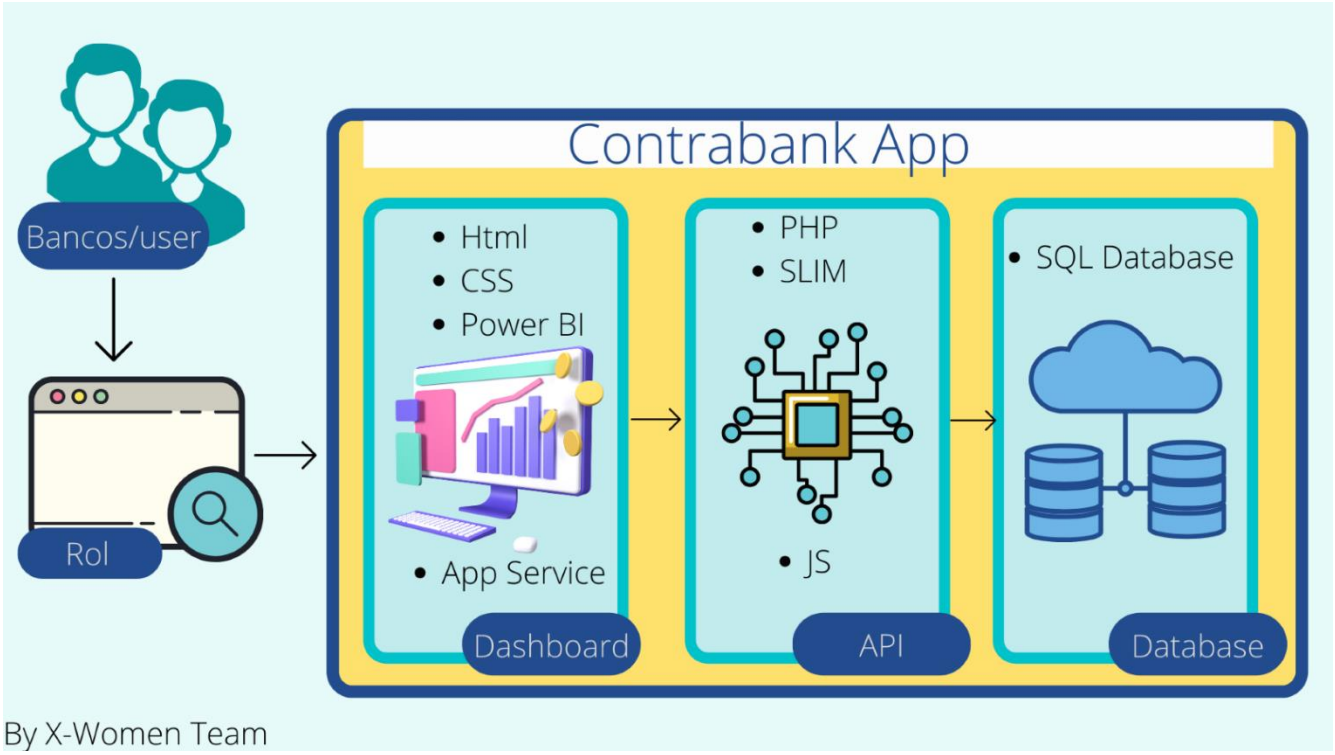
SQL Database: Nuestra base de datos y los métodos usados para agregar y visualizar datos se encuentran alojados en la nube de Azure, abriendo posibilidad a que pueda mudarse hacia AWS.

App Service: Despligue de la aplicación web “Contrabank”.

## Lista las principales tecnologías que utilizas.

HTML, CSS, JS, PHP, Power BI, SQL Database, App Service (Azure)

## Archivo




**Día 3: Entregables 10:00h**

***Guion del pitch:***

¿Sabías que en México existe un alto índice de contracargos? Al menos en 2020, se registraron 3, 451, 219 contracargos en compras con tarjetas de crédito y débito. ¿Y qué el 88.8% resulta en una resolución favorable para el tarjetahabiente? Por otro lado, los fraudes cibernéticos subieron 1% respecto a 2019 y representan cada año una mayor proporción; en 2016 representaba el 32% y en 2020 llegaron a 69%.

Por lo anterior nace la necesidad de crear una solución para que bancos, procesadores de pagos y comercios nacionales interesados, puedan compartir o acceder a esta información de usuarios que han llevado a cabo contracargos frecuentemente brindando así la opción de protegerse del fraude.

Es así como surge CONTRABANK, una aplicación web basada en la nube donde cualquier entidad previamente aprobada podrá registrarse para compartir y visualizar fácilmente datos bancarios de los distintos tipos de contracargos.

Esto a través de un dashboard creado en Power BI, que ofrece una fácil y dinámica visualización para el usario, ofreciendo información relevante y de fácil entendimiento a la entidad gracias a los distintos filtros y al código de colores implementado.

Así mismo, se ha creado una API () de fácil y libre acceso a las entidades previamente registradas, quienes podrán consumirla para dar de alta reporte de contracargo y así mismo obtener información de suma relevancia.

Como sabemos que los datos bancarios son datos de carácter privado y confidencial nos hemos dado a la tarea de brindar seguridad a la aplicación para salvaguardar la información y por consiguiente a los usuarios, para que de esta forma se les brinde la oportunidad y la confianza de compartir y consultar la información en beneficio propio y de la red que conformará CONTRABANK.

Cabe destacar que CONTRABANK es una solución tecnológica basada en la nube, con la finalidad de ofrecer un modelo óptimo, dinámico, escalable y ágil. De esta forma abrimos la posibilidad de que la solución pueda ajustarse a las necesidades e incluso migrarse a otras nubes.

Repositorio de código: [FanyEstAg/BBVA_X-Women (github.com)]

## Arquitectura del Proyecto

### Análisis de la arquitectura de acuerdo a MVC

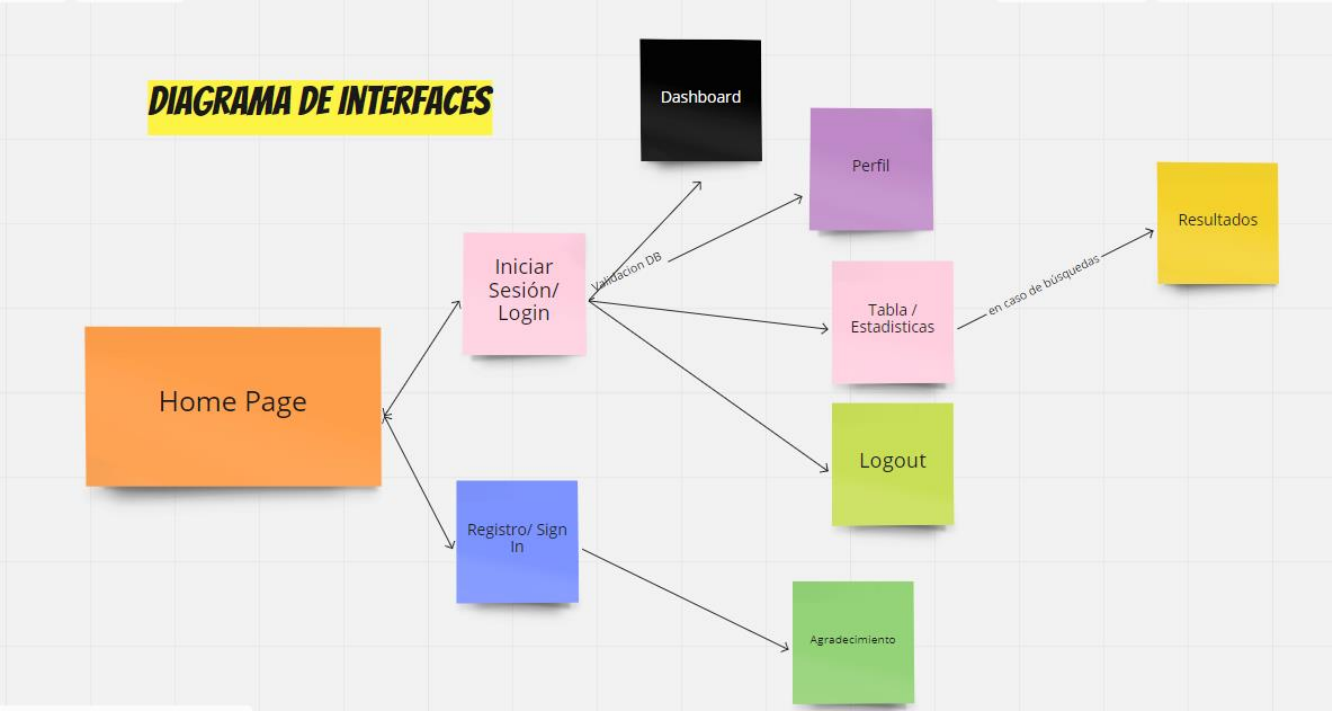
Cuando el **Usuario** acceda al sitio se le mostrará la ventana de bienvenida (**Vista**) y le mostrará las 2 opciones principales del sitio: Iniciar Sesión y Registrarse.

Supongamos que este es nuevo, así que dará clic en “Registrarse” y se le redirigirá a otra ventana con un formulario, el usuario deberá llenar los datos que se requieran y al dar “Enviar”, dicho formulario tendrá una serie de verificadores (**Controlador**), los cuales mandaran mensajes de error en caso de que algún dato no esté en el formato correcto o incluso si esta mal escrito, esto sucederá hasta que el formulario esté libre de errores. Una vez que suceda lo anterior, un código PHP (**Controlador**) se encargará de invocar a la Base de Datos (BD) (**Modelo**) para almacenar dichos datos y también para notificar a los administradores o entidades (en su dashboard) sobre dicho registro o petición. Mientras al **Usuario** le aparecerá una pantalla de agradecimiento y la leyenda “sus datos serán revisados”. Cabe aclarar que si los administradores aceptan al nuevo usuario sus datos permanecerán dentro de la BD y serán dados de alta a través de una API tipo REST (**Modelo**), de lo contrario serán eliminados

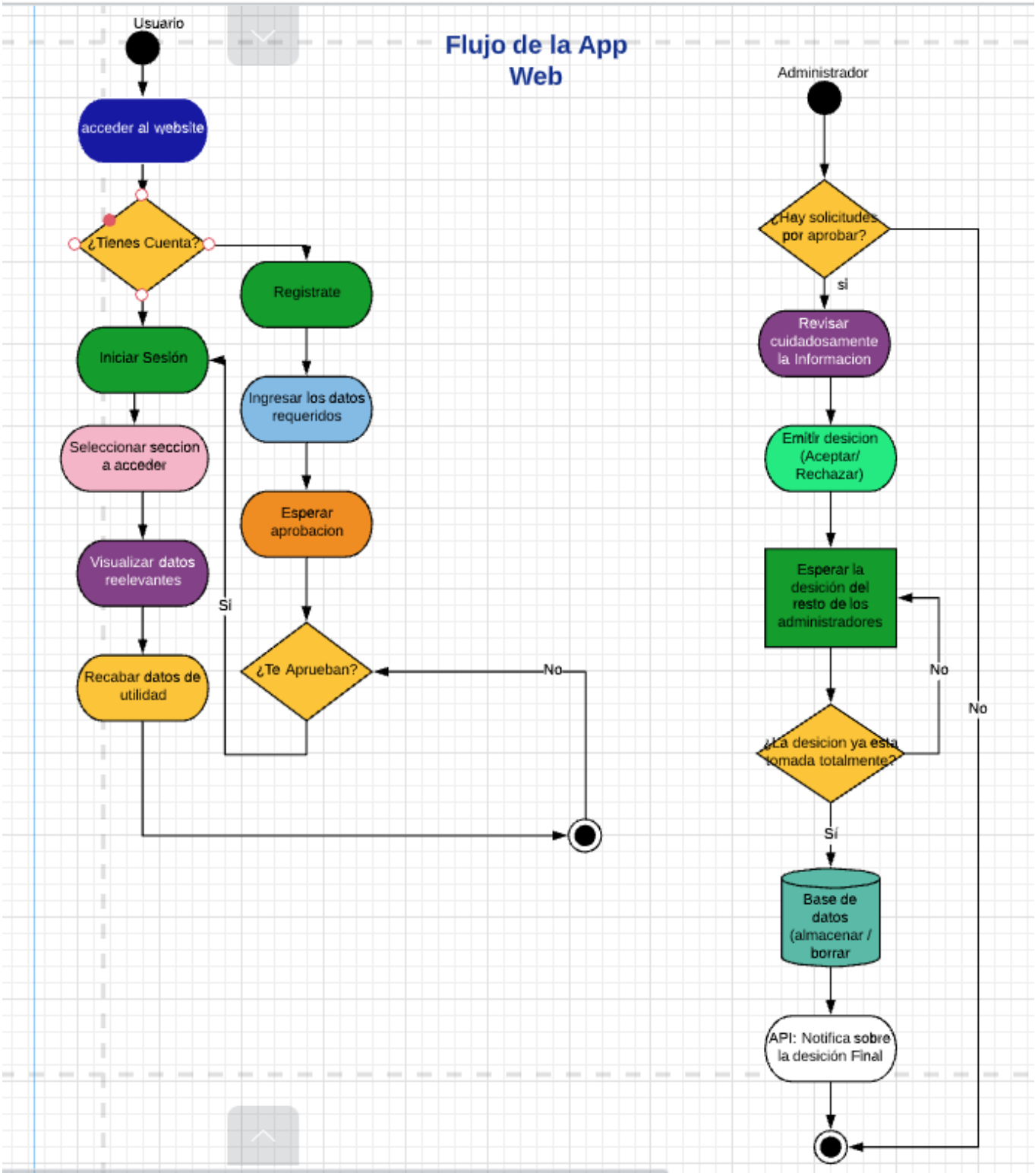
(**Controlador**), e independientemente de eso serán notificados cuando una decisión sea tomada y si son aceptados recibirían sus datos de acceso vía Email [tentativamente](**Vista**)

Mientras el **usuario** solicita iniciar sesión en la aplicación escribe usuario y contraseña a través de cualquier dispositivo con acceso a internet enviará esto a verificar si está registrado en la base de datos del sistema (**modelo**), y, si estos existen, el usuario podrá acceder a todas las opciones disponibles dentro del sitio, de lo contrario, mandará un mensaje de "usuario o contraseña incorrectos" (**controlador**) y en la pantalla de su dispositivo mostrará dicha información de manera entendible para el usuario (**vistas**).

## Diagrama de Interfaces




## Diagrama de Flujo


Flujo de la App Web
Usuario
acceder al website
¿Tienes Cuenta?
Registrate
Iniciar Sesión
Ingresar los datos requeridos
Seleccionar seccion a acceder
Esperar aprobacion
Visualizar datos reelevantes
Sí
¿Te Aprueban?
No
Recabar datos de utilidad
Administrador
¿Hay solicitudes por aprobar?
si
Revisar cuidadosamente la Informacion
Emitir desicion (Aceptar/ Rechazar)
Esperar la desición del resto de los administradores
No
¿La desicion ya esta tomada totalmente?
No
Sí
Base de datos (almacenar / borrar
API: Notifica sobre la desición Final