Reto Técnico (Node JS / React JS)

En esta ocasión se requiere de la creación de un aplicativo web el cual constará de venta de productos financieros de un banco. Para la realización de esta App nos gustaría que implementes todos tus conocimientos sobre programación. El código del proyecto debe estar creado en un repositorio GIT a su elección, funcional 100% y disponible para la elaboración de pruebas con los pasos para poder correr el proyecto.

En este desarrollo queremos que uses Node JS para el Backend; y en el Frontend usaremos la biblioteca de componentes React Js. Se puede usar como base de datos, MySQL o PostgreSQL (base de datos relacional)

***Lo que buscamos en la realización de la prueba es crear un Sistema de Logeo y generar un Crud.**

Ok, la información para el aplicativo es la siguiente.
Debe contar con un administrador para realizar las acciones CRUD de las ventas de los productos, usuarios y otros datos.

- Para ingresar al administrador debo iniciar sesión con email y contraseña por medio de un login.
- Para el sistema de logueo se deberá contar con un captcha que pueda validarse desde el back end de la aplicación
- Se mostrará un menú con las opciones Usuarios (Si soy administrador) y Radicar Venta

- Puedo dentro del administrador hacer CRUD de usuarios, estos deben tener:
  - ID (Único, puede ser autoincrementable)
  - Nombre (50 caracteres tipo texto obligatorio)
  - Correo electrónico (50 caracteres tipo texto con validaciones de correo válido obligatorio)
  - Contraseña (20 caracteres, encriptada, obligatorio)
  - Tipo de usuario (Lista desplegable con las opciones “Administrador”, “Asesor” obligatorio) (De ser posible una tabla relacional para los roles)
  - Fecha de creación (tipo fecha y hora obligatorio)
  - Fecha de actualización (tipo fecha y hora obligatorio)

- Para el CRUD de las ventas de productos, necesitamos los siguiente:

  - ID (Único, puede ser autoincrementable)
  - Producto (Lista desplegable con las opciones “Credito de Consumo”, “Libranza Libre Inversión”, “Tarjeta de Credito” obligatorio)
  - Cupo Solicitado (20 caracteres tipo miles ej: 1.000.000)
  - Franquicia (Lista desplegable con las opciones “AMEX”, “VISA”, “MASTERCARD” obligatorio si la opción es tarjeta de crédito, sino no se debe mostrar dicho campo)
  - Tasa (4 caracteres 2 numeros y 2 decimales ej: 10.58 obligatorio si la opción en el campo Producto es Credito de Consumo o Libranza Libre Inversión, sino no se debe mostrar dicho campo)
  - Fecha de creación de la venta (tipo fecha y hora obligatorio)
  - Usuario quien crea la venta (obligatorio, debe de guardarse el ID del usuario quien radicó el producto)
  - Fecha de actualización (tipo fecha y hora obligatorio)
  - Usuario quien actualiza la venta (obligatorio, Debe de guardarse el ID del usuario quien actualizó el producto)

- Al momento de entrar a la aplicación se podrá ver el listado de productos radicados, la estética de este lo dejamos a tu imaginación. Si el rol es Administrador podrá ver las ventas de todos los Usuarios, Si el rol es Asesor únicamente podrá ver sus ventas.
- Dentro del listado de productos se debe mostrar el nombre del producto, cupo solicitado, la fecha de la creación de la venta y el usuario que creo la venta, también tendrá un campo llamado acciones en donde se podrá visualizar, editar, eliminar la venta

- Se debe de mostrar arriba de la lista de productos la sumatoria del cupo solicitado de todos los productos radicados.

Opcional, pero es un plus:

- Una buena estética por lo menos en la parte de usuario final, puedes usar cualquier UI Framework.
- Implementación del uso de contenedores, como docker.
- Aplicación desplegada en AWS.
- Tener un módulo de back office de gestión de ventas (que todos los productos vengan con estado “Abierto”, se pueda modificar a estado “En Proceso” y por último “Finalizado” y tenga sus fechas y usuarios de quienes modifican dichos casos”
- Tener un módulo grafico de estadísticas (cantidad de ventas realizadas por asesor, sumatoria de cupos por producto, cantidad de ventas por fecha de creación, etc)

Para el desarrollo de esta App es necesario:

- Usar **JWT** para el login de usuarios.
- Recuerda usar **React JS**, para el Frontend
- Recuerda usar **Node JS,** para el Back End
- Usar una base de datos relacional (puede ser MySQL o PostgreSQL)
- Para el login y los formularios usar **validaciones** tanto en Frontend como en Backend
- Programación orientada a objetos, Clases y Modelos es indispensable, además de uso de componentes funcionales, redux, hooks.
- Cualquier otra característica funcional es un plus y se tomará en cuenta.