

Universidad Tecnológica De Bolívar

Blockchain App

Integrantes:

- **Sergio Sánchez**
- **Juan Agamez**

María Eugenia.

Ingeniería de Software.

22 - Marzo - 2022.

Tabla de Contenido

1. Introducción	3
1.1 Descripción general	3
1.2 Descripción de los requerimientos	3
2. Descripción de los Stakeholders	4
3. Composición de la arquitectura	5
3.1 Objetivos de la arquitectura	5
3.2 Consideraciones de la arquitectura	5
4. Esquemas	6
4.1 Diagramas de secuencia.....	6 - 7 - 8
4.2 Casos de Uso	9
5. Diseño Interfaz de Usuario	
5.1 Log-in	10
5.2 Sign-up	10

1. Introducción

Con esta arquitectura, se busca generar una transformación en el sistema de cómo se desarrolla un registro de información que se genera en un entorno de producción y consumo de energía. Proveniente de fuentes renovables, en nuestro caso la fuente de generación de información hace alusión a lo producido y demandado de parques solares.

Con lo cual el principal objetivo de este proyecto es desarrollar una aplicación la cual se le pueda mostrar a los usuarios información certera en tiempo real de los procesos de producción, para de esta forma nuestros clientes puedan tomar decisiones sobre si vender o comprar energía dependiendo de cómo se comporte el mercado energético sobre el cual estamos enfocados.

1.1 Descripción General

En esta arquitectura se detalla la composición del software que da solución al manejo de datos suministrados por diferentes parques solares ubicados en diferentes Zonas de Bolívar departamento en el norte de Colombia. Se busca con este software dar seguridad al usuario de que los diferentes procesos que se realizarán con estos datos estarán asegurados de tal forma que se presenten la menor cantidad de percances con dichos datos; Se busca también que cada usuario pueda acceder a los diferentes datos que son suministrados desde el parque solar y consumido por una api creada y usada en este software.

Para las vistas en el que los usuarios podrán tener un resumen de las transacciones de datos que se realizan entre los parques solares y las diferentes entidades finales, se realiza una analítica de datos que permite la correcta visualización de dichos datos.

1.2 Descripción de los requerimientos

1.2.1. Requerimientos Funcionales

- Registrar el historial de transacciones de datos realizadas entre el productor y el consumidor.
- Utilizar datos de consumo eléctrico reales.
- Implementar una arquitectura en capas o microservicios.
- Garantizar un acceso al sistema con doble verificación de seguridad.
- Garantizar seguridad para almacenar y administrar los datos.
- Proporcionar a los clientes esquemas de precios según la matriz de suministro de energía.
- Controlar el acceso de los usuarios a la información del sistema desde una vista de administrador.
- Conectar al sistema con API de Exia para obtener la data.
- Analizar los datos y crear alertas cuando se registren anomalías en la data.

- Visualizar la información de la data mediante graficas con valores puntuales (día, hora, minuto, semana, mes), en tablas con valores numéricos y líneas de tiempo.
- Gestionar el registro, acceso y manejo de recursos en el sistema.

1.2.2. Requerimientos No Funcionales

- Todas las comunicaciones externas entre la base de datos, la página web y el cliente debe estar cifrado en algoritmo RSA.
- Análisis descriptivo y prescriptivo de rendimiento.

1.2.3 Requerimientos Del Sistema

- Realizar el despliegue de la página web.
- Modelar un sistema orientado a Servicios.
- El sistema debe tener características Responsive.
- El desarrollo del Backend se realizará con Django en Python 3.
- El desarrollo del Frontend será con react.js.
- Soporte de concurrencia.
- El sistema debe ser acoplable a los diferentes navegadores y SO.
- El sistema debe ser escalable.

2. Descripción de los Stakeholders del sistema

Todos aquellos que a través del endpoint principal de la aplicación accedan a la vista principal de dicha aplicación podrán acceder a datos almacenados de en la base de datos, más sin embargo, habrán datos y acciones que solo podrán ser revisados y/o modificados/eliminados/creados por ciertos usuarios, los cuales hayan recibido por parte de un usuario cuyo rol sea el de “Admin”.

- **Admin:** Este es el que asigna el rol a cada nuevo usuario que se registra en la aplicación, este posee permisos para poder crear, modificar y/o eliminar datos de la base de datos en caso tal se haya violado alguna política que se haya impuesto para esta aplicación, o en dado caso un usuario con otro usuario con otro rol lo haya solicitado y dicha solicitud sea admitida.
- **User_Generator:** Este rol le permite a un usuario generar nuevas ofertas de recursos energéticos, en estas ofertas puede especificar la cantidad de recursos energéticos ofrecidos así como también el precio de dichas ofertas.
- **User_Consumer:** Este rol le permite a un usuario visualizar las ofertas que se encuentran realizadas por otro usuario cuyo rol puede ser: User_Generator o Prosumer.
- Este usuario puede leer más acerca de la oferta realizada por otro usuario y realizar la compra de dicha oferta..
- **User_prosumer:** Este rol le permite al usuario ver las cantidades de recursos que genera, así como también la cantidad de recursos que compra, en dado caso este

usuario se encuentre en una situación en la que no genera lo suficiente como para poder cubrir la demanda que requieren sus ventas

- **Invited:** Este rol le permite al usuario solo ver las ofertas de los usuarios con rol “User_Generator” y “Prosumer”, esta es la única acción que puede realizar un usuario con este rol.

Composición de la arquitectura

2.1 Objetivos de la arquitectura

- Brindar un canal de comunicación a través del cual se puedan generar de manera eficiente las diferentes transacciones.
- Garantizar la veracidad de la información que se maneja en cada una de las transacciones.
- Crear un canal seguro de flujo de información.

2.2 Consideraciones de la arquitectura

Con la implementación de esta arquitectura se busca generar un sistema con un alto nivel de seguridad el cual ofrezca a los usuarios la seguridad de que su información, así como la de todas y cada una de las transacciones que realicen mantengan su veracidad en cualquier momento. Además de ofrecerles una alternativa de fácil manejo y que cumpla con lo requerido al momento de llevar un registro de producción y consumo.

2.3 Descripción de la Arquitectura

La arquitectura de software

3. Esquemas

3.1 Diagramas de secuencia

4.1.1 Access.

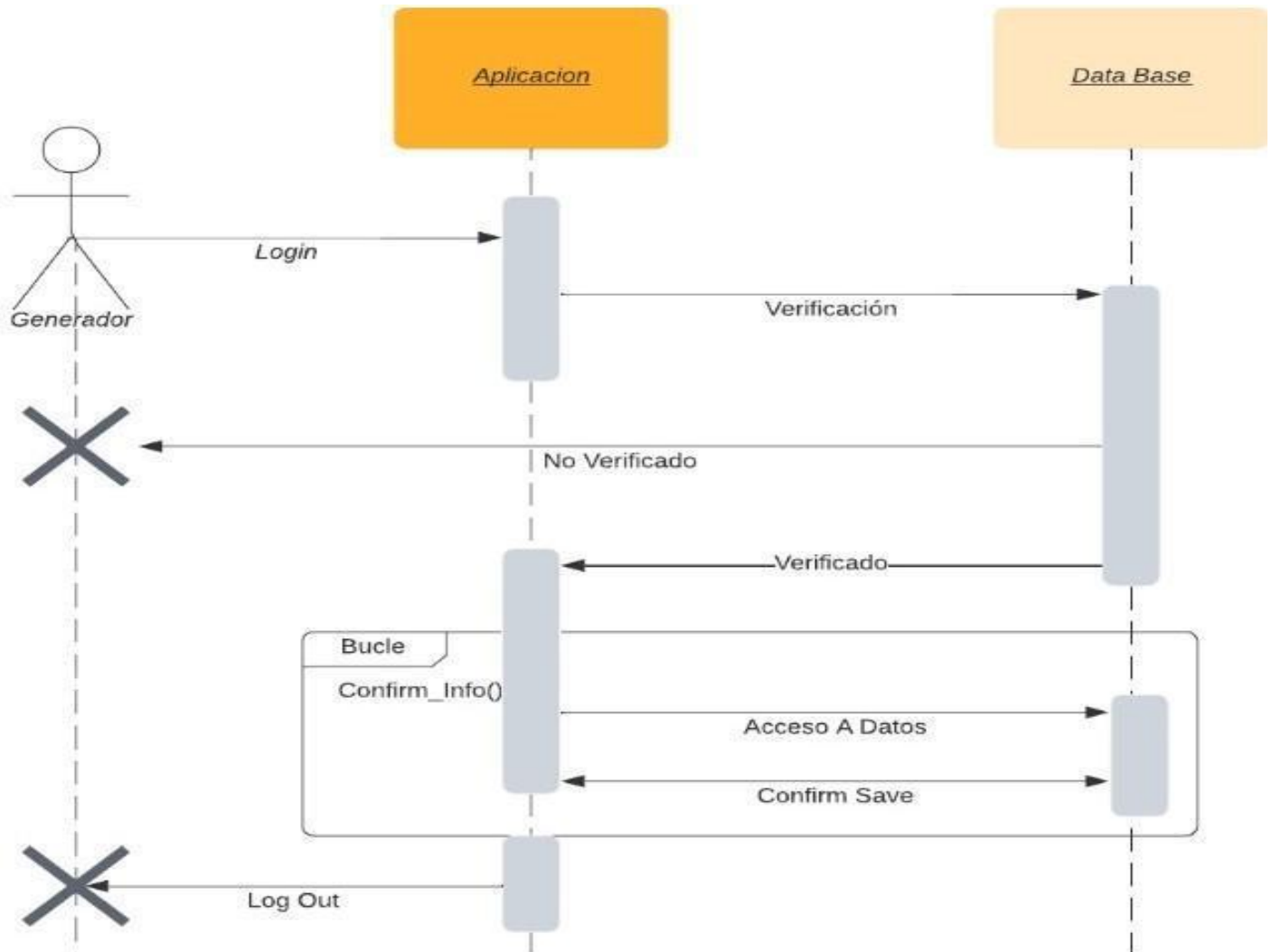


Figura 1: Los usuarios que posean un rol “Generador”, podrán generar ofertas de energía, se guarda esta oferta en la base de datos para mantener una constancia de dicha oferta con la cual pueda un usuario con rol Consumidor adquirir esta oferta y pasar al proceso de compra.

4.1.2. Log-in.

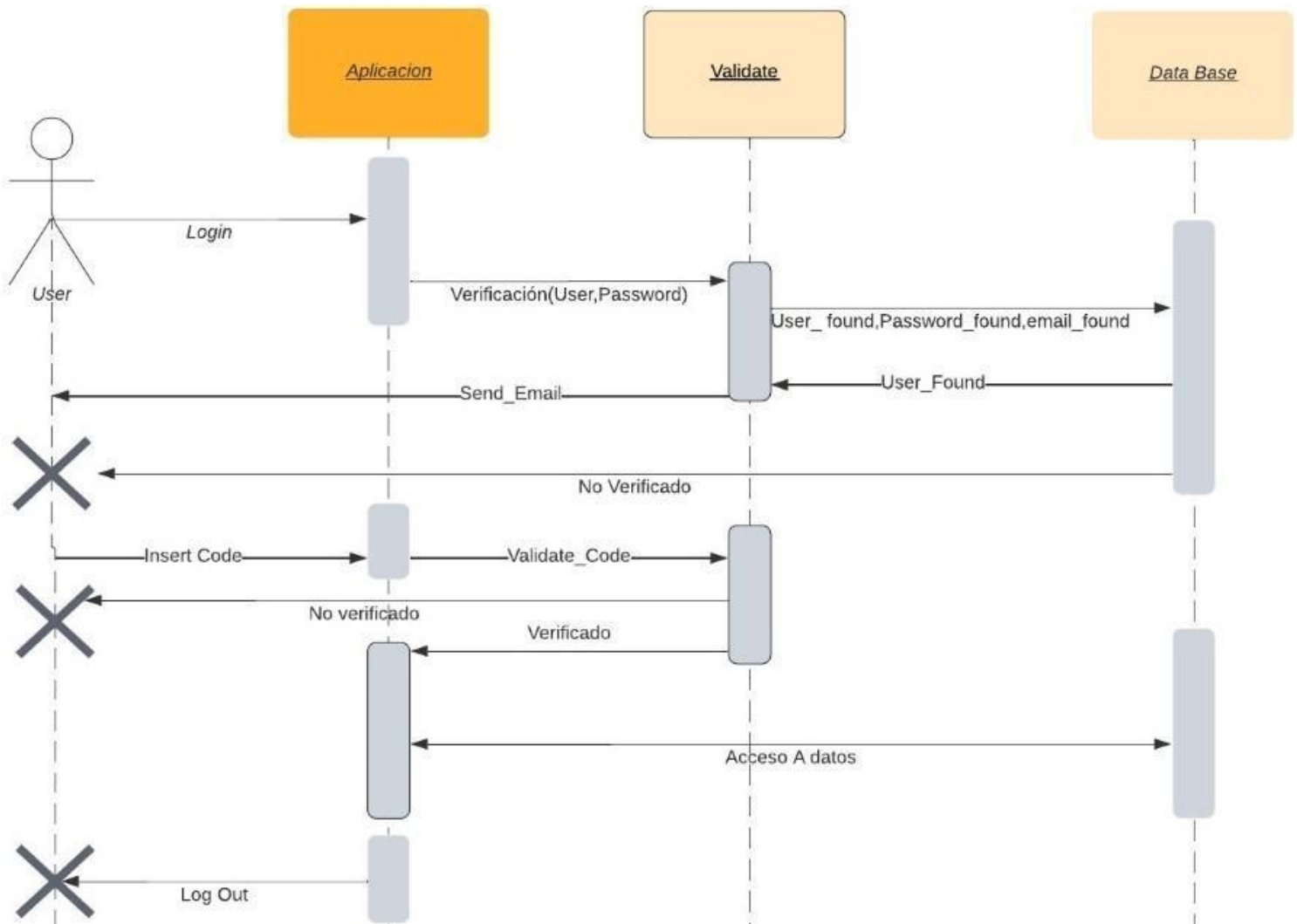


Figura 2: Para mantener un alto estándar de seguridad, se generará un log-in de 2 pasos, por medio del cual, un usuario podrá entrar a la aplicación siempre y cuando se haya verificado su ingreso por medio de un correo electrónico enviado a este, gracias a la información anteriormente suministrada por el usuario.

4.1.3. Rol Consumer.

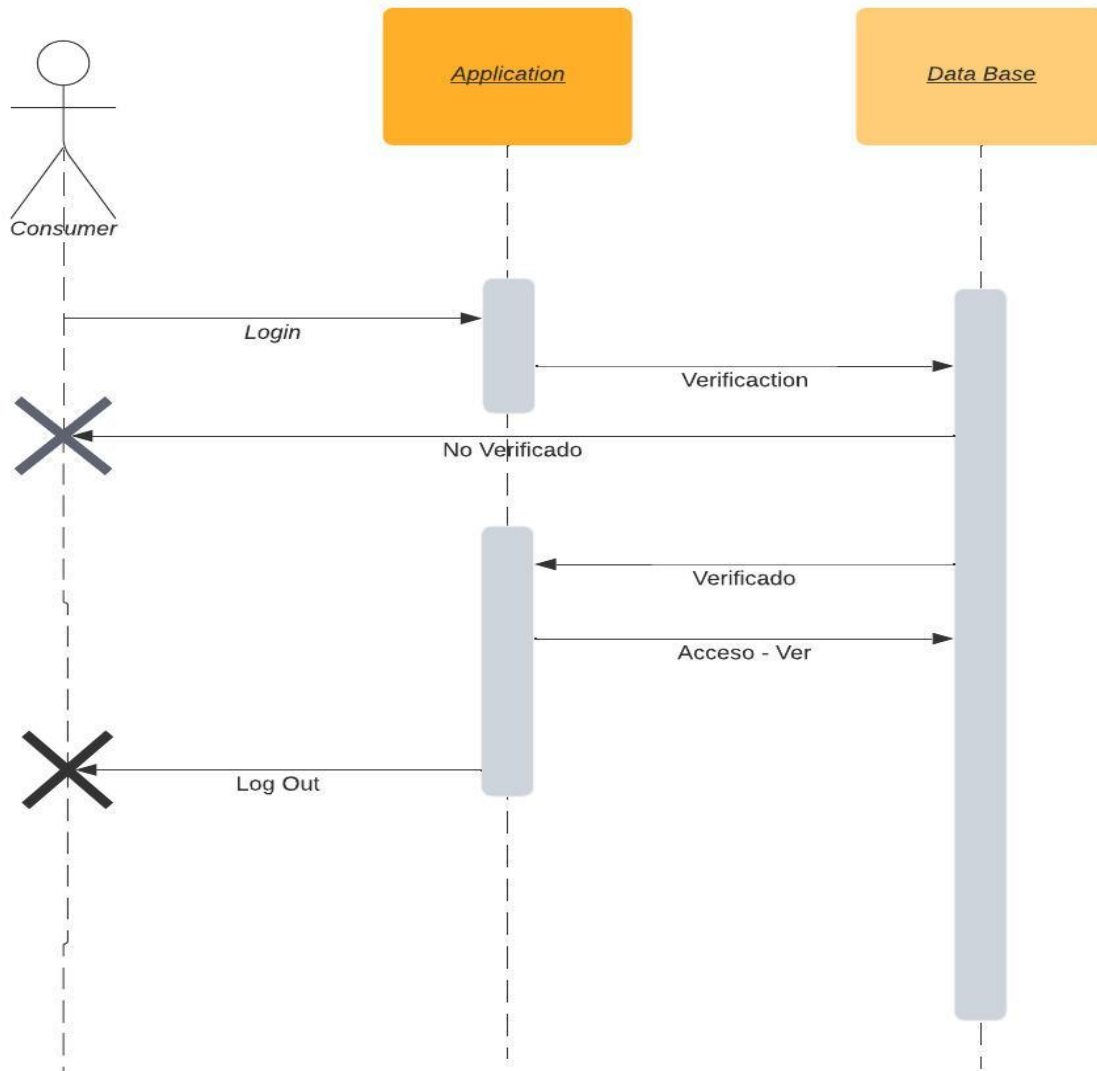


Figura 3: Se muestra el diagrama de secuencia del rol “Consumer”, éste sólo podrá ver el comportamiento que tiene la energía que está consumiendo, este análisis se realizará de forma gráfica.

4.2 Use of case

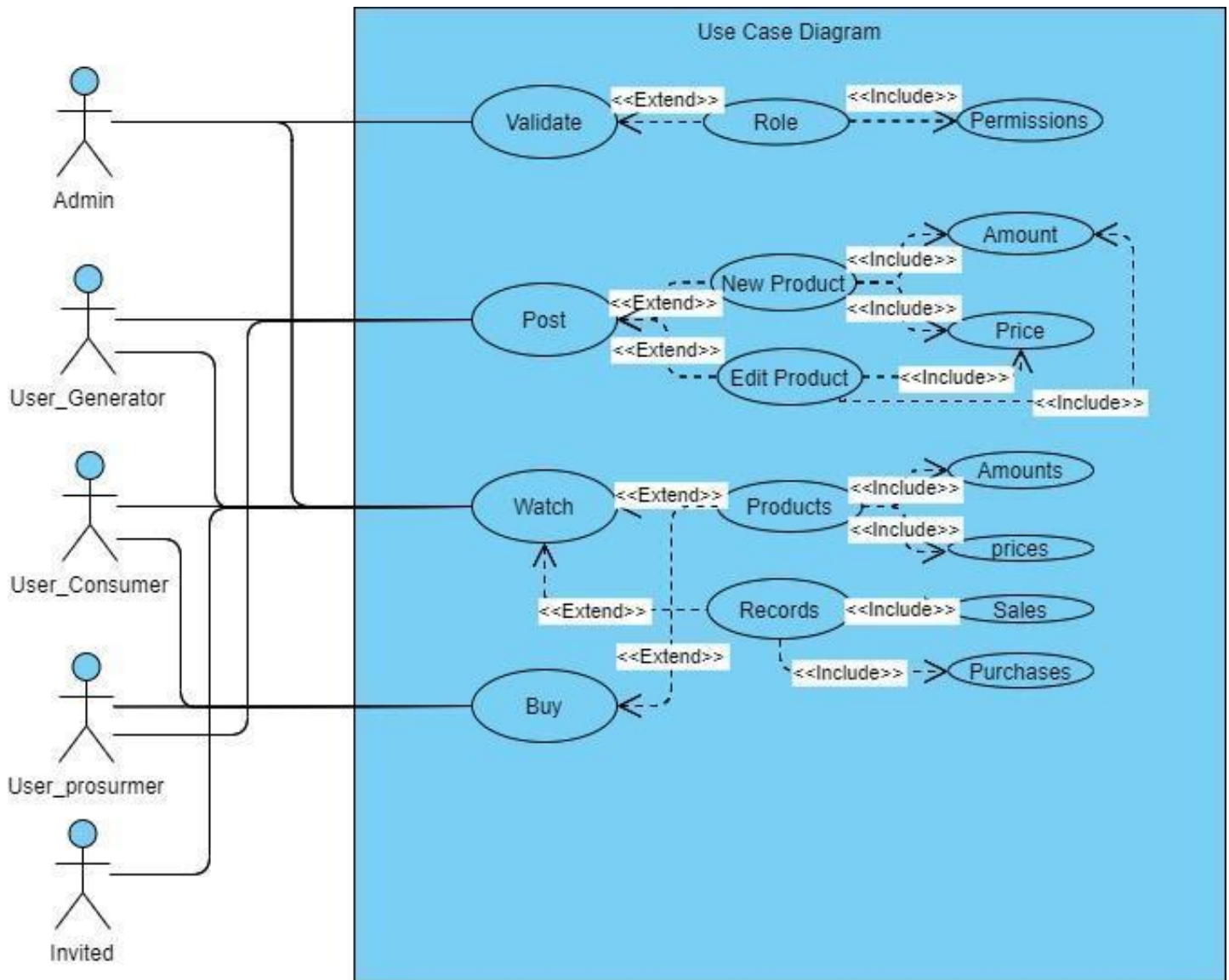


Figura 4: Diagrama de caso de uso donde se pueden ver los diferentes usuarios que interactuaran con el producto una vez implementado y los roles y permisos que cada uno puede ejercer dentro del entorno.

4.2.1. Admin use of case

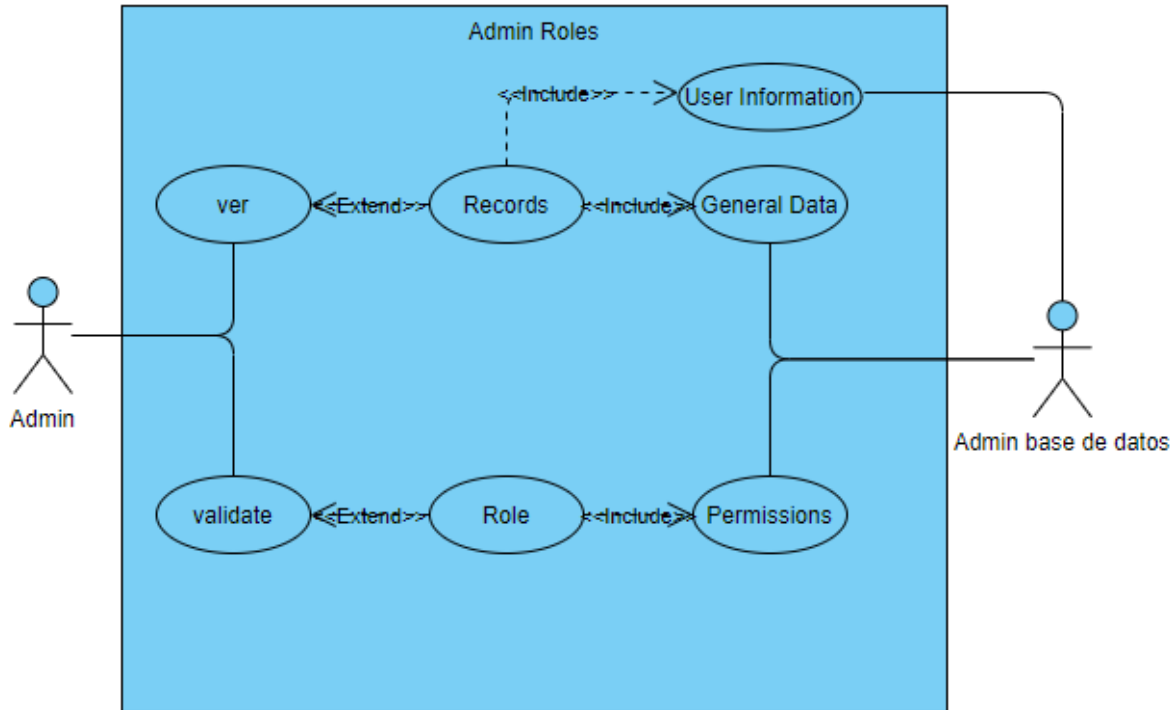


Figura 5. Diagrama de caso de uso del rol Admin, en donde se describen las actividades que este usuario puede realizar, como validar datos ingresados por los usuarios, asignar roles y otorgar ciertos permisos.

4.4 Diagrama de clases

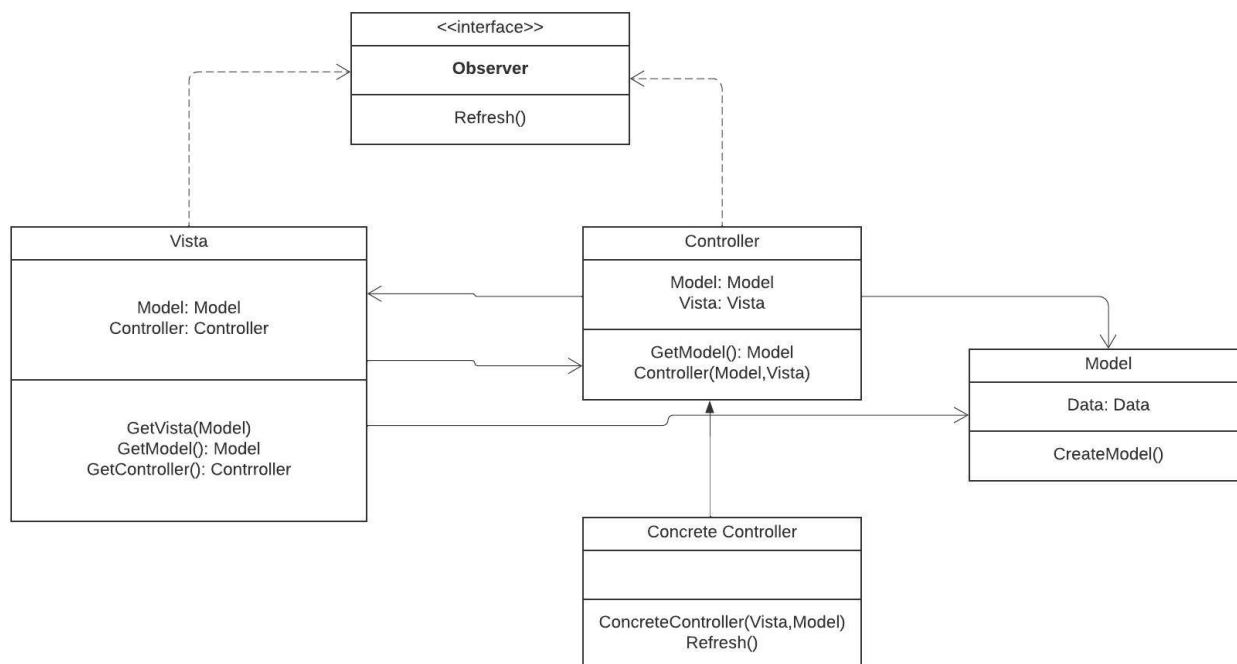


Figura 6.

4.3 Tabla de Roles

Resumen Privilegio	Rol Predeterminado				
	Consumer	Generator	Prosumer	Invited	Admin
Crear	●	●	●	●	●
Modificar	●	●	●	●	●
Eliminar	●	●	●	●	●
Ver	●	●	●	●	●
Validar	●	●	●	●	●

Figura 6. Tabla que describe detalladamente los permisos que cada usuario con sus respectivos roles le permiten.

4.4 Descripciones del Sistema

	System	Note
Programming Language (Frontend)	Javascript	Es un lenguaje de programación interpretado, cuyo uso se da principalmente en la web, tanto del lado del servidor como del usuario.
Framework (Frontend)	React	Biblioteca escrita en javascript, creada para facilitar la creación de componentes que puedan interactuar entre sí y que sean reutilizables.
Programming Language(Backend)	Python	Es un lenguaje de programación de alto nivel utilizado tanto del lado del servidor como del lado del usuario; Así como también es utilizado ampliamente en la ciencia de datos.
Framework (Backend)	Django	Este es un framework que está escrito en python, es usado principalmente del lado del servidor, este framework posee características de seguridad firmes contra ataques de SQL injection o XSS.
Database	Mysql	Es un sistema de gestión de base de datos relacionales.
Data Analytic	Python	La analítica de datos se hará en Python, con su librería “Pandas”, la cual permitirá realizar una clara y excelente analítica de los datos que son suministrados desde los parques solares.

5. Diseño Interfaz de Usuario

5.1.Vista General Log-in

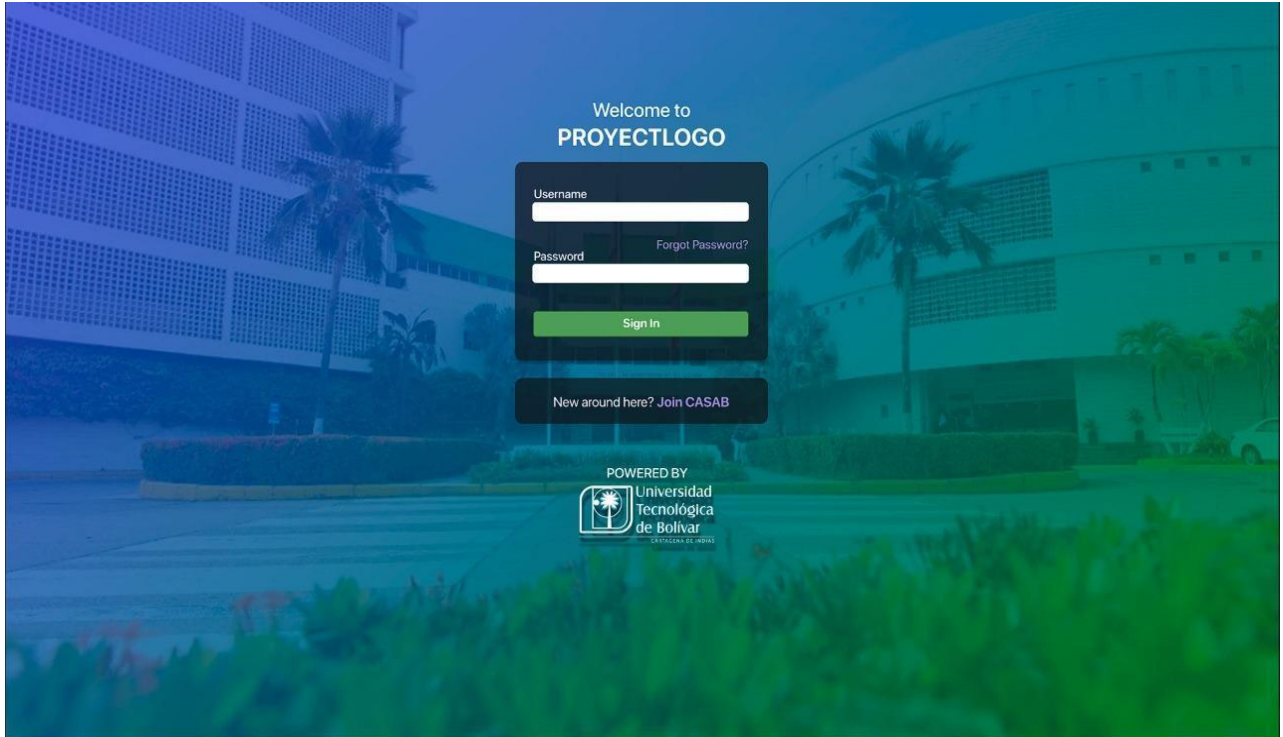
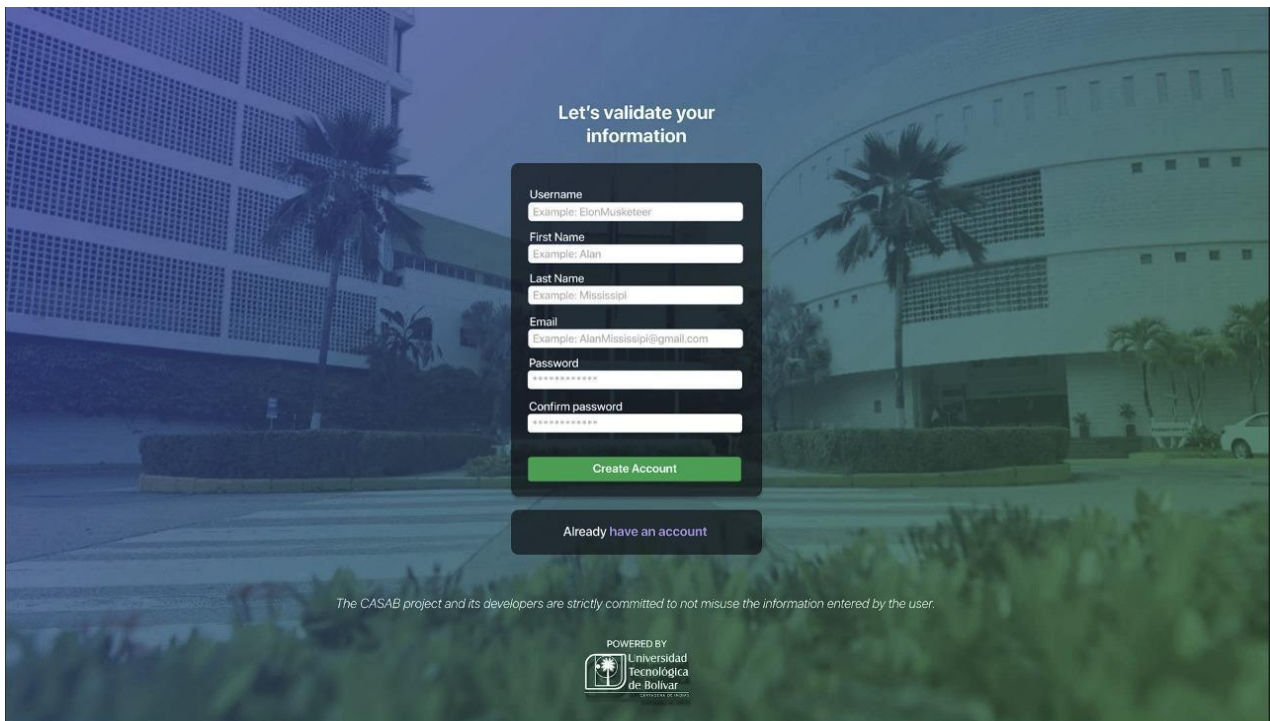
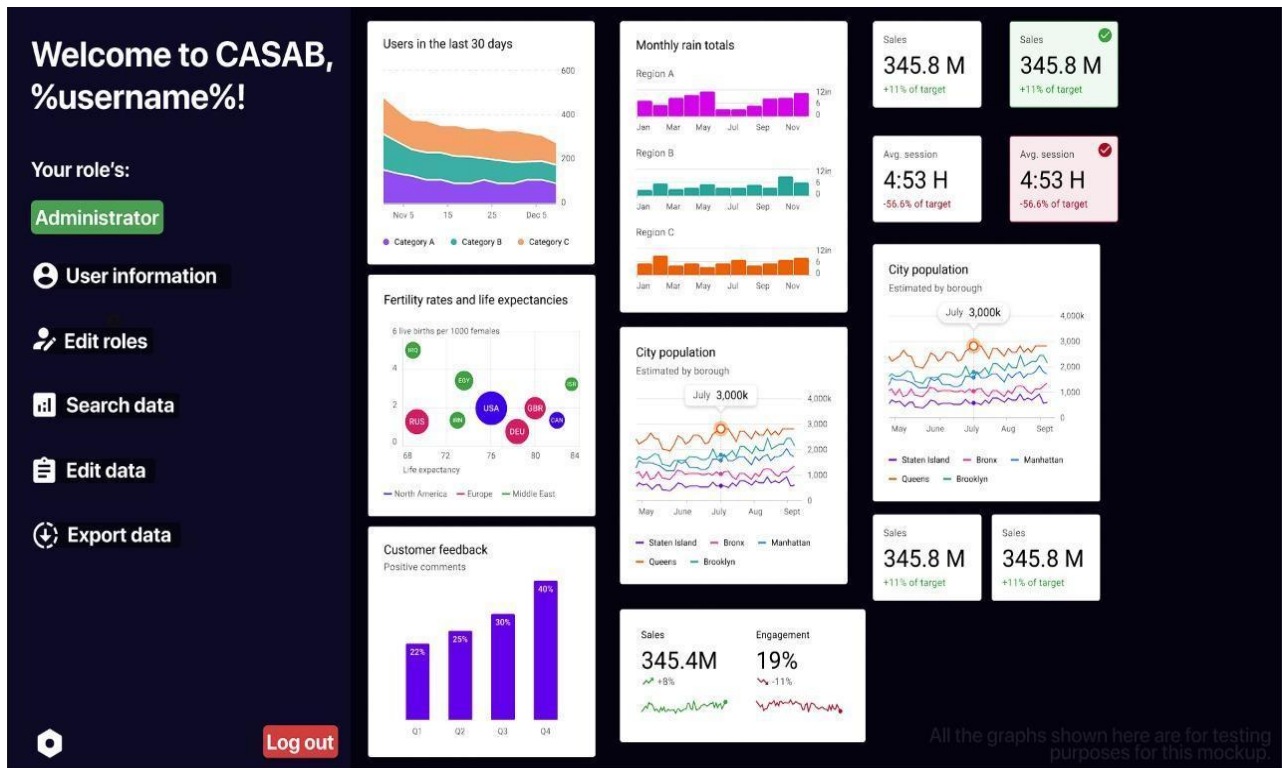


Figura 8.

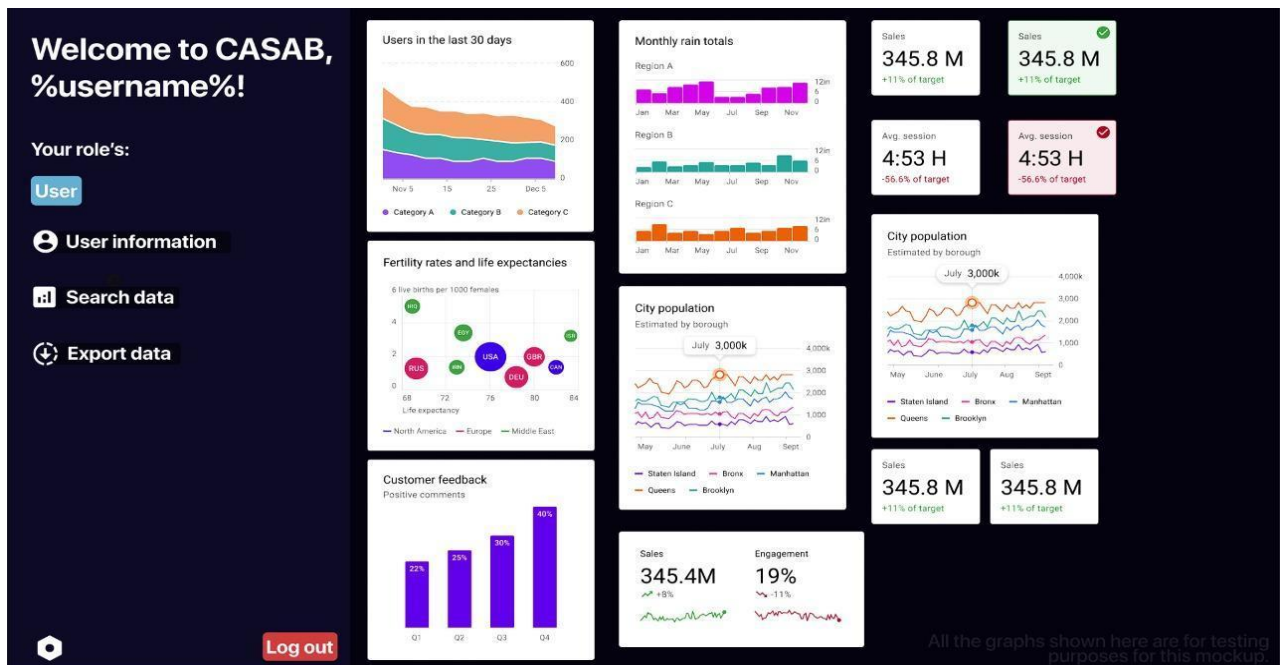
5.2. Vista general Sign-up



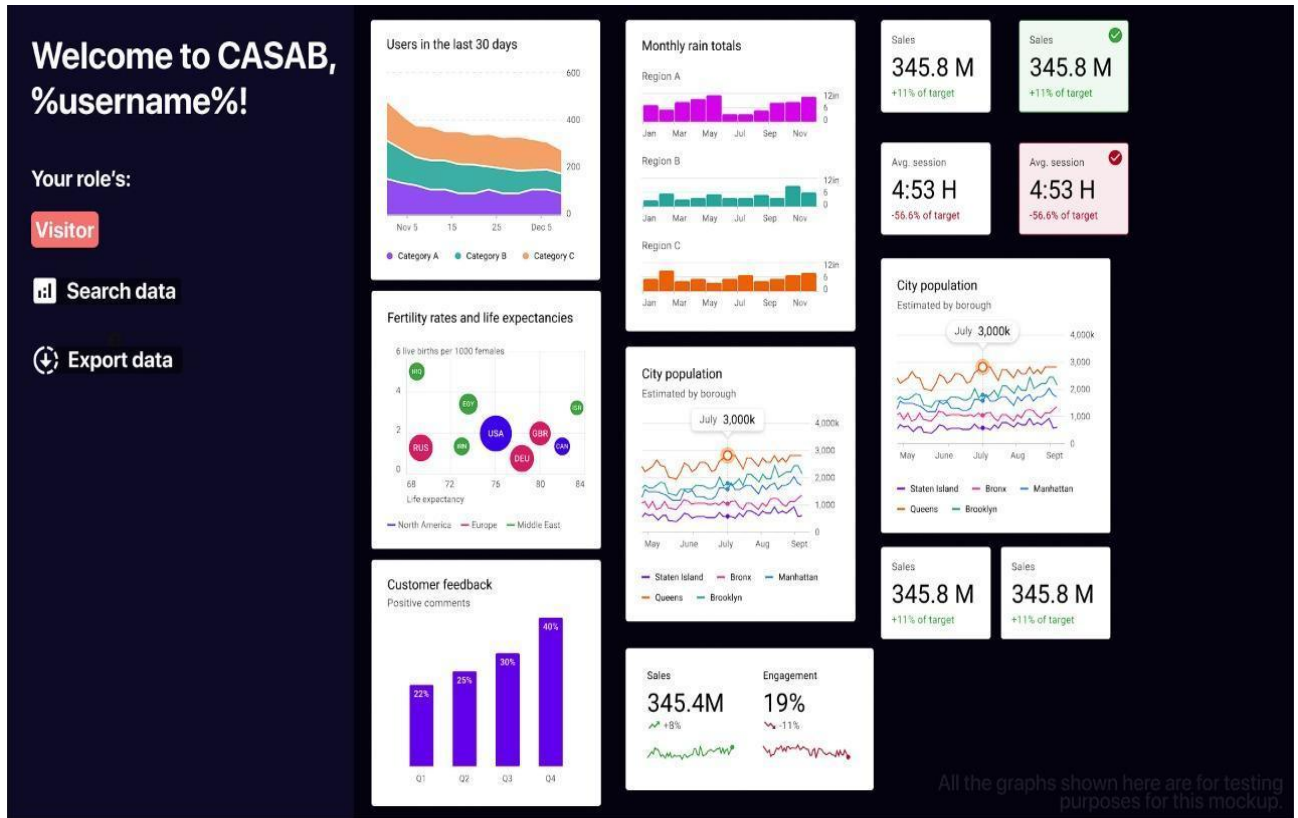
5.3. Vista General “Usuario Rol Administrador”



5.4. Vista General “Usuario Rol Producer/Consumer/Prosumer”



5.5. Vista General “Usuario Rol Visitor”



El diseño de estas interfaces de usuario son 100% veraces de cómo se implementarán, aunque el maquetado puede estar sujeto a cambios al momento de su desarrollo en los respectivos entornos en los cuales se realicen.