# PRUEBA TECNICA APUESTA TOTAL

**Postulante: SAMUEL ISAAC ROMÁN VELASQUEZ**

Contenido

[PRUEBA TECNICA APUESTA TOTAL 1](#_Toc167423323)

[ANALISIS Y ESPECIFICACIÓN DE REQUERIMIENTOS 2](#_Toc167423324)

[DISEÑO PRELIMINAR DEL MODELO DE BASE DE DATOS 3](#_Toc167423325)

[DESARROLLO DE LA APLICACIÒN WEB 5](#_Toc167423326)

[USUARIOS Y CONTRASEÑA DE PRUEBA 15](#_Toc167423327)

## ANALISIS Y ESPECIFICACIÓN DE REQUERIMIENTOS

1.- Requerimientos funcionales de todos los tipos de usuarios del sistema

|  |  |  |
| --- | --- | --- |
| Código | Función | Descripción |
| RF-01 | Acceso a la interfaz de Login | Cualquier persona puede acceder al Login de sin ningún impedimento. |
| RF-02 | Mostrar la opción de registrarse | Cualquier persona puede crear un nuevo registro de cliente. |
| RF-03 | Mostrar la opción de restablecer contraseña | Todos los usuarios que están registrados tienen la posibilidad de restablecer su contraseña mediante verificación del su correo asociado. |

2.- Requerimientos funcionales del Bloque del Cliente

|  |  |  |
| --- | --- | --- |
| Código | Función | Descripción |
| RF-04 | Acceso a la interfaz de inicio del Cliente | Se muestra un panel de inicio del cliente donde se le presenta información relevante, como la cantidad de recargas, su saldo, etc. |
| RF-05 | Mostrar historial de recargas | Se muestra una tabla con el historial de todas sus recargas registradas. |
| RF-06 | Mostrar historial de apuestas | Se muestra una tabla con el historial de todas sus apuestas realizadas. |
| RF-07 | Acceso al módulo de canales | Se muestra los accesos directos a los canales por donde el usuario puede enviar sus vauchers para que un asesor realice una recarga a su cuenta. |

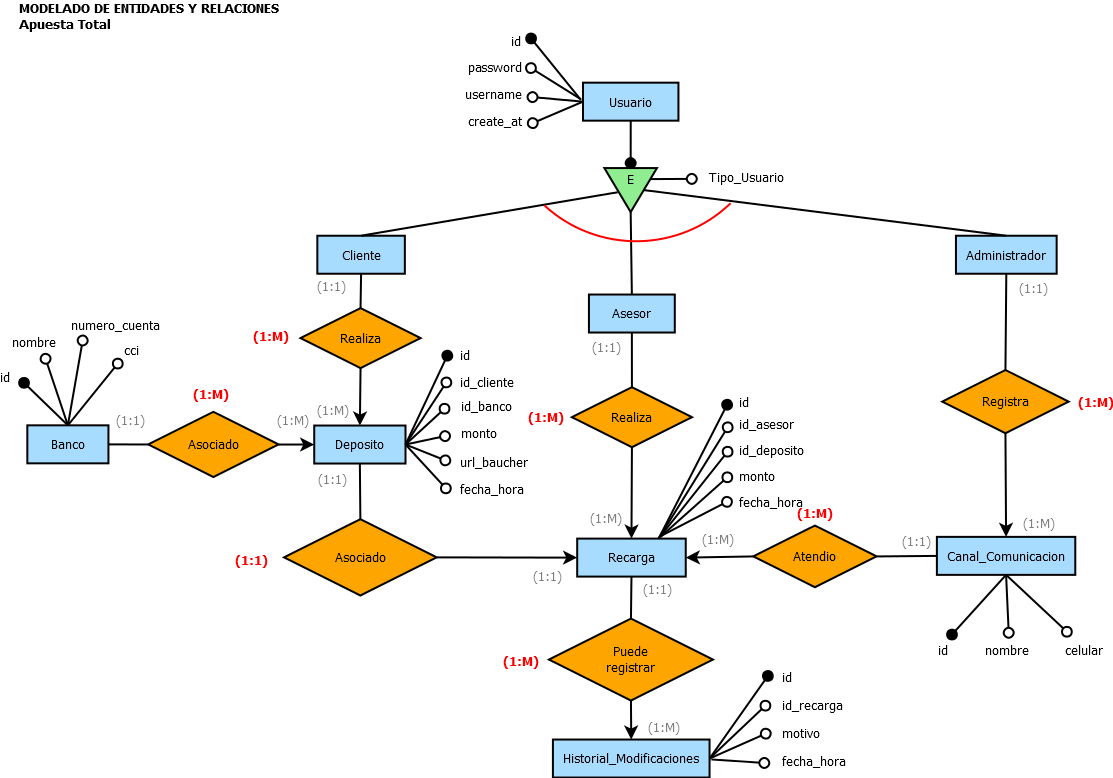
3.- Requerimientos funcionales del Bloque del Asesor

|  |  |  |
| --- | --- | --- |
| Código | Función | Descripción |
| RF-08 | Acceso a la interfaz de inicio del Asesor | Se muestra un panel de inicio del asesor donde se le presenta información relevante. |
| RF-09 | Mostrar historial de recargas aprobadas | Se muestra una lista con todas las recargas que el asesor ha realizado. |
| RF-10 | Mostrar historial de recargas Pendientes de aprobación | Se muestra una lista con todas las recargas que están pendientes de aprobación. |
| RF-11 | Mostrar Opción de editado de recarga | Cada recarga tiene habilitada la opción de editar el monto asignado. |
| RF-12 | Mostrar Opción de anular recarga | Cada recarga tiene habilitado la opción de anular la recarga, por consiguiente, se reduce el saldo del cliente. |
| RF-13 | Mostrar Modulo de Reportería | Se muestra un módulo de reportes. |
| RF-14 | Recargar Saldo a cliente | Se muestra un módulo donde se puede recargar el saldo respectivo al cliente mediante su código, la foto del baucher, indicando además del canal de atención, el banco por el cual se hizo la transacción y el monto correspondiente. |

4.- Requerimientos funcionales del Bloque del Administrador

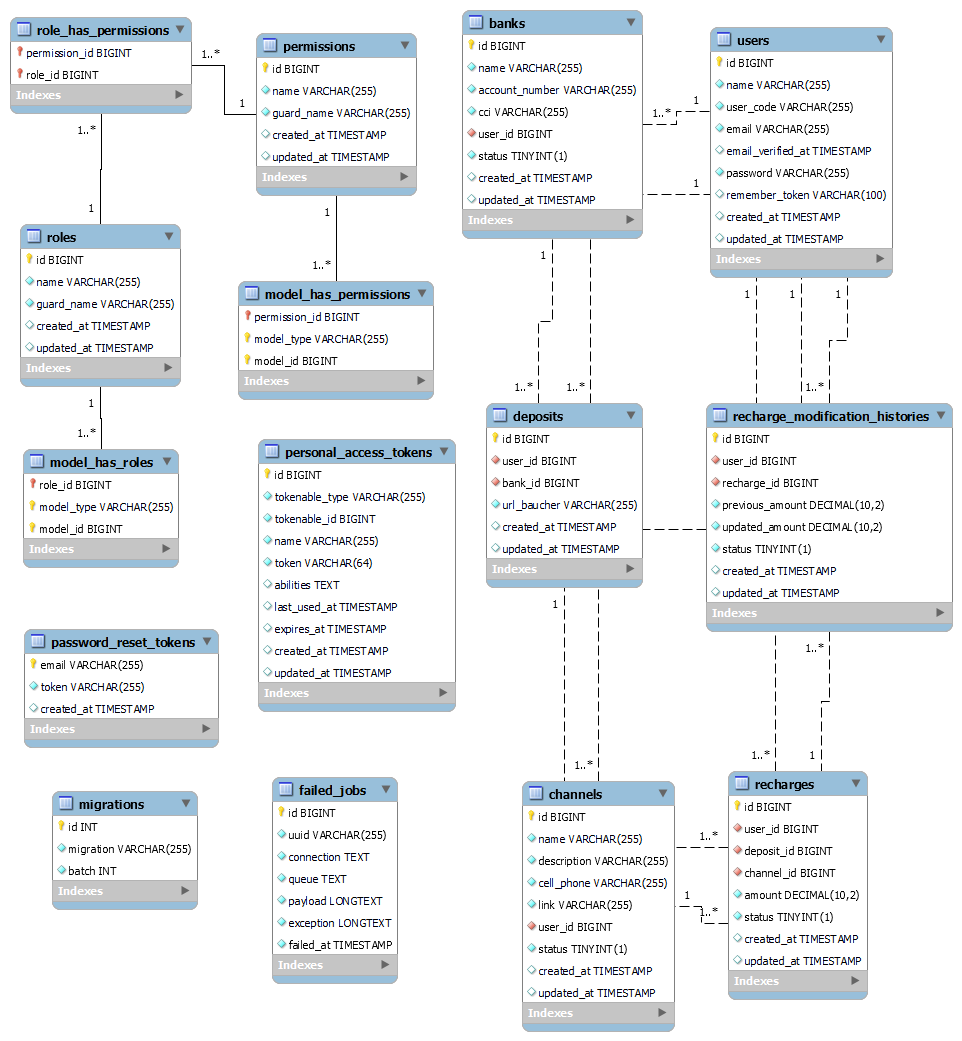
|  |  |  |
| --- | --- | --- |
| Código | Función | Descripción |
| RF-15 | Acceso a la interfaz de inicio del Administrador | Se muestra un panel de inicio del administrador donde se le presenta información relevante. |
| RF-16 | Mostrar historial de todas las recargas aprobadas | Se muestra una lista con todas las recargas de todos los asesores. |
| RF-17 | Mostrar historial de recargas Pendientes de aprobación | Se muestra una lista con todas las recargas que están pendientes de aprobación. |
| RF-18 | Mostrar Opción de editado de recarga | Cada recarga tiene habilitada la opción de editar el monto asignado. |
| RF-19 | Mostrar Opción de anular recarga | Cada recarga tiene habilitado la opción de anular la recarga, por consiguiente, se reduce el saldo del cliente. |
| RF-20 | Mostrar Modulo de Reportería | Se muestra un módulo de reportes. |
| RF-21 | Mostrar Modulo de Clientes | Se muestra una lista de todos los clientes registrados |
| RF-22 | Mostrar Asesores | Se muestra una lista de todos los asesores registrados |
| RF-23 | Modulo para crear un nuevo Asesor | Se muestra el modulo para poder crear un nuevo asesor. |

## DISEÑO PRELIMINAR DEL MODELO DE BASE DE DATOS



Al considerar los requerimientos iniciales y al analizar la mejor manera de almacenar la información generada, se diseñó un modelo de base de datos inicial. Este modelo incluyó la entidad "usuario", la cual poseía especificaciones de roles tales como administrador, asesor y cliente. Cada uno de estos roles estaba asociado a otras entidades; por ejemplo, un cliente podía realizar un depósito el cual estaría vinculado a un banco específico. Además, dicho depósito estaba relacionado con una recarga que era generada por un asesor. Este último tenía la capacidad de crear la recarga y modificar sus montos, lo que generaba un historial de modificaciones asociado a la recarga. Además, esta recarga era gestionada a través de un canal de comunicación, el cual podía ser activado por un administrador.

Después de haber realizado el modelamiento inicial, pude mejorar aún más el diseño mediante la normalización de la base de datos. El resultado de este proceso es el siguiente esquema de base de datos:



En el nuevo esquema de base de datos resultante, se observa la creación de tablas adicionales en comparación con el modelo inicial. Se implementó un sistema de roles y permisos para dar soporte a los tres tipos de usuarios identificados durante el análisis de requerimientos. Respecto a las tablas que aún no están relacionadas, estas corresponden a tablas generadas por el framework Laravel para el manejo de tokens y control de migraciones asociadas a la creación de la base de datos.

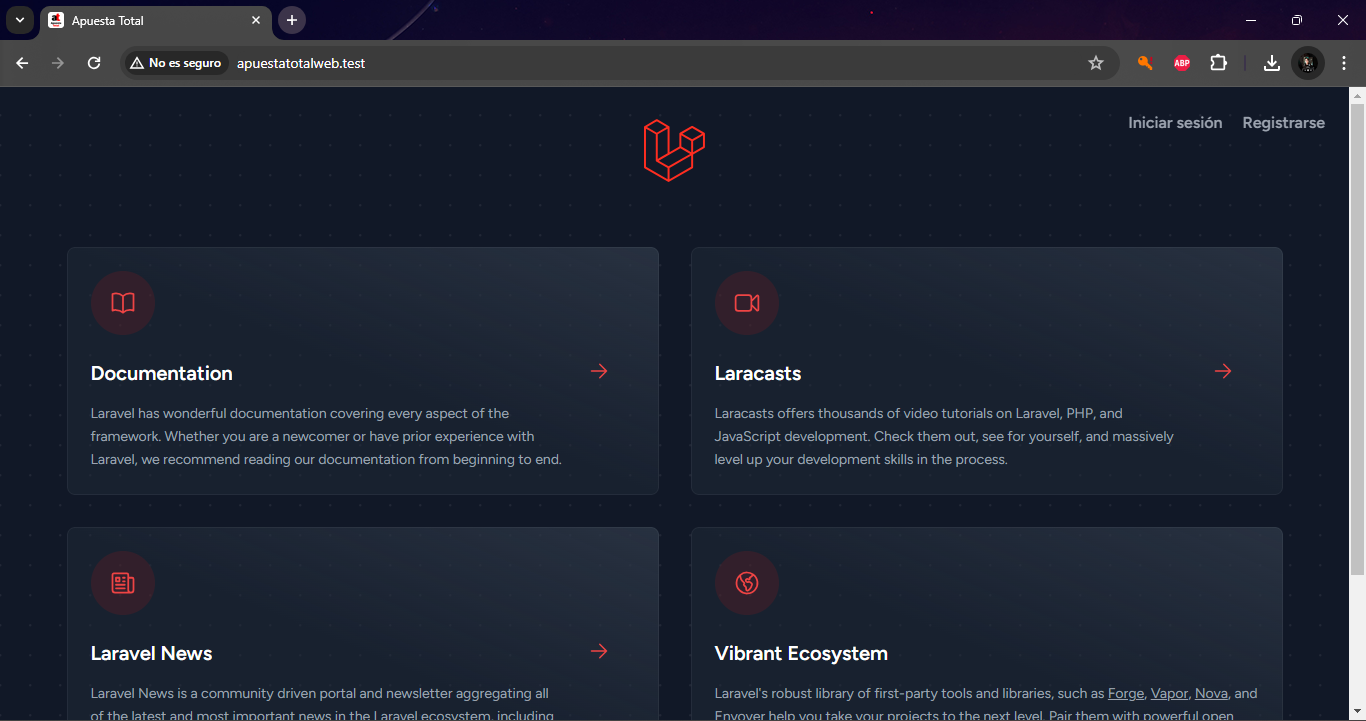
Aunque el diseño actual de la base de datos cumple con los requerimientos iniciales, todavía se puede mejorar y normalizar aún más. Se podría considerar establecer relaciones adicionales con otros tipos de entidades, como por ejemplo una tabla que gestione el historial de apuestas del cliente.

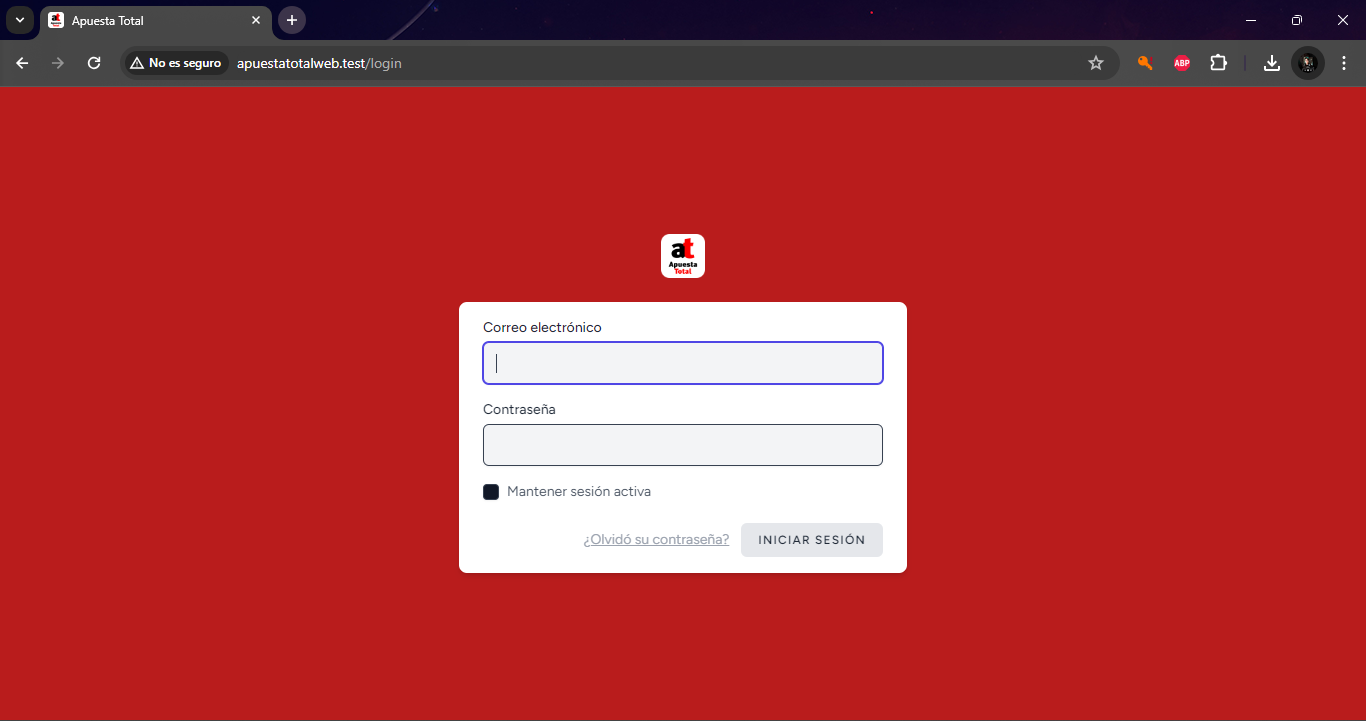
Es importante destacar que el modelo presentado está normalizado, lo que se refleja en la segregación de las distintas tablas para mantener la integridad y eficiencia de la base de datos.

## DESARROLLO DE LA APLICACIÒN WEB

El desarrollo de la aplicación en términos generales fue el siguiente:

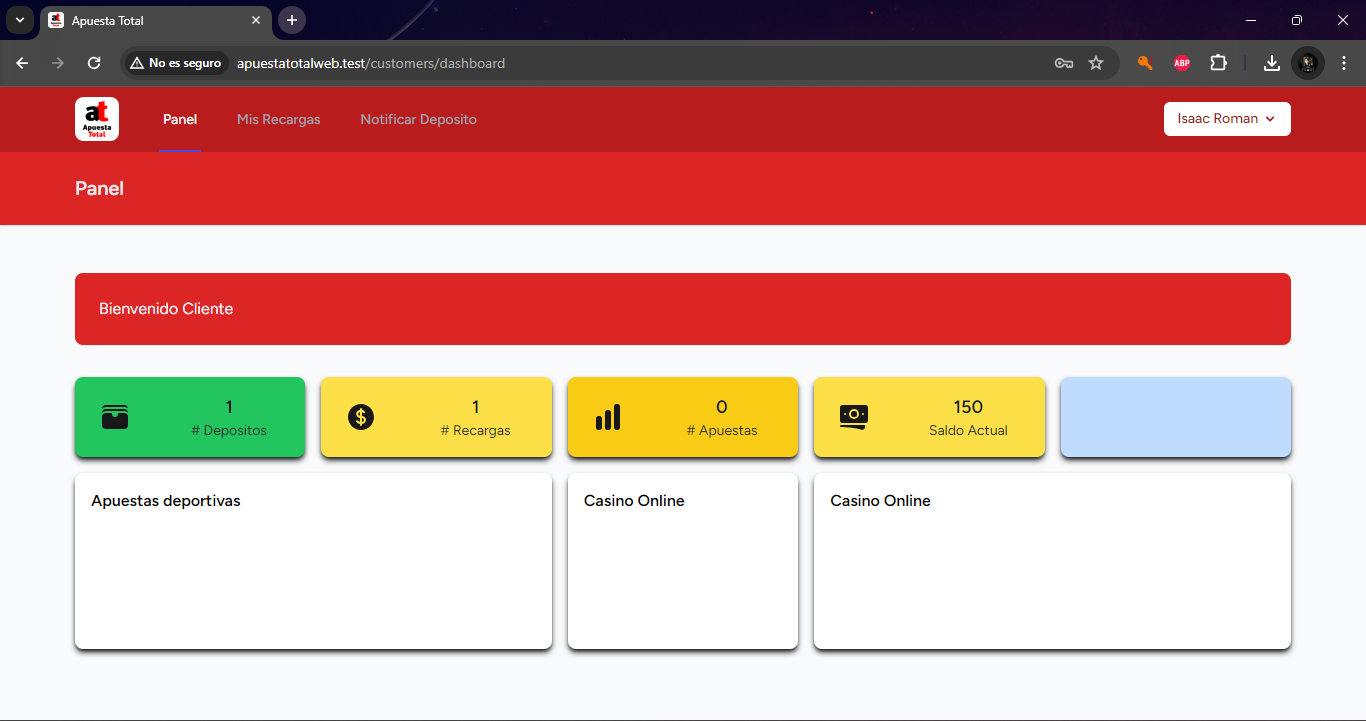
1. **Creación del Proyecto**: Inicialmente, se creó el proyecto base en Laravel y se estableció un repositorio en GitHub.
2. **Gestión de Ramas**: Una vez publicado el repositorio en GitHub, se creó una rama de desarrollo llamada dev-isaac, donde se cargaban los avances conforme se iba programando. Los procesos o módulos completos se cargaban a la rama master, siguiendo la guía de GitFlow.
3. **Configuración de Entorno**: Se configuraron las variables de entorno, especificando el nombre de la aplicación, la URL del hosting local, las credenciales para el envío de correos, entre otros detalles necesarios.
4. **Creación de la Base de Datos**: Se creó la base de datos utilizando el paquete Blueprint, que facilitó la generación de migraciones, modelos y factories.
5. **Pruebas de la Base de Datos**: Una vez creadas las migraciones, se probó el diseño de la base de datos mediante Tinker. Verificado que el diseño funcionaba correctamente con las relaciones y consultas en Eloquent, se instaló Laravel Breeze para gestionar la autenticación de la aplicación.
6. **Generación de Datos de Prueba**: Para la creación de vistas, se necesitaron datos de prueba. Se crearon Seeders para los modelos de la base de datos, lo que facilitó la creación de vistas y rutas sin autenticar para clientes, asesores y administradores.
7. **Implementación de MVC**: Siguiendo el patrón de diseño MVC en Laravel, se aseguraba que cada vez que se declaraba una ruta en el archivo web.php, se tuviera creado el controlador, los modelos de base de datos y las vistas correspondientes.
8. **Seguridad en Formularios**: Al diseñar los formularios, se agregó el token CSRF para incrementar la seguridad en el envío de formularios.
9. **Pruebas y Migraciones**: En cada módulo creado, se generaban pruebas correspondientes. Esto a veces requería añadir atributos extras a las tablas, lo cual se manejaba sin problemas actualizando las migraciones y reflejando los cambios instantáneamente.
10. **Protección de Rutas**: Finalmente, se crearon middleware para proteger las rutas según los roles de cada usuario, asegurando el acceso adecuado a diferentes partes de la aplicación.



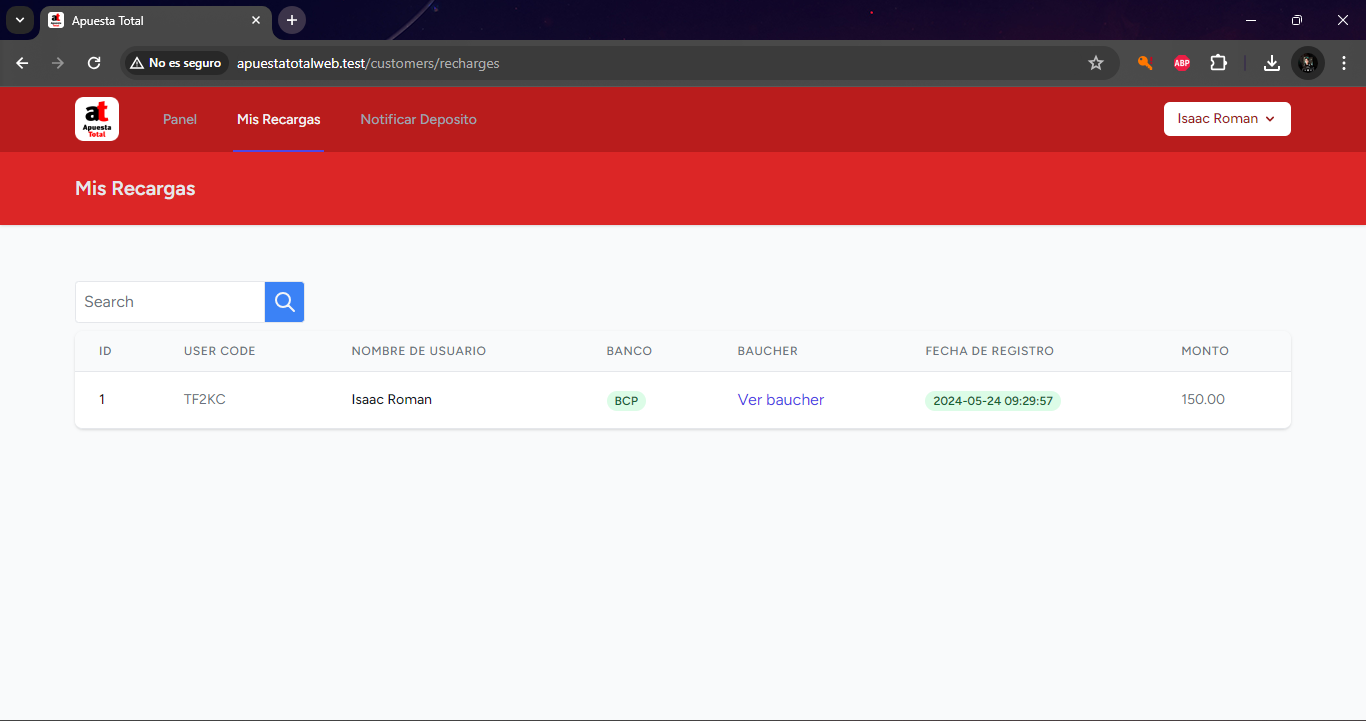


CAPTURA DE PANTALLA DE LOS MODULOS DEL CLIENTE

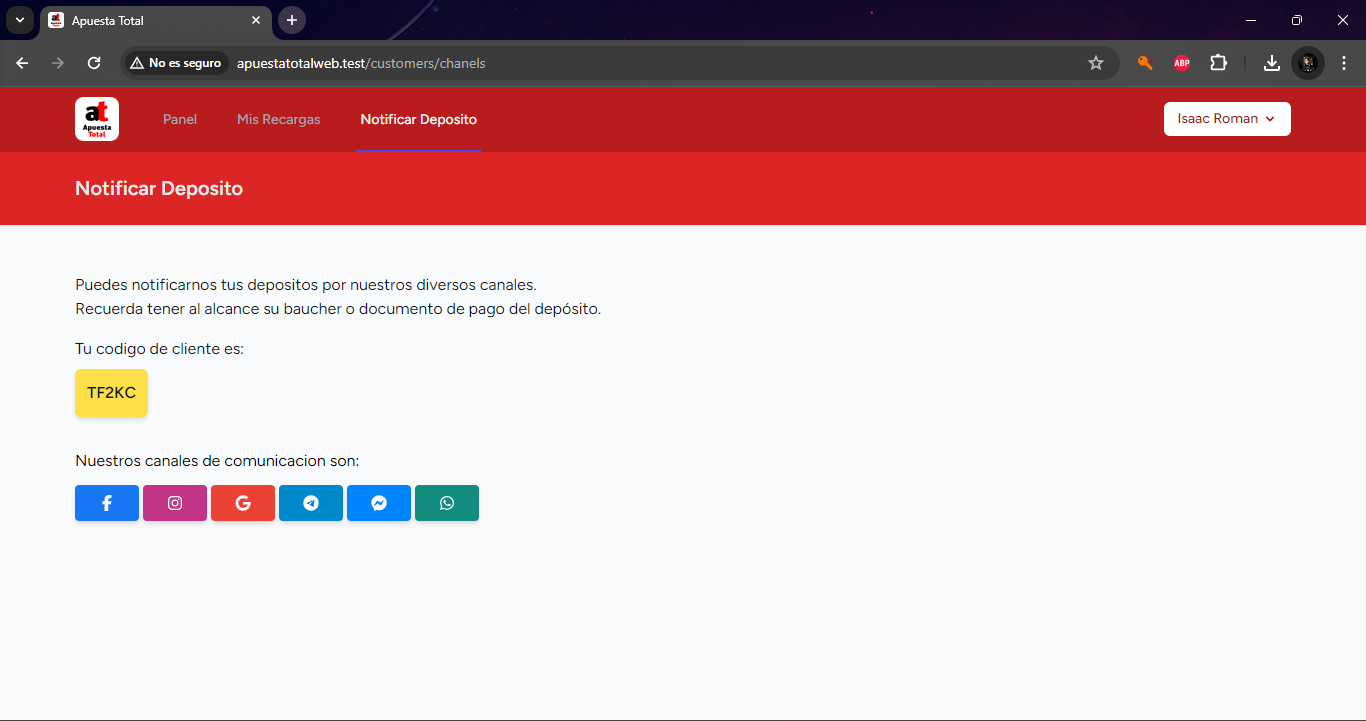
PANEL PRINCIPAL



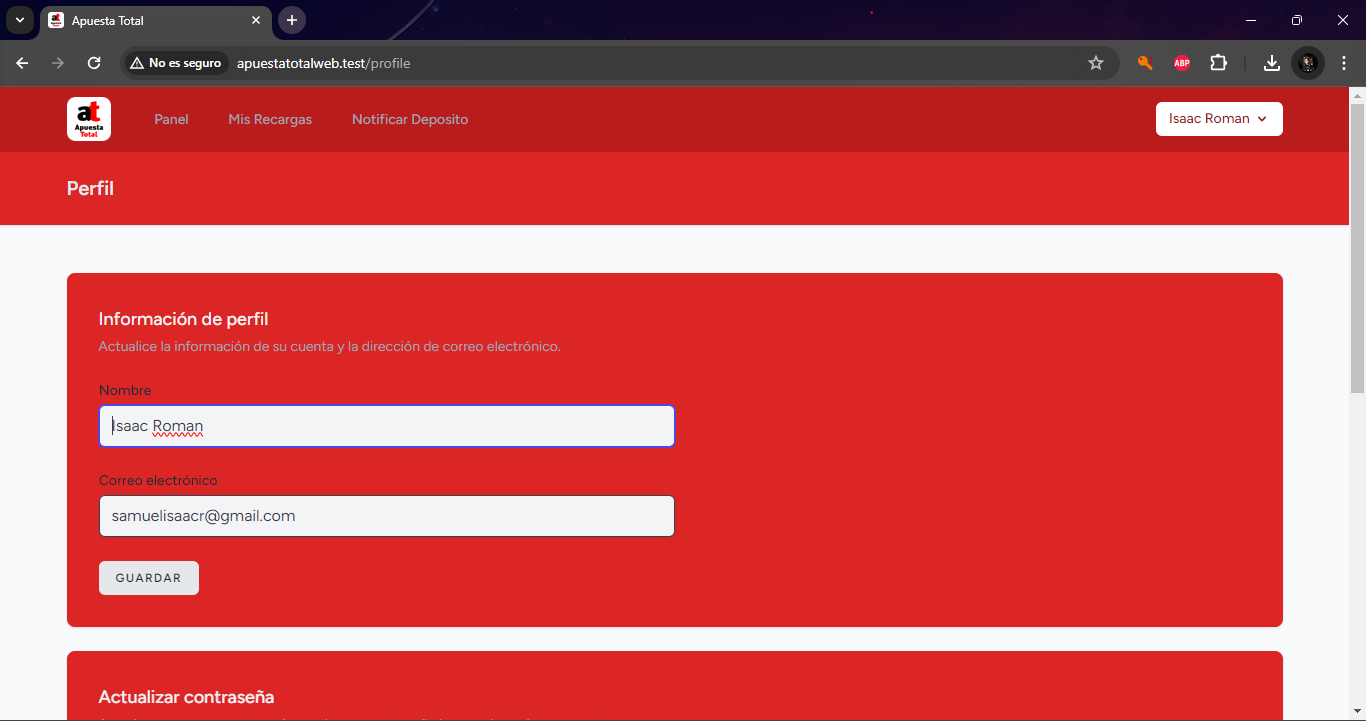
PANEL DE RECARGAS DEL CLIENTE



PANEL SOBRE LOS CANALES DE ATENCIÓN

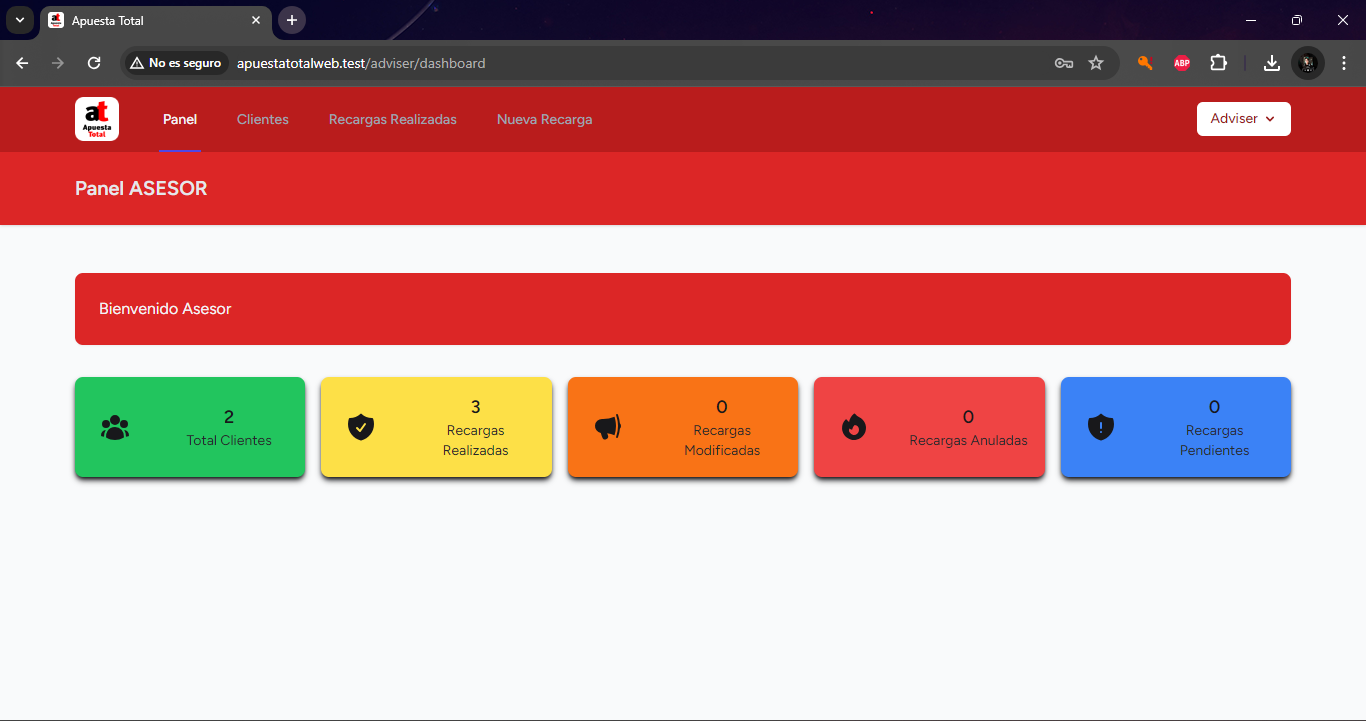


PANEL DE INFORMACIÓN DE LA CUENTA DEL CLIENTE

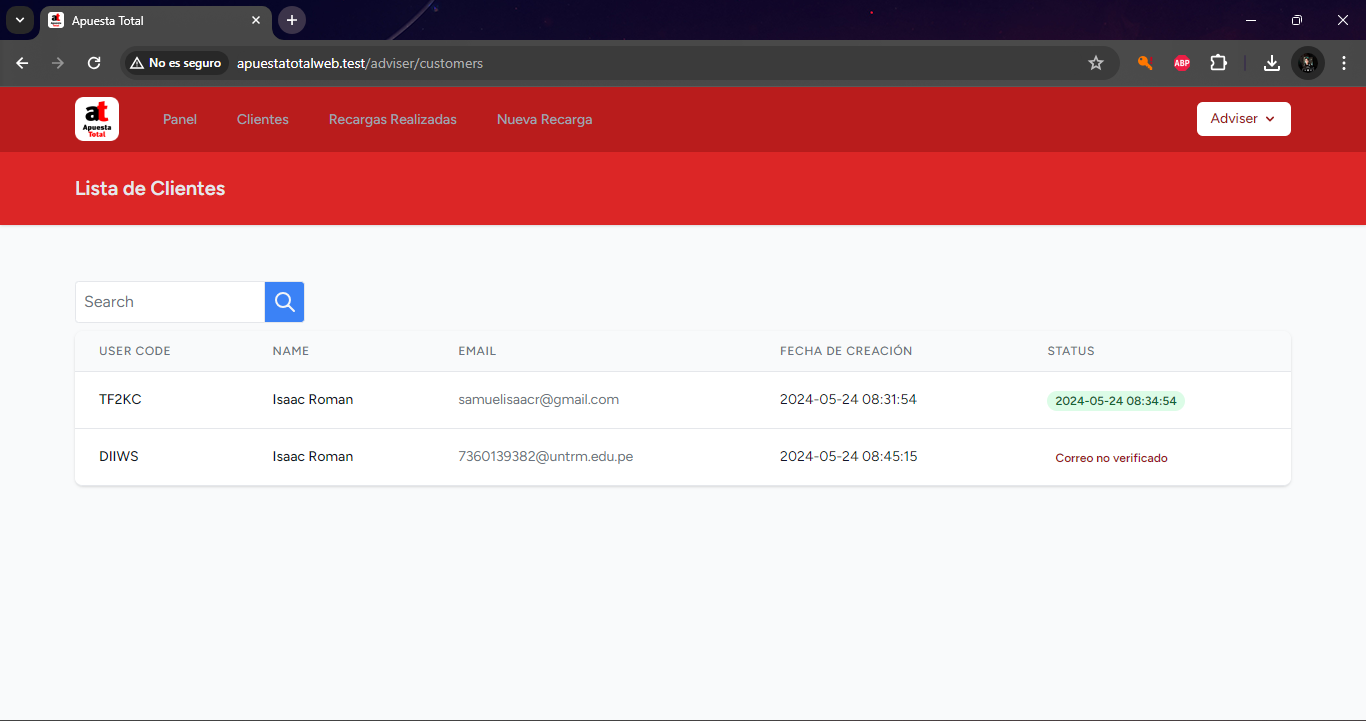


CAPTURAS DE PANTALLA DE LOS MODULOS DEL ASESOR

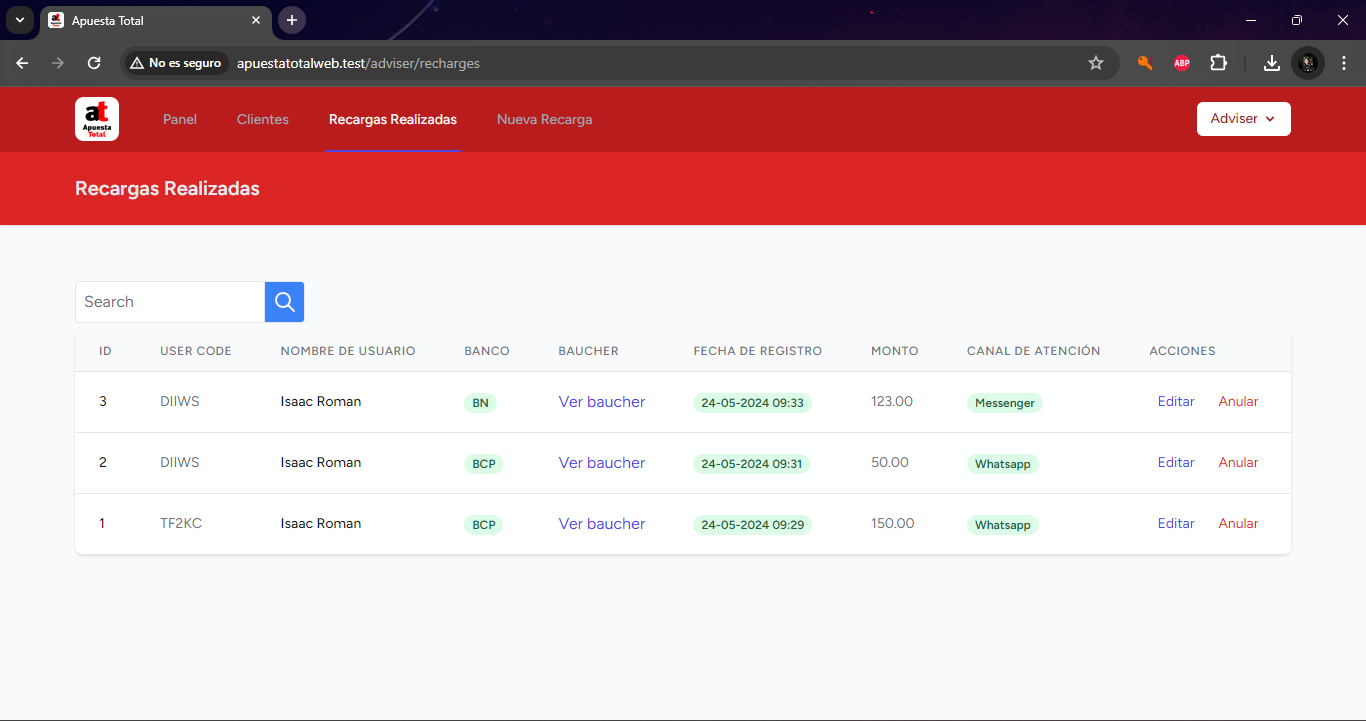
PANEL PRINCIPAL DEL ASESOR



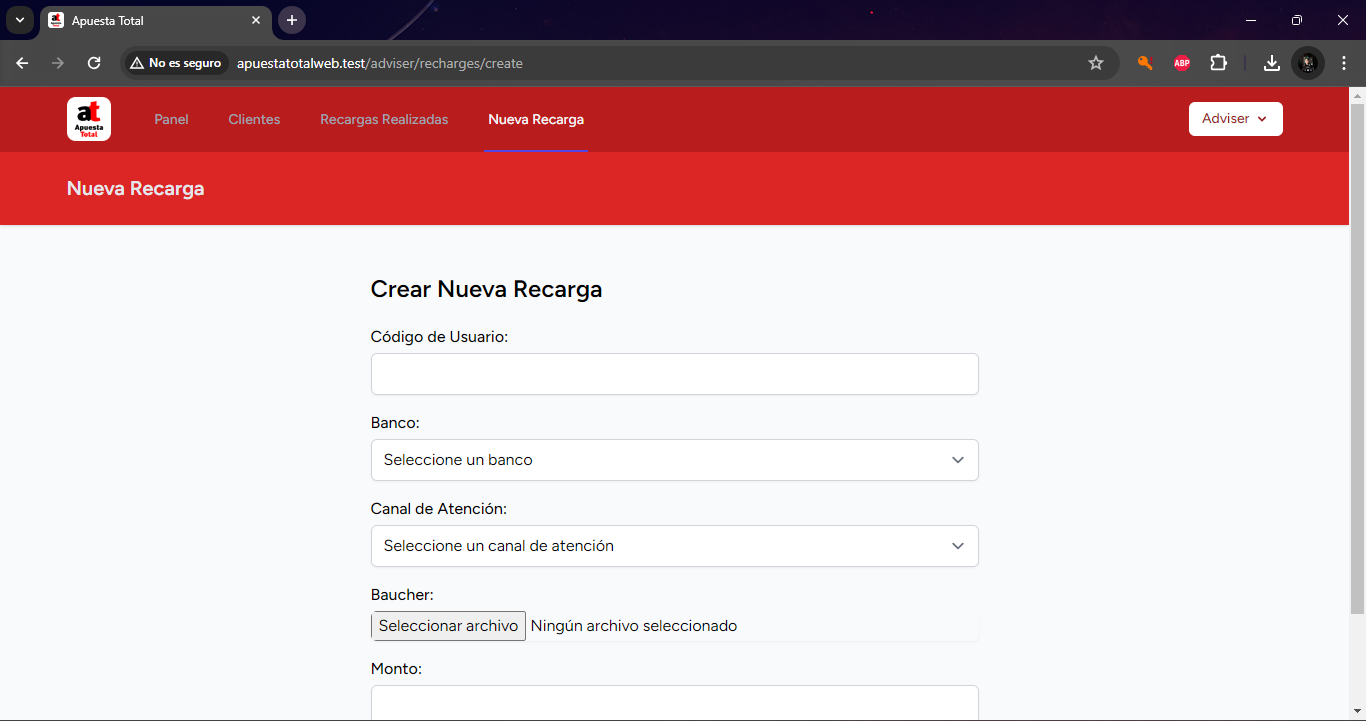
LISTA DE CLIENTES



RECARGAS REALIZADAS

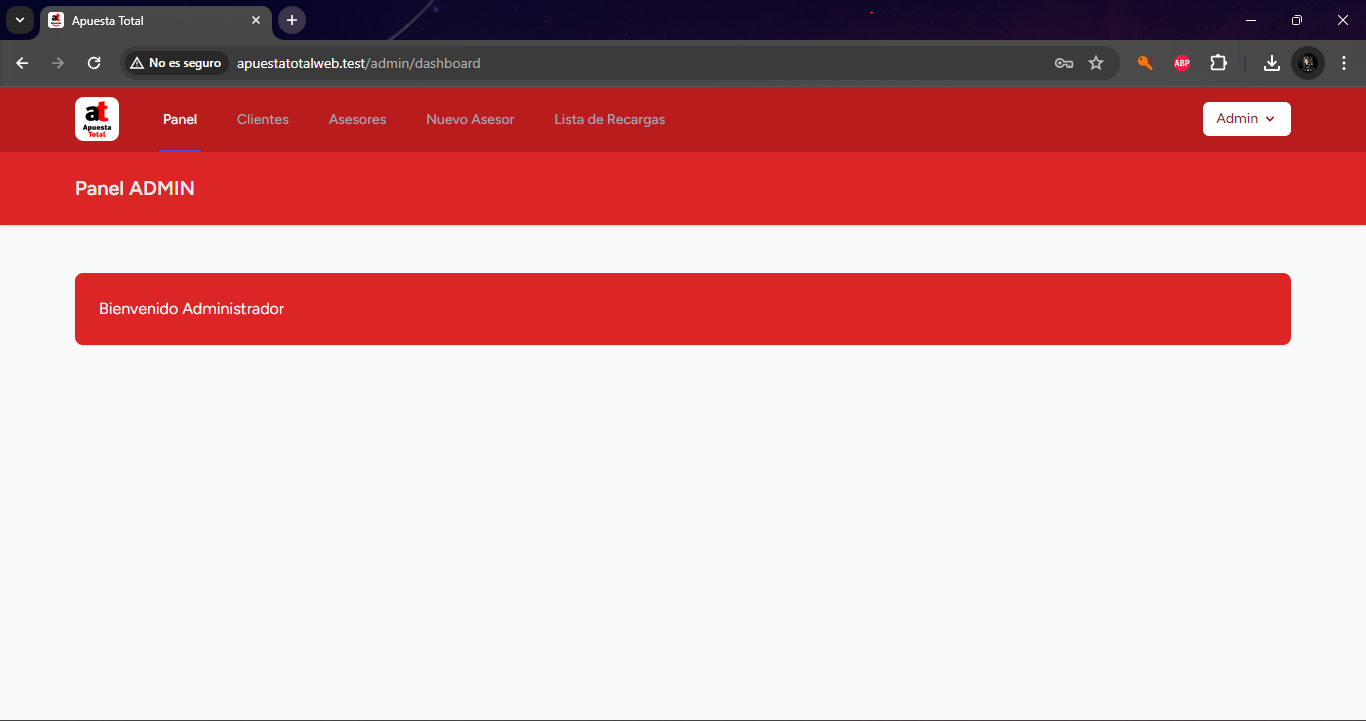


NUEVA RECARGA

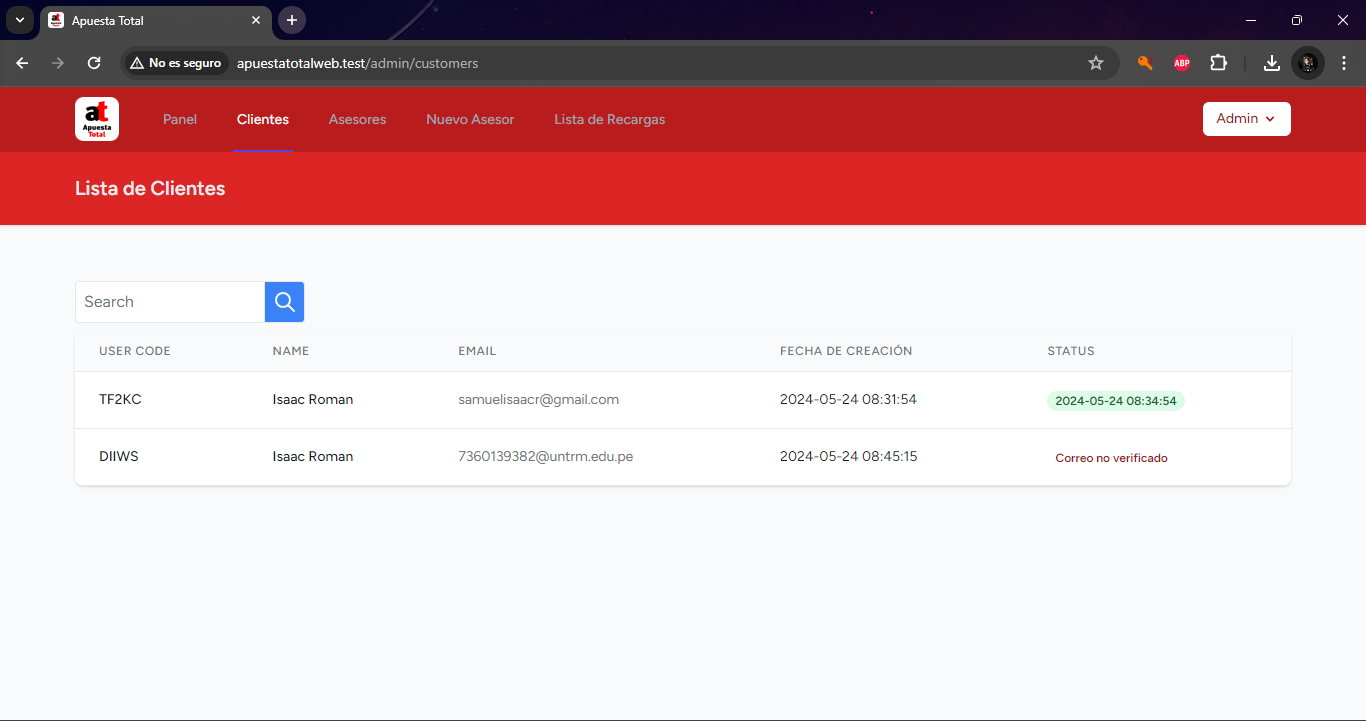


CAPTURA DE PANTALLAS DE LOS MODULOS DEL ADMINISTRADOR

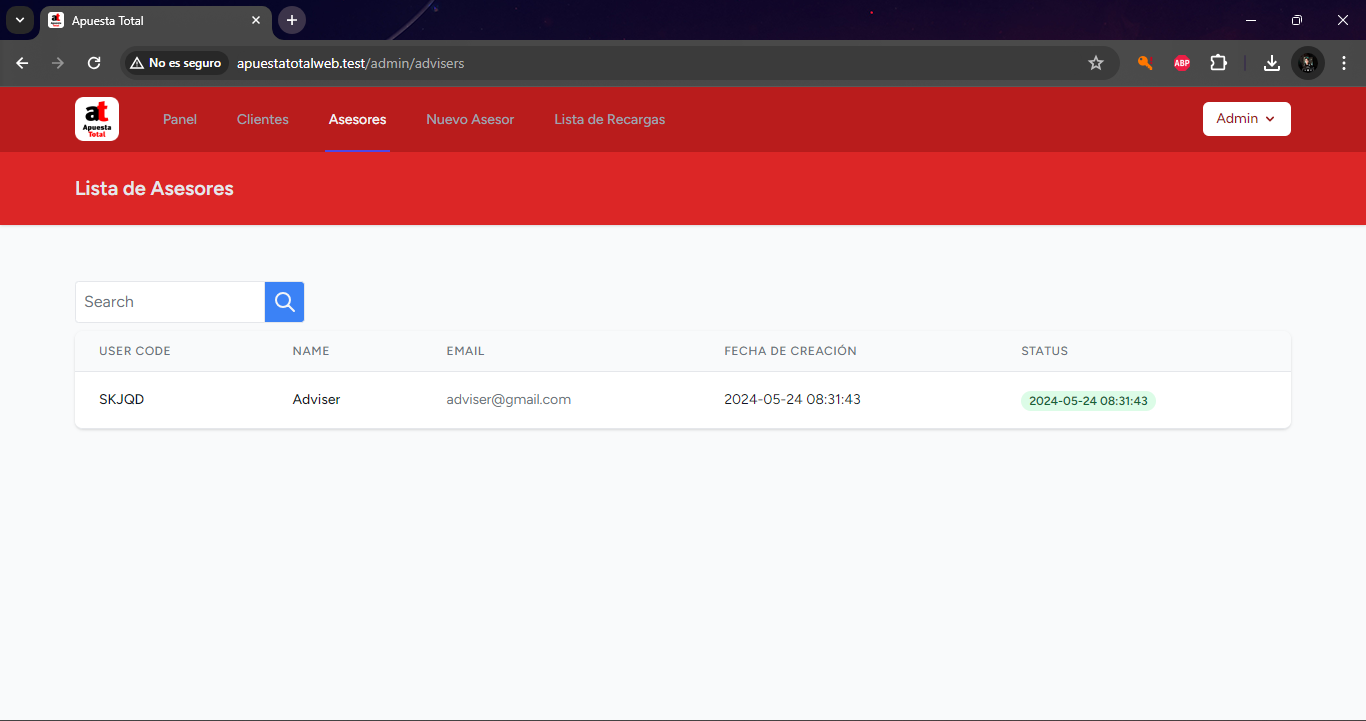
PANEL PRINCIPAL



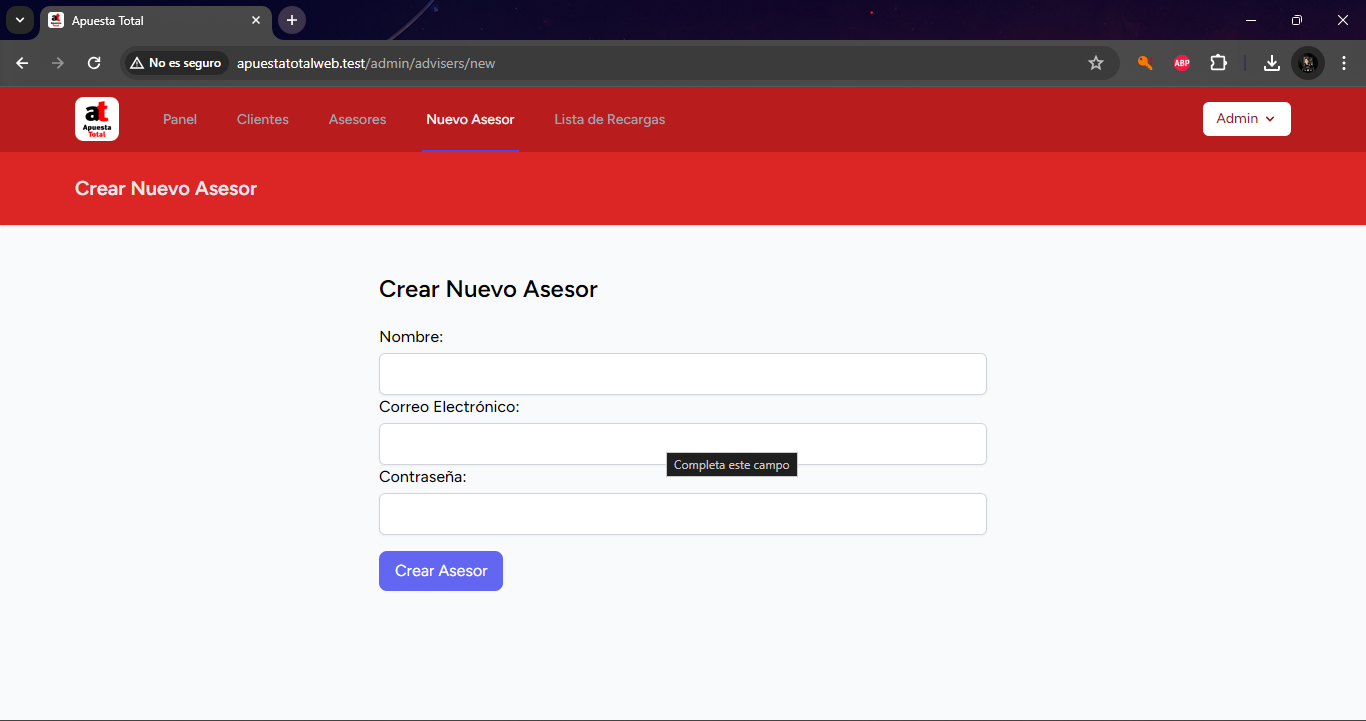
LISTA DE CLIENTES



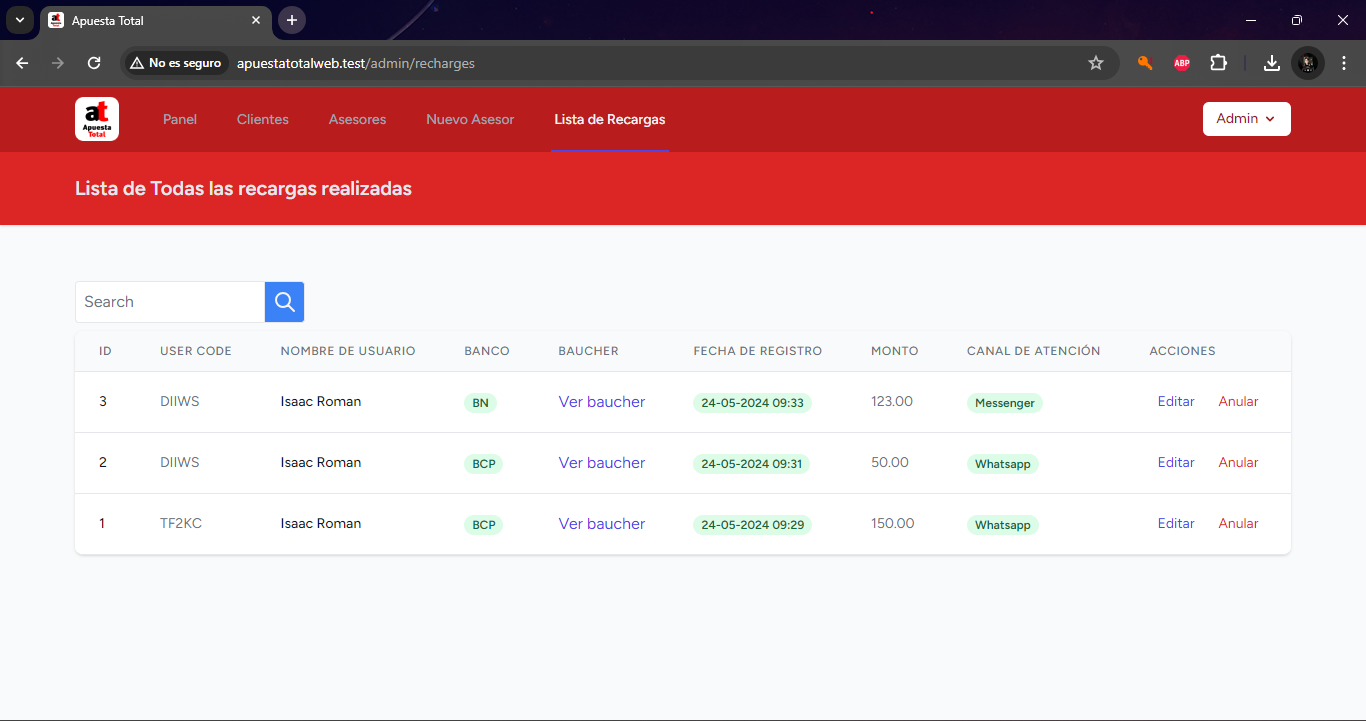
LISTA DE ASESORES



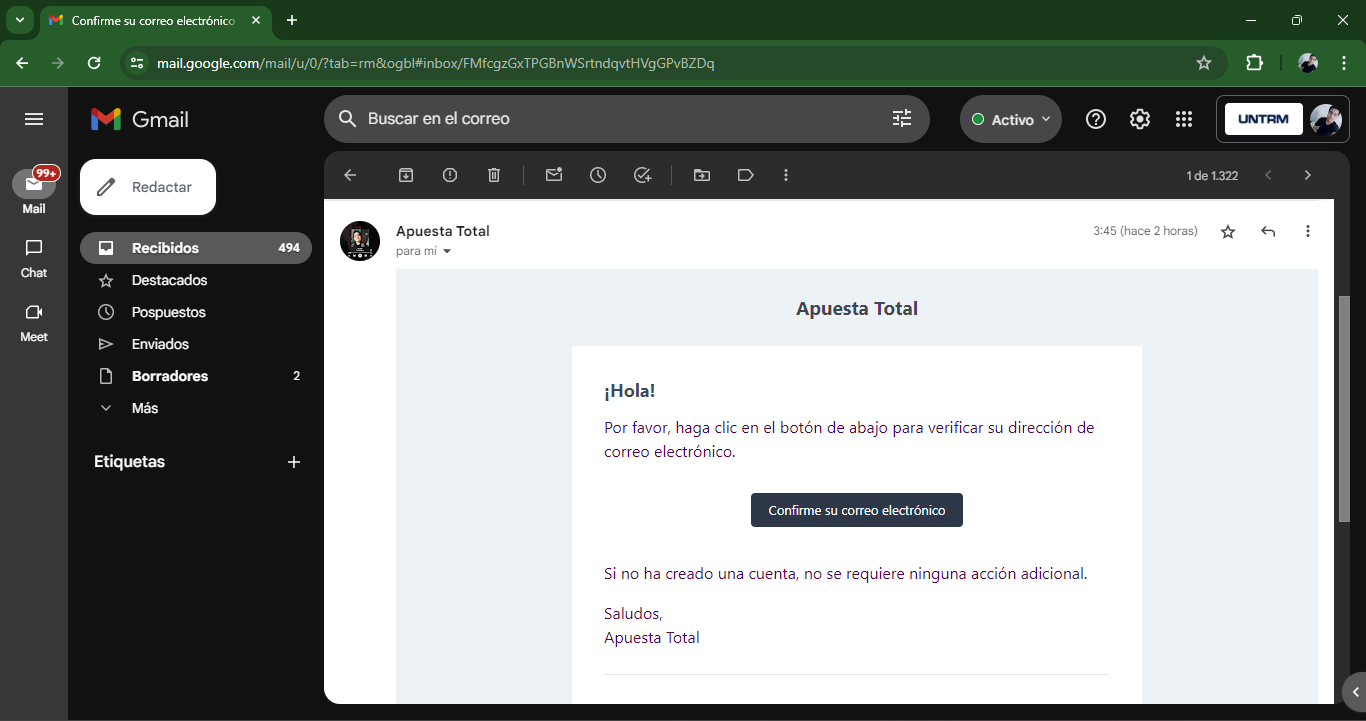
CREAR NUEVO ASESOR



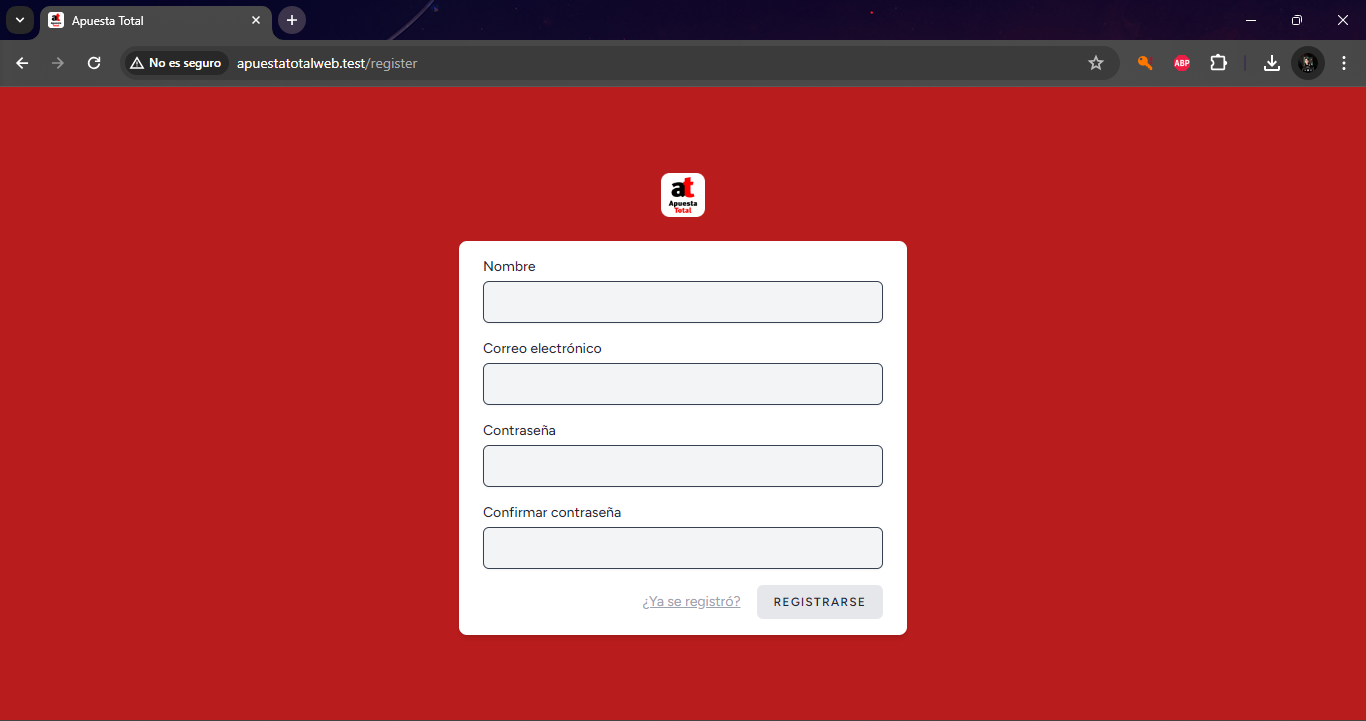
LISTA DE TODAS LAS RECARGAS



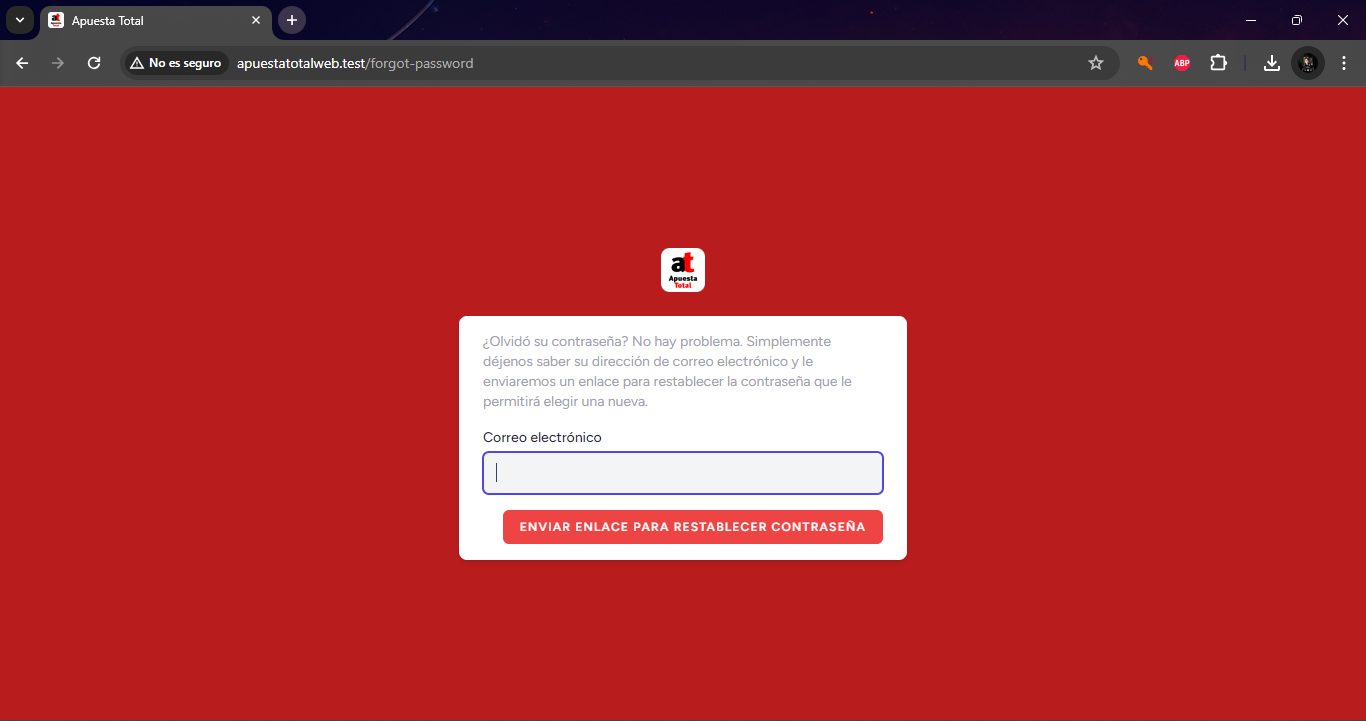
CORREOS DE CONFIRMACIÓN



REGISTRO DE NUEVO CLIENTE



RESTABLECIMIENTO DE CONTRASEÑA



## USUARIOS Y CONTRASEÑA DE PRUEBA

Para ingresar y probar la aplicación puede usted acceder mediante estos usuarios de prueba generados en las migraciones mediante los seeders.

Revise la documentación del README del repositorio, allí se detalla como debería ser el despliegue en un entorno local.

**Usuarios tipo Cliente**

* Usuario: cliente01@gmail.com
* Contraseña: apuestatotal$2024

**Usuarios tipo Asesor**

* Usuario: adviser@gmail.com
* Contraseña: apuestatotal$2024

**Usuario tipo Administrador**

* Usuario: admin@gmail.com
* Contraseña: apuestatotal$2024