

Informe de proyecto [Atención en el instituto de seguro IESS en el Ecuador]

Amanda Rivera Verdezoto^{L00384392} and Nayeli Michelle Tipantiza Cumbal^{L00073321}

Universidad de las Fuerzas Armadas "ESPE"
alrivera4@espe.edu.ec
nmtipantiza@espe.edu.ec

Abstract

El presente informe describe cómo generar una aplicación para dar solución a la atención en el insitituto de seguro IESS en el Ecuador con back-end y el front-end. La propuesta permitiría a los usuarios acceder a los servicios de atención médica y otros servicios ofrecidos por el IESS de manera más eficiente y fácil. Para lograr esto, el desarrollo de la aplicación involucra la creación de una interfaz de usuario atractiva y fácil de usar que permita a los usuarios registrarse y acceder a los servicios del IESS. Además, se debe diseñar y desarrollar un back-end robusto que maneje las solicitudes de los usuarios y proporcione una respuesta rápida y precisa. La aplicación contará con un sistema de gestión de bases de datos para el almacenamiento de información y una capa de seguridad sólida para proteger la información del usuario.

1 Introducción

La atención que brinda la institución de Seguridad Social IESS no es la adecuada. Gran parte de sus casos presenta problemas tediosos, y los usuarios no son atendidos en el momento adecuado. Es por eso que se ha realizado un modelo que logre abarcar la atención que brinda el IESS, para así tener un sistema capaz de organizar toda la información y lo servicios o encargados de brindar los mismos a las personas que cuentan con una afiliación y lo realicen de la manera más adecuada posible. El informe detalla el desarrollo de un proyecto final que consiste en la creación de una aplicación para dar solución al problema que se presenta en el insitituto de seguro IESS en el Ecuador.

Para el back-end de la aplicación se han considerado diversas consideraciones importantes, como la nomenglatura de las estructuras y obejtos establecidos anteriormente, la optimización de consultas de los procesadores almacenados o funciones y la creacióde ínides necesarios para la optimización de procesos. Además, se han implementado disparadores de importancia, incluyendo procesos de auditoría, esto con el fin de garantizar la integridad y seguridad de los datos manejados por la aplicación. También se han creado vista para facilitar el acceso y visualización de información relevante para los usuarios finales. Finalmente se ejecutan comandos de procesos de optimización sobre diferentes consultas que ayudarán al sistema a mejorar el rendimiento y eficiencia del mismo.

El objetivo de este proyecto es demostrar la habilidad para diseñar y desarrollar una aplicación funcional y eficiente, considerando las mejores prácticas y tecnologías disponibles. El primer paso antes de diseñar una BD, es pensar bien en los datos que se quieren almacenar. Lo que se necesita es organizar la información que queremos registrar para nuestro proyecto. Se crearán datos, para poder entender mejor cómo funcionan los atributos y a la vez la función de cada entidad. La solución propuesta ofrece una respuesta práctica y efectiva a un problema real y se espera que tenga impacto positivo en los usuarios finales. La implementación de un back-end bien estructurado y optimizado, junto con un front-end atractivo e intuitivo, garantiza la calidad y usabilidad de la aplicación desarrollada.

Contexto y motivación Se quiere demostrar la capacidad de los estudiantes para aplicar conocimientos y habilidades adquiridas durante el curso, diseñar y desarrollar una aplicación funcional y eficiente, considerando las mejores prácticas y tecnologías disponibles.

La motivación para realizar este proyecto es doble. En primer lugar, permite a los estudiantes aplicar los conocimientos teóricos y prácticos adquiridos durante el curso, a través del diseño y desarrollo de una aplicación real. En segundo lugar, la generación de una aplicación funcional y eficiente puede tener un impacto positivo en los usuarios finales, al ofrecer una solución práctica a un problema concreto.

Para el desarrollo del proyecto se estudió toda la información necesaria acerca de la administración, prestación de bienes y servicios del Instituto IESS, luego de analizar esta información se han definido los requerimientos que este sistema tiene que satisfacer. Se ha realizado una elección de los procedimientos que se consideran necesarios para este proyecto. El sistema será capaz de organizar la información necesaria para poder brindar a la persona afiliada el servicio de manera adecuada y a tiempo.

Problema de estudio El problema radica en la mala atención existente en el instituto y las personas necesitan un mejor esmero por parte del sistema, resulta que la forma de solicitar una cita o adquirir un servicio es muy complicado e ineficiente, por ende muchas veces la atención no es la requerida en ese instante sino luego de mucho tiempo, además de existir más problemas como es la falta de personal médico y medicamentos.

Se trata de una problemática real que afecta a los ciudadanos que dependen de los servicios del IESS, y que puede tener un impacto negativo en su salud, bienestar y calidad de vida. Entre los problemas específicos que pueden afectar la calidad del servicio de atención en el IESS, se pueden mencionar: largas filas de espera, demoras en la atención, falta de personal suficiente y capacitado, problemas de infraestructura y equipamiento, falta de información clara y precisa para los usuarios, entre otros.

Trabajo relacionado La Empresa Despensa Peruana se dedica a la venta de abarrotes, y posee un sistema que registra toda esta información de los procesos de la empresa, debido a las pocas computadoras hace que los vendedores asistan presencialmente solo para su registro y necesitan un sistema que les permita registrar a los pedidos y no dependiendo de un equipo de cómputo para registrar al mismo [1]. Para recibir todos los servicios del IESS, se necesita que el afiliado esté día a día con el pago de sus reportes [2]. Se realizaron investigaciones donde participaban los afiliados, el seguro general y el seguro campesino. El seguro campesino es un régimen del seguro universal obligatorio del IESS que protege a poblaciones de sectores rurales y a las personas que están dedicadas a la pesca artesanal del Ecuador [3]. Se habló sobre la situación hospitalaria del IESS ya que son de gran preocupación, entre ellas está la insatisfacción de los servicios que oferta el hospital, otro de ellos es el rechazo del Call Center, escasez de médicos especialistas, etc. Sin embargo, el 70 por ciento de estas personas respondió que prefieren ser

atendida en el hospital del IESS y el otro 30 por ciento en otros centros de salud. Los afiliados y jubilados informan que las principales dificultades que tienen para ser atendidos es el Call Center. El Hospital IESS tiene requerimientos insatisfechos entre ellos: Consulta externa, que debe ser desmantelada, Emergencia, Cirugías programadas, Farmacia, Laboratorio y Rayos X y necesidad de especialistas adicionales, envío de medicamentos, reparación de equipos de laboratorio y rayos X. La demanda general y urgente es aumentar las unidades de atención primaria y a la vez reemplazar la administración [4].

Resumen de contribución La contribución de este proyecto es proporcionar una solución efectiva para mejorar la calidad del servicio de atención en el Instituto de Seguro IESS en Ecuador, a través de la creación de una aplicación que integra tanto el back-end como el front-end. Se centra en la mejora de la calidad del servicio de atención al cliente en el IESS, lo que puede tener un impacto positivo en la salud, el bienestar y la calidad de vida de los ciudadanos que dependen de los servicios de esta institución.

2 Antecedentes

Algunos antecedentes relevantes que pueden ser de interés para este proyecto son los problemas de atención al cliente en el Instituto de Seguro Social (IESS): En los últimos años, ha habido un aumento en las quejas de los pacientes sobre los tiempos de espera prolongados y la falta de personal en los hospitales y clínicas del IESS en Ecuador. Esto ha llevado a una disminución en la satisfacción del cliente y la percepción negativa del IESS como proveedor de servicios de salud. Otro antecedente es el uso de tecnología en la atención al cliente, con el aumento de la demanda de servicios de atención al cliente, muchas empresas están implementando soluciones tecnológicas para mejorar la eficiencia y la calidad de su atención al cliente. Estas soluciones incluyen aplicaciones móviles, chatbots, sistemas de seguimiento de problemas y analítica de datos para identificar áreas de mejora. Experiencias de otras instituciones en la implementación de soluciones tecnológicas, es importante revisar experiencias de otras instituciones en la implementación de soluciones tecnológicas para mejorar la calidad de la atención al cliente y la gestión de procesos internos. Esto puede ayudar a identificar mejores prácticas y evitar errores comunes en la implementación de soluciones.

3 Método

Antes de desarrollar nuestro producto de software, se ha analizado el problema de este proyecto, el cual consiste en la mala atención por parte del área de administración y se ha realizado las respectivas investigaciones. Se ha separado cada uno de los objetos que pertenecen a la misma, convirtiéndose así en entidades, cada entidad cuenta con sus atributos necesarios para poder desarrollar nuestros modelamientos del problema.

Descripción del diseño de la aplicación: Es importante describir en detalle el diseño de la aplicación que se ha desarrollado para solucionar el problema identificado. Esto incluye los requisitos del sistema, las funcionalidades necesarias y los componentes técnicos. Para el diseño se ha utilizado a Gliffy.

Selección de herramientas y tecnologías: Se debe explicar las herramientas y tecnologías utilizadas para el desarrollo de la aplicación, como el lenguaje de programación, en este caso Python, la base de datos de SQL, el servidor web, entre otros.

Descripción del back-end y front-end: Incluye las funciones de cada componente, la estructura de la base de datos, las consultas y los procedimientos almacenados utilizados, así

como los diseños de la interfaz de usuario.

Descripción de los resultados:: En esta sección se deben presentar los resultados de la aplicación, incluyendo los indicadores de éxito, la eficacia y eficiencia de la aplicación, así como los beneficios para los usuarios y para el IESS.

3.1 Diseño

Diagrama Entidad Relación

En la Figura 1 se puede observar el Diagrama de Entidad Relación de nuestro proyecto creado en Gliffy, todos tienen sus relaciones a otras entidades, las cuales tienen datos que se han considerado importantes para sí mismas. Nos permite representar de manera muy simplificada todos los "componentes", en este caso entidades que van a participar en este proceso que es sobre el IESS, y estarán relacionados entre sí. Este diagrama tiene descripción alto nivel de dominio de datos, organiza los requisitos de datos, y define las entidades y relaciones entre ellas.

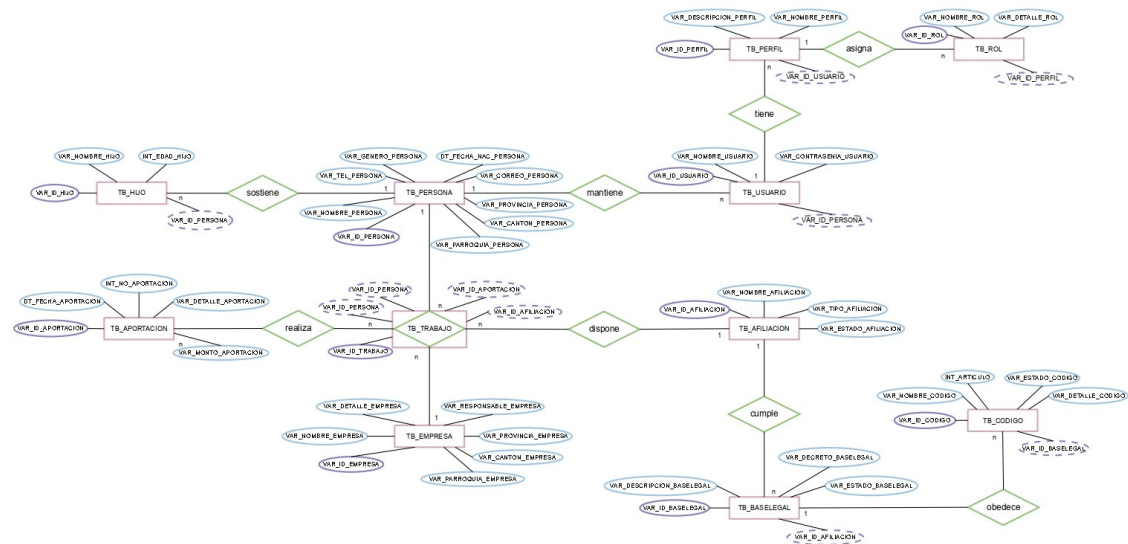


Figure 1: Diagrama Entidad Relación.

Diagrama Lógico

En la Figura 2 se puede observar el Diagrama Lógico de nuestro proyecto, al cual también se le ha creado en Gliffy, este diagrama tiene tecnología independiente y más detalles que el modelo de datos conceptual, define las entidades y sus atributos, las relaciones y su cardinalidad. El modelo describe aspectos relacionados con las necesidades del insituto IEES y recopila los datos y las relaciones entre esos aspectos.

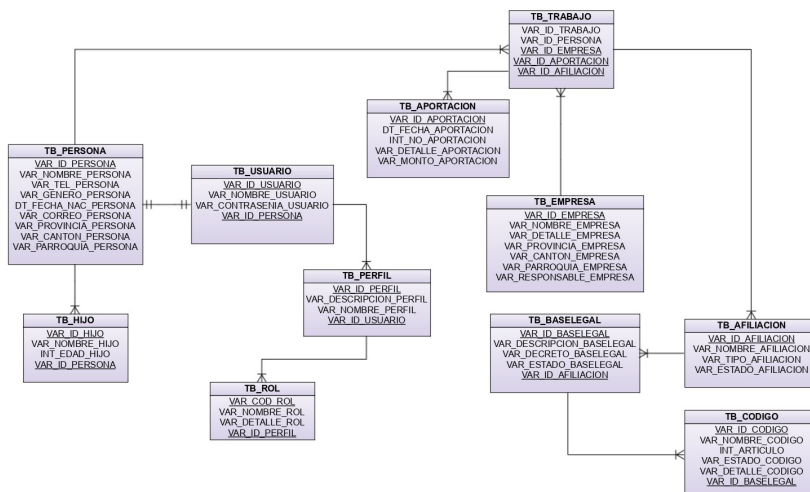


Figure 2: Diagrama Lógico.

Diagrama Físico

En la Figura 3 se puede observar el Diagrama Físico de nuestro proyecto, al mismo también se le ha creado en Gliffy, este diagrama es un modelo específico de la data base, representa objetos de datos relaciones como tablas, columnas, claves primarias, claves foráneas y sus relaciones, contiene más información que los anterires diagramas, y es una base de cómo estará formado la data base de nuestro proyecto, ya que contiene mayor información.

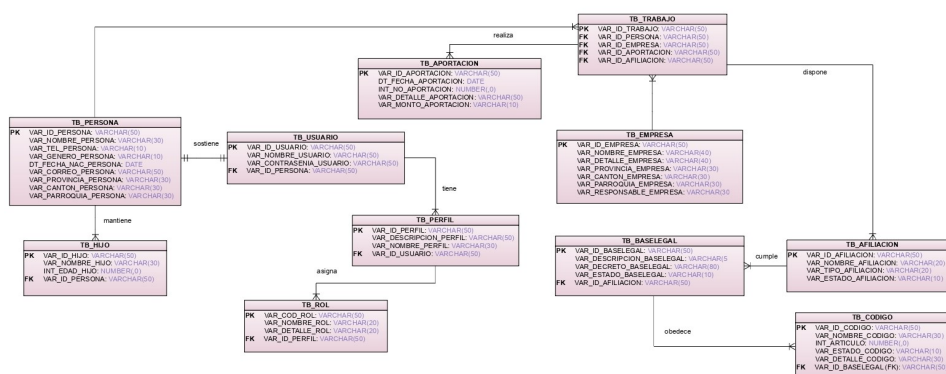


Figure 3: Diagrama Físico.

3.2 Análisis

El primer diseño que se realizó contaba solo con entidades mismas que eran demasiado generalizadas y no cumplían con la primera forma de normalización ya que no tenían la atomicidad necesaria para un diseño, algunas entidades que antes fueron relaciones, fueron eliminadas, ya que las relaciones que se habían hecho, estaban mal especificadas, al tiempo también no abarcaba todos los aspectos importantes necesarios para una institución como es la del IESS. Se considera correcto a este diseño porque se han envuelto todos los aspectos necesarios para que la institución cuente con un sistema capaz de manejar toda la información de manera que las consultas y prestación de servicio se realice de manera eficaz.

El diseño esta enfocado de acuerdo a toda la información teórica obtenida de la página oficial del IESS y fuentes con información de la misma, donde se puede encontrar desde la base legal hasta cuáles son los tipos de servicios, afiliados, etc. Es por esto, que las relaciones establecidas entre las entidades se implementó de acuerdo al conocimiento adquirido, de igual forma la creación de datos sintéticos para probar el funcionamiento del diseño se basó en esta misma información para llenar o intentar llenar los campos.

3.3 Optimización

Para la creación se simuló un diagrama de flujo que nos permita saber como funcionarían distintas situaciones de acuerdo al diseño del diagrama entidad relación para descubrir si las relaciones están bien realizadas, todo esto se realizó de acuerdo a la información obtenida en la página oficial donde se encuentra la información correcta sobre toda esta institución del IESS.

En algunas situaciones se requiere saber qué tipo de afiliación tiene una persona, y es por esto que se relaciona desde la entidad trabajo a afiliacion ya que de acuerdo al tipo de afiliación la persona podrá solicitar un servicio, de misma manera todas las personas afiliadas cuentan con un usuario y este de acuerdo a su tipo, tiene un perfil y por ende cumple con un rol. Y así existen varias consultas que se podrán realizar de manera adecuada de acuerdo a la información que maneja el IESS. Por esto que se establece que el modelo está realizado de manera correcta ya que cumple con los requerimientos establecidos y con las necesidades del usuarios.

3.4 Implementation

ÍNDICES

Análisis del Index

El index que se ha creado en cada tabla fue analizado de manera que no reemplace un campo que sea primary key, pero si un campo que sea único es decir que su información no se repita, en base a esto se han creado índices con el fin de obtener un funcionamiento eficaz en la base de datos.

INDEX - TB_PERSONA

En la entidad de persona se crea en index para el campo de correo de persona ya que este dato no se repite y es único para cada persona, sin embargo es un campo que ocupa espacio y con el índice se busca generar consultas de manera más rápida sin la necesidad de ocupar mucho espacio.

```
1 CREATE UNIQUE INDEX ix_uniq_persona
2 ON TB_PERSONA (VAR_CORREO_PERSONA);
```

Listing 1: Index1.

INDEX - TB_USUARIO

En la entidad de usuario se crea en index para el campo de nombre de usuario, ya que este dato no se repite y es único para cada usuario, sin embargo, es un campo que ocupa espacio y con el índice se busca generar consultas de manera más rápida sin la necesidad de ocupar mucho espacio.

```
1 CREATE UNIQUE INDEX ix_uniq_usuario
2 ON TB_USUARIO (VAR_NOMBRE_USUARIO);
```

Listing 2: Index2.

INDEX - TB_EMPRESA

En la entidad de empresa se crea en index para el campo de nombre empresa ya que este dato no se repite y es único para cada empresa, sin embargo es un campo que ocupa espacio y con el índice se busca generar consultas de manera más rápida sin la necesidad de ocupar mucho espacio.

```
1 CREATE UNIQUE INDEX ix_uniq_empresa
2 ON TB_EMPRESA (VAR_NOMBRE_EMPRESA);
```

Listing 3: Index3.

INDEX - TB_BASELEGAL

En la entidad de base legal se crea en index para el campo de descripción de base legal, ya que este dato no se repite debido a que cada base legal tiene una distinta función por ende la descripción será distinta para cada uno y es único para cada base legal, sin embargo, es un campo que ocupa espacio y con el índice se busca generar consultas de manera más rápida sin la necesidad de ocupar mucho espacio.

```
1 CREATE UNIQUE INDEX ix_uniq_baselegal
2 ON TB_BASELEGAL (VAR_DESCRIPCION_BASELEGAL);
```

Listing 4: Index4.

INDEX - TB_CODIGO

En la entidad de código se crea en index para el campo de detalle código, ya que este dato no se repite debido a que cada código tiene un distinto funcionamiento dentro de la constitución y es único para cada código, sin embargo, es un campo que ocupa espacio y con el índice se busca generar consultas de manera más rápida sin la necesidad de ocupar mucho espacio,

```
1 CREATE UNIQUE INDEX ix_uniq_codigo
2 ON TB_CODIGO (VAR_DETALLE_CODIGO);
```

Listing 5: Index5.

TRIGGERS

Análisis del Trigger

Un trigger es un tipo de procedimiento almacenado que se ejecuta automáticamente en respuesta a ciertas acciones realizadas en la base de datos, como insertar, actualizar o eliminar registros en una tabla. En otras palabras, los triggers son eventos que se activan en la base de datos y desencadenan automáticamente una serie de acciones predefinidas. Se utilizan comúnmente para mantener la integridad de los datos y garantizar la consistencia de los registros en la base de datos. Los triggers se definen en la base de datos y se pueden activar antes o después de la acción

que desencadena el evento. Por lo general, se escriben en lenguajes de programación como SQL, PL/SQL o T-SQL, y se pueden personalizar para realizar acciones específicas en la base de datos.

PRIMER TRIGGER

En el Listing 6 como se comienza a crear una base de datos donde se va a almacenar los cambios que se desean auditar.

```
1 CREATE TABLE TB_AUDITORIA (  
2   id INT,  
3   var_nombre_tabla VARCHAR2(30),  
4   var_accion VARCHAR2(30),  
5   dt_fecha DATE  
6 );
```

Listing 6: Crear TB_AUDITORIA.

La tabla que se va auditar en ese caso es la tabla de Afiliacion.

```
1 CREATE OR REPLACE NONEDITIONABLE TRIGGER TG_AUDITORIA_AFILIADO  
2 AFTER INSERT OR UPDATE OR DELETE ON TB_AFILIACION  
3 FOR EACH ROW  
4 DECLARE  
5   V_ACCION VARCHAR2(30);  
6 BEGIN  
7   IF INSERTING THEN  
8     V_ACCION := 'INSERT';  
9   ELSIF UPDATING THEN  
10    V_ACCION := 'UPDATE';  
11  ELSIF DELETING THEN  
12    V_ACCION := 'DELETE';  
13  END IF;  
14  INSERT INTO TB_AUDITORIA (ID, VAR_NOMBRE_TABLA, VAR_ACCION, DT_FECHA)  
15  VALUES (:NEW.VAR_ID_AFILIACION, 'TB_AFILIACION', V_ACCION, SYSDATE);  
16 END;  
17 /
```

Listing 7: Trigger en TB_AUDITORIA.

Para comprobar el trigger se actualiza una tabla de afiliación, pero antes se mostrará:

```
1 select * from TB_AFILIACION WHERE VAR_ID_AFILIACION = '18  
   cd7c7aa37047e782caa79f1d0d5de9';
```

Listing 8: Comprobar por Select.

Ahora vamos a actualizar la misma fila con el nombre:

```
1 UPDATE TB_AFILIACION  
2 SET VAR_NOMBRE_AFILIACION = 'Asegurado', VAR_ESTADO_AFILIACION = 'Inactivo'  
3 WHERE VAR_ID_AFILIACION = '18cd7c7aa37047e782caa79f1d0d5de9';
```

Listing 9: Update del Trigger.

SEGUNDO TRIGGER

En el Listing 10 se ha creado un trigger referente a si el hijo de alguna persona afiliada es mayor a 18 años, este tendrá el mensaje de que ya no cuenta con el seguro por parte del padre o madre.

```
1 CREATE OR REPLACE TRIGGER Tg_mensaje_edad_18  
2 AFTER UPDATE ON TB_HIJO  
3 FOR EACH ROW
```



```

4 BEGIN
5   IF :NEW.INT_EDAD_HIJO >= 18 THEN
6     DBMS_OUTPUT.PUT_LINE('Ya no cuenta con seguro de salud por parte del padre
7     ');
8   END IF;
9 END;
/

```

Listing 10: Trigger en TB_HIJO.

Ahora vamos a observar la fila que queremos actualizar.

```

1 select * from TB_HIJO where VAR_ID_HIJO = 'b7f626e2685b46c6aabbcd26ce072188';
2 UPDATE TB_HIJO
3 SET INT_EDAD_HIJO = 18 WHERE VAR_ID_HIJO = 'b7f626e2685b46c6aabbcd26ce072188';

```

Listing 11: Update EDAD_HIJO.

TERCER TRIGGER

En el Listing 12 se ha creado un trigger en caso de que una persona actualice el número de aportaciones al número 24 o mayor, este se observará a manera de mensaje que ya puede reclamar sus fondos de cesantía.

```

1 CREATE OR REPLACE TRIGGER Tg_mensaje_fondos
2 AFTER UPDATE ON TB_APORTACION
3 FOR EACH ROW
4 BEGIN
5   IF :NEW.INT_NO_APORTACION >= 24 THEN
6     DBMS_OUTPUT.PUT_LINE('Ya puede reclamar los fondos de cesant a');
7   END IF;
8 END;
9 /

```

Listing 12: Trigger en TB_APORTACION.

Ahora vamos a observar la fila que queremos actualizar.

```

1 select * from TB_APORTACION where VAR_ID_APORTACION = '82
2   a9098ce7694d5382c777bfe7195eb6';
3 UPDATE TB_APORTACION
4 SET INT_NO_APORTACION = 24 WHERE VAR_ID_APORTACION = '82
5   a9098ce7694d5382c777bfe7195eb6';

```

Listing 13: Update INT_NO_APORTACION.

CUARTO TRIGGER

En el Listing 14 se ha creado un trigger para verificar si una persona es pensionista (es decir, si su edad es mayor o igual a 65 años) y muestre un mensaje indicando que es acreedora a la pensión otorgada por la institución.

```

1 CREATE OR REPLACE TRIGGER TRG_VERIFICAR_PENSIONISTA
2 BEFORE INSERT OR UPDATE ON TB_PERSONA
3 FOR EACH ROW
4 DECLARE
5   edad NUMBER;
6 BEGIN
7   -- Calculamos la edad de la persona a partir de su fecha de nacimiento
8   edad := TRUNC(MONTHS_BETWEEN(SYSDATE, :NEW.DT_FECHA_NAC_PERSONA) / 12);
9   -- Si la persona tiene 65 años o más, es pensionista
10  IF edad >= 65 THEN

```

```

11 DBMS_OUTPUT.PUT_LINE('La persona es acreedora a la pensi n otorgada
    por la instituci n');
12 END IF;
13 END;
14 /

```

Listing 14: Trigger en TB_PERSONA.

Y para comprobar ingresaremos el dato de una persona.

```

1 INSERT INTO TB_PERSONA
2 (VAR_ID_PERSONA, VAR_NOMBRE_PERSONA, VAR_TEL_PERSONA, VAR_GENERO_PERSONA,
    DT_FECHA_NAC_PERSONA, VAR_CORREO_PERSONA, VAR_PROVINCIA_PERSONA,
    VAR_CANTON_PERSONA, VAR_PARROQUIA_PERSONA)
3 VALUES ('1845612547', 'Victor Borquez', '0961245754', 'Masculino', TO_DATE('
    1950-01-01', 'YYYY-MM-DD'), 'victorcz@gmail.com', 'Pichincha', 'Quito', '
    La Floresta');

```

Listing 15: Insert TB_PERSONA.

QUINTO TRIGGER

En el Listing 16 se ha creado un trigger para verificar si el nombre de usuario ya existe en la tabla y evitar su inserción si ya existe.

```

1 CREATE OR REPLACE TRIGGER TRG_TB_USUARIO_BEFORE_INSERT
2 BEFORE INSERT ON TB_USUARIO
3 FOR EACH ROW
4 DECLARE
5     V_COUNT INTEGER;
6 BEGIN
7     SELECT COUNT(*) INTO V_COUNT FROM TB_USUARIO WHERE VAR_NOMBRE_USUARIO = :
    NEW.VAR_NOMBRE_USUARIO;
8     IF V_COUNT > 0 THEN
9         RAISE_APPLICATION_ERROR(-20001, 'El nombre de usuario ya existe en la
    tabla. ');
10    END IF;
11 END;
12 /

```

Listing 16: Trigger en TB_USUARIO.

Y para comprobar ingresaremos el dato de una persona.

```

1 INSERT INTO TB_USUARIO (VAR_ID_USUARIO, VAR_NOMBRE_USUARIO,
    VAR_CONTRASENIA_USUARIO, VAR_ID_PERSONA)
2 VALUES ('965', 'nayelmtc', 'ponymalta123', '1752373142');

```

Listing 17: Insert TB_USUARIO.

PROCEDIMIENTOS ALMACENADOS

Análisis del Procedimiento

Un procedimiento almacenado es un conjunto de instrucciones SQL que se guardan en la base de datos como un objeto. Estas instrucciones pueden incluir consultas SQL, declaraciones condicionales, bucles y otros tipos de lógica empresarial. Se utilizan comúnmente para automatizar tareas complejas en la base de datos y mejorar la eficiencia del procesamiento de datos. Además, los procedimientos almacenados se pueden reutilizar en múltiples aplicaciones y consultas, lo que ahorra tiempo y reduce el riesgo de errores en la programación. También pueden proporcionar

una capa adicional de seguridad al limitar el acceso directo a la base de datos y controlar quién puede ejecutar ciertas tareas.

PRIMER PROCEDIMIENTO

En el Listing 18 se ha creado un procedimiento que indique si las persona tiene mas de 24 aportaciones para poder ser acreedora de el préstamo quirografario. El préstamo quirografario es destinado para que los afiliados, jubilados y pensionistas, sujetos de crédito, puedan obtener recursos para solventar sus necesidades de consumo.

```
1 CREATE OR REPLACE PROCEDURE sp_consulta_aportaciones AS
2   c1 SYS_REFCURSOR;
3 BEGIN
4   OPEN c1 FOR SELECT
5     tb_persona.var_nombre_persona,
6     tb_aportacion.int_no_aportacion,
7     tb_aportacion.var_monto_aportacion
8   FROM tb_trabajo
9        INNER JOIN tb_persona ON tb_trabajo.var_id_persona = tb_persona.
10       var_id_persona
11       INNER JOIN tb_aportacion ON tb_trabajo.var_id_aportacion =
12       tb_aportacion.var_id_aportacion
13   WHERE tb_aportacion.int_no_aportacion >= 24;
14   dbms_sql.return_result(c1);
15 END;
16 /
```

Listing 18: Procedimiento From TB_TRABAJO.

Comprobar si el procedimiento ha sido creado.

```
1 EXEC sp_consulta_aportaciones;
```

Listing 19: Exec en primer procedimiento.

SEGUNDO PROCEDIMIENTO

En el Listing 20 se crea un procedimiento donde se requiere a las personas que tengan mas de 3 aportaciones para poder tener acceso al servicio de salud.

```
1 CREATE OR REPLACE PROCEDURE sp_aportacion_salud AS
2   c1 SYS_REFCURSOR;
3 BEGIN
4   OPEN c1 FOR SELECT
5     tb_persona.var_nombre_persona,
6     tb_aportacion.int_no_aportacion,
7     tb_aportacion.var_monto_aportacion
8   FROM tb_trabajo
9        INNER JOIN tb_persona ON tb_trabajo.var_id_persona = tb_persona.
10       var_id_persona
11       INNER JOIN tb_aportacion ON tb_trabajo.var_id_aportacion =
12       tb_aportacion.var_id_aportacion
13   WHERE tb_aportacion.int_no_aportacion >= 3;
14   dbms_sql.return_result(c1);
15 END;
16 /
```

Listing 20: Procedimiento From TB_TRABAJO.

Comprobar si el procedimiento ha sido creado.

```
1 EXEC sp_aportacion_salud;
```

Listing 21: Exec en segundo procedimiento.

TERCER PROCEDIMIENTO

En el Listing 22 se ha creado un procedimiento en donde se requiere mostrar los artículos con el código al que pertenezcan indicando que han sido cambiados o suspendidos con el paso del tiempo y han afectado a las personas aseguradas.

```
1  Crear CREATE OR REPLACE PROCEDURE sp_codigo_afiliacion AS
2      c1 SYS_REFCURSOR;
3  BEGIN
4      OPEN c1 FOR SELECT TB_CODIGO.INT_ARTICULO, TB_CODIGO.VAR_NOMBRE_CODIGO,
5      TB_CODIGO.VAR_ESTADO_CODIGO, TB_AFILIACION.VAR_NOMBRE_AFILIACION
6  FROM TB_CODIGO
7      INNER JOIN TB_BASELEGAL ON TB_CODIGO.VAR_ID_BASELEGAL = TB_BASELEGAL.
8      VAR_ID_BASELEGAL
9      INNER JOIN TB_AFILIACION ON TB_BASELEGAL.VAR_ID_AFILIACION = TB_AFILIACION
10     .VAR_ID_AFILIACION
11     WHERE VAR_ESTADO_CODIGO = 'Inactivo' AND VAR_NOMBRE_AFILIACION = '
    Asegurado';
    dbms_sql.return_result(c1);
END;
/
```

Listing 22: Procedimiento From TB_CODIGO.

Comprobar si el procedimiento ha sido creado.

```
1 EXEC sp_codigo_afiliacion;
```

Listing 23: Exec en tercer procedimiento.

CUARTO PROCEDIMIENTO

En el Listing 24 se crea un procedimiento en donde requiere mostrar las personas de Pichincha que han realizado aportaciones menores a 55 dólares y la fecha en que realizaron dicha aportación, esto para realizar cálculos de lo que al afiliado le corresponde entregar un aporte al IESS del 9,45 por ciento de su sueldo o salario; mientras que al empleador, el 11,15 por ciento del salario del trabajador.

```
1 CREATE OR REPLACE PROCEDURE sp_consulta_aportacion AS
2     c1 SYS_REFCURSOR;
3 BEGIN
4     OPEN c1 FOR SELECT TB_PERSONA.VAR_NOMBRE_PERSONA, TB_PERSONA.
5     VAR_PROVINCIA_PERSONA,
6     TB_APORTACION.VAR_MONTO_APORTACION AS "MONTO APORTACION",
7     TB_APORTACION.DT_FECHA_APORTACION AS "FECHA APORTACION"
8 FROM TB_PERSONA
9     INNER JOIN TB_TRABAJO ON TB_TRABAJO.VAR_ID_PERSONA=TB_PERSONA.
10    VAR_ID_PERSONA
11    INNER JOIN TB_APORTACION ON TB_TRABAJO.VAR_ID_APORTACION=TB_APORTACION
12    .VAR_ID_APORTACION
13    WHERE TB_APORTACION.VAR_MONTO_APORTACION < 900 AND TB_PERSONA.
14    VAR_PROVINCIA_PERSONA = 'Pichincha';
15    dbms_sql.return_result(c1);
END;
/
```

Listing 24: Crear TB_AUDITORIA.

Comprobar si el procedimiento ha sido creado.

```
1 EXEC sp_consulta_aportacion;
```

Listing 25: Exec en cuarto procedimiento.

QUINTO PROCEDIMIENTO

En el Listing 26 se ha creado un procedimiento en donde se requiere mostrar las personas que sean pensionistas y consten en el sistema del Iess para contabilizar cuantas personas son acreedoras de la pensión otorgada por la institución.

```
1 CREATE OR REPLACE PROCEDURE sp_consulta_usuario
2 AS
3     c1 SYS_REFCURSOR;
4 BEGIN
5     OPEN c1 FOR
6         SELECT TB_USUARIO.VAR_NOMBRE_USUARIO, TB_ROL.VAR_NOMBRE_ROL, TB_PERFIL.
7             VAR_NOMBRE_PERFIL
8         FROM TB_ROL
9         INNER JOIN TB_PERFIL ON TB_ROL.VAR_ID_PERFIL = TB_PERFIL.VAR_ID_PERFIL
10        INNER JOIN TB_USUARIO ON TB_PERFIL.VAR_ID_USUARIO= TB_USUARIO.
11        WHERE VAR_NOMBRE_ROL = 'Pensionista' AND VAR_NOMBRE_PERFIL = 'Usuario';
12     dbms_sql.return_result(c1);
13 END;
```

Listing 26: Procedimiento From TB_ROL.

Comprobar si el procedimiento ha sido creado.

```
1 EXEC sp_consulta_usuario;
```

Listing 27: Exec en quinto procedimiento.

VISTAS

Análisis de Vistas

Una vista es una representación virtual de una tabla o combinación de tablas. Se puede definir como una consulta predefinida que devuelve un conjunto de filas y columnas seleccionadas de una o varias tablas en la base de datos. Las vistas proporcionan una forma conveniente de acceder y manipular datos complejos sin tener que escribir consultas SQL complejas. En lugar de eso, se pueden crear vistas personalizadas que resumen y presentan los datos de la manera deseada. También pueden utilizarse para ocultar información confidencial o restringir el acceso a ciertas partes de la base de datos. Por ejemplo, se puede crear una vista que muestre sólo los datos relevantes para un usuario determinado, mientras que oculta los datos sensibles.

PRIMERA VISTA

Se ha creado una vista que liste a todos los hijos que sean mayores de edad de dicha persona. Se comienza con la creación de la vista como se observa en el Listing 28.

```
1 CREATE VIEW vista_hijo_mayores AS
2     SELECT
3         var_nombre_hijo,
4         int_edad_hijo
5     FROM
6         tb_hijo
7     WHERE
```

```
8      int_edad_hijo >= 18;
```

Listing 28: Vista From TB_HIJO.

Ejecución de la Vista con el comando mostrado en el Listing 29.

```
1  Select * from vista_hijo_mayores;
```

Listing 29: Select en primera vista.

SEGUNDA VISTA

Se ha creado una vista que indique todos los códigos inactivos y la base legal a la que pertenezcan.

```
1  CREATE VIEW vista_codigo_inactivo AS
2  SELECT
3      tb_codigo.var_nombre_codigo as "Codigo",
4      tb_codigo.var_estado_codigo as "Estado delCodigo",
5      tb_baselegal.var_decreto_baselegal as "Base Legal"
6  FROM
7      tb_codigo
8      INNER JOIN tb_baselegal ON tb_codigo.var_id_baselegal = tb_baselegal.
9      var_id_baselegal
10 WHERE
11     tb_codigo.var_estado_codigo = 'Inactivo';
```

Listing 30: Vista Codigo_Inactivo

Ejecución de la Vista con el comando cmostrado en el Listing 31.

```
1  Select * from vista_codigo_inactivo;
```

Listing 31: Select en segunda vista.

TERCERA VISTA

Se ha creado una vista que muestre todas las persona afiliadas que aporten de manera voluntaria al instituto.

```
1  CREATE VIEW vista_afiliado_voluntario AS
2  SELECT
3      tb_persona.var_nombre_persona as "Persona",
4      tb_afiliacion.var_nombre_afiliacion as "Afiliaci n",
5      tb_afiliacion.var_tipo_afiliacion as "Tipo Afiliacion"
6  FROM
7      tb_trabajo
8      INNER JOIN tb_persona ON tb_trabajo.var_id_persona = tb_persona.
9      var_id_persona
10     INNER JOIN tb_afiliacion ON tb_trabajo.var_id_afiliacion =
11     tb_afiliacion.var_id_afiliacion
12 WHERE
13     tb_afiliacion.var_nombre_afiliacion = 'Voluntario'
14     AND tb_afiliacion.var_tipo_afiliacion = 'Afiliado';
```

Listing 32: Vista From TB_TRABAJO.

Ejecución de la Vista con el comando mostrado en el Listing 33.

```
1  SELECT * FROM vista_afiliado_voluntario;
```

Listing 33: Select en tercera vista.

CUARTA VISTA

Se ha creado una vista que muestre los usuarios que tengan un perfil creado hace dos años.

```
1 CREATE VIEW vista_usuario_perfil AS
2 SELECT
3     tb_usuario.var_nombre_usuario,
4     tb_perfil.var_descripcion_perfil
5 FROM
6     tb_perfil
7 INNER JOIN tb_usuario ON tb_perfil.var_id_usuario = tb_usuario.
8 var_id_usuario
9 WHERE
    tb_perfil.var_descripcion_perfil = 'Creado hace 2 años';
```

Listing 34: Vista From TB_PERFIL.

Ejecución de la Vista con el comando mostrado en el Listing 35.

```
1 SELECT * FROM
2     vista_usuario_perfil;
```

Listing 35: Select en cuarta vista.

DISEÑO DE TEMPLATES EN HTML

Primero se va a crear un home o página inicial como se observa en el Listing 36 que es lo primero que va a observar el usuario para esto y facilitar la navegación del usuario se ha realizado un navbar en el cual constan las pestañas de home, trámites, servicios, contactenos y base legal.

```
1 <nav class="navbar navbar-expand-lg navbar-dark bg-primary">
2   <div class="container-sm">
3     <form class="d-flex">
4       
6     </form>
7     <div class="collapse navbar-collapse" id="navbarSupportedContent">
8       <ul class="navbar-nav me-auto mb-2 mb-lg-0">
9         <li class="nav-item">
10          <a class="nav-link active" aria-current="page" href="{ url_for
11            ('home') }}">Instituci n</a>
12        </li>
13        <li class="nav-item">
14          <a class="nav-link active" href="{ url_for('tramites') }}">
15            Tr mites</a>
16        </li>
17        <li class="nav-item">
18          <a class="nav-link active" aria-current="page" href="{ url_for
19            ('servicios') }}">
20            Servicios
21          </a>
22        </li>
23        <li class="nav-item">
24          <a class="nav-link active" href="{ url_for('contactenos') }}"
25            aria-disabled="true">Contactenos</a>
26        </li>
27        <li class="nav-item">
28          <a class="nav-link active" href="{ url_for('baselegal') }}"
29            aria-disabled="true">Base Legal</a>
30        </li>
31      </ul>
32    </div>
33  </div>
34 </nav>
```

```

27 <a class="btn btn-outline-light" aria-current="page" href="{
28 url_for('logout')}>Iniciar Sesion</a>
29 </form>
30 </div>
31 </nav>

```

Listing 36: Home o página inicial.

Cabe recalcar que este menú se mantendrá para las 5 pestañas externas al login por ende el mismo proceso se repitió en todas las pestañas, a continuación, se da diseño a la página de trámites dentro de un contenedor se crean las pestañas con diseño de bootstrap 5 para plasmar la información referente.

Para no llenar el reporte del código de las demás pestañas que han sido creadas en HTML, se ha subido al repositorio de GitHub.

```

1 <h3 class="display-4">Trámites destacados</h3>
2 <div class="container">
3 <div class="card-group mt-3">
4 <div class="card text-center border-info">
5 <div class="card-body">
6 
8 <h4 class="card-title">Jubilación por vejez</h4>
9 <p class="card-text">Cumplir con el derecho a pensiones mensuales
10 vitalicias a los afiliados.</p>
11 </div>
12 </div>
13 <div class="card text-center border-info">
14 <div class="card-body">
15 
17 <h4 class="card-title">Denuncias</h4>
18 <p class="card-text">Si tienes alguna queja o reclamo sobre uno de
19 nuestros servicios puedes hacerlos desde la pestaña contactenos</p>
20 </div>
21 </div>
22 <div class="card text-center border-info">
23 <div class="card-body">
24 
26 <h4 class="card-title">Rol de pensión Jubilados</h4>
27 <p class="card-text">Información detallada de la mensualidad que
28 acredita el IESS.</p>
29 </div>
30 </div>
31 </div>

```

Listing 37: Menú externo en HTML.

Una vez terminada la parte externa se comienza con la parte del login como indica el Listing 38. El mismo será usado y creado para que se pueda ingresar como un administrador o un usuario y ver sus datos en una imagen. Para ingresar, se debe usar un usuario y contraseña que exista en la base de datos, de lo contrario no se podrá utilizar los servicios dentro de la página.

Todo el código se encuentra subido en el repositorio, el cual puede ser descargado y ejecutado.

```

1 <form class="form-signin" action="/login" method="POST">
2 

```



```

3      {% with messages = get_flashed_messages() %}
4      {% if messages %}
5      <br />
6      {% for message in messages %}
7      <div class="alert alert-primary alert-dismissible" role="alert">
8          <strong>{{ message }}</strong>
9          <button type="button" class="btn-close" data-bs-dismiss="alert"
            aria-label="Close"></button>
10     </div>
11     {% endfor %}
12     {% endif %}
13     {% endwith %}
14     <h1 class="h3 mb-3 fw-normal">Login</h1>
15     <div class="form-floating">
16         <input type="text" class="form-control" id="username" name="username"
            placeholder="Username">
17         <label for="username">Username</label>
18     </div>
19     <div class="form-floating mt-2">
20         <input type="password" class="form-control" name="password"
            placeholder="Password">
21         <label for="password">Password</label>
22     </div>
23     <button class="w-100 btn btn-lg btn-primary" type="submit">Sign in</
24     button>
</form>

```

Listing 38: Validación de usuario y contraseña.

CONFIGURACIÓN DE TEMPLATES CON PYTHON

Se debe conectar la base que ya ha sido creada en SQL Plus e ingresamos el usuario y la contraseña.

```

1  import oracledb
2  import getpass
3  import csv
4
5  p_username = "system"
6  pw = "0034"
7  p_password = pw
8  p_dns = "localhost/xe"
9  p_port = "1521"
10
11 con = oracledb.connect(user=p_username, password=p_password, dsn=p_dns, port=
    p_port)

```

Listing 39: Login.

Se valida el usuario y contraseña, estas deben ser las mismas que la base cargada para poder acceder al sistema como se observa en el Listing 40.

```

1  if request.method == 'POST':
2      username = request.form.get("username")
3      password = request.form.get("password")
4      logged_user = ModelUser.login(db, username)
5      if logged_user != None:
6          if logged_user.password == password:
7              if logged_user.perfil == "Usuario":
8                  return render_template("ingreso.html", result = logged_user
9
10 )

```

```

9         else:
10             return render_template("admin.html", result = logged_user)
11     else:
12         flash("Contrase a Incorrecta...")
13         return render_template('auth/login.html')
14     else:
15         flash("Usuario no registrado...")
16         return render_template('auth/login.html')
17 else:
18     return render_template('auth/login.html')

```

Listing 40: Validar Login.

4 Evaluación Experimental

Nuestro proyecto ha sido realizado luego de hacer los respectivos análisis, se han definido entidades, de esas entidades se han seleccionado listas de atributos que son necesarias e importantes, se ha creado relaciones entre las entidades para que tengan un mismo fin y se procede a analizar los datos sintéticos obtenidos por medio de consultas o queries, y así se prueba su funcionamiento. Además de llegar a crear vistas, triggers y procedimientos con todo el conocimiento que se ha adquirido en la materia.

4.1 Configuración experimental

Para el desarrollo de este proyecto se ha utilizado dos laptops con un sistema operativo Windows 10. A la vez se ha usado el software "Visual Studio Code" con lenguaje de programación Python, que es un editor de código fuente, este fue desarrollado por Microsoft y se lo puede usar en Windows, Linux, macOS y Web. Para la creación de funciones, queries, triggers, vistas y cargar cualquier tipo de datos se ha utilizado a SQL Plus, que es el cmd de Oracle que nos sirve para interactuar con la base de datos. Básicamente, es un intérprete SQL con algunas opciones de edición y formateo de resultados. Con ayuda de Gliffy se han diseñado el diagrama entidad relación, diagrama lógico y diagrama físico de este proyecto.

En la Figura 4 se muestra que se ha subido el código fuente y también el script de la database de nuestro proyecto y los scripts de objetos del mismo. Se conoce a GitHub como un repositorio online totalmente gratis que permite gestionar los archivos de este proyecto y controlar nuestros códigos, también permite crear y subir repositorios de código para mantenerlo almacenado.

Para una mejor visualización de los documentos generados, visitar el repositorio: Sistemas-de-Bases-De-Datos-8393/Tercer Parcial/Proyecto 3 Unidad/
<https://github.com/NayeliMTC/Sistemas-de-Bases-De-Datos-8393.git>

4.2 Resultados

Con ayuda de los tablas y los cinco mil datos ue han sido cargados en SQL Plus en las anteriores unidades se ha podido realizar los procedimientos, triggers, vistas y crear consultas con la base de datos. Con ayuda de los cinco mil datos se han podido realizar las consultas de ejemplo para probar su funcionamiento y sacar valore estadísticos, estos serán útiles, ya que como son falsos, no afectarán a ningunda entidad real. Es importante tener bien creadas las tablas y bien establecidas sus relaciones para que los resultados sean correctos.

Índices creados

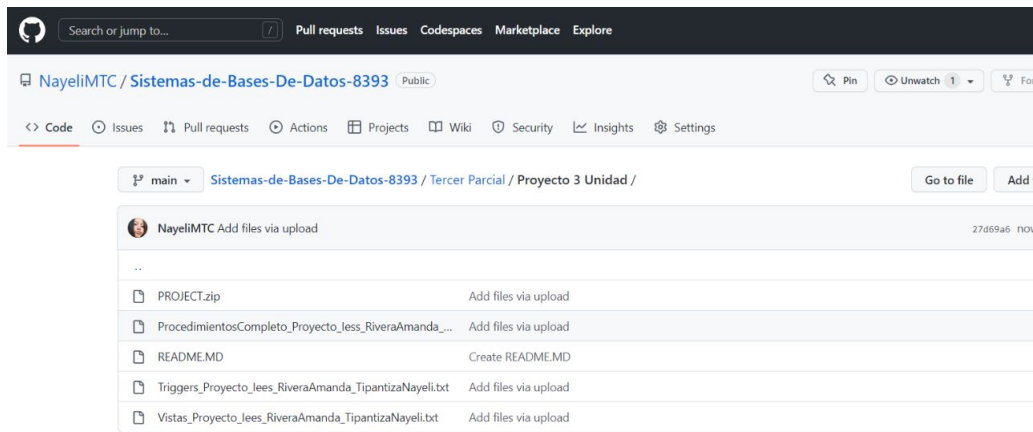


Figure 4: Carpetas creadas en GitHub.

Se ha creado un script de los index que se han considerado necesarios para la optimización de procesos, cada uno ha sido ejecutado en el SQL Plus y se han creado con éxito.

El primer índice se puede visualizar en la Figura 5. El cual se ha creado en la tabla de Persona, considerando que otro dato único aparte del ID de esta tabla podría ser el correo de la Persona.

```
SQL> CREATE UNIQUE INDEX ix_uniq_persona
2 ON TB_PERSONA(VAR_CORREO_PERSONA);

Índice creado.
```

Figure 5: Índice creado en TB_PERSONA.

El otro índice creado se puede visualizar en la Figura 6. El cual se ha creado en la tabla de Usuario, considerando que otro dato único aparte del ID de esta tabla podría ser el nombre de Usuario.

```
SQL> CREATE UNIQUE INDEX ix_uniq_usuario  
2 ON TB_USUARIO(VAR_NOMBRE_USUARIO);  
  
Índice creado.
```

Figure 6: Índice creado en TB_PERSONA.

El siguiente índice creado se puede visualizar en la Figura 7. El cual se ha creado en la tabla de Empresa, considerando que otro dato único aparte del ID de esta tabla podría ser el nombre de la Empresa.

```
SQL> CREATE UNIQUE INDEX ix_uniq_empresa  
2 ON TB_EMPRESA(VAR_NOMBRE_EMPRESA);  
  
Índice creado.
```

Figure 7: Índice creado en TB_PERSONA.

Se puede visualizar en la Figura 8 a otro índice que ha sido creado. El cual se ha creado en la tabla de Base Legal, considerando que otro dato único aparte del ID de esta tabla podría ser la descripción de la Base Legal.

```
SQL> CREATE UNIQUE INDEX ix_uniq_baselegal  
2 ON TB_BASELEGAL(VAR_DESCRIPCION_BASELEGAL);  
  
Índice creado.
```

Figure 8: Índice creado en TB_BASELEGAL.

Y el último índice que ha sido creado es el que se observa en la Figura 9. El cual se ha creado en la tabla de Código, considerando que otro dato único aparte del ID de esta tabla podría ser el detalle del Código.

```
SQL> CREATE UNIQUE INDEX ix_uniq_codigo  
2 ON TB_CODIGO(VAR_DETALLE_CODIGO);  
  
Índice creado.
```

Figure 9: Índice creado en TB_CODIGO.

Triggers creados

Se han considerado cinco triggers de importancia para este proyecto, a continuación veremos los resultados de su ejecución.

El primer Trigger que se observa en la Figura 10 ha sido creado luego de crear la tabla auditoría, aquí se observa el nombre de la misma y la acción que se realiza en el Trigger, en este caso un Update.

```
SQL> select * from TB_AUDITORIA;
```

VAR_ID_AUDITORIA	VAR_NOMBRE_TABLA	VAR_ACCION	DT_FECHA
18cd7c7aa37047e782caa79f1d0d5de9	TB_AFILIACION	UPDATE	02/03/23

Figure 10: Resultado del primer Trigger.

El siguiente Trigger que se observa en la Figura 11 ha sido creado para ser disparado al momento en que la edad del hijo sea mayor o igual a 18 años indicando que ya no cuenta con el seguro.

```
SQL> select * from TB_HIJO where VAR_ID_HIJO = 'b7f626e2685b46c6aabbcd26ce072188';
```

VAR_ID_HIJO	VAR_NOMBRE_HIJO	INT_EDAD_HIJO	VAR_ID_PERSONA
b7f626e2685b46c6aabbcd26ce072188	Laura Garza	18	81ca504246914af1b1d26d820ab592c7

Figure 11: Resultado del segundo Trigger.

El Trigger que se observa en la Figura 12 ha sido creado para cuando se actualice el número de aportaciones, en caso de ser mayor de 24, esa persona puede reclamar sus fondos de cesantías.

```
SQL> UPDATE TB_APORTACION
2 SET INT_NO_APORTACION = 24 WHERE VAR_ID_APORTACION = '82a9098ce7694d5382c777bfe7195eb6';
Ya puede reclamar los fondos de cesantía
1 fila actualizada.
```

Figure 12: Resultado del tercer Trigger.

El cuarto Trigger que se observa en la Figura 13 ha sido creado para que en el momento en que se inserte los datos de una persona, y esta sea mayor de 65 años indicará un mensaje que es acreedora a la pensión otorgada por el IESS.

```
Disparador creado.
SQL> INSERT INTO TB_PERSONA
2 (VAR_ID_PERSONA, VAR_NOMBRE_PERSONA, VAR_TEL_PERSONA, VAR_GENERO_PERSONA, DT_FECHA_NAC_PERSONA, VAR_CORREO_PERSONA, VAR_PROVINCIA_PERSONA, VAR_CANTON_PERSONA, VAR_PARROQUIA_PERSONA)
3 VALUES ('1845612547', 'Victor Borquez', '0961245754', 'Masculino', TO_DATE('1950-01-01', 'YYYY-MM-DD'), 'victorcz@gmail.com', 'Pichincha', 'Quito', 'La Floresta');
La persona es acreedora a la pensión otorgada por la institución
INSERT INTO TB_PERSONA
```

Figure 13: Resultado del cuarto Trigger.

El último Trigger que se observa en la Figura 14 ha sido creado para la inserción del nombre de usuario, en caso de que ya exista, el dato nuevo no será creado e indicará un mensaje donde dice que el usuario ya ha sido creado indicando que hay un error durante la ejecución del disparador.

```
SQL> INSERT INTO TB_USUARIO (VAR_ID_USUARIO, VAR_NOMBRE_USUARIO, VAR_CONTRASENIA_USUARIO, VAR_ID_PERSONA)
2 VALUES ('965', 'nayelimtc', 'ponymalta123', '1752373142');
INSERT INTO TB_USUARIO (VAR_ID_USUARIO, VAR_NOMBRE_USUARIO, VAR_CONTRASENIA_USUARIO, VAR_ID_PERSONA)
*
ERROR en línea 1:
ORA-20001: El nombre de usuario ya existe en la tabla.
ORA-06512: en "NAIESS2.TRG_TB_USUARIO_BEFORE_INSERT", línea 6
ORA-04088: error durante la ejecución del disparador
'NAIESS2.TRG_TB_USUARIO_BEFORE_INSERT'
```

Figure 14: Resultado del quinto Trigger.

Procedimientos creados

A continuación, se mostrarán las ejecuciones de los procesos que han sido creados en SQL Plus, y lo que se obtiene como resultado de su ejecución.

El primer proceso indica a las personas que tienen más de 24 procesos, de ser así, todas esas personas pueden ser acreedoras al préstamo quirografario.

```
SQL> EXEC sp_consulta_aportaciones;

Procedimiento PL/SQL terminado correctamente.

N|| de Juego de Resultados 1
```

VAR_NOMBRE_PERSONA	INT_NO_APORTACION	VAR_MONTO_APORTACION
Mr. Joseph Brown	75	757
Leslie Frost	34	190
Robert Webb	86	832
Michelle Henry	83	651
Dana Cuevas	80	661
John Berger	32	291
Sarah Walker	30	870
Kevin Hensley	54	129
Jessica Bell	87	859
Jeffrey Green	65	836
Scott Wilson	46	734

Figure 15: Resultado del primer Proceso.

El siguiente proceso indica a las personas que tienen acceso al servicio de salud, porque tienen más de 3 aportaciones realizadas en el IESS como se observa en la Figura 16.

VAR_NOMBRE_PERSONA	INT_NO_APORTACION	VAR_MONTO_APORTACION
Beth Fisher	47	037
Ashley Rivera	76	177
Ana Johnston	48	701
Christine Lee	30	636
Daniel Smith	44	402
Kimberly Francis	96	453
David Gutierrez	40	614
Mr. William Maxwell	37	143
Denise Wright	85	806
Jacob Harris	91	254
Brian Hobbs	43	525

Figure 16: Resultado del segundo Proceso.

El tercer proceso muestra los artículos con su código, los cuales han sido cambiados con el tiempo a un estado inactivo, esto ha llegado a afectar a las personas que eran aseguradas.

```
SQL> EXEC sp_codigo_afiliacion;
```

Procedimiento PL/SQL terminado correctamente.

N|| de Juego de Resultados 1

INT_ARTICULO	VAR_NOMBRE_CODIGO	VAR_ESTADO	VAR_NOMBRE_AFILIACION
31	Codigo del trabajo	Inactivo	Asegurado

Figure 17: Resultado del tercer Proceso.

El siguiente proceso muestra a las personas de la provincia de Pichincha que han realizado aportaciones menores a 55 dólares y la fecha en que realizaron dicha aportación, esto ayuda realizar cálculos de de lo que al afiliado le corresponde entregar un aporte al IESS del 9,45% de su sueldo o salario, mientras que al empleador, el 11,15% del salario del trabajador como se observa en la Figura 18.

```
SQL> EXEC sp_consulta_aportacion;
```

Procedimiento PL/SQL terminado correctamente.

N|| de Juego de Resultados 1

VAR_NOMBRE_PERSONA	VAR_PROVINCIA_PERSONA	MONTO APOR
FECHA AP		
Nayeli Cumbal	Pichincha	897
09/07/22		

Figure 18: Resultado del cuarto Proceso.

El último proceso que se ha creado es para mostrar solo a las personas que son Pensionistas y consten en el sistema del Iess para contabilizar cuantas personas son acreedoras de la pensión otorgada por la institución tal y como lo muestra la Figura 19.

```

SQL> EXEC sp_consulta_usuario;

Procedimiento PL/SQL terminado correctamente.

Nº de Juego de Resultados 1

VAR_NOMBRE_USUARIO                                VAR_NOMBRE_ROL
-----
VAR_NOMBRE_PERFIL
-----
kyrat                                              Pensionista
Usuario
gtipantiza                                       Pensionista
Usuario

```

Figure 19: Resultado del quinto Proceso.

Vistas creadas

Se han creado cuatro vistas para esta base de datos, a continuación se verán las ejecuciones de cada una y los resultados que indican las vistas que han sido creadas.

La primera vista es para visualizar a todos los niños que son mayores de edad, es decir, que tienen 18 años y más, por lo tanto, ya no podrán tener accesos a los servicios del IESS, se les puede observar en la Figura 20.

```

SQL> Select * from vista_hijo_mayores;

VAR_NOMBRE_HIJO      INT_EDAD_HIJO
-----
Rachel Blair         18
Laura Garza          18
Taylor Morgan        18
Tonya Whitehead      18
Mark Little          18
Randall Watson       18
Deborah Jordan       18
Melissa Garza        18
Kelly Johnson        18

```

Figure 20: Resultado de la primera Vista.

La segunda vista indica a los códigos que se encuentran inactivos y muestra la base legal a la que perteneces como se visualiza en la Figura 21.


```
SQL> Select * from vista_codigo_inactivo;
```

Codigo	Estado del	Base Legal
Código Orgánico Integral	Inactivo	Ley Orgánica del Sistema Nacional de Contratación Pública
Código del Trabajo	Inactivo	Norma Creación
Código Civil	Inactivo	Norma Creación
Código Civil	Inactivo	Ley de Seguridad Social
Código del Trabajo	Inactivo	Ley de Seguridad Social
Código Civil	Inactivo	Ley Orgánica del Sistema Nacional de Contratación Pública
Código del Trabajo	Inactivo	Ley de Seguridad Social
Código del Trabajo	Inactivo	Norma Creación
Código del Trabajo	Inactivo	Norma Creación
Código del Trabajo	Inactivo	Ley de Seguridad Social
Código del Trabajo	Inactivo	Ley Orgánica del Sistema Nacional de Contratación Pública

Figure 21: Resultado de la segunda Vista.

La siguiente vista ejecutada como se muestra en la Figura 22 muestra solo a las personas que son afiliadas y realizan aportaciones de manera voluntaria.

```
SQL> SELECT
2      *
3  FROM
4      vista_afiliado_voluntario;
```

Persona	Afiliación	Tipo Afiliacion
Christopher Campos	Voluntario	Afiliado
Stephanie Mitchell	Voluntario	Afiliado
Dr. Wendy Castro	Voluntario	Afiliado
Susan Fuller	Voluntario	Afiliado
Joseph Smith	Voluntario	Afiliado
Kelly Dixon	Voluntario	Afiliado
Jacob Rogers	Voluntario	Afiliado
Dr. Michael Carr	Voluntario	Afiliado
Theresa Mathews	Voluntario	Afiliado
Rebecca Houston	Voluntario	Afiliado
Angel Nunez	Voluntario	Afiliado

Figure 22: Resultado de la tercera Vista.

La cuarta vista como se muestra en la Figura 23 muestra solo a las personas que tienen su perfil creado hace dos años, puede indicar que no tienen mucho tiempo en el sistema.

```
SQL> SELECT
2      *
3  FROM
4      vista_usuario_perfil;
```

VAR_NOMBRE_USUARIO	VAR_DESCRIPCION_PERFIL
Tracy Conner	Creado hace 2 años
Shelby Reynolds	Creado hace 2 años
Mr. Gregory Douglas	Creado hace 2 años
Brandon Salinas	Creado hace 2 años
Kayla Hansen	Creado hace 2 años
Tara Wagner	Creado hace 2 años
Anna Owens	Creado hace 2 años
Tanya Jarvis	Creado hace 2 años
Carol Lang	Creado hace 2 años
Linda Roman	Creado hace 2 años
Bruce Ford	Creado hace 2 años

Figure 23: Resultadon de la cuarta Vista.

Ventanas del aplicativo o sistema creado

Luego de crear el sistema con ayuda de la herramienta Visual Studio Code, usando códigos de programación y validando los datos con la base de datos, se han podido crear un sistema que ayudará al insituto con sus servicios y atenciones al usuario, el cual será sencillo de manejar.

En la página principal se encontrará información esencial acerca de quiénes somos y una pequeña introducción del sistema.

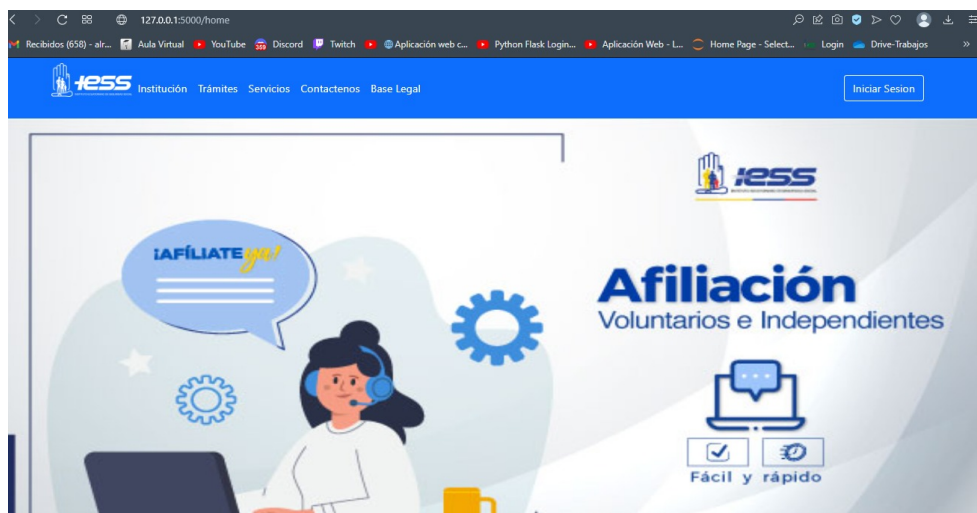


Figure 24: Home.

En la Figura 25 se encuentra la sección de los trámites destacados donde se obtiene información de la gestión o diligenciamiento que se realiza para obtener un resultado, en pos de algo, o los formulismos necesarios para resolver una cosa.

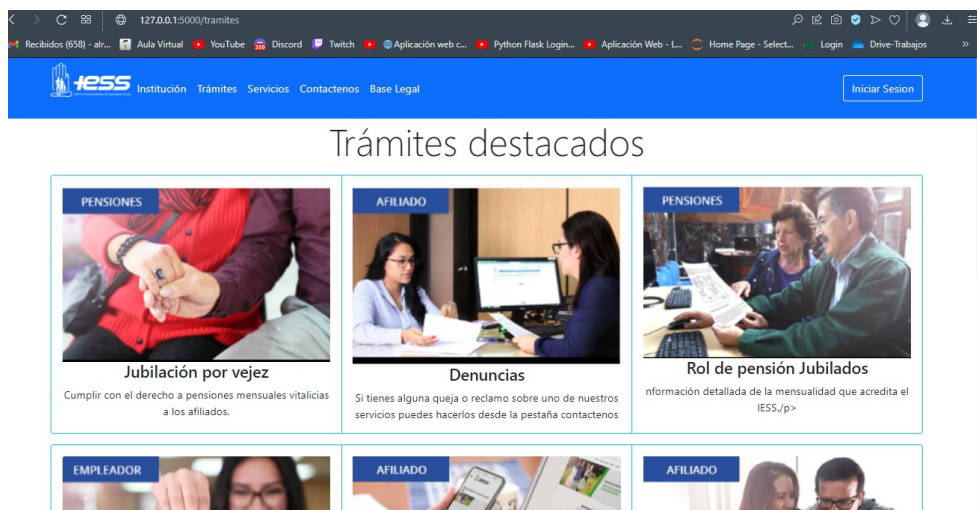


Figure 25: Trámites.

La Figura 26 muestra los servicios que ofrece el instituto, entre ellos están la asistencia médica, quirúrgica, farmacéutica, hospitalaria y de rehabilitación. Pensiones mensuales temporales o permanentes.

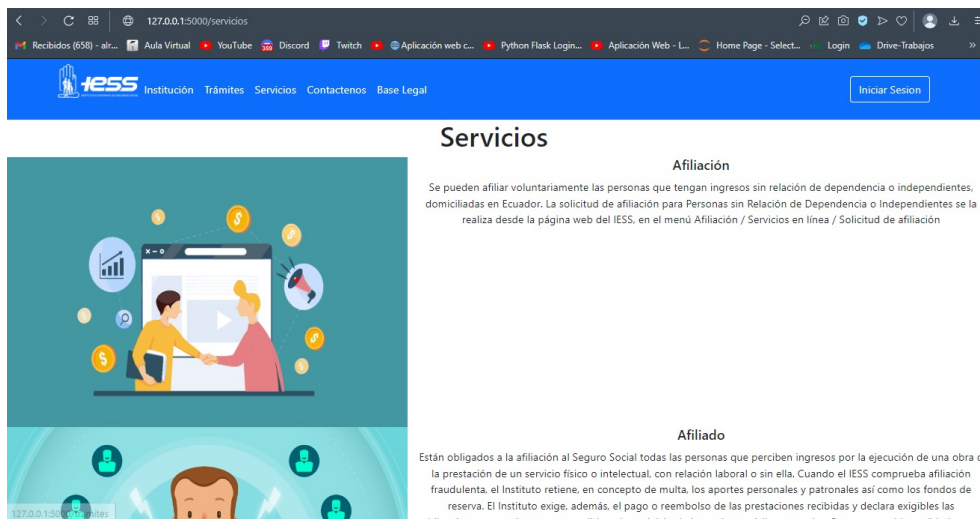


Figure 26: Servicios.

En la Figura 27 se observa la página en donde se encuentra un formulario de contacto para que el usuario pueda llenarlo realizando preguntas sin necesidad de acceder a la página, ya que puede ser una persona que no cuenta con su usuario y contraseña, pero necesita información.

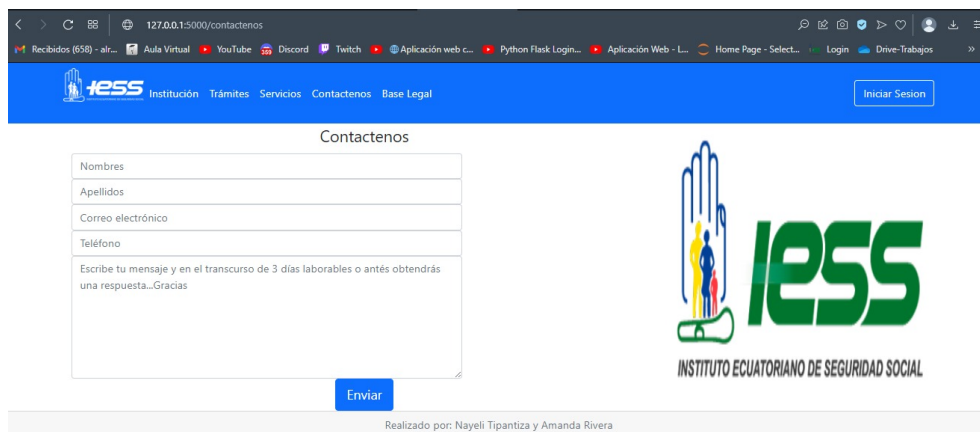


Figure 27: Contacto con el instituto.

La Figura 28 se observa la página en donde se encuentra la base legal del insitituto. Estos son documentos de carácter oficial y legal que se utiliza cuando se requiere salvaguardar a las dos partes en una negociación, cuando se refiere a marketing de contenidos, siendo una serie de leyes o regulaciones que protegen a los propietarios de plagio o copias de trabajos originales.

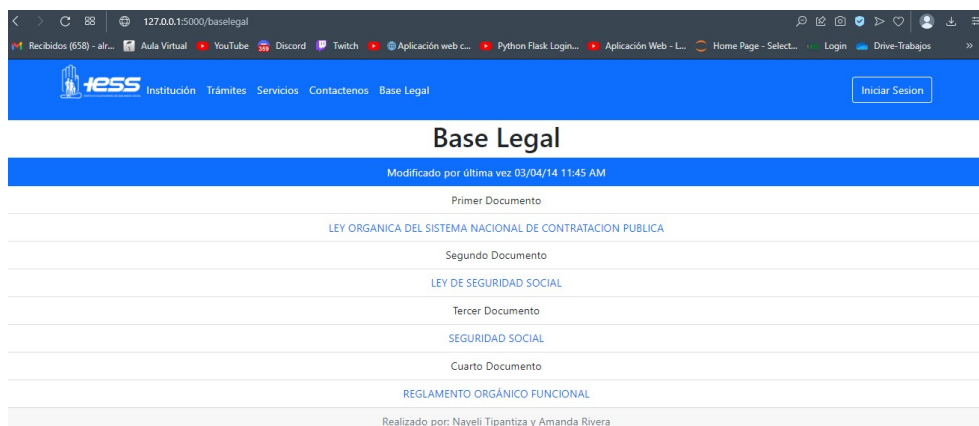


Figure 28: Base Legal.

Una vez visitadas todas las páginas externas del menú principal, se necesita realizar un Login para ver las acciones que un usuario con acceso a este sistema puede realizar y los servicios a los que puede acceder.

Lo primero es dar clic en donde dice Iniciar sesión y nos enviará a la página de acceso en donde se debe ingresar el usuario y contraseña del usuario o administrador, en la Figura 29 se puede observar que se ha ingresado con los datos de un administrador.

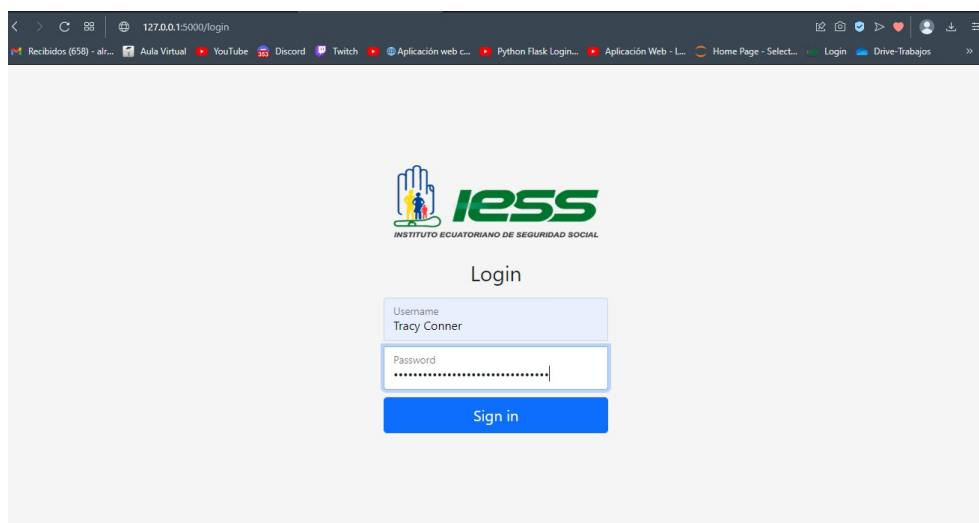


Figure 29: Login.

Luego de dar clic en Sign in, podremos observar a lo que tiene acceso un administrador, en la Figura 30 podemos visualizar que se puede actualizar los datos de un usuario, gestionar y un listado de los mismos.

127.0.0.1:5000/buscar

Recibidos (658) - alr... Aula Virtual YouTube Discord Twitch Aplicación web c... Python Flask Login... Aplicación Web - L... Home Page - Select... Login Drive-Trabajos

IESS Inicio Gestión de Usuarios Listado de Usuarios Cerrar Sesión

Ingrese nombre del afiliado Buscar

Sean Benitez
1988-10-21
Pensionista

Hombre
Voluntario
Activo

Actualizar

Realizado por: Nayeli Tipantiza y Amanda Rivera

Figure 30: Actualizar Usuario.

En la Figura 31 podemos visualizar a todos los usuarios que se encuentran registrados en la base de datos del IESS, esta lista solo la puede visualizar el administrador.

127.0.0.1:5000/epc

Recibidos (658) - alr... Aula Virtual YouTube Discord Twitch Aplicación web c... Python Flask Login... Aplicación Web - L... Home Page - Select... Login Drive-Trabajos

IESS Inicio Gestión de Usuarios Listado de Usuarios Cerrar Sesión

Gestión de Usuarios

Nombre	Genero	Fecha de Nacimiento	Afiliación	Tipo Afiliación	Estado Afiliación
Mr. Joseph Brown	Hombre	1981-07-04 00:00:00	Voluntario	Pensionista	Inactivo
Lisa Guerra	Hombre	1994-04-20 00:00:00	Asegurado	Pensionista	Activo
Leslie Frost	Mujer	2003-10-16 00:00:00	Voluntario	Pensionista	Activo
Connie Welch	Mujer	1982-11-17 00:00:00	Voluntario	Afiliado	Inactivo
Robert Webb	Hombre	2001-07-06 00:00:00	Asegurado	Afiliado	Inactivo
Michelle Henry	Hombre	2002-07-02 00:00:00	Asegurado	Pensionista	Activo
Dana Cuevas	Hombre	2005-11-22 00:00:00	Voluntario	Afiliado	Activo
Craig Butler	Hombre	1996-10-08 00:00:00	Voluntario	Pensionista	Inactivo
John Berger	Mujer	1997-07-05 00:00:00	Voluntario	Pensionista	Activo
James Baker	Hombre	1990-01-19 00:00:00	Asegurado	Pensionista	Activo

Figure 31: Listado de usuarios.

Ahora observaremos lo que un usuario puede realizar o visualizar desde su inicio de sesión en el sistema. En la Figura 32 podemos ver la información principal de un usuario, todos estos datos se encuentran en la base de datos.

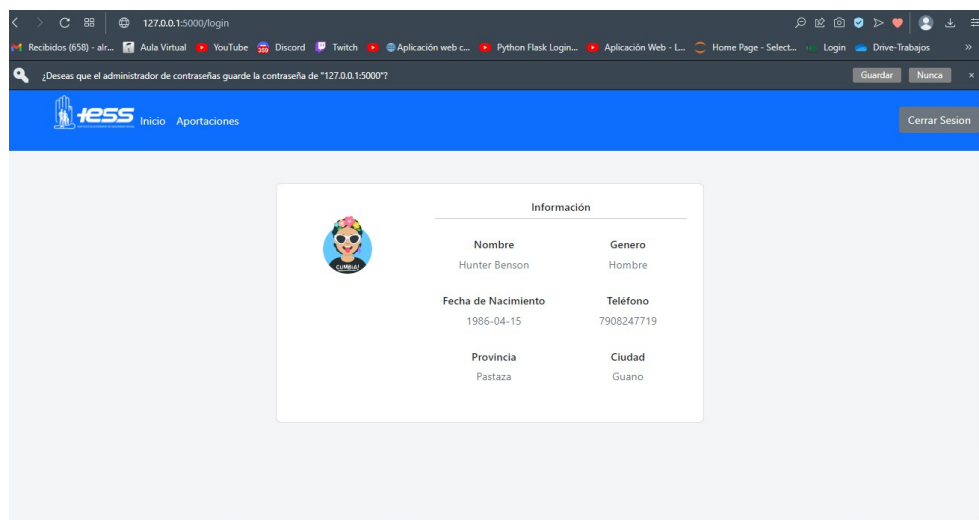


Figure 32: Perfil usuario.

El usuario puede visualizar las aportaciones que ha realizado, el monto y la fecha de su última aportación al sistema como se observa en la Figura 33. Luego si desea puede cerrar sesión asegurándose de que nadie puede ingresar a ver sus datos o pedir servicios.

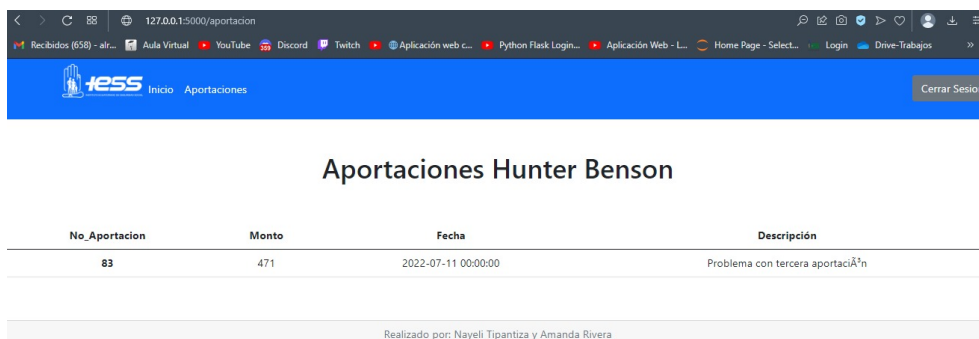


Figure 33: Aportaciones de usuario.

5 Conclusión

El proyecto final de la creación de una aplicación para solucionar la problemática en el Instituto de Seguro IESS en Ecuador ha sido un desafío importante en términos de diseño, implementación y optimización del back-end. Como ya se sabe, uno de los objetivos de este proyecto es demostrar la habilidad para diseñar y desarrollar una aplicación funcional y eficiente, considerando las mejores prácticas y tecnologías disponibles. La solución propuesta ofrece una respuesta práctica y efectiva

a un problema real que es sobre la mala intención del instituto IESS y se espera que tenga impacto positivo en los usuarios finales. En conclusión, la aplicación propuesta para mejorar la atención en el IESS una herramienta esencial para simplificar y mejorar la experiencia de los usuarios. Además, la implementación de tecnologías modernas en el desarrollo de la aplicación asegurará una respuesta rápida y eficiente para las solicitudes de los usuarios. La implementación de un back-end bien estructurado y optimizado, junto con un front-end atractivo e intuitivo, garantiza la calidad y usabilidad de la aplicación desarrollada.

Este proyecto ha demostrado la importancia de un diseño cuidadoso del back-end en una aplicación para garantizar la eficiencia y la escalabilidad. Se espera que la aplicación propuesta contribuya a mejorar la calidad de la atención en el Instituto de Seguro IESS en Ecuador y brinde una solución efectiva a la problemática planteada.

References

- [1] D. J., “Desarrollo de un sistema informático para gestionar el registro de eventos,” *Proyecto de tesis*, 2022.
- [2] Iess., “Afiliados y servicios,” *Entrevista a afiliados del iess*, 2022.
- [3] Barreiro, “El caso del seguro social campesino,” *Iess y proteccion*, 2022.
- [4] M. Z. Gabriel, “Servicios y prestaciones,” *Iess y sus problemas*, 2022.
- [5] Ortega, “¿quiénes somos?” *Iess*, 2022.
- [6] F. J. Webinars, “Qué son las estructuras de datos y por qué son tan útiles,” *Estructuras de datos más conocidas*, 2019.