

Universidad Tecnológica de El Salvador



FACULTAD DE INFORMÁTICA Y CIENCIAS APLICADAS ESCUELA DE INFORMÁTICA

PROGRAMACION 2

DOCENTE:

JAVIER ANTONIO VALLE CAMPOS

SECCIÓN: 3

TEMA DE INVESTIGACION:

EQUIPO

TEMIS

INTEGRANTES:

N.º	CARNET	APELLIDO	NOMBRE	PORCENTAJE DE TRABAJO
1	27-1207-2023	Andrade García	Rolando Antonio	100
2	25-1819-2017	Gamez Molina	Nelson Edgardo	100
3	27-1414-2016	Guzman Martinez	Walter Alejandro	100
4	27-1293-2023	Estrada Gonzalez	Jesus Alejandro	100
5	25-3389-2022	Corea Espino	José Luis	100

20/03/2024

Índice

TEMIS	4
Problema investigado.....	5
Justificación	5
Objetivos de la investigación.....	6
Objetivo General	6
Objetivos Específicos.....	6
Limitaciones del Proyecto	7
Limitaciones Económicas	7
Tecnología y Desarrollo	7
Regulaciones y Cumplimiento.....	7
Resistencia al Cambio.....	7
Factibilidad del Proyecto	8
Demanda del Mercado.....	8
Público Objetivo Amplio.....	8
Competencia Limitada	8
Beneficios Operativos	8
Viabilidad Técnica.....	8
Cronograma de Actividades (16 Semanas)	9
Fase 1: Planificación y Análisis (2 semanas).....	9
Fase 2: Diseño (2 semanas).....	9
Fase 3: Desarrollo (8 semanas)	10
Fase 4: Pruebas (3 semanas).....	10
Fase 5: Presentación del Proyecto (1 semana)	11
Marco Teórico	11
Capacidades y herramientas básicas:	11
Bases de datos.....	11
Casos de uso.....	13
Creación de una nuestra base de datos.....	14
Diccionario de Datos	14
Relaciones entre Tablas.....	16
Creación de Tablas	17
Clientes.....	17

Abogados.....	17
Asistentes	19
Casos	19
Facturas	20
Usuarios	20
Documentos	21
Procedimientos Almacenados.....	21
Crear Abogado	21
Listar los Abogados	21
Actualizar abogados	22
Eliminar Abogados por ID	22
Creación de Clientes.....	23
Buscar Clientes	23
Actualización de cliente	24
Eliminación de Cliente.....	24
Crear Usuario	25
Buscar Usuarios.....	25
Actualizar Usuario	26
Eliminar Usuario.....	26
Entidad Relación.....	27
Seguridad del sistema	27
Claves de acceso	27
Roles de usuario	28
Privilegios	28
Glosario de términos.....	29

TEMIS

"TEMIS" es una aplicación web diseñada para revolucionar la gestión de perfiles de abogados y la asignación inteligente de casos en empresas de servicios legales. Esta plataforma ofrece una solución integral para optimizar la eficiencia operativa y la experiencia del usuario tanto para abogados como para clientes.

Con TEMIS, las empresas legales pueden gestionar de manera centralizada los perfiles de sus abogados, incluyendo información personal, académica y profesional, lo que facilita la identificación y asignación de abogados a casos específicos. Además, cuenta con un algoritmo inteligente que, basado en diversos criterios como la especialización del abogado, la carga de trabajo y la ubicación geográfica, asigna automáticamente el abogado más adecuado a cada cliente que se registra en la plataforma.

La seguridad de los datos sensibles es una prioridad en TEMIS, con medidas robustas de protección de la información, como el cifrado de datos, el control de acceso y la auditoría de registros, garantizando así la confidencialidad y la integridad de la información almacenada en la plataforma.

En resumen, TEMIS ofrece una solución completa para mejorar la gestión de perfiles de abogados, la asignación de casos y la seguridad de los datos en empresas de servicios legales, impulsando la eficiencia operativa y la satisfacción del cliente.

Problema investigado

La falta de eficiencia en la gestión de perfiles de abogados y asignación de casos en empresas de servicios legales, lo cual puede conducir a una distribución inadecuada de la carga de trabajo, retrasos en la atención de clientes, y riesgos de seguridad en el manejo de datos sensibles.

Justificación

La necesidad de desarrollar una aplicación web que permita a una empresa de servicios legales gestionar de manera eficiente los registros de sus abogados, así como asignar automáticamente un abogado y su especialización a los clientes al registrarse en la plataforma. Se está proponiendo manejar mejor la gestión de perfiles de abogados, la asignación inteligente de abogados a clientes, la seguridad de los datos sensibles y la experiencia del usuario tanto para abogados como para clientes.

Y es por esta necesidad que nace TEMIS, del Griego THEMIS, era una diosa que representa la justicia y la equidad, y es por dicha diosa que nos inspiramos en ella para nombrar esta aplicación web de servicios legales.

Objetivos de la investigación

Objetivo General

Desarrollar una aplicación web que optimice la gestión de perfiles de abogados, la asignación automatizada de abogados a clientes y la seguridad de los datos sensibles, mejorando así la experiencia del usuario para tanto para abogados como para clientes de una empresa de servicios legales.

Objetivos Específicos

- Implementar un sistema de gestión de perfiles de abogados que permita una actualización eficiente de la información personal, académica y profesional de cada abogado, facilitando así su identificación y asignación a casos específicos.
- Desarrollar un algoritmo de asignación inteligente que, basado en criterios como la especialización del abogado, la carga de trabajo y la ubicación geográfica, asigne automáticamente el abogado más adecuado a cada nuevo cliente registrado en la plataforma.
- Garantizar la seguridad de los datos sensibles de los clientes y los abogados mediante la implementación de medidas robustas de protección de la información, como el cifrado de datos, el control de acceso y la auditoría de registros, asegurando así la confidencialidad y la integridad de la información almacenada en la plataforma.

Limitaciones del Proyecto

Nuestro proyecto presenta varias limitaciones que deben ser consideradas:

Limitaciones Económicas: Una de las principales limitaciones es la falta de recursos financieros suficientes para expandir el proyecto a gran escala. Esto puede afectar la capacidad para desarrollar todas las funcionalidades planificadas, contratar personal adicional, y realizar campañas de marketing extensivas.

Tecnología y Desarrollo: La implementación de una aplicación web que cumpla con todos los requisitos técnicos, especialmente en términos de seguridad de datos, puede ser compleja y requerir conocimientos especializados. La falta de acceso a tecnología avanzada o a desarrolladores altamente cualificados puede ralentizar el progreso del proyecto.

Regulaciones y Cumplimiento: La gestión de datos sensibles en el ámbito legal está sujeta a estrictas regulaciones y leyes de protección de datos. Asegurarse de cumplir con todas las normativas vigentes puede ser desafiante y puede requerir asesoramiento legal y técnico continuo.

Resistencia al Cambio: Las empresas de servicios legales pueden mostrar resistencia a adoptar nuevas tecnologías debido a la inercia operativa y la preferencia por métodos tradicionales. La aceptación y adaptación de los usuarios finales (abogados y clientes) puede tomar tiempo.

Factibilidad del Proyecto

A pesar de las limitaciones, el proyecto presenta una alta factibilidad basada en los siguientes factores:

Demanda del Mercado: Existe una necesidad significativa de mejorar la eficiencia en la gestión de perfiles de abogados y la asignación de casos en empresas de servicios legales. La optimización de estos procesos puede proporcionar una ventaja competitiva y satisfacer una demanda creciente por soluciones tecnológicas en el sector legal.

Público Objetivo Amplio: El mercado objetivo para este proyecto es considerablemente grande, incluyendo bufetes de abogados de todos los tamaños, departamentos legales corporativos y organizaciones que requieren servicios legales especializados. La adopción de una plataforma eficiente puede atraer a una amplia gama de usuarios.

Competencia Limitada: Actualmente, la competencia en este nicho específico es limitada, lo que brinda una oportunidad para establecerse como un líder del mercado. La falta de soluciones integrales similares permite captar una cuota de mercado significativa rápidamente.

Beneficios Operativos: La implementación de un sistema que gestione eficientemente los perfiles de abogados y asigne casos de manera inteligente puede mejorar considerablemente la productividad y la satisfacción del cliente. Además, la mejora en la seguridad de los datos puede atraer a clientes preocupados por la confidencialidad y la integridad de su información.

Viabilidad Técnica: Con una planificación adecuada y el uso de tecnologías modernas, es viable desarrollar una aplicación que cumpla con los requisitos funcionales y de seguridad

establecidos. La adopción de metodologías ágiles puede ayudar a superar algunos de los desafíos técnicos y asegurar una implementación exitosa.

Cronograma de Actividades (16 Semanas)

Fase 1: Planificación y Análisis (2 semanas)

Semana 1:

Reunión de inicio del proyecto

Definición de requisitos del proyecto

Análisis de requisitos y elaboración del documento de especificaciones

Semana 2:

Identificación de recursos necesarios (tecnológicos y humanos)

Planificación del proyecto y cronograma detallado

Revisión y aprobación del plan del proyecto

Fase 2: Diseño (2 semanas)

Semana 3:

Diseño de la arquitectura de la aplicación

Creación de wireframes y prototipos de la interfaz de usuario

Semana 4:

Diseño de la base de datos

Revisión y aprobación del diseño

Fase 3: Desarrollo (8 semanas)

Semana 5:

Configuración del entorno de desarrollo en ASP.Net MVC en C#

Configuración de la base de datos

Semana 6-7:

Desarrollo del módulo de gestión de perfiles de abogados (Backend y Frontend)

Semana 8-9:

Desarrollo del algoritmo de asignación inteligente (Backend)

Integración del algoritmo de asignación con la interfaz de usuario

Semana 10-11:

Implementación de medidas de seguridad (cifrado de datos, control de acceso, auditoría de registros)

Semana 12-13:

Desarrollo de funcionalidades adicionales y ajustes según feedback

Fase 4: Pruebas (3 semanas)

Semana 14:

Pruebas unitarias y de integración

Semana 15:

Pruebas de seguridad y performance

Corrección de errores y ajustes finales

Fase 5: Presentación del Proyecto (1 semana)

Semana 16:

Preparación de la presentación del proyecto

Presentación del proyecto

Marco Teórico

Una base de datos es un sistema organizado que permite almacenar, manipular y recuperar información de manera eficiente. Esta información se organiza de acuerdo con un modelo específico, que puede ser relacional, orientado a objetos, jerárquica, entre otros, también pueden clasificarse según el contenido y ámbito de su aplicación, como texto completo que se almacenan fuentes primarias artículos que se utilizan para gestionar finanzas de una empresa.

Microsoft SQL Server es un sistema de gestión de base de datos relacional, desarrollado por la empresa Microsoft. El lenguaje de desarrollo utilizado es Transact-SQL, una implementación del estándar ANSI del lenguaje SQL, utilizado para manipular y recuperar datos, crear tablas y definir relaciones entre ellas.

Capacidades y herramientas básicas:

Bases de datos

Una base de datos es un conjunto organizado y estructurado de información que se almacena de forma sistemática para facilitar su gestión, recuperación y análisis. Las bases de datos son utilizadas en una

amplia variedad de aplicaciones y sectores, como empresas, sector público, científico y bibliotecas.

Algunos puntos importantes sobre las bases de datos son:

Las bases de datos pueden ser estáticas o dinámicas.

Las bases de datos estáticas solo se utilizan para consultar datos y no pueden ser modificadas, mientras que las bases de datos dinámicas permiten tanto la consulta como la modificación de los datos según sea necesario.

Existen diferentes tipos de bases de datos, como las bases de datos relacionales, las bases de datos NoSQL y las bases de datos orientadas a grafos. Cada tipo tiene sus propias características y se utiliza en diferentes contextos y aplicaciones.

Los sistemas de gestión de bases de datos (DBMS, por sus siglas en inglés) son software que permite crear, gestionar y manipular bases de datos. Algunos ejemplos de DBMS populares son MySQL, Oracle, SQL Server.

Se llama base de datos, o también banco de datos, a un conjunto de informaciones de perteneciente un mismo contexto, ordenada de modo sistemático para su posterior recuperación, o transmisión. Existen actualmente muchas formas de bases de datos, que van desde una biblioteca hasta los vastos conjuntos de datos de usuarios de una empresa grande de telecomunicaciones.

En resumen, una base de datos es una herramienta esencial en la gestión de información, que permite almacenar, gestionar y recuperar datos de manera eficiente. Existen diferentes tipos de bases de datos y sistemas de gestión de bases de datos que se utilizan en diversas aplicaciones y contextos. Las bases de datos son fundamentales para la toma de decisiones basada en datos y la optimización de procesos en las organizaciones.

Casos de uso

Las operaciones CRUD (Crear, Leer, Actualizar, Borrar) son las acciones básicas para administrar datos en MySQL. Por ejemplo, la sentencia SELECT nos permite consultar los datos almacenados en una tabla de la base de datos. Por otro lado, la sentencia UPDATE de SQL se utiliza para modificar los valores de un conjunto de registros existentes en una tabla.

Las bases de datos del sistema:

master - Todos los procedimientos, funciones y tablas del sistema que están utilizadas por parte de todas las bases de datos y que están instaladas automáticamente, tanto como las que han sido creado por parte de los administradores del sistema. Además, todas las definiciones en respecto a la seguridad a nivel del servidor, están almacenadas en esta base de datos.

msdb - Almacenamiento de las tareas del agente, los códigos de CLR combinados en el sistema, los paquetes de SSIS, y otros más.

model - El molde de las bases de datos. Cada nueva base de datos se crea como una copia de esta base de datos, menos que algo más estaba definido explícitamente.

tempdb - Base de datos temporal que se crea de nuevo cada vez que el servicio reinicia. Se utiliza para almacenar tablas temporales creadas por parte de los usuarios o el sistema (por ejemplo, en ordenaciones complejos).

Tablas fijas y temporales

Desde la perspectiva lógica, los datos almacenados en las bases de datos en tablas, que mediante ellas se implementa la teoría de las bases de datos relacionales. La tabla se divide en filas y columnas (A veces se les conoce como registros y campos). Las tablas pueden ser fijas o temporales, mientras que en el segundo caso existen físicamente en la base de datos tempdb, y se borran automáticamente en caso de desconexión de la sesión o de la conexión al servidor, depende en el tipo de la tabla temporal.

Tipos de datos

Para cada columna en una tabla y a cada variable o parámetro, se define un tipo de datos que sean almacenados en él, entre ellos:

Números: Números enteros y no enteros en distintos tamaños, y en diferentes niveles de precisión; y auto incremento opcional. [int] [double] [float]

Textos: Cadenas de distintas longitudes, y distintas capacidades de apoyar distintas lenguas. [varchar()] [char]

Fechas: Fechas en distintos niveles de precisión, desde días completos hasta fracciones menores de un segundo, que apoyan fechas a partir del principio del siglo XX o del calendario gregoriano, y la capacidad de diferenciar entre distintos usos de horarios.

Creación de una nuestra base de datos

Diccionario de Datos

Tabla Abogados

Nombre de Campo	Tipo de Dato	Restricciones	Descripción
ID_Abogado	int	PK, NOT NULL	Identificador único del abogado
pNombre	varchar(50)	NULL	Primer nombre del abogado
sNombre	nchar(10)	NULL	Segundo nombre del abogado
pApellido	nchar(10)	NULL	Primer apellido del abogado
sApellido	nchar(10)	NULL	Segundo apellido del abogado
DUI	int	NULL	Documento Único de Identidad
Especialidad	varchar(100)	NULL	Especialidad del abogado
Telefono	varchar(50)	NULL	Número de teléfono del abogado
Email	varchar(100)	NULL	Correo electrónico del abogado

Tabla Asistentes

Nombre de Campo	Tipo de Dato	Restricciones	Descripción
ID_Asistente	int	PK, NOT NULL	Identificador único del asistente
pNombre	varchar(50)	NULL	Primer nombre del asistente
sNombre	varchar(50)	NULL	Segundo nombre del asistente
pApellido	varchar(50)	NULL	Primer apellido del asistente
sApellido	varchar(50)	NULL	Segundo apellido del asistente
DUI	int	NULL	Documento Único de Identidad
WorkedHrs	float	NULL	Horas trabajadas

Tabla Casos

Nombre de Campo	Tipo de Dato	Restricciones	Descripción
Case_ID	int	PK, NOT NULL	Identificador único del caso
Caso_Nombre	varchar(50)	NULL	Nombre del caso
Tipo_Facturacion	varchar(50)	NULL	Tipo de facturación del caso
Facturacion	float	NULL	Monto de facturación del caso

Tabla Clientes

Nombre de Campo	Tipo de Dato	Restricciones	Descripción
ID_Cliente	int	PK, NOT NULL	Identificador único del cliente
PrimNombre	varchar(50)	NULL	Primer nombre del cliente
SegNombre	varchar(50)	NULL	Segundo nombre del cliente
PrimApellido	varchar(50)	NULL	Primer apellido del cliente
SegApellido	varchar(50)	NULL	Segundo apellido del cliente
DUI	int	NULL	Documento Único de Identidad
Client_Edad	int	NULL	Edad del cliente
Nacionalidad	varchar(50)	NULL	Nacionalidad del cliente
Ocupacion	varchar(50)	NULL	Ocupación del cliente
Direccion	varchar(500)	NULL	Dirección del cliente
Telefono	varchar(50)	NULL	Número de teléfono del cliente
Email	varchar(100)	NULL	Correo electrónico del cliente

Tabla Documentos

Nombre de Campo	Tipo de Dato	Restricciones	Descripción
ID_Documento	int	NULL	Identificador único del documento
Nombre_Documento	varchar(50)	NULL	Nombre del documento
ID_Cliente	int	NULL	Identificador del cliente asociado
NombreCliente	nchar(10)	NULL	Nombre del cliente
Case_ID	int	NULL	Identificador del caso asociado

Tabla Facturas

Nombre de Campo	Tipo de Dato	Restricciones	Descripción
ID_Factura	int	PK, NOT NULL	Identificador único de la factura
Case_ID	int	NULL	Identificador del caso asociado
WorkedHrs	float	NULL	Horas trabajadas facturadas
Caso_Nombre	varchar(50)	NULL	Nombre del caso
Tipo_Facturacion	varchar(50)	NULL	Tipo de facturación
ID_Cliente	int	NULL	Identificador del cliente asociado
ID_Abogado	int	NULL	Identificador del abogado asociado
Nombre_Cliente	varchar(100)	NULL	Nombre del cliente
Nombre_Abogado	varchar(1000)	NULL	Nombre del abogado

Tabla Usuarios

Nombre de Campo	Tipo de Dato	Restricciones	Descripción
ID_Usuario	int	PK, NOT NULL	Identificador único del usuario
Nombre_Usuario	varchar(50)	NULL	Nombre de usuario
Contraseña	varchar(50)	NULL	Contraseña del usuario
Usuario_Inserta	varchar(50)	NULL	Usuario que realizó la inserción
Fecha_Inserta	datetime	NULL	Fecha de inserción del registro

Usuario_Modifica	varchar(50)	NULL	Usuario que realizó la modificación
Fecha_Modifica	datetime	NULL	Fecha de modificación del registro
ID_Asistente	int	NULL	Identificador del asistente asociado
ID_Abogado	int	NULL	Identificador del abogado asociado
Rol	varchar(50)	NULL	Rol del usuario

Relaciones entre Tablas

Clientes - Documentos: Un cliente puede tener múltiples documentos. La relación se establece a través de ID_Cliente en la tabla Documentos.

Clientes - Facturas: Un cliente puede tener múltiples facturas. La relación se establece a través de ID_Cliente en la tabla Facturas.

Abogados - Facturas: Un abogado puede estar asociado con múltiples facturas. La relación se establece a través de ID_Abogado en la tabla Facturas.

Casos - Facturas: Un caso puede tener múltiples facturas. La relación se establece a través de Case_ID en la tabla Facturas.

Casos - Documentos: Un caso puede tener múltiples documentos. La relación se establece a través de Case_ID en la tabla Documentos.

```
USE [master]
GO
/***** Object: Database [TemisDB]    Script Date: 3/18/2024 9:06:42 PM *****/
CREATE DATABASE [TemisDB]
    CONTAINMENT = NONE
    ON PRIMARY
    ( NAME = N'TemisDB', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\DATA\TemisDB.mdf' , SIZE = 8192KB , MAXSIZE = UNLIMITED, FILEGROWTH = 65536KB )
    LOG ON
    ( NAME = N'TemisDB_log', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\DATA\TemisDB_log.ldf' , SIZE = 8192KB , MAXSIZE = 2048GB , FILEGROWTH = 65536KB )
    WITH CATALOG_COLLATION = DATABASE_DEFAULT, LEDGER = OFF
GO
ALTER DATABASE [TemisDB] SET COMPATIBILITY_LEVEL = 160
GO
IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))
begin
EXEC [TemisDB].[dbo].[sp_fulltext_database] @action = 'enable'
end
```

```
CREATE DATABASE [TemisDB]
    CONTAINMENT = NONE
    ON PRIMARY
    ( NAME = N'TemisDB', FILENAME = N'C:\Program Files\Microsoft SQL
Server\MSSQL16.MSSQLSERVER\MSSQL\DATA\TemisDB.mdf' , SIZE = 8192KB , MAXSIZE = UNLIMITED,
FILEGROWTH = 65536KB )
    LOG ON
    ( NAME = N'TemisDB_log', FILENAME = N'C:\Program Files\Microsoft SQL
Server\MSSQL16.MSSQLSERVER\MSSQL\DATA\TemisDB_log.ldf' , SIZE = 8192KB , MAXSIZE = 2048GB
, FILEGROWTH = 65536KB )
    WITH CATALOG_COLLATION = DATABASE_DEFAULT, LEDGER = OFF
```



```
GO
ALTER DATABASE [TemisDB] SET COMPATIBILITY_LEVEL = 160
GO
```

Creación de Tablas



Cientes



Permitirá guardar la información de los clientes que contraten a la firma de abogados. Esta es información esencial para a ver seguimientos pertinentes y tener información relevante para el caso.

R.TemisDB - dbo.Ta...isDB - dbo.Cientes				R.TemisDB - dbo.Table_1*			
	Column Name	Data Type	Allow Nulls				
🔑	ID_Cliente	int	<input type="checkbox"/>				
	PrimNombre	varchar(50)	<input checked="" type="checkbox"/>				
	SegNombre	varchar(50)	<input checked="" type="checkbox"/>				
	PrimApellido	varchar(50)	<input checked="" type="checkbox"/>				
	SegApellido	varchar(50)	<input checked="" type="checkbox"/>				
	DUI	int	<input checked="" type="checkbox"/>				
	Client_Edad	int	<input checked="" type="checkbox"/>				
	Nacionalidad	varchar(50)	<input checked="" type="checkbox"/>				
	Ocupacion	varchar(50)	<input checked="" type="checkbox"/>				
	Direccion	varchar(500)	<input checked="" type="checkbox"/>				
	Telefono	varchar(50)	<input checked="" type="checkbox"/>				
	Email	varchar(100)	<input checked="" type="checkbox"/>				
▶			<input type="checkbox"/>				

Abogados

Permite llevar información básica del abogado encargado del caso. Esta información es la única relevante para el cliente, se mantiene solo del lado de la firma.

R.TemisDB - dbo.T...sDB - dbo.Abogados   R.TemisDB - dbo.T...B - dbo

	Column Name	Data Type	Allow Nulls
	ID_Abogado	int	<input type="checkbox"/>
	pNombre	varchar(50)	<input checked="" type="checkbox"/>
	sNombre	nchar(10)	<input checked="" type="checkbox"/>
	pApellido	nchar(10)	<input checked="" type="checkbox"/>
	sApellido	nchar(10)	<input checked="" type="checkbox"/>
	DUI	int	<input checked="" type="checkbox"/>
	Especialidad	varchar(100)	<input checked="" type="checkbox"/>
	Telefono	varchar(50)	<input checked="" type="checkbox"/>
	Email	varchar(100)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Asistentes

Personas encargadas de dar seguimiento a los casos, pedir y guardar documentos relevantes para que el Abogado pueda acceder fácilmente a la información.

R.TemisDB - dbo.Asistentes			
	Column Name	Data Type	Allow Nulls
PK	ID_Asiistente	int	<input type="checkbox"/>
	pNombre	varchar(50)	<input checked="" type="checkbox"/>
	sNombre	varchar(50)	<input checked="" type="checkbox"/>
	pApellido	varchar(50)	<input checked="" type="checkbox"/>
	sApellido	varchar(50)	<input checked="" type="checkbox"/>
	DUI	int	<input checked="" type="checkbox"/>
	WorkedHrs	float	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Casos

Detalles del caso, el tipo de facturación, horas trabajadas. Esta será una tabla para asociar la facturación.

R.TemisDB - dbo.Casos			
	Column Name	Data Type	Allow Nulls
PK	Case_ID	int	<input type="checkbox"/>
	Caso_Nombre	varchar(50)	<input checked="" type="checkbox"/>
	Tipo_Facturacion	varchar(50)	<input checked="" type="checkbox"/>
	Facturacion	float	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Facturas

Sera utilizada para llevar el conteo de horas trabajadas por cada caso, debe haber una diferencia de precios por Abogados y asistentes. También existen casos Pro-Bono que no se le cobra al cliente; y casos que se manejan en paquetes de resolución. Dependiendo del abogado y del tipo de caso.

R.TemisDB - dbo.Facturas		R.TemisDB - dbo.T...emisDB - dbo.Casos	
	Column Name	Data Type	Allow Nulls
▶	ID_Factura	int	<input type="checkbox"/>
	Case_ID	int	<input checked="" type="checkbox"/>
	WorkedHrs	float	<input checked="" type="checkbox"/>
	Caso_Nombre	varchar(50)	<input checked="" type="checkbox"/>
	Tipo_Facturacion	varchar(50)	<input checked="" type="checkbox"/>
	ID_Cliente	int	<input checked="" type="checkbox"/>
	ID_Abogado	int	<input checked="" type="checkbox"/>
	Nombre_Cliente	varchar(100)	<input checked="" type="checkbox"/>
	Nombre_Abogado	varchar(1000)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Usuarios

Información relevante para la aplicación y el mantenimiento. Se deberá respetar la información que se ingresa o se cambia por cada rol. Y habrá diferentes niveles preestablecidos. También se registra los cambios por usuario y hora por auditorias.

R.TemisDB - dbo.Table_1*		R.TemisDB - dbo.Facturas		R.TemisDB - dbo.Casos	
	Column Name	Data Type	Allow Nulls		
▶	ID_Usuario	int	<input type="checkbox"/>		
	Nombre_Usuario	varchar(50)	<input checked="" type="checkbox"/>		
	Contraseña	varchar(50)	<input checked="" type="checkbox"/>		
	Usuario_Inserta	varchar(50)	<input checked="" type="checkbox"/>		
	Fecha_Inserta	datetime	<input checked="" type="checkbox"/>		
	Usuario_Modifica	varchar(50)	<input checked="" type="checkbox"/>		
	Fecha_Modifica	datetime	<input checked="" type="checkbox"/>		
	ID_Asistente	int	<input checked="" type="checkbox"/>		
	ID_Abogado	int	<input checked="" type="checkbox"/>		
	Rol	varchar(50)	<input checked="" type="checkbox"/>		
			<input type="checkbox"/>		

Documentos

Esta tabla será usada para guardar los documentos pertinentes a cada caso. Ya sean documentos de identidad o pruebas relevantes. (aun en investigación para recopilar lo deseado)

R.TemisDB - dbo.T...B - dbo.Documentos			
Column Name	Data Type	Allow Nulls	
ID_Documento	int	<input checked="" type="checkbox"/>	
Nombre_Documento	varchar(50)	<input checked="" type="checkbox"/>	
ID_Cliente	int	<input checked="" type="checkbox"/>	
NombreCliente	nchar(10)	<input checked="" type="checkbox"/>	
Case_ID	int	<input checked="" type="checkbox"/>	
		<input type="checkbox"/>	

Procedimientos Almacenados

Crear Abogado

```
-- Procedimiento almacenado para crear un nuevo abogado
CREATE PROCEDURE sp_CrearAbogado
    @pNombre varchar(50),
    @sNombre nchar(10),
    @pApellido nchar(10),
    @sApellido nchar(10),
    @DUI int,
    @Especialidad varchar(100),
    @Telefono varchar(50),
    @Email varchar(100)
AS
BEGIN
    INSERT INTO Abogados (pNombre, sNombre, pApellido, sApellido, DUI, Especialidad, Telefono, Email, Usuario_Modifica, Fecha_Modifica)
    VALUES (@pNombre, @sNombre, @pApellido, @sApellido, @DUI, @Especialidad, @Telefono, @Email, @pNombre, GETDATE())
END
GO
```

Listar los Abogados

```
-- Procedimiento almacenado para leer todos los abogados
CREATE PROCEDURE sp_LeerAbogados
AS
BEGIN
    SELECT * FROM Abogados
END
GO
```

Actualizar abogados

-- Procedimiento almacenado para actualizar la información de un abogado

```
CREATE PROCEDURE sp_ActualizarAbogado
    @ID_Abogado int,
    @pNombre varchar(50),
    @sNombre nchar(10),
    @pApellido nchar(10),
    @sApellido nchar(10),
    @DUI int,
    @Especialidad varchar(100),
    @Telefono varchar(50),
    @Email varchar(100)
AS
BEGIN
    UPDATE Abogados
    SET pNombre = @pNombre,
        sNombre = @sNombre,
        pApellido = @pApellido,
        sApellido = @sApellido,
        DUI = @DUI,
        Especialidad = @Especialidad,
        Telefono = @Telefono,
        Email = @Email,
        Usuario_Modifica = ISNULL(@pNombre, Usuario_Modifica),
        Fecha_Modifica = ISNULL(GETDATE(), Fecha_Modifica)
    WHERE ID_Abogado = @ID_Abogado
END
GO
```

Eliminar Abogados por ID

-- Procedimiento almacenado para eliminar un abogado por su ID

```
CREATE PROCEDURE sp_EliminarAbogado
    @ID_Abogado int
AS
BEGIN
    DELETE FROM Abogados WHERE ID_Abogado = @ID_Abogado
END
GO
```

Creación de Clientes

-- Procedimiento almacenado para crear un nuevo cliente

```
CREATE PROCEDURE sp_CrearCliente
    @PrimNombre varchar(50),
    @SegNombre varchar(50),
    @PrimApellido varchar(50),
    @SegApellido varchar(50),
    @DUI int,
    @Client_Edad int,
    @Nacionalidad varchar(50),
    @Ocupacion varchar(50),
    @Direccion varchar(500),
    @Telefono varchar(50),
    @Email varchar(100)
AS
BEGIN
    INSERT INTO Clientes (PrimNombre, SegNombre, PrimApellido, SegApellido, DUI, Client_Edad, Nacionalidad, Ocupacion, Direccion, Telefono, Email)
    VALUES (@PrimNombre, @SegNombre, @PrimApellido, @SegApellido, @DUI, @Client_Edad, @Nacionalidad, @Ocupacion, @Direccion, @Telefono, @Email)
END
GO
```

Buscar Clientes

-- Procedimiento almacenado para leer todos los clientes

```
CREATE PROCEDURE sp_LeerClientes
```

```
AS
```

```
BEGIN
```

```
    SELECT * FROM Clientes
```

```
END
```

```
GO
```

-- Procedimiento almacenado para leer un cliente por su ID

```
CREATE PROCEDURE sp_LeerClientePorID
```

```
    @ID_Cliente int
```

```
AS
```

```
BEGIN
```

```
    SELECT * FROM Clientes WHERE ID_Cliente = @ID_Cliente
```

```
END
```

```
GO
```

Actualización de cliente

-- Procedimiento almacenado para actualizar la información de un cliente

```
CREATE PROCEDURE sp_ActualizarCliente
```

```
    @ID_Cliente int,  
    @PrimNombre varchar(50),  
    @SegNombre varchar(50),  
    @PrimApellido varchar(50),  
    @SegApellido varchar(50),  
    @DUI int,  
    @Client_Edad int,  
    @Nacionalidad varchar(50),  
    @Ocupacion varchar(50),  
    @Direccion varchar(500),  
    @Telefono varchar(50),  
    @Email varchar(100)
```

AS

```
BEGIN
```

```
    UPDATE Clientes
```

```
    SET PrimNombre = @PrimNombre,  
        SegNombre = @SegNombre,  
        PrimApellido = @PrimApellido,  
        SegApellido = @SegApellido,  
        DUI = @DUI,  
        Client_Edad = @Client_Edad,  
        Nacionalidad = @Nacionalidad,  
        Ocupacion = @Ocupacion,  
        Direccion = @Direccion,  
        Telefono = @Telefono,  
        Email = @Email
```

```
    WHERE ID_Cliente = @ID_Cliente
```

```
END
```

```
GO
```

Eliminación de Cliente

-- Procedimiento almacenado para eliminar un cliente por su ID

```
CREATE PROCEDURE sp_EliminarCliente
```

```
    @ID_Cliente int
```

AS

```
BEGIN
```

```
    DELETE FROM Clientes WHERE ID_Cliente = @ID_Cliente
```

```
END
```

```
GO
```


Crear Usuario

```
-- Procedimiento almacenado para crear un nuevo usuario
CREATE PROCEDURE sp_CrearUsuario
    @Nombre_Usuario varchar(50),
    @Contraseña varchar(50),
    @Usuario_Inserta varchar(50),
    @Fecha_Inserta datetime,
    @ID_Asistente int,
    @ID_Abogado int,
    @Rol varchar(50)
AS
BEGIN
    INSERT INTO Usuarios (Nombre_Usuario, Contraseña, Usuario_Inserta, Fecha_Inserta, Usuario_Modifica, Fecha_Modifica, ID_Asistente, ID_Abogado, Rol)
    VALUES (@Nombre_Usuario, @Contraseña, @Usuario_Inserta, @Fecha_Inserta, @Usuario_Inserta, @Fecha_Inserta, @ID_Asistente, @ID_Abogado, @Rol)
END
GO
```

Buscar Usuarios

```
-- Procedimiento almacenado para leer todos los usuarios
CREATE PROCEDURE sp_LeerUsuarios
AS
BEGIN
    SELECT * FROM Usuarios
END
GO

-- Procedimiento almacenado para leer un usuario por su ID
CREATE PROCEDURE sp_LeerUsuarioPorID
    @ID_Usuario int
AS
BEGIN
    SELECT * FROM Usuarios WHERE ID_Usuario = @ID_Usuario
END
GO
```

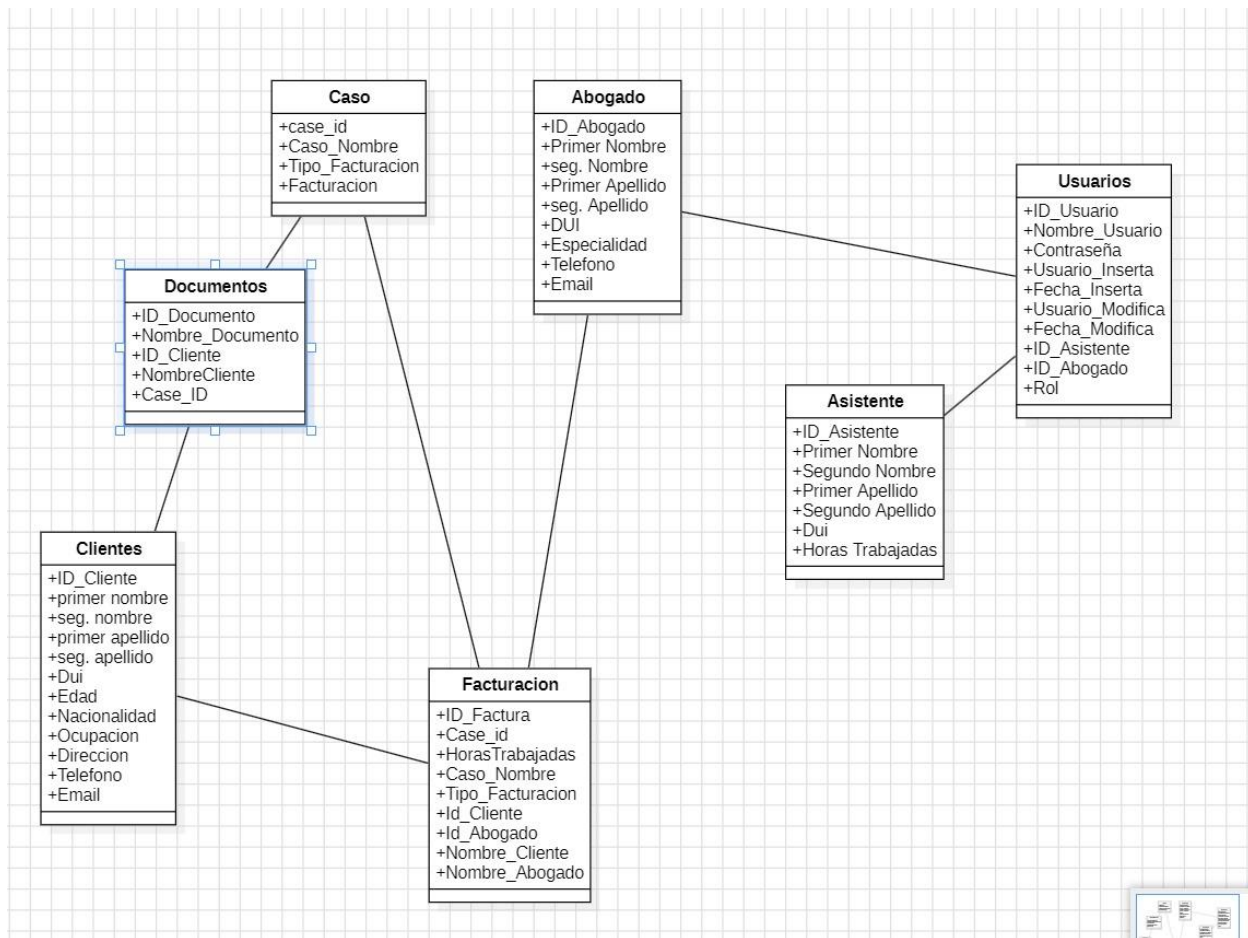
Actualizar Usuario

```
-- Procedimiento almacenado para actualizar la información de un usuario
CREATE PROCEDURE sp_ActualizarUsuario
    @ID_Usuario int,
    @Nombre_Usuario varchar(50),
    @Contrasena varchar(50),
    @Usuario_Modifica varchar(50),
    @Fecha_Modifica datetime,
    @ID_Asistente int,
    @ID_Abogado int,
    @Rol varchar(50)
AS
BEGIN
    UPDATE Usuarios
    SET Nombre_Usuario = @Nombre_Usuario,
        Contraseña = @Contraseña,
        Usuario_Modifica = @Usuario_Modifica,
        Fecha_Modifica = @Fecha_Modifica,
        ID_Asistente = @ID_Asistente,
        ID_Abogado = @ID_Abogado,
        Rol = @Rol
    WHERE ID_Usuario = @ID_Usuario
END
GO
```

Eliminar Usuario

```
-- Procedimiento almacenado para eliminar un usuario por su ID
CREATE PROCEDURE sp_EliminarUsuario
    @ID_Usuario int
AS
BEGIN
    DELETE FROM Usuarios WHERE ID_Usuario = @ID_Usuario
END
GO
```

Entidad Relación



Seguridad del sistema

La seguridad de nuestro sistema es fundamental para garantizar la confidencialidad, integridad y disponibilidad de la información. Para lo cual se ha realizado:

Claves de acceso

Las claves de acceso en este proyecto son una parte esencial de nuestros servicios, por lo que para ello utilizaremos una combinación de longitud y complejidad:

¿Qué significa esto?, significa que las contraseñas deben tener una longitud mínima de 6 caracteres y combinar letras mayúsculas, minúsculas, números y símbolos.

Roles de usuario

Los roles de los usuarios que gestionan nuestros servicios y accesos a nuestra página son esenciales. Por lo que definimos los roles y limitamos el acceso a la información según la necesidad del usuario.

- **Administrador:** Este ostenta el control total de la base, puede gestionar roles e información sensible.
- **Abogado:** Este tiene acceso a los casos asignados y a la información del cliente.
- **Asistente Legal:** Tiene acceso limitado y acceso limitado a la información de los clientes
- **Cliente:** Tiene acceso solamente a su propio expediente y la posibilidad de comunicarse con su abogado.

Privilegios

En este apartado nos referimos a cuanto acceso y control tiene un usuario sobre el sistema y de su relevancia con respecto a la información. Por tanto, se debe de tener ciertas medidas para mantener la salud de nuestro sistema.

- **Privilegios mínimos:** Los usuarios solo deben tener los permisos necesarios para realizar sus labores, reduciendo así las posibilidades de vulneración de cuentas.
- **Roles y acceso:** La implementación del modelo de roles nos da la facilidad de gestionar y auditar los permisos de los usuarios en cuanto a que información pueden tener a nuestra base.

Glosario de términos

Administrador:

Usuario con acceso completo al sistema que puede gestionar usuarios, roles y toda la información sensible.

Asistente Legal:

Usuario con acceso limitado a documentación y casos asignados, con permisos para ver y editar información bajo supervisión.

Base de Datos:

Sistema organizado que permite almacenar, manipular y recuperar información de manera eficiente.

Cliente:

Usuario con acceso restringido a su propio expediente y la capacidad de comunicarse con su abogado.

CRUD:

Acrónimo que representa las operaciones básicas de bases de datos: Crear, Leer, Actualizar y Borrar.

Entidad Relación:

Modelo utilizado para definir la estructura de una base de datos, mostrando las entidades y las relaciones entre ellas.

Facturas:

Tabla utilizada para llevar el conteo de horas trabajadas por cada caso y gestionar la facturación, incluyendo casos Pro-Bono y diferentes precios según el tipo de caso y abogado.

Procedimientos Almacenados:

Conjunto de comandos SQL guardados en la base de datos que pueden ser reutilizados y ejecutados cuando sea necesario, como crear, listar, actualizar y eliminar registros.

Privilegios:

Niveles de acceso y control que un usuario tiene sobre el sistema y la información que contiene.

Revisión y Auditoría de Privilegios:

Proceso periódico para revisar y auditar los permisos asignados a cada usuario y rol, asegurando que estén actualizados según las necesidades del sistema.

Tipos de Datos:

Definiciones para cada columna en una tabla y cada variable o parámetro, especificando el tipo de datos que se almacenarán, como números, textos y fechas.