

INSTITUTO TECNOLÓGICO DE COSTA RICA
ESCUELA DE INGENIERÍA EN COMPUTACIÓN

IC-8621 Diseño de Software

E4 Base de Datos

Profesor:

Mario Chacon Rivas

Elaborado por:

José Miguel González Barrantes 2023087564

Alejandro Gutiérrez Chaves 2023066266

Felipe Masís Calderón 2023047882

12 de septiembre del 2024

II semestre, 2024

Tabla de Contenidos

1. Introducción	3
2. Modelo de Datos	4
3. Estándares	18
3.1 Formas de normalización:	18
4. Diagramas	19
4.1 Diagrama de Contexto:	20
4.2 Diagrama de Ecosistemas:	20
4.3 Diagrama de Base de Datos:	21
5. Conclusiones	23
6. Referencias	23

1. Introducción

En el siguiente documento se presentan todos aquellos componentes fundamentales que permiten la salida y la entrada de datos del sistema a desarrollar. Este documento está conformado por varios apartados esenciales que detallan cada aspecto crítico del diseño y funcionamiento del sistema. El primer apartado, denominado “Modelo de datos”, se centra en explicar la utilidad y los aspectos generales de las entidades y sus respectivas relaciones. Este apartado es fundamental, ya que proporciona una visión detallada del contexto del proyecto desde una perspectiva técnica en el área de la computación. Se busca explicar estos conceptos de manera clara y accesible, incluso para aquellos que no están profundamente familiarizados con los aspectos técnicos de la computación. Este enfoque es fundamental, ya que estos documentos deben ser comprensibles para todos los involucrados en el desarrollo del software.

Después, se presenta el apartado de “Estándares”, donde se abordan las pautas, consideraciones y “reglas” bien conocidas en el desarrollo y presentación de modelos de bases de datos. Se tomó en cuenta un modelo relacional, en donde se tomaba en cuenta uno de los estándares más reconocidos; la normalización. En este documento, se han considerado las tres primeras formas de normalización, que son las más recomendadas debido a sus beneficios en la organización y eficiencia de los datos. Estas formas de normalización se explicarán en detalle más adelante. Además, se han implementado aspectos fundamentales como la optimización y la escalabilidad de las entidades, asegurando que el modelo de datos no solo sea eficiente, sino también adaptable a futuros cambios y expansiones.

El apartado de “Diagramas” presenta el modelo realizado del proyecto, incluyendo el diagrama que ilustra la estructura y las relaciones entre las entidades. Este diagrama no solo facilita la comprensión del modelo de datos, sino que también destacan los beneficios y elementos generales a considerar en el diseño de bases de datos. Se mencionan algunas consideraciones que se tomaron en cuenta para cumplir con los estándares que se mencionaron anteriormente.

Finalmente, en el apartado de “Conclusiones”, se presentan las ideas, mejoras y aspectos que se concretaron durante el desarrollo del modelo de base de datos. Este apartado resume las decisiones clave y las lecciones aprendidas,

proporcionando una visión general de cómo el equipo ha abordado los desafíos y obstáculos, así logrando un entendimiento del problema planteado. Estas conclusiones son fundamentales para asegurar que todos los miembros del equipo y otros stakeholders comprendan plenamente la lógica y la estructura del modelo de datos, facilitando así su implementación y mantenimiento.

2. Modelo de Datos

1. Tabla USUARIO

USUARIO				
Campo		Tipo de Dato	Descripción	
ID		INT	PK autoincremental	
Nombre		VARCHAR(50)	Nombre del usuario	
Apellido		VARCHAR(75)	Apellidos del usuario	
Correo		VARCHAR(100)	Correo electrónico, único	
Contraseña		VARCHAR(255)	Contraseña encriptada de la cuenta	
ID_rol		TINYINT(1) UNSIGNED	FK1 tipo de usuario	
Fecha_Registro		DATETIME	Fecha de registro, valor por defecto actual	
ID_escuela		TINYINT(1) UNSIGNED	FK2 Escuela a la que pertenece la persona	

Análisis del tamaño del registro y de la tabla

Esta será la tabla donde se registrarán los datos de las cuentas de las personas que usen el sistema, con los siguientes datos:

- ID: Debido a que se pueden registrar un gran número de cuentas, se decidió usar un tipo de dato int que sea incremental (4 bytes).
- Nombre: Los nombres no suelen ocupar muchos caracteres pero para algún caso especial se decidió usar un varchar de 50 (50 bytes).
- Apellido: De igual forma, para los apellidos se decidió un tamaño un poco más grande por casos especiales, además de considerar que debe ser más largo que el nombre, en este caso un varchar de 75 (75 bytes).

- Correo: Un correo electrónico típico no excede los 100 caracteres (dirección + dominio), por lo que se escogió un varchar de 100 (100 bytes).
- Contraseña: Se decidió tener contraseñas encriptadas por temas de seguridad, por lo que se escogió un varchar de 255 (255 bytes).
- ID_rol: Será una llave foránea para hacer la unión con el tipo de cuenta que es, por lo que no serán muchos registros y por esta razón se escogió un tinyint que abarca de 0 a 255 (1 byte).
- Fecha_Registro: Será la fecha de creación de la cuenta que se manejará con fecha y hora (8 bytes).
- ID_escuela: Será una llave foránea para hacer la unión con la escuela a la que pertenece el usuario (en caso de no tener contendrá *NULL*), por lo que no serán muchos registros y por esta razón se escogió un tinyint que abarca de 0 a 255 (1 byte).

De esta forma, se espera que el uso de estos tipos de datos optimice el espacio de almacenamiento de la tabla, permitiendo gestionar eficientemente un gran número de cuentas. A medida que el sistema se utilice y crezca, la estructura elegida facilitará su escalabilidad sin un impacto significativo en el rendimiento ni en el consumo de recursos.

2. Tabla ROL

ROL		
Campo	Tipo Dato	Descripción
ID	TINYINT(1) UNSIGNED	PK autoincremental
Nombre	VARCHAR(100)	Tipo de rol
Descripción	VARCHAR(255)	Descripción de cada rol

Análisis del tamaño del registro y de la tabla

Esta será la tabla donde se registrarán los distintos tipos de usuarios que manejará el sistema, con los siguientes datos :

- ID: Debido a que la tabla tendrá alrededor de 3 tipos de cuenta (extensionista, administrador y externo), se decidió usar tinyint que abarca de 0 a 255 (1 byte).
- Nombre: Los nombres de los roles no suelen abarcar muchos caracteres pero para algún caso especial se decidió usar un varchar de 50 (50 bytes).
- Descripción: será una descripción muy breve de qué es el tipo de cuenta/rol, por lo que se decidió usar un varchar de 255 (255 bytes).

Al ser una tabla que no va a recibir prácticamente cambios y tendrá muy pocos registros no ocupará mucho espacio por lo que no será algo significativo en cuanto a rendimiento, mas sí se pensaron los tipos de datos para ocupar lo menos posible de almacenamiento.

3. Tabla Escuela

ESCUELA					
Campo	Tipo Dato		Descripción		
ID	TINYINT(1) UNSIGNED		PK autoincremental		
Nombre	VARCHAR(100)		Nombre de la escuela		

Análisis del tamaño del registro y de la tabla

Esta será la tabla donde se registrarán las distintas escuelas a las que pertenecen los usuarios, con los siguientes datos :

- ID: Debido a que la tabla tendrá pocos registros, se decidió usar tinyint que abarca de 0 a 255 (1 byte) .
- Nombre: Los nombres de las escuelas no suelen ocupar muchos caracteres pero para algún caso especial se decidió usar un varchar de 100 (100 bytes).

Al ser una tabla que no va a recibir prácticamente cambios y tendrá muy pocos registros no ocupará mucho espacio por lo que no será algo significativo en cuanto a rendimiento, mas sí se pensaron los tipos de datos para ocupar lo menos posible de almacenamiento.

4. Tabla PROPUESTA

PROPUESTA		
Campo	Tipo Dato	Descripción
ID	INT	PK autoincremental
NombrePropuesta	VARCHAR(255)	Nombre de la propuesta
IdentificaciónAcuerdo	VARCHAR (50)	Identificación única de la propuesta
Descripción	VARCHAR(MAX)	Descripción de la propuesta
TipoExtensión	VARCHAR(100)	Tipo de extensión de la propuesta
ODS	VARCHAR(MAX)	Objetivo de desarrollo sostenible
DeclaraciónFinal	VARCHAR (MAX)	Declaración final por parte del responsable
FechaAprobación	DATETIME	Fecha que aprobó a nivel de escuela
fechaSolicitud	DATETIME	Fecha de solicitud de la propuesta
ID_Institucion	INT	FK1 Id. de empresa relacionada(s) con la propuesta
ID_UsuarioPromotor	INT	FK2 Id. de extensionista promotor
ID_Estado	INT	FK3 estado que se encuentra la propuesta

Análisis del tamaño del registro y de la tabla

Esta será la tabla donde se registrarán los datos de las propuestas realizadas y también para los proyectos (se diferenciarán por el tipo de estado), con los siguientes datos:

- ID: Debido a que se pueden registrar un gran número de propuestas, se decidió usar un tipo de dato int que sea incremental (4 bytes).

- NombrePropuesta: Los nombres de las propuestas pueden ser extensos por lo que se seleccionó un varchar de 255 (255 bytes).
- IdentificaciónAcuerdo: Al ser un identificador pequeño pero que dependiendo de la cantidad de acuerdos que haya según cantidad de propuestas, se decidió usar un varchar de 50 (50 bytes).
- Descripción: Al ser un texto muy amplio se decidió usar el máximo tamaño que un varchar puede almacenar (2 gb).
- TipoExtensión: Al ser solo un nombre pequeño se decidió usar un varchar de 100 (100 bytes).
- DeclaraciónFinal: Al ser un texto muy amplio se decidió usar el máximo tamaño que un varchar puede almacenar (2 gb).
- FechaAprobación: Se guardará la fecha en la que se aprobó la idea del proyecto por parte de una escuela para que se pudiera hacer una propuesta, por lo que se usará un datetime (8 bytes)
- fechaSolicitud: Se guardará la fecha en la que se hizo el envío de la propuesta, por lo que se usará un datetime (8 bytes)
- ID_UsuarioPromotor: Será una llave foránea para hacer la unión con el usuario que envió la propuesta, por lo que al igual que en la tabla usuario se decidió usar int (4 bytes).
- ID_Estado: Será una llave foránea para hacer la unión con el estado de la propuesta, por lo que no serán muchos registros y por esta razón se escogió un tinyint que abarca de 0 a 255 (1 byte).

De esta forma, se espera que el uso de estos tipos de datos optimice el espacio de almacenamiento de la tabla, permitiendo gestionar eficientemente un gran número de propuestas. A medida que el sistema se utilice y crezca, la estructura elegida facilitará su escalabilidad sin un impacto significativo en el rendimiento ni en el consumo de recursos, mas se sabe que por campos que usar varchar(max) el almacenamiento que ocupará la tabla será significativo, por dicha razón se trató de separar otro datos de las propuestas para disminuir el peso en cuanto a almacenamiento que representará esta tabla.

5. Tabla ESTADO_PROYECTO

ESTADO_PROYECTO		
Campo	Tipo Dato	Descripción
ID	TINYINT(1) UNSIGNED	PK autoincremental
Nombre	VARCHAR(50)	Estado de proyecto

Análisis del tamaño del registro y de la tabla

Esta será la tabla donde se registrarán los distintos tipos de estados que puede tener una propuesta, con los siguientes datos :

- ID: Debido a que la tabla tendrá alrededor de 4 tipos de cuenta (rechazada, enviada, en revisión, aprobada, se decidió usar tinyint que abarca de 0 a 255 (1 byte).
- Nombre: Los nombres de los estados no suelen abarcar muchos caracteres pero para algún caso especial se decidió usar un varchar de 50 (50 bytes).

Al ser una tabla que no va a recibir prácticamente cambios y tendrá muy pocos registros no ocupará mucho espacio por lo que no será algo significativo en cuanto a rendimiento, mas sí se pensaron los tipos de datos para ocupar lo menos posible de almacenamiento.

6. Tabla INSTITUCION_EXT

INSTITUCION_EXT		
Campo	Tipo Dato	Descripción
ID	INT	PK autoincremental
Institución	VARCHAR(100)	Nombre de la institución externa
Contacto	VARCHAR(255)	Formas de contactar a la institución
ID_propuesta	INT	FK id de la propuesta

Análisis del tamaño del registro y de la tabla

Esta será la tabla donde se registrarán los datos de las instituciones externas que participarán en cada propuesta, incluido la forma de contactarse, con los siguientes datos:

- ID: Debido a que se pueden registrar un gran número de propuestas, puede que haya un gran número de instituciones externas que participen, por lo que se decidió usar un tipo de dato int que sea incremental (4 bytes).
- Institución: Los nombres de instituciones no suelen ocupar muchos caracteres pero para algún caso especial se decidió usar un varchar de 100 (100 bytes).
- Contacto: Debido a que en el formulario se pedirá toda la información junta (Nombre de persona para hacer el contacto, correo y número), se decidió usar un varchar de 255 (255 bytes)
- ID_Propuesta: Será una llave foránea para hacer la unión con la propuesta en la que colaborará la institución, por lo que al igual que en la tabla propuesta se decidió usar int (4 bytes).

De esta forma, se espera que el uso de estos tipos de datos optimice el espacio de almacenamiento de la tabla, permitiendo gestionar eficientemente un gran número de miembros del equipo de la propuesta. A medida que el sistema se utilice y crezca, la estructura elegida facilitará su escalabilidad sin un impacto significativo en el rendimiento ni en el consumo de recursos. Aún así se sabe que puede haber instituciones repetidas, pero no con la misma propuesta o información de contacto, por eso la selección de datos.

7. Tabla GRUPO_EXTENSIONISTAS

GRUPO_EXTENSIONISTAS		
Campo	Tipo Dato	Descripción
ID	INT	PK autoincremental
ID_usuario	INT	FK1 Estado de propuesta
ID_propuesta	INT	FK2 Id. de la propuesta que pertenece el usuario
Condicion	VARCHAR2(50)	Condición otorgada hacia el extorsionista
Nombramiento	VARCHAR(50)	Nombramiento del usuario en el grupo
FechaInicio	VARCHAR(50)	Fecha inicio de operación
FechaFin	VARCHAR(50)	Fecha fin de operación

Análisis del tamaño del registro y de la tabla

Esta será la tabla donde se registrarán los datos de las personas que participarán en cada propuesta, incluido el promotor y también sus funciones dentro del proyecto, con los siguientes datos:

- ID: Debido a que se pueden registrar un gran número de propuestas, habrá un gran número de personas que participen, por lo que se decidió usar un tipo de dato int que sea incremental (4 bytes).
- ID_Usuario: Será una llave foránea para hacer la unión con el usuario que participa en la propuesta, por lo que al igual que en la tabla usuario se decidió usar int (4 bytes).

- ID_Propuesta: Será una llave foránea para hacer la unión con la propuesta en la que participa el usuario, por lo que al igual que en la tabla propuesta se decidió usar int (4 bytes).
- Condición: Al ser un texto corto se decidió usar un varchar de 50 (50 bytes).
- Nombramiento: Condición: Al ser un texto corto se decidió usar un varchar de 50 (50 bytes).
- FechaInicio: Se guardará la fecha en la que inició a trabajar en el proyecto el usuario, por lo que se usará un datetime (8 bytes)
- FechaFin: Se guardará la fecha en la que terminó de trabajar el usuario en el proyecto (en caso de no tener será *NULL*) , por lo que se usará un datetime (8 bytes)

De esta forma, se espera que el uso de estos tipos de datos optimice el espacio de almacenamiento de la tabla, permitiendo gestionar eficientemente un gran número de miembros del equipo de la propuesta. A medida que el sistema se utilice y crezca, la estructura elegida facilitará su escalabilidad sin un impacto significativo en el rendimiento ni en el consumo de recursos.

8. Tabla FINANCIAMIENTO

FINANCIAMIENTO		
Campo	Tipo Dato	Descripción
ID	INT	PK, autoincremental
monto_Aprobado	DECIMAL(10,3)	monto aprobado como financiamiento
fecha_Revision	DATETIME	Fecha en la que se revisó la propuesta
ID_propuesta	INT	FK, id de la propuesta

Análisis del tamaño del registro y de la tabla

Esta será la tabla donde se registrarán los datos del financiamiento otorgado a cada propuesta (solo para aquellas aprobadas), con los siguientes datos:

- ID: Debido a que se pueden registrar un gran número de propuestas, puede que haya muchos financiamientos aprobados, por lo que se decidió usar un tipo de dato int que sea incremental (4 bytes).
- monto_Aprobado: Será para guardar el monto asignado para la propuesta, se pensó en un decimal por si quiere tener mayor presión en caso de tener que hacer cambios de tipos de moneda o algo relacionado, usando 10 espacios para la parte entera y 3 para la decimal (9 bytes).
- FechaInicio: Se guardará la fecha en la que se aprobó el financiamiento para la propuesta, por lo que se usará un datetime (8 bytes)
- ID_Propuesta: Será una llave foránea para hacer la unión con la propuesta a la que pertenece el financiamiento, por lo que al igual que en la tabla propuesta se decidió usar int (4 bytes).

De esta forma, se espera que el uso de estos tipos de datos optimice el espacio de almacenamiento de la tabla, permitiendo gestionar eficientemente un gran número de miembros del equipo de la propuesta. A medida que el sistema se utilice y crezca, la estructura elegida facilitará su escalabilidad sin un impacto significativo en el rendimiento ni en el consumo de recursos.

9. Tabla CALIFICACION

CALIFICACION		
Campo	Tipo Dato	Descripción
ID	INT	PK, autoincremental
Nota	DECIMAL(3,2)	Nota de revisión de la propuesta
fecha_Revision	DATETIME	Fecha en la que se revisó la propuesta
ID_propuesta	INT	FK, id de la propuesta

Análisis del tamaño del registro y de la tabla

Esta será la tabla donde se registrarán los datos de la calificación otorgada a cada propuesta, con los siguientes datos:

- ID: Debido a que se pueden registrar un gran número de propuestas, puede que haya muchas calificaciones, por lo que se decidió usar un tipo de dato int que sea incremental (4 bytes).
- Nota: Será para guardar la nota asignada para la propuesta, se pensó en un decimal porque están van de 0 a 100 y con dos decimales (5 bytes).
- FechaInicio: Se guardará la fecha en la que se realizó la evaluación para la propuesta, por lo que se usará un datetime (8 bytes)
- ID_Propuesta: Será una llave foránea para hacer la unión con la propuesta a la que pertenece la calificación, por lo que al igual que en la tabla propuesta se decidió usar int (4 bytes).

De esta forma, se espera que el uso de estos tipos de datos optimice el espacio de almacenamiento de la tabla, permitiendo gestionar eficientemente un gran número de miembros del equipo de la propuesta. A medida que el sistema se utilice y crezca, la estructura elegida facilitará su escalabilidad sin un impacto significativo en el rendimiento ni en el consumo de recursos.

10. Tabla Beneficiario

BENEFICIARIO		
Campo	Tipo Dato	Descripción
ID	INT	PK autoincremental
Beneficiario	VARCHAR(100)	Nombre del beneficiario
ID_propuesta	INT	FK, id de la propuesta
Descripción	VARCHAR(255)	Descripción de como se beneficia

Análisis del tamaño del registro y de la tabla

Esta será la tabla donde se registrarán los datos de las personas que se van a ver beneficiadas por la propuesta realizadas y también para los proyectos (se diferenciarán por el tipo de estado), con los siguientes datos:

- ID: Debido a que se pueden registrar un gran número de propuestas, habrá muchos beneficiarios, por lo que se decidió usar un tipo de dato int que sea incremental (4 bytes).
- Beneficiario: Los nombres del grupo que se beneficiará no suele abarcar muchos caracteres pero para algún caso especial se decidió usar un varchar de 100 (100 bytes).
- ID_Propuesta: Será una llave foránea para hacer la unión con la propuesta relacionada con los beneficiarios, por lo que al igual que en la tabla propuesta se decidió usar int (4 bytes).
- Descripción: Al ser un texto no muy amplio con una pequeña explicación de cómo se beneficiará el grupo, se seleccionó un varchar de 255 (255 bytes).

De esta forma, se espera que el uso de estos tipos de datos optimice el espacio de almacenamiento de la tabla, permitiendo gestionar eficientemente un gran número de propuestas. A medida que el sistema se utilice y crezca, la estructura elegida facilitará su escalabilidad sin un impacto significativo en el rendimiento ni en el consumo de recursos.

11. Tabla BENEFICIO

BENEFICIO		
Campo	Tipo Dato	Descripción
ID	INT	PK autoincremental
ID_tipo	INT	ID tipo de beneficio
ID_propuesta	INT	FK id de la propuesta
Descripción	VARCHAR(255)	Descripción específica del beneficio

Análisis del tamaño del registro y de la tabla

Esta será la tabla donde se registrarán los datos de las beneficios específicos que brindará la propuesta, con los siguientes datos:

- ID: Debido a que se pueden registrar un gran número de propuestas, habrá muchos beneficiarios y por ende muchos beneficios, por lo que se decidió usar un tipo de dato int que sea incremental (4 bytes).
- ID_tipo: Será una llave foránea para hacer la unión con el tipo de beneficio que es, por lo que no serán muchos registros y por esta razón se escogió un tinyint que abarca de 0 a 255 (1 byte).
- ID_Propuesta: Será una llave foránea para hacer la unión con la propuesta relacionada con los beneficios, por lo que al igual que en la tabla propuesta se decidió usar int (4 bytes).
- Descripción: Al ser un texto no muy amplio con una pequeña explicación de en qué consiste el beneficio, se seleccionó un varchar de 255 (255 bytes).

De esta forma, se espera que el uso de estos tipos de datos optimice el espacio de almacenamiento de la tabla, permitiendo gestionar eficientemente un gran número de propuestas. A medida que el sistema se utilice y crezca, la estructura elegida facilitará su escalabilidad sin un impacto significativo en el rendimiento ni en el consumo de recursos.

12. Tabla TIPO_BENEFICIO

TIPO_BENEFICIO					
Campo		Tipo Dato		Descripción	
ID		TINYINT(1) UNSIGNED		PK autoincremental	
Nombre		VARCHAR(100)		Tipo de beneficio que otorga el proyecto	

Análisis del tamaño del registro y de la tabla

Esta será la tabla donde se registrarán los distintos tipos de beneficios que manejará el sistema, con los siguientes datos :

- ID: Debido a que la tabla tendrá alrededor de 4 tipos (se podrían agregar más) de beneficio (Económicos, Institucionales, Sociales, Académicos), se decidió usar tinyint que abarca de 0 a 255 (1 byte).
- Nombre: Los nombres de los tipos de beneficio no suelen abarcar muchos caracteres pero para algún caso especial se decidió usar un varchar de 50 (50 bytes).

Al ser una tabla que no va a recibir prácticamente cambios y tendrá muy pocos registros no ocupará mucho espacio por lo que no será algo significativo en cuanto a rendimiento, mas sí se pensaron los tipos de datos para ocupar lo menos posible de almacenamiento.

Relaciones generales

- Una escuela puede tener muchos usuarios pero un usuario solo una escuela.
- Un rol puede tener muchos usuarios pero un usuario solo un rol.
- Un usuario puede estar en varios grupos de extensionistas y un grupo de extensionistas puede tener varios usuarios.
- Una propuesta tiene un financiamiento, y ese financiamiento es solo de esa propuesta.
- Un usuario puede hacer varias propuestas pero una propuesta solo tiene un usuario.
- Una institución externa puede participar en varias propuestas y una propuesta puede tener varias instituciones externas.
- Un estado de proyecto puede tener muchos proyectos (muchos proyectos pueden tener el mismo estado), pero un proyecto solo puede tener un estado de proyecto.
- Una calificación solo tiene una propuesta y una propuesta solo tiene una calificación.
- Una propuesta puede tener muchos beneficios, y esos beneficios pueden estar en varias propuestas.

- Un tipo de beneficio puede tener muchos sub-beneficios, pero un sub-beneficio pertenece a un solo tipo de beneficio.
- Un grupo de beneficiarios puede ser beneficiado por varias propuestas, y una propuesta puede tener varios beneficiarios.

3. Estándares

Para el desarrollo del modelo de base de datos, se realizó uno de tipo “relacional” en el cual se toma en cuenta la idea de que existen entidades las cuales serán vistas como “tablas”; estas pueden relacionarse con otras entidades mediante la implementación de llaves foráneas. Adicionalmente se tomaron en cuenta las tres formas de normalización, escalabilidad y optimización; estas serán explicadas a continuación.

3.1 Formas de normalización:

- Primera Forma Normal (1FN): Es un requisito en donde se establece que cada columna debe ser un valor atómico. Esto significa que son indivisibles y que un dato no depende de varios datos o significados. Además cada registro debe ser único para mantener la integridad de los datos. Ej: Si existe un campo de periodo de participación de tiempo, dividirlo en dos columnas; una para fecha de inicio y otra para fecha final.
- Segunda Forma Normal (2FN): Debe cumplir con la “Primera Forma Normal” y además los atributos que no sean llaves primarias no deben depender en su totalidad de la llave primaria. Ej: Separar una tabla transaccional en dos; una de “encabezado” y otra de “detalles”.
- Tercera Forma Normal (3FN): Debe cumplir con la “Segunda Forma Normal”, además aquellos atributos que no sean llaves primarias deben de ser independientes entre sí. En pocas palabras, las columnas no pueden depender de otras. Ej: Utilización de tablas separadas para evitar dependencias entre sí.

Como cualquier tipo de estándar, existen ventajas y desventajas al momento de implementar estas formas normales, en este caso se nombrarán algunas que se identificaron:

Ventajas:

- Evitar repeticiones: Al dividir mucho los datos, se logra reducir mucho la duplicación de la información.
- Garantiza la integridad de los datos: A partir de tantas medidas, se reduce muchísimo la probabilidad de que se atente con la integridad de los datos.
- Altos niveles de mantenibilidad: Puesto a que se garantiza la integridad de los datos, esto influye positivamente a que el ingreso y retiro de información sea óptimo.

Desventajas:

- Puede retardar el rendimiento: Al momento de aplicar las tres formas de normalización, casi siempre se termina teniendo una gran cantidad de tablas; esto se traduce en más carga para el motor de base de datos.
- Complejidad y facilidad para equivocarse: El aplicar correctamente lo mencionado anteriormente es difícil y suele haber muchas retroalimentaciones y revisiones.

Para los demás atributos de calidad como la optimización y escalabilidad, estos también fueron tomados en cuenta al momento de desarrollar el diagrama de bases de datos, especialmente al momento de crear las entidades y sus respectivos atributos.

4. Diagramas

En el caso del sistema a implementar se decidió realizar tres diferentes tipos de diagramas; diagrama de contexto, diagrama de ecosistemas y diagrama de base de datos. Cada diagrama tiene un propósito claro el cual será explicado a continuación:

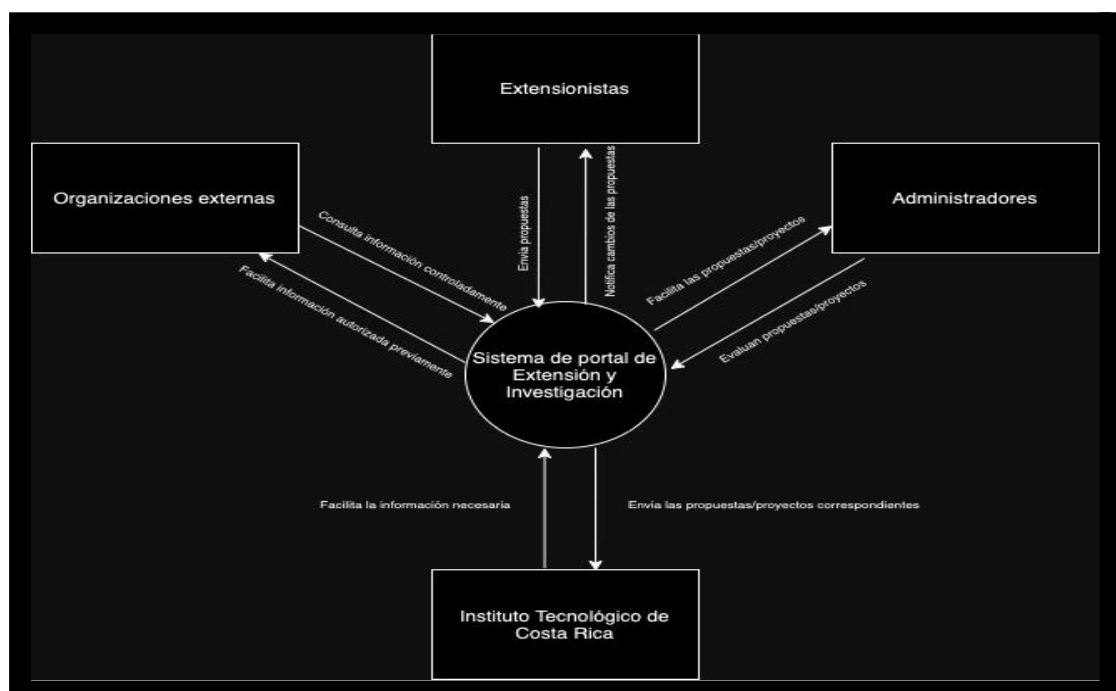
4.1 Diagrama de Contexto:

El diagrama de contexto es un modelo en donde se presentan aquellas entes externas las cuales influyen al sistema en sí, en este caso se puede identificar las organizaciones externas, las cuales serían empresas o organizaciones que no pertenecen al Tecnológico de Costa Rica. Los extensionistas vendrían siendo aquellos investigadores, estudiantes y profesores que pueden subir y pertenecer a diferentes propuestas. Por otro lado los administradores serían los evaluadores, el comité, entre otros; estos pueden evaluar propuestas, dar presupuestos, asignar propuestas a evaluadores, etc.

Este diagrama es realmente útil puesto a que permite que los stakeholders y desarrolladores tengan las siguientes facilidades:

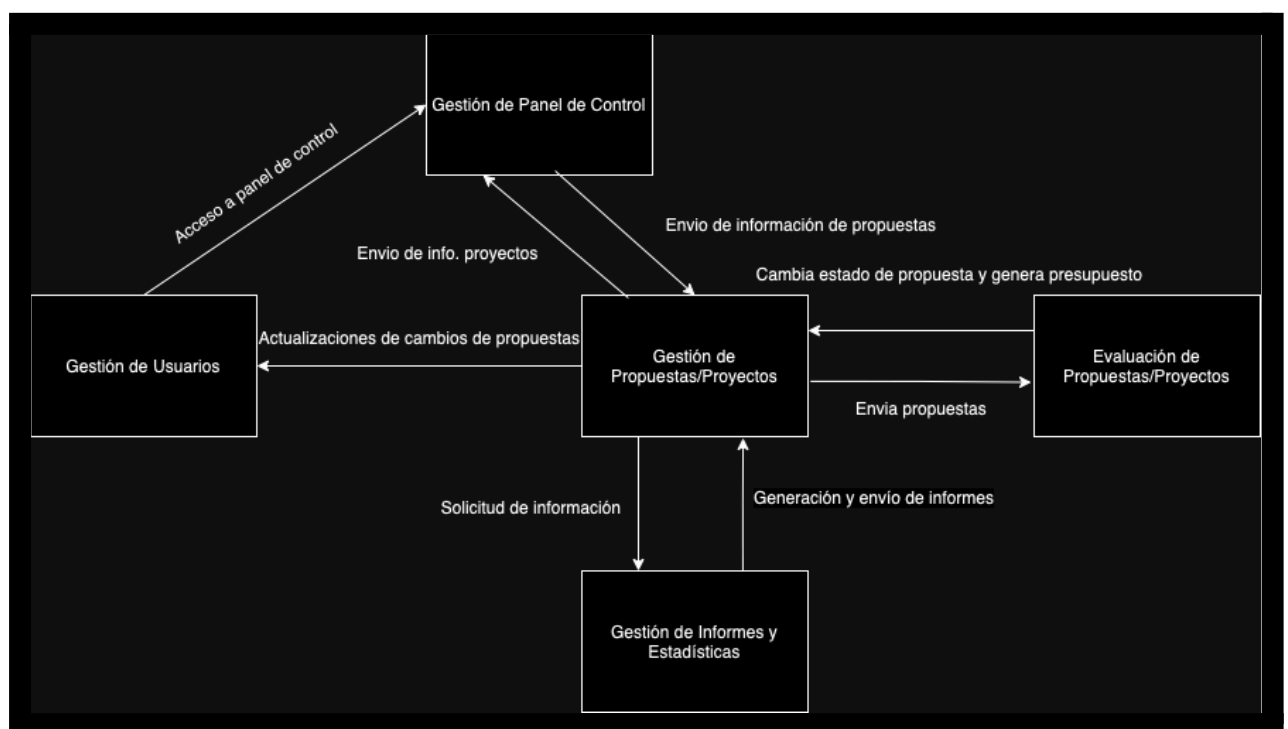
- Visión General: Muestra cómo el sistema interactúa con usuarios y otros sistemas, ayudando a entender el alcance del proyecto.
- Identificación de Interacciones: Ayuda a ver cómo se comunican las diferentes partes del sistema con el exterior.
- Comunicación Clara: Facilita que todos los involucrados entiendan el sistema, incluso si no son técnicos.
- Detección de Errores: Permite encontrar y corregir errores en los requisitos del proyecto desde el principio.

4.2 Diagrama de Ecosistemas:



En el caso del diagrama de ecosistemas, este permite representar de manera más clara y menos técnica aquellos elementos internos que participan en el sistema. Además se identifican aquellos módulos o gestores que son fundamentales para el funcionamiento de este mismo; tanto así que sin estos el sistema no funcionará correctamente. Este diagrama permite también entender de manera resumida el cómo y de dónde vienen las entradas y salidas, además del procesamiento de las mismas.

Para los stakeholders y los desarrolladores del software este diagrama les permite ventajas como la visualización clara y sencilla, una facilidad para identificar las dependencias entre sí, la escalabilidad relacionada con la mantenibilidad, y la optimización de recursos para así tener una planificación del tamaño del software que se está desarrollando.



4.3 Diagrama de Base de Datos:

En este diagrama, se detallan tanto las tablas que componen la base de datos del sistema, como la relación entre estas tablas. Este diagrama es fundamental para visualizar cómo la información será estructurada, almacenada y

Otros beneficios del diagrama de la base de datos es la integridad de los datos, ya que ayuda a detectar errores en el diseño de las tablas y en sus relaciones, también es bastante útil para optimizar el diseño de la base de datos, ya que da mejor visión de cómo interactúan y permite ver si se puede optimizar más

A diferencia de los otros diagramas, se va a incluir un link además de la imagen, ya que al ser un diagrama más grande, se dificulta entenderlo a partir de una imagen.



Se recomienda visitar el siguiente link que lleva al diagrama de la base de datos, ya que en la imagen no se pueden apreciar muy bien las tablas y sus relaciones,

- https://viewer.diagrams.net/?tags=%7B%7D&lightbox=1&highlight=0000ff&edit=_blank&layers=1&nav=1&title=Diagrama%20de%20Bases%20de%20Datos#Uhttps%3A%2F%2Fdrive.google.com%2Fuc%3Fid%3D1HkCixD-0f07bkxdRKnxJIZILSw9hry5Q%26export%3Ddownload

5. Conclusiones

El desarrollo del modelo de la base de datos permitió aprender aspectos clave para la implementación del proyecto, resolviendo varias dudas en el proceso. Uno de los mayores desafíos fue convertir el formulario y toda la información relacionada en tablas que siguieran los estándares y tuvieran coherencia. Esto requirió un análisis cuidadoso de los datos, ya que el formulario era extenso y estaba en formato escrito.

Otro aspecto clave y complicado fue establecer las relaciones entre las tablas. Aunque fue un proceso complicado, valió la pena, ya que ayuda a tener un mejor entendimiento del funcionamiento de los datos dentro del proyecto, además, permitió desarrollar una estructura más lógica, dicha estructura va a poder adaptarse a futuros cambios (si es necesario) y será más fácil de implementar.

Para finalizar, se puede concluir que este proyecto nos permitió enfrentar desafíos importantes en cuanto al diseño de la base de datos, pero que sin duda alguna estos desafíos fueron algo bueno ya que nos permitieron mejorar nuestro entendimiento del proyecto, para así poder realizar una mejor implementación de este.

6. Referencias

Araneda, P. (2022, 18 octubre). *Capítulo 6 Normalización | Base de datos*.

<https://bookdown.org/paranedagarcia/database/normalizacion.html>