Imagen que contiene Diagrama

Descripción generada automáticamente

ESTACION CLIMATOLOGICA

PROYECTO FERIA CIENTIFICA JUNIO 2024

ANALISIS Y DISENO BASE DE DATOS

ROLANDO ROQUE

LESLIE URBINA

JORGE RIVERA

FERNADO CHAVEZ

ENRIQUE COLINDRES

Contenido

[Introducción 4](#_Toc169990225)

[Planteamiento del Escenario 4](#_Toc169990226)

[Presupuesto y Tiempo de Implementación 4](#_Toc169990227)

[Detalle del Cronograma 5](#_Toc169990228)

[Descripción del Problema 6](#_Toc169990229)

[Justificación del Problema 6](#_Toc169990230)

[Objetivo General 7](#_Toc169990231)

[Objetivos Específicos 7](#_Toc169990232)

[Diseño Conceptual 8](#_Toc169990233)

[Relaciones Existentes 9](#_Toc169990234)

[Diagrama Entidad relación 9](#_Toc169990235)

[Restricciones 10](#_Toc169990236)

[Cardinalidad 10](#_Toc169990237)

[Diseño Lógico 10](#_Toc169990238)

[Esquema Entidad Relación 10](#_Toc169990239)

[Afinación del Modelo Relacional a través de la Normalización 11](#_Toc169990240)

[Restricciones de Integridad Referencial 11](#_Toc169990241)

[Diccionario de Datos – Diseño Lógico 12](#_Toc169990242)

[Descripción de las Tablas del Modelo Lógico 13](#_Toc169990243)

[Modelo Físico 13](#_Toc169990244)

[Restricciones de Dominio 13](#_Toc169990245)

[Sentencias SQL para Implementar la Base de Datos 14](#_Toc169990246)

[Referencias Bibliográficas 14](#_Toc169990247)

# Introducción

La gestión eficiente de datos es crucial para cualquier sistema de información. Este documento propone la implementación de una base de datos para automatizar la gestión de una estación meteorológica, facilitando la recopilación, almacenamiento y análisis de datos meteorológicos, que son esenciales para la predicción del clima y su registro historico

La base de datos webAppClima se diseñará siguiendo las mejores prácticas en el diseño de bases de datos, garantizando la integridad, consistencia y accesibilidad de los datos. A lo largo de este documento, se describirá en detalle el proceso de diseño e implementación de la base de datos, desde el análisis de requisitos hasta la ejecución de de las sentencias de implementacion.

# Planteamiento del Escenario

Actualmente, la estación meteorológica para registrar sus lecturas de temperatura, presión y altitud es una idea planteado como proyecto de la feria científica Unitec 2024.

# Presupuesto y Tiempo de Implementación

**Presupuesto de Gastos**

1. **Tiempo de Implementación:** 4 semanas
   * **Análisis de Requisitos:** $300
   * **Diseño de la Base de Datos:**
     + Diseño conceptual: $600
     + Diseño lógico: $600
     + Diseño físico: $600
   * **Validación y Pruebas:**
     + Pruebas de integridad de datos: $400
   * **Documentación:**
     + Elaboración de diagramas y documentación técnica: $300
     + Documentación de usuario: $300

**Total:** $3100.00

# Detalle del Cronograma

1. **Semana 1: Análisis de Requisitos**
   * Reuniones con los stakeholders
   * Recopilación de datos y definición de requisitos
2. **Semana 2: Diseño Conceptual**
   * Identificación de entidades y atributos
   * Elaboración del Modelo Entidad-Relación (ERD)
3. **Semana 3: Diseño Lógico y Físico**
   * Transformación del ERD a un modelo lógico
   * Normalización de las tablas
   * Definición de restricciones y claves
   * Creación de sentencias SQL para la implementación
4. **Semana 4: Validación, Pruebas y Documentación**
   * Ejecución de pruebas de integridad de datos
   * Revisión y validación con los stakeholders
   * Elaboración de documentación técnica y de usuario

# Descripción del Problema

El método actual de registro de datos meteorológicos en hojas de cálculo es ineficiente y propenso a errores. Las hojas de cálculo son susceptibles a errores de entrada, duplicación de datos y falta de integridad referencial. Además, carecen de la capacidad para realizar análisis de datos en profundidad, lo que limita la generación de informes precisos y útiles para la toma de decisiones.

Los problemas específicos incluyen:

* **Errores de Entrada:** La entrada manual de datos aumenta la probabilidad de errores tipográficos y de formato.
* **Duplicación de Datos:** La falta de restricciones y controles adecuados puede llevar a la duplicación de datos.
* **Falta de Integridad Referencial:** Las hojas de cálculo no pueden garantizar la integridad referencial entre diferentes conjuntos de datos.
* **Dificultad en el Análisis de Datos:** Las hojas de cálculo no están diseñadas para manejar grandes volúmenes de datos y realizar análisis complejos.

# Justificación del Problema

Centralizar y organizar la información de las lecturas meteorológicas en una base de datos permitirá una gestión más eficiente y precisa de los datos. La automatización del registro de datos meteorológicos ayuda a garantizar la consistencia y precisión de los datos, eliminando errores humanos.

Las ventajas de implementar una base de datos incluyen:

* **Mejora en la Precisión de los Datos:** La automatización reduce los errores humanos y garantiza la consistencia de los datos.
* **Facilidad de Acceso y Análisis:** Los datos centralizados permiten un acceso más rápido y eficiente, así como la capacidad de realizar análisis complejos.
* **Generación de Informes:** La base de datos facilita la generación de informes precisos y útiles.

# Objetivo General

Implementar una base de datos para automatizar el proceso de registro y gestión de lecturas meteorológicas de una estación meteorológica, mejorando la eficiencia y precisión en la gestión de la información.

# Objetivos Específicos

Para alcanzar el objetivo general, se establecen los siguientes objetivos específicos:

* **Análisis de Requisitos:** Identificar los requisitos para la automatización del proceso de registro y gestión de lecturas meteorológicas.
* **Diseño Conceptual:** Realizar el diseño conceptual de la base de datos, identificando entidades y atributos, y elaborando el Modelo Entidad-Relación.
* **Diseño Lógico:** Diseñar la base de datos, normalizando las entidades, describiendo campos, atributos, claves primarias y foráneas.
* **Diseño Físico:** Definir el motor de base de datos a ser utilizado, en este caso MySQL, y crear las sentencias SQL necesarias.
* **Validación y Pruebas:** Realizar pruebas de integridad de datos para asegurar la calidad de la base de datos.
* **Documentación:** Elaborar la documentación técnica y de usuario para garantizar el correcto uso y mantenimiento de la base de datos.

# Diseño Conceptual

El diseño conceptual de la base de datos se desarrolló para satisfacer las necesidades de almacenamiento y gestión de información de manera eficiente y estructurada. Se identificaron las entidades principales y sus atributos clave, así como las relaciones entre ellas. Además, se utilizaron tablas catálogo para representar datos estáticos y controlar los valores predefinidos en ciertos atributos.

**Entidades y Atributos**

Las entidades y atributos identificados son:

* **Usuario:**
  + usuario\_id: Identificador único del usuario.
  + user: Nombre de usuario.
  + passwd: Contraseña del usuario.
* **Lecturas:**
  + lectura\_id: Identificador único de la lectura.
  + fecha\_hora: Fecha y hora de la lectura.
  + temperatura: Valor de la temperatura registrada.
  + presion: Valor de la presión atmosférica registrada.
  + altitud: Valor de la altitud registrada.
* **Logs:**
  + log\_id: Identificador único del log.
  + usuario\_id: Identificador del usuario que realizó la actividad.
  + actividad\_id: Identificador de la actividad realizada.
  + fecha\_hora: Fecha y hora de la actividad registrada.

# Relaciones Existentes

Las relaciones existentes entre las entidades son:

* **Usuario y Logs:** Existe una relación de uno a muchos entre la entidad Usuario y la entidad Logs. Un usuario puede tener múltiples logs asociados, pero cada log está relacionado con un solo usuario. Es una relación uno a muchos (1

).

# Diagrama Entidad relación

Diagrama

Descripción generada automáticamente

# Restricciones

Las restricciones definidas para garantizar la integridad y consistencia de los datos son:

* **Claves Primarias:**
  + tblUsuarios: usuario\_id
  + tblLecturas: lectura\_id
  + tblLogs: log\_id
* **Claves Foráneas:**
  + tblLogs: usuario\_id (FK a usuario\_id en tblUsuarios)

# Cardinalidad

Las relaciones de cardinalidad definidas son:

* **Usuario y Logs:** Relación uno a muchos (1

). Un usuario puede tener varios logs, pero un log pertenece a un solo usuario.

# Diseño Lógico

El diseño lógico de la base de datos utilizó técnicas de normalización para eliminar redundancias y garantizar la integridad de los datos. Se identificaron entidades y atributos, se definieron relaciones entre entidades a través de claves primarias y foráneas, y se procedió a transformar las entidades y atributos en modelos de tabla y campos, tipos de datos, restricciones y relaciones.

# Esquema Entidad Relación

El Esquema Entidad Relación muestra las relaciones entre las entidades tblUsuarios, tblLecturas, y tblLogs, así como sus atributos.

# 

# Afinación del Modelo Relacional a través de la Normalización

1. **Tabla tblUsuarios:**
   * Se aplicó la primera forma normal (1FN) para garantizar la simplicidad de los atributos y evitar la repetición de datos.
   * Se aplicó la segunda forma normal (2FN) para eliminar las dependencias parciales, identificando y separando los atributos que dependían solo de una parte de la clave primaria.
2. **Tabla tblLecturas:**
   * Se aplicó la primera forma normal (1FN) para asegurar la simplicidad de los atributos y evitar la repetición de datos.
   * Se aplicó la segunda forma normal (2FN) para eliminar las dependencias parciales, identificando y separando los atributos que dependían solo de una parte de la clave primaria.
3. **Tabla tblLogs:**
   * Se aplicó la primera forma normal (1FN) para garantizar la simplicidad de los atributos y evitar la repetición de datos.
   * Se aplicó la segunda forma normal (2FN) para eliminar las dependencias parciales, identificando y separando los atributos que dependían solo de una parte de la clave primaria.

# Restricciones de Integridad Referencial

Las restricciones de integridad referencial aseguran que las relaciones entre tablas se mantengan consistentes.

* **Tabla tblLogs:** El campo usuario\_id es FK con referencia a tblUsuarios. Solo se pueden insertar valores válidos para usuario\_id.

# Diccionario de Datos – Diseño Lógico

El diccionario de datos proporciona una descripción detallada de los campos en cada tabla.

* **tblUsuarios:**
  + usuario\_id (PK, INT, NOT NULL, AUTO\_INCREMENT): Identificador único del usuario.
  + user (VARCHAR(8), NOT NULL): Nombre de usuario.
  + passwd (VARCHAR(12), NOT NULL): Contraseña del usuario.
* **tblLecturas:**
  + lectura\_id (PK, INT, NOT NULL, AUTO\_INCREMENT): Identificador único de la lectura.
  + fecha\_hora (DATETIME, NOT NULL): Fecha y hora de la lectura.
  + temperatura (FLOAT): Valor de la temperatura registrada.
  + presion (FLOAT): Valor de la presión atmosférica registrada.
  + altitud (FLOAT): Valor de la altitud registrada.
* **tblLogs:**
  + log\_id (PK, INT, NOT NULL, AUTO\_INCREMENT): Identificador único del log.
  + usuario\_id (FK, INT, NOT NULL, REFERENCES tblUsuarios(usuario\_id)): Identificador del usuario que realizó la actividad.
  + actividad\_id (INT): Identificador de la actividad realizada.
  + fecha\_hora (DATETIME, NOT NULL): Fecha y hora de la actividad registrada.

# Descripción de las Tablas del Modelo Lógico

* **tblUsuarios:** Almacena la información de los usuarios de la estación meteorológica.
* **tblLecturas:** Contiene los registros de las lecturas meteorológicas.
* **tblLogs:** Registra las actividades de los usuarios.

# Modelo Físico

El modelo físico detalla la implementación real de la base de datos, incluyendo los tipos de campos y las restricciones de dominio.

**Tipos de Campos**

* **INT:** Utilizado para campos que almacenan enteros.
* **VARCHAR:** Utilizado para campos que almacenan cadenas de texto.
* **DATETIME:** Utilizado para campos que almacenan fecha y hora.
* **FLOAT:** Utilizado para campos que almacenan números decimales.
* **AUTO\_INCREMENT:** Utilizado para campos que incrementan automáticamente su valor.
* **PRIMARY KEY:** Define la clave primaria de una tabla.
* **FOREIGN KEY:** Define una clave foránea que referencia una clave primaria en otra tabla.

# Restricciones de Dominio

Las restricciones de dominio aseguran que los valores ingresados en los campos cumplan con los tipos de datos y restricciones definidas.

* **usuario\_id:** Debe ser un número entero y único.
* **user:** Debe ser una cadena de texto de hasta 8 caracteres.
* **passwd:** Debe ser una cadena de texto de hasta 12 caracteres.
* **lectura\_id:** Debe ser un número entero y único.
* **fecha\_hora:** Debe ser una fecha y hora válidas.
* **temperatura:** Debe ser un número decimal.
* **presion:** Debe ser un número decimal.
* **altitud:** Debe ser un número decimal.
* **log\_id:** Debe ser un número entero y único.
* **usuario\_id:** Debe ser un número entero que exista en la tabla tblUsuarios.
* **actividad\_id:** Debe ser un número entero.
* **fecha\_hora:** Debe ser una fecha y hora válidas.

# Sentencias SQL para Implementar la Base de Datos

create database webappclima;

use webappclima;

create table tblusuarios (usuario\_id int auto\_increment primary key, user varchar(8) not null,passwd varchar(12) not null);

create table tbllecturas (lectura\_id int auto\_increment primary key, usuario\_id int not null,fecha\_hora datetime not null,temperatura float, presion float, altitud float, foreign key (usuario\_id) references tblusuarios(usuario\_id));

create table tbllogs (log\_id int auto\_increment primary key, usuario\_id int not null, actividad\_id int, fecha\_hora datetime not null,

foreign key (usuario\_id) references tblusuarios(usuario\_id)

# Referencias Bibliográficas

* Fundamentos de bases de datos, Abraham Silberschatz, Henry F. Korth y S. Sudarshan (2006).
* Bases de Datos, C.J. Date (2003).