Diseño de Base de Datos - TaskManager

**Fecha:** 05/09/2025

**Autor:Erick Aguila Barrientos, Pool Apaza Aguirre, Victor Condori Mamani, Crystian Garcia Huallpa**

**Versión:** 1.0

# 1. Introducción

El presente proyecto tiene como propósito el diseño y construcción de un prototipo de invernadero automatizado para la comunidad de Chihuaco, ubicada en Sicuani, Perú. El invernadero será gestionado a través de la plataforma Arduino, que permitirá el control eficiente de recursos clave como la temperatura, humedad y luz, utilizando tecnologías accesibles y sostenibles. Este sistema automatizado facilitará la labor de los agricultores, quienes podrán monitorear y controlar las condiciones ambientales de manera remota, aumentando la productividad de los cultivos y reduciendo los costos operativos.

Este documento tiene como objetivo presentar el diseño de la base de datos necesaria para gestionar la información generada por los sensores y sistemas automatizados del invernadero. La base de datos será fundamental para almacenar las lecturas de los sensores, los ajustes realizados, los eventos generados y la información de los usuarios que operan el sistema.

# 2. Diagrama Entidad-Relación (ERD)

## 2.1. Entidades y sus Atributos

**1. Usuarios**

* id\_usuario (PK): Identificador único del usuario.
* nombre: Nombre completo del usuario.
* email: Correo electrónico del usuario (único).
* fecha\_registro: Fecha de registro del usuario.

**2. Sensores**

* id\_sensor (PK): Identificador único del sensor.
* tipo\_sensor: Tipo de sensor (ejemplo: temperatura, humedad, luz).
* fecha\_instalacion: Fecha de instalación del sensor.

**3. Lecturas**

* id\_lectura (PK): Identificador único de la lectura.
* id\_sensor (FK, referencia a Sensores): Identificador del sensor que generó la lectura.
* valor: Valor de la lectura registrada por el sensor (ejemplo: 25°C, 70% de humedad).
* fecha\_lectura: Fecha y hora en que se registró la lectura.

**4. Controles**

* id\_control (PK): Identificador único del control.
* id\_sensor (FK, referencia a Sensores): Identificador del sensor relacionado con el ajuste.
* valor\_ajuste: Valor ajustado por el usuario (por ejemplo, temperatura deseada).
* fecha\_ajuste: Fecha en que se realizó el ajuste.

**5. Eventos**

* id\_evento (PK): Identificador único del evento.
* descripcion: Descripción del evento (ejemplo: error en el sensor, ajuste manual realizado).
* fecha\_evento: Fecha en que ocurrió el evento.

## 2.2. Relaciones

* **Usuarios y Sensores**: Un usuario puede gestionar múltiples sensores, pero cada sensor tiene un único propietario.
* **Sensores y Lecturas**: Un sensor puede generar múltiples lecturas, y cada lectura está asociada a un único sensor.
* **Sensores y Controles**: Un sensor puede tener múltiples ajustes, y cada ajuste corresponde a un único sensor.
* **Usuarios y Eventos**: Un usuario puede registrar varios eventos, pero cada evento está asociado a un único usuario..

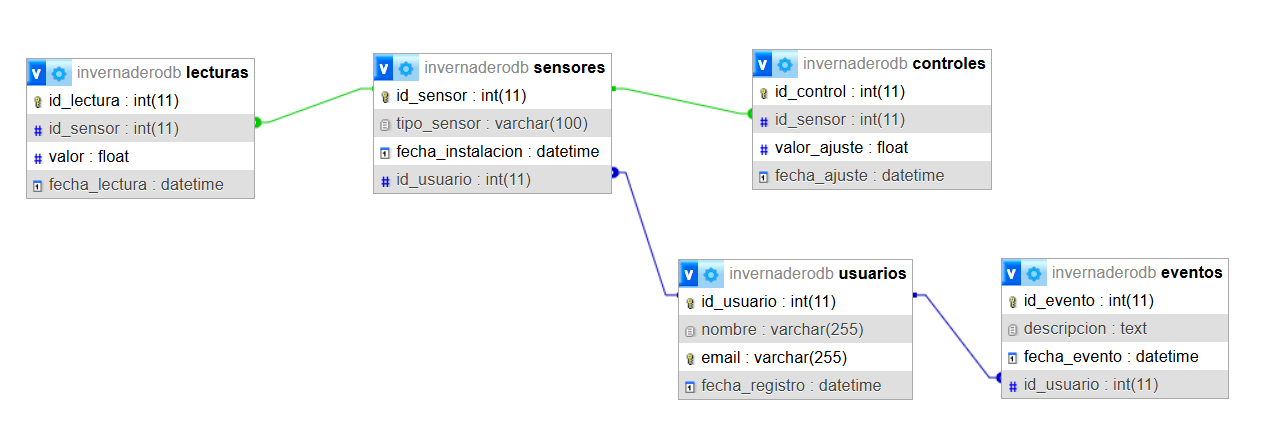
## 2.3. Reglas y Restricciones

* El email de un usuario debe ser único.
* Un sensor debe estar vinculado a un solo usuario.
* Las lecturas y los controles deben estar asociados a un único sensor.
* Los eventos deben tener una descripción detallada y deben estar relacionados a un usuario que los haya registrado.

## 2.4. Diagrama de Base de Datos (E/R)

* **Usuarios (1) → (N) Sensores**
* **Sensores (1) → (N) Lecturas**
* **Sensores (1) → (N) Controles**
* **Usuarios (1) → (N) Eventos**

# 3. Diagrama Relacional



# 4. Diccionario de Datos

**Tabla Usuarios**

| Campo | Tipo de Dato | Descripción | Restricciones |
| --- | --- | --- | --- |
| id\_usuario | INT | Identificador único del usuario. | PRIMARY KEY, AUTO\_INCREMENT |
| nombre | VARCHAR(255) | Nombre completo del usuario. | NOT NULL |
| email | VARCHAR(255) | Dirección de correo electrónico del usuario. | UNIQUE, NOT NULL |
| fecha\_registro | DATETIME | Fecha y hora en que el usuario se registró. | NOT NULL |

**Tabla Sensores**

| Campo | Tipo de Dato | Descripción | Restricciones |
| --- | --- | --- | --- |
| id\_sensor | INT | Identificador único del sensor. | PRIMARY KEY, AUTO\_INCREMENT |
| tipo\_sensor | VARCHAR(255) | Tipo de sensor (temperatura, humedad, etc.) | NOT NULL |
| fecha\_instalacion | DATETIME | Fecha de instalación del sensor | NULLABLE |
|  |  |  |  |

**Tabla Lecturas**

| Campo | Tipo de Dato | Descripción | Restricciones |
| --- | --- | --- | --- |
| id\_lectura | INT | Identificador único de la lectura. | PRIMARY KEY, AUTO\_INCREMENT |
| id\_sensor | INT | Identificador del sensor asociado. | FOREIGN KEY (Sensores) |
| valor | float | Valor de la lectura (temperatura, humedad, etc.) | NULLABLE |
| fecha\_lectura | DATETIME | Fecha y hora de la lectura | NOT NULL |

**Tabla Controles**

| Campo | Tipo de Dato | Descripción | Restricciones |
| --- | --- | --- | --- |
| id\_control | INT | Identificador único del control. | PRIMARY KEY, AUTO\_INCREMENT |
| id\_sensor | INT | Identificador del sensor ajustado. | FOREIGN KEY (Sensores) |
| valor\_ajuste | FLOAT | Valor ajustado (por ejemplo, límite de temperatura) | NOT NULL |
| fecha\_ajuste | DATETIME | Fecha del ajuste | NOT NULL |

**Tabla Eventos**

| Campo | Tipo de Dato | Descripción | Restricciones |
| --- | --- | --- | --- |
| id\_evento | INT | Identificador único del evento. | PRIMARY KEY, AUTO\_INCREMENT |
| descripción | VARCHAR | Descripción del evento (ajuste de sistema, error, etc.). | NOT NULL |
| fecha\_evento | DATETIME | Fecha en que ocurrió el evento | NOT NULL |

## 4.1. Relaciones y Restricciones:

**Usuarios ↔ Sensores**

* Un **usuario** puede gestionar múltiples **sensores**, pero cada **sensor** está asociado a un único **usuario**.
* La relación entre **Usuarios** y **Sensores** se establece mediante el campo id\_usuario en la tabla **Sensores**, que actúa como una clave foránea que hace referencia al campo id\_usuario en la tabla **Usuarios**.
* **Restricción**: Cada **sensor** debe pertenecer a un solo **usuario**, lo que garantiza que no haya confusión en la propiedad y gestión de los sensores.

**Sensores ↔ Lecturas**

* Un **sensor** puede generar muchas **lecturas**, pero cada **lectura** está asociada a un único **sensor**.
* La relación entre **Sensores** y **Lecturas** se establece mediante el campo id\_sensor en la tabla **Lecturas**, que es una clave foránea que hace referencia al campo id\_sensor en la tabla **Sensores**.
* **Restricción**: Cada **lectura** debe pertenecer a un solo **sensor**, lo que asegura que las mediciones se registren de manera organizada y precisa.

**Sensores ↔ Controles**

* Un **sensor** puede ser ajustado múltiples veces a lo largo del tiempo, pero cada **control** está vinculado a un único **sensor**.
* La relación entre **Sensores** y **Controles** se representa mediante el campo id\_sensor en la tabla **Controles**, que es una clave foránea que hace referencia al campo id\_sensor en la tabla **Sensores**.
* **Restricción**: Cada **control** debe corresponder a un único **sensor**, lo que permite llevar un registro detallado de los ajustes realizados.

**Usuarios ↔ Eventos**

* Un **usuario** puede generar múltiples **eventos**, pero cada **evento** está asociado a un único **usuario**.
* La relación entre **Usuarios** y **Eventos** se establece mediante el campo id\_usuario en la tabla **Eventos**, que es una clave foránea que hace referencia al campo id\_usuario en la tabla **Usuarios**.
* **Restricción**: Cada **evento** debe tener un único **usuario** responsable, lo que asegura la trazabilidad de las acciones en el sistema.

**Usuarios ↔ Controles**

* Un usuario puede realizar múltiples controles a lo largo del tiempo, pero cada control está asociado a un único usuario.
* La relación entre Usuarios y Controles se establece mediante el campo id\_usuario en la tabla Controles, que actúa como una clave foránea que hace referencia al campo id\_usuario en la tabla Usuarios.
* Restricción: Cada control debe estar vinculado a un único usuario, lo que asegura la trazabilidad de las acciones y permite identificar con precisión quién realizó cada ajuste en el sistema.

# 5. Consideraciones de Escalabilidad

* **Índices**: Crear índices en campos de búsqueda frecuente, como email, id\_sensor y fecha\_lectura.
* **Integridad Referencial**: Uso de claves foráneas con opción de eliminación en cascada (ON DELETE CASCADE) para garantizar la integridad de los datos cuando un sensor o usuario se elimina.
* **Particionamiento de Tablas**: Las tablas de lecturas y controles pueden ser particionadas por sensor para optimizar el rendimiento con el crecimiento de los datos.

# 6. Conclusiones

Este modelo de base de datos relacional permitirá gestionar de manera eficiente la información generada por los sensores y sistemas automatizados del invernadero. La correcta estructuración de las tablas y relaciones garantiza una escalabilidad adecuada, permitiendo la expansión futura del sistema sin comprometer la integridad de los datos. Además, la implementación de medidas de integridad referencial y optimización de consultas asegurará un rendimiento adecuado a medida que se vayan incorporando más sensores y datos al sistema. El proyecto proporcionará a la comunidad de Chihuaco una herramienta valiosa para mejorar la eficiencia de sus cultivos mediante el uso de tecnologías de automatización de bajo costo.