

Buenas noches, aprovecho la oportunidad de enviar, además, la Entrega N° 1, en este documento. Saludos.

Entrega N°1

Gestión de Asistencia_RRHH

1. Introducción

El presente proyecto desarrolla una Base de Datos orientada a la gestión de la asistencia del personal dentro de una organización. A través de la misma, se busca registrar entradas y salidas de empleados, sus sectores de pertenencia, así como información clave de cada trabajador, permitiendo a Recursos Humanos y otras áreas operativas (como Contabilidad o Logística) gestionar y analizar el desempeño laboral, cumplimiento horario, asignación de presupuestos y dimensionamiento de recursos humanos.

2. Objetivo

El objetivo principal del proyecto es implementar una base de datos funcional que permita:

- Registrar y controlar las asistencias de los empleados.
- Asociar cada empleado a un sector específico de la empresa.
- Llevar un control de las jornadas laborales, horas extra y llegadas tarde.
- Mejorar la gestión de personal mediante informes analíticos basados en la información registrada.
- Brindar información contable precisa sobre la cantidad de horas trabajadas y salarios, ayudando en los cierres de nómina.

El proyecto es transversal a diferentes áreas de la organización: Recursos Humanos, Administración, Finanzas y Logística.

3. Situación Problemática

Actualmente, la empresa registra la asistencia de sus empleados en planillas manuales, lo que genera:

- Errores de carga y pérdida de información.
- Demoras en la generación de reportes de presentismo.
- Dificultad para calcular horas extras y ausencias justificadas.
- Falta de visibilidad sobre la distribución de empleados por sector y su presupuesto.

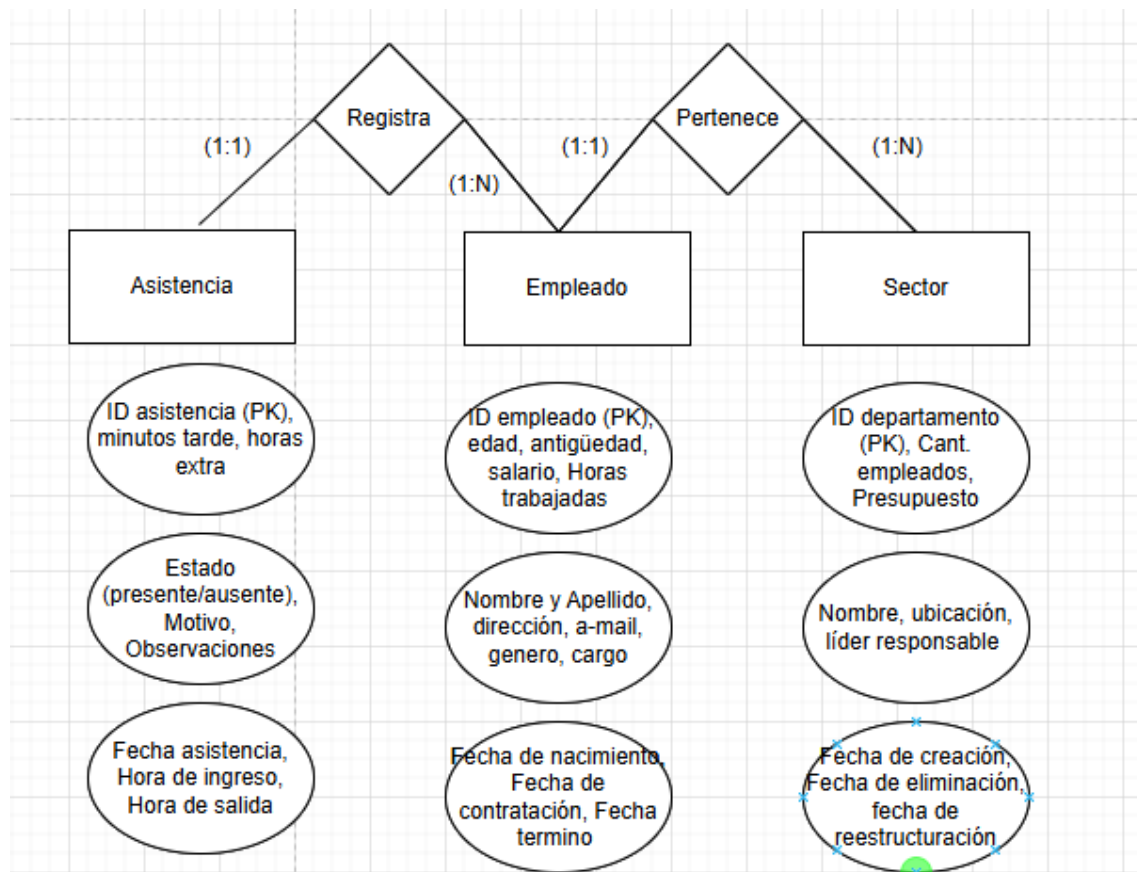
La implementación de esta base de datos permitirá solucionar estas brechas, optimizando el control de la asistencia, automatizando cálculos y centralizando toda la información crítica en un solo sistema.

4. Modelo de Negocio

Se plantea una organización mediana-grande, con diferentes sectores (Administración, Producción, Logística, etc.), cada uno con su propio presupuesto y equipo de empleados.

Recursos Humanos es el área encargada de registrar datos de los empleados y controlar su asistencia diaria.

5. Diagrama Entidad-Relación (E-R)



- Cada Empleado pertenece a un Sector.
- Cada Empleado registra múltiples Asistencias.
- Relaciones 1:N entre Empleado-Asistencia y Sector-Empleado.

6. Listado de Tablas

Tabla	Descripción	Campos	Abreviaturas	Tipo de Dato	Claves
Sector	Información de los sectores de la empresa	ID_depto, cant_emp, presupuesto, nombre, ubicacion, lider_resp, fecha_creac, fecha_eli, fecha_reestr	ID_depto, cant_emp, presupuesto, nombre, ubicacion, lider_resp, fecha_creac, fecha_eli, fecha_reestr	INT, DECIMAL, VARCHAR, DATE	PK: ID_depto
Empleado	Datos personales y laborales de los empleados	ID_empleado, edad, antigüedad, salario, horas_trab, nombre_apellido, direccion, email, genero, cargo, fecha_nac, fecha_contr, fecha_term, ID_depto	ID_empleado, edad, antigüedad, salario, horas_trab, nombre_apellido, direccion, email, genero, cargo, fecha_nac, fecha_contr, fecha_term, ID_depto	INT, DECIMAL, VARCHAR, CHAR, DATE	PK: ID_empleado, FK: ID_depto
Asistencia	Registro diario de asistencias	ID_asistencia, minutos_tarde, horas_extra, estado, motivo, observaciones, fecha_asist, hora_ingreso, hora_salida, ID_empleado	ID_asistencia, minutos_tarde, horas_extra, estado, motivo, observaciones, fecha_asist, hora_ingreso, hora_salida, ID_empleado	INT, DECIMAL, VARCHAR, TEXT, DATE, TIME	PK: ID_asistencia, FK: ID_empleado

7. Script SQL de creación de la base de datos

[Repositorio en Github](#)

Entrega N° 2

1. Listado de Vistas

Vista: Vista_AsistenciaCompleta

- **Descripción:** Muestra un resumen de asistencias con datos del empleado y del sector.
- **Objetivo:** Facilita consultas administrativas y reportes de RRHH.
- **Tablas involucradas:** Asistencia, Empleado, Sector.

```
CREATE VIEW Vista_AsistenciaCompleta AS
SELECT A.ID_asistencia, E.nombre_apellido, S.nombre AS sector, A.fecha_asist,
A.hora_ingreso, A.hora_salida,
      A.minutos_tarde, A.horas_extra, A.estado
FROM Asistencia A
JOIN Empleado E ON A.ID_empleado = E.ID_empleado
JOIN Sector S ON E.ID_depto = S.ID_depto;
```

Vista: Vista_ResumenHorasPorEmpleado

- **Descripción:** Totaliza las horas trabajadas, horas extra y minutos tarde por empleado.
- **Objetivo:** Brindar informes de productividad individual.
- **Tablas involucradas:** Asistencia, Empleado.

```
CREATE VIEW Vista_ResumenHorasPorEmpleado AS
SELECT E.ID_empleado, E.nombre_apellido,
      SUM(A.horas_extra) AS TotalHorasExtra,
      SUM(A.minutos_tarde) AS TotalMinutosTarde
FROM Asistencia A
JOIN Empleado E ON A.ID_empleado = E.ID_empleado
GROUP BY E.ID_empleado, E.nombre_apellido;
```

2. Listado de Funciones

Función: calcularEdad(fecha_nac DATE)

- **Descripción:** Devuelve la edad de un empleado a partir de su fecha de nacimiento.
- **Objetivo:** Cálculos demográficos y agrupamientos por rangos etarios.
- **Tablas involucradas:** Se usa sobre campos de la tabla Empleado.

```
DELIMITER $$
CREATE FUNCTION calcularEdad(fecha_nac DATE)
```

```

RETURNS INT
DETERMINISTIC
BEGIN
    RETURN TIMESTAMPDIFF(YEAR, fecha_nac, CURDATE());
END$$
DELIMITER ;

```

Función: totalHorasEmpleado(idEmp INT)

- **Descripción:** Devuelve el total de horas extra registradas por un empleado.
- **Objetivo:** Consultas individuales desde el frontend.
- **Tablas involucradas:** Asistencia.

```

DELIMITER $$
CREATE FUNCTION totalHorasEmpleado(idEmp INT)
RETURNS DECIMAL(10,2)
DETERMINISTIC
BEGIN
    DECLARE total DECIMAL(10,2);
    SELECT SUM(horas_extra) INTO total FROM Asistencia WHERE ID_empleado = idEmp;
    RETURN total;
END$$
DELIMITER ;

```

3. Listado de Stored Procedures

SP: sp_InsertarEmpleado

- **Descripción:** Inserta un nuevo registro de empleado.
- **Objetivo:** Automatiza inserciones desde formularios.
- **Tablas involucradas:** Empleado.

```

DELIMITER $$
CREATE PROCEDURE sp_InsertarEmpleado (
    IN p_nombre_apellido VARCHAR(100),
    IN p_edad INT,
    IN p_salario DECIMAL(10,2),
    IN p_ID_depto INT
)
BEGIN
    INSERT INTO Empleado(nombre_apellido, edad, salario, ID_depto)
    VALUES (p_nombre_apellido, p_edad, p_salario, p_ID_depto);
END$$
DELIMITER ;

```

SP: sp_AsisntenciaDelDia

- **Descripción:** Retorna las asistencias registradas para una fecha determinada.
- **Objetivo:** Generación diaria de reportes.
- **Tablas involucradas:** Asistencia, Empleado.

```
DELIMITER $$
CREATE PROCEDURE sp_AsisntenciaDelDia (IN p_fecha DATE)
BEGIN
    SELECT A.*, E.nombre_apellido
    FROM Asistencia A
    JOIN Empleado E ON A.ID_empleado = E.ID_empleado
    WHERE A.fecha_asist = p_fecha;
END$$
DELIMITER ;
```

4. Listado de Triggers

Trigger: trg_ActualizarCantEmp

- **Descripción:** Actualiza el campo cant_emp en la tabla Sector cuando se inserta un nuevo empleado.
- **Objetivo:** Mantener actualizada la cantidad de empleados por sector.
- **Tablas involucradas:** Empleado, Sector.

```
DELIMITER $$
CREATE TRIGGER trg_ActualizarCantEmp
AFTER INSERT ON Empleado
FOR EACH ROW
BEGIN
    UPDATE Sector
    SET cant_emp = cant_emp + 1
    WHERE ID_depto = NEW.ID_depto;
END$$
DELIMITER ;
```

Trigger: trg_DecrementarCantEmp

- **Descripción:** Disminuye en 1 el campo cant_emp en la tabla Sector cuando se elimina un empleado de la tabla Empleado.
- **Objetivo:** Mantener actualizada la cantidad de empleados por sector también cuando se produce una baja.
- **Tablas involucradas:** Empleado, Sector.

```

DELIMITER $$
CREATE TRIGGER trg_DecrementarCantEmp
AFTER DELETE ON Empleado
FOR EACH ROW
BEGIN
    UPDATE Sector
    SET cant_emp = cant_emp - 1
    WHERE ID_depto = OLD.ID_depto;
END$$
DELIMITER ;

```

5. Script de inclusión de datos

Sectores

```

INSERT INTO Sector (cant_emp, presupuesto, nombre, ubicacion, lider_resp,
fecha_creac)
VALUES
(0, 100000, 'Administración', 'Piso 1', 'Ana Torres', '2023-01-01'),
(0, 150000, 'Producción', 'Planta Baja', 'Carlos Ruiz', '2023-01-01');

```

Empleados

```

INSERT INTO Empleado (edad, antigüedad, salario, horas_trab, nombre_apellido,
direccion, email, genero, cargo, fecha_nac, fecha_contr, ID_depto)
VALUES
(30, 5, 55000, 160, 'Lucia Romero', 'Calle Falsa 123', 'lucia@empresa.com', 'F',
'Analista', '1993-06-15', '2018-04-10', 1),
(40, 10, 70000, 180, 'Pedro Gómez', 'Av. Siempre Viva 742', 'pedro@empresa.com',
'M', 'Supervisor', '1984-11-20', '2013-09-01', 2);

```

Asistencias

```

INSERT INTO Asistencia (minutos_tarde, horas_extra, estado, motivo, observaciones,
fecha_asist, hora_ingreso, hora_salida, ID_empleado)
VALUES
(0, 2.5, 'Presente', NULL, '', '2024-10-01', '08:00:00', '17:00:00', 1),
(15, 0, 'Tarde', 'Tráfico', 'Demorado por embotellamiento', '2024-10-01', '08:15:00',
'17:00:00', 2);

```

- Archivos: [Repositorio en Github](#)