



Manual Técnico

Proyecto: Sistema de Control de Empleados para
Recursos Humanos

Programación II

Catedrático: Ing. Alan G. Ucelo Morán

Universidad Mariano Gálvez de Guatemala

Ingeniería en Sistemas de la Información

Ingeniería en Sistemas de Información y Ciencias de la
Computación

**Sistema de Control de Empleados para
Recursos Humanos
MANUAL TÉCNICO**



Sistema de Control de Empleados para Recursos Humanos

MANUAL TÉCNICO

Integrantes:

Juan Enrique Samayoa Reyes

9390-23-2010

**Sistema de Control de Empleados para
Recursos Humanos
MANUAL TÉCNICO**

Contenido

Introducción	1
Objetivos	2
Alcance	3
Requisitos de Instalación	4
Requerimientos Técnicos.....	4
Herramientas Utilizadas para el Desarrollo.....	5
Instalación.....	5
Usuarios.....	6
Usuarios de aplicaciones.....	6
Aspectos Importantes	7
Procedimientos y Funciones.....	7
Diagrama de Clases	16
Diagrama Entidad-Relación	17
Glosario.....	18

Introducción

Este proyecto consiste en un sistema informático de gestión de personal orientado a las necesidades de un departamento de Recursos Humanos. Desarrollado en Java y utilizando una base de datos SQL, el sistema permitirá gestionar información de empleados, realizar procesos de contratación y administrar el control de acciones y movimientos de personal.

Objetivos

General:

Implementar un sistema en Java que permita la administración de personal para un departamento de Recursos Humanos, cubriendo necesidades de registro, consulta y modificación de datos de empleados, así como el seguimiento de acciones y movimientos del personal.

Específicos:

- Desarrollar una interfaz gráfica amigable.
- Implementar funciones de registro y modificación de datos de empleados.
- Integrar el sistema con una base de datos para almacenar y gestionar la información.
- Incorporar controles de seguridad mediante la gestión de usuarios y roles.
- Facilitar la generación de reportes para la toma de decisiones.

Alcance

El sistema cubre las siguientes funcionalidades:

- Registro de empleados y actualización de su información.
- Gestión de contrataciones.
- Control de acciones de personal (permisos, vacaciones, etc.).
- Gestión de movimientos del personal (transferencias, promociones).
- Generación de reportes (empleados por departamento, movimientos, contrataciones recientes).

Este documento está dirigido a: Todos los operadores, administrativos, propietario(s) y personas interesadas.

Conocimientos básicos en: Operación de sistemas informáticos, Office y Windows

Requisitos de Instalación

Requerimientos Técnicos

Requerimientos Mínimos de Hardware:

- Procesador: Intel Core i3 o superior.
- Memoria RAM: 4 GB.
- Espacio en disco: 500 MB libres.
- Resolución de pantalla: 1280x720 píxeles.

Requerimientos Mínimos de Software:

- Java Development Kit (JDK) versión 8 o superior.
- NetBeans IDE (preferentemente versión 12 o superior).
- SQL Server o MySQL para gestión de la base de datos.

Herramientas Utilizadas para el Desarrollo.

- *Lenguaje de Programación: Java.*
- *Entorno de Desarrollo: NetBeans IDE.*
- *Sistema Gestor de Base de Datos: SQL Server o MySQL.*
- *Documentación y Diagramas: Utilización de diagramas UML para clases y relaciones.*

Instalación

- Configuración del Entorno: Instalar JDK y configurar NetBeans.
- Base de Datos: Crear una base de datos en SQL Server o MySQL y ejecutar los scripts de SQL para crear tablas y relaciones.
- Configuración de Conexión: Configurar el archivo de propiedades en Java para establecer la conexión a la base de datos.

Usuarios

Usuarios de aplicaciones

Los usuarios de la aplicación se clasifican según los roles: Administrador, Reclutador y Operador, cada uno con permisos específicos que garantizan la seguridad y la integridad de los datos.

- **Administrador:**
Acceso completo al sistema, incluyendo gestión de usuarios y generación de reportes.
- **Reclutador:**
Acceso al módulo de gestión y contratación de personal.
- **Operador:**
Acceso al control de acciones y movimientos de personal.

Aspectos Importantes

- Seguridad: Uso de roles y contraseñas seguras para acceder al sistema.
- Escalabilidad: Diseño modular para facilitar la adición de nuevas funcionalidades en el futuro.
- Integridad de Datos: Mantenimiento de registros de auditoría para modificaciones y accesos.

Procedimientos y Funciones

Clase Usuario

Atributos:

- int idUsuario: Numero entero para identificar cada Usuario en la base de datos.
- String nombreUsuario: Cadena de caracteres para almacenar el nombre del Usuario registrado.
- String nombreCompleto: Cadena de caracteres para almacenar el nombre completo del usuario asociado.
- String correoElectronico: Cadena de caracteres para almacenar el correo electrónico asociado al usuario con el formato "*nombre@dominio.com*".
- String teléfono: Cadena de caracteres para almacenar los distintos números telefónicos.
- String contraseña: Cadena de caracteres en formato hash para poder almacenarse de manera seguro.
- String tipoUsuario: Cadena de caracteres que únicamente puede ser "Administrador", "Operador" o "Reclutador".
- ArrayList<Usuario> usuarios: Arreglo de tipo Usuarios para poder almacenar todos los usuarios creados en el sistema.

Métodos

- crearUsuario(String usuario, String nombre, String nombreC, String correo, String telefono, String contra, String tipoUsr) boolean: Método que devuelve un booleano para confirmar si ha sido creado el usuario exitosamente o no.
- iniciarSesion(String usr, String con) boolean: Método que devuelve un booleano si el usuario y contraseña es ingresado correctamente.

Obteniendo los datos de la base de datos.

- verificarPermisos(Usuario usuario) String: Devuelve una cadena de caracteres según el usuario de tipo Usuario que se ha ingresado y valida su permiso asignado.
- obtenerTodosLosUsuarios() void: Método que obtiene todos los usuarios almacenados en la base de datos y los ingresa uno a uno, en el arreglo de usuarios para poder tener acceso a todos los usuarios.
- obtenerUsuario(String nombreUsuario) Usuario: Devuelve un usuario de tipo Usuario en base a su nombre de usuario dentro del arreglo de usuarios.
- getNextIdUsuario() int: Devuelve un entero con el último ID asignable a un usuario para poder agregarlo dentro de la base de datos.
- actualizarUsuario(int idUsuario, String nombreUsuario, String nombreCompleto, String correoElectronico, String telefono, String tipoUsuario) boolean: Devuelve un dato booleano para confirmar si el usuario ha sido actualizado correctamente en la base de datos.
- eliminarUsuario(int idUsuario) boolean: Devuelve un booleano si el usuario ha sido encontrado o no dentro de la base de datos.
- encriptarContraseña(String pass) String: Devuelve una cadena de caracteres en base a otra cadena de caracteres para poder devolver una contraseña encriptada.

Clase Empleado

Atributos

- int idEmpleado: Identificador único del empleado. Este número permite diferenciar a cada empleado en el sistema.
- String nombre: Almacena el primer nombre del empleado.
- String apellido: Almacena el apellido del empleado.
- String fechaContratacion: Fecha en la que el empleado fue contratado. Generalmente en formato de texto para facilitar su almacenamiento.
- String cargo: Describe el cargo o posición que ocupa el empleado en la empresa (por ejemplo, "Gerente de Recursos Humanos").
- double salario: Almacena el salario asignado al empleado, permitiendo cálculos salariales u otros procesos de gestión.
- String departamento: Describe el departamento al cual pertenece el empleado (como "Auditoría Interna" o "Administración").
- ArrayList<AccionPersonal> accionPersonal: Lista de acciones relacionadas con el personal (por ejemplo, permisos o vacaciones) asociadas a este empleado. Cada elemento de esta lista es una instancia de la clase AccionPersonal.

- `ArrayList<MovimientoPersonal> movimientoPersonal`: Lista que registra los movimientos del empleado dentro de la organización, como ascensos o transferencias. Cada movimiento es una instancia de la clase `MovimientoPersonal`.
- `ArrayList<Contratacion> contrataciones`: Lista de contratos que el empleado ha tenido con la organización. Permite tener un historial de sus contrataciones.
- `ArrayList<Empleado> empleados`: Lista que almacena todos los empleados en el sistema. Es útil para gestionar y realizar operaciones a nivel de todo el personal.

Métodos

- `getInformacionEmpleado(int id) Empleado`: Busca y retorna la información completa de un empleado a partir de su `idEmpleado`. Si el empleado existe, devuelve un objeto `Empleado` con su información; de lo contrario, retorna `null`.
- `modificarInformacionEmpleado(int id, String nom, String ape, String fecha, String car, double sal, String dep) boolean`: Permite modificar la información básica de un empleado (nombre, apellido, fecha de contratación, cargo, salario, departamento) a partir de su `idEmpleado`. Retorna `true` si la modificación fue exitosa y `false` en caso de error (por ejemplo, si el empleado no existe).
- `calcularVacaciones(int id) int`: Calcula el número de días de vacaciones disponibles para el empleado según su `idEmpleado`. La lógica exacta puede basarse en la fecha de contratación, el cargo o las políticas de la empresa.
- `getNextIdEmpleado() int`: Genera y retorna el siguiente `idEmpleado` único para asignar a un nuevo empleado. Este método asegura que cada empleado tenga un identificador único dentro del sistema.
- `agregarEmpleado(String nombre, String apellido, String fechaContratacion, String cargo, double salario, String departamento) boolean`: Añade un nuevo empleado a la lista de empleados (`empleados`). Recibe la información básica del empleado y devuelve `true` si el empleado se ha agregado exitosamente o `false` si ocurre algún error.
- `eliminarEmpleado(int id) boolean`: Elimina un empleado específico identificado

por su idEmpleado de la lista empleados. Retorna true si la eliminación fue exitosa y false si el empleado no existe.

- obtenerTodosLosEmpleados() void: Muestra la información de todos los empleados almacenados en el sistema. Es útil para obtener una vista general de los empleados.
- convertirAFechaISO(String fecha) String: Convierte una fecha en formato estándar a formato ISO (por ejemplo, de dd/MM/yyyy a yyyy-MM-dd). Es útil para mantener uniformidad en el almacenamiento y manipulación de fechas.
- convertirAFechaEstandar(String fecha) String: Realiza la conversión inversa, es decir, transforma una fecha en formato ISO a un formato estándar. Facilita la presentación de fechas en un formato más comprensible para el usuario.

Clase Contratación

Atributos

- int idContratacion: Identificador único de cada contrato. Permite diferenciar cada contrato en el sistema.
- int idEmpleado: Identificador del empleado asociado a este contrato, vinculando así al empleado con su contrato específico.
- String fechaInicio: Fecha en la que el contrato entra en vigor. Usualmente almacenada en formato de texto.
- String tipoContrato: Tipo de contrato, por ejemplo, "indefinido", "temporal", "por obra", etc., indicando la naturaleza del vínculo laboral.
- int duracionContrato: Duración del contrato en meses o años, dependiendo del tipo de contrato.
- double salario: Salario acordado para el empleado en este contrato, almacenado como un valor numérico que puede usarse para cálculos o reportes salariales.
- ArrayList<Contratacion> contratosActivos: Lista de todos los contratos actualmente activos en el sistema. Permite llevar un registro y gestionar contratos en vigor.

Métodos

- `crearContrato(int idE, String fechaIn, String tipoCon, int dur, double sal)` boolean: Crea un nuevo contrato para un empleado específico. Recibe el ID del empleado, fecha de inicio, tipo de contrato, duración y salario. Devuelve true si el contrato fue creado exitosamente o false si hubo algún problema.
- `finalizarContrato(int idContrato)` boolean: Termina un contrato específico identificado por su idContratacion. El contrato se marca como finalizado, y el método retorna true si la finalización fue exitosa o false si el contrato no existe.
- `consultarContratosVigentes()` ArrayList<Contratacion>: Devuelve una lista de todos los contratos que aún están vigentes. Esto facilita la consulta de contratos activos sin incluir aquellos ya finalizados.
- `getNextIdContratacion()` int: Genera y retorna el siguiente idContratacion único para asignar a un nuevo contrato. Esto asegura que cada contrato tenga un identificador único en el sistema.
- `agregarContrato(int idEmpleado, String fechaInicio, String tipoContrato, int duracionContrato, double salario)` boolean: Agrega un nuevo contrato a la lista contratosActivos. Recibe la información básica del contrato y retorna true si se añadió correctamente o false en caso de error.
- `modificarContrato(int id, int idEmpleado, String fechaInicio, String tipoContrato, int duracionContrato, double salario)` boolean: Permite modificar los detalles de un contrato específico, identificando el contrato por su idContratacion. Retorna true si la modificación fue exitosa o false si el contrato no existe.
- `convertirAFechaISO(String fecha)` String: Convierte una fecha en formato estándar a formato ISO (yyyy-MM-dd). Esta conversión es útil para uniformizar el formato de las fechas en el sistema.
- `convertirAFechaEstandar(String fecha)` String: Convierte una fecha en formato ISO al formato estándar (dd/MM/yyyy), facilitando su presentación en un formato comprensible para el usuario final.

Clase AccionPersonal

Atributos

- `int idAccion`: Identificador único de cada acción registrada. Ayuda a diferenciar cada acción en el sistema.
- `int idEmpleado`: Identificador del empleado al que está asociada la acción. Este atributo permite vincular la acción a un empleado específico.
- `String tipoAccion`: Describe el tipo de acción realizada, como "permiso", "vacaciones", "licencia", entre otros, para clasificar las diferentes acciones de personal.
- `String fechaAccion`: Fecha en la que se realiza la acción, generalmente almacenada en formato de texto.
- `String detalle`: Contiene información adicional sobre la acción realizada. Este atributo permite registrar comentarios específicos o descripciones adicionales.
- `ArrayList<AccionPersonal> acciones`: Lista de todas las acciones registradas en el sistema. Sirve para almacenar y gestionar múltiples acciones de empleados.

Métodos

- `registrarAccion(int idEmpleado, String tipo, String fecha, String detalle) boolean`: Registra una nueva acción asociada a un empleado. Recibe el ID del empleado, el tipo de acción, la fecha y detalles adicionales. Devuelve true si el registro fue exitoso o false en caso contrario.
- `consultarAccionesEmpleado(int idEmpleado) ArrayList<AccionPersonal>`: Retorna una lista de todas las acciones realizadas por un empleado específico, utilizando el `idEmpleado` para filtrar la información. Es útil para revisar el historial de acciones de un empleado en particular.
- `generarReporteAcciones(String fechaI, String fechaF) ArrayList<AccionPersonal>`: Genera y retorna un reporte de todas las acciones registradas dentro de un rango de fechas especificado por `fechaI` (fecha de inicio) y `fechaF` (fecha final). Permite obtener un resumen de las acciones ocurridas en un periodo determinado.
- `convertirAFechaISO(String fecha) String`: Convierte una fecha en formato estándar a formato ISO (yyyy-MM-dd). Ayuda a uniformizar las fechas dentro del sistema para simplificar su gestión y almacenamiento.

- `convertirAFechaEstandar(String fecha)` String: Realiza la conversión inversa, es decir, de formato ISO a un formato estándar (dd/MM/yyyy). Esto facilita la presentación de fechas de una forma más comprensible para el usuario final.

Clase MovimientoPersonal

Atributos

- `int idMovimiento`: Identificador único de cada movimiento registrado. Permite diferenciar cada movimiento dentro del sistema.
- `int idEmpleado`: Identificador del empleado al que está asociado el movimiento. Este atributo vincula el movimiento con un empleado específico.
- `String tipoMovimiento`: Describe el tipo de movimiento realizado, como "promoción", "transferencia", "cambio de puesto", etc., para clasificar los distintos movimientos.
- `String fechaMovimiento`: Fecha en la que el movimiento se lleva a cabo, generalmente almacenada como un texto.
- `String detalle`: Información adicional sobre el movimiento. Este atributo permite registrar comentarios específicos o detalles adicionales.
- `ArrayList<MovimientoPersonal> movimientos`: Lista de todos los movimientos registrados en el sistema. Sirve para almacenar y gestionar el historial de movimientos de cada empleado.

Métodos

- `registrarMovimiento(int idEmpleado, String tipo, String fecha, String detalle)` boolean: Registra un nuevo movimiento asociado a un empleado. Recibe el ID del empleado, el tipo de movimiento, la fecha y los detalles adicionales. Retorna true si el registro fue exitoso o false en caso contrario.
- `consultarMovimientosEmpleado(int idEmpleado)` ArrayList<MovimientoPersonal>: Devuelve una lista de todos los movimientos registrados para un empleado específico, filtrando por su idEmpleado. Es útil para revisar el historial de movimientos de un empleado.
- `generarReporteMovimientos(String fechaI, String fechaF)` ArrayList<MovimientoPersonal>: Genera y retorna un reporte de todos los

movimientos realizados dentro de un rango de fechas especificado por fechaI (fecha de inicio) y fechaF (fecha final). Permite obtener un resumen de los movimientos ocurridos en un periodo determinado.

- obtenerFechaActual() String: Obtiene y retorna la fecha actual en un formato estándar. Este método es útil para registrar movimientos con la fecha actual sin necesidad de ingresarla manualmente.
- convertirAFechaISO(String fecha) String: Convierte una fecha de formato estándar a formato ISO (yyyy-MM-dd). Ayuda a uniformizar las fechas en el sistema.
- convertirAFechaEstandar(String fecha) String: Convierte una fecha en formato ISO al formato estándar (dd/MM/yyyy). Facilita la presentación de fechas en un formato más comprensible para el usuario final.

Clase Reportes

Atributos

Atributos Generales de Empleado: Estos atributos se usan en distintos reportes para identificar al empleado involucrado:

- int idEmpleado: Identificador único del empleado, útil para enlazar datos de diversas tablas.
- String nombre y String apellido: Nombre y apellido del empleado, comunes en reportes de contrataciones, movimientos y acciones.
- String fechaContratacion: Fecha en que el empleado fue contratado. Puede usarse en reportes de contratación o al analizar la duración de empleo.

Atributos de Contrato

- String fechaInicio: Fecha de inicio del contrato.
- String tipoContrato: Tipo de contrato (indefinido, temporal, etc.).
- int duracion: Duración del contrato en meses o años.
- double salario: Salario acordado en el contrato.

Atributos de Departamento

- String departamento: Departamento al que pertenece el empleado.
- int cantidadEmpleados: Cantidad total de empleados en cada departamento. Este dato es calculado en el reporte para obtener un resumen por cada área.
- Atributos de Acciones (usados en generarReporteAusenciasVacaciones):

- String tipoAccion: Tipo de acción realizada por el empleado, como "permiso", "vacaciones", etc.
- String fechaAccion: Fecha en la que la acción tuvo lugar.
- String detalleAccion: Información adicional sobre la acción.

Atributos de Movimientos

- String tipoMovimiento: Tipo de movimiento (por ejemplo, promoción, transferencia).
- String fechaMovimiento: Fecha en la que se realizó el movimiento.
- String detalleMovimiento: Descripción adicional del movimiento realizado.
- ArrayList<Reporte> listaReportes: Lista que almacena múltiples reportes generados en el sistema, permitiendo consultar o exportar los reportes generados.

Métodos

- generarReporteEmpleadosPorDepartamento() ArrayList<Reporte>: Genera un reporte que lista la cantidad de empleados en cada departamento. Este método utiliza consultas JOIN para contar empleados y obtener datos por departamento, como departamento y cantidadEmpleados.
- generarReporteMovimientosPersonal() ArrayList<Reporte>: Crea un reporte con detalles de los movimientos realizados por los empleados, incluyendo promociones, transferencias, y otros cambios. Los datos incluyen tipoMovimiento, fechaMovimiento y detalleMovimiento, junto con la información básica del empleado.
- generarReporteContratacionesRecientes() ArrayList<Reporte>: Genera un reporte de los contratos más recientes, obteniendo datos como fechaInicio, tipoContrato, duracion y salario, junto con la información de identificación del empleado. Este reporte ayuda a analizar la actividad de contratación reciente.
- generarReporteAusenciasVacaciones() ArrayList<Reporte>: Produce un reporte de ausencias y vacaciones, detallando cada acción tomada por el empleado con datos como tipoAccion, fechaAccion y detalleAccion. Es útil para ver el historial de permisos y vacaciones.

Diagrama de Clases

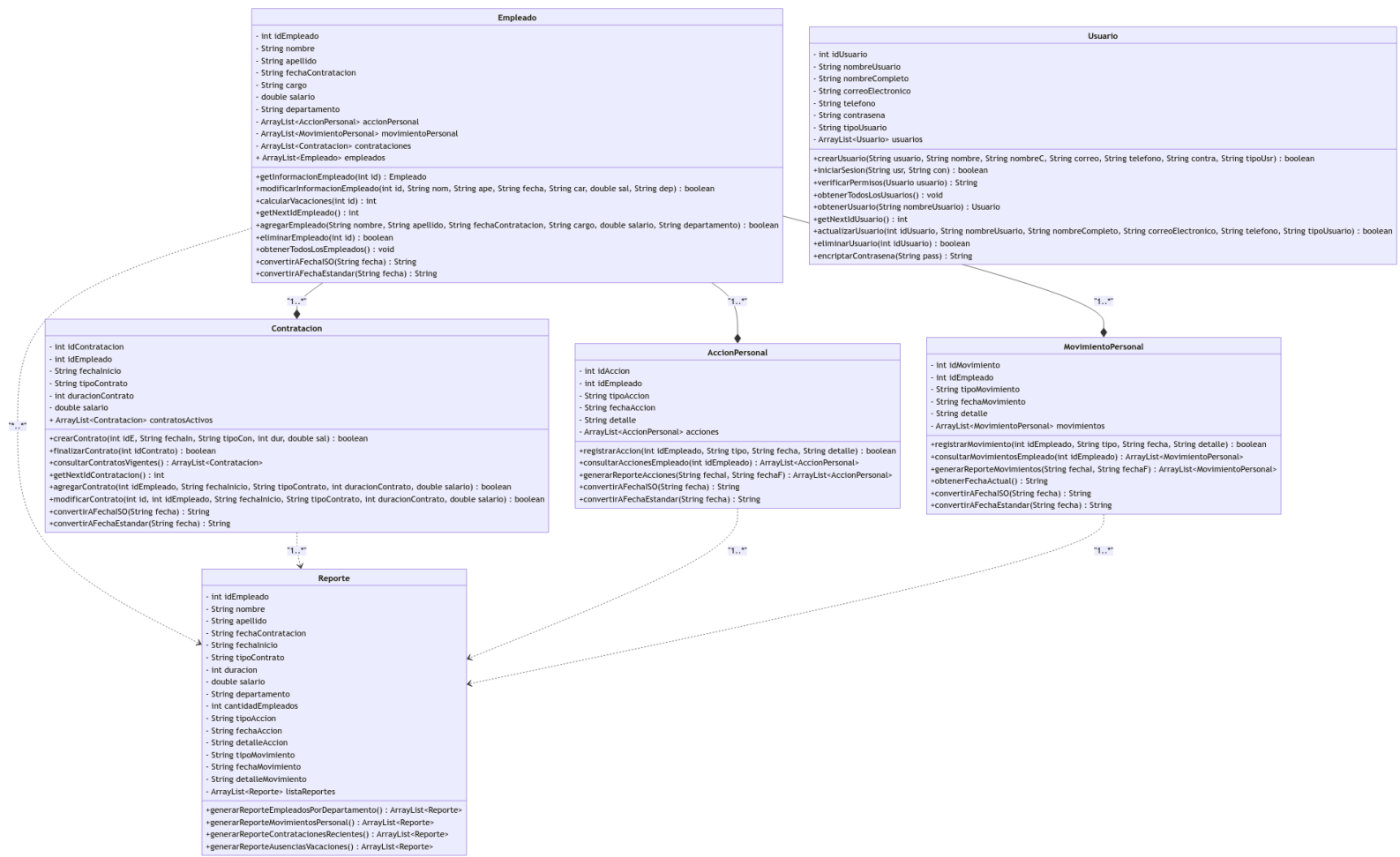
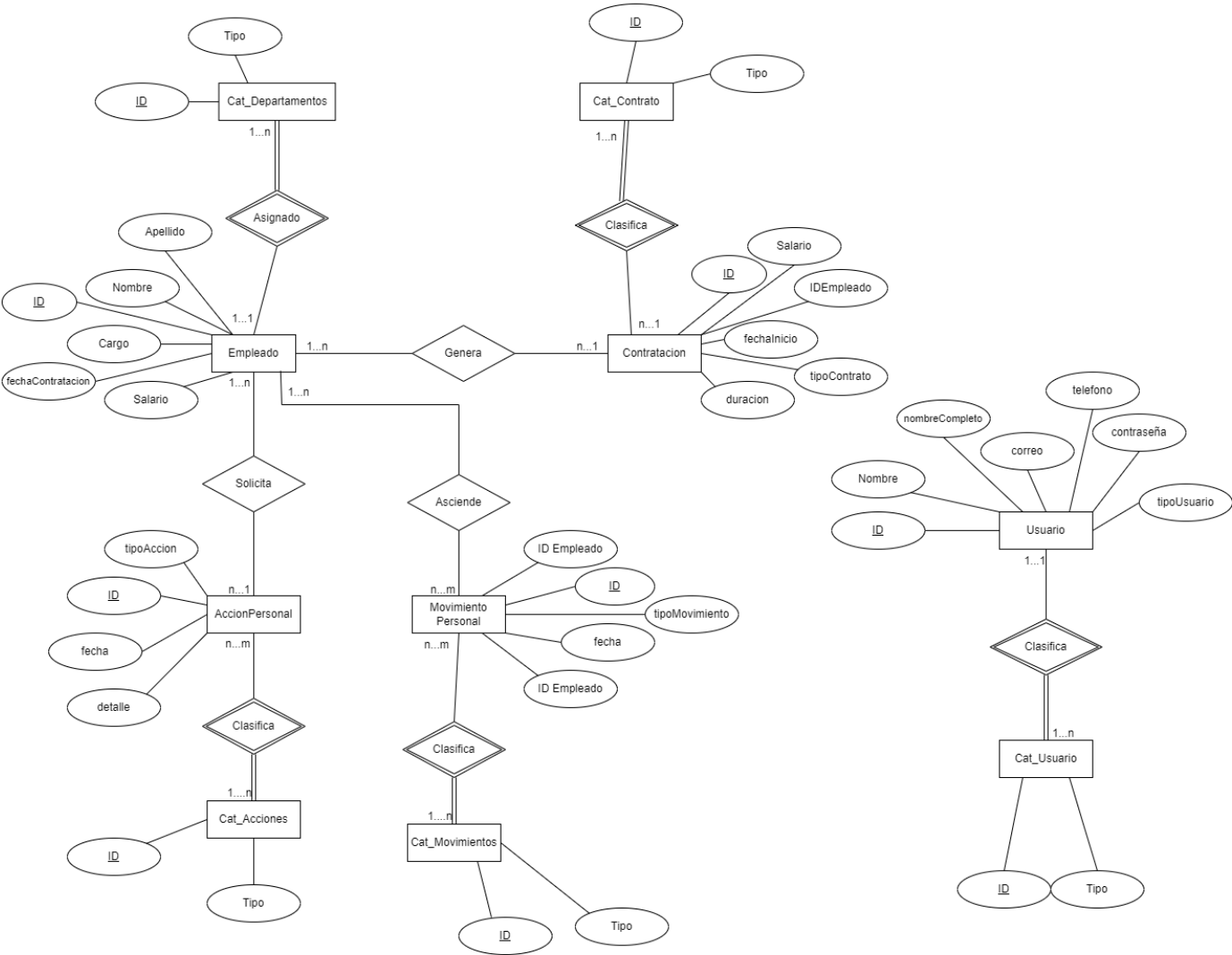


Diagrama Entidad-Relación



Glosario

AccionPersonal

Registro de eventos administrativos como permisos, licencias y vacaciones de un empleado.

Administrador

Usuario con acceso completo al sistema, incluyendo la gestión de usuarios y la generación de reportes.

Contratación

Proceso de incorporar un nuevo empleado a la empresa, incluyendo detalles del contrato como tipo, duración y salario.

Departamento

Sección o división dentro de una empresa a la cual pertenece un empleado, como "Recursos Humanos" o "Finanzas".

DuracionContrato

Período durante el cual el contrato del empleado es válido, medido en meses o años.

Empleado

Persona que trabaja para la empresa y cuya información es gestionada en el sistema.

FechaContratacion

La fecha en que un empleado fue oficialmente contratado por la empresa.

FechaInicio

Fecha en la que comienza el contrato de un empleado.

MovimientoPersonal

Registro de cambios en la posición o departamento de un empleado, como promociones, transferencias y cambios de puesto.

Operador

Usuario con acceso limitado al sistema, enfocado en el control de acciones y movimientos de personal.

Reclutador

Usuario encargado de gestionar la contratación de nuevo personal, con acceso a los módulos correspondientes.

Salario

Remuneración acordada que un empleado recibe por su trabajo, especificada en el contrato.

TipoAccion

Clasificación del tipo de evento administrativo registrado, como "permiso", "vacaciones" o "licencia".

TipoContrato

Clasificación del contrato de un empleado, como "indefinido", "temporal" o "por obra".

TipoMovimiento

Clasificación del tipo de cambio en la posición o departamento de un empleado, como "promoción" o "transferencia".

TipoUsuario

Clasificación del nivel de acceso de un usuario dentro del sistema, como "Administrador", "Operador" o "Reclutador".

Usuario

Persona que utiliza el sistema y que tiene un perfil registrado con un conjunto específico de permisos y accesos.