

## **Documentación del Proyecto: Sistema de Gestión para Microempresas v5.1**

### **1. Introducción**

Este documento detalla el desarrollo y las funcionalidades del "Sistema de Gestión para Microempresas v5.1", una aplicación de escritorio diseñada para centralizar y automatizar la gestión de clientes, proyectos, empleados, tareas, gastos y facturación. El software tiene como objetivo principal optimizar la organización, el seguimiento de las operaciones y la toma de decisiones dentro de una microempresa del sector servicios, mejorando su eficiencia y control general.

### **2. Análisis del Modelo de Negocio**

Actualmente, muchas microempresas utilizan métodos manuales (hojas de cálculo, documentos de texto, agendas) para la gestión de sus operaciones diarias. Esta situación presenta los siguientes problemas recurrentes:

- Ineficiencia: Dificultad para acceder rápidamente a la información consolidada de clientes, proyectos y finanzas.
- Duplicidad de Datos: Alto riesgo de inconsistencia, errores humanos y registros desactualizados.
- Falta de Control y Visibilidad: Complicaciones para generar informes, tener una visión clara del estado de los proyectos, controlar los gastos y dar seguimiento a las tareas pendientes.

El sistema propuesto resuelve estos problemas al centralizar toda la información en una única base de datos, permitiendo un acceso rápido y seguro, reduciendo errores y habilitando la capacidad de generar reportes de forma automatizada.

### **3. Requisitos del Sistema**

#### **3.1. Requisitos Funcionales (Nuevas Funciones Incluidas)**

- Gestión de Clientes:
  - ABMC (Alta, Baja, Modificación y Consulta): Creación, lectura, actualización y eliminación de registros de clientes.
  - Filtrado y Búsqueda: Búsqueda dinámica de clientes por nombre o empresa.
  - Exportación: Exportación de la lista de clientes a formatos CSV y PDF.
  - Gestión de Estado: Clasificación de clientes como "Activo", "Inactivo" o "Potencial".
- Gestión de Empleados:
  - ABMC: Registro, actualización y eliminación de los datos de los empleados (nombre, cargo, email, teléfono).
- Gestión de Proyectos:
  - ABMC: Creación, lectura, actualización y eliminación de proyectos.
  - Asociación: Vinculación obligatoria de cada proyecto a un cliente existente.
  - Gestión de Estado: Selección de estado desde una lista predefinida ("En curso", "Finalizado", "Cancelado").
- Gestión de Tareas:
  - ABMC: Creación, actualización y eliminación de tareas.
  - Asociación y Asignación: Vinculación de cada tarea a un proyecto y asignación a un empleado responsable.
  - Seguimiento: Registro del estado de la tarea ("Pendiente", "En progreso", "Completada").
- Módulo Financiero:
  - Registro de Gastos: Inclusión de gastos detallados (descripción, monto, fecha) asociados a un proyecto específico.
  - Gestión de Facturas: Creación y eliminación de facturas con código único, asociadas a un

proyecto.

- Registro de Horas:
  - Control de Tiempo: Registro de las horas trabajadas por cada empleado en tareas específicas, con fecha y descripción.
- Gestión de Usuarios y Roles:
  - Control de Acceso: Sistema de login para autenticar a los usuarios.
  - Roles y Permisos: Definición de tres roles ("Administrador", "Gestor", "Empleado") con diferentes niveles de acceso a los módulos del sistema.
  - Cierre de Sesión: Funcionalidad para cerrar la sesión actual y permitir el ingreso de otro usuario.
- Dashboard e Informes:
  - Dashboard Principal: Visualización de métricas clave en tiempo real: proyectos activos, facturas pendientes, tareas por vencer e ingresos del mes.
  - Notificaciones: Alertas automáticas sobre tareas próximas a vencer.

### 3.2. Requisitos no Funcionales

- Usabilidad: Interfaz gráfica de usuario (GUI) intuitiva, con un diseño limpio, organizado en pestañas para facilitar la navegación.
- Portabilidad: La aplicación, desarrollada en Java, es compatible con cualquier sistema operativo que soporte Java (Windows, macOS, Linux).
- Seguridad: Manejo controlado de errores y excepciones, mostrando mensajes claros al usuario sin interrumpir la ejecución del programa.
- Eficiencia: Operaciones de lectura y escritura en la base de datos optimizadas para ser rápidas y no bloquear la interfaz de usuario, usando SwingWorker para tareas en segundo plano.

## 4. Arquitectura y Metodología de Desarrollo

Se utiliza una arquitectura de tres capas para el desarrollo del sistema:

- Capa de Presentación (Paquete gui): Implementada en Java con la librería Swing. Se encarga de la interfaz gráfica y la interacción con el usuario.
- Capa de Lógica de Negocio/Datos (Paquete dao): Clases DAO (Data Access Object) que encapsulan la lógica para interactuar con la base de datos (consultas SQL).
- Capa de Dominio (Paquete modelo): Clases que representan las entidades del negocio (Cliente, Proyecto, Tarea, etc.).

La metodología de desarrollo es el Proceso Unificado de Desarrollo (PUD), con un enfoque en la iteración y la entrega incremental, asegurando un desarrollo continuo y de alta calidad.

## 5. Diseño de la Base de Datos

La base de datos se llama `gestion_empresa` y se compone de las siguientes tablas relacionales:

- `clientes`: Almacena la información de los clientes.
- `empleados`: Guarda los datos de los empleados de la empresa.
- `proyectos`: Contiene los detalles de los proyectos, vinculados a un cliente.
- `tareas`: Registra las tareas de cada proyecto, asignadas a un empleado.
- `gastos`: Almacena los gastos asociados a cada proyecto.
- `registros_horas`: Lleva un registro de las horas trabajadas por los empleados en cada tarea.
- `facturas`: Guarda la información de las facturas emitidas por proyecto.
- `usuarios`: Almacena las credenciales y roles de los usuarios del sistema.

## 6. Nuevos Casos de Uso

### Caso de Uso 1: Asignar Tarea a Empleado

- Actor: Usuario del sistema (Administrador, Gestor).
- Descripción: El usuario crea una nueva tarea y la asigna a un empleado dentro de un proyecto.
- Precondición: Deben existir proyectos y empleados registrados en el sistema.
- Flujo de Eventos:
  1. El usuario se dirige a la pestaña "Tareas".
  2. Selecciona un proyecto y un empleado de las listas desplegables.
  3. Completa la descripción y la fecha límite de la tarea.
  4. Hace clic en "Agregar Tarea".
  5. El sistema valida que todos los campos obligatorios estén completos.
  6. Si la validación es correcta, se inserta el nuevo registro en la tabla tareas.
  7. Se muestra un mensaje de confirmación.
- Postcondición: La nueva tarea aparece en la lista general y está visible para el empleado asignado.

### Caso de Uso 2: Registrar Horas en una Tarea

- Actor: Usuario del sistema (Empleado, Gestor, Administrador).
- Descripción: Un usuario registra el tiempo invertido en una tarea específica.
- Precondición: El usuario ha iniciado sesión y existen tareas asignadas.
- Flujo de Eventos:
  1. El usuario navega a la pestaña "Registro de Horas".
  2. Selecciona la tarea correspondiente de la lista.
  3. Selecciona su nombre de empleado.
  4. Ingresa el número de horas trabajadas y una descripción opcional.
  5. Hace clic en "Registrar Horas".
- Postcondición: El registro de horas queda asociado a la tarea y es visible en el historial.

### Caso de Uso 3: Exportar Lista de Clientes a PDF

- Actor: Usuario del sistema (Administrador, Gestor).
- Descripción: El usuario exporta la lista completa de clientes a un archivo en formato PDF para informes externos.
- Precondición: La aplicación está en ejecución y hay clientes registrados en la base de datos.
- Flujo de Eventos:
  1. El usuario va a la pestaña "Clientes".
  2. Hace clic en el botón "Exportar a PDF" en la sección de "Opciones de Exportación".
  3. El sistema abre un cuadro de diálogo para que el usuario elija la ubicación y el nombre del archivo.
  4. Una vez confirmada la ubicación, el sistema genera el archivo PDF con los datos de la tabla de clientes.
  5. Se muestra un mensaje de éxito.
- Postcondición: Se crea un archivo PDF con la lista de clientes en la ubicación seleccionada por el usuario.

## 7. Código Fuente

El código fuente del sistema está modularizado en varios paquetes (gui, dao, modelo, util) para una mejor organización y mantenimiento. El archivo principal que inicia la aplicación es SistemaGestion.java, el cual se encarga de la conexión a la base de datos, la inicialización de la interfaz principal y el manejo del login de usuarios.