

# Variables de Clase Principales y Flujo de Trabajo

## Clase Principal: ImputacionesApp

### Variables de Control de Tiempo

- `_running` (bool): Controla si el temporizador está activo
- `_paused` (bool): Controla si el temporizador está pausado
- `elapsed_time` (int): Tiempo transcurrido en segundos
- `timer_id`: ID del callback de tkinter para cancelar el temporizador

### Variables de Datos

- `register_dic` (dict): Diccionario con información del registro actual
  - Incluye: empresa, cif, concepto, descripcion, fecha\_creacion, id, vinculada
- `empresas_dic` (dict): Diccionario empresa → CIF de todas las empresas disponibles
- `empresas_suasor` (dict): Diccionario empresa → CIF solo de empresas en SUASOR

### Variables de Instancias de Gestores

- `sql_server_manager`: Gestiona conexión y operaciones con SQL Server
- `db_manager`: Gestiona operaciones con SQLite local
- `session`: Gestiona información de usuario y sesión
- `systray`: Gestiona el ícono de bandeja del sistema
- `tasks_admin`: Gestiona la tabla de registros (TreeView)
- `search_frame`: Gestiona la búsqueda y selección de empresas

---

## Flujo de Trabajo Principal

### 1. INICIALIZACIÓN

```
main() → load_heavy_modules() → ImputacionesApp.__init__()
├─ verificar_o_crear_carpeta_archivos()
├─ SQLServerManager() → conectar a SQL Server
├─ DatabaseManager() → sincronizar empleados y empresas
├─ SessionManager() → cargar configuración de usuario
└─ SystrayManager() → crear ícono de bandeja
```

### 2. ESTADOS DE LA APLICACIÓN

## Estado: Sin Usuario

- `session.logged_in = False`
- `user_label = "Usuario no logado"`
- Mostrar popup de login

## Estado: Usuario Logueado

- `session.logged_in = True`
- `session.user` = nombre del usuario
- `session.department` = departamento del usuario
- Cargar `empresas_dic` y `empresas_suasor`
- Mostrar interfaz principal

## Estado: Empresa Seleccionada

- `register_dic` contiene información de la empresa
- `register_dic["vinculada"]` indica si la empresa existe en SUASOR
- Habilitar campos de concepto y descripción
- Mostrar botón de Play

## Estado: Imputando Tiempo

- `_running = True`, `_paused = False`
- `elapsed_time` se incrementa cada segundo
- Deshabilitar selección de empresa
- Actualizar base de datos cada 30 segundos

## Estado: Pausado

- `_running = True`, `_paused = True`
- Temporizador parpadea
- Mantener tiempo actual sin incrementar

---

## Clase SessionManager

### Variables Principales

- `user` (str): Nombre del usuario actual
- `user_id` (int): ID del usuario en la base de datos

- `department` (str): Departamento del usuario
- `empleados_dict` (dict): Diccionario nombre → ID de empleados
- `empleados_df` (DataFrame): DataFrame con información de empleados
- `empresa_no_creada` (str): Constante para empresas temporales

## Métodos Clave

- `logged_in` (property): Evalúa si hay usuario activo
  - `selected_user()`: Establece usuario y guarda configuración
  - `return_empresas_combo_values()`: Devuelve empresas disponibles
- 



## Clase TreeviewManager

### Variables de Control

- `interaccion_treeview` (bool): Habilita/deshabilita interacción con la tabla
- `seleccionado`: ID del registro seleccionado
- `orden_actual` (dict): Estado de ordenación por columnas
- `tree`: Widget TreeView de tkinter
- `bottom_frame`: Frame con botones de acción (borrar, imputar)

### Flujo de Interacción

1. Click izquierdo → `click_izquierdo()` → Toggle checkbox o recuperar registro
  2. Click derecho → `mostrar_menu_contextual()` → Menú editar/borrar/imputar
  3. Checkbox marcado → `actualizar_estado_frame()` → Mostrar botones de acción
- 



## Clase DatabaseManager (SQLite)

### Variables Principales

- `db_path`: Ruta del archivo SQLite
- `conexion`: Conexión a SQLite
- `cursor`: Cursor para ejecutar consultas
- `columnas`: Definición de columnas de la tabla registros

### Operaciones Principales

- `sincronizar_empleados/empresas()`: Sync desde SQL Server

- `agregar_registro()`: Crear nuevo registro
  - `actualizar_registro()`: Actualizar tiempo y datos
  - `obtener_registros_imputando()`: Registros pendientes de subir
- 

## Clase **SQLServerManager**

### **Variables de Conexión**

- `connection`: Conexión activa a SQL Server
- `cursor`: Cursor para ejecutar consultas
- `servidor_activo`: Servidor que está funcionando
- `conectado` (bool): Estado de la conexión

### **Sistema de Fallback**

Prueba múltiples servidores según configuración:

1. Servidor principal (dominio)
  2. Servidor de respaldo (IP)
  3. Servidores adicionales
- 

## **Flujo de Datos Completo**

### **Crear Nuevo Registro**

1. Seleccionar empresa → `register_dic` creado
2. `start_stop_timer()` → `nuevo_registro()`
3. `DatabaseManager.agregar_registro()` → SQLite
4. `TreeviewManager.agregar_fila()` → UI actualizada
5. Temporizador inicia → `update_timer()` cada segundo

### **Imputar Registros**

1. Marcar checkboxes → seleccionar registros
2. `imputar()` → validar empresas vinculadas
3. `set_imputacion()` → estado "imputando" en SQLite
4. `SQLServerManager.subir_registros()` → sync a SQL Server
5. Estado "imputado" → registro completado

### **Gestión de Estados GUI**

set\_logout\_state() → Sin usuario  
set\_login\_state() → Usuario logueado  
set\_empresa\_seleccionada\_state() → Empresa activa  
set\_play\_state() → Imputando tiempo  
set\_pause\_state() → Pausado  
set\_detener\_state() → Entre tareas

---

## Variables Críticas por Funcionalidad

### Control de Tiempo

- `_running`: Determina si mostrar botón Stop vs Play
- `_paused`: Determina si incrementar tiempo o parpadear
- `elapsed_time`: Valor mostrado en pantalla y guardado en BD

### Gestión de Empresa

- `register_dic["vinculada"]`: Determina si se puede imputar directamente
- `empresas_dic` vs `empresas_suasor`: Diferencia empresas temporales de reales

### Sincronización de Datos

- `state` en SQLite: "working" → "imputando" → "imputado"
- `sql_server_manager.conectado`: Determina funcionalidad disponible

### Interfaz de Usuario

- `interaccion_treeview`: Habilita/deshabilita selección de registros
- `seleccionado`: Registro activo para edición/borrado
- `bottom_frame.visible`: Muestra botones según checkboxes marcados