



Dirección de Sistemas MSCB

Planificación de desarrollo de Software: Sistema de Gestión del Cementerio MSCB

El desarrollo de este proyecto se realizará en cuatro etapas:

1. Etapa 1

1.1 Relevamiento y análisis de requerimientos

1.2 Alcances del proyecto

1.2.1 Límites del proyecto

1.2.2 Restricciones y dependencias

1.2.3 Gestión de expectativas

2. Etapa 2: Funcionamiento del sistema y requisitos específicos

2.1 Requerimientos funcionales

2.2 Requerimientos no funcionales

2.3 Requisitos de interfaz externa

2.4 Interfaz de usuario y casos de uso

3. Etapa 3: Desarrollo del Software

3.1 Diseño del sistema

3.2 Configuración del entorno de desarrollo

3.3 Definir los componentes de software

3.4 Desarrollo del frontend (HTML + CSS + JS)

3.5 Desarrollo del backend (PHP + MySQL)

4. Etapa 4: pruebas e implementación

4.1 Implementación a nivel de software

4.2 Pruebas y análisis en tiempo de ejecución

4.3 Despliegue

4.4 Mantenimiento del software y posibles mejoras

1. Etapa 1

1.1 Relevamiento y análisis de requerimientos

- Entrevistar al personal del cementerio municipal.
- Identificar los procesos actuales: gestión de difuntos, parcelas, nichos, pagos, vencimientos, traslados, etc.
- Recolectar requerimientos funcionales y no funcionales.
- Análisis del software existente: desarrollado en VisualFoxPro, año 1996.

Documentos a realizar:

- Documento de Requisitos Funcionales.
- Casos de uso.
- Diagramas esquemáticos.

1.2 Alcances del proyecto

Esta aplicación web brindará una plataforma moderna para la gestión cotidiana del cementerio municipal, mejorando la atención mediante la automatización de procesos básicos.

1.2.1 Límites del proyecto

- Implementación de una aplicación web enfocada en la carga y consulta de datos del cementerio.
- La funcionalidad se limitará a procesos internos como: registro de difuntos, asignación de parcelas, búsqueda y generación de reportes, historial de ubicación por difunto e historial de pago.

1.2.2 Restricciones y dependencias

- Requiere estabilidad en la conexión de red.
- Depende del correcto funcionamiento de los dispositivos de acceso (PCs municipales).
- El desarrollo estará limitado al stack actual: HTML, CSS, JavaScript, PHP (vs 8.1.0), MySQL(vs 5.7.36) y DBeaver.

1.2.3 Gestión de expectativas

El proyecto está planteado como una base modular que puede extenderse en el futuro para incluir funcionalidades adicionales, como:

- Integración con sistemas externos (ej. planilla de excel).
- Incorporación de notificaciones (ej. de pagos próximos).
- Mejoras en la interfaz de usuario (accesibilidad a dispositivo móvil).
- Envío de recibos de pago o documentación específica a través de correo electrónico.

2. Etapa 2: Funcionamiento del sistema y requisitos específicos

Se busca la escalabilidad y modularidad del sistema.

2.1 Requerimientos funcionales

- Iniciar y cerrar sesión de usuarios con diferentes permisos.
- Gestión de usuarios del sistema.

- Registrar un nuevo difunto en la base de datos.
- Buscar un difunto por nombre y fecha.
- Asignar y registrar en un historial la ubicación de la parcela y sus actualizaciones.
- Asignaciones e historial de pagos y vencimientos.

2.2 Requerimientos no funcionales

- **Rendimiento:** el sistema debe estar disponible durante el horario laboral.
- **Seguridad:** control de acceso por niveles, protección de datos personales.
- **Fiabilidad:** manejo de errores y registros de auditoría.
- **Usabilidad:** interfaz intuitiva y accesible.
- **Escalabilidad:** preparado para futuras funciones y más usuarios.

2.3 Requisitos de interfaz externa

- **Interfaz de usuario:** formularios, tablas, reportes, menús.
- **Interfaz de software:** conexión con base de datos MySQL, posibles APIs a futuro.

2.4 Interfaz de usuario y casos de uso

- Representación de los distintos escenarios de uso del sistema.
- Interacción de los usuarios mediante formularios, listados y acciones.
- Casos de uso que detallan cada funcionalidad desde la perspectiva del usuario final. Por ejemplo, el ingreso de datos a partir de las actas de defunción y ficha de pago.

3. Etapas 3: Desarrollo del Software

3.1 Diseño del sistema

- **Modelo de base de datos relacional:** entidades como *Difuntos*, *Parcelas*, *Usuarios*, *Contratos*, *Familiares*, *Pagos*.

- **Arquitectura de la aplicación:** Arquitectura centralizada y basada en servicios con PHP. Con especificación de SO Windows. Separación de lógica, presentación y datos.
- **Diseño de interfaces:** formularios, listados, filtros, reportes.
- **Seguridad y autenticación.** Control de sesiones (login/acceso a la página web).

Documentos a realizar:

- Wireframes (diagrama visual) de pantallas clave (o servicios a ofrecer).
- Estructura de carpetas del proyecto.
- Diagrama entidad-interrelación de Bases de Datos.
 - Diseño conceptual
 - Diseño lógico
 - Diseño físico

3.2 Configuración del entorno de desarrollo

- Apache + PHP + MySQL (WampServer o similar para local).
- Git para control de versiones.
- Estructura inicial del repositorio (basada en el modelo de proyectos realizados en la dirección de sistemas).

3.3 Definir los componentes de software

- Definición de controladores, modelos y vistas.
- Módulos principales: login, gestión de difuntos (ingresos, modificaciones, actualizaciones y borrado de datos), parcelas, reportes, usuarios.

3.4 Desarrollo del frontend (HTML + CSS + JS)

- Formularios accesibles, intuitivos y responsivos.
- Uso de librerías de estilos (ej: Bootstrap).
- JS para validaciones, alertas, autocompletado, AJAX, para mejorar la experiencia del usuario.

3.5 Desarrollo del backend (PHP + MySQL)

- Creación de la base de datos SQL.
- Conexión segura a la base de datos.
- CRUD (crear, leer, actualizar y borrar) de las entidades principales.
- Sistema de login: Gestión de sesiones y roles de usuario.
- Validaciones y control de errores del lado del servidor.
- Migración de datos (desde el sistema anterior).

4. Etapa 4: pruebas e implementación

4.1 Implementación a nivel de software

- Instalación del sistema en entorno de prueba y producción.
- Configuración de la base de datos en el servidor.

4.2 Pruebas y análisis en tiempo de ejecución

- Pruebas funcionales de cada módulo.
- Pruebas con datos reales (importación del sistema anterior).
- Pruebas de usuario (feedback del personal del municipio).
- Documentación de errores encontrados y ajustes realizados.

4.3 Despliegue y capacitación

- Servidor de prueba municipal.
- Migración de datos del sistema anterior al servidor municipal.
- Capacitación del personal.
- Elaborar documentos y/o manuales de especificaciones técnicas y de usuario.

4.4 Mantenimiento del software y posibles mejoras

- Corrección de errores según el uso diario del sistema.
- Incorporación de nuevas funcionalidades solicitadas.
- Optimización de consultas SQL y mejoras de rendimiento.