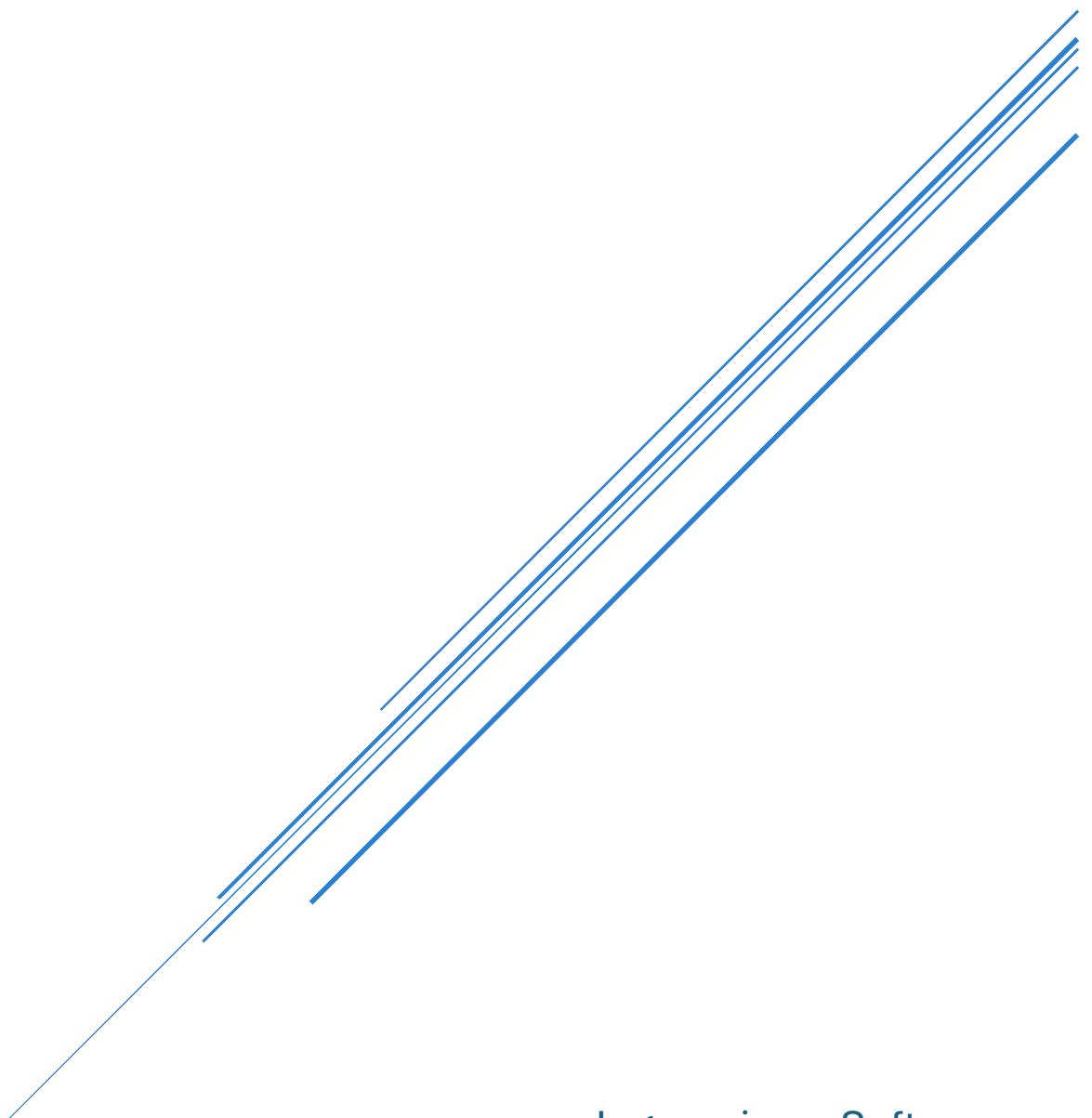


SOFTWARE DE GESTION DE PERSONAL

Escritorio y móvil



Ingeneria en Software
Maestra. Yadira Fleitas

1. Documento de Visión / Requerimientos

1.1 Objetivo del Sistema

Objetivo del Sistema

El Sistema de Gestión de Empleados constituye una solución integral diseñada para administrar el ciclo completo de información laboral del personal de la organización. El sistema facilita el registro

gestión y consulta de datos de empleados, con especial énfasis en el control preciso de asistencia, incluyendo el monitoreo de entradas, salidas y cálculo de horas laboradas efectivas.

Desarrollado como una aplicación completamente autónoma con capacidad de operación local —sin dependencia de conectividad a internet—, el sistema proporciona funcionalidades avanzadas para

la generación de reportes detallados sobre jornadas laborales, análisis de desempeño y simulaciones de nómina.

Esta solución tecnológica tiene como propósito optimizar los procesos administrativos de recursos humanos, garantizar la trazabilidad completa de la asistencia del personal y proporcionar información

confiable para la toma de decisiones estratégicas, mediante un seguimiento eficiente y sistemático de la actividad laboral.

1.2 Usuarios del Sistema

-Empleado

*Registro de entrada y salida

*Consulta de historial personal

*Visualización de horas trabajadas y salario estimado

- Administrador / Supervisor (versión de escritorio en Python)
- *Gestión de empleados (alta, edición y eliminación)
- *Consulta de reportes y generación de PDFs
- *Control general del personal

1.3 Requerimientos Funcionales

- RF-01 Registrar empleados (alta, edición y eliminación)
- RF-02 Registrar entradas y salidas de empleados
- RF-03 Calcular automáticamente horas trabajadas por día
- RF-04 Generar resumen semanal y mensual
- RF-05 Simular nómina quincenal a partir de horas trabajadas
- RF-06 Visualizar historial de registros
- RF-07 Mostrar gráficas de horas trabajadas
- RF-08 Autenticación básica mediante PIN

1.4 Requerimientos No Funcionales

- RNF-01: El sistema debe funcionar sin conexión a internet
- RNF-02: La aplicación móvil debe ser compatible con Android 8.0 o superior
- RNF-03: Persistencia local usando base de datos
- RNF-04: Interfaz intuitiva y fácil de usar
- RNF-05: Tiempo de respuesta menor a 1 segundo en operaciones comunes

1.5 Cronograma Final de Actividades (Planeado vs Real)

Sprint	Planeado	Real	Actividades
Sprint 1	3 dias	4 dias	Análisis, definición de requerimientos
Sprint 2	3 dias	3 dias	Desarrollo sistema Python (GUI + MongoDB)
Sprint 3	3 dias	3 dias	Resúmenes, gráficas, UI/UX y documentación
Sprint 4	3 dias	5 dias	App Android (registro y base de datos local)

1.6 Alcance Final del Producto

- ✓ Sistema de escritorio funcional en Python
- ✓ Aplicación móvil Android en Kotlin + Jetpack Compose
- ✓ Control de asistencia y cálculo de horas
- ✓ Resúmenes semanales y mensuales
- ✓ Simulación de nómina quincenal
- ✓ Funcionamiento offline

1.7 Costo Total del Desarrollo (Estimado)

Concepto	Horas	Costo estimado
Análisis y diseño	40 h	\$6,000 MXN
Desarrollo Python	60 h	\$9,000 MXN
Desarrollo Android	70 h	\$10,500 MXN
Pruebas y documentación	30 h	\$4,500 MXN
Total	200 h	\$30,000 MXN

2. Arquitectura del Sistema

2.1 Descripción General

El sistema implementa una arquitectura multicapa distribuida, compuesta por dos aplicaciones cliente independientes que operan de manera autónoma, cada una con su propia capa de persistencia local.

Esta arquitectura garantiza la disponibilidad del sistema sin dependencia de infraestructura de red o servicios en la nube.

2.2 Componentes Principales

El ecosistema del sistema está conformado por los siguientes componentes:

-----Aplicación de Escritorio

Tecnologías: Python + Tkinter

Orientada a funciones administrativas y de gestión

Base de datos local independiente

-----Aplicación Móvil

Tecnologías: Android + Kotlin

Enfocada en registro de asistencia y consultas en campo

Base de datos local independiente

2.3 Modelo Arquitectónico por Capas

El sistema adopta un patrón arquitectónico de tres capas, aplicado de manera consistente en ambas plataformas:

Capa de Presentación (UI/UX)

Gestiona la interfaz de usuario y la interacción con el sistema

Captura eventos y presenta información al usuario final

Implementación específica por plataforma (Tkinter para escritorio, Android SDK para móvil)

Capa de Lógica de Negocio

Contiene las reglas de negocio y procesos centrales del sistema

Coordina el flujo de información entre la presentación y la persistencia

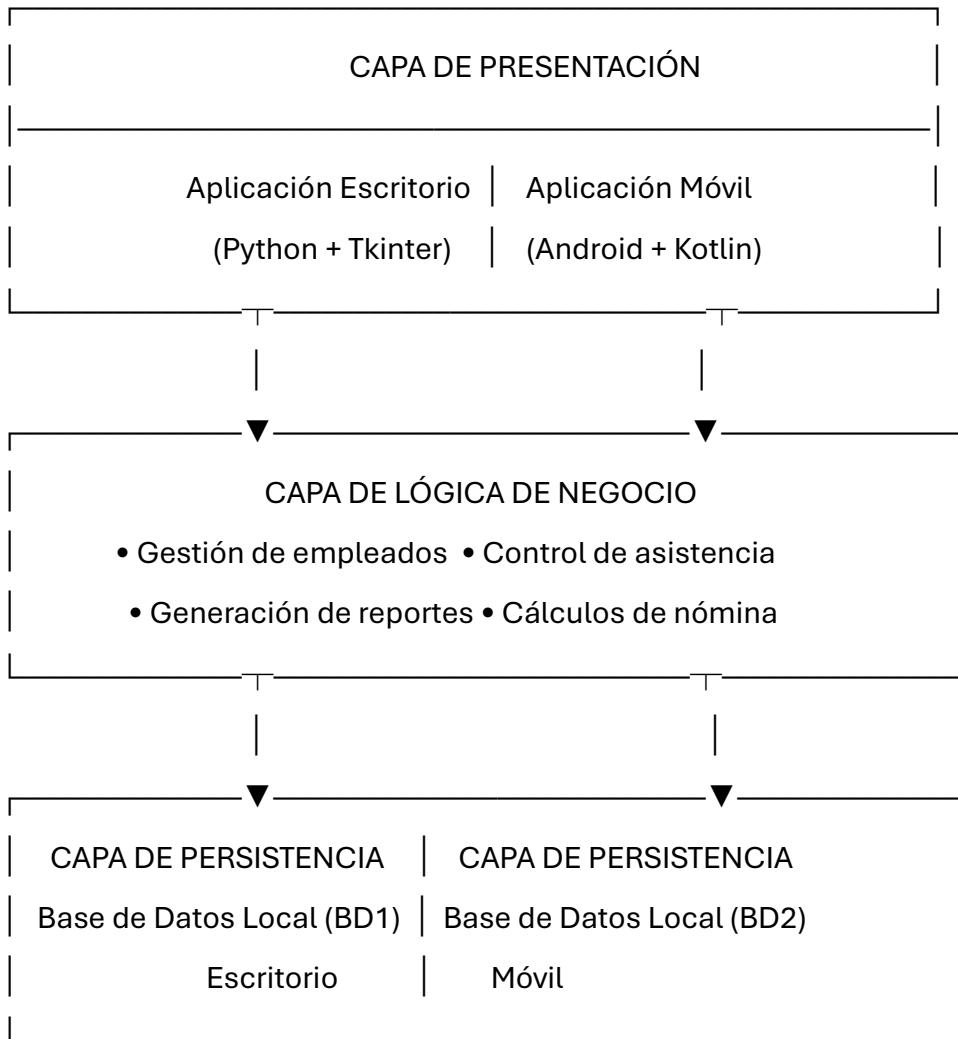
Ejecuta validaciones, cálculos y operaciones de transformación de datos

Capa de Persistencia

Gestiona el acceso y almacenamiento de datos

Abstactae las operaciones de base de datos

Garantiza la integridad y consistencia de la información



2.5 Características Arquitectónicas

Independencia de Plataformas

Cada aplicación opera de manera autónoma con su propio repositorio de datos

No existe dependencia de sincronización en tiempo real

Operación Offline

Funcionalidad completa sin requerimientos de conectividad

Todos los datos se gestionan localmente y tener Escalabilidad Modular

La arquitectura por capas facilita el mantenimiento y evolución del sistema

Posibilidad de incorporar nuevas funcionalidades sin afectar componentes existentes

2.6 Tecnologías Utilizadas

El sistema integra un stack tecnológico robusto y moderno, seleccionado estratégicamente para garantizar rendimiento, mantenibilidad y experiencia de usuario óptima en ambas plataformas.

2.6.1 Aplicación de Escritorio

Lenguaje de Programación

Python 3: Lenguaje de alto nivel que proporciona flexibilidad, amplio ecosistema de librerías y facilidad de mantenimiento para el desarrollo de aplicaciones empresariales.

Framework de Interfaz Gráfica

Tkinter: Biblioteca estándar de Python para desarrollo de interfaces gráficas de usuario, que ofrece portabilidad multiplataforma y bajo consumo de recursos.

Sistema de Persistencia

MongoDB: Base de datos NoSQL orientada a documentos que proporciona flexibilidad en el esquema de datos, alto rendimiento en operaciones locales y escalabilidad horizontal.

Generación de Reportes

ReportLab: Biblioteca especializada para la generación programática de documentos PDF con capacidades avanzadas de diseño y formato, ideal para reportes empresariales.

2.6.2 Aplicación Móvil

Lenguaje de Programación

Kotlin: Lenguaje oficial para desarrollo Android, moderno y conciso, que ofrece seguridad de tipos, interoperabilidad con Java y sintaxis expresiva.

Framework de Interfaz de Usuario

Jetpack Compose: Toolkit declarativo moderno para construcción de interfaces nativas Android, que simplifica el desarrollo UI mediante programación reactiva y reduce significativamente el código boilerplate.

Sistema de Persistencia

Room: Biblioteca de abstracción sobre SQLite que proporciona una capa de acceso a datos robusta con validación en tiempo de compilación, mapeo objeto-relacional y soporte para operaciones asíncronas.

Diseño de Interfaz

Material Design 3: Sistema de diseño de Google que garantiza consistencia visual, accesibilidad y adherencia a las mejores prácticas de UX/UI en la plataforma Android.

2.6.3 Justificación Técnica

La selección de estas tecnologías responde a criterios específicos:

Madurez y Estabilidad: Todas las tecnologías seleccionadas cuentan con comunidades activas, documentación exhaustiva y soporte a largo plazo.

Rendimiento Local: Tanto MongoDB como Room están optimizados para operaciones de lectura/escritura en entornos locales sin latencia de red.

Productividad de Desarrollo: Las herramientas modernas como Jetpack Compose y las capacidades de Python aceleran el ciclo de desarrollo y reducen errores.

Experiencia de Usuario: Material Design 3 y las capacidades gráficas de Tkinter/ReportLab aseguran interfaces intuitivas y documentos profesionales.

2.7 Diseño de la base de datos

Entidad: Empleado

Almacena la información completa del personal de la organización, incluyendo datos personales, laborales y jerárquicos.

Campo	Tipo	Descripción	Restricciones
id	String/ObjectId	Identificador único del empleado	PK, Obligatorio, Único
nombre	String	Nombre completo del empleado	Obligatorio, 3-100 caracteres
dpi	String	Documento Personal de Identificación	Obligatorio, Único, 13 dígitos
cargo	String	Puesto o posición laboral	Obligatorio
departamento	String	Área o departamento de adscripción	Obligatorio
salario_hora	Decimal	Tarifa por hora trabajada	Obligatorio, > 0
tipo_empleado	Enum	Categoría del empleado	Valores: OPERATIVO, SUPERVISOR, JEFE_AREA, RECURSOS_HUMANOS, ADMINISTRATIVO

Campo	Tipo	Descripción	Restricciones
supervisor_id	String/ObjectId	ID del supervisor directo	FK → Empleado(id), Nullable
fecha_ingreso	Date	Fecha de inicio de relación laboral	Obligatorio
estado	Enum	Estado actual del empleado	Valores: ACTIVO, INACTIVO, SUSPENDIDO, VACACIONES
email	String	Correo electrónico corporativo	Opcional, Único si existe
telefono	String	Número de contacto	Opcional
fecha_creacion	DateTime	Timestamp de creación del registro	Auto-generado
fecha_modificacion	DateTime	Timestamp de última actualización	Auto-actualizado

Entidad: Registro

Captura cada evento de entrada o salida del personal, permitiendo el cálculo preciso de horas trabajadas y generación de reportes de asistencia.

Campo	Tipo	Descripción	Restricciones
id	String/ObjectId	Identificador único del registro	PK, Obligatorio, Único
empleado_id	String/ObjectId	ID del empleado asociado	FK → Empleado(id), Obligatorio

Campo	Tipo	Descripción	Restricciones
tipo	Enum	Tipo de marcación	Valores: ENTRADA, SALIDA; Obligatorio
fecha_hora	DateTime	Timestamp exacto de la marcación	Obligatorio, Indexado
ubicacion	String	Punto de registro (terminal/dispositivo)	Opcional
metodo_registro	Enum	Medio utilizado para el registro	Valores: MANUAL, BIOMETRICO, MOVIL, WEB
observaciones	String	Notas o comentarios adicionales	Opcional, Max 500 caracteres
aprobado_por	String/ObjectId	ID del supervisor que aprobó	FK → Empleado(id), Nullable
estado_registro	Enum	Estado de validación	Valores: PENDIENTE, APROBADO, RECHAZADO, CORREGIDO
latitud	Decimal	Coordenada geográfica (si aplica)	Opcional, -90 a 90
longitud	Decimal	Coordenada geográfica (si aplica)	Opcional, -180 a 180
fecha_creacion	DateTime	Timestamp de creación del registro	Auto-generado

3. Interfaz de Usuario y Experiencia (UI/UX)

3.1 Descripción de Wireframes

El sistema presenta una arquitectura de navegación intuitiva y minimalista, diseñada para maximizar la eficiencia operativa y reducir la curva de aprendizaje del usuario.

Pantalla de Autenticación (Login)

Propósito: Validación segura de identidad mediante PIN numérico.

Elementos principales:

- Campo de entrada numérico para PIN (4-6 dígitos)
- Teclado numérico visual para dispositivos táctiles
- Indicador de estado de autenticación
- Opción de recuperación de acceso (solo para usuarios autorizados)
- Selector de idioma (español/inglés)

Consideraciones de seguridad:

- Ocultamiento de caracteres durante la entrada
- Bloqueo temporal tras 3 intentos fallidos
- Registro de intentos de acceso en log del sistema

Menú Principal (Dashboard)

Propósito: Hub central de navegación con información contextual del usuario.

Elementos principales:

- Tarjeta de perfil con información del empleado (nombre, cargo, foto)
- Estado actual de jornada (dentro/fuera, última marcación)
- Accesos rápidos a funciones principales mediante cards
- Notificaciones y alertas pendientes
- Indicador de sincronización (si aplica)

Navegación primaria:

- Registro de Entrada/Salida (acción destacada)
 - Historial de Asistencia
 - Resumen de Horas y Nómina
 - Configuración/Perfil
 - Cerrar Sesión
-

Pantalla de Registro de Entrada/Salida

Propósito: Captura rápida y eficiente de eventos de asistencia.

Elementos principales:

- Reloj digital con fecha y hora actual en tiempo real
- Botón principal contextual (REGISTRAR ENTRADA o REGISTRAR SALIDA)
- Visualización del último registro realizado
- Indicador de estado de conexión/sincronización
- Campo opcional para observaciones
- Confirmación visual tras registro exitoso

Flujo de interacción:

1. Usuario visualiza estado actual (última marcación)
 2. Presiona botón de acción correspondiente
 3. Sistema valida y registra con timestamp
 4. Feedback visual inmediato (animación + mensaje)
 5. Actualización automática del dashboard
-

Pantalla de Historial de Registros

Propósito: Consulta y auditoría de asistencia histórica.

Elementos principales:

- Filtros por rango de fechas (día, semana, mes, personalizado)
- Lista cronológica inversa de registros
- Cards individuales por jornada mostrando:
 - Fecha y día de la semana
 - Hora de entrada y salida
 - Total de horas trabajadas
 - Estado (completa, incompleta, pendiente)
- Indicadores visuales por tipo de registro (iconos diferenciados)
- Función de búsqueda por fecha específica
- Exportación a PDF (según permisos)

Indicadores de estado:

- ✓ Jornada completa (verde)
 - △ Jornada incompleta (amarillo)
 - X Sin registros (rojo)
 - ⚡ Horas extras (azul)
-

Pantalla de Resumen de Horas y Salario

Propósito: Visualización analítica del desempeño y proyección salarial.

Elementos principales:

- Selector de período (semanal, quincenal, mensual)
- Tarjetas de métricas clave:
 - Total de horas trabajadas
 - Promedio diario
 - Días laborados
 - Proyección de nómina

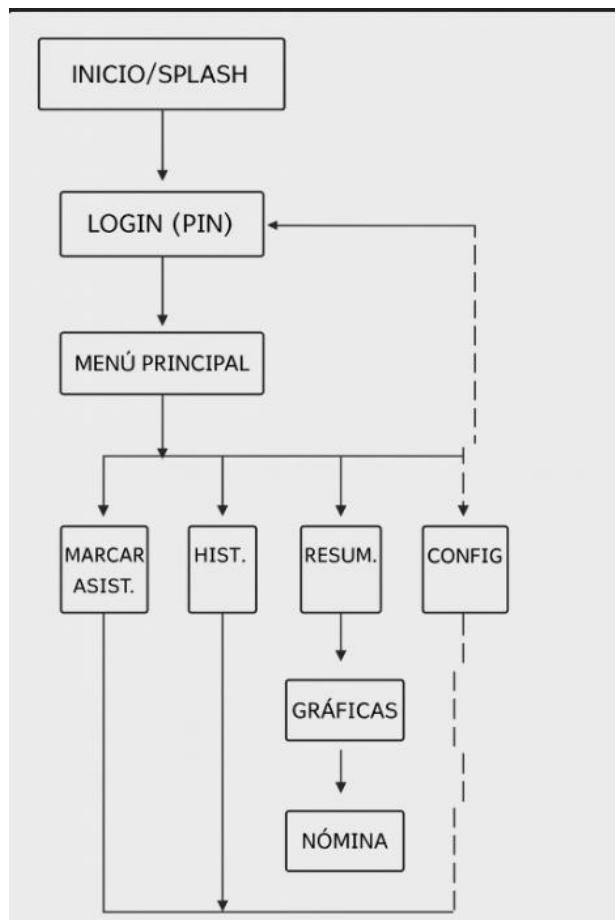
- Gráfica de barras de horas por día
- Gráfica de tendencia semanal/mensual
- Desglose detallado:
 - Horas regulares
 - Horas extras
 - Ausencias
 - Deducciones (si aplica)
- Botón para generar reporte detallado

Visualización de datos:

- Gráficos interactivos con tooltips informativos
- Código de colores consistente
- Animaciones suaves al cargar datos
- Responsive para diferentes tamaños de pantalla

3.2 Flujo de Navegación del Sistema

Diagrama de Flujo Principal



Flujos Secundarios

Flujo de Registro de Asistencia:

Menú Principal → Marcar Asistencia → Confirmación Visual →

Actualización Dashboard → Retorno a Menú Principal

Flujo de Consulta Histórica:

Menú Principal → Historial → Seleccionar Filtro →

Visualizar Registros → [Opcional: Exportar PDF] →

Retorno a Menú Principal

Flujo de Análisis de Nómina:

Menú Principal → Resumen → Seleccionar Período →
Ver Gráficas → Desglose Detallado →
[Opcional: Generar Reporte] → Retorno a Menú Principal

3.3 Estilo Visual y Decisiones de Diseño

3.3.1 Sistema de Diseño Adoptado

Material Design 3 (Material You)

Justificación de selección:

- Estándar de facto para aplicaciones Android modernas
- Sistema de diseño adaptativo y accesible
- Componentes pre-construidos que aceleran desarrollo
- Soporte nativo para temas dinámicos y personalización
- Excelente documentación y comunidad activa

3.3.2 Paleta de Colores

Esquema Cromático: Neutro Profesional

Categoría	Color	Código Hex	Uso
Primario	Azul Corporativo	#1976D2	Botones principales, encabezados
Secundario	Verde Confirmación	#43A047	Estados exitosos, registros completos
Terciario	Gris Neutral	#757575	Textos secundarios, bordes
Superficie	Blanco/Gris Claro	#FFFFFF / #F5F5F5	Fondos, tarjetas

Categoría	Color	Código Hex	Uso
Error	Rojo Advertencia	#D32F2F	Errores, validaciones fallidas
Advertencia	Amarillo Atención	#FFA000	Alertas, pendientes
Información	Azul Información	#0288D1	Mensajes informativos

Principios de aplicación:

- Contraste mínimo 4.5:1 para cumplir WCAG 2.1 AA
- Uso conservador del color para evitar fatiga visual
- Código de colores consistente en toda la aplicación
- Modo oscuro disponible como opción de accesibilidad

3.3.3 Componentes y Patrones de Diseño

Sistema de Tarjetas (Cards)

Las tarjetas son el componente principal para agrupar información relacionada:

- **Elevación:** Sombra sutil (2dp-4dp) para jerarquía visual
- **Bordes:** Redondeados (8dp-12dp) para suavidad visual
- **Espaciado:** Padding interno 16dp, márgenes 8dp
- **Contenido:** Máximo 3 niveles de información por tarjeta
- **Interacción:** Efecto ripple en tarjetas interactivas

Botones y Controles

- **Botones primarios:** Filled, elevados, color primario
- **Botones secundarios:** Outlined, sin relleno
- **FAB (Floating Action Button):** Acción principal contextual
- **Altura mínima táctil:** 48dp según Material Guidelines
- **Estados visuales claros:** Normal, Pressed, Disabled, Focused

3.3.4 Tipografía

Familia tipográfica: Roboto (fuente predeterminada de Material Design)

Elemento	Estilo	Tamaño Peso	Uso
Encabezado H1	Display	32sp	Medium Títulos principales
Encabezado H2	Headline	24sp	Regular Secciones
Cuerpo Principal	Body	16sp	Regular Texto general
Cuerpo Secundario	Body	14sp	Regular Información adicional
Botones	Button	14sp	Medium Etiquetas de botones
Captions	Caption	12sp	Regular Metadatos, timestamps

Principios tipográficos:

- Jerarquía visual clara mediante tamaño y peso
- Interlineado 1.5x para legibilidad óptima
- Máximo 60-70 caracteres por línea en textos extensos
- Alineación izquierda para idiomas occidentales

3.3.5 Visualización de Datos

Gráficas y Elementos Analíticos

Gráfica de Barras (Horas Trabajadas)

- Barras verticales con colores diferenciados por día
- Eje Y con escala automática adaptativa
- Etiquetas claras en cada barra
- Grid horizontal sutil para referencia
- Animación de entrada progresiva

Gráfica de Tendencia Lineal

- Línea suavizada para visualizar patrones
- Puntos de datos interactivos con tooltips
- Área bajo la curva con transparencia sutil
- Leyendas descriptivas
- Zoom y pan para exploración detallada

Principios de visualización:

- Simplicidad sobre complejidad
- Un gráfico = una idea principal
- Colores consistentes con el sistema
- Escalas proporcionales y honestas
- Accesibilidad para usuarios con daltonismo

3.3.6 Iconografía

Sistema de Íconos: Material Symbols (Rounded variant)

- Estilo redondeado para coherencia con el diseño general
- Tamaño estándar: 24dp
- Íconos rellenos para acciones primarias
- Íconos outlined para navegación secundaria
- Consistencia semántica (mismo ícono = misma función)

Íconos clave del sistema:

-  Home: Menú principal
-  Schedule: Registro de tiempo
-  Analytics: Resúmenes y gráficas
-  History: Historial
-  Settings: Configuración

-  Logout: Cerrar sesión

3.3.7 Espaciado y Layout

Sistema de Grid de 8dp

Todo el espaciado se basa en múltiplos de 8:

- Elementos pequeños: 8dp, 16dp
- Separación media: 24dp, 32dp
- Separación grande: 48dp, 64dp

Márgenes y Padding:

- Margen lateral de pantalla: 16dp
- Padding interno de cards: 16dp
- Separación entre secciones: 24dp
- Separación entre elementos relacionados: 8dp

3.3.8 Animaciones y Transiciones

Principios de movimiento:

- Duración estándar: 200-300ms
- Curva de easing: Material standard (cubic-bezier)
- Transiciones entre pantallas: Slide + Fade
- Feedback táctil: Ripple effect
- Carga de datos: Skeleton screens o shimmer effect

Animaciones específicas:

- Entrada de tarjetas: Fade in + Slide up
- Actualización de datos: Pulse suave
- Confirmación de acciones: Check mark animado
- Estados de carga: Circular progress indicator

3.3.9 Accesibilidad

Cumplimiento de estándares:

- WCAG 2.1 nivel AA
- Soporte para lectores de pantalla
- Navegación por teclado (versión escritorio)
- Tamaños táctiles mínimos de 48dp
- Texto escalable sin pérdida de funcionalidad
- Modo alto contraste disponible

4. Bitácora de Cambios y Control de Versiones

4.1 Historial de Versiones

Versión 0.1.0 (Lanzamiento Inicial) — [Fecha: 03-dic-2025]

Estado: Stable Release

Tipo: Major Release

Características principales:

- Sistema base de gestión de empleados
- CRUD completo de registros de personal
- Modelo de datos inicial con entidades Empleado y Registro
- Aplicación de escritorio funcional (Python + Tkinter)
- Persistencia en MongoDB
- Autenticación básica mediante PIN
- Registro manual de entradas y salidas
- Interfaz administrativa para gestión de empleados

Alcance funcional:

- Registro de información básica de empleados (nombre, DPI, cargo, salario)
- Captura de eventos de entrada/salida con timestamp
- Consultas simples de historial

Limitaciones conocidas:

- Sin sincronización entre dispositivos
- Sin generación de reportes avanzados
- Sin cálculos automáticos de nómina

Versión 0.5.0 (Analytics y Nómina) — [Fecha: 10-dic-2025]

Estado: Stable Release

Tipo: Minor Release (Feature Enhancement)

Nuevas características:

- Cálculo automático de horas trabajadas por jornada
- Resúmenes estadísticos (semanal, quincenal, mensual)
- Simulación de nómina quincenal con desglose
- Gráficas interactivas de horas trabajadas
- Dashboard analítico con métricas clave
- Exportación de reportes a PDF
- Sistema de notificaciones para jornadas incompletas
- Detección automática de horas extras

Funcionalidades agregadas:

- Algoritmo de cálculo de diferencia entre entrada/salida
- Motor de generación de reportes con ReportLab
- Componentes de visualización con gráficas de barras y líneas
- Filtros avanzados por período en historial
- Preview de nómina antes de procesamiento

Mejoras de UX/UI:

- Rediseño de pantalla de resumen con cards informativas
- Animaciones suaves en transiciones
- Feedback visual mejorado en registros
- Indicadores de progreso en cálculos largos

Optimizaciones:

- Índices de base de datos para consultas de rango de fechas
- Caché de cálculos frecuentes
- Lazy loading en listas de historial extenso

Versión 1.0.0 (Expansión Móvil) — [Fecha: 8 dic- 15 dic2025]

Estado: Stable Release

Tipo: Minor Release (Feature Addition)

Nuevas características:

- Aplicación móvil nativa Android (Kotlin + Jetpack Compose)
- Operación completamente offline
- Persistencia local con Room (SQLite)
- Interfaz Material Design 3
- Sincronización manual de datos (exportación/importación)
- Registro de asistencia desde dispositivos móviles
- Geolocalización opcional en registros

Mejoras:

- Arquitectura multicapa implementada en ambas plataformas
- Validaciones mejoradas en formularios
- Performance optimizado en consultas de historial

Correcciones:

- Ajustes en validación de PIN
- Corrección de timezone en timestamps

Mejoras en manejo de errores de persistencia

4.2 Funcionalidades Agregadas y Estado Actual

Módulo de Control de Asistencia

Funcionalidad	Estado	Versión	Notas
Registro de entrada	Completado	V0.1.0	Funcionalidad base
Registro de salida	Completado	V0.1.0	Funcionalidad base
Validación de secuencia entrada/salida	Completado	v1.0.0	Previene registros duplicados
Geolocalización de registros	Completado	v1.0.0	Solo en app móvil
Registro con observaciones	Completado	V0.1.0	Campo opcional
Aprobación por supervisor	En progreso	v1.1.0	Próxima versión

Módulo de Cálculos y Analytics

Funcionalidad	Estado	Versión	Notas
Cálculo automático de horas diarias	Completado	V0.5.0	Diferencia entre entrada/salida

Funcionalidad	Estado	Versión Notas
Resumen semanal de horas	Completado	V0.5.0 Agregación por semana
Resumen mensual de horas	Completado	V0.5.0 Agregación por mes
Detección de horas extras	Completado	V0.5.0 Configurable por jornada
Simulación de nómina quincenal	Completado	V0.5.0 Basado en salario_hora
Cálculo de deducciones	Planificado v1.1.0	IGSS, ISR, etc.
Proyección de ingresos anuales	Planificado v1.1.0	Basado en tendencias

Módulo de Reportes y Visualización

Funcionalidad	Estado	Versión Notas
Gráfica de barras (horas/día)	Completado v1.0.0	Visualización semanal
Gráfica de tendencia lineal	Completado v1.0.0	Análisis temporal
Exportación a PDF	Completado v1.0.0	Reportes individuales
Reportes departamentales	Planificado v1.1.0	Para jefes de área
Dashboard ejecutivo	Planificado v1.2.0	Métricas organizacionales
Exportación a Excel	Planificado v1.2.0	Análisis avanzado

Módulo de Gestión Jerárquica

Funcionalidad	Estado	Versión Notas
Tipos de empleado	Completado v1.2.5	5 categorías definidas

Funcionalidad	Estado	Versión	Notas
Relación supervisor-empleado	Completado	v1.2.5	Auto-referencial
Permisos por nivel	En progreso	v1.3.0	Control de acceso
Workflow de aprobaciones	Planificado	v1.3.0	Validación jerárquica

4.3 Registro de Decisiones Técnicas (ADR - Architecture Decision Records)

ADR-001: Selección de Room para Persistencia Móvil

Fecha: [12 dic 2025]

Estado: Aprobado

Contexto:

La aplicación móvil requiere almacenamiento local robusto con las siguientes características:

- Operación offline completa
- Relaciones entre entidades
- Consultas complejas con joins
- Type-safety en tiempo de compilación
- Soporte de Android Jetpack

Alternativas evaluadas:

1. **SQLite puro:** Mayor control pero verboso, propenso a errores SQL en strings
2. **Realm:** Excelente performance pero licencia comercial, comunidad en declive
3. **Room:** Abstracción oficial de Google sobre SQLite

Decisión:

Se seleccionó **Room** como capa de persistencia para la aplicación móvil.

Justificación:

- Parte oficial de Android Jetpack con soporte a largo plazo

- Validación de consultas en tiempo de compilación
- Integración perfecta con Kotlin Coroutines y Flow
- Type-safe y reduce boilerplate vs SQLite puro
- Migrations manejadas declarativamente
- Excelente documentación y comunidad activa
- Sin costos de licenciamiento

Consecuencias:

- Desarrollo más rápido y seguro
 - Menos bugs relacionados con SQL
 - Facilita testing con bases de datos in-memory
 - Curva de aprendizaje inicial para el equipo
 - Abstracción puede limitar optimizaciones avanzadas (mitigable con queries directos)
-

ADR-002: Adopción de Jetpack Compose para UI Móvil

Fecha: [08-dic-2025]

Estado: Aprobado

Contexto:

El proyecto móvil requiere una interfaz moderna, mantenible y con capacidad de escalar:

- UI reactiva que refleje cambios de estado automáticamente
- Reducción de código boilerplate
- Facilidad para crear componentes reutilizables
- Soporte a largo plazo

Alternativas evaluadas:

1. **XML + View Binding:** Enfoque tradicional, maduro pero verboso

2. **Jetpack Compose:** Declarativo, moderno, recomendado por Google
3. **Flutter:** Cross-platform pero requiere Dart, equipo con experiencia en Kotlin

Decisión:

Se adoptó **Jetpack Compose** como framework de UI.

Justificación:

- Paradigma declarativo reduce complejidad
- Menos código (~30-40% menos que XML views)
- Composición sobre herencia facilita reutilización
- Hot reload acelera iteraciones de diseño
- Integración nativa con ViewModels y StateFlow
- Futuro de Android UI según Google
- Material Design 3 pre-implementado
- Equipo ya domina Kotlin

Consecuencias:

- Desarrollo UI significativamente más rápido
- Código más testeable (composables son funciones puras)
- Mejor performance en listas largas (lazy composites)
- Versión mínima Android API 21+ requerida
- Curva de aprendizaje de paradigma declarativo
- Algunas librerías legacy requieren interoperabilidad

ADR-003: MongoDB para Aplicación de Escritorio

Fecha: [25-nov-2025]

Estado: Aprobado

Contexto:

La aplicación de escritorio necesita persistencia flexible para:

- Iteración rápida en esquema de datos (proyecto en evolución)
- Consultas complejas de agregación para reportes
- Escalabilidad futura sin migraciones complejas
- Operación local sin servidor externo

Alternativas evaluadas:

1. **SQLite**: Ligero, sin instalación, pero esquema rígido
2. **PostgreSQL**: Potente pero requiere servidor, overkill para uso local
3. **MongoDB**: NoSQL, flexible, embedded mode disponible

Decisión:

Se utilizó **MongoDB** en modo embebido para la aplicación de escritorio.

Justificación:

- Esquema flexible permite iteración ágil
- Documentos JSON alinean con Python dictionaries
- Pipeline de agregación potente para reportes
- No requiere migraciones complejas al agregar campos
- PyMongo tiene excelente soporte
- Fácil migración futura a arquitectura cliente-servidor si se necesita
- Consultas expresivas para reportes complejos

Consecuencias:

- Desarrollo iterativo más rápido
- Queries legibles y cercanas a lógica de negocio
- Escalable sin refactorización mayor
- Mayor consumo de espacio vs bases relacionalmente normalizadas
- Requiere disciplina para mantener consistencia de datos
- Learning curve para equipo acostumbrado a SQL

ADR-004: Arquitectura Desacoplada sin Sincronización Automática

Fecha: [14-dic]

Estado: Aprobado (Temporal - Revisión en v2.0)

Contexto:

Se requiere decidir el nivel de integración entre aplicación móvil y escritorio:

- Recursos limitados de desarrollo en fase inicial
- Necesidad de lanzar MVP rápidamente
- Complejidad de sincronización bidireccional confiable
- Casos de uso actuales no requieren datos en tiempo real

Alternativas evaluadas:

1. **Sincronización automática en tiempo real:** Máxima funcionalidad pero alta complejidad
2. **Sincronización periódica con resolución de conflictos:** Balance medio
3. **Operación independiente con exportación manual:** Mínima complejidad

Decisión:

Arquitectura desacoplada con bases de datos independientes y **sin sincronización automática** en v1.x.

Justificación:

- Time-to-market reducido significativamente
- Menor superficie de bugs críticos
- Cada aplicación optimizada para su caso de uso
- No requiere infraestructura de red adicional
- Permite validar modelo de negocio antes de inversión mayor
- Exportación/importación manual suficiente para casos actuales

Consecuencias:

- Lanzamiento más rápido

- Cada aplicación independiente y robusta
- Simplicidad operativa
- Usuarios deben gestionar dos bases de datos
- Riesgo de divergencia de datos
- Workflow manual de consolidación

Revisión planificada:

- v2.0 evaluará arquitectura cliente-servidor con API REST
 - Backend centralizado con PostgreSQL o continuar MongoDB
 - Sincronización bidireccional con resolución de conflictos
 - Autenticación centralizada y control de acceso
-

ADR-005: Material Design 3 como Sistema de Diseño

Fecha: [10-dic]

Estado: Aprobado

Contexto:

Necesidad de definir lineamientos visuales consistentes:

- Aplicación móvil requiere look & feel profesional
- Mantener coherencia entre plataformas (escritorio/móvil)
- Reducir tiempo de diseño de componentes
- Accesibilidad y mejores prácticas incorporadas

Decisión:

Adopción de **Material Design 3** como sistema de diseño base.

Justificación:

- Estándar de industria con amplia aceptación
- Componentes pre-diseñados reducen trabajo
- Guías de accesibilidad incluidas

- Soporte nativo en Jetpack Compose
- Consistencia cross-platform
- Documentación exhaustiva
- Temas dinámicos (Material You) mejoran UX

Consecuencias:

- Desarrollo UI acelerado
- Interfaz familiar para usuarios Android
- Accesibilidad garantizada (WCAG 2.1)
- Menos diferenciación visual vs otras apps
- Escritorio (Tkinter) no tiene Material nativo (adaptación manual)

4.4 Roadmap de Versiones Futuras

Versión 1.3.0 (Sistema de Aprobaciones) — Q[X] 202[X]

Planificado:

- Sistema de roles y permisos granular
- Workflow de aprobación de registros
- Notificaciones push para supervisores
- Dashboard para jefes de área
- Reportes departamentales
- Auditoría de cambios en registros

Versión 1.4.0 (Sincronización y Backend) — Q[X] 202[X]

Planificado:

- API REST centralizada
- Sincronización bidireccional automática

- Autenticación JWT
 - Backend con Node.js/Express o Django
 - Base de datos centralizada
 - Resolución de conflictos de sincronización
-

Versión 2.0.0 (Enterprise Features) — 202[X]

Planificado:

- Módulo de vacaciones y ausencias
- Integración con sistemas de nómina externos
- Reportería avanzada con Business Intelligence
- Aplicación web (Progressive Web App)
- Exportación masiva de datos
- Configuración multi-empresa
- Cumplimiento GDPR/regulaciones locales