**Guía Informe final de proyecto de Desarrollo de SIA**

# Introducción

El control de asistencia es fundamental para una gestión eficiente del personal en cualquier empresa. Este proyecto propone el desarrollo de un sistema automatizado de toma de asistencia que permita registrar entradas y salidas de los empleados de forma precisa, reduciendo errores manuales y mejorando la eficiencia administrativa.

# Etapa 1: Visión y Alcance del Proyecto

**1.1 Visión.**

Un sistema moderno de control de asistencia que simplifica el registro con códigos QR y mejora la gestión del tiempo y la información. Asistencia rápida, precisa y sin contacto.

1.2 Problema y contexto

Queremos resolver la dificultad y el tiempo que implica registrar manualmente la asistencia de personas en una organización, evitando errores humanos y mejorando la eficiencia del proceso.

En evidencia documentada se reportan deficiencias técnicas y operativas con los métodos manuales para toma de asistencia, registros ilegibles, incorrectos e incompletos, algunos como falsificación de datos o falta de respaldo. También se han reportado ineficiencias operativas como dificultad para realizar reportes.

**1.3 Usuarios y stakeholders.**

Empleados: necesitan registrar su asistencia de forma rápida y sin errores. Supervisores: requieren controlar puntualidad y detectar ausencias fácilmente.

Recursos Humanos: busca automatizar el registro y reducir errores en nómina.

Gerencia: necesita datos confiables para tomar decisiones y mejorar productividad. Auditores: requieren registros claros y trazables.

Proveedores de software: necesitan un sistema compatible y fácil de mantener.

**1.4 Propuesta de valor.**

El software permitirá registrar asistencias de forma rápida, segura y automatizada usando códigos QR, reduciendo tiempos de registro, evitando suplantaciones y permitiendo generar reportes confiables en tiempo real. El proyecto será exitoso si: el sistema puede escanear los QR y registrar asistencia de forma inmediata, se pueden generar reportes de asistencia por fecha, persona y hora, los usuarios valoran el sistema como más eficiente que el registro manual.

# Etapa 2: Organización y Planificación del Proyecto

En esta fase, se debe realizar toda la organización administrativa del proyecto. Esto incluye la planificación de tareas, asignación de roles, creación de una carta Gantt, y selección de herramientas de gestión de proyectos. Este paso es crucial para asegurar que el proyecto avance de manera ordenada y eficiente.

**2.1 Definición de Roles y Responsabilidades.**

|  |  |  |
| --- | --- | --- |
| ROL | RESPONSABLE | RESPONSABILIDAD |
| Líder del proyecto | Sebastian Mejias | Coordina el equipo, realiza seguimiento del proyecto, comunica con stakeholders. |
| Desarrollador Backend | Paula Henriquez | Implementa lógica de negocio, base de datos y seguridad del sistema. |
| Desarrollador Frontend | Paula Henriquez | Diseña e implementa la interfaz de usuario. |
| Analista de Requisitos | Alison Ramirez | Recolecta requerimientos funcionales y no funcionales. |
| Tester / QA | Alison Ramirez | Verifica la calidad del software y realiza pruebas funcionales y no funcionales. |
| Administrador DevOps | Sebastian Mejias | Gestiona el entorno de despliegue, integración continua, backups, y monitoreo. |

* 1. **Planificación de Tareas**

|  |  |  |  |
| --- | --- | --- | --- |
| Fase | Tareas clave | Duración estimada | Encargado |
| Análisis de requisitos | Levantamiento de requerimientos | 1 semana | Analista de Requisitos |
| Diseño del sistema | Arquitectura, diagramas, prototipos UI | 1 semana | Desarrollador Frontend / Administrador DevOps |
| Desarrollo Backend | API, base de datos, seguridad | 2 semanas | Desarrollador Backend / Administrador DevOps |
| Desarrollo Frontend | UI, integraciones con API | 2 semanas | Administrador DevOps / Desarrollador Backend |
| Pruebas y QA | Testing, corrección de errores | 1 semana | Tester |
| Despliegue y documentación | Deploy, manuales, capacitación | 1 semana | Todo el equipo |

**2.3 Herramientas de Organización y Gestión del Proyecto.**

Para asegurar el cumplimiento de los objetivos del proyecto y facilitar la colaboración entre los miembros del equipo, se utilizarán diversas herramientas de organización y gestión. Estas permitirán planificar, asignar tareas, monitorear avances y garantizar la comunicación efectiva.

### a) Herramientas de Gestión de Tareas y Proyectos

* **Trello / Gira**: Plataformas basadas en tableros Kanban que permiten organizar las tareas en columnas (*Pendiente, En progreso, Completado*), asignarlas a responsables, establecer prioridades y fechas de entrega.
* **Click Up / Asana**: Alternativas para gestión ágil de proyectos, con integración de cronogramas, dependencias y seguimiento del progreso.

### b) Herramientas de Comunicación y Colaboración

* **Slack / Microsoft Teams**: Canales de mensajería y videollamadas para mantener una comunicación fluida en tiempo real entre los miembros del equipo y los interesados.
* **Google Workspace (Docs, Sheets, Drive)**: Para la creación, almacenamiento y edición colaborativa de documentos y reportes del proyecto.

### c) Herramientas de Control de Versiones

* **GitHub / GitLab**: Repositorios de código que permiten el trabajo colaborativo, el control de versiones y la integración continua, garantizando la trazabilidad de los cambios en el software.

### d) Herramientas de Seguimiento del Tiempo y Productividad

* **Clockify / Toggl**: Aplicaciones para registrar el tiempo invertido en cada tarea, facilitando la medición de productividad y la estimación de esfuerzo.

### e) Herramientas de Documentación Técnica

* **Notion / Confluence**: Plataformas para centralizar la documentación técnica, manuales, diagramas y decisiones de diseño, asegurando la accesibilidad de la información.

**2.4 Matriz de riesgos inicial (seguridad, datos, continuidad).**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| | ***Riesgo identificado*** | ***Categoría (Seguridad/Datos/Continuidad)*** | ***Probabilidad (Alta/Media/Baja)*** | ***Impacto (Alto/Medio/Bajo)*** | ***Mitigación inicial*** | ***Responsable*** | | --- | --- | --- | --- | --- | --- | | *Acceso no autorizado a la BD* | *Seguridad* | *Media* | *Alto* | *Autenticación fuerte, cifrado, roles RBAC* | *Dev Backend* | | *Pérdida de información crítica* | *Datos* | *Media* | *Alto* | *Backups automáticos, control de versiones* | *DevOps* | | *Caída del servidor durante clase/demo* | *Continuidad* | *Alta* | *Medio* | *Infraestructura escalable en la nube, monitoreo* | *DevOps* | | *Fuga de datos sensibles de clientes* | *Seguridad* | *Baja* | *Alto* | *Cifrado en tránsito y en reposo, NDAs, políticas de acceso* | *Líder del proyecto* | | *Error humano al borrar registros* | *Datos* | *Media* | *Medio* | *Confirmaciones dobles, control de acceso, backups diarios* | *QA / Dev Backend* | |

2.5 **Checklist de decisión rápida**.

|  |
| --- |
| 1. Producto instalado vs SaaS (Software como servicio) Producto instalado (On-Premise)  SaaS (Servicio en la nube)  Justificación: Evita costos iniciales de infraestructura, fácil mantenimiento, escalabilidad, y actualizaciones automáticas. 2. Infraestructura: Nube vs On-Premise On-Premise**.**  Nube (IaaS/PaaS/DBaaS)Justificación:Mayor flexibilidad, escalabilidad, reducción de costos operativos y alta disponibilidad. 3. Arquitectura: Monolito vs Microservicios/Servicios Monolito modular  Microservicios / SOAJustificación:Menor complejidad para el tamaño del proyecto, permite desarrollo ágil y simplifica mantenimiento inicial. 4. Proceso de desarrollo: Plan-Dirigido vs Ágil/DevOps Plan-dirigido (Cascada / V-Model)  Ágil / DevOps (Scrum, Kanban, XP)Justificación:Permite iteraciones rápidas, adaptación al cambio, integración continua y mejora incremental. |

# Etapa 3: Selección del Modelo de Desarrollo

En esta fase, deberás seleccionar el modelo de desarrollo de software que utilizarás para guiar el proyecto. Justifica tu elección basándote en las características del proyecto y las características de cada modelo.

**3.1 Descripción del Proyecto**

El proyecto consiste en el desarrollo de un **sistema de control de asistencia para empresas mediante QR**, orientado a gestionar de manera eficiente la presencia y puntualidad de sus empleados. El sistema busca reemplazar métodos manuales o sistemas obsoletos, ofreciendo una solución moderna, segura y accesible desde distintos dispositivos.

Entre sus principales funciones se consideran:

* Registro de asistencia y salida mediante escaneo de QR.
* Validación en tiempo real de la identidad del empleado.
* Reportes y estadísticas de asistencia y puntualidad.
* Accesibilidad multiplataforma (web y dispositivos móviles).
* Seguridad en el manejo de datos.

**3.2 Modelos de Desarrollo Considerados.**

1. **Modelo en Cascada  
   -** Estructurado, con fases secuenciales (análisis, diseño, desarrollo, pruebas, implementación).  
   - Adecuado para proyectos con requerimientos muy claros y poco cambiantes.  
   - Desventaja: poca flexibilidad ante cambios.
2. **Modelo Incremental  
   -** Desarrollo en entregas parciales (módulos funcionales que se van sumando).  
   - Permite mostrar avances al cliente y obtener retroalimentación temprana.  
   - Requiere una planificación cuidadosa para integrar los incrementos.
3. **Modelo Iterativo  
   -** El producto se construye y mejora en versiones sucesivas, refinando requisitos y funcionalidades.  
   - Permite aprendizaje y adaptación conforme avanza el proyecto.  
   - Riesgo: requiere coordinación constante y revisiones frecuentes.
4. **Modelo Ágil (Scrum, Kanban, etc.)  
   -** Orientado a la colaboración, flexibilidad y entregas rápidas.  
   -Favorece la adaptación a cambios en los requerimientos.  
   - Requiere un equipo comprometido, comunicación continua y gestión dinámica.

**3.3 Justificación del Modelo Seleccionado.**

Para este proyecto,cómo este modelo se adapta al ciclo de vida del proyecto elegido. El **modelo de desarrollo Ágil** resulta el más adecuado, principalmente bajo el marco de **Scrum**, debido a las siguientes razones:

* **Requerimientos cambiantes:** Al tratarse de un sistema nuevo, es probable que las empresas necesiten ajustes durante el desarrollo (nuevos reportes, mejoras de seguridad, funciones adicionales). El modelo ágil facilita estas adaptaciones.
* **Entrega temprana de valor:** Con sprints cortos, se pueden liberar versiones parciales (ejemplo: primero el módulo de registro QR, luego reportes, luego estadísticas), permitiendo a los usuarios empezar a utilizar el sistema sin esperar la versión final.
* **Retroalimentación constante:** Los clientes pueden probar cada incremento y dar sugerencias, lo que garantiza que el producto final cumpla con las expectativas.
* **Reducción de riesgos:** Al trabajar en iteraciones, los problemas se detectan y corrigen de manera temprana, evitando fallas críticas en etapas avanzadas.
* **Escalabilidad:** Si el sistema se expande (integración con nóminas, geolocalización, control biométrico), el modelo ágil permite añadir funcionalidades sin rehacer el proyecto completo.

En conclusión, el **modelo ágil** se adapta mejor al ciclo de vida de este proyecto porque combina **flexibilidad, entregas rápidas, colaboración activa y mejora continua**, asegurando que el sistema de control de asistencia mediante QR evolucione según las necesidades de las empresas.

# Etapa 4: Recolección y Análisis de Requerimientos

En esta fase, deberás identificar y documentar los requisitos del sistema. Estos pueden dividirse en requisitos funcionales (qué debe hacer el sistema) y requisitos no funcionales (cómo debe comportarse el sistema). Identificar y justificar la técnica que ocuparon para identificar los requisitos (Como documento Anexo deben entregar el DER)

**4.1 Técnica de Levantamiento de Requisitos.**

Para la definición de los requisitos funcionales y no funcionales del sistema de control de asistencia, se utilizó la técnica de encuesta tipo formulario aplicada a los stakeholders y potenciales clientes del sistema.

El propósito de esta técnica fue recopilar de manera estructurada las necesidades, expectativas y observaciones de los usuarios finales y de quienes supervisan el proceso de asistencia dentro de la empresa. A través de preguntas cerradas y abiertas, el formulario permitió identificar las funciones esenciales del sistema (como el registro de entrada y salida mediante código QR, la generación de reportes y la consulta de asistencias) y los aspectos de calidad esperados (como disponibilidad, seguridad, facilidad de uso y rendimiento).

La información recolectada sirvió como base para concluir y priorizar los requisitos que guiarán el desarrollo del sistema, asegurando que el producto final responda a las demandas reales de los usuarios y a los objetivos de la organización.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

**4.2 Requisitos Funcionales y No Funcionales.**

*Requisitos Funcionales:*

* **Requisito Funcional 1:** Registro de Usuarios
* **Requisito Funcional 2**: Registro de Entrada de Turno
* **Requisito Funcional 3**: Registro de Salida de Turno
* **Requisito Funcional 4**: Consulta de Asistencia Personal
* **Requisito Funcional 5:** Panel de Supervisión en Tiempo Real
* **Requisito Funcional 6:** Consulta de Faltas
* **Requisito Funcional 7:** Reporte Diario de Asistencia
* **Requisito Funcional 8:** Búsqueda de Usuarios
* **Requisito Funcional 9:** Edición de Datos de Usuario
* **Requisito Funcional 10:** Desactivación de Usuario
* **Requisito Funcional 11:** Accesibilidad Multiplataforma para Escaneo de QR
* **Requisito Funcional 12**: Almacenamiento Seguro de Información
* **Requisito Funcional 13**: Autenticación de Usuario para Acceso al Sistema
* **Requisito Funcional 14**: Registro de Auditoría (Logs)
* **Requisito Funcional 15**: Exportación General de Reportes

*Requisitos No Funcionales:*

* **Requisito No Funcional 1:** Disponibilidad del Sistema
* **Requisito No Funcional 2:** Rendimiento de Procesamiento
* **Requisito No Funcional 3:** Escalabilidad
* **Requisito No Funcional 4:** Seguridad de la Información
* **Requisito No Funcional 5:** Respaldo de Información
* **Requisito No Funcional 6:** Control de Accesos por Roles
* **Requisito No Funcional 7:** Trazabilidad y Auditoría
* **Requisito No Funcional 8:** Portabilidad Multiplataforma
* **Requisito No Funcional 9:** Usabilidad de la Interfaz
* **Requisito No Funcional 10:** Tiempo de Respuesta en Consultas
* **Requisito No Funcional 11:** Exportación Estándar
* **Requisito No Funcional 12:** Integridad de los Datos
  1. **Requisitos de Seguridad y Privacidad.**

1. **Autenticación Segura de Usuarios**: El sistema deberá requerir que cada usuario (empleado, supervisor, administrador) inicie sesión con credenciales únicas (usuario y contraseña)

**Criterio:** No se permitirá acceso sin autenticación válida.

1. **Roles y Permisos Diferenciados:** El sistema deberá asignar permisos según el rol del usuario:

* Empleado: solo acceso a su propia asistencia.
* Supervisor: acceso al estado de su equipo.
* Administrador: acceso total a la plataforma.

**Criterio:** Un usuario no podrá ver ni modificar información a la que no tiene permiso.

1. **Almacenamiento Seguro de Datos Personales**: Toda la información de empleados (nombre, RUT, correo, asistencias) deberá almacenarse en una base de datos protegida con cifrado estándar (ejemplo: AES-256).

**Criterio:** Los datos sensibles no podrán visualizarse en texto plano.

1. **Protección de la Sesión del Usuario:** El sistema deberá cerrar automáticamente la sesión de un usuario tras un periodo de inactividad (ejemplo: 10 minutos).

**Criterio:** Prueba con usuarios inactivos para validar cierre automático.

1. **Registro de Auditoría (Logs):** El sistema deberá guardar un registro de las acciones importantes realizadas (ejemplo: cambios en datos de usuario, generación de reportes).

**Criterio:** El log deberá indicar usuario, acción realizada, fecha y hora.

1. **Acceso Seguro a la Plataforma:** Todas las comunicaciones entre los usuarios y el sistema deberán realizarse bajo protocolo seguro HTTPS.

**Criterio:** No se permitirá conexión mediante HTTP sin cifrado.

1. **Enmascaramiento de Datos Sensibles:** El sistema deberá mostrar solo la información estrictamente necesaria en pantalla. Ejemplo: ocultar parcialmente correos o RUT en listados.

**Criterio:** Los datos sensibles completos solo serán visibles para administradores autorizados.

1. **Respaldo Seguro de Información.** El sistema deberá generar copias de seguridad automáticas diarias de la base de datos y almacenarlas de manera protegida.

**Criterio:** Validar restauración correcta de respaldos en pruebas periódicas.

1. **Control de Intentos de Inicio de Sesión:** El sistema deberá bloquear temporalmente una cuenta tras varios intentos fallidos de inicio de sesión (ejemplo: 5 intentos).

**Criterio:** El sistema mostrará mensaje de bloqueo y permitirá recuperación de acceso mediante administrador o correo de recuperación.

1. **Eliminación Segura de Usuarios Inactivos:** El sistema deberá permitir la desactivación de cuentas de empleados que ya no pertenezcan a la organización, sin borrar su historial.

**Criterio**: El usuario quedará marcado como “inactivo” y no podrá iniciar sesión ni registrar asistencias.

* 1. **Otros Requisitos.**

1. **Gestión de Roles:** El sistema debe permitir crear, editar y desactivar usuarios según su rol (empleado, supervisor, administrador).
2. **Autenticación de Actores:** Cada actor debe autenticarse con credenciales únicas para acceder al sistema.
3. **Acceso Diferenciado:** Los supervisores y administradores deben tener acceso a reportes y paneles de control, mientras que los empleados solo a su historial
4. **Permisos de Edición:** El sistema debe diferenciar permisos de edición de datos, permitiendo solo a los administradores modificar información sensible.
5. **Visualización de Perfil:** El sistema debe permitir que cada actor vea su propio perfil y configuraciones asociadas a su rol.

**Requisitos de Operación:**

1. **Registro Rápido de Asistencia:** El sistema debe registrar entradas y salidas mediante código QR en menos de 2 segundos.
2. **Generación de Reportes:** Debe generar reportes diarios y exportarlos en formatos PDF o Excel.
3. **Respaldo Automático de Datos:** Debe almacenar automáticamente los datos en la base de datos con respaldo diario.
4. **Multiplataforma:** El sistema debe funcionar en distintos dispositivos (web, móviles, tablets) con adaptación automática de la interfaz.
5. **Operación Offline:** Debe permitir la operación incluso sin conexión, guardando los registros para sincronizar al volver a conectarse.

**Requisitos de Interacción y Complementarios**

1. **interfaz Intuitiva:** La interfaz debe ser clara e intuitiva, permitiendo que un usuario nuevo aprenda a registrar asistencia en menos de 5 minutos.
2. **Confirmación de Acciones:** El sistema debe mostrar mensajes visuales o sonoros al registrar asistencia o salida correctamente.
3. **Notificación de Errores:** Debe notificar errores de escaneo o códigos QR no válidos de forma clara y comprensible.
4. **Búsqueda de Usuarios:** Debe permitir búsqueda rápida de usuarios mediante filtros como nombre, cargo o área (solo para roles autorizados).
5. **Soporte de Idiomas:** La plataforma debe ser accesible en español y permitir futuras configuraciones de idioma para usuarios multilingües.
   1. **Priorización de Requerimientos.**

**Requerimientos de Alta Prioridad**

RF-01: Registro de Usuarios

RF-02: Registro de Entrada de Turno

RF-03: Registro de Salida de Turno

RF-05: Panel de Supervisión en Tiempo Real

RF-07: Reporte Diario de Asistencia

RF-11: Accesibilidad Multiplataforma para Escaneo de QR

RF-12: Almacenamiento Seguro de Información

RF-13: Autenticación de Usuario para Acceso al Sistema

RF-14: Registro de Auditoría (Logs

RNF-01: Disponibilidad del Sistema

RNF-02: Rendimiento de Procesamiento

RNF-04: Seguridad de la Información

RNF-05: Respaldo de Información

RNF-06: Control de Accesos por Roles

RNF-07: Trazabilidad y Auditoría

RNF-08: Portabilidad Multiplataforma

RNF-10: Tiempo de Respuesta en Consultas

RNF-12: Integridad de los Datos

**Requerimientos de Media Prioridad**

RF-04: Consulta de Asistencia Personal

RF-06: Consulta de Faltas

RF-09: Edición de Datos de Usuari

RF-10: Desactivación de Usuario

RF-15: Exportación General de Reporte

RNF-03: Escalabilidad

RNF-09: Usabilidad de la Interfaz

RNF-011: Exportación Estándar

**Requerimientos de Baja Prioridad**

RF-08: Búsqueda de Usuarios