Acerca de arc42

arc42, La plantilla de documentación para arquitectura de sistemas y de software.

Por Dr. Gernot Starke, Dr. Peter Hruschka y otros contribuyentes.

Revisión de la plantilla: 7.0 ES (basada en asciidoc), Enero 2017

1. Introducción y Metas

1.2.3 Requerimientos No Funcionales

Id	Requirement	Explanation
RNF1	Disponibilidad	El sistema debe estar disponible 24/7. Backend con redundancia en servidores críticos.
RNF2	Escalabilidad	Soporte para miles de usuarios concurrentes (ejemplo: universidades grandes).
RNF3	Seguridad	Comunicación cifrada (HTTPS + TLS). Tokens JWT / OAuth2. Cumplimiento de GDPR/Habeas Data.
RNF4	Rendimiento	Tiempo de respuesta < 2 segundos en operaciones comunes. Manejo eficiente de reportes masivos.
RNF5	Portabilidad	App disponible en Android e iOS. Versión web compatible con navegadores modernos.
		Interfaz intuitiva, multilenguaje y accesible según normas

2. Restricciones de la Arquitectura

2.1 Restricciones Tecnológicas

- La app debe desarrollarse en **Flutter** para asegurar compatibilidad en Android e iOS.
- El backend debe implementarse en **ORACLE Apex**, priorizando escalabilidad y modularidad.
- La base de datos debe ser **Oracle**, con **Redis** como caché para optimizar el rendimiento.
- Toda comunicación debe realizarse mediante **HTTPS/TLS**.

2.2 Restricciones Operativas

- El sistema debe estar disponible **24/7**, con un máximo de **3 horas de inactividad**

3. Alcance y Contexto del Sistema

3.1 Alcance del Sistema

El sistema de **Control de Asistencia** tiene como objetivo principal **digitalizar y automatizar el control de asistencia** en instituciones educativas y organizaciones empresariales, reduciendo procesos manuales y mejorando la precisión en los registros.

3.1.1 Funcionalidades dentro del alcance

- Registro de asistencia mediante **app móvil** (entrada y salida).
- Validación de asistencia mediante **QR**.
- **Autenticación segura** con usuario/contraseña o SSO (Google/Microsoft).
- Gestión de **horarios, grupos, cursos, jornadas**.

4.2.2 Interfaz de Usuario (App Móvil)

- Aplicación híbrida (Flutter).
- Funcionalidad offline con sincronización en línea.
- Autenticación segura mediante JWT u OAuth 2.0.

4.2.3 Comunicación y Backend

- API RESTful para validación y procesamiento de datos.
- Backend ligero (Oracle Apex).
- Control de accesos basado en roles (RBAC).
- Validaciones de integridad en tiempo real.

4.2.4 Gestión de Datos

- Base de datos relacional (Oracle).
- Cacheo con Redis para mejorar rendimiento.
- Almacenamiento histórico de asistencia con trazabilidad.

4.2.5 Seguridad

- Encriptación en tránsito (HTTPS + TLS).
- Autenticación con tokens JWT.
- Cumplimiento con normativas de protección de datos.

4.2.6 Integración con Sistemas Externos

- Reportes automáticos en PDF / Excel.

4.2.7 Infraestructura

- Despliegue con contenedores Docker.
- Despliegue en la nube (AWS, Azure, GCP) o en servidores locales.
- Balanceadores de carga para asegurar disponibilidad.

4.2.8 Monitoreo y Observabilidad

- Métricas con Prometheus + Grafana.

4.3 Decisiones Arquitectónicas

- Arquitectura modular con servicios desacoplados.
- Uso de tecnologías ligeras y escalables (Oracle Apex, Flutter).
- Base de datos relacional con soporte a integridad y relaciones complejas.
- Seguridad como prioridad en autenticación, cifrado y auditoría.

4.4 Trade-offs

- **Microservicios vs Monolito:** Se inicia con servicios modulares (menor complejidad) con visión futura hacia microservicios completos.
- **Infraestructura en la nube vs on-premise:** Dependerá del presupuesto y políticas de la institución.
- **Funcionalidad offline:** Aumenta complejidad técnica pero garantiza continuidad del servicio.

4.5 Riesgos Potenciales

- Gestión de seguridad y privacidad de datos sensibles.
- Complejidad en el mantenimiento de sincronización offline/online.
- Escalabilidad limitada si no se implementa orquestación adecuada en fases iniciales.

5. Vista de Bloques

5.1 Visión general

El sistema se compone de seis bloques principales:

1. **Aplicación móvil (UI/UX)**
2. **Módulo de autenticación y gestión de sesión**

5.2 Jerarquía de bloques

- **Nivel 1 (Subsistemas):**
  - UI
  - Autenticación
  - Asistencia
  - Sincronización
  - Almacenamiento
  - Utilitarios
- **Nivel 2 (Componentes internos):**
  - Autenticación → login manager, token handler.
  - Asistencia → validador de horarios, gestor local de eventos, sincronizador de asistencias.

5.3 Tabla de bloques principales

Bloque	Responsabilidades	Interfaces	Dependencias
UI móvil	Interacción con el usuario, captura de datos	Pantallas y formularios	Autenticación, Asistencia
Autenticación	Validar credenciales, emitir tokens, controlar sesión	`login()`, `logout()`, `refresh()`	UI, Almacenamien local
Registro de asistencia	Captura de entradas/salidas, validación de ubicación	`registrarAsistencia()`, `historial()`	Autenticación, Geolocalizació Almacenamien

5.4 Diagramas

```
flowchart TB
    subgraph UI["Aplicación móvil (UI/UX)"]
        UI1[Pantallas de usuario]
    end

    subgraph Auth["Autenticación"]
        A1[Login Manager]
        A2[Token Handler]
    end

    subgraph Asistencia["Registro de asistencia"]
        R1[Asistencia Validator]
        R2[Asistencia Local Store]
        R3[Asistencia Sync Manager]
    end

    subgraph Sync["Sincronización"]
        S1[Cliente HTTP]
        S2[Manejador de colas]
        S3[Adaptador de datos]
    end

    subgraph DB["Almacenamiento (Oracle APEX)"]
        D1[DAO]
        D2[Repositorios]
    end

    subgraph Utils["Utilitarios"]
        U2[Validaciones]
        U3[Logging]
    end

    %% Relaciones
    UI1 --> A1
    UI1 --> R1

    A1 --> A2
    A1 --> D1

    R1 --> R2
    R1 --> R3
    R3 --> S1

    S1 --> S2
    S1 --> S3

    R2 --> D1

    U2 --> R1
    U3 --> UI1
    U3 --> A1
    U3 --> R1
    U3 --> S1
    U3 --> D1
```

5.5 Consideraciones de modularidad

- Los bloques están diseñados para minimizar dependencias circulares.
- El almacenamiento local y utilitarios son reutilizables por múltiples módulos.
- La separación entre registro de asistencia y sincronización permite un uso offline robusto.

5.6 Relación con otras vistas

- En la **vista runtime**, los bloques se coordinan para casos de uso como *“registro de asistencia con validación en línea”*.
- En la **vista de despliegue**, la app móvil (bloques UI, lógica y almacenamiento) corre en el dispositivo, mientras la sincronización se conecta al backend desplegado en la nube.

6. Vista de Ejecución

La vista runtime describe cómo los distintos componentes del sistema colaboran en escenarios de ejecución concretos. A continuación se presentan los principales casos de uso.

6.1 Escenario: Inicio de sesión

Objetivo: Validar credenciales y establecer sesión segura.

Secuencia:

1. El usuario ingresa credenciales en la **Aplicación móvil (UI/UX)**.
2. El módulo de **Autenticación** envía las credenciales al **Almacenamiento (Oracle APEX)**.
3. Oracle APEX valida el usuario y devuelve un token.
4. El **Token Handler** guarda el token para futuras peticiones.
5. Se notifica a la UI que el inicio de sesión fue exitoso.

6.2 Escenario: Registro de asistencia sin conexión

Objetivo: Permitir al usuario registrar asistencia aun sin conexión a internet.

Secuencia:

1. El usuario marca asistencia desde la **Aplicación móvil (UI/UX)**.
2. El **Validador de asistencia** revisa la información.
3. Si no hay conexión, los datos se guardan en el **Almacenamiento local temporal** dentro del dispositivo.
4. El **Sync Manager** marca la asistencia como pendiente.
5. Se notifica a la UI que la asistencia fue registrada localmente.

6.3 Escenario: Sincronización de asistencias

Objetivo: Subir registros locales pendientes a Oracle APEX.

Secuencia:

1. El **Sync Manager** detecta conexión disponible.
2. El **Cliente HTTP** empaqueta los registros pendientes.
3. El **Manejador de colas** organiza los envíos.
4. Oracle APEX recibe los registros y responde confirmando almacenamiento.
5. El **Sync Manager** actualiza el estado local y notifica a la UI.

6.4 Escenario: Consulta de historial de asistencias

Objetivo: Mostrar al usuario las asistencias registradas.

Secuencia:

1. El usuario solicita el historial en la **Aplicación móvil (UI/UX)**.
2. La UI llama al módulo de **Registro de asistencia**.
3. El módulo consulta primero en la caché local.
4. Si no existe la información completa, el **Cliente HTTP** consulta en **Oracle APEX**.
5. El **Adaptador de datos** transforma la respuesta en un formato amigable.
6. La UI despliega la información al usuario.

```
sequenceDiagram
    actor U as Usuario
    participant UI as Aplicación móvil
    participant Auth as Autenticación
    participant Sync as Sincronización
    participant DB as Oracle APEX
```

7. Vista de Despliegue

La vista de despliegue describe la infraestructura técnica donde se ejecuta el sistema, así como la asignación de los componentes principales a dicha infraestructura.

7.1 Nodos principales

- **Dispositivo móvil (Android/iOS):**
  Ejecuta la aplicación móvil que incluye la interfaz de usuario, validaciones básicas, almacenamiento temporal de asistencias y el cliente de sincronización.
- **Servidor Oracle APEX (Cloud / On-Premise):**
  Plataforma de base de datos y backend que gestiona usuarios, asistencias, autenticación y reportes.
- **Servidor de Autenticación (opcional):**
  Puede estar integrado en Oracle APEX o desplegado como un servicio separado para validar credenciales y emitir tokens.
- **Servicios de Sincronización / API REST:**
  Interfaz expuesta en Oracle APEX (o como microservicio externo) para recibir los datos de asistencia y proveer consultas de historial.

7.2 Relaciones

- La **Aplicación móvil** se conecta mediante internet (HTTPS) al **servidor Oracle APEX**.
- El **Servidor APEX** puede apoyarse en:
  - **Módulo de Autenticación** (si está separado).
  - **Servicios de sincronización** para recibir registros desde los móviles.
- El **Dispositivo móvil** almacena datos localmente cuando no hay conexión, y sincroniza con Oracle APEX cuando la conexión se restablece.

7.3 Consideraciones técnicas

- **Protocolos de comunicación:** HTTPS con JSON.
- **Seguridad:** Uso de tokens de autenticación (JWT u OAUTH2).
- **Disponibilidad:** El servidor Oracle APEX debe estar altamente disponible para garantizar la sincronización de múltiples dispositivos.
- **Escalabilidad:** Posibilidad de balanceo de carga sobre el servidor APEX o servicios asociados si la demanda crece.

```
flowchart TB
    subgraph Mobile["📱 Dispositivo móvil (Android/iOS)"]
        UI["Aplicación móvil (UI/UX + Registro + Sync)"]
    end

    subgraph Cloud["☁️ Oracle APEX Server"]
        DB["Gestión de asistencia y usuarios"]
        API["Servicios REST / Sincronización"]
        AUTH["Módulo de Autenticación"]
```