

Equipo Linguardia

Linguardia – Gestión Digital de Guardias Docentes

Diciembre 14, 2025



Integrantes y roles Scrum

- Jazmín – Mobile Developer (Android)
 - Desarrollo de la aplicación móvil en Android Studio utilizando Kotlin.
- Fernando – Backend Integrator
 - Integración de la base de datos y gestión de la comunicación cliente-servidor.
- Darío – Web Developer / Documentación
 - Apoyo en la configuración del entorno Docker, análisis del servidor y redacción de documentación técnica.
- Lucía – Database Developer
 - Diseño e implementación de la base de datos MySQL.

-
- Nuria – Web Developer / Documentación
 - Desarrollo de la aplicación web, elaboración de la documentación técnica y trabajo de investigación.
 - Sara – UX Designer / DevOps
 - Diseño de la interfaz en Figma y creación y despliegue del entorno Docker.

1. Introducción al proyecto

Descripción general de la aplicación

Linguardia es una aplicación de escritorio y móvil diseñada para facilitar la gestión de guardias docentes en centros educativos. Permite el seguimiento de guardias de clase y recreo, la notificación automática a los profesores y la administración centralizada por parte de jefatura y administradores.

Problema que resuelve


La gestión de guardias en los centros educativos suele realizarse de forma manual o con herramientas poco eficientes, lo que provoca errores, falta de información actualizada y dificultades de comunicación. Linguardia digitaliza este proceso, centralizando la información y automatizando avisos y cambios.

Público objetivo

- Profesores de centros educativos
- Jefatura de estudios
- Administradores de sistemas del centro

Arquitectura prevista

Arquitectura **cliente-servidor**:

- **Cliente:** Aplicación de escritorio y aplicación móvil.
 - **Servidor:** Docker con Apache.
 - **Backend:** Kotlin.
 - **Base de datos:** MySQL.
 - **Control de versiones:** Git y GitHub
- 

2. Resumen de avances por Sprint

Sprint 1

Fechas: 22 septiembre – 6 octubre

Objetivos:

- Definición del proyecto Linguardia
- Análisis del problema y usuarios
- Organización inicial del repositorio

Trabajo realizado:

- Definición funcional de la app (usuarios, guardias, notificaciones)
- Creación de estructura base del proyecto
- Inicio de documentación (anteproyecto)

Evidencias:

- 00_documentation/Anteproyecto - LINGUARDIA.pdf
 - README.md
-

Sprint 2

Fechas: 7 octubre – 20 octubre

Objetivos:

- Diseño visual de la aplicación
- Prototipos de la app y la web

Trabajo realizado:

- Diseño completo en Figma
- 

-
- Mockups de login, calendario, notificaciones y perfil
 - Definición de identidad visual (colores, iconos)
 - Inicio de documentación (anteproyecto)

Evidencias:

- 01_concept_designs/app-prototypes/
 - 01_concept_designs/Linguardia_palette.png
 - 02_assets/
-

Sprint 3

Fechas: 21 octubre – 3 noviembre

Objetivos:

- Diseño de la base de datos
- Modelo entidad-relación

Trabajo realizado:

- Diseño del modelo de datos
- Creación del esquema MySQL
- Primera versión de la base de datos

Evidencias:

- 03_database/modelo.mwb
 - 03_database/modelo_grafico.png
 - 03_database/base_de_datos_old.sql
-

Sprint 4

Fechas: 4 noviembre – 17 noviembre



Objetivos:

- Desarrollo inicial de la aplicación Android
- Estructura base del proyecto móvil

Trabajo realizado:

- Creación del proyecto en Android Studio
- Pantallas principales (login, registro, calendario)
- Navegación básica

Evidencias:

- 04_app/app/src/main/java/...
 - LoginActivity.kt
 - SignupActivity.kt
 - MainActivity.kt
-


Sprint 5

Fechas: 18 noviembre – 1 diciembre

Objetivos:

- Funcionalidades principales de la app
- Notificaciones y gestión de guardias

Trabajo realizado:

- Pantalla de notificaciones
 - Vista de calendario y horarios
 - Gestión de datos del usuario
- 

Evidencias:

- NotificationsActivity.kt
 - ScheduleActivity.kt
 - CalendarWidget.kt
-

Sprint 6

Fechas: 2 diciembre – 14 diciembre


Objetivos:

- Web de administración y landing page
- Dockerización del proyecto
- Integración base de datos – aplicación

Trabajo realizado:

- Desarrollo de la web administrativa
- Landing page con sistema de ayuda (bot)
- Docker con Apache y MySQL
- Conexión BD y aplicación

Evidencias:

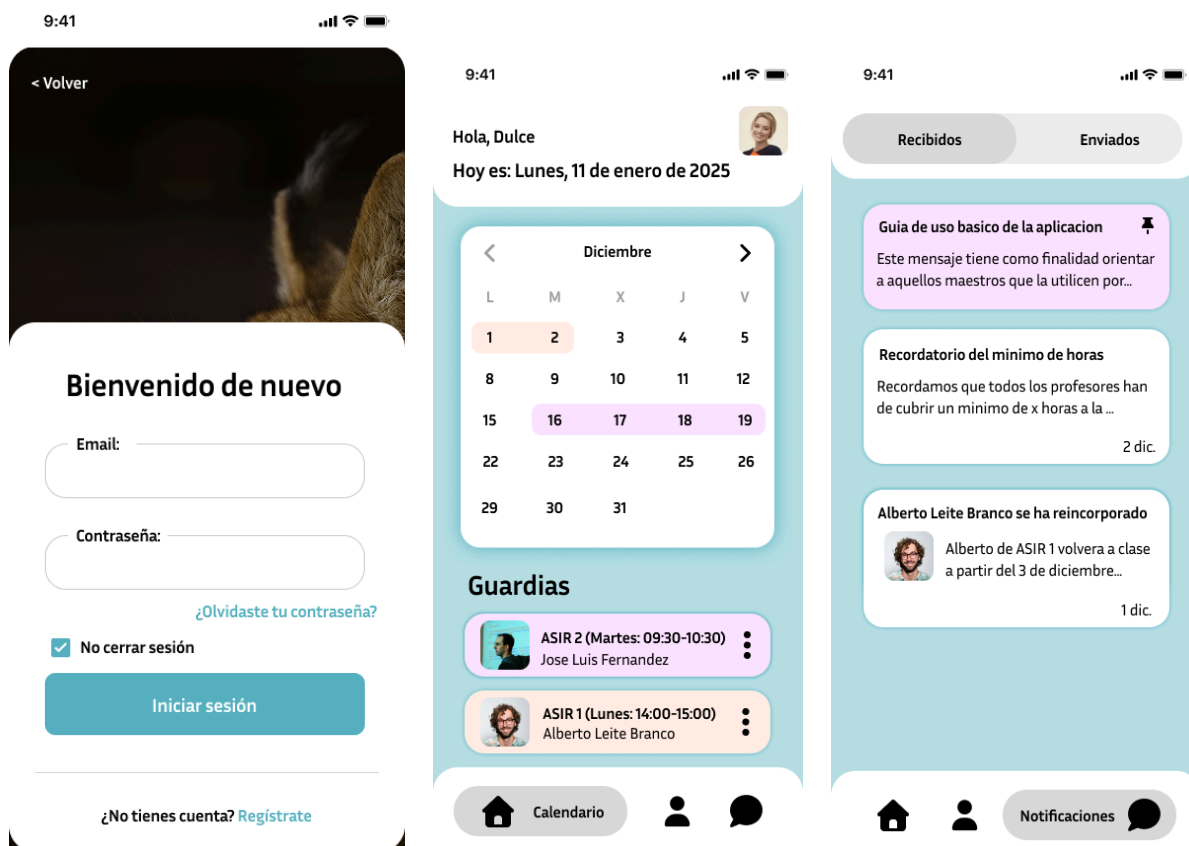
- 05_landing_and_administrator_website/
 - 06_docker_deployment/docker-compose.yml
 - db-init/init.sql
- 

3. Avances técnicos

Funcionalidades implementadas

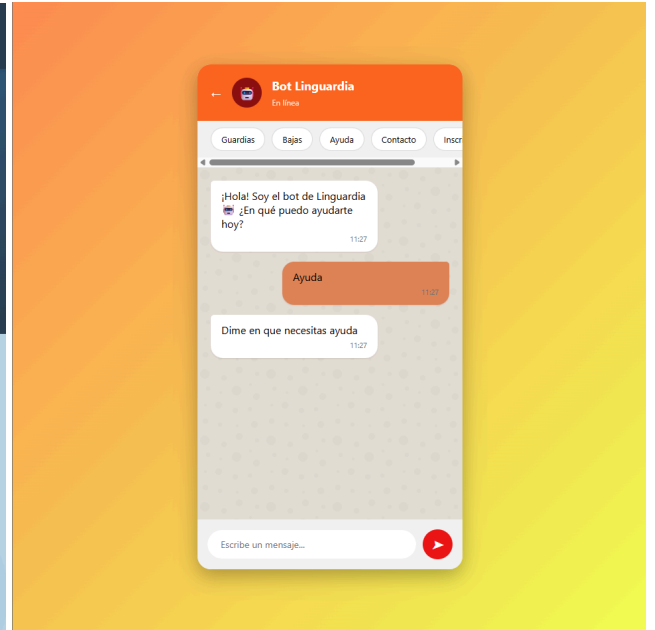
Aplicación móvil (Android)

- Sistema de inicio de sesión y registro de usuarios.
- Pantalla principal con acceso a calendario y notificaciones.
- Visualización del calendario de guardias (diario y semanal).
- Sistema de notificaciones para avisos de guardias y cambios.
- Gestión del perfil de usuario (datos básicos y cierre de sesión).



Aplicación web (Landing y administración)

- Landing page informativa con descripción del proyecto y acceso.
- Portal web administrativo para gestión básica.
- Sección de ayuda con bot integrado.
- Diseño responsive y coherente con la identidad visual de la app.



Bienvenido al portal de gestión de guardias

Usuario:

Contraseña:

☒ No cerrar sesión [¿Olvidaste tu contraseña?](#)

Iniciar sesión



Base de datos

- Base de datos MySQL estructurada para:
 - Usuarios
 - Profesores
 - Guardias
- Horarios
- Relaciones entre tablas correctamente definidas.
- Script de inicialización automática de la base de datos.

Infraestructura


- Entorno Dockerizado con:
 - Servidor web Apache
 - Base de datos MySQL
- Uso de docker-compose para facilitar el despliegue.
- Separación clara entre código, base de datos y recursos web.

Explicación técnica

Arquitectura del sistema

- El proyecto Linguardia utiliza una arquitectura cliente-servidor, donde los clientes (aplicación móvil Android y aplicación web) se comunican con un servidor central desplegado mediante Docker.
- El servidor Apache gestiona las peticiones de los clientes y se conecta a una base de datos MySQL donde se almacenan los datos de usuarios, profesores y guardias.

Aplicación móvil

- La aplicación móvil está desarrollada en Android Studio utilizando Kotlin.
 - La estructura se organiza en actividades independientes para cada funcionalidad principal:
- 

-
- LoginActivity y SignupActivity: autenticación de usuarios.
 - MainActivity: punto de entrada principal.
 - ScheduleActivity y CalendarWidget: visualización de horarios y guardias.
 - NotificationsActivity: gestión de avisos y notificaciones.
 - DataUserActivity: gestión del perfil de usuario.
 - Los recursos visuales se gestionan desde la carpeta res, permitiendo mantener una interfaz coherente y adaptable a distintos dispositivos.

Aplicación web

- La web está desarrollada con HTML, CSS y JavaScript, separando estructura, estilo y comportamiento.
- Incluye:
 - Página principal (index.html)
 - Portal de administración
 - Bot de ayuda mediante scripts JavaScript
- El diseño sigue la misma identidad visual definida en los prototipos de Figma.

Base de datos

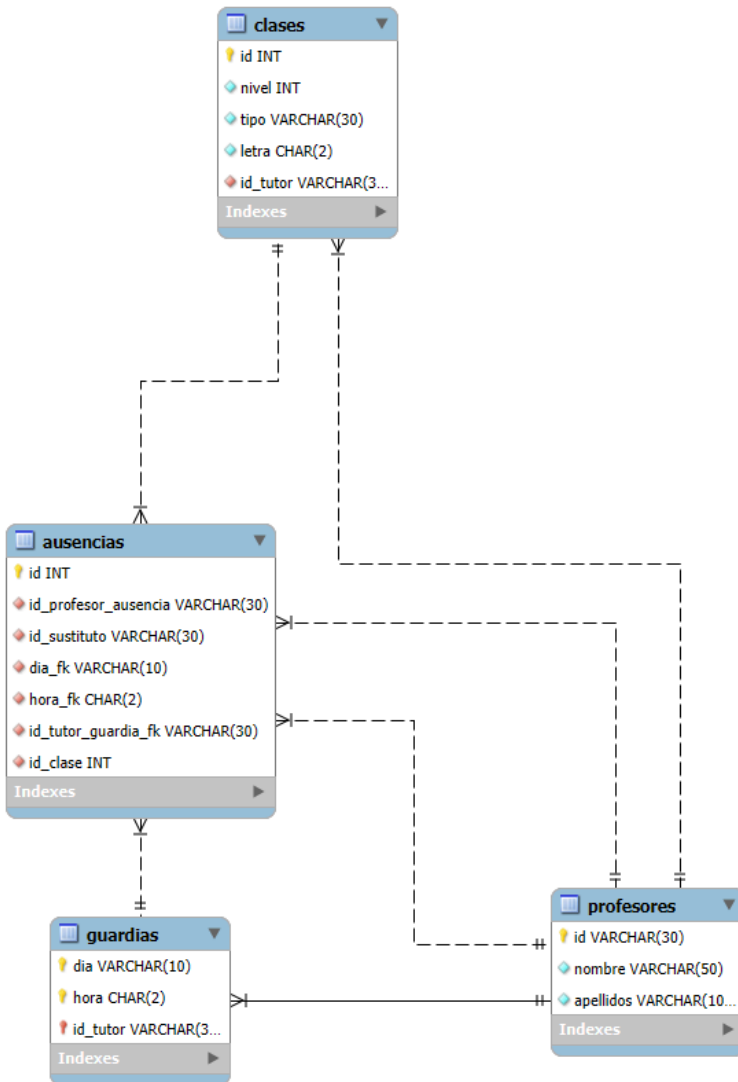
- La base de datos MySQL ha sido diseñada previamente mediante un modelo gráfico y posteriormente implementada mediante scripts SQL.
- Incluye:
 - Claves primarias y foráneas.
 - Relaciones entre profesores, usuarios y guardias.
 - Scripts de creación y carga inicial de datos.

Docker y despliegue

- Se ha utilizado Docker para asegurar un entorno de ejecución estable y replicable.
- docker-compose.yml define los servicios:
 - Apache (web)
 - MySQL (base de datos)
- El contenedor de base de datos se inicializa automáticamente mediante scripts SQL.

- Facilita la puesta en marcha del proyecto sin configuraciones manuales complejas.

Diagramas



4. Demo

El proyecto Linguardia puede demostrarse mediante la ejecución del entorno Docker incluido. Desde la carpeta 06_docker_deployment, basta con ejecutar el siguiente comando:

```
docker compose up
```

Este comando levanta automáticamente el servidor Apache y la base de datos MySQL, dejando operativa la aplicación web y el entorno necesario para su evaluación, sin configuraciones adicionales.

