Diseño de Arquitectura del Sistema - TaskManager

**Fecha:**

**Autor:**

**Versión:** 1.0

# 1. Introducción

Este documento describe la arquitectura del sistema Braille Express, una solución tecnológica que busca mejorar el acceso a la lectura de personas con discapacidad visual mediante un dispositivo físico y una aplicación móvil que traduce texto y voz a Braille.  
El sistema está compuesto por una app Android y un microcontrolador Arduino Nano, comunicados vía Bluetooth. Se define la estructura general del sistema, los componentes que lo conforman, sus interacciones y las tecnologías empleadas.

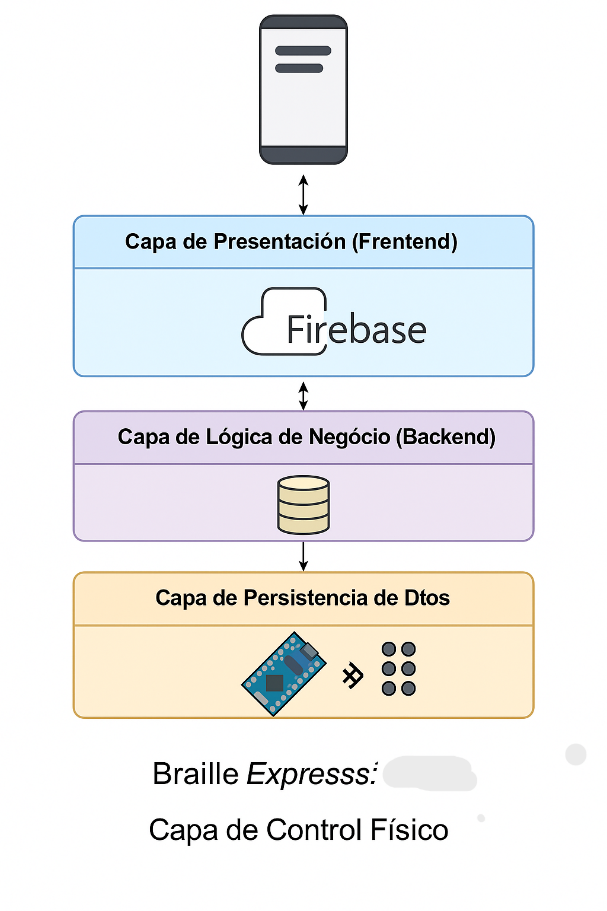
# 2. Visón General de la Arquitectura

## 2.1. Estilo Arquitectónico

El sistema adopta un modelo cliente-servidor con arquitectura en capas, y se apoya en un esquema Backend-as-a-Service (BaaS) usando Firebase. Se identifican cuatro capas principales:

* **Capa de Presentación (Frontend)**: Interfaz con la que interactúa el usuario.
* **Capa de Lógica de Negocio** **(Backend)**: Lógica de traducción a Braille y conexión con servicios.
* **Capa de Persistencia de Datos (Base de datos)**: Manejo de usuarios y sincronización en la nube.
* **Capa de Control Físico**: Representación física de los caracteres Braille usando servomotores.

## 2.2. Diagrama General de Arquitectura:



# 3. Descripción de Componentes Principales

## 3.1. Frontend (Presentación)

* Lenguaje: Java
* IDE: Android Studio
* Estilo visual: XML Layouts
* Responsabilidades:
* Captura de texto manual y por voz.
* Traducción automática del texto a Braille.
* Conexión con Firebase para login, configuración y almacenamiento.
* Comunicación Bluetooth con el dispositivo Braille.
* Notificación de errores, acciones exitosas y actualizaciones.

## 3.2. Backend (Lógica de negocio)

* **Ubicación:** Lógica embebida en la app móvil, complementada con funciones de Firebase.
* **Responsabilidades:**
  + Registro e inicio de sesión con DNI.
  + Validación de datos ingresados.
  + Conversión automática de texto a Braille.
  + Control de flujos de autenticación y sesión.
  + Integración con reconocimiento de voz.
  + Gestión de reglas de negocio: tareas como notificaciones o reconexión Bluetooth.

### 3.2.1. Módulos del Backend

* RegisterActivity.java y LoginActivity.java: Registro e inicio de sesión con DNI.
* MainActivity.java y Mapa.java: Conversión de texto plano a representación Braille.
* Conexión.java: Conexión y transmisión de datos hacia el dispositivo.
* MainActivity.java: Procesamiento de voz a texto mediante API externa.
* MainActivity.java: Manejo de alertas y confirmaciones visuales o sonora

## 3.3. Capa de Persistencia de Datos (Firebase)

* Sistema Gestor: Firebase
* Responsabilidad:
  + Registro y login de usuarios.
  + Almacenamiento de configuraciones del usuario.
  + Sincronización de datos entre sesiones y dispositivos.
  + Gestión segura mediante reglas de acceso y encriptación de datos.

## 3.4. Capa de Control Físico (Hardware)

* Componente Principal: Arduino Nano con módulo Bluetooth HC-06
* Elementos clave:
* 6 servomotores SG90 conectados a Shield Nano IO V1.0
* Comunicación inalámbrica vía Bluetooth
* **Responsabilidades:**
* Recepción de caracteres desde la app.
* Activación de servomotores para representar los puntos Braille.
* Reinicio de conexión en caso de errores.

# 4. Integraciones Externas (Opcionales)

* **Firebase:** Autenticación y almacenamiento remoto.
* **API de Reconocimiento de Voz:** Google Speech-to-Text (para entrada por voz).
* **Bluetooth HC-06:** Comunicación inalámbrica entre la app y el Arduino.

# 5. Seguridad

* Autenticación segura: Firebase con validación de DNI.
* Protección de datos: Almacenamiento cifrado y seguro de la información en la nube.
* Comunicación Bluetooth: Conexión limitada a dispositivos emparejados.
* Validación de entradas: Validar de texto para evitar errores o comandos maliciosos.
* Privacidad: No se almacena contenido de texto traducido en la nube (solo configuración de usuario).

# 6. Escalabilidad y Despliegue

Posibles problemas que pueden surgir durante el desarrollo.

* Aplicación Móvil:
* Compatible con Android 7.0 o superior.
* Desplegable mediante Google Play Store.
* Backend Firebase:
* Escalable automáticamente según la demanda.
* Dispositivo Físico:
* Modular y replicable con componentes accesibles.
* Escalable en instituciones educativas mediante múltiples kits Braille.
* Posibles desafíos:
* Conectividad Bluetooth limitada a 10 metros.
* Capacidad limitada del Arduino en procesamiento.

# 7. Conclusiones

La arquitectura propuesta para Braille Express busca un equilibrio entre simplicidad técnica, accesibilidad y escalabilidad. Aprovecha tecnologías conocidas y confiables como Android, Firebase y Arduino, y se enfoca en ofrecer una solución tangible y funcional para la comunidad con discapacidad visual. La estructura modular permite su mejora futura mediante nuevas funcionalidades, sin comprometer su rendimiento ni accesibilidad.