

# Documento de Diseño de Software

Versión: 1.0.0

Autor(es): Fabricio Daniel Lara Valencia / Juan Pablo Gutierrez Garcia / Pablo Portillo

Madera

Fecha: 9 / 30 / 2025

Historial de revisiones:

---

## 1. Introducción (Caso de uso)

- **Descripción general del sistema :** La skill Biblioteca Pines es una skill de Aleza que permite que los usuarios administren de manera sencilla y rápida una biblioteca personal mediante comandos de voz. La skill contiene funcionales como agregar libros, buscar información, listar libros que estén disponibles o prestados, también puede registrar préstamos y devoluciones, así como consultar historiales. Todo para que la búsqueda de libros y la gestión sea hecha de la manera más accesible y sencilla.
- **Objetivo del documento :** Organizar y presentar de manera clara los casos de usos principales de la skill, requerimientos funcionales y no funcionales, especificar aspectos, interacciones, alcances y limitaciones, aportando mucho en el desarrollo, validación y pruebas del sistema, asegurando que las funciones cumplan con los requerimientos definidos.
- **Alcance :**
  - Incluye : Invocación de la skill / Gestión de libros / Consultar estado de préstamos y devoluciones / Soporte a comandos adicionales
  - **No incluye :** Integración con bases de datos externas o bibliotecas digitales comerciales/ Sincronización en múltiples dispositivos / Gestión avanzada de usuarios con roles o autenticación
- **Actores principales :** Usuario / Sistema Alexa / Biblioteca pines

## **Casos de uso :**

- CU01 – Invocar la skill: Como usuario, quiero abrir “Biblioteca Pines” para acceder a mis libros.
- CU02 – Mostrar opciones: Como usuario, quiero conocer qué puedo hacer para interactuar con la skill.
- CU03 – Listar libros: Como usuario, quiero ver los libros disponibles, prestados o todos, para conocer mi biblioteca.
- CU04 – Buscar libro: Como usuario, quiero buscar un libro específico por título para obtener su información.
- CU05 – Agregar libro: Como usuario, quiero registrar un libro con título, autor y tipo para almacenarlo en la biblioteca.
- CU06 – Prestar libro: Como usuario, quiero prestar un libro a una persona para llevar control de mis préstamos.
- CU07 – Devolver libro: Como usuario, quiero registrar la devolución de un libro prestado para mantener actualizado mi inventario.
- CU08 – Consultar préstamos: Como usuario, quiero ver qué libros tengo prestados y a quién.
- CU09 – Consultar devueltos: Como usuario, quiero ver qué libros ya fueron devueltos para consultar el historial.
- CU10 – Siguiendo página: Como usuario, quiero avanzar en la lista de libros cuando hay muchos resultados.
- CU11 – Salir del listado: Como usuario, quiero detener la enumeración de libros cuando ya no deseo escuchar más.
- CU12 – Limpiar caché: Como usuario, quiero reiniciar los datos de la biblioteca en caso de inconsistencias.
- CU13 – Respuesta general: Como usuario, quiero poder responder libremente a preguntas de Alexa para completar información.
- CU14 – Ayuda: Como usuario, quiero obtener ayuda para entender cómo usar la skill.

- CU15 – Cancelar acción: Como usuario, quiero cancelar una acción en curso si me equivoqué.
  - CU16 – Detener skill: Como usuario, quiero salir completamente de la skill cuando ya no la necesito.
  - CU17 – Regresar al inicio: Como usuario, quiero reiniciar la skill para volver al menú principal.
  - CU18 – Manejar errores: Como usuario, quiero que la skill me dé una guía cuando no entiende lo que digo.
- 

## **2. Problem Statement (Declaración del problema)**

La Biblioteca Pines sufre de ineficiencia operativa debido a la gestión manual y a la dependencia de sistemas informáticos lentos para el control de inventario, préstamos y devoluciones. Esta fricción en los procesos se traduce en tiempos de espera excesivos para los usuarios y en una visibilidad incompleta del stock de libros. No existe una herramienta de acceso rápido y manos libres que permita a los bibliotecarios realizar tareas críticas con la voz.

El proyecto busca implementar una skill de Alexa que funcione como una interfaz de voz completa. Esta skill debe permitir automatizar todas las funciones de gestión de la biblioteca, incluyendo el registro de nuevos títulos, el listado filtrado de libros, y el ciclo completo de préstamo/devolución. El objetivo es ofrecer una experiencia de usuario fluida, con mecanismos integrados para la ayuda, cancelación y manejo de errores.

Si no se implementa esta solución, la biblioteca continuará perdiendo eficiencia, aumentando los errores de inventario y disminuyendo la satisfacción del usuario. La principal limitación es que toda la funcionalidad debe ser totalmente operable por comandos de voz y ser compatible con la arquitectura de Amazon Alexa. El éxito se define por la capacidad del sistema de agilizar y precisar la administración del inventario.

---

### **3. Requerimientos funcionales**

#### **RF-01 : Invocar Skill**

**Descripción:** El sistema debe permitir invocar la skill con el nombre “biblioteca pines”.

**Actor:** Usuario

**Prioridad:** Alta

**Criterios de aceptación:** EL usuario dice “Alexa, abre biblioteca pines” -> Alexa responde con un mensaje de bienvenida.

#### **RF-02 : Mostrar opciones**

**Descripción:** El sistema debe mostrar al usuario las opciones disponibles de la skill.

**Actor:** Usuario

**Prioridad:** Alta

**Criterios de aceptación:** El usuario pide opciones -> Alexa enumera las funcionalidades disponibles.

#### **RF-03 : Listar libros**

**Descripción:** El sistema debe listar los libros filtrando por tipo (prestados, disponibles, todos) o por autor.

**Actor:** Usuario

**Prioridad:** Alta

**Criterios de aceptación:** El usuario solicita libros con o sin filtros -> Alexa responde con la lista de correspondientes.

#### **RF-04 : Buscar libro**

**Descripción:** El sistema debe buscar un libro específico según su título.

**Actor:** Usuario

**Prioridad:** Alta

**Criterios de aceptación:** El usuario va a decir el título de un libro -> Alexa devuelve si el libro existe

#### **RF-05 : Agregar libro**

**Descripción:** El sistema debe permitir registrar un nuevo libro indicando título, autor y tipo.

**Actor:** Usuario

**Prioridad:** Alta

**Criterios de aceptación:** El usuario dicta los datos del libro -> Alexa confirma que el libro fue agregado

#### **RF-06 : Prestar libro**

**Descripción:** El sistema debe registrar el préstamo de un libro a una persona.

**Actor:** Usuario

**Prioridad:** Alta

**Criterios de aceptación:** El usuario dice que el libro y a quien se presta -> Alexa confirma que el préstamo fue registrado

#### **RF-07 : Devolver libro**

**Descripción:** El sistema debe registrar la devolución de un libro prestado, ya sea por título o ID de préstamo.

**Actor:** Usuario

**Prioridad:** Alta

**Criterios de aceptación:** El usuario solicita devolver un libro -> Alexa confirma que la devolución fue registrada

#### **RF-08 : Consultar préstamos**

**Descripción:** El sistema debe mostrar la lista de préstamos activos.

**Actor:** Usuario

**Prioridad:** Media-alta

**Criterios de aceptación:** El usuario pregunta por sus préstamos -> Alexa muestra los libros prestados y a quien

#### **RF-09: Consultar devueltos**

**Descripción:** El sistema debe mostrar el historial de libros devueltos.

**Actor:** Usuario

**Prioridad:** Media

**Criterios de aceptación:** El usuario pide el historial de devoluciones -> Alexa enumera los libros devueltos

#### **RF-10 : Siguiendo página**

**Descripción:** El sistema debe mostrar la siguiente página de resultados al listar libros.

**Actor:** Usuario

**Prioridad:** Media

**Criterios de aceptación:** El usuario dice “seguir” -> Alexa muestra mas libros de la lista

#### **RF-11 : Salir del listado**

**Descripción:** El sistema debe detener la lectura o listado de libros a petición del usuario.

**Actor:** Usuario

**Prioridad:** Media

**Criterios de aceptación:** El usuario dice “Salir de la lista” -> Alexa detiene la enumeración de libros

#### **RF-12 : Limpiar caché**

**Descripción:** El sistema debe permitir limpiar la memoria o reiniciar los datos de la biblioteca.

**Actor:** Usuario

**Prioridad:** Baja

**Criterios de aceptación:** El usuario solicita limpiar o renunciar datos -> Alexa confirma que la cachpe fue eliminada.

### **RF-13 : Respuesta general**

**Descripción:** El sistema debe aceptar respuestas abiertas para completar información faltante en otros intentos.

**Actor:** Usuario

**Prioridad:** Media

**Criterios de aceptación:** Alexa pide infomación faltante -> El usuario responde lo que queira y Alexa lo registra

### **RF-14 : Ayuda**

**Descripción:** El sistema debe proporcionar ayuda al usuario sobre cómo usar la skill.

**Actor:** Usuario

**Prioridad:** Media

**Criterios de aceptación:** El usuario pide ayuda -> Alexa explica las funciones principales

### **RF-15 : Cancelar acción**

**Descripción:** El sistema debe permitir cancelar una acción en curso.

**Actor:** Usuario

**Prioridad:** Media

**Criterios de aceptación:** El usuario dice "Cancelar" -> Alexa le quita le seguimiento a la acción en curso

#### **RF-16 : Detener Skill**

**Descripción:** El sistema debe detener completamente la skill cuando el usuario lo solicite.

**Actor:** Usuario

**Prioridad:** Alta

**Criterios de aceptación:** El usuario dice “Para” -> Alexa cierra la sesión de la skill

#### **RF-17 : Regresar al inicio**

**Descripción:** El sistema debe regresar al inicio de la skill cuando se indique

**Actor:** Usuario

**Prioridad:** Baja

**Criterios de aceptación:** El usuario dice “ir al inicio” -> Alexa reinicia la interacción al principio.

#### **RF-18 : Manejar errores**

**Descripción:** El sistema debe manejar las frases no reconocidas mostrando un mensaje de error o guía.

**Actor:** Usuario

**Prioridad:** Alta

**Criterios de aceptación:** El usuario dice algo que no se entiende o que no este en contexto -> Alexa responde con un mensaje de fallback indicando opciones validas.

---

### **4. Requerimientos no funcionales**

#### **RNF-01 :**

**Atributo :** Rendimiento



**Descripción :** El sistema debe responder a cada comando de voz en un tiempo aceptable

**Métricas :** Tiempo de respuesta con máximo de 3 segundos en el 90% de los casos

#### **RNF-02 :**

**Atributo :** Disponibilidad

**Descripción :** El sistema debe estar disponible para los usuarios la mayor parte del tiempo

**Métricas :** Disponible al 99% del tiempo a menos de un contratiempo inesperado.

#### **RNF-03 :**

**Atributo :** Usabilidad

**Descripción :** La skill debe ser fácil de usar e intuitiva, con respuestas claras y concisas.

**Métricas :** 80% de los usuarios deben poder completar una task sin problemas.

#### **RNF-04 :**

**Atributo :** Compatibilidad

**Descripción :** La skill debe funcionar en todos los dispositivos compatibles con Alexa.

**Métricas :** Ejecución en 100% de los dispositivos que sean compatibles.

#### **RNF-05 :**

**Atributo :** Seguridad

**Descripción :** Los datos de los usuarios y sus libros deben manejarse de forma segura

**Métricas :** Cumple la política de privacidad de Amazon al 100%

#### **RNF-06 :**

**Atributo :** Escalabilidad

**Descripción :** El sistema debe soportar un crecimiento en el numero de libros y préstamos.

**Métricas :** Soporta escalabilidad sin degradación en la calidad.

#### **RNF-07 :**

**Atributo :** Mantenibilidad

**Descripción :** La skill debe ser fácil de actualizar y corregir.

**Métricas :** Hacer pruebas unitarias al incorporar cambios en el 50% del código

#### **RNF-08 :**

**Atributo :** Accesibilidad

**Descripción :** La skill debe ser entendible por distintos usuarios, evitando respuestas técnicas.

**Métricas :** Al menos 90% de respuestas validadas como claras por los usuarios prueba.

---

## **5. Arquitectura / Diseño – C4 Diagrams**

- Descripción general de la arquitectura / componentes principales
- **Nota:** incluir aquí los enlaces a los diagramas C4 que ya existan (GitHub, Notion, Figma, SharePoint, etc.)

Diagrama de Contexto C1

[https://github.com/Fabricio1210/AlexaSkill/blob/main/Diagramas/C1\\_Context.txt](https://github.com/Fabricio1210/AlexaSkill/blob/main/Diagramas/C1_Context.txt)

Diagrama de Contenedores C2

[https://github.com/Fabricio1210/AlexaSkill/blob/main/Diagramas/C2\\_Containers.txt](https://github.com/Fabricio1210/AlexaSkill/blob/main/Diagramas/C2_Containers.txt)

Diagrama de Componentes C3

[https://github.com/Fabricio1210/AlexaSkill/blob/main/Diagramas/C3\\_Components.txt](https://github.com/Fabricio1210/AlexaSkill/blob/main/Diagramas/C3_Components.txt)

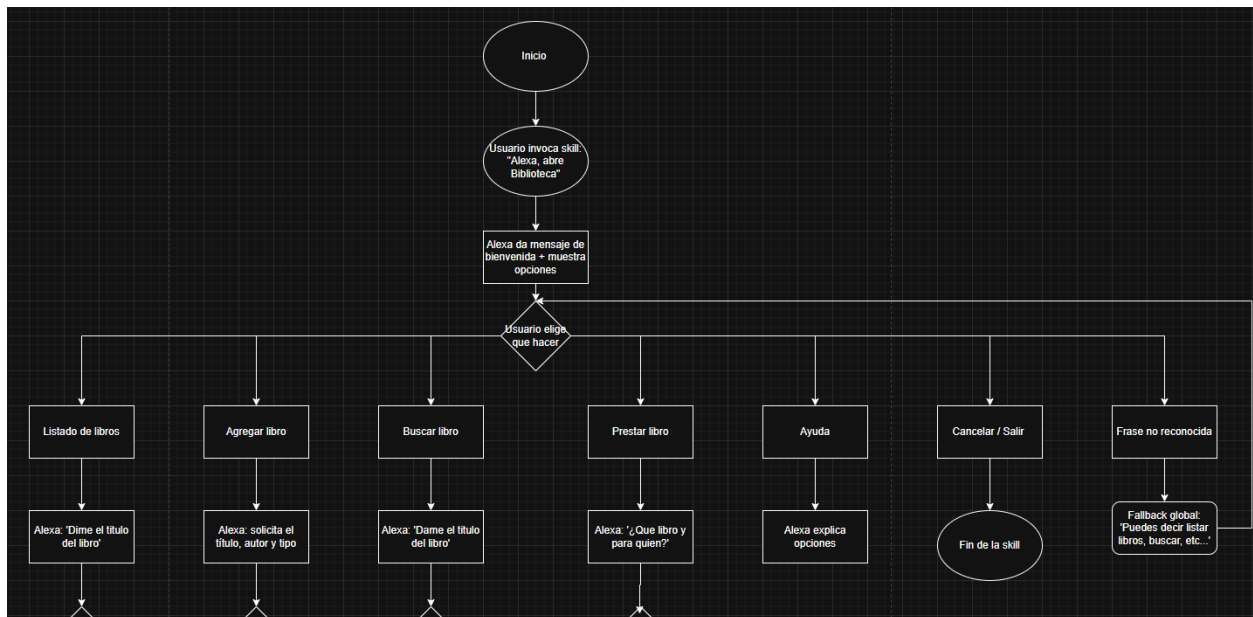
(Opcional) Diagrama de nivel de clases / código

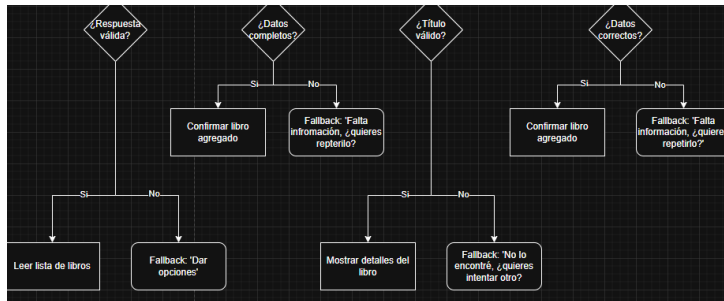
[https://github.com/Fabricao1210/AlexaSkill/blob/main/Diagramas/C4\\_Code.txt](https://github.com/Fabricao1210/AlexaSkill/blob/main/Diagramas/C4_Code.txt)

## 6. Diseño VUI / Diagramas de flujo de voz

- Objetivo del diseño VUI: Facilitar que el usuario gestione su biblioteca de forma rápida y sin fricción, utilizando únicamente la voz.
- Estilo / tono / lenguaje de voz: El tono de la skill tiene que ser amigable y respetuoso, utilizando un lenguaje directo con respuestas claras y breves.
- Escenarios de uso por voz (cuándo, dónde, dispositivos, entorno) : La skill está pensada para usarse principalmente en contextos cotidianos como su casa o en un escritorio, mientras el usuario realiza otras actividades y tiene las manos ocupadas, por lo que los comandos deben ser breves y tolerantes a intervenciones espontáneas.

Diagrama de voz:





---

## 8. Apéndices (si aplica)

### Decisiones de diseño

Primeramente creemos que era necesario generar algunos objetos como lo son préstamo y libro ya que libro es el protagonista en nuestras acciones por lo que tiene atributos fáciles de abstraer y ya puestos en forma de JSON en cambio préstamo es un poco más difícil de abstraer porque no es un objeto físico sin embargo como lo trataba el programa base con un identificador con fechas y estos atributos que tiene era fácil pensar que pudiese ser manejado como un objeto estos objetos se guardaron en la carpeta de models siendo los modelos de estos dos tipos de objetos solo se le ponen unos cuantos métodos para evitar la sobrecarga de este, luego se hace otra clase llamada LibraryService que esta seria la clase que relaciona estos dos porque los servicios de nuestra biblioteca pueden manipular libros y prestamos y es por eso que aquí están la mayoría de métodos que afectan directamente a los intents como agregar o eliminar y aquí se ponen los métodos, adicionalmente se creyó conveniente crear una clase de PhrasesManager debido al principio de encapsulamiento y no dejar variables globales volando, también porque se repetía muchas veces el tener que hacer random para obtener una frase lo cual lo hacia redundante por lo que se decidió solo declararlo una vez y traerlo siempre que se necesite.

Se decidió poner todo lo relacionado de persistencia y temas de cache en un archivo aparte para encapsularlo de un posible mal uso ya que es una de nuestras partes mas importante, igualmente tienen bastante relación entre si y por eso se decido abstraer del código. Por último el archivo Config.py representa configuraciones globales como ocupamos en uno u otro momento lo usamos como un .env de cierta manera

---

## 9. Revisión y mantenimiento del documento

<b>Versión</b>	<b>Fecha</b>	<b>Autor(es)</b>	<b>Cambios Realizados</b>	<b>Aprobado por / Revisor principal</b>	<b>Comentarios adicionales</b>
0.1	2025-09-23	Pablo	Creación inicial del documento; estructura básica establecida	Juan Pablo	Primer borrador
0.2	2025-09-27	Juan Pablo	Adición de requerimientos no funcionales; ajuste a caso de uso	Pablo	Feedback de equipo/del profesor
0.3	2025-09-28	Fabricio	Incorporación de diagramas VUI; mejora de flujos de voz	Juan pablo, Pablo	Versiones de diagramas revisadas
1.0	2025-09-30	Juan Pablo, Fabricio y Pablo	Documento aprobado para uso / entrega final	Juan Pablo, Fabricio y Pablo	Versión oficial para implementación