

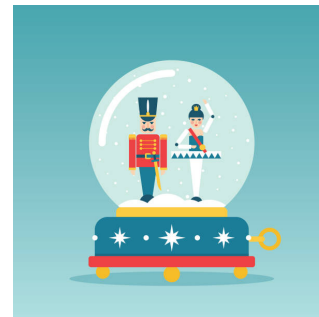
Notesbox

Contexto general

Notesbox es una empresa familiar que se dedica a la comercialización de cajitas musicales.

La empresa cuenta con un catálogo de melodías, de las que ya tienen el diseño de sus cilindros¹ y que puede enviar a fabricar ante demanda. Para agilizar tiempos y abaratar costos, Notesbox mantiene también un stock de cajitas con melodías comunes que están listas para vender.

Y como parte de su objetivo de expandir su negocio, han decidido automatizar algunos aspectos de su proceso.



Relevamiento

El sitio: Notesbox necesita un sitio que permita a los clientes realizar órdenes. Para ello, el cliente deberá crear una cuenta, cargar su dirección de entrega, recorrer su catálogo de melodías y agregar a su carrito de compras una cierta cantidad de cajitas de una melodía dada. Finalmente, deberá poder realizar una orden que incluya todas las cajitas del carrito y listar todas las órdenes, tanto las ya *entregadas* como las que no.

Direcciones de entrega: Cada cliente tiene una única dirección, pero ésta puede variar a lo largo del tiempo. Cuando una orden se crea, la dirección asociada a la misma es la dirección actual del cliente

El catálogo: las melodías del catálogo tienen un nombre que las identifica y un precio predefinido y están agrupadas en categorías. Cada una cuenta también con su secuencia MIDI, la cual es utilizada tanto para permitir reproducirla desde el navegador² como para generar la especificación de construcción que se le envía a cualquiera de las fábricas con las que Notesbox trabaja.

Las especificaciones de construcción: cuando una melodía es agregada al catálogo, se debe generar una especificación textual con las instrucciones de construcción del cilindro, que luego se envía a las fábricas con las que Notesbox trabaja para que sepan qué producir. El formato de dicha especificación está fuera de alcance, dado que nos han provisto un script ya desarrollado que toma un archivo MIDI, realiza la conversión a este formato y genera un archivo con el resultado. Nos han recomendado que utilicemos este script sin modificarlo.

Fabricantes: Notesbox trabaja con varios fabricantes, que se encargan de la fabricación de los cilindros y el resto de las piezas, que luego Notesbox ensambla. El contacto con estos fabricantes, tanto para consultar si tienen disponibilidad para fabricar como para realizarles el encargo, es telefónico.

¹ El cilindro es un componente metálico de una caja de música que contiene una muesca por cada nota, y su forma exacta por tanto variará para cada melodía.

² Los navegadores modernos cuentan con soporte nativo para reproducir MIDI



Stock: Notesbox utiliza un software de manejo de stock de tipo SAAS, con el cual debemos integrarnos mediante un API REST. Cuando un cliente realiza una orden, si la misma puede ser completada íntegramente con cajitas de stock, la misma debe quedar lista para envío y ser marcada como *pendiente de entrega*. En caso contrario, la orden debe quedar en un estado intermedio entre *pendiente de entrega* y *entregada: pendiente de fabricación*. Para mantener el stock actualizado, los administradores pueden realizar órdenes especiales de reposición, pero esto queda fuera de alcance.

La aplicación administrativa: Notesbox necesita que los administradores del sistema también cuenten con una aplicación web que les permita realizar su flujo de trabajo principal:

1. listar las fábricas,
2. listar las órdenes pendientes de fabricación que no hayan sido asignadas aún a fábricas,
3. asignar las órdenes a la primera fábrica que encuentren disponible
4. marcar una orden como *pendiente de entrega* cuando regrese de la fábrica

Además, debe permitirles consultar a qué fábrica fue asignada una orden, a fines de control. Por último, debe permitirles agregar nuevas melodías al catálogo y disparar el proceso de conversión de la secuencia MIDI a la especificación de construcción.

Envío de órdenes: la entrega de las cajitas al cliente queda fuera de alcance.

Alcance y Requerimientos:

1. Como cliente, poder registrarme cargando los datos básicos de contacto (nombre, apellido, email) y dirección de recepción de las órdenes.
2. Como cliente, poder consultar el catálogo de melodías y ordenar una o más cajitas musicales para mi dirección actual, enviando a fabricar en caso de ser necesario.
3. Como cliente, poder listar todas mis órdenes.
4. Como persona administradora, poder agregar cajas musicales al catálogo, disparando el proceso de conversión de la secuencia MIDI a la especificación de construcción.
5. Como persona administradora, poder revisar las órdenes *pendientes de fabricación* y asignarles una fábrica a su elección que esté en condiciones de aceptar la orden.
6. Como persona administradora, poder actualizar el estado de una orden.
7. Como persona administradora, poder consultar a qué fábrica fue asignada una orden.

Punto 1 – Arquitectura (30 puntos)

1. **(15 Puntos)** Como parte del proceso de diseño, se ha propuesto que el sistema presente una arquitectura de microservicios, siendo uno el servicio de clientes y el otro el de administradores.
 - a. ¿Qué opinás de esta propuesta? Justificá en base a las cualidades de diseño de Escalabilidad y Facilidad de mantenimiento.
 - b. Si estás de acuerdo, realizá un diagrama de despliegue de cómo quedaría la solución. En caso contrario, proponé un diagrama de despliegue que muestre tu contrapropuesta
2. **(15 Puntos)** Notesbox ha sido puesto en producción y nos han informado dos problemas. Para cada uno de ellos, explicá cual es una causa probable y qué mejoras de de arquitectura pueden realizarse para evitarlos:
 - a. Cuando los administradores del sitio cargan una melodía nueva en el catálogo, el sitio les arroja un error de timeout y la conversión jamás termina.



- b. La cantidad de usuarios ha crecido notoriamente y estamos empezando a notar que los pedidos HTTP empiezan a demorar cada vez más e incluso hemos detectado caídas del servidor.

Punto 2 - Modelo de Dominio (40 puntos)

Realizar y comunicar el diseño del componente que resuelve la lógica de negocio aplicando el paradigma OO.

- A. **(20 Puntos)** Documentar la solución utilizando diagramas UML (diagrama de clases obligatorio).
- B. **(10 Puntos)** Justificar las decisiones de diseño que se tomen, por ejemplo, haciendo referencia a los principios que guían al diseño o las consecuencias de aplicar un determinado patrón.
- C. **(10 Puntos)** Justificar en forma detallada cómo resuelve el siguiente requerimiento: “Como persona administradora, poder agregar cajas musicales al catálogo, disparando el proceso de conversión de la secuencia MIDI a la especificación de construcción”.

Punto 3 – Persistencia (30 puntos)

- A. **(20 Puntos)** Diseñar el modelo de datos del punto anterior para poder persistir en una base de datos relacional. Indicando las entidades, sus campos, claves primarias, las foráneas, cardinalidad, modalidad y las restricciones según corresponda.
- B. **(10 Puntos)** Justificar:
- Qué elementos del modelo es necesario persistir.
 - Cómo resolvió los *impedance mismatches*.
 - Las estructuras de datos que deban ser desnormalizadas, si corresponde.

NOTA: Explicar supuestos y justificar decisiones de diseño.

Condiciones de aprobación: Para aprobar debe sumar como mínimo 60 puntos y no menos del 50 % en cada sección.