

# Escaleta-Teclado GC

# Especificación de realización de casos de uso

**Versión <1.0>**

# Historial de revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor** |
| **13/09/2023** | **1.0** | **Primera versión** | **Wílmer E. León** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Prefacio**

Este documento de *Especificación de Realización de Casos de Uso* tiene como objetivo proporcionar una guía detallada sobre la implementación y ejecución de los casos de uso identificados para el sistema en desarrollo. La especificación se centra en describir de manera clara y concisa cómo cada caso de uso se llevará a cabo, los actores involucrados, las interacciones necesarias y las especificaciones de diagramas que muestran cada uno de los procesos implicados en la gestión de tareas del aplicativo.

La realización de casos de uso es una parte fundamental del proceso de desarrollo de software, ya que permite al equipo de desarrollo comprender las necesidades del usuario y traducirlas en funcionalidades concretas. Este documento servirá como referencia para los desarrolladores, diseñadores y cualquier otro miembro del equipo involucrado en la implementación del sistema.

# Tabla de contenidos

[Escaleta-Teclado GC 1](#_Toc177459513)

[Especificación de realización de casos de uso 1](#_Toc177459514)

[Historial de revisiones 2](#_Toc177459515)

[Tabla de contenidos 4](#_Toc177459516)

[Figuras 5](#_Toc177459517)

[Especificación de realización de casos de uso 6](#_Toc177459518)

[1. Introducción 6](#_Toc177459519)

[1.1. Propósito 6](#_Toc177459520)

[1.2. Alcance 6](#_Toc177459521)

[1.3. Definiciones, acrónimos y abreviaturas 6](#_Toc177459522)

[1.4. Referencias 6](#_Toc177459523)

[Documento 6](#_Toc177459524)

[Versión 6](#_Toc177459525)

[Fecha de versión 6](#_Toc177459526)

[1.5. Descripción general 7](#_Toc177459527)

[1. CASO DE USO <Ingresar al Módulo DCSHA> 7](#_Toc177459528)

[2. CASO DE USO <Seleccionar plantilla> 14](#_Toc177459529)

[3. CASO DE USO <Editar escaleta> 19](#_Toc177459530)

[4. CASO DE USO <Insertar ID> 24](#_Toc177459531)

[5. CASO DE USO <Editar escaleta> 25](#_Toc177459532)

[6. CASO DE USO <Editar escaleta> 25](#_Toc177459533)

[7. CASO DE USO <Editar escaleta> 26](#_Toc177459534)

[8. CASO DE USO <Editar escaleta> 26](#_Toc177459535)

[9. CASO DE USO <Editar escaleta> 27](#_Toc177459536)

[10. CASO DE USO <Editar escaleta> 27](#_Toc177459537)

[11. CASO DE USO <Editar escaleta> 28](#_Toc177459538)

[12. CASO DE USO <Editar escaleta> 28](#_Toc177459539)

[13. CASO DE USO <Editar escaleta> 29](#_Toc177459540)

[14. CASO DE USO <Editar escaleta> 30](#_Toc177459541)

[15. CASO DE USO <Editar escaleta> 30](#_Toc177459542)

[16. CASO DE USO <Editar escaleta> 31](#_Toc177459543)

[17. CASO DE USO <Editar escaleta> 31](#_Toc177459544)

[18. CASO DE USO <Editar escaleta> 32](#_Toc177459545)

[19. CASO DE USO <Editar escaleta> 32](#_Toc177459546)

[20. CASO DE USO <Editar escaleta> 33](#_Toc177459547)

[21. CASO DE USO <Editar escaleta> 33](#_Toc177459548)

# Figuras

[Ilustración 1: Caso de uso 1 | Ingresar a módulo DCSHA 9](#_Toc177459606)

[Ilustración 2: Diagrama de secuencia | Caso de uso 1 11](#_Toc177459607)

[Ilustración 3: Diagrama de colaboración | Caso de uso 1 12](#_Toc177459608)

[Ilustración 4: Diagrama de objetos | Caso de uso 1 13](#_Toc177459609)

[Ilustración 5: Caso de uso 2 | Seleccionar plantilla 15](#_Toc177459610)

[Ilustración 6: Diagrama de secuencia | Caso de uso 2 16](#_Toc177459611)

[Ilustración 7: Diagrama de objetos | Caso de uso 2 17](#_Toc177459612)

[Ilustración 8: Caso de uso 3 | Editar escaleta 20](#_Toc177459613)

[Ilustración 9: Diagrama de secuencia | Caso de uso 2 22](#_Toc177459614)

[Ilustración 10: Diagrama de objetos | Caso de uso 2 23](#_Toc177459615)

[Ilustración 11: Caso de uso 4 | Insertar ID 25](#_Toc177459616)

[Ilustración 12: Diagrama de secuencia | Caso de uso 26](#_Toc177459617)

[Ilustración 13: Diagrama de objetos | Caso de uso 27](#_Toc177459618)

[Ilustración 14: Caso de uso | 27](#_Toc177459619)

[Ilustración 15: Diagrama de secuencia | Caso de uso 27](#_Toc177459620)

[Ilustración 16: Diagrama de objetos | Caso de uso 27](#_Toc177459621)

[Ilustración 17: Caso de uso | 27](#_Toc177459622)

[Ilustración 18: Diagrama de secuencia | Caso de uso 28](#_Toc177459623)

[Ilustración 19: Diagrama de objetos | Caso de uso 28](#_Toc177459624)

[Ilustración 20: Caso de uso | 28](#_Toc177459625)

[Ilustración 21: Diagrama de secuencia | Caso de uso 28](#_Toc177459626)

[Ilustración 22: Diagrama de objetos | Caso de uso 28](#_Toc177459627)

[Ilustración 23: Caso de uso | 28](#_Toc177459628)

[Ilustración 24: Diagrama de secuencia | Caso de uso 29](#_Toc177459629)

[Ilustración 25: Diagrama de objetos | Caso de uso 29](#_Toc177459630)

[Ilustración 26: Caso de uso | 29](#_Toc177459631)

[Ilustración 27: Diagrama de secuencia | Caso de uso 29](#_Toc177459632)

[Ilustración 28: Diagrama de objetos | Caso de uso 29](#_Toc177459633)

[Ilustración 29: Caso de uso | 30](#_Toc177459634)

[Ilustración 30: Diagrama de secuencia | Caso de uso 30](#_Toc177459635)

[Ilustración 31: Diagrama de objetos | Caso de uso 30](#_Toc177459636)

[Ilustración 32: Caso de uso | 30](#_Toc177459637)

[Ilustración 33: Diagrama de secuencia | Caso de uso 30](#_Toc177459638)

[Ilustración 34: Diagrama de objetos | Caso de uso 31](#_Toc177459639)

[Ilustración 35: Caso de uso | 31](#_Toc177459640)

[Ilustración 36: Diagrama de secuencia | Caso de uso 31](#_Toc177459641)

[Ilustración 37: Diagrama de objetos | Caso de uso 31](#_Toc177459642)

[Ilustración 38: Caso de uso | 31](#_Toc177459643)

[Ilustración 39: Diagrama de secuencia | Caso de uso 31](#_Toc177459644)

[Ilustración 40: Diagrama de objetos | Caso de uso 32](#_Toc177459645)

[Ilustración 41: Caso de uso | 32](#_Toc177459646)

[Ilustración 42: Diagrama de secuencia | Caso de uso 32](#_Toc177459647)

[Ilustración 43: Diagrama de objetos | Caso de uso 32](#_Toc177459648)

[Ilustración 44: Caso de uso | 32](#_Toc177459649)

[Ilustración 45: Diagrama de secuencia | Caso de uso 33](#_Toc177459650)

[Ilustración 46: Diagrama de objetos | Caso de uso 33](#_Toc177459651)

[Ilustración 47: Caso de uso | 33](#_Toc177459652)

[Ilustración 48: Diagrama de secuencia | Caso de uso 33](#_Toc177459653)

[Ilustración 49: Diagrama de objetos | Caso de uso 33](#_Toc177459654)

[Ilustración 50: Caso de uso | 33](#_Toc177459655)

[Ilustración 51: Diagrama de secuencia | Caso de uso 34](#_Toc177459656)

[Ilustración 52: Diagrama de objetos | Caso de uso 34](#_Toc177459657)

[Ilustración 53: Caso de uso | 34](#_Toc177459658)

[Ilustración 54: Diagrama de secuencia | Caso de uso 34](#_Toc177459659)

[Ilustración 55: Diagrama de objetos | Caso de uso 34](#_Toc177459660)

[Ilustración 56: Caso de uso | 35](#_Toc177459661)

[Ilustración 57: Diagrama de secuencia | Caso de uso 35](#_Toc177459662)

[Ilustración 58: Diagrama de objetos | Caso de uso 35](#_Toc177459663)

# Especificación de realización de casos de uso

## Introducción

## Propósito

Este documento proporciona una descripción general completa del sistema, utilizando varios diagramas diferentes para representar las funciones del sistema.

## Alcance

El presente documento tiene como finalidad establecer un marco claro y conciso para la especificación de los casos de uso del sistema *Escaleta-Teclado GC,* una herramienta innovadora diseñada para optimizar la creación y gestión de escaletas mediante un teclado virtual intuitivo. Este sistema busca mejorar la eficiencia y la experiencia del usuario en la producción de contenido, permitiendo a los usuarios interactuar de manera fluida y efectiva con la aplicación. En este contexto, el alcance del documento abarca la identificación y descripción detallada de los actores involucrados, así como los casos de uso que representan las interacciones entre estos actores y el sistema. Se abordarán los requisitos funcionales y no funcionales que el sistema debe cumplir, así como las limitaciones y exclusiones que definirán el perímetro del proyecto. Además, se incluirán consideraciones sobre la interfaz de usuario y el manejo de la confidencialidad de los datos, asegurando que el sistema no solo cumpla con las expectativas de funcionalidad, sino que también garantice la seguridad y la integridad de la información. Este enfoque integral permitirá al equipo de desarrollo y a las partes interesadas tener una comprensión compartida de los objetivos y requisitos del sistema, facilitando así un proceso de desarrollo más alineado y eficiente.

### Definiciones, acrónimos y abreviaturas

|  |  |
| --- | --- |
| Término / acrónimo | Definición |
| ERS | Especificación de requerimientos de ‘software’: Especificación de requerimientos de *software*, que puede ser sinónimo de ERS en algunos contextos. |
| UML | Unified Modeling Language: Lenguaje de modelado unificado utilizado para especificar, visualizar, construir y documentar los artefactos de un sistema de *software*. |
| ETgc | Escaleta-Teclado GC: Aplicativo de montaje de escaleta de tiempos y teclados de previsualización de contenidos gráficos con texto para transmisiones de programas en Máster. |
|  |  |
|  |  |

### Referencias

|  |  |  |
| --- | --- | --- |
| Documento | Versión | Fecha de versión |
| Carta de proyecto Estimación de tiempo y recursos | 1.0 | 13 de septiembre de 2024 |
| Especificación de requerimientos | 1.0 | 10 de septiembre de 2024 |
| Formatos Plan de Mejoramiento 1- 2 - 3 | 1.0. | 10 de septiembre de 2024 |
| Enunciado de creación del diagrama de casos de uso | 1.0 | 12 de septiembre de 2024 |
| Documento de arquitectura de software | 1.0 | 12 de septiembre de 2024 |

### Descripción general

Los casos de uso en el sistema *Escaleta-Teclado GC* son representaciones fundamentales que describen cómo los diferentes actores interactúan con la aplicación para lograr objetivos específicos relacionados con la creación y gestión de escaletas. Cada caso de uso captura una funcionalidad particular del sistema, proporcionando un marco claro para entender las expectativas y necesidades de los usuarios finales, así como los requisitos técnicos que deben ser implementados por el equipo de desarrollo.

En este sistema, los casos de uso abarcan una variedad de acciones que los usuarios pueden realizar, tales como la creación de nuevas escaletas, la edición de escaletas existentes, la eliminación de contenido no deseado y la visualización de escaletas en diferentes formatos. Además, se contemplan interacciones más complejas, como la integración de elementos multimedia y la colaboración entre múltiples usuarios, lo que permite un trabajo en equipo más eficiente.

Cada caso de uso se documentará con un enfoque en los flujos de trabajo típicos, así como en las excepciones y errores que pueden surgir durante la interacción con el sistema. Esto incluye la identificación de precondiciones necesarias para iniciar un caso de uso, el flujo principal de acciones que se llevan a cabo, y los flujos alternativos que describen cómo manejar situaciones imprevistas.

La documentación de los casos de uso no solo servirá como guía para el desarrollo del sistema, sino que también facilitará la comunicación entre los diferentes interesados, asegurando que todos tengan una comprensión clara de las funcionalidades que se están implementando. Al proporcionar una visión estructurada de las interacciones del usuario con el sistema, los casos de uso en *Escaleta-Teclado GC* son esenciales para garantizar que la solución final cumpla con las expectativas de los usuarios y se alinee con los objetivos del proyecto.

### CASO DE USO <Ingresar al Módulo DCSHA>

#### Breve descripción

El productor abre la aplicación e ingresa al módulo DCSHA.

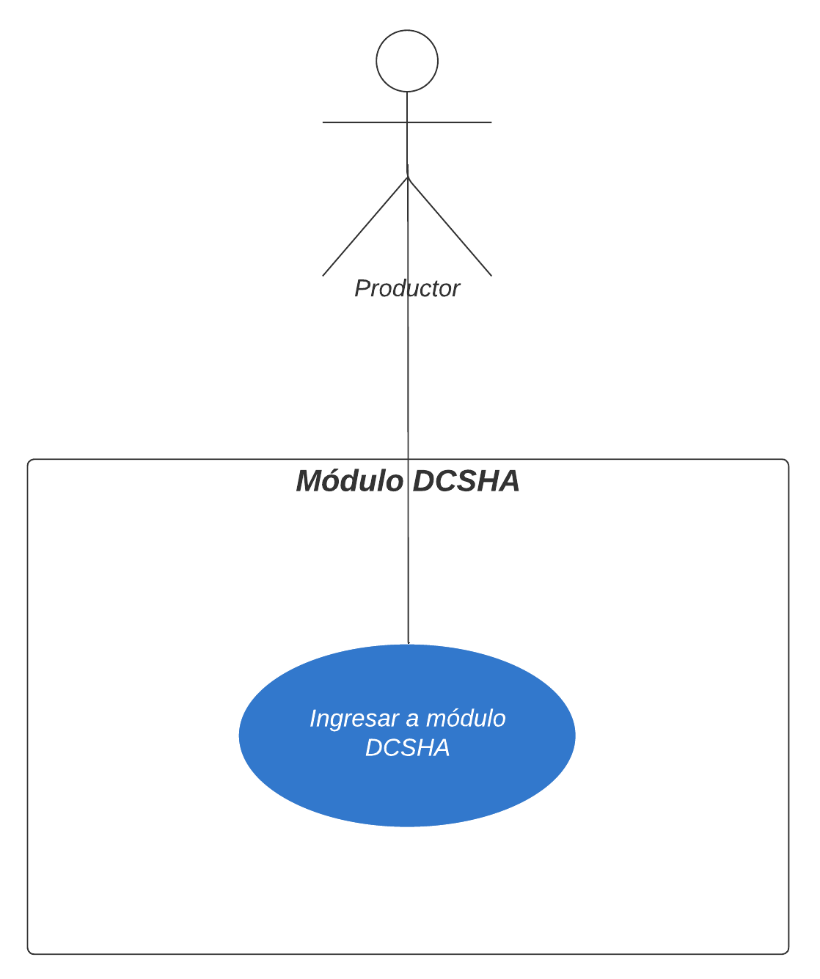


Ilustración 1: Caso de uso 1 | Ingresar a módulo DCSHA

#### Eventos de flujo

* El productor abre la aplicación.
* El productor selecciona el módulo DCSHA.
* El sistema verifica que el módulo DCSHA existe.
* El sistema muestra la interfaz del módulo DCSHA.

#### Diagramas de interacción

* **Actor:**

**Productor:** Representa al usuario que interactúa con la aplicación. En este contexto, el productor es quien inicia el proceso.

* **Participantes:**

**Aplicación (App):** Es el sistema que el productor utiliza. Se encarga de mostrar la interfaz y gestionar las interacciones del usuario.

**Sistema (Sys):** Es un componente que verifica la existencia del módulo solicitado por el productor.

* **Flujo de Interacciones:**

**Paso 1:** El productor inicia la interacción abriendo la aplicación (Productor -> App: Abrir aplicación).

**Paso 2:** La aplicación responde mostrando la interfaz al productor (App -> Productor: Mostrar interfaz).

**Paso 3:** El productor selecciona el módulo DCSHA en la aplicación (Productor -> App: Seleccionar módulo DCSHA).

**Paso 4:** La aplicación envía una solicitud al sistema para verificar si el módulo DCSHA existe (App -> Sys: Verificar existencia del módulo).

**Paso 5:** El sistema responde a la aplicación confirmando la existencia del módulo (Sys -> App: Confirmar existencia).

**Paso 6:** Finalmente, la aplicación muestra la interfaz del módulo DCSHA al productor (App -> Productor: Mostrar interfaz del módulo DCSHA).

* + 1. *Diagrama de secuencia*

El diagrama representa el proceso de un usuario (Productor) que interactúa con una aplicación para acceder a un módulo específico (DCSHA). Este tipo de diagrama es útil para visualizar cómo se llevan a cabo las interacciones en un sistema y cómo se gestionan las solicitudes y respuestas entre los diferentes componentes.

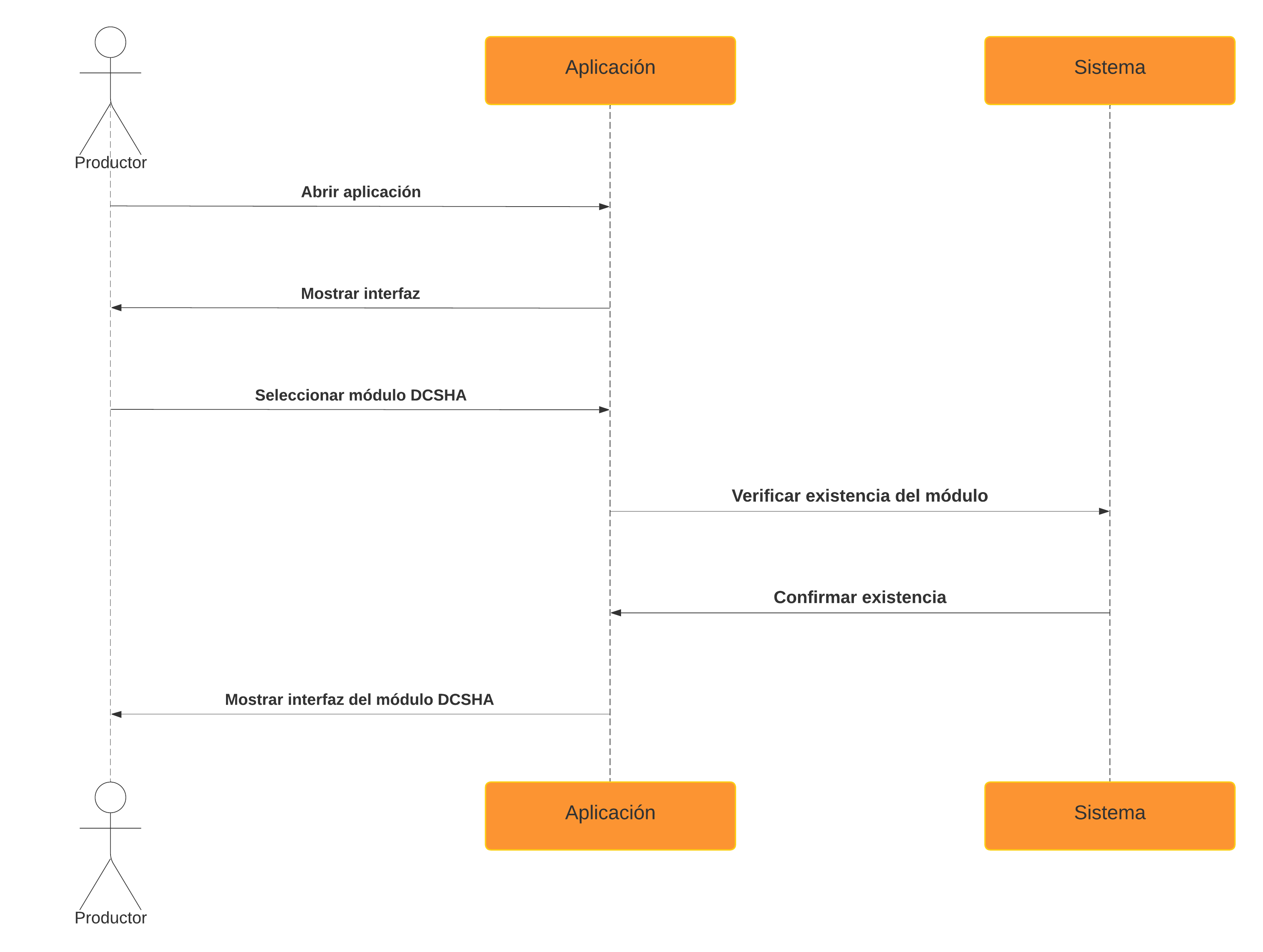


Ilustración 2: Diagrama de secuencia | Caso de uso 1

* **Inicio de la Interacción:**
  + **Productor -> App: Abrir aplicación**
    - El proceso comienza cuando el Productor envía un mensaje a la Aplicación para abrirla. Esto indica que el usuario desea iniciar su trabajo en el sistema.
* **Respuesta de la Aplicación:**
  + **App -> Productor: Mostrar interfaz**
    - La Aplicación responde mostrando su interfaz al Productor. Esto significa que la aplicación ha sido cargada y está lista para ser utilizada.
* **Selección del Módulo:**
  + **Productor -> App: Seleccionar módulo DCSHA**
    - Una vez que la interfaz está visible, el Productor selecciona un módulo específico llamado "DCSHA". Este paso implica que el usuario está interesado en acceder a funcionalidades o datos relacionados con ese módulo.
* **Verificación de Existencia:**
  + **App -> Sys: Verificar existencia del módulo**
    - La Aplicación envía una solicitud al Sistema para verificar si el módulo DCSHA está disponible. Este paso es crucial para asegurarse de que el módulo puede ser utilizado antes de proceder.
* **Confirmación del Sistema:**
  + **Sys -> App: Confirmar existencia**
    - El Sistema responde a la Aplicación confirmando que el módulo DCSHA efectivamente existe. Esta confirmación es necesaria para que la Aplicación pueda continuar con el flujo de trabajo.
* **Mostrar Interfaz del Módulo:**
  + **App -> Productor: Mostrar interfaz del módulo DCSHA**
    - Finalmente, la Aplicación muestra al Productor la interfaz del módulo DCSHA. Esto permite al usuario interactuar con las funcionalidades específicas de ese módulo.

*1.3.2 Diagrama de colaboración*

* Visualizar Interacciones: El diagrama muestra cómo los actores (en este caso, el “Productor”) y los componentes del sistema (la “Aplicación” y el “Sistema”) interactúan entre sí a lo largo del tiempo. Esto ayuda a entender el flujo de mensajes y la secuencia de acciones.
* Describir el Comportamiento del Sistema: Permite documentar y analizar el comportamiento del sistema en respuesta a las acciones del usuario. Es útil para identificar cómo se manejan las solicitudes y las respuestas.
* Facilitar la Comunicación: Sirve como una herramienta de comunicación entre los desarrolladores, diseñadores y otros interesados, proporcionando una representación clara de cómo se espera que funcione el sistema.

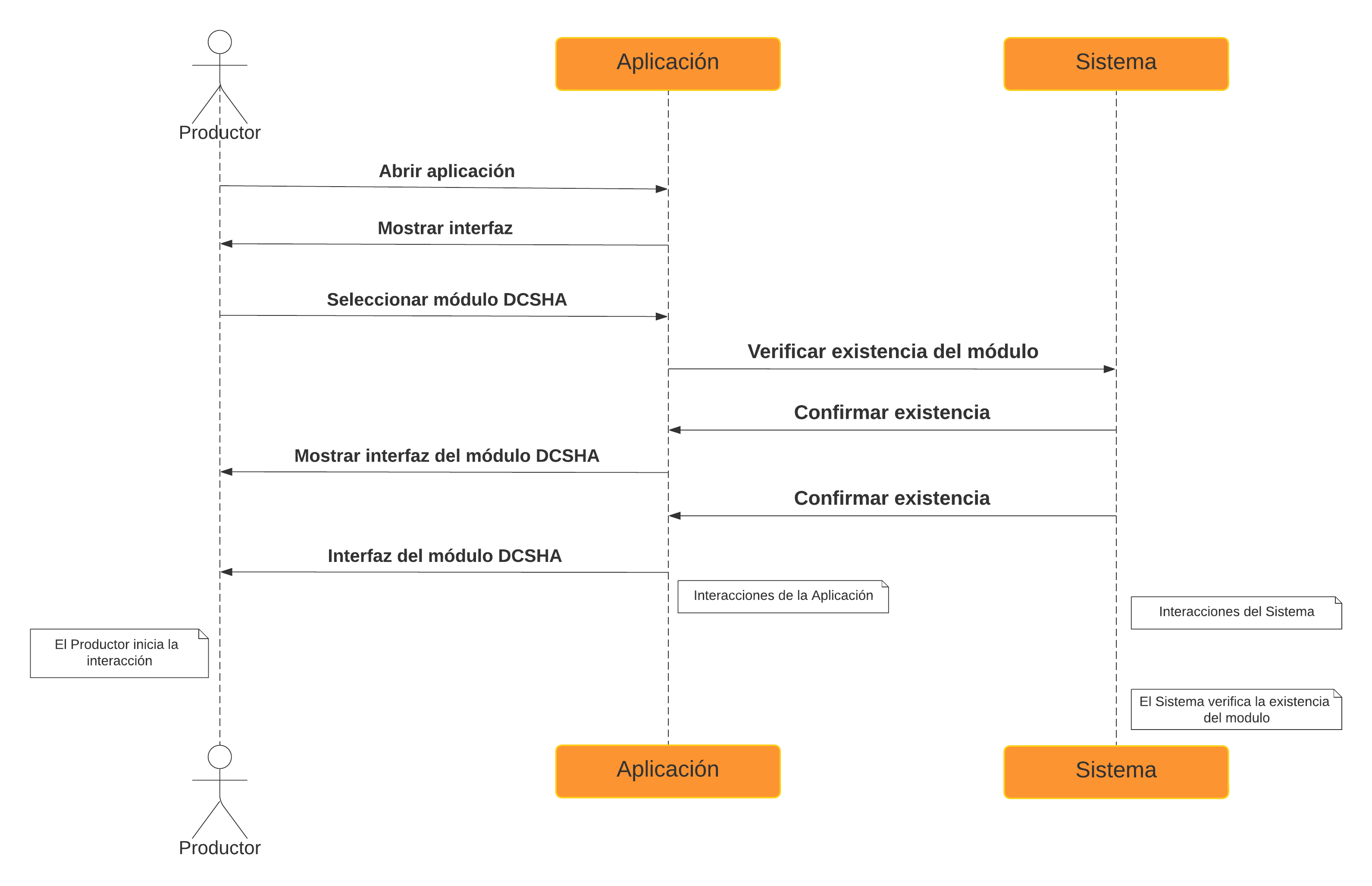


Ilustración 3: Diagrama de colaboración | Caso de uso 1

#### Objetos participantes

* Productor.
* Aplicación.
* Sistema.

#### 1.5 Diagrama de objetos

Este diagrama de objetos ilustra cómo se organiza un sistema en capas, donde cada capa tiene responsabilidades específicas. La **Capa de Presentación** se encarga de la interfaz de usuario, la **Capa de Control** gestiona la lógica de control y la **Capa de Negocio** maneja la lógica de negocio y los datos. Las relaciones entre las clases muestran cómo interactúan y se comunican entre sí, lo que facilita la comprensión de la arquitectura del sistema y su funcionamiento.

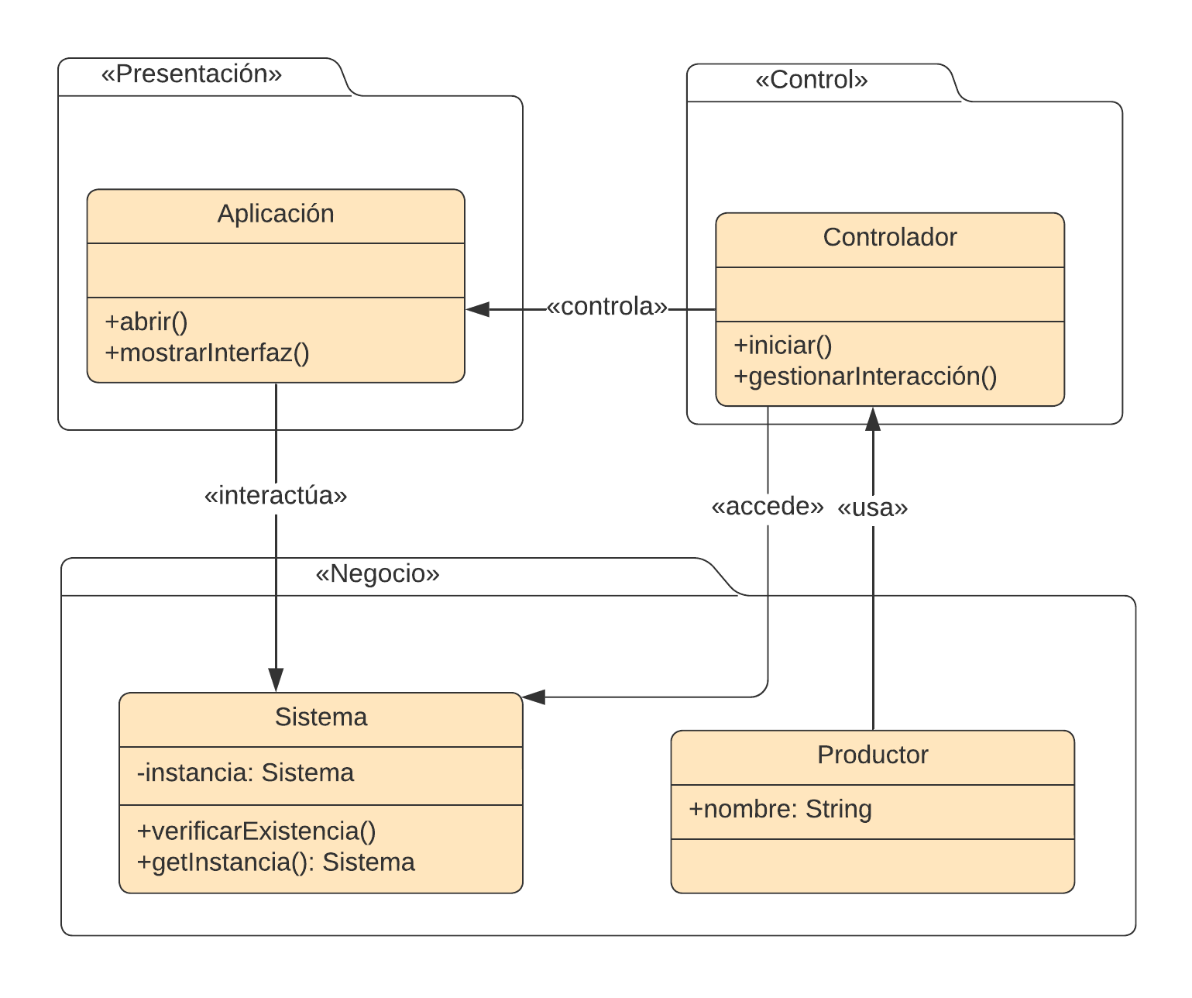


Ilustración 4: Diagrama de objetos | Caso de uso 1

* **Capa de Presentación:**
  + **Clase Aplicación:**
    - **Métodos:**
      * +abrir(): Método público que se utiliza para iniciar la aplicación.
      * +mostrarInterfaz(): Método público que se encarga de mostrar la interfaz de usuario al usuario final.
    - **Función:** Esta clase es responsable de la interacción con el usuario. Se encarga de presentar la información y recibir las entradas del usuario.
* **Capa de Control:**
  + **Clase Controlador:**
    - **Métodos:**
      * +iniciar(): Método público que inicia el proceso de control, inicializando la aplicación o configurando el entorno.
      * +gestionarInteracción(): Método público que maneja las interacciones del usuario, actuando como intermediario entre la interfaz de usuario y la lógica de negocio.
    - **Función:** Esta clase actúa como un intermediario entre la capa de presentación y la capa de negocio. Su responsabilidad principal es gestionar la lógica de control y coordinar las acciones entre las diferentes capas.
* **Capa de Negocio:**
  + **Clase Sistema:**
    - **Métodos:**
      * +verificarExistencia(): Método público que probablemente se utiliza para verificar la existencia de un elemento o entidad en el sistema.
      * -instancia: Sistema: Atributo privado que sugiere que esta clase puede estar implementando el patrón *Singleton*, donde solo puede existir una instancia de Sistema.
      * +getInstancia(): Sistema: Método público que devuelve la única instancia de Sistema.
    - **Función:** Esta clase contiene la lógica de negocio del sistema. Se encarga de las operaciones fundamentales y de la gestión de datos.
  + **Clase Productor:**
    - **Atributos:**
      * +nombre: String: Atributo público que representa el nombre del productor.
    - **Función:** Esta clase representa una entidad del negocio, que puede ser utilizada por el sistema para realizar operaciones relacionadas con los productores.

**Relaciones entre Clases**

* **Productor --> Controlador: <<usa>>:**
  + Indica que la clase Controlador utiliza la clase Productor. Esto sugiere que el controlador necesita acceder a la información o los métodos de Productor para realizar su función.
* **Controlador --> Aplicación: <<controla>>:**
  + Indica que el Controlador controla la Aplicación. Esto significa que el controlador gestiona la lógica de la interfaz de usuario y coordina las acciones que se deben realizar en respuesta a las interacciones del usuario.
* **Controlador --> Sistema: <<accede>>:**
  + Indica que el Controlador accede a la clase Sistema. Esto implica que el controlador puede invocar métodos de Sistema para realizar operaciones de negocio.
* **Aplicación --> Sistema: <<interactúa>>:**
  + Indica que la Aplicación interactúa con Sistema. Esto sugiere que la interfaz de usuario puede llamar a métodos de Sistema para obtener datos o realizar acciones.

### CASO DE USO <Seleccionar plantilla>

#### Breve descripción

El productor selecciona una plantilla base, una plantilla del día anterior o importa de Histórico. Este caso de uso permite al productor elegir la plantilla adecuada para su trabajo.

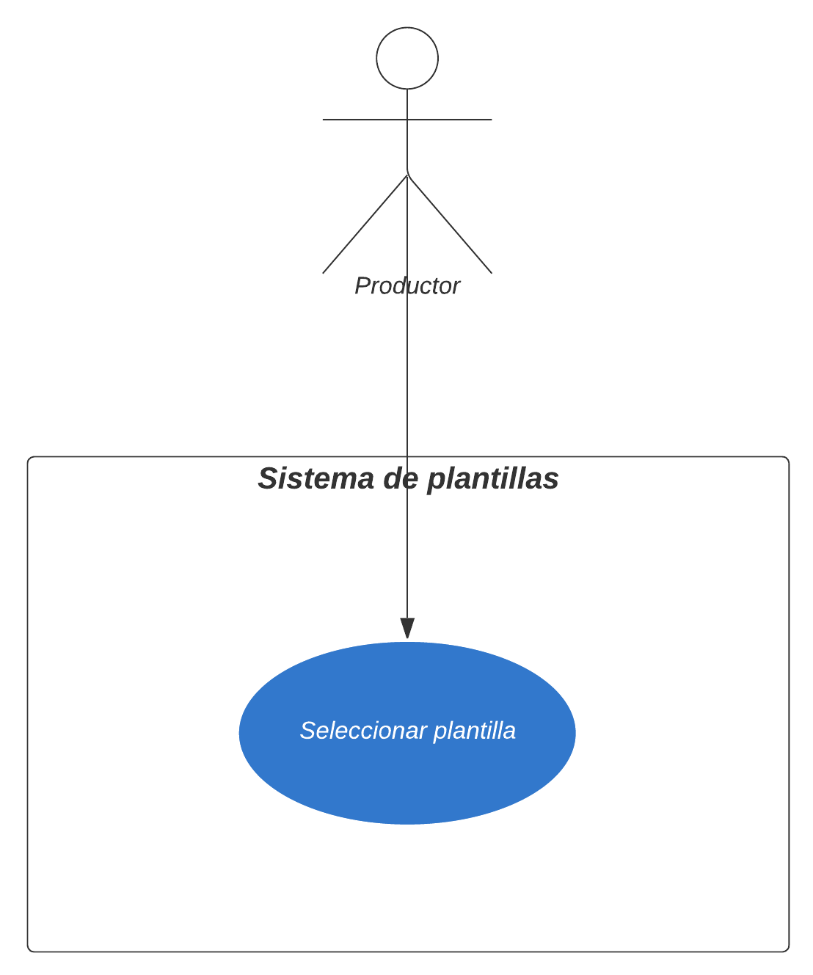


Ilustración 5: Caso de uso 2 | Seleccionar plantilla

#### Eventos de flujo

* El productor selecciona la opción de plantilla base, plantilla del día anterior o importar de Histórico.
* El sistema verifica la disponibilidad de la opción seleccionada.
* El sistema carga la plantilla seleccionada.

#### Diagramas de interacción

* Actor:
  + Productor: Representa a un usuario o entidad externa que interactúa con el sistema. En este contexto, el productor es quien selecciona y utiliza las plantillas disponibles en el sistema.
* Participante:
  + Sistema de Plantillas (Plantillas): Representa el sistema que gestiona las plantillas. Este sistema es responsable de ofrecer opciones al productor y cargar la plantilla seleccionada.

Interacciones en el Diagrama

* Productor -> Plantillas: Seleccionar plantilla:
  + El productor inicia la interacción al seleccionar una plantilla del sistema. Este es el primer paso en el proceso, donde el productor expresa su intención de trabajar con una plantilla específica.
* Plantillas -> Productor: Mostrar opciones:
  + En respuesta a la selección del productor, el sistema de plantillas muestra las opciones disponibles. Esto implica que el sistema presenta al productor una lista o un conjunto de plantillas que puede elegir.
* Productor -> Plantillas: Elegir opción:
  + Después de revisar las opciones, el productor elige una de las plantillas disponibles. Este paso indica que el productor ha tomado una decisión sobre qué plantilla utilizar.
* Plantillas -> Productor: Cargar plantilla:
  + Finalmente, el sistema de plantillas carga la plantilla seleccionada y la presenta al productor. Este paso completa el proceso, permitiendo al productor acceder a la plantilla elegida para su uso.
    1. *Diagrama de secuencia*

En este diagrama se ilustra la interacción entre un actor (en este caso, el “Productor”) y un sistema (denominado “Sistema de Plantillas”). Este tipo de diagrama se utiliza para mostrar cómo los objetos interactúan en un sistema a lo largo del tiempo, destacando el orden de los mensajes intercambiados.

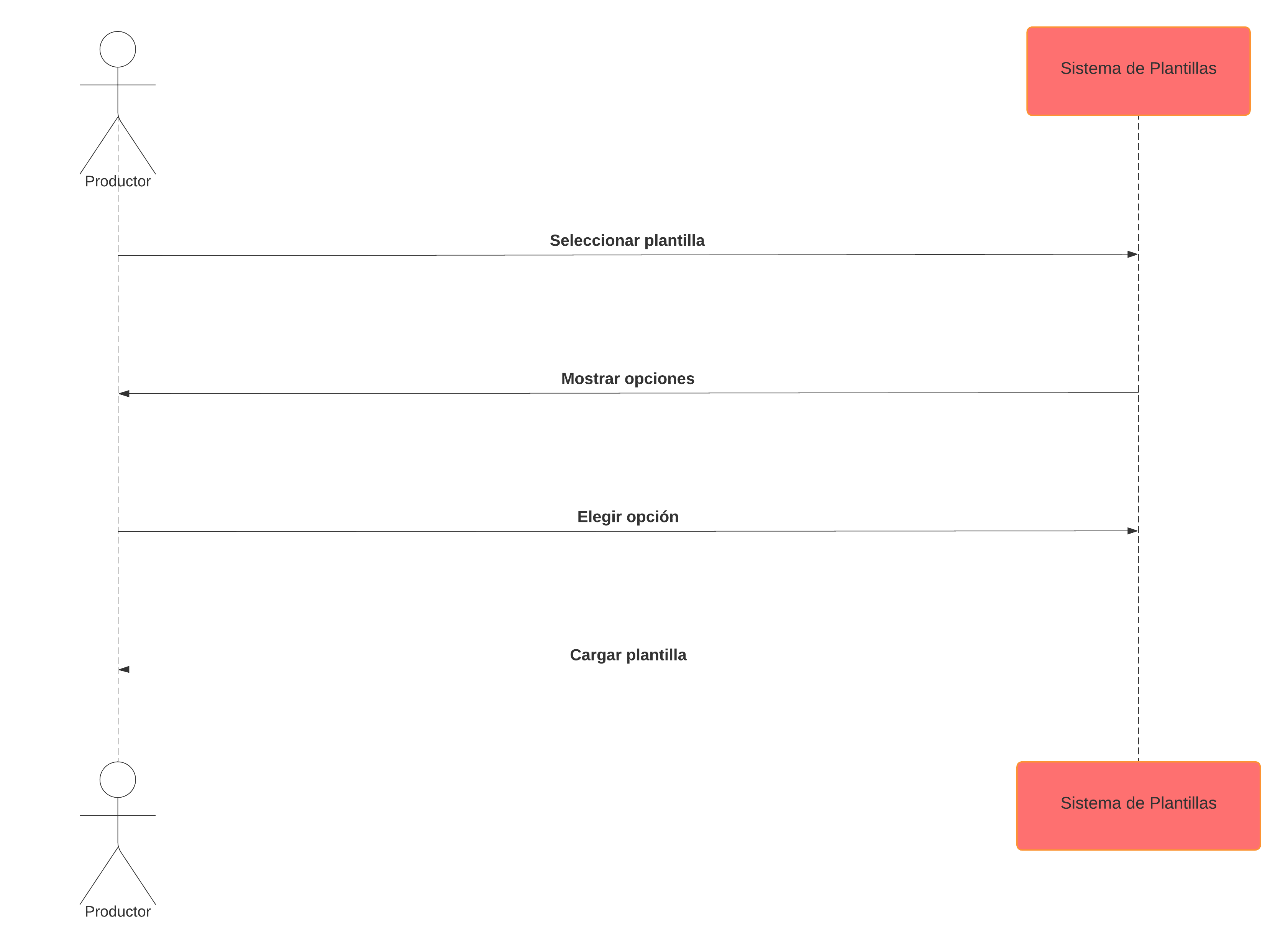


Ilustración 6: Diagrama de secuencia | Caso de uso 2

#### Objetos participantes

* Productor.
* Sistema de plantillas

#### 2.5 Diagrama de objetos

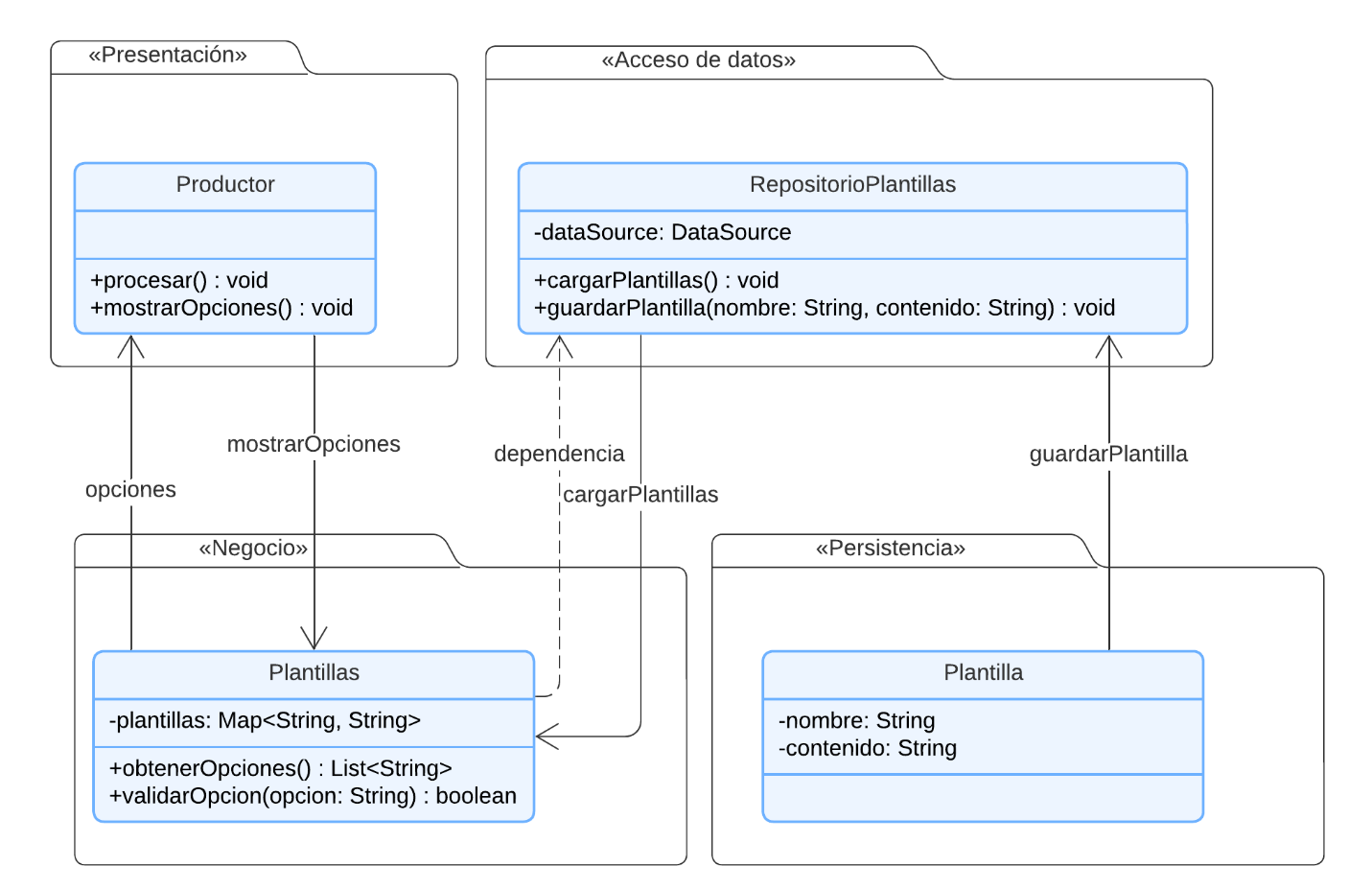


Ilustración 7: Diagrama de objetos | Caso de uso 2

* **Capa de Presentación:**
  + **Productor:**
    - Métodos:
      * procesar(): void: Método que inicia el proceso de selección de plantillas.
      * mostrarOpciones(): void: Método que permite al productor ver las opciones de plantillas disponibles.
* **Capa de Negocio:**
  + **Plantillas:**
    - Atributos:
      * plantillas: Map <String, String>: Atributo que llama al entorno de plantillas disponibles.
    - Métodos:
      * seleccionar(): void: Método que permite al productor seleccionar una plantilla.
      * obtenerOpciones(): List<String>: Método que devuelve una lista de opciones de plantillas disponibles para el productor.
      * validarOpcion(opción: String): boolean: Método que valida la selección de la plantilla.
* **Capa de Acceso a Datos:**
  + **RepositorioPlantillas:**
    - Atributos:
      * dataSource: DataSource: Atributo que llama la conexión a la base de datos del contenido de las plantillas disponibles.
    - Métodos:
      * obtenerPlantillas(): List<String>: Método que obtiene la lista de plantillas desde la base de datos o fuente de datos.
      * guardarPlantilla(plantilla: String): void: Método que guarda una plantilla en la base de datos o fuente de datos.
* **Capa de Persistencia:**
  + **Plantilla:**
    - Atributos:
      * nombre: String: Atributo que permite el almacenamiento del nombre de la plantilla.
      * contenido: String: Atributo que permite el almacenamiento del contenido de la plantilla.

**Relaciones entre Clases**

* **Relación entre Productor y Plantillas:**

**Tipo de relación: Asociación**

**Descripción:**El Productor inicia la interacción con la clase Plantillas al llamar al método obtenerOpciones(). Esta relación indica que el Productor depende de la clase Plantillas para realizar su tarea de selección de plantillas. En términos de diseño, esto significa que el Productor tiene conocimiento de la existencia de Plantillas y puede invocar sus métodos para interactuar con el sistema de plantillas.

* **Relación entre Plantillas y RepositorioPlantillas:**

**Tipo de relación: Asociación**

**Descripción:**La clase Plantillas se comunica con la clase RepositorioPlantillas/cargarPlantillas(): void para obtener la lista de plantillas disponibles mediante el método obtenerOpciones(). Esta relación indica que Plantillas necesita acceder a los datos almacenados en RepositorioPlantillas para poder mostrar las opciones al Productor. Aquí, Plantillas actúa como un intermediario que utiliza los datos proporcionados por RepositorioPlantillas para cumplir con su funcionalidad.

**Detalle de las Relaciones**

* **Asociación:**

La asociación es una relación estructural que indica que una clase utiliza o tiene una referencia a otra clase. En este caso, tanto Productor como Plantillas están asociados, ya que el Productor necesita interactuar con Plantillas para seleccionar y cargar plantillas. Asimismo, Plantillas está asociada con RepositorioPlantillas para acceder a los datos necesarios.

* **Flujo de Interacción**

El Productor inicia la selección:

Cuando el Productor llama al método mostrarOpciones(), se establece la interacción con la clase Plantillas. Esto puede implicar que el Productor solicita ver las opciones de plantillas disponibles.

* **Plantillas obtiene las opciones:**

La clase Plantillas invoca el método obtenerOpciones() de RepositorioPlantillas para recuperar la lista de plantillas. Esta acción es crucial, ya que Plantillas no tiene los datos por sí misma; depende de RepositorioPlantillas para obtener la información necesaria.

* **Mostrar opciones al Productor:**

Una vez que Plantillas tiene la lista de plantillas, utiliza el método mostrarOpciones() para presentar estas opciones al Productor. Esto completa la interacción inicial.

* **Selección y carga de la plantilla:**

Después de que el Productor elige una opción, llama al método validarOpciones() en Plantillas, que a su vez puede implicar que Plantillas interactúe nuevamente con RepositorioPlantillas para cargar la plantilla seleccionada, utilizando el método cargarPlantillas() y; luego, la función de salvado va almacenando los datos en plantilla, en la capa de Persistencia.

### CASO DE USO <Editar escaleta>

#### Breve descripción

El productor edita una escaleta con una estructura similar a Excel. Este caso de uso permite al productor modificar la escaleta según sus necesidades.

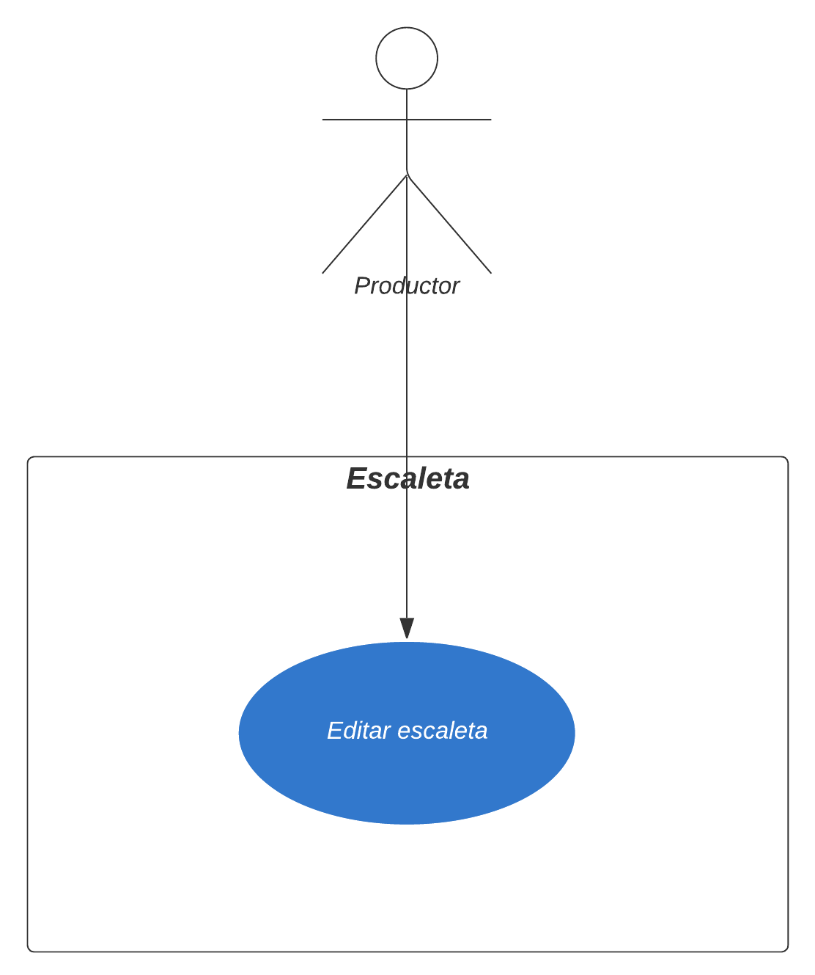


Ilustración 8: Caso de uso 3 | Editar escaleta

#### Eventos de flujo

* El sistema despliega una ventana con la escaleta editable.
* El productor asigna nombres a las filas.
* El productor define el tipo de filas.
* El productor asigna un ID a cada fila.

#### Diagramas de interacción

* Actor:
  + Productor: Representa a un usuario o entidad externa que interactúa con el sistema. En este contexto, el productor es quien selecciona y utiliza las plantillas disponibles en el sistema.
* Participante:
  + Sistema de Plantillas (Plantillas): Representa el sistema que gestiona las plantillas. Este sistema es responsable de ofrecer opciones al productor y cargar la plantilla seleccionada.

Interacciones en el Diagrama

* Productor -> Plantillas: Seleccionar plantilla:
  + El productor inicia la interacción al seleccionar una plantilla del sistema. Este es el primer paso en el proceso, donde el productor expresa su intención de trabajar con una plantilla específica.
* Plantillas -> Productor: Mostrar opciones:
  + En respuesta a la selección del productor, el sistema de plantillas muestra las opciones disponibles. Esto implica que el sistema presenta al productor una lista o un conjunto de plantillas que puede elegir.
* Productor -> Plantillas: Elegir opción:
  + Después de revisar las opciones, el productor elige una de las plantillas disponibles. Este paso indica que el productor ha tomado una decisión sobre qué plantilla utilizar.
* Plantillas -> Productor: Cargar plantilla:
  + Finalmente, el sistema de plantillas carga la plantilla seleccionada y la presenta al productor. Este paso completa el proceso, permitiendo al productor acceder a la plantilla elegida para su uso.
    1. *Diagrama de secuencia*

El Este diagrama ilustra la interacción entre un actor, denominado “Productor”, y un componente del sistema llamado “Escaleta”. Este diagrama representa el flujo de acciones que el Productor realiza al interactuar con la Escaleta, que es una herramienta utilizada para la planificación y organización de contenido.

En este escenario, el Productor inicia la interacción abriendo la escaleta, lo que provoca que la Escaleta muestre una versión editable al Productor. A partir de ahí, el Productor puede realizar varias acciones, como asignar nombres a las filas, definir el tipo de filas y asignar identificadores (ID) a cada fila. Este flujo de trabajo es fundamental para la correcta configuración y uso de la escaleta, permitiendo al Productor organizar la información de manera efectiva.

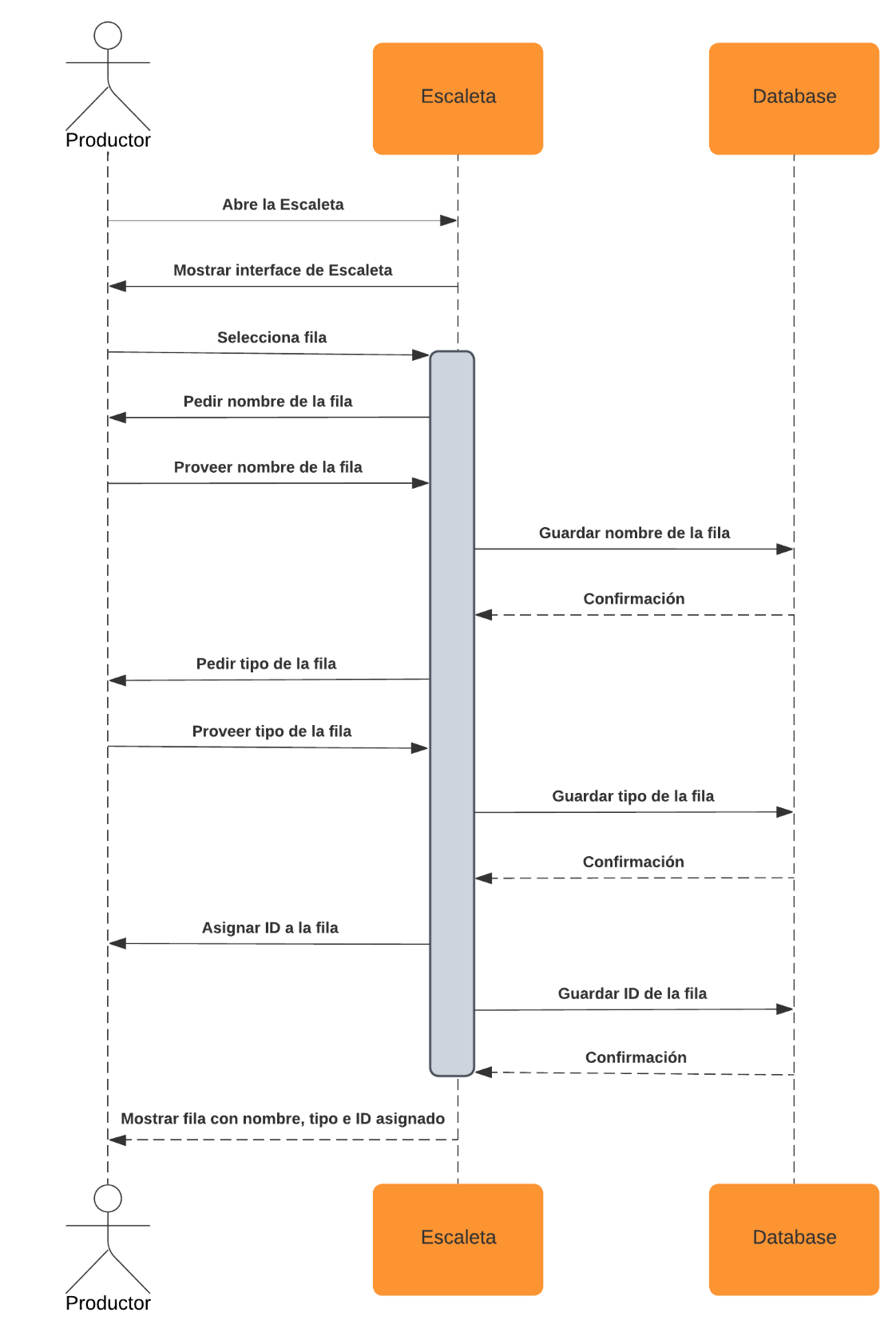


Ilustración 9: Diagrama de secuencia | Caso de uso 2

#### Objetos participantes

* Productor.
* Sistema de plantillas

#### Diagrama de objetos

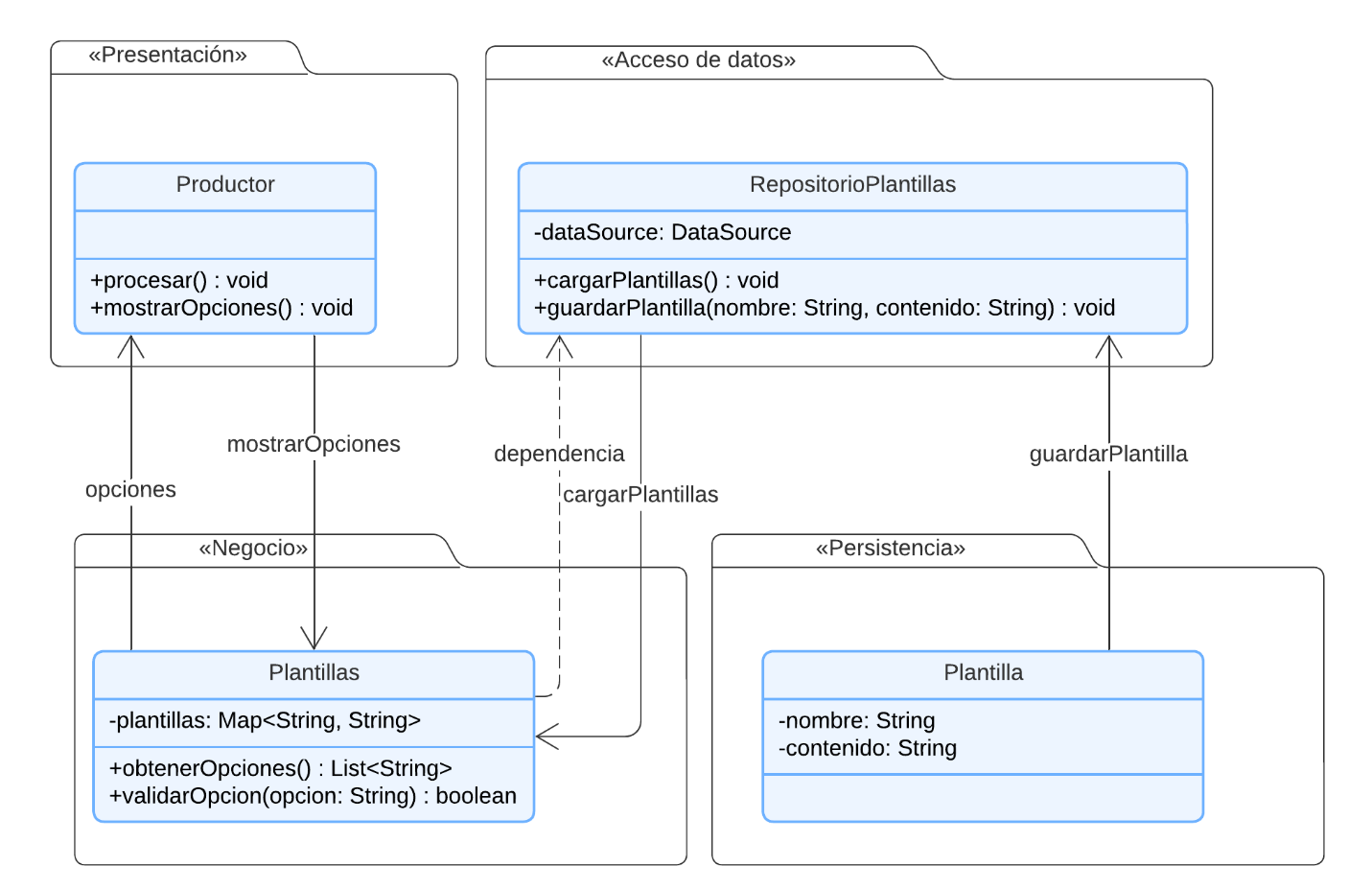


Ilustración 10: Diagrama de objetos | Caso de uso 2

* **Capa de Presentación:**
  + **Productor:**
    - Métodos:
      * procesar(): void: Método que inicia el proceso de selección de plantillas.
      * mostrarOpciones(): void: Método que permite al productor ver las opciones de plantillas disponibles.
* **Capa de Negocio:**
  + **Plantillas:**
    - Atributos:
      * plantillas: Map <String, String>: Atributo que llama al entorno de plantillas disponibles.
    - Métodos:
      * seleccionar(): void: Método que permite al productor seleccionar una plantilla.
      * obtenerOpciones(): List<String>: Método que devuelve una lista de opciones de plantillas disponibles para el productor.
      * validarOpcion(opción: String): boolean: Método que valida la selección de la plantilla.
* **Capa de Acceso a Datos:**
  + **RepositorioPlantillas:**
    - Atributos:
      * dataSource: DataSource: Atributo que llama la conexión a la base de datos del contenido de las plantillas disponibles.
    - Métodos:
      * obtenerPlantillas(): List<String>: Método que obtiene la lista de plantillas desde la base de datos o fuente de datos.
      * guardarPlantilla(plantilla: String): void: Método que guarda una plantilla en la base de datos o fuente de datos.

**Relaciones entre Clases**

* **Relación entre Productor y Plantillas:**

**Tipo de relación: Asociación**

**Descripción:**El Productor inicia la interacción con la clase Plantillas al llamar al método obtenerOpciones(). Esta relación indica que el Productor depende de la clase Plantillas para realizar su tarea de selección de plantillas. En términos de diseño, esto significa que el Productor tiene conocimiento de la existencia de Plantillas y puede invocar sus métodos para interactuar con el sistema de plantillas.

* **Relación entre Plantillas y RepositorioPlantillas:**

**Tipo de relación: Asociación**

**Descripción:**La clase Plantillas se comunica con la clase RepositorioPlantillas/cargarPlantillas(): void para obtener la lista de plantillas disponibles mediante el método obtenerOpciones(). Esta relación indica que Plantillas necesita acceder a los datos almacenados en RepositorioPlantillas para poder mostrar las opciones al Productor. Aquí, Plantillas actúa como un intermediario que utiliza los datos proporcionados por RepositorioPlantillas para cumplir con su funcionalidad.

**Detalle de las Relaciones**

* **Asociación:**

La asociación es una relación estructural que indica que una clase utiliza o tiene una referencia a otra clase. En este caso, tanto Productor como Plantillas están asociados, ya que el Productor necesita interactuar con Plantillas para seleccionar y cargar plantillas. Asimismo, Plantillas está asociada con RepositorioPlantillas para acceder a los datos necesarios.

* **Flujo de Interacción**

El Productor inicia la selección:

Cuando el Productor llama al método mostrarOpciones(), se establece la interacción con la clase Plantillas. Esto puede implicar que el Productor solicita ver las opciones de plantillas disponibles.

* **Plantillas obtiene las opciones:**

La clase Plantillas invoca el método obtenerOpciones() de RepositorioPlantillas para recuperar la lista de plantillas. Esta acción es crucial, ya que Plantillas no tiene los datos por sí misma; depende de RepositorioPlantillas para obtener la información necesaria.

* **Mostrar opciones al Productor:**

Una vez que Plantillas tiene la lista de plantillas, utiliza el método mostrarOpciones() para presentar estas opciones al Productor. Esto completa la interacción inicial.

* **Selección y carga de la plantilla:**

Después de que el Productor elige una opción, llama al método validarOpciones() en Plantillas, que a su vez puede implicar que Plantillas interactúe nuevamente con RepositorioPlantillas para cargar la plantilla seleccionada, utilizando el método cargarPlantillas() y; luego, la función de salvado va almacenando los datos en plantilla, en la capa de Persistencia.

### CASO DE USO <Insertar ID>

#### Breve descripción

El productor inserta ID de forma alfanumérica en la escaleta. Este caso de uso permite al productor identificar cada fila de manera única.

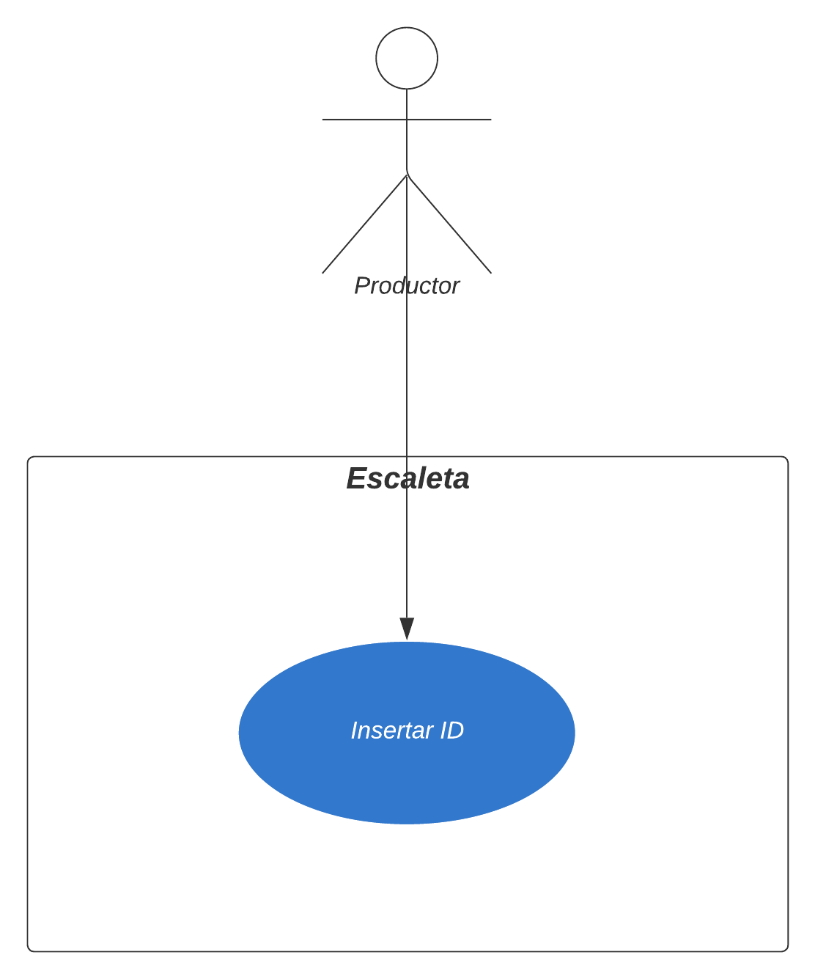


Ilustración 11: Caso de uso 4 | Insertar ID

#### Eventos de flujo

* El productor inserta un ID en una fila.
* El sistema verifica si el ID es repetido.
* Si el ID es repetido, el sistema muestra un mensaje de aviso.

#### Diagramas de interacción

En este diagrama se refleja una serie de interacciones entre el actor “Productor” y el componente “Escaleta”. A continuación, se describen las interacciones que se llevan a cabo en este flujo:

* **Asignar Nombres a Filas:**
  + Una vez que la escaleta está visible y editable, el Productor procede a asignar nombres a las filas. Esta acción permite al Productor identificar y organizar el contenido de manera efectiva, facilitando la planificación y la gestión de la información.
* **Definir Tipo de Filas:**
  + A continuación, el Productor define el tipo de filas. Esta interacción sugiere que el Productor puede categorizar las filas según diferentes criterios, lo que puede incluir tipos de contenido, prioridades o cualquier otra clasificación relevante para el proyecto.
* **Asignar ID a Filas:**
  + Finalmente, el Productor asigna identificadores (ID) a las filas. Esta acción es importante para la referencia y el seguimiento de cada fila dentro de la escaleta, asegurando que cada elemento sea único y fácilmente accesible.
    1. *Diagrama de secuencia*

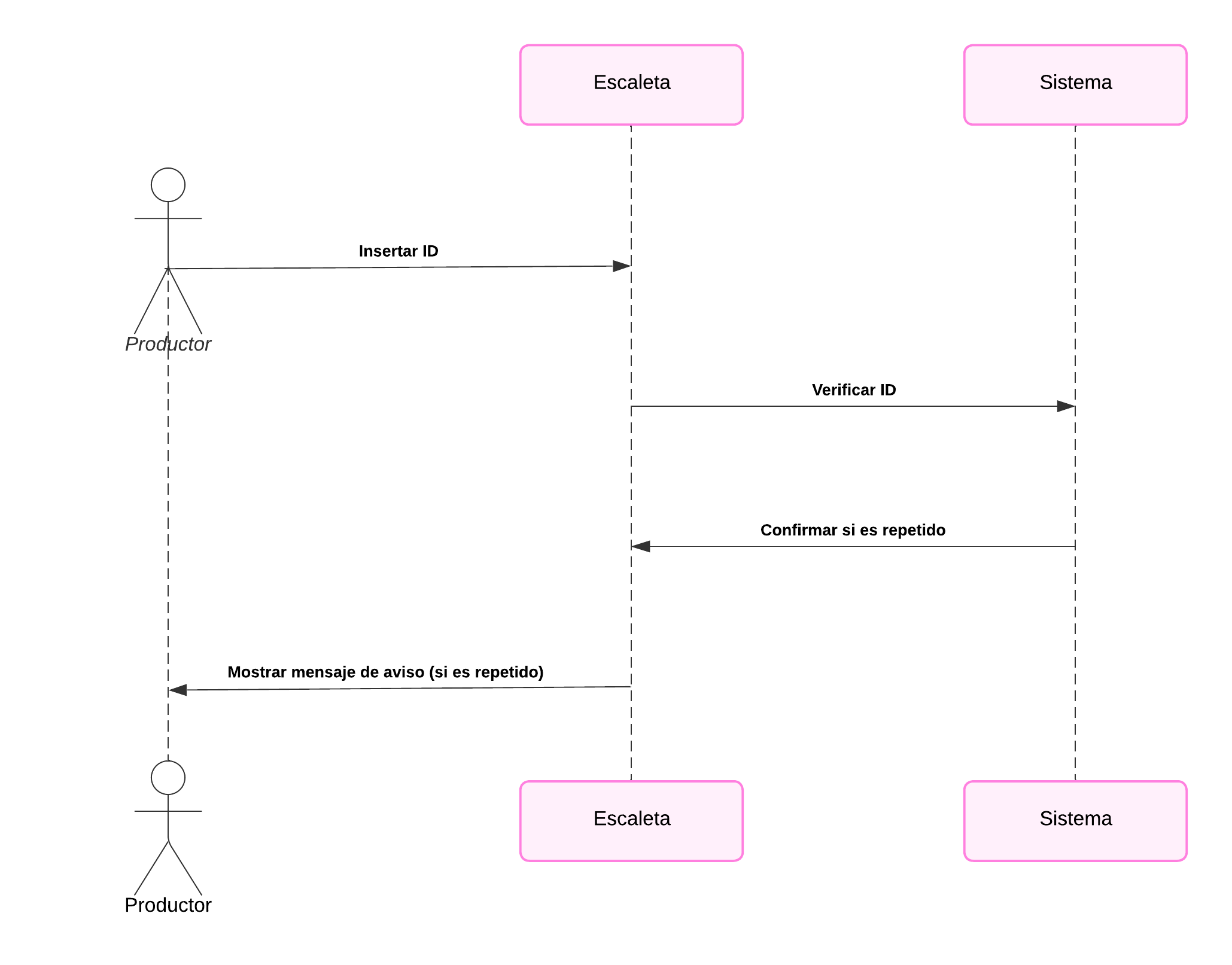


Ilustración 10: Diagrama de secuencia | Caso de uso 4

#### Objetos participantes

* Productor
* Escaleta
* Sistema

#### Diagrama de objetos

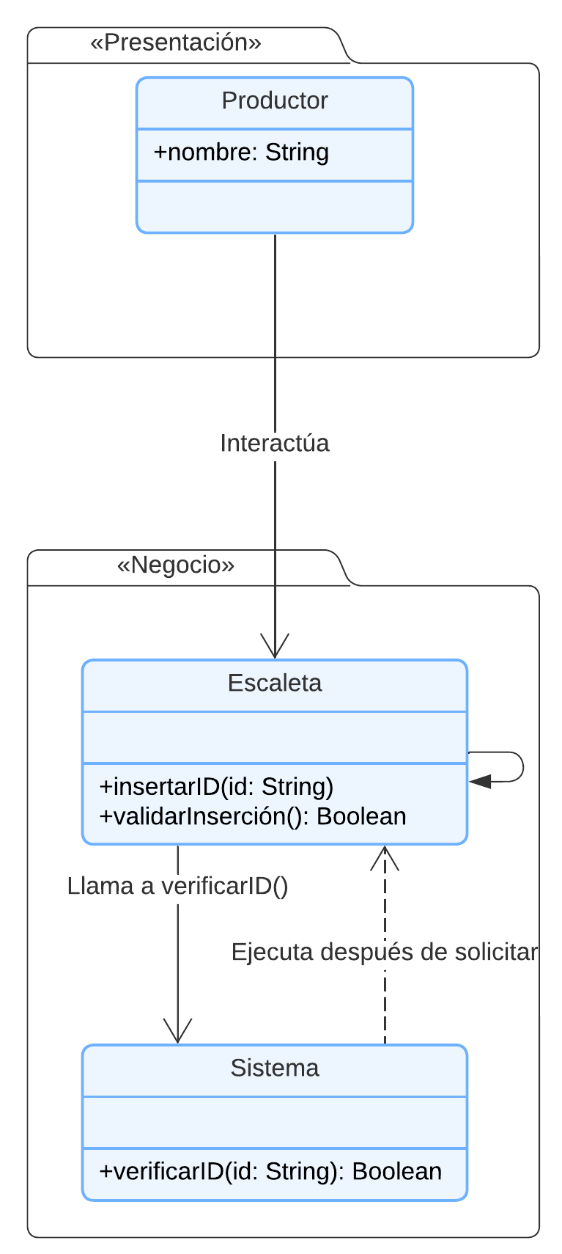


Ilustración 11: Diagrama de objetos | Caso de uso 4

### CASO DE USO <Asignar tiempo total de emisión>

#### Breve descripción

El caso de uso *Asignar tiempo total de emisión* permite al productor establecer el tiempo total que se destinará a la emisión de un bloque específico y a la emisión total. Este tiempo se muestra en el panel superior de la escaleta y se cuenta de manera regresiva durante la emisión.

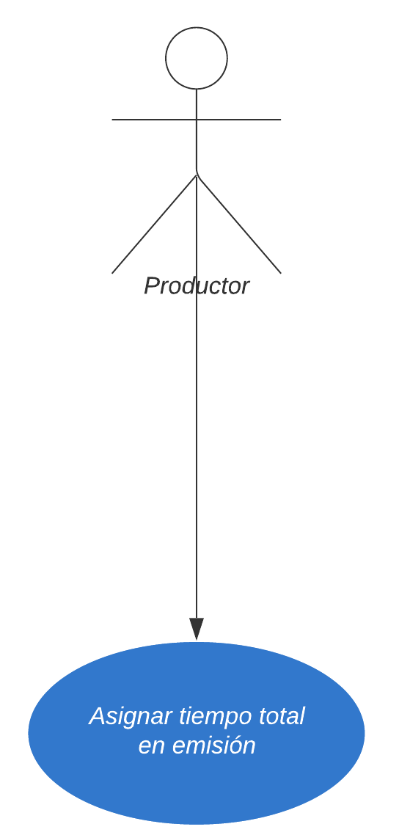


Ilustración 12: Caso de uso 5 | Asignar tiempo total de emisión

#### Eventos de flujo

* El productor asigna el tiempo total de la emisión al final del separador de bloque.
* El sistema muestra el tiempo total en el panel superior de la escaleta.
* Durante la emisión, el sistema muestra el tiempo en regresivo.

#### Diagramas de interacción

Este diagrama la interacción entre el “Productor” y el “Sistema” en el contexto de la asignación y gestión del tiempo. A continuación, se ofrece una descripción detallada de lo que hace el sistema según el diagrama:

* **Actor: Productor**
  + El actor principal en este diagrama es el “Productor”, que representa a un usuario o entidad que interactúa con el sistema para gestionar el tiempo.
* **Interacción Inicial: Asignar tiempo total**
  + El Productor envía un mensaje al Sistema para **asignar un tiempo total**. Este paso implica que el Productor especifica la duración que desea establecer para una cuenta regresiva o un temporizador.
* **Mostrar tiempo total en panel**
  + Una vez que el Sistema recibe la solicitud de asignación de tiempo, realiza una acción interna para **mostrar el tiempo total en un panel**. Esto sugiere que el Sistema tiene una interfaz de usuario donde se visualiza el tiempo asignado, permitiendo al Productor verificar que la duración se ha establecido correctamente.
* **Iniciar cuenta regresiva**
  + Finalmente, el Sistema inicia una **cuenta regresiva**. Este paso indica que, tras la asignación y visualización del tiempo, el Sistema comienza a contar hacia atrás desde el tiempo total asignado, lo que puede ser útil para tareas que requieren un límite de tiempo.
    1. *Diagrama de secuencia*

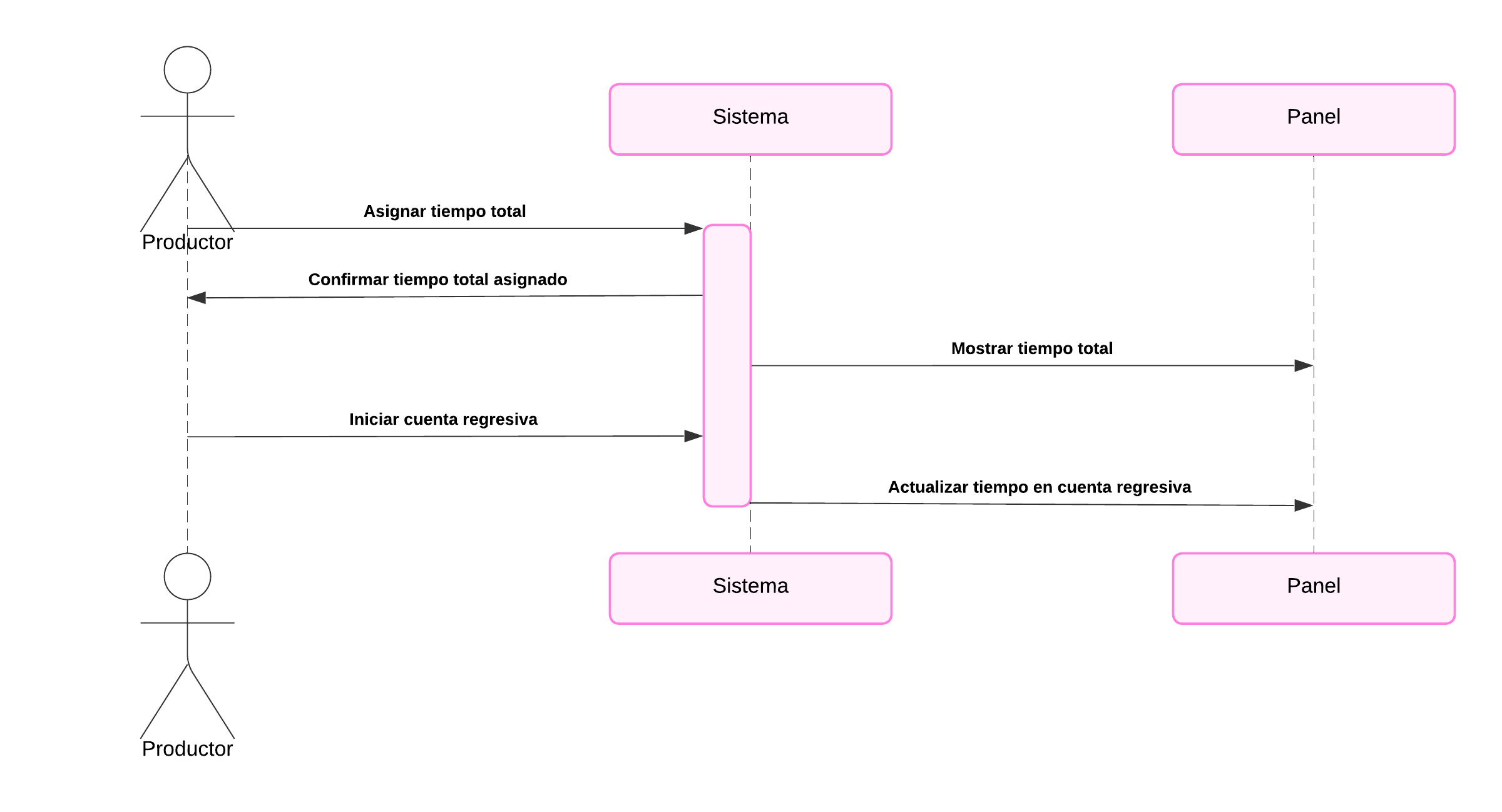


Ilustración 13: Diagrama de secuencia | Caso de uso 5

#### Objetos participantes

* Productor
* Sistema

#### Diagrama de objetos

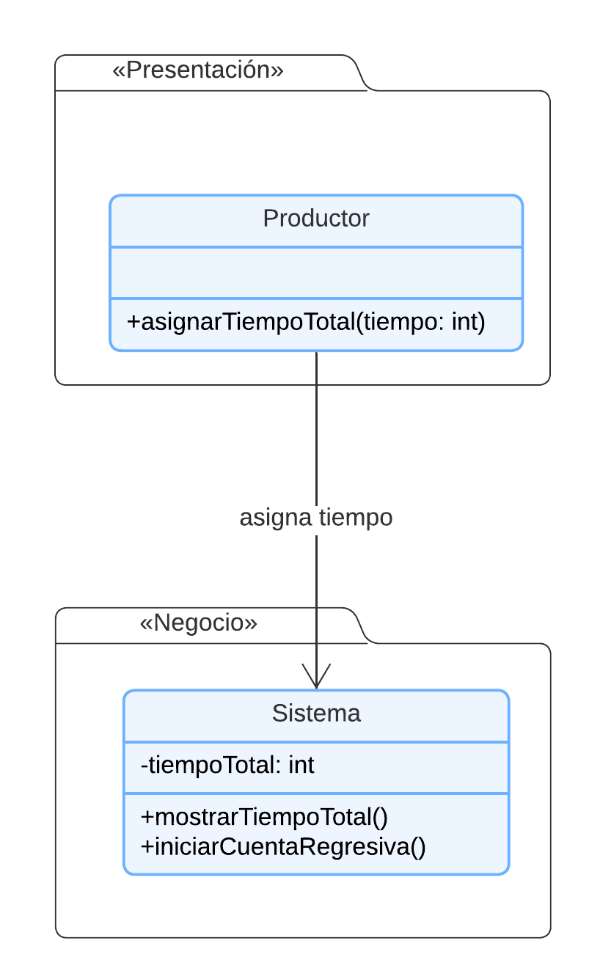


Ilustración 14: Diagrama de objetos | Caso de uso 5

### CASO DE USO <Asignar tiempos a bloques>

#### Breve descripción

El caso de uso *Asignar Tiempos a Bloques* permite al productor asignar tiempos específicos a cada bloque de la escaleta, asegurando que el tiempo total de la emisión se respete.

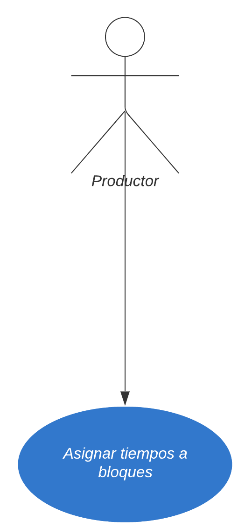


Ilustración 15: Caso de uso 6 | Asignar tiempos a bloques

#### Eventos de flujo

* El productor asigna tiempos a cada bloque.
* El sistema deduce el tiempo asignado del tiempo total de la emisión.
* El sistema muestra el tiempo faltante para cumplir con el tiempo total.

#### Diagramas de interacción

* **Actor: Productor**
  + Representa a un usuario o entidad que interactúa con el sistema. En este contexto, el Productor es responsable de asignar tiempos a diferentes bloques de trabajo o tareas.
* **Participante: Sistema**
  + Representa el sistema que recibe las instrucciones del Productor y realiza las operaciones necesarias para gestionar el tiempo asignado.

**Interacciones Representadas**

1. **Asignar tiempos a bloques**
   * **Interacción:** Productor -> Sistema: Asignar tiempos a bloques
   * **Descripción:** El Productor envía un mensaje al Sistema para asignar tiempos específicos a diferentes bloques o tareas. Esta acción inicia el proceso de gestión del tiempo dentro del sistema.
2. **Deduce tiempo asignado**
   * **Interacción:** Sistema -> Sistema: Deduce tiempo asignado
   * **Descripción:** El Sistema realiza una operación interna para deducir o calcular el tiempo total que ha sido asignado a los bloques. Esto puede implicar sumar los tiempos de cada bloque o realizar otras operaciones lógicas para determinar el tiempo total.
3. **Mostrar tiempo faltante**
   * **Interacción:** Sistema -> Sistema: Mostrar tiempo faltante
   * **Descripción:** El Sistema, tras deducir el tiempo asignado, puede mostrar el tiempo que queda disponible o el tiempo que falta para completar las tareas asignadas. Esta operación es crucial para que el Productor tenga visibilidad sobre el progreso y la gestión del tiempo.

**Representación para el Sistema**

* **Flujo de Trabajo:**
  + El diagrama representa un flujo de trabajo claro donde el Productor interactúa con el Sistema para gestionar el tiempo. Esto es fundamental en aplicaciones donde la gestión del tiempo es crítica, como en la planificación de proyectos, cronogramas o sistemas de control de tareas.
* **Responsabilidades del Sistema:**
  + El Sistema no solo recibe la asignación de tiempos, sino que también realiza cálculos internos y proporciona retroalimentación al Productor sobre el tiempo restante. Esto implica que el Sistema tiene lógica de negocio incorporada para manejar el tiempo de manera efectiva.
* **Interacción Dinámica:**
  + El diagrama muestra una interacción dinámica entre el Productor y el Sistema, lo que sugiere que el sistema es reactivo a las acciones del usuario. Esto es importante para la experiencia del usuario, ya que permite una gestión activa y en tiempo real de las tareas.
    1. *Diagrama de secuencia*

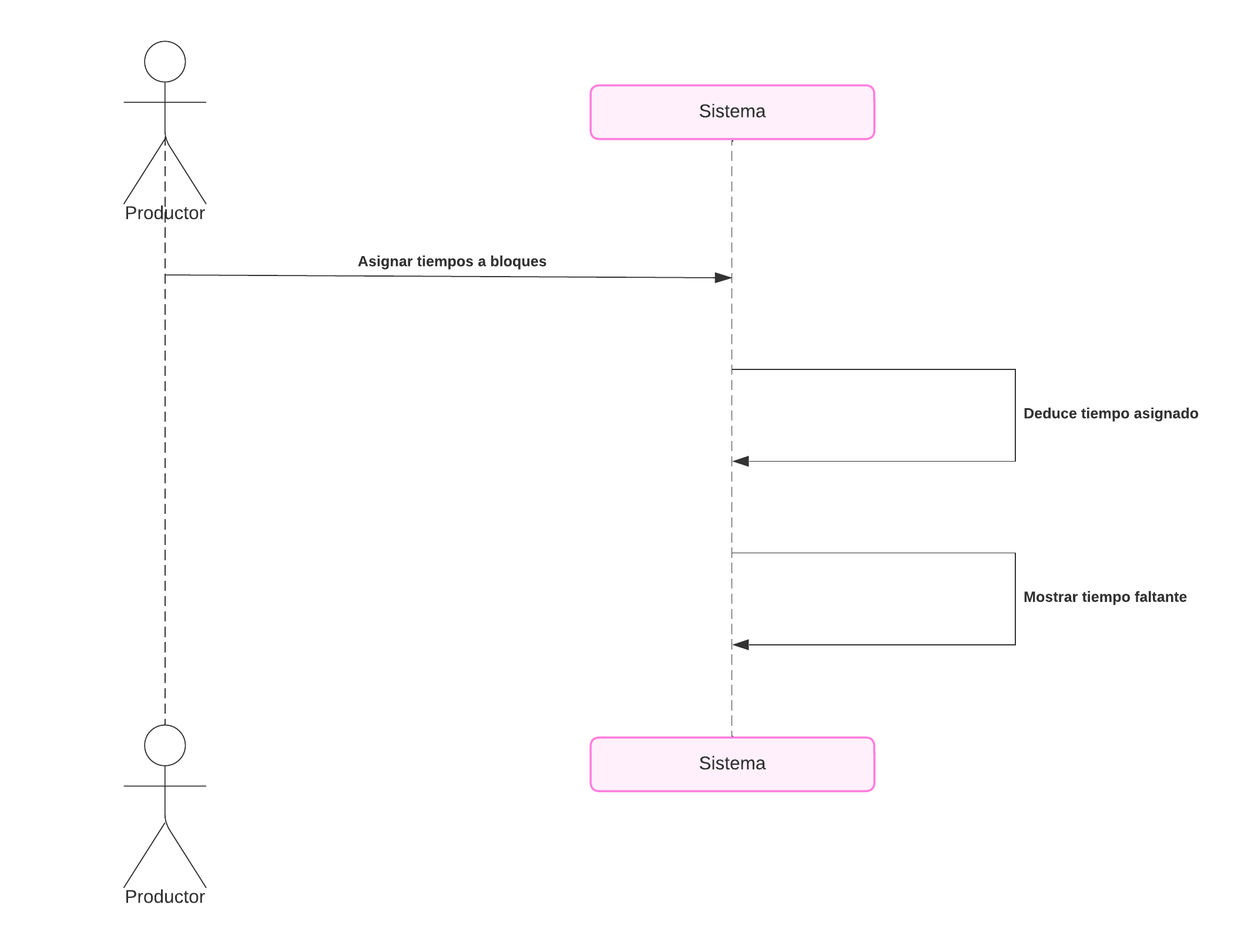


Ilustración 12: Diagrama de secuencia | Caso de uso 6

#### Objetos participantes

* Productor
* Sistema

#### Diagrama de objetos

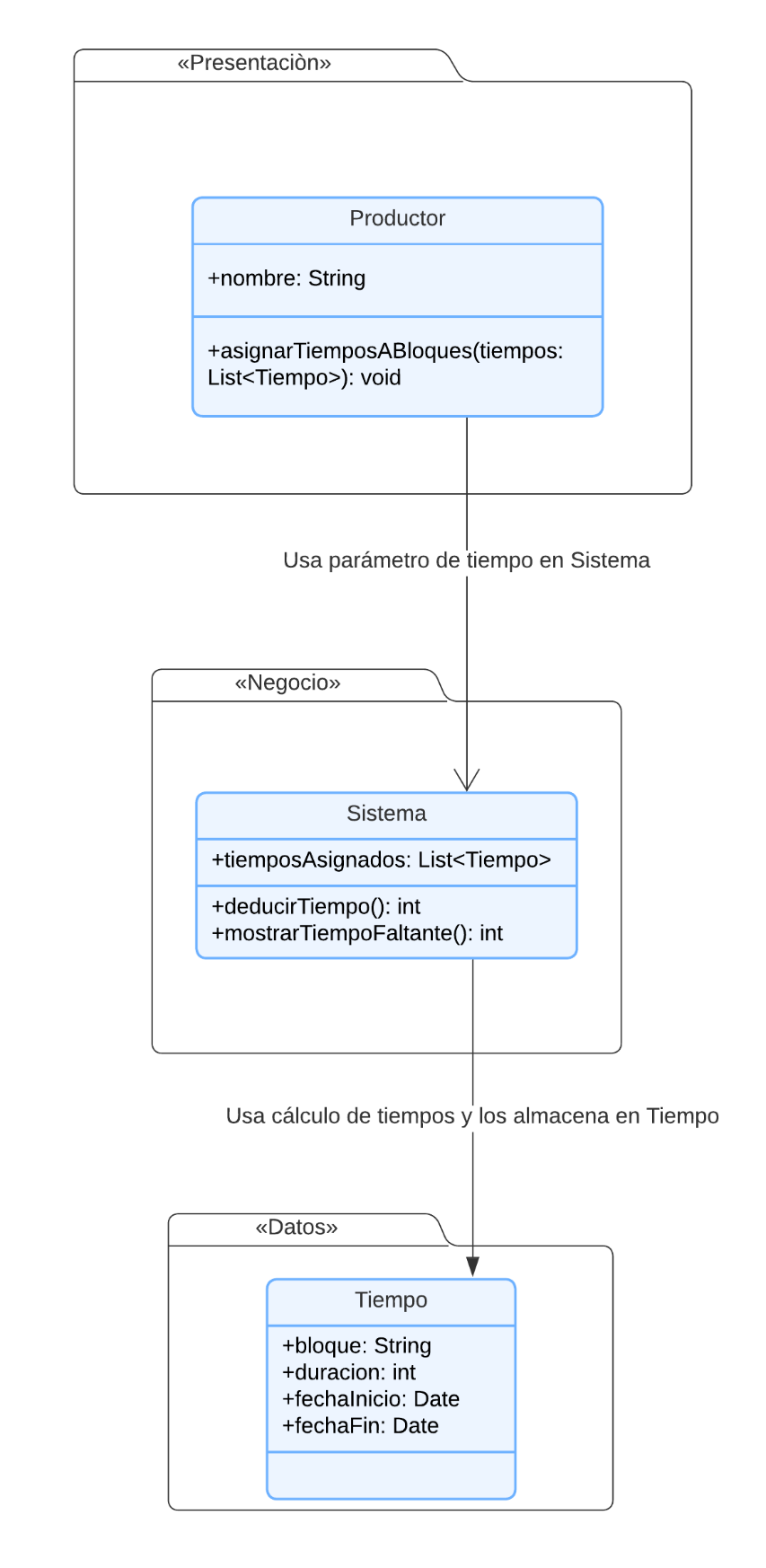


Ilustración 13: Diagrama de objetos | Caso de uso 6

### CASO DE USO <Asignar números de inserción>

#### Breve descripción

El caso de uso *Asignar Números de Inserción* permite al sistema asignar automáticamente números de inserción a cada identidad de los Zócalos de manera ascendente.

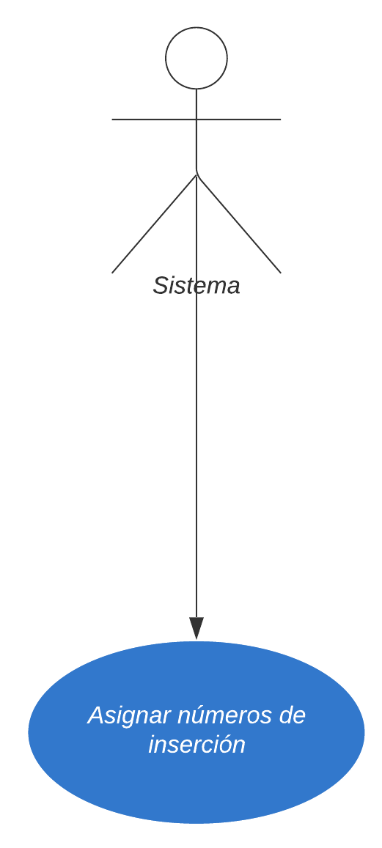


Ilustración 14: Caso de uso 7 | Asignar números de inserción

#### Eventos de flujo

* El sistema asigna números de inserción de forma ascendente a cada identidad de los Zócalos.

#### Diagramas de interacción

* **Inicio de la Asignación:**
  + El sistema inicia el proceso de asignación de números de inserción.
* **Asignación para TITULOS, CATASTROFE y 2 LINEAS:**
  + Para cada uno de estos tipos, el sistema asigna números del 1 al 200.
  + Se itera sobre cada fila, verificando el tema actual y asignando un número de zócalo.
  + Si hay un cambio de fila, el sistema salta un número para dejar un espacio en el Excel.
* **Asignación para VENTAS y TEXTUALES:**
  + El sistema lee las celdas ID y asigna números considerando el número de filas que abarcan (2 para VENTAS y 4 para TEXTUALES).
* **Asignación para PLACAS:**
  + El sistema solo considera los datos de la columna ID.
  + Dependiendo del tipo de placa, se asignan diferentes cantidades de filas:
    - **DATOS y DATOS EQUIPOS:** 12 filas por placa.
    - **COMPARATIVA:** 9 filas por placa.
    - **DATOS:** 6 filas por placa.
    - **PVENTASX3:** 6 filas por placa.
    - **PVENTASX1:** 4 filas por placa.
* **Finalización:**
  + El sistema finaliza el proceso de asignación de números de inserción.
    1. *Diagrama de secuencia*

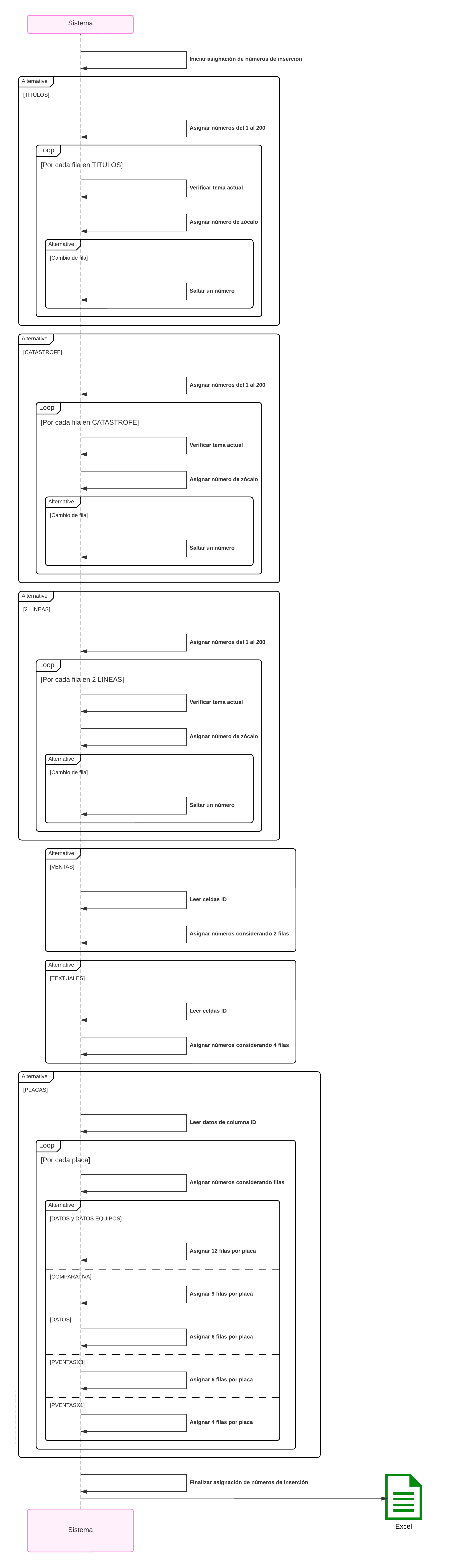


Ilustración 15: Diagrama de secuencia | Caso de uso 7

#### Objetos participantes

* Sistema

#### Diagrama de objetos

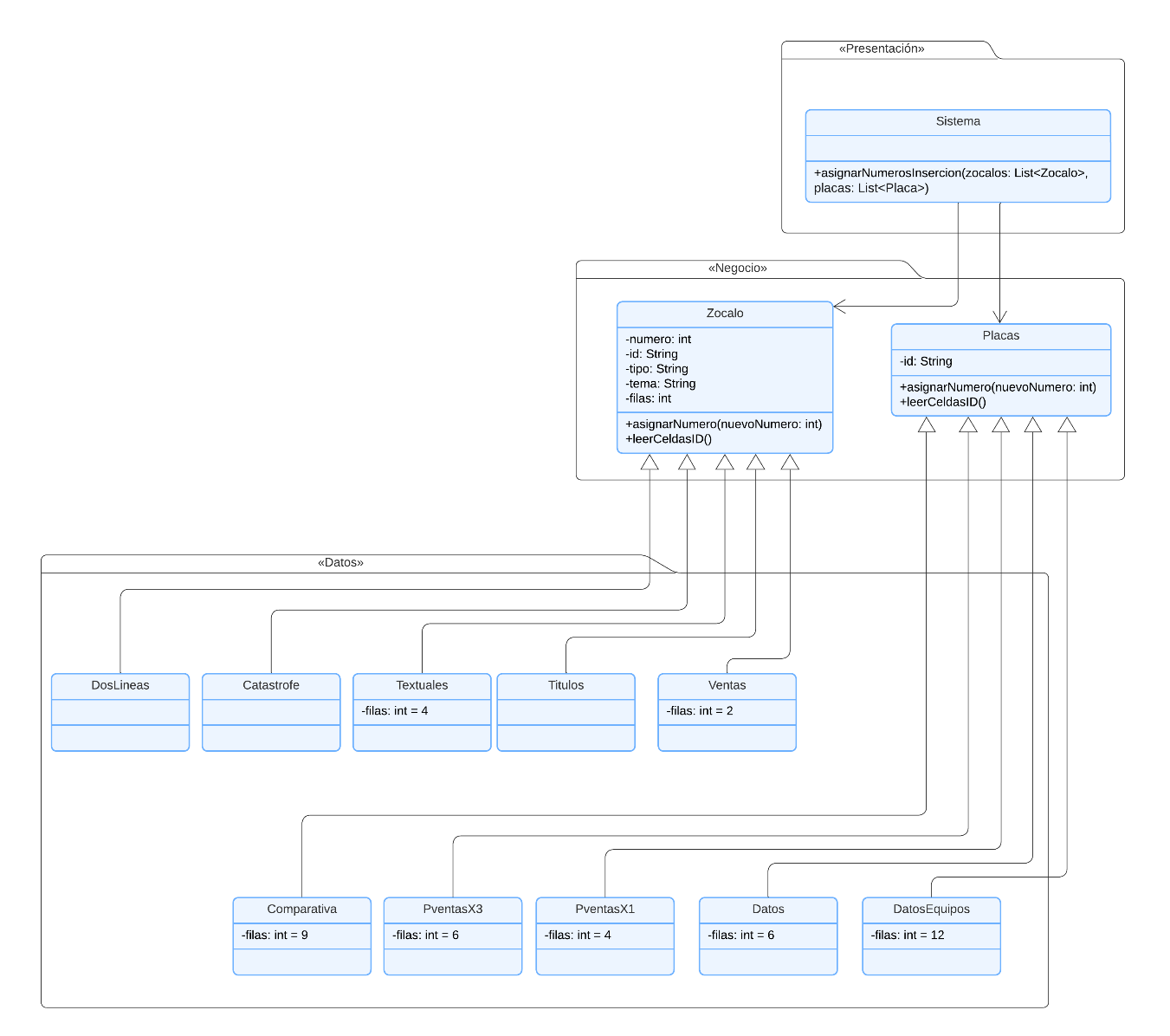


Ilustración 16: Diagrama de objetos | Caso de uso 7

**Capa de Presentación**

* **Clase Sistema**:
  + Método público asignarNumerosInsercion(zocalos: List<Zocalo>, placas: List<Placa>): Este método permite al sistema asignar números a una lista de objetos Zocalo y Placa. Por lo tanto, el sistema tiene la responsabilidad de gestionar la interacción del usuario y la lógica de presentación, actuando como intermediario entre la interfaz de usuario y la lógica de negocio.

**Capa de Negocio**

* **Clase Zocalo**:
  + Atributos privados: numero, id, tipo, tema, filas. Estos atributos definen las características de un zócalo, lo que permite una gestión detallada de cada instancia.
  + Métodos públicos:
    - asignarNumero(nuevoNumero: int): Permite asignar un nuevo número al zócalo.
    - leerCeldasID(): Permite leer los identificadores de las celdas y que el zócalo puede interactuar con datos específicos.
* **Clase Placas**:
  + Atributo privado: id. Esto permite identificar cada placa de manera única.
  + Métodos públicos:
    - leerCeldas(): Permite leer las celdas asociadas a la placa.
    - asignarNumero(nuevoNumero: int): Similar a Zocalo, permite asignar un número a la placa.

**Capa de Datos**

* **Clases de Datos (Titulos, Catastrofe, DosLineas, Ventas, Textuales, Datos, DatosEquipos, Comparativa, PventasX3, PventasX1)**:
  + Estas clases representan diferentes entidades de datos que el sistema puede manejar. Cada una tiene un atributo filas que indica la cantidad de filas que puede gestionar; por lo tanto, están diseñadas para almacenar y gestionar diferentes tipos de información.
  + La clase Ventas tiene un atributo filas inicializado en 2, Textuales en 4, Datos en 6, DatosEquipos en 12, Comparativa en 9, PventasX3 en 6 y PventasX1 en 4, lo que indica que cada clase puede tener un número diferente de filas para gestionar como representación de un único ID asociado a diferente cantidad de filas para cada tipo de zócalo y placa.

**Relaciones**

* **Dependencias**:
  + Sistema --> Zocalo: Indica que la clase Sistema depende de la funcionalidad de la clase Zocalo, lo que significa que el sistema utiliza la lógica de negocio relacionada con los zócalos.
  + Sistema --> Placas: Similar a la relación anterior, el sistema también depende de la clase Placas.
* **Herencia**:
  + Zocalo <|-- [Clases de Datos]: Las clases Titulos, Catastrofe, DosLineas, Ventas, Textuales heredan de Zocalo, lo que sugiere que son tipos específicos de zócalos con características adicionales.
  + Placas --> [Clases de Datos]: Las clases Datos, DatosEquipos, Comparativa, PventasX3, y PventasX1 están relacionadas con Placas, lo que indica que estas clases de datos pueden ser utilizadas o gestionadas por la clase Placas.

### CASO DE USO < Acceder y crear filas>

#### Breve descripción

El caso de uso *Acceder y Crear filas* permite al asistente de producción acceder a las filas de la escaleta y crear nuevas a partir de una celda de inserción seleccionada (la creación de filas está restringidas a VTR/Nota, VTR/Full, Entrada y Placa), según sea necesario.

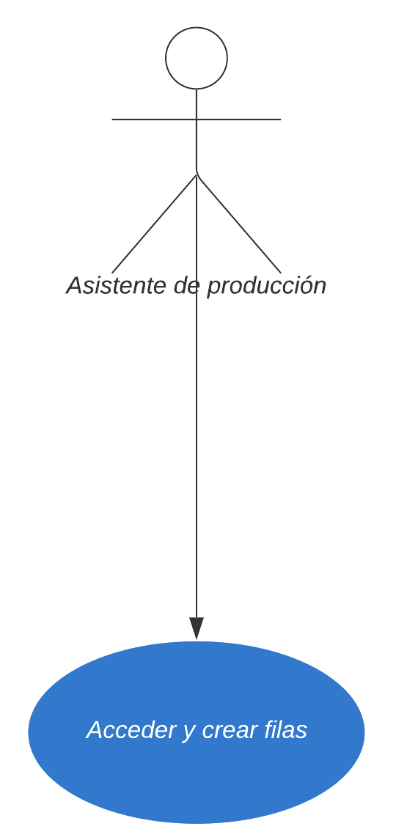


Ilustración 17: Caso de uso 8 | Acceder y crear celdas

#### Eventos de flujo

* El asistente de producción accede a las filas de la escaleta.
* El asistente de producción crea filas (solo Entrada, VTR/Nota, VTR/Full y Placa).

#### Diagramas de interacción

**Actores:**

*Asistente de Producción:* Este es el actor que interactúa con el sistema. Su rol es acceder y crear filas en la escaleta.

*Sistema*: Este es el actor que representa el sistema que gestiona las filas y celdas.

**Acceso a Fila:**

*Asistente de Producción -> Sistema:* Acceder a fila: El asistente inicia el proceso solicitando acceso a una fila específica en el sistema. Este paso indica que el asistente está listo para trabajar con las filas de la escaleta.

**Interacción con la Celda de Inserción:**

*Sistema -> Asistente de Producción:* Dar clic a celda de inserción: El sistema responde indicando que el asistente debe hacer clic en la celda de inserción. Esto sugiere que el sistema tiene una interfaz gráfica donde el asistente puede seleccionar dónde desea agregar la nueva fila.

**Solicitud de Creación de Fila:**

*Asistente de producción -> Sistema:* Solicitar creación de fila: Una vez que el asistente ha hecho clic en la celda de inserción, solicita formalmente la creación de una nueva fila.

**Selección del Tipo de Fila:**

*Sistema -> AsistenteDeProducción:* Solicitar tipo de fila (Entrada, VTR/Nota, VTR/Full, Placa): El sistema solicita al asistente que seleccione el tipo de fila que desea crear. Esto es importante porque la creación de filas está restringida a ciertos tipos, lo que asegura que el asistente elija una opción válida.

Selección del Tipo de Fila:

*Asistente de Producción -> Sistema*: Seleccionar tipo de fila (ej. VTR/Nota): El asistente elige un tipo de fila, como "VTR/Nota". Este paso es crucial para definir la naturaleza de la fila que se va a crear.

**Solicitud de Detalles de la Fila:**

*Sistema -> Asistente de Producción*: Solicitar detalles de la celda (nombre, descripción, etc.): El sistema solicita al asistente que proporcione detalles específicos sobre la fila que está creando, como el nombre y la descripción. Esto es necesario para completar la creación de la fila.

**Proveer Detalles de la Fila:**

*Asistente de producción -> Sistema:* Proveer detalles de la fila: El asistente envía los detalles requeridos al sistema.

Validación de Detalles:

*Sistema -> Sistema*: Validar detalles de la fila: El sistema realiza una validación interna de los detalles proporcionados para asegurarse de que son correctos y cumplen con los requisitos.

Resultado de la Validación:

***alt Validación exitosa***: Si la validación es exitosa:

*Sistema -> Sistema*: Crear nueva fila: El sistema procede a crear la nueva fila en la escaleta.

*Sistema -> Asistente de Producción*: Confirmar creación de fila: El sistema confirma al asistente que la fila ha sido creada exitosamente.

***else Validación fallida***: Si la validación falla:

*Sistema -> Asistente de Producción:* Mostrar mensaje de error: El sistema informa al asistente que hubo un error en la creación de la fila, indicando que debe corregir los detalles proporcionados.

* + 1. *Diagrama de secuencia*

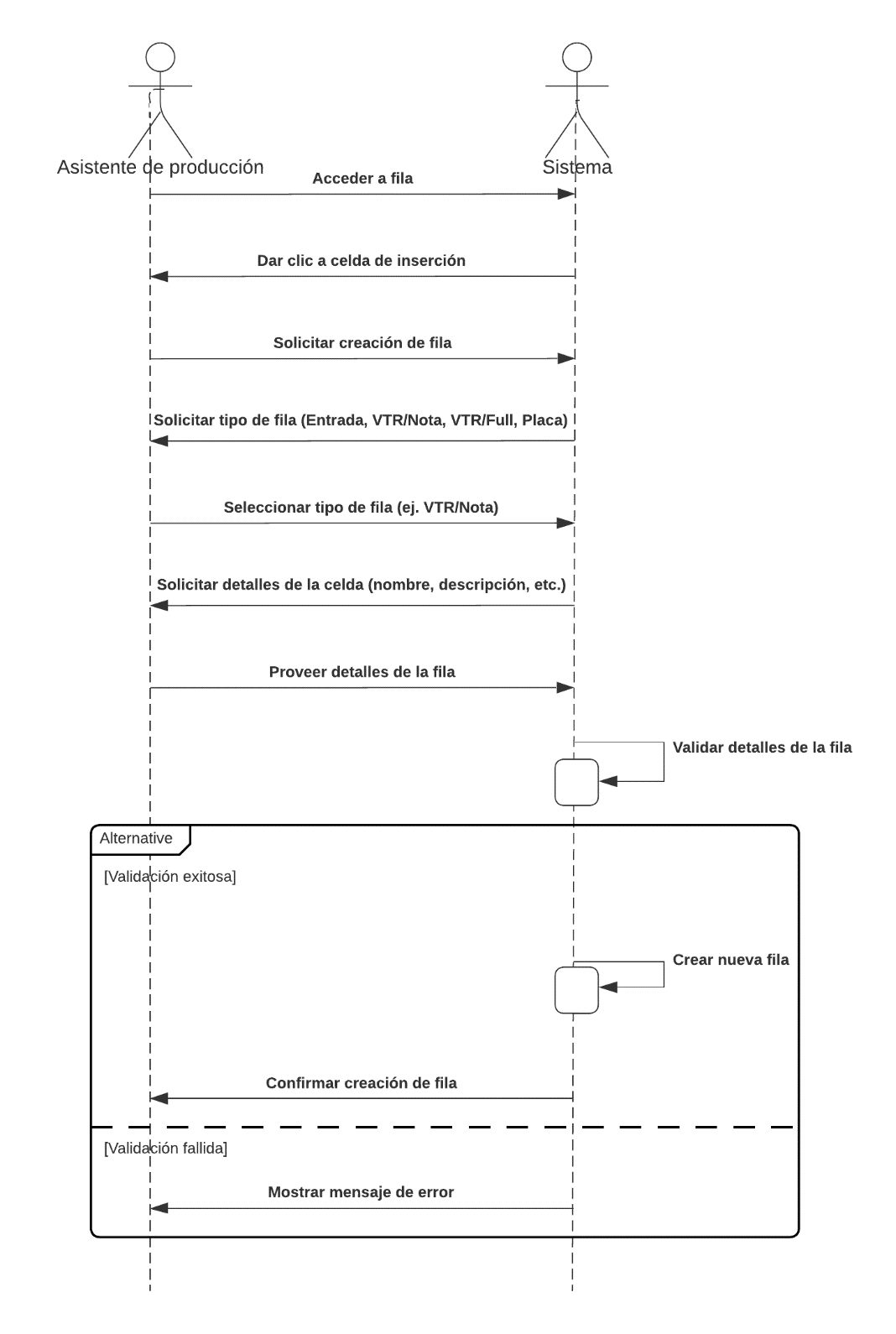


Ilustración 18: Diagrama de secuencia | Caso de uso

#### Objetos participantes

* Asistente de producción
* Sistema

#### Diagrama de objetos

Ilustración 19: Diagrama de objetos | Caso de uso

### CASO DE USO <Editar escaleta>

#### Breve descripción

Ilustración 20: Caso de uso 9 |

#### Eventos de flujo

#### Diagramas de interacción

* + 1. *Diagrama de secuencia*

Ilustración 21: Diagrama de secuencia | Caso de uso

#### Objetos participantes

#### Diagrama de objetos

Ilustración 22: Diagrama de objetos | Caso de uso

### CASO DE USO <Editar escaleta>

#### Breve descripción

Ilustración 23: Caso de uso |

#### Eventos de flujo

#### Diagramas de interacción

* + 1. *Diagrama de secuencia*

Ilustración 24: Diagrama de secuencia | Caso de uso

#### Objetos participantes

#### Diagrama de objetos

Ilustración 25: Diagrama de objetos | Caso de uso

### CASO DE USO <Editar escaleta>

#### Breve descripción

Ilustración 26: Caso de uso |

#### Eventos de flujo

#### Diagramas de interacción

* + 1. *Diagrama de secuencia*

Ilustración 27: Diagrama de secuencia | Caso de uso

#### Objetos participantes

#### Diagrama de objetos

Ilustración 28: Diagrama de objetos | Caso de uso

### CASO DE USO <Editar escaleta>

#### Breve descripción

Ilustración 29: Caso de uso |

#### Eventos de flujo

#### Diagramas de interacción

* + 1. *Diagrama de secuencia*

Ilustración 30: Diagrama de secuencia | Caso de uso

#### Objetos participantes

#### Diagrama de objetos

Ilustración 31: Diagrama de objetos | Caso de uso

### CASO DE USO <Editar escaleta>

#### Breve descripción

Ilustración 32: Caso de uso |

#### Eventos de flujo

#### Diagramas de interacción

* + 1. *Diagrama de secuencia*

Ilustración 33: Diagrama de secuencia | Caso de uso

#### Objetos participantes

#### Diagrama de objetos

Ilustración 34: Diagrama de objetos | Caso de uso

### CASO DE USO <Editar escaleta>

#### Breve descripción

Ilustración 35: Caso de uso |

#### Eventos de flujo

#### Diagramas de interacción

* + 1. *Diagrama de secuencia*

Ilustración 36: Diagrama de secuencia | Caso de uso

#### Objetos participantes

#### Diagrama de objetos

Ilustración 37: Diagrama de objetos | Caso de uso

### CASO DE USO <Editar escaleta>

#### Breve descripción

Ilustración 38: Caso de uso |

#### Eventos de flujo

#### Diagramas de interacción

* + 1. *Diagrama de secuencia*

Ilustración 39: Diagrama de secuencia | Caso de uso

#### Objetos participantes

#### Diagrama de objetos

Ilustración 40: Diagrama de objetos | Caso de uso

### CASO DE USO <Editar escaleta>

#### Breve descripción

Ilustración 41: Caso de uso |

#### Eventos de flujo

#### Diagramas de interacción

* + 1. *Diagrama de secuencia*

Ilustración 42: Diagrama de secuencia | Caso de uso

#### Objetos participantes

#### Diagrama de objetos

Ilustración 43: Diagrama de objetos | Caso de uso

### CASO DE USO <Editar escaleta>

#### Breve descripción

Ilustración 44: Caso de uso |

#### Eventos de flujo

#### Diagramas de interacción

* + 1. *Diagrama de secuencia*

Ilustración 45: Diagrama de secuencia | Caso de uso

#### Objetos participantes

#### Diagrama de objetos

Ilustración 46: Diagrama de objetos | Caso de uso

### CASO DE USO <Editar escaleta>

#### Breve descripción

Ilustración 47: Caso de uso |

#### Eventos de flujo

#### Diagramas de interacción

* + 1. *Diagrama de secuencia*

Ilustración 48: Diagrama de secuencia | Caso de uso

#### Objetos participantes

#### Diagrama de objetos

Ilustración 49: Diagrama de objetos | Caso de uso

### CASO DE USO <Editar escaleta>

#### Breve descripción

Ilustración 50: Caso de uso |

#### Eventos de flujo

#### Diagramas de interacción

* + 1. *Diagrama de secuencia*

Ilustración 51: Diagrama de secuencia | Caso de uso

#### Objetos participantes

#### Diagrama de objetos

Ilustración 52: Diagrama de objetos | Caso de uso

### CASO DE USO <Editar escaleta>

#### Breve descripción

Ilustración 53: Caso de uso |

#### Eventos de flujo

#### Diagramas de interacción

* + 1. *Diagrama de secuencia*

Ilustración 54: Diagrama de secuencia | Caso de uso

#### Objetos participantes

#### Diagrama de objetos

Ilustración 55: Diagrama de objetos | Caso de uso

### CASO DE USO <Editar escaleta>

#### Breve descripción

Ilustración 56: Caso de uso |

#### Eventos de flujo

#### Diagramas de interacción

* + 1. *Diagrama de secuencia*

Ilustración 57: Diagrama de secuencia | Caso de uso

#### Objetos participantes

#### Diagrama de objetos

Ilustración 58: Diagrama de objetos | Caso de uso