

PROYECTO PRACTICO DE APLICACION INTEGRADOR

FLUJO DE TRABAJO DE ANALISIS



Cátedra: Diseño de Sistemas

Año lectivo: 2023

Entrega N°3

Caso de Estudio: Sistema de Respuesta de Voz Interactiva (IVR)

Caso de Uso N°17- Registrar respuesta de operador

Curso:

Fecha de entrega: 21/12/2023

Docentes: Ing. Judith Meles, Ing. Mónica Lovay, Ing. María Sol Zanel

Docente Tutor: Ing. Mónica Lovay

Grupo N°:12

Integrantes:

- Carranza, Celia Mabel - 14059 – carranzacelia69@gmail.com
- Giménez Tomás- 14137- tomasgimenez7.tg@gmail.com
- Griffone Bruno- 14243- brunogriffone15@gmail.com



Contenido

Rediseño de la realización de casos de uso de análisis aplicando los 2 Patrones de Diseño de Gamma.....	2
a. Identificación del patrón implementado: descripción del problema que proponen resolver con una breve justificación.	2
b. Vista de la Estructura de la realización de caso de uso de diseño utilizando un diagrama de clases, que debe incluir la especificación de tipos de datos, tipos de retornos y tipos de parámetros y visibilidad de métodos – privados, públicos; de la estructura resultante de la aplicación del patrón.	3
ITERATOR.....	3
STATE.....	4
c. Vista Dinámica de la realización de caso de uso de diseño, utilizando un diagrama de secuencia; donde se modele la parte del escenario que se rediseña debido a la aplicación del patrón.....	5
ITERATOR.....	5
STATE.....	6
Implementación del caso de uso rediseñado aplicando un patrón de diseño respetando los patrones de IHM para la interfaz de usuario.....	7
a. Lenguaje de programación utilizado (Python/Java son los lenguajes para los que la cátedra dará soporte)	7
b. Framework de programación si se utiliza.....	7
c. Tecnología (Web/Escritorio).....	7
d. Base de datos y framework de persistencia utilizado.....	7



Rediseño de la realización de casos de uso de análisis aplicando los 2 Patrones de Diseño de Gamma.

a. Identificación del patrón implementado: descripción del problema que proponen resolver con una breve justificación.

La implementación del patrón State se justifica por la necesidad de encapsular y modularizar el comportamiento específico asociado a cada estado de una llamada. Esto facilita la comprensión del código, reduce la complejidad al evitar condicionales excesivos y permite la fácil extensión del sistema para manejar nuevos estados en el futuro.

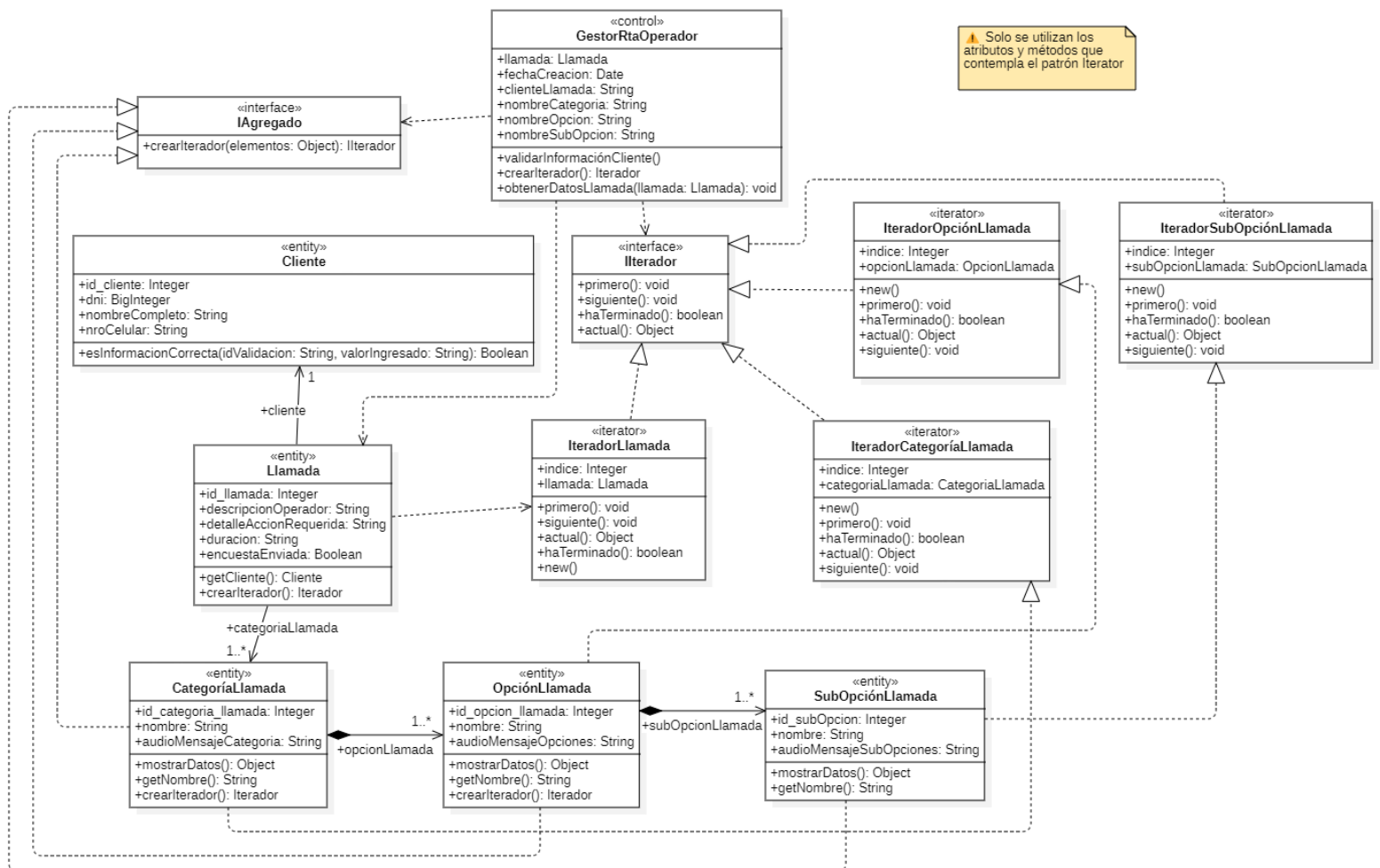
Esto se consigue creando una clase abstracta “Estado” para definir los comportamientos comunes y tantas clases concretas como sean necesarias.

La implementación del patrón Iterator se justifica por la necesidad de proporcionar al sistema una interfaz uniforme para iterar sobre las validaciones de una llamada y sobre las acciones requeridas. Esto desacopla el código del cliente de la estructura interna de la colección, facilita la adaptación a cambios en la lógica de iteración y mejora la flexibilidad del sistema para manejar diferentes tipos de colecciones en el futuro.

Esto se consigue creando una interfaz IAgregado donde se define el método para creación de iteradores, una clase abstracta IIterador donde se definen los comportamientos comunes y tantos iteradores como clases concretas haya que recorrer.

b. Vista de la Estructura de la realización de caso de uso de diseño utilizando un diagrama de clases, que debe incluir la especificación de tipos de datos, tipos de retornos y tipos de parámetros y visibilidad de métodos – privados, públicos; de la estructura resultante de la aplicación del patrón.

ITERATOR

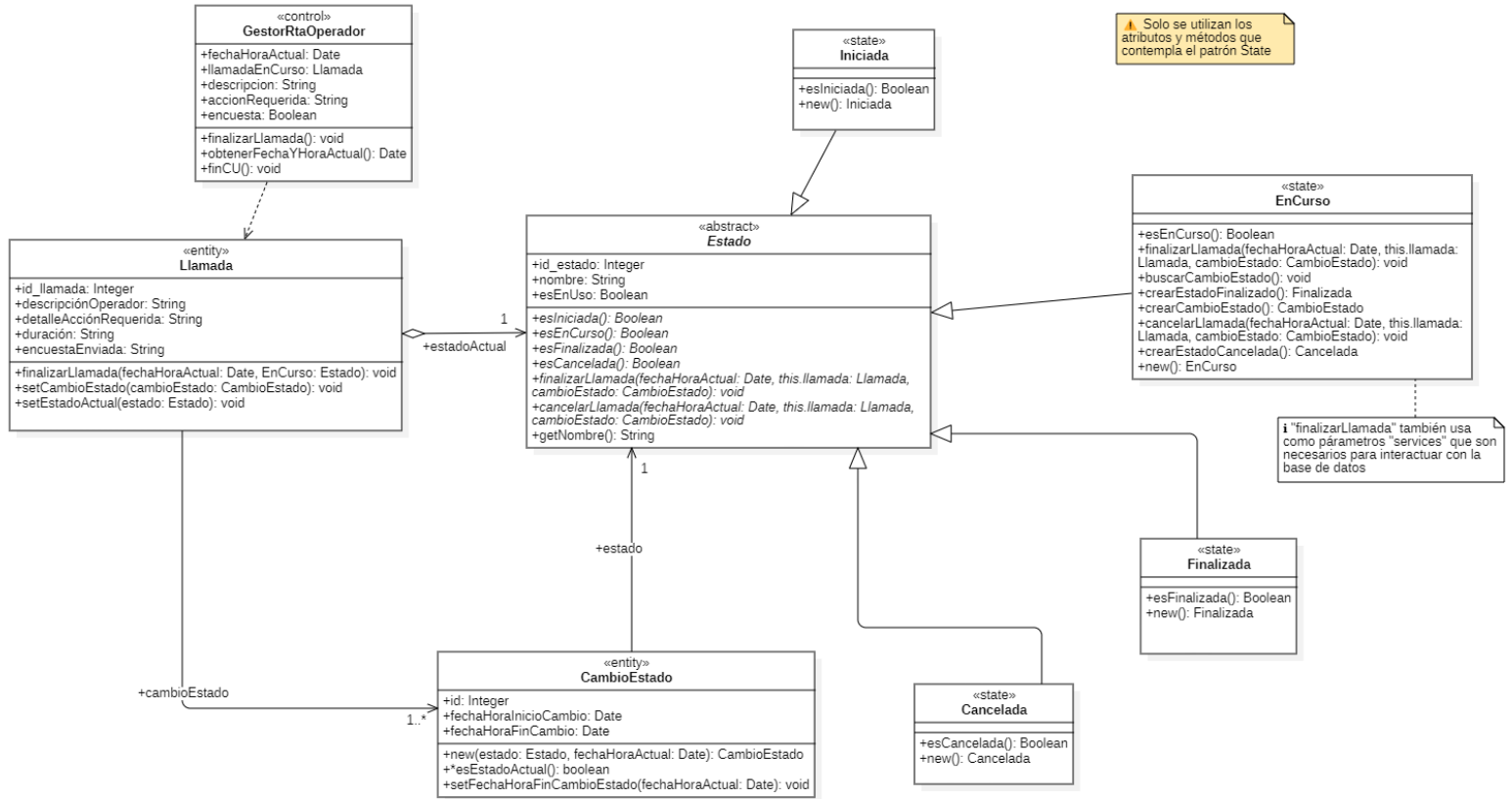


Links de acceso para mejor visualización:

https://prnt.sc/J_1S22_oeAl9



STATE



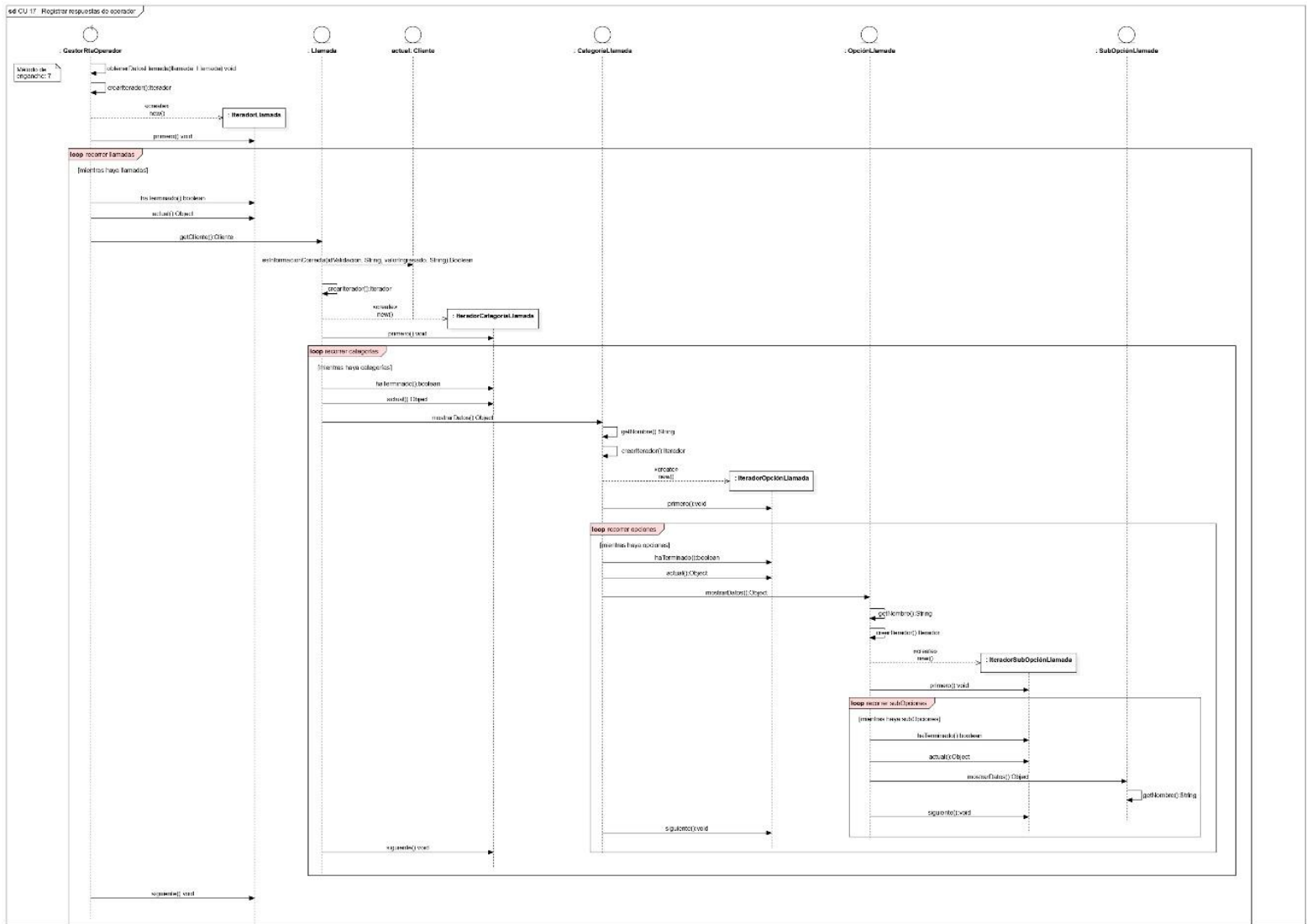
Links de acceso para mejor visualización:

<https://prnt.sc/Zq5NVX2MczAt>



c. Vista Dinámica de la realización de caso de uso de diseño, utilizando un diagrama de secuencia; donde se modele la parte del escenario que se rediseña debido a la aplicación del patrón.

ITERATOR

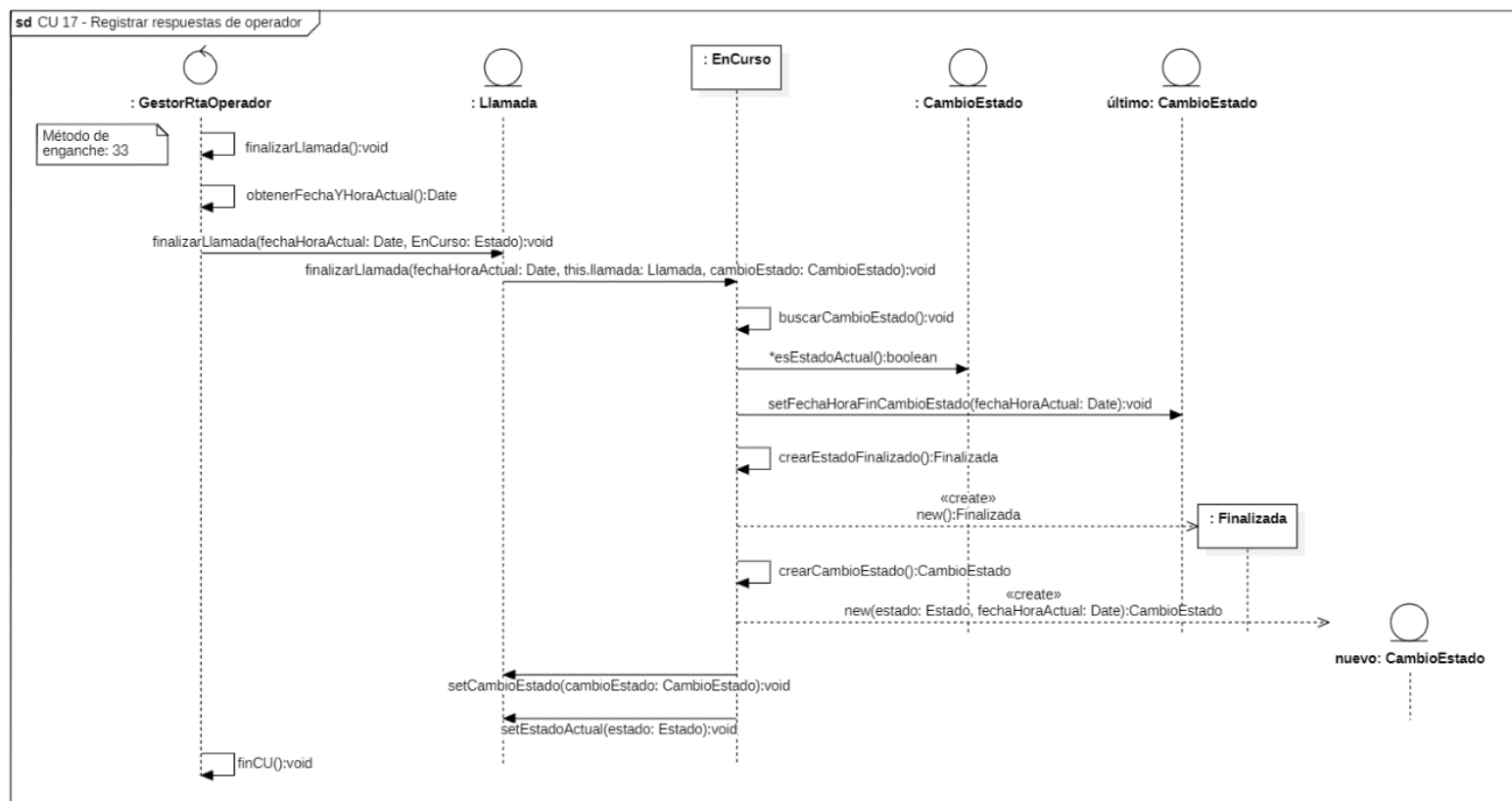


Links de acceso para mejor visualización:

https://prnt.sc/dl_zlHqV2mgU



STATE



Links de acceso para mejor visualización:

<https://prnt.sc/1s7kAOWb8Y0a>



Implementación del caso de uso rediseñado aplicando un patrón de diseño respetando los patrones de IHM para la interfaz de usuario.

a. Lenguaje de programación utilizado (Python/Java son los lenguajes para los que la cátedra dará soporte)

El lenguaje de programación utilizado es **Java**

b. Framework de programación si se utiliza

El Framework utilizado es **Springboot**

c. Tecnología (Web/Escritorio)

Usamos **tecnología web**, con plantillas **HTML** y estilos con **Bootstrap**.

d. Base de datos y framework de persistencia utilizado

Usamos una Base de datos relacional **SQL** con el gestor **MySQL**.

El framework de persistencia utilizado es **JPA**.