

**UNIVERSIDAD AUTÓNOMA DE MADRID**

**ESCUELA POLITÉCNICA SUPERIOR**



**Grado en Ingeniería Informática**

# **MEMORIA DE LA PRÁCTICA 5**

**ANÁLISIS Y DISEÑO DE SOFTWARE**

**Miguel Lozano Alonso  
Sergio Larriba Moreno**

# ÍNDICE

---

Apartado 1.....	2
Apartado 2.....	2
Apartado 3.....	3
Apartado 4.....	3
Diagrama de clases.....	4

## Apartado 1

---

La solución implementada para este apartado consiste en la creación de 3 clases diferentes:

- CoffeeType (Enum)
- Parameter
- StructuredPhrase

La primera de las 3 consiste simplemente en una enumeración que contiene los tipos de café que el usuario puede pedir.

La clase Parameter es una clase genérica que contendrá la información del segundo argumento de las funciones tanto with como settings. Hemos decidido hacerla genérica para facilitar el trabajo a la hora de hacer el ejercicio 2 y para poder capturar de algún modo, el segundo argumento de las funciones justo mencionadas.

La clase structured phrase contiene en un mapa, la relación entre el primer parámetro (un String siempre) y el segundo (un tipo genérico), de tal forma que podamos saber qué tipo de café quiere o cuantas unidades quiere sabiendo la frase en lenguaje natural que ha introducido el usuario.

## Apartado 2

---

La solución elegida en este apartado consiste en la implementación de 5 clases nuevas:

- CoffeeOrder
- ContextIntent
- Intent
- IntentHelper
- ParameterOfContext

La primera de todas, CoffeeOrder, es una clase simple que contiene una relación entre el numero de café y el tipo de café.

La clase ContextIntent, extiende de la clase padre Intent y agrega unas características adicionales como el manejo de parámetros y la creación de objetos a partir de los datos de entrada del usuario. Define los parámetros de entrada de un intento (por ejemplo, el número y el tipo de café en el caso de una orden de café), los valores que se extraen de la entrada del usuario y el método para crear un objeto a partir de estos valores. La clase también define una respuesta que se generará después de procesar la entrada del usuario. También, utiliza un mapa para almacenar los valores de los parámetros, lo que permite una fácil recuperación de los valores de los parámetros en cualquier momento.

La clase Intent define un modelo de intención o acción que un usuario puede querer llevar a cabo, por ejemplo, realizar un pedido de café. La clase contiene una lista de frases estructuradas que indican cómo el usuario puede expresar la intención y una respuesta para el chatbot cuando se reconoce la intención.

La clase `IntentHelper` es una clase de utilidad que proporciona métodos estáticos para trabajar con las intenciones. En particular, contiene un método `containsIgnoreCase` que verifica si una cadena dada contiene alguna de las opciones enumeradas proporcionadas.

Finalmente, La clase `ParameterOfContext` define un parámetro que se puede extraer de una cadena de entrada en el contexto de una intención. Un parámetro tiene un nombre, un predicado que se usa para verificar si el valor de entrada cumple con ciertos criterios, y una función que se usa para convertir el valor de entrada en un tipo específico de objeto. Esta clase se utiliza junto con la clase `ContextIntent` para modelar los parámetros de entrada de una intención y su correspondiente valor en una cadena de entrada.

## Apartado 3

---

La solución empleada para la resolución de este apartado es mucho más simple que la implementada en el apartado anterior. En este caso, consta de una sola clase:

- `Chatbot`

Ésta clase se encarga de gestionar las respuestas a las entradas del usuario, proporciona una respuesta determinada en función de la solicitud del usuario. Dicha clase tiene diferentes métodos, como `withIntent()` para agregar intenciones al chatbot, `withFallback()` para establecer una respuesta predeterminada en caso de que no haya una intención que coincida con la entrada del usuario y `reactTo()` para responder a la entrada del usuario. También, el método `getObject()` devuelve el objeto de la última interacción del chatbot.

## Apartado 4

---

La solución diseñada en este apartado consiste en la creación de 2 nuevas clases:

- `ChatbotBuilder`
- `ChitChatChatbotBuilder`

La clase `ChatbotBuilder` es una clase genérica que define un constructor para crear un objeto `Chatbot`. Este constructor permite crear un nuevo `Chatbot` proporcionando su nombre. Los métodos `withIntent` y `withFallback` son métodos que permiten agregar objetos de tipo `Intent` y respuestas por defecto al `Chatbot`. Además, el método `build` devuelve el objeto `Chatbot` construido.

La clase `ChitChatbotBuilder` es una clase genérica que extiende de la clase padre `Chatbot` y tiene la misma funcionalidad que “`Chatbot`”, pero con la adición de los intents por defecto de saludo, cortesía y despedida. Al crear una instancia de esta clase, se creará automáticamente un chatbot con estos intents, además de una respuesta por defecto de fallback.

# Diagrama de clases

