

Práctica 3 - Primeros módulos Odoo

Sistemas de Gestión Empresarial

Cristian Fernández

10 de noviembre de 2025

Índice

1. Introducción	2
2. Requisitos funcionales	3
3. Requisitos no funcionales	4
4. Casos de uso	5
5. Historias de usuario	6
6. Herramientas de desarrollo	7
7. Lenguajes de programación	8
8. Interfaz de la aplicación	9

1. Introducción

El objetivo de esta práctica consiste en

El programa debe seguir estas pautas:

- Crear una clase Cliente con los siguientes atributos (nombre, tiempoEspera).
- Crear una clase Camarero con un método *prepararCafe* que acepte como argumento un objeto Cliente y simule el tiempo que lleva prepararle dicho brebaje.
- Crear varios objetos Cliente (con diferentes nombres y tiempos de espera) y varios objetos Camarero en la clase principal.
- Simular la llegada de los clientes como hilos separados.
- Cada cliente debe intentar pedir un café y esperar el tiempo especificado. Los camarero deben preparar los cafés para los clientes en orden de llegada. Si un cliente se va antes de que su café esté listo, el camarero debe continuar con el siguiente cliente.
- El programa debe continuar ejecutándose hasta que todos los clientes hayan sido atendidos.

2. Requisitos funcionales

A continuación se listan los requisitos funcionales del sistema:

RF-1: Ver una leyenda de colores que informen del estado de los Hilos.

RF-2: Pulsar un botón que inicie la ejecución del programa.

RF-3: Mostrar dinámicamente todo el proceso de ejecución.

3. Requisitos no funcionales

A continuación se listan los requisitos NO funcionales del sistema:

RNF-1: El tiempo de respuesta al pulsar el botón Iniciar no debe superar los 2 segundos.

RNF-2: La interfaz gráfica de usuario ha de ser fácilmente entendible.

RNF-3: Cada acción realizada por los hilos debe ser lo suficientemente descriptiva para el usuario.

RNF-4: La interfaz debe avisar de forma clara que la ejecución de la aplicación ha terminado.

4. Casos de uso

A continuación se muestran los casos de uso mediante la siguiente tabla:

Caso de uso	Descripción
Consultar leyenda colores	El usuario consulta el funcionamiento de los hilos de la aplicación.
Iniciar aplicación	El usuario puede iniciar la aplicación pulsando un botón.

5. Historias de usuario

A continuación se muestran las historias de usuario mediante la siguiente tabla:

ID	Como...	Quiero...	Para...
HU-01	Usuario	Pulsar un botón	Iniciar la aplicación
HU-02	Usuario	Entender el comportamiento de los hilos	Deducir cómo funciona la concurrencia

6. Herramientas de desarrollo

IntelliJ Community Edition es un IDE gratuito y de código abierto, ideal para el desarrollo en lenguajes JVM y Android. Sus características principales incluyen un potente editor de código con autocompletado inteligente y refactorización, soporte para control de versiones, un depurador integrado, y herramientas de pruebas unitarias. Está disponible para todas las plataformas.

Características principales:

- **Editor de código potente:** Ofrece resaltado de sintaxis, análisis en tiempo real, sugerencias de código, e inspecciones y correcciones rápidas.
- **Soporte para múltiples lenguajes:** Es compatible con Java, Kotlin, Groovy, Scala, y otros lenguajes JVM.
- **Integración con control de versiones:** Permite la integración con sistemas como Git, SVN, y Mercurial.

7. Lenguajes de programación

JavaFX se caracteriza por su capacidad de crear aplicaciones enriquecidas visualmente, integrando multimedia (audio, video) y gráficos vectoriales. Permite desarrollar aplicaciones multiplataforma (escritorio, web, móvil, smart TV). Otras características clave son la integración fluida con el ecosistema Java y herramientas como FXML y Scene Builder para un desarrollo más eficiente y colaborativo.

Características principales:

- **Interfaces gráficas de usuario (GUI) enriquecidas:** Permite crear interfaces modernas y atractivas con animaciones, gráficos vectoriales, y elementos personalizables.
- **Integración multimedia:** Facilita la inclusión de contenido de audio, video y otros medios dentro de las aplicaciones.
- **Desarrollo declarativo (FXML):** Utiliza FXML, un lenguaje basado en XML, para describir la interfaz de usuario, lo que facilita el trabajo colaborativo entre diseñadores y desarrolladores.

8. Interfaz de la aplicación

A continuación unas capturas de las diferentes vistas de la interfaz.