

# Trabajo Práctico – Diseño de Aplicaciones Web.

## Modelado Orientado a Objetos: Oktubre Fest.

### Introducción.

Cada año, Múnich se llena de vida durante la famosa Oktubre Fest, una feria que combina música tradicional, comidas típicas y, por supuesto, mucha cerveza. La organización de la feria necesita un sistema de software que permita **modelar y controlar lo que sucede en la fiesta**: cómo los asistentes compran y consumen cerveza, cómo se gestionan las carpas, y cómo se registran las interacciones entre personas, bebidas y espacios.

El objetivo de este trabajo práctico es que desarrollen un **modelo orientado a objetos completo**, identificando **entidades clave, atributos, comportamientos, relaciones y restricciones** a partir de la narrativa. No se proporcionan listas de clases ni requisitos explícitos: deberán descubrirlos a partir de la historia y de las situaciones planteadas.

### Contexto narrativo.

Al llegar a la Oktubre Fest, los **asistentes** se encuentran con enormes carpas cerveceras. Cada **carpa** tiene un estilo propio: algunas ofrecen música tradicional, mientras que otras solo venden cerveza. **La entrada a cada carpa depende de las preferencias personales, la disponibilidad de espacio y la cantidad de asistentes presentes.**

Dentro de las carpas, los asistentes compran jarras de cerveza de distintos tamaños. Cada **jarra** pertenece a una **marca** de cerveza, y la cantidad de alcohol que contiene depende del **tipo de cerveza** y su contenido de lúpulo. Los asistentes disfrutan más de unas cervezas que de otras, y algunas personas solo consumen cervezas de cierta graduación o con características específicas según su nacionalidad.

Cada persona tiene un **nivel de aguante**, que indica cuánta cerveza puede consumir antes de emborracharse. También **tienen preferencias de música**: algunos disfrutan de la música tradicional y otros no. A partir de estas características, una persona decide si desea ingresar a una carpa y si puede hacerlo según las reglas de la misma.

Algunas personas siguen un **patrón de consumo**, pueden ser **compatibles con otros asistentes** según las cervezas que han probado, o muestran **lealtad a su país de origen** en las cervezas que eligen. Las carpas aplican distintas **políticas de venta, calculando recargos** según distintos criterios: un porcentaje fijo, según la cantidad de asistentes o según el nivel de ebriedad general.

## Historia integrada de uso.

Lukas, un alemán aficionado a la música tradicional, llega a la carpa “Hofbräu”. La carpa tiene música en vivo y vende únicamente cerveza roja. Lukas se detiene frente a la entrada, evaluando si desea ingresar. El encargado de la carpa abre su tablet con el sistema de gestión y verifica si Lukas **cumple con todas las condiciones** para entrar: le gusta la cerveza, la música coincide con sus preferencias, la carpa no ha alcanzado su límite de asistentes y no está ebrio según su historial de consumo. Todas las condiciones se cumplen, y el sistema registra a Lukas como asistente presente. La cantidad de asistentes se actualiza automáticamente y Lukas entra a la carpa.

Dentro de la carpa “Guinness”, Anna, checa y fanática de la cerveza negra, se acerca al mostrador y pide una jarra de 0.5 litros. El encargado selecciona a Anna en la lista de asistentes y el sistema muestra las capacidades disponibles. Se genera automáticamente un objeto **Jarra**, registrando la marca, la capacidad y la carpa que la sirve. El sistema actualiza el historial de consumo de Anna, calcula la cantidad de alcohol ingerido y su estado de ebriedad. La venta se confirma y el encargado queda listo para atender al siguiente pedido.

Peter y María, asistentes de distintas nacionalidades, están en la misma carpa y desean socializar. La aplicación cruza sus historiales de consumo y determina la **compatibilidad** entre ellos según las coincidencias y diferencias en las cervezas que han probado. El sistema confirma la compatibilidad y ambos pueden interactuar dentro de la carpa.

Markus ha comprado varias jarras en distintas carpas y quiere conocer su gasto total. La aplicación recopila todas las jarras adquiridas, incluyendo capacidad, marca, precio base por litro y recargo aplicado por cada carpa. Calcula el precio de cada jarra, suma el total y determina cuál fue la jarra más cara. Markus recibe esta información y puede decidir si seguirá comprando o si se retira, mientras el sistema registra la consulta en su historial.

Durante todo el evento, cada asistente toma decisiones basadas en **sus preferencias, nivel de aguante, tipos de cerveza y políticas de las carpas**. La interacción entre personas, cervezas y carpas genera un flujo dinámico de consumo, socialización y registro de información que el sistema debe modelar de manera precisa.

## Objetivos del trabajo práctico.

1. **Identificación de entidades y relaciones.**
  - Descubrir clases principales y subclases a partir de la narrativa.
2. **Definición de atributos y comportamientos coherentes.**
  - Determinar atributos relevantes (peso, nivel de aguante, capacidad de jarra, preferencias, etc.) y métodos clave (`estaEbria()`, `contenidoAlcohol()`, `permiteIngreso()`, `gastoTotal()`, `verificarCompatibilidad()`).
3. **Diseño de diagrama de clases completo y preciso.**
  - Incluir multiplicidades, asociaciones etiquetadas, composición/agregación, herencia y polimorfismo aplicados correctamente.

**4. Aplicación de buenas prácticas de diseño.**

- Mantener cohesión y responsabilidad única en cada clase, encapsulación de datos y claridad en la definición de métodos.

**5. Razonamiento crítico y justificación de decisiones.**

- Ser capaz de explicar por qué se modelaron ciertas clases, relaciones o métodos de determinada manera, justificando las elecciones de diseño.