

**Facultad de Informatica**

**P**rogramación **M**odular **y** **Orientación a** **Objetos**

Pub de Jon Etxebarria

Autor: Jon Etxebarria Sasia

Fecha: 22/05/2022

Grupo: B

**Presentación**

1 Diagrama de clases UML [3](#_heading=h.30j0zll)

1.1 Imágenes [3](#_heading=h.1fob9te)

1.2 Sobreescritura de métodos [3](#_heading=h.3znysh7)

1.3 Interfaces [3](#_heading=h.2et92p0)

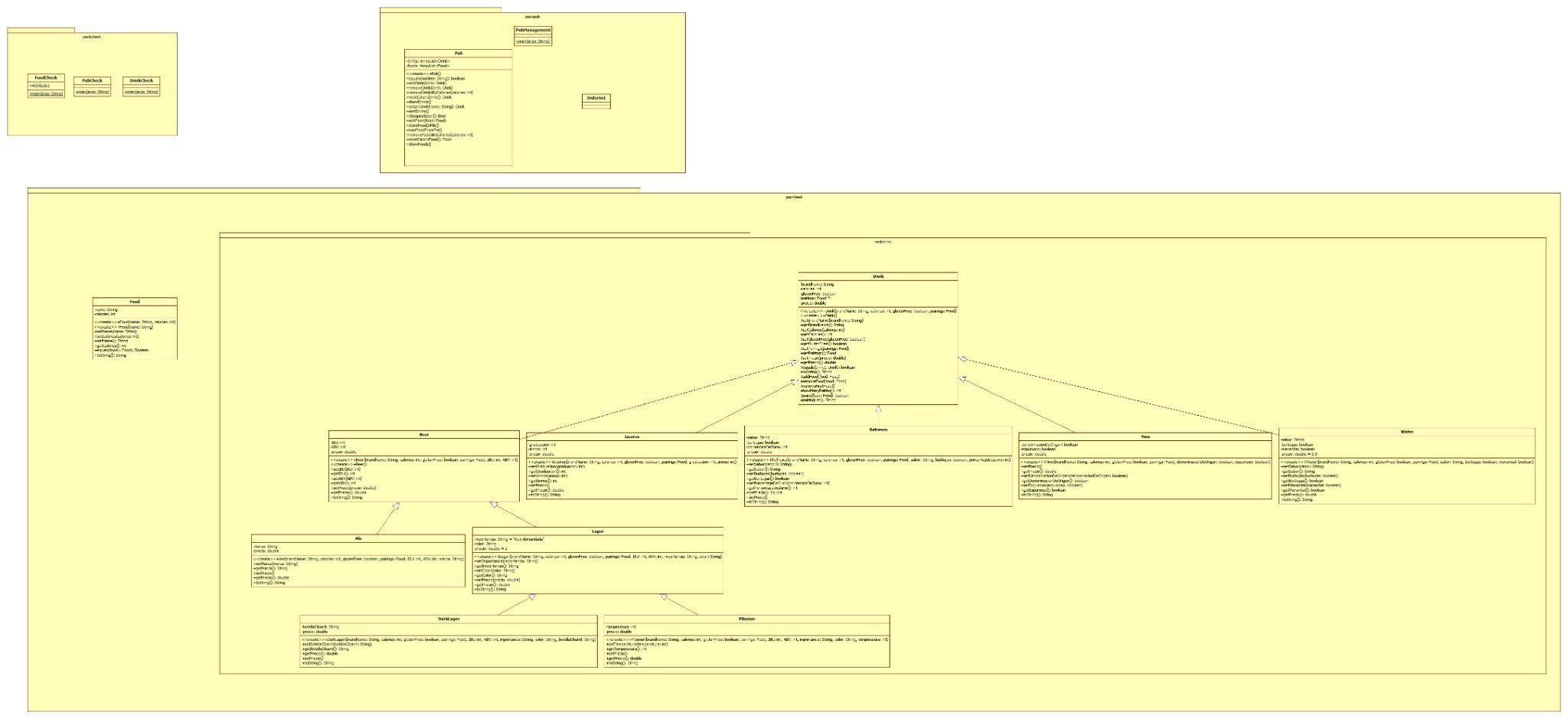
2 Problemas y las soluciones realizadas [4](#_heading=h.3dy6vkm)

3 Manual para la ejecución correcta del proyecto [5](#_heading=h.1t3h5sf)

4 Valoración personal [6](#_heading=h.4d34og8)

# Diagrama de clases UML

## Imágenes



Al haber muchas clases y con mucho contenido en la imagen general queda la letra con un tamaño muy reducido, en el archivo zip adjuntado en el proyecto viene el jpg por si quiere verlo con más detalle

## Sobreescritura de métodos

(Qué métodos has sobrescritos, dónde y por qué)

En la clase llamada Pub he sobreescrito el método equals ya que para comparar una bebida necesitas acceder al ArrayList de la clase Pub.

En Food he sobreescrito los métodos equals y toString de los cuales el primero compara solo el nombre de Food llamando a .getName y devuelve true si son iguales, y el segundo método me devuelve un string con todos los datos de la instancia.

Y para la clase Drink y sus subclases el método toString también está sobreescrito por el mismo motivo que en Food pero además en cada subclase se añaden los atributos de esa subclase en concreto.

Pero para el método equals en esta clase solo está sobreescrito en la clase más alta, Drink, en el cual también se devuelve true si el .getName de una es igual que el otro que se especifica.

## Interfaces

(Interfaces utilizadas, dónde y por qué )

# Problemas y las soluciones realizadas

(Indicar qué problemas han surgido y las soluciones realizadas. Por ejemplo: con la ayuda de compañeros o compañeras, bibliografía, tutorías ...)

La gran mayoría de problemas que he tenido y en los que me he quedado unas horas sin saber porque no funcionaba como yo quería han sido simplemente por no tener en cuenta algún archivo. Por ejemplo al hacer el archivo pubcheck y no darme cuenta de que no había iniciado ninguna instancia de la clase Food y por eso no guardaba nada en el archivo al que estaba apuntando.

También la jerarquía de Drink, teniendo bastantes subclases y estas además con sus subclases pero todas estas teniendo que compartir algunos atributos se me ha complicado en algún momento ya que tenía que tener en cuenta muchos archivos a la vez.

Para solucionar esto lo único que he hecho ha sido fijarme más en cada archivo e ir cerrándolos cuando lo comprobaba.

Y en general si tenía algún problema en el que no me daba el resultado que esperaba preguntaba a algún compañero y si me saltaba algún error de excepciones o cosas en las que el código deja de ejecutarse lo buscaba en internet.

# Manual para la ejecución correcta del proyecto

(Establece los pasos a seguir para ejecutar correctamente tu proyecto)

1. Descomprime el archivo .zip. Esto te va a dar un par de archivos y otro archivo .zip.
2. Este segundo .zip lo descomprimes también y esto ya será el proyecto.
3. Pero todavía falta una forma de ejecutarlo, para esto te tienes que descargar algo para ello. Para esto recomendamos Eclipse, si te diriges a su página principal lo podrás descargar sin ningún inconveniente.
4. Ahora abre Eclipse y dale a la pestaña File < import… y eliges la carpeta en la que hayas guardado el proyecto.
5. Por último tienes que darle a este botón  para ejecutarlo.
6. De ahí en adelante tienes que introducir el número de la opción que quieres elegir.

# Valoración personal

(planificación, tiempo dedicado a cada parte, propuestas de mejora,...)

La verdad es que empecé a hacerlo bastante pronto y bastante bien planificado pero esta última parte la fui dejando y me ha pillado un poco el toro. En general estoy bastante contento con cómo me ha quedado el proyecto, el menú me ha tomado bastante más tiempo del que me esperaba en un principio pero creo que está bastante bien organizado.

En mi opinión obligaría un poco más a usar cosas más variadas, con esto me refiero a interfaces, clases abstractas, meter algo gráfico o en general más funciones propias de java.