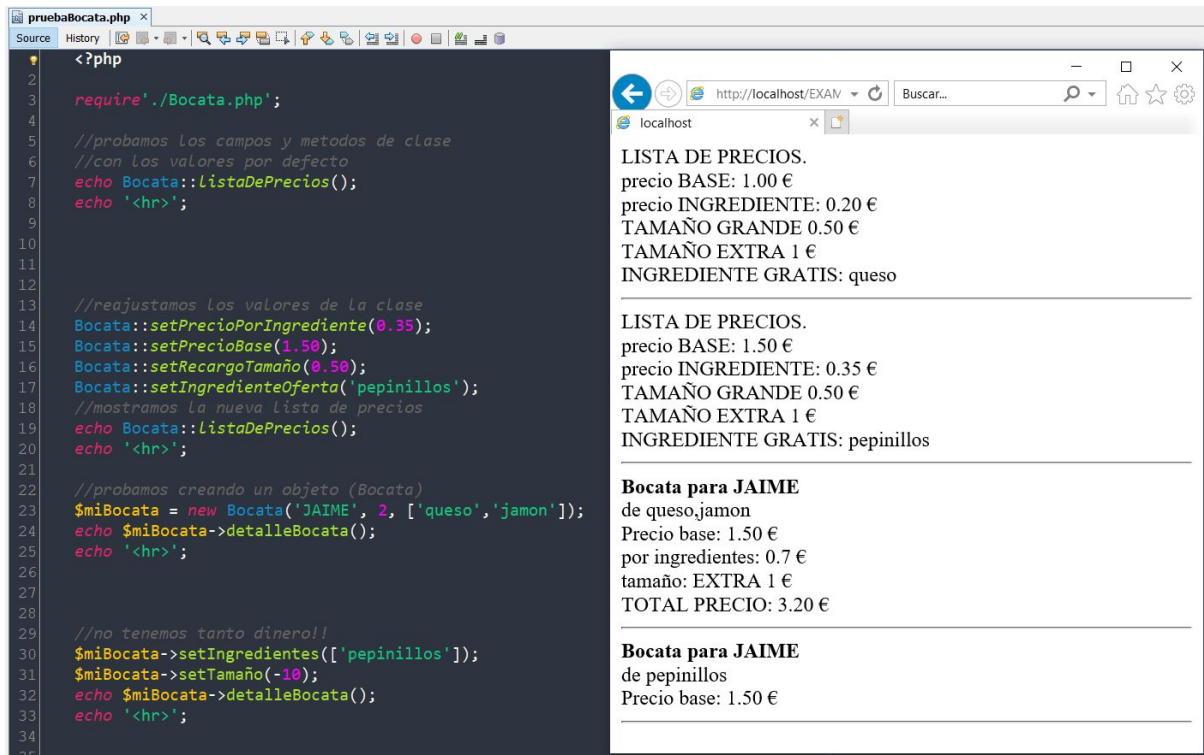


EJERCICIO PRACTICO DE PROGRAMACION ORIENTADA A OBJETOS 2º ASIR C 27/12/2021 - IMPLANTACIÓN DE APLICACIONES WEB

Se pide la clase **Bocata.php**, de manera que se valida con el código **pruebaBocata.php**,



ESPECIFICACIONES DE LA CLASE:

IMPORTANTE (CÓDIGO PERSONAL, en el enunciado es 00):

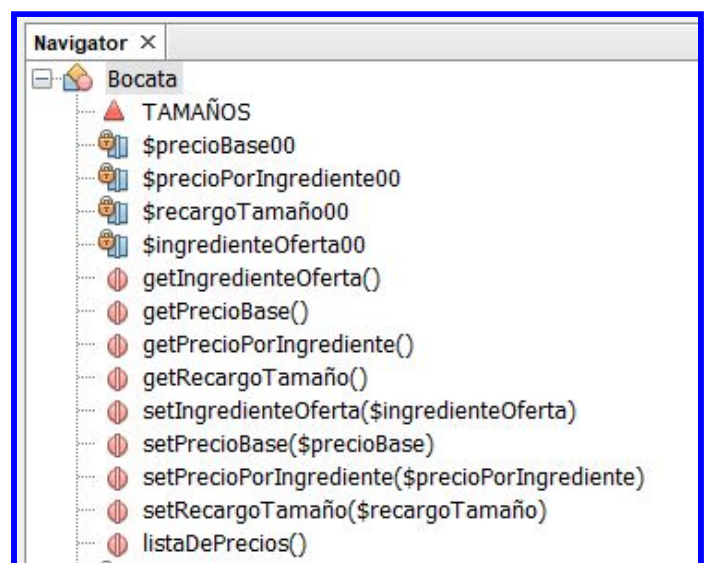
cuando le déis nombre a los **campos de clase y de objeto** le debéis añadir al final el **CÓDIGO PERSONAL** de dos dígitos que os daré en el momento de conectar con vosotros, así si un campo de clase se va a llamar `$estadoCamion` y vuestro CODIGO PERSONAL es 43, la declaración sería: `private static $estadoCamion43` y si es un campo de objeto, p.e. `$velocidad`, sería `private $velocidad43`.

Constante TAMAÑOS-> contiene un array
['NORMAL', 'GRANDE', 'EXTRA']

Campos de clase (self) PRIVADOS (con valores iniciales) : `$precioBase00` (1),
`$precioPorIngrediente00` (0.20),
`$recargoTamaño00` (0.50),
`$ingredienteOferta00` ('queso')

Setter Y Getters de CLASE automáticos.

Método público de clase `listaDePrecios()`



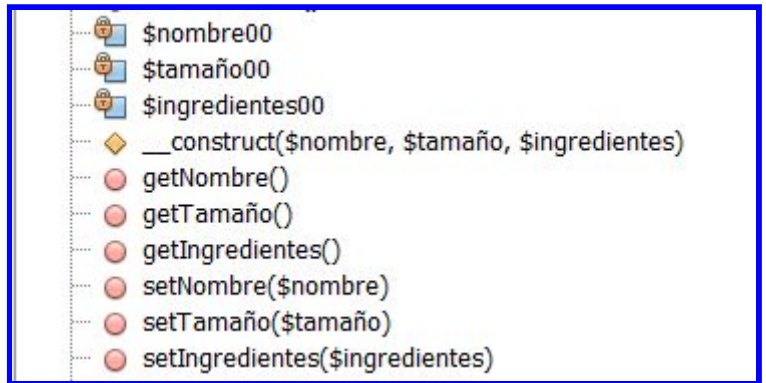
EJERCICIO PRACTICO DE PROGRAMACION ORIENTADA A OBJETOS

2º ASIR C 27/12/2021 - IMPLANTACIÓN DE APLICACIONES WEB

Campos de objeto (\$this) PRIVADOS :

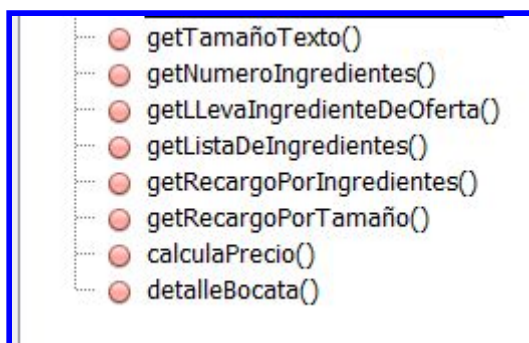
\$nombre00, **\$tamaño00**,
\$ingredientes.

Setter Y Getters de OBJETO
automáticos (excepto setTamaño).
Modificar **setTamaño(\$tamaño)**, para
que si nos intentan poner un tamaño
que no sea 0,1 o 2 lo ponga a 0.



Métodos público de objeto:

- **getTamañoTexto()**: devuelve la constante correspondiente de self::TAMAÑOS, según sea el campo \$this->tamaño, por ejemplo si es 1, devuelve 'GRANDE'.
- **getNumeroIngredientes()** devuelve el número de ingredientes del objeto.
- **getLLevaIngredienteDeOferta()** devuelve true si entre la lista de ingredientes del bocata (\$this->getIngredientes) está el de oferta (self::getIngredienteOferta), p.e, 'queso'.
- **getListadelIngredientes()**, devuelve una cadena con la lista de ingredientes separados por ',', p.e. 'queso,jamón,pepinillos'. (**implode**)
- **getRecargoPorIngredientes()**, devuelve el importe de este recargo, será el número de ingredientes por el self::getPrecioPorIngrediente, si por ejemplo lleva 3 ingredientes y son a 0.20 por ingrediente, sería 0.60. Si alguno de los ingredientes es el de oferta hay que restarle uno, si saliera negativo porque el único ingrediente que lleva es el de oferta, devuelve 0.
- **getRecargoPorTamaño()**, devuelve el recargo por el tamaño, multiplica el tamaño (\$this) por el recargoTamaño (self).
- **calculaPrecio()**, devuelve el precio del bocata, con dos decimales forzados (**number_format**).
- **detalleBocata()**, devuelve una cadena HTML con el ticket detallado del bocata,



Se sugiere ir desarrollando la clase en el mismo orden, y que vayáis probando en un php vuestro de prueba.

Se incluye el php pruebaBocata.php (NO MODIFICAR).

Solo teneis que subir **Bocata.php**