

EJECUCIÓN DEL PROGRAMA

```

138     return function ($this, callbackfn, that, specificCreate) {
139         var O = toObject($this);
140         var self = IndexedObject(O);
141         var boundFunction = bind(callbackfn, that, 3);
142         var length = toLength(self.length);
143         var index = 0;
144         var create = specificCreate || arraySpeciesCreate;
145         var target = IS_MAP ? create($this, length) : $this;
146         var value, result;
147         for (; length > index; index++) if (!IS_HOLES || !IS_UNDEFINED)
148             value = self[index];
149             result = boundFunction(value, index, 0);
150             if (TYPE) {
151                 if (IS_MAP) target[index] = result; // map
152                 else if (result) switch (TYPE) {
153                     case 3: return true;
154                     case 5: return value; // some
155                     case 6: return index; // find
156                     case 2: push.call(target, value); // filter
157                 } else if (IS_EVERY) return false; // every
158             }
159         );
160     );
161 };
162 });
163 module.exports =
164 // `Array.prototype.find` polyfill
165 // https://github.com/ljharb/es6-promise/pull/227

```

En esta sección se presentarán pruebas de la ejecución funcional del código final de la aplicación, con todas sus características y usos.

EJECUCIÓN

→ Inicio del programa:

El programa se ejecuta por primera vez y muestra el menú con todas las opciones.

```

-preview' '-XX:+ShowCodeDetailsInExceptionMessages'
paceStorage\16e8cf31981c1fea7a06a74b31be0fd\redhat

MÁQUINA BEBIDAS -> GESTOR
1. Recargar Ingredientes
2. Recargar Utensilios
3. Consultar estado de depositos y utensilios
4. Seleccionar Bebida
5. Cerrar
-> []

```

→ Seleccionar Bebida:

Seleccionamos la opción escribiendo **4** delante de -> en la consola. Ahí obtendremos un menú que nos mostrará las bebidas disponibles, seleccionamos la que queramos y nos hará la bebida con éxito siempre que haya stock.

```
-> 4
Selecciona una bebida:
1. Café Expresso
2. Café Largo
3. Café con Leche
4. Te
5. Chocolate
-> 5
Chocolate preparado. Has pedido 1 bebidas. Muchas gracias!
```

Cuando lo prepare, volveremos al menú principal inmediatamente.

```
Chocolate preparado. Has pedido 1 bebidas. Muchas gracias!

MÁQUINA BEBIDAS -> GESTOR
1. Recargar Ingredientes
2. Recargar Utensilios
3. Consultar estado de depositos y utensilios
4. Seleccionar Bebida
5. Cerrar
-> 1
```

→ Consultar estado de depositos y utensilios:

Si seleccionamos la **opción 3** en el menú podremos consultar **el estado del stock de la máquina**, indicando **cuánta cantidad** de cada ingrediente y utensilio queda.

Como hemos pedido una bebida, de chocolate, se ha restado la cantidad correspondiente de los ingredientes usados para dicha bebida.

```
-> 3
Ingrediente: Agua - Stock: 3850.0ml
Ingrediente: Leche - Stock: 1000.0g
Ingrediente: Café - Stock: 1000.0g
Ingrediente: Cacao - Stock: 980.0g
Ingrediente: Té - Stock: 1000.0g
Ingrediente: Azúcar - Stock: 1000.0g
Utensilio: Vaso - Stock: 199
Utensilio: Cuchara - Stock: 99
```

→ Recargar Utensilios:

Esta opción, seleccionando la **opción 2** en el menú, sirve para **recargar al máximo la cantidad de stock** de las cucharas y vasos de la máquina, para que en caso de agotarse se renueve.

Al usarla, mostrará un mensaje “**Utensilios cargados al máximo.**”

```
MÁQUINA BEBIDAS -> GESTOR
1. Recargar Ingredientes
2. Recargar Utensilios
3. Consultar estado de depositos y utensilios
4. Seleccionar Bebida
5. Cerrar
-> 2
Utensilios recargados al maximo.
```

Después de recargar al máximo los utensilios, podemos verificar su funcionalidad seleccionando la **opción 3** de nuevo y verificar que en efecto se han recargado.

```
-> 3
Ingrediente: Agua - Stock: 3850.0ml
Ingrediente: Leche - Stock: 1000.0g
Ingrediente: Café - Stock: 1000.0g
Ingrediente: Cacao - Stock: 980.0g
Ingrediente: Té - Stock: 1000.0g
Ingrediente: Azúcar - Stock: 1000.0g
Utensilio: Vaso - Stock: 200
Utensilio: Cuchara - Stock: 100
```

Como podemos ver, los utensilios **Vaso** y **Cuchara** tienen el stock de nuevo al máximo.

→ Recargar Ingredientes:

Esta opción cumple la misma función que la anterior, **recargando los ingredientes a su máximo stock.**

En este caso, hemos pedido un chocolate, gastando stock de agua y cacao, seleccionando la **opción 1**, podremos recargar el stock de nuevo.

```
-> 1
Ingredientes recargados al maximo.

MÁQUINA BEBIDAS -> GESTOR
1. Recargar Ingredientes
2. Recargar Utensilios
3. Consultar estado de depositos y utensilios
4. Seleccionar Bebida
5. Cerrar
-> 3
Ingrediente: Agua - Stock: 4000.0ml
Ingrediente: Leche - Stock: 1000.0g
Ingrediente: Café - Stock: 1000.0g
Ingrediente: Cacao - Stock: 1000.0g
Ingrediente: Té - Stock: 1000.0g
Ingrediente: Azúcar - Stock: 1000.0g
Utensilio: Vaso - Stock: 200
Utensilio: Cuchara - Stock: 100
```

EXCEPCIONES

→ Opcion no valida:

En caso de que el usuario seleccione una opción que no se encuentra en el menú retornará un mensaje de error indicando que selecciones una opción correcta.

Esto funciona a través de la opción **default** del switch que controla el menú.

```
MÁQUINA BEBIDAS -> GESTOR
1. Recargar Ingredientes
2. Recargar Utensilios
3. Consultar estado de depositos y utensilios
4. Seleccionar Bebida
5. Cerrar
-> ?
Selecciona una opción correcta.
```

```
default:
    System.out.println("Selecciona una opción correcta.");
```

→ Valor no válido:

En caso de que el usuario introduzca un valor no válido, recibirá como respuesta una excepción **Input Mismatch**, que le indicará que el valor es invalido.

Retornara de nuevo el menú de opciones para introducir una correcta.

```
MÁQUINA BEBIDAS -> GESTOR
1. Recargar Ingredientes
2. Recargar Utensilios
3. Consultar estado de depositos y utensilios
4. Seleccionar Bebida
5. Cerrar
-> hola
Error al detectar input: java.util.InputMismatchException
```

```
MÁQUINA BEBIDAS -> GESTOR
1. Recargar Ingredientes
2. Recargar Utensilios
3. Consultar estado de depositos y utensilios
4. Seleccionar Bebida
5. Cerrar
-> 
```

```
} catch (Exception e) { //Por si el usuario introduce un valor no valido.
    System.out.println("Error al detectar input: " + e);
    sc.nextLine(); //Devuelve el menu en la siguiente iteracion.
}
```

→ Longitud del Array no válida:

Esta excepción aparece cuando queremos seleccionar una bebida que no existe, si introducimos un valor que sale del array que guarda las opciones de bebidas, recibiremos un mensaje que nos indicará que debemos seleccionar una opción correcta, y nos devolverá al menú principal de la máquina.

```
MÁQUINA BEBIDAS -> GESTOR
1. Recargar Ingredientes
2. Recargar Utensilios
3. Consultar estado de depositos y utensilios
4. Seleccionar Bebida
5. Cerrar
-> 4
Selecciona una bebida:
1. Café Expresso
2. Café Largo
3. Café con Leche
4. Te
5. Chocolate
-> 7
Elije una opción correcta.
```

Lo mismo ocurre si introducimos un valor no válido.

```
Selecciona una bebida:
1. Café Expresso
2. Café Largo
3. Café con Leche
4. Te
5. Chocolate
-> fd
Elije una opción numérica
```