**UNIVERSIDAD UTE**

**FACULTAD CIENCIAS DE LA INGENIERIA E INDUSTRIAS**

**CARRERA:** INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN E INFORMÁTICA

**CURSO:** 7mo

**FECHA:** 2020/10/28

**MATERIA:** Aplicaciones Distribuidas

**INTEGRANTES:**

* Diego Flores
* Vanessa Paucar
* Juan Carlos Morocho

**MÉTODO DEFAULT**

Lo primero que tenemos que saber es que los métodos default de Java sólo se permiten introducir en interfaces, por lo tanto no existen en clases. Estos métodos, al igual que todos los demás en las interfaces, son de manera implícita públicos. Su principal diferencia es que no son abstractos como el resto y necesitan proporcionar una implementación para pasar la fase de compilación.

**ESTRUCTURA DE UN MÉTODO DEFAULT DE JAVA**

Los siguientes puntos son obligatorios para crear un método default en el lenguaje Java:

* Comenzará por la palabra reservada default.
* Proporcionará una implementación al método.
* Debe de encontrarse dentro de una interfaz



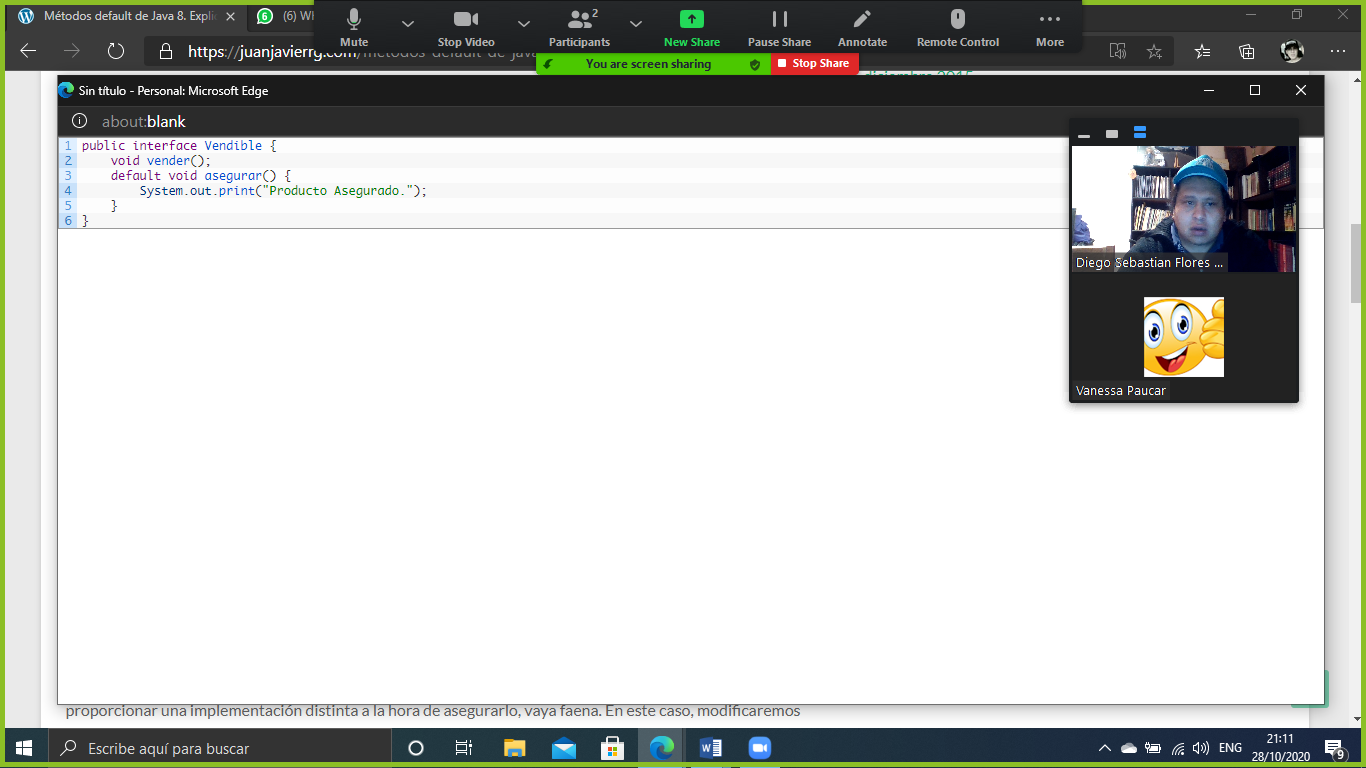
**¿QUÉ NOS PERMITE UN MÉTODO DEFAULT EN JAVA?**

Lo que nos permite es de manera flexible modificar el patrón de abstracción que hasta ahora las interfaces proporcionaban. Podremos modificar el contrato de la interfaz y automáticamente propagarlo a todas las clases que la implementan sin tener que editarlas todas (que se hubieran roto al no contemplar el nuevo método añadido). Ahora el ejemplo.

Si en la actualidad tuviéramos una interfaz implementada por 20 clases en nuestra aplicación y necesitamos añadir un nuevo método a esa interfaz para que nuestro diseño siga siendo rígido, tendríamos que modificar la interfaz y posteriormente modificar todas las clases que la implementan.

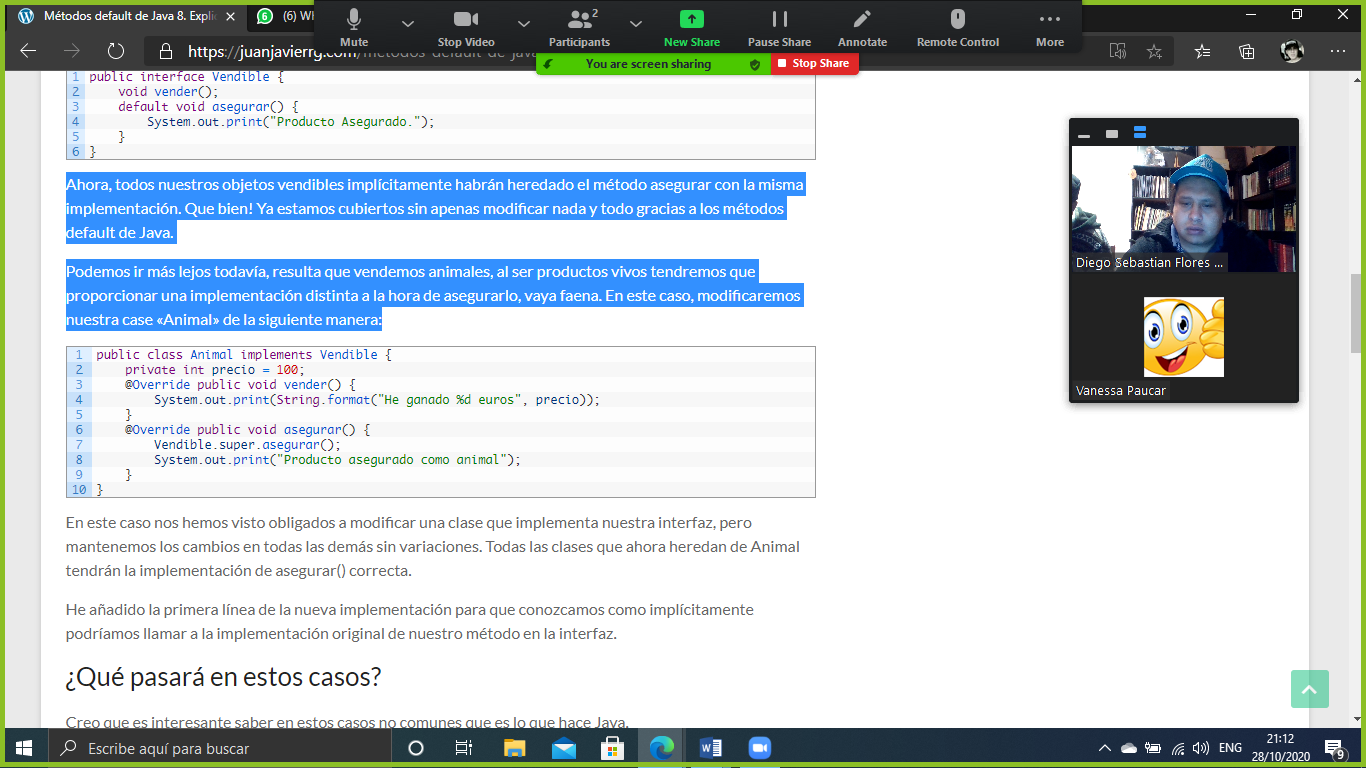


Cómo vemos, Lavadora es una de nuestras 20 clases y tiene el método vender que la interfaz Vendible le está obligando a implementar. Todo correcto hasta ahora. Pero llega el día en el que nos obligan a asegurar todos los productos que vendemos en nuestra aplicación. Por lo tanto todos nuestros productos vendibles son asegurables. Haremos un pequeño cambio a nuestra interfaz:



Ahora, todos nuestros objetos vendibles implícitamente habrán heredado el método asegurar con la misma implementación. Que bien! Ya estamos cubiertos sin apenas modificar nada y todo gracias a los métodos default de Java.

Podemos ir más lejos todavía, resulta que vendemos animales, al ser productos vivos tendremos que proporcionar una implementación distinta a la hora de asegurarlo, vaya faena. En este caso, modificaremos nuestra case «Animal» de la siguiente manera:



En este caso nos hemos visto obligados a modificar una clase que implementa nuestra interfaz, pero mantenemos los cambios en todas las demás sin variaciones. Todas las clases que ahora heredan de Animal tendrán la implementación de asegurar() correcta.

**CONCLUSIONES:**

* Los métodos default son sin duda una gran utilidad, sobre todo porque nos evita tener que tener que implementar clases abstractas, pero como ya lo mencionamos no las sustituyen.

# **Bibliografía**

* beginnersbook.com. 2020. Java 8 Interface Changes – Default Method And Static Method. [online] Available at: <https://beginnersbook.com/2017/10/java-8-interface-changes-default-method-and-static-method/> [Accessed 28 October 2020].
* Club de Tecnología. 2020. Java 8: Métodos Default En Interfaces. [online] Available at: <https://www.clubdetecnologia.net/blog/2016/java-8-metodos-default-en-interfaces/> [Accessed 28 October 2020].

**AUTOEVALUACIÓN**

|  |  |
| --- | --- |
| **NOMBRES** | **NOTA** |
| DIEGO FLORES | 95% |
| VANESSA PAUCAR | 95% |
| JUAN CARLOS MOROCHO | 95% |
| **OBSERVACIONES** |  |
| NUESTRA NOTA ES DEL 95%  YA QUE CUMPLIMOS CON EL TIEMPO ESTIMADO Y LA CONSULTA ESTUVO APROPIADA Y BASADA EN EL TEMA CONSULTADO.  CON UNA BUENA PRESENTACION |  |