1. **Utilización de Objetos**

Los objetos son instancias de las clases. Cada objeto tiene una identidad propia y tiene sus propios atributos. Todos los objetos transitorios residen en el contexto de una sesión interna. Una clase puede tener un número indefinido de instancias.

Un objeto existe mientras se está usando en el programa, lo que quiere decir que existe siempre que al menos una referencia apunte hacia él o al menos un método del objeto este registrando como método manejador de eventos.

1. **Referencias a Objetos**

Son punteros a los objetos ABAP. Se usan para acceder a un objeto desde un programa ABAP. En ABAP los objetos están siempre contenidos en variables referenciadas.

Las variables referenciadas o bien contienen el valor inicial o bien contienen la referencia a un objeto ya existente. La identidad de un objeto depende de su referencia. Una variable referenciada que apunta a un objeto es la que conoce la identidad del objeto. Los usuarios no pueden acceder a la identidad del objeto directamente.

Las variables referenciadas en ABAP son tratadas como cualquier otro objeto de datos elemental. Esto quiere decir que una variable referenciada puede contener una tabla interna o una estructura.

ABAP contiene un tipo de datos predefinido para las referencias, comparable a los tipos de datos para las estructuras o para las tablas internas. El tipo de datos completo no esta definido hasta la declaracion en el programa ABAP. El tipo de datos de la vatriable referenciada determina como el programa actua con su valor, o sea, con la referencia al objeto.

Hay dos tipos principales de referencias, la referencia a clases y la referencia a interfaces.

Las referencias a clases se definen usando la siguiente adición:

# TYPE REF TO <class>.

Ejemplo:

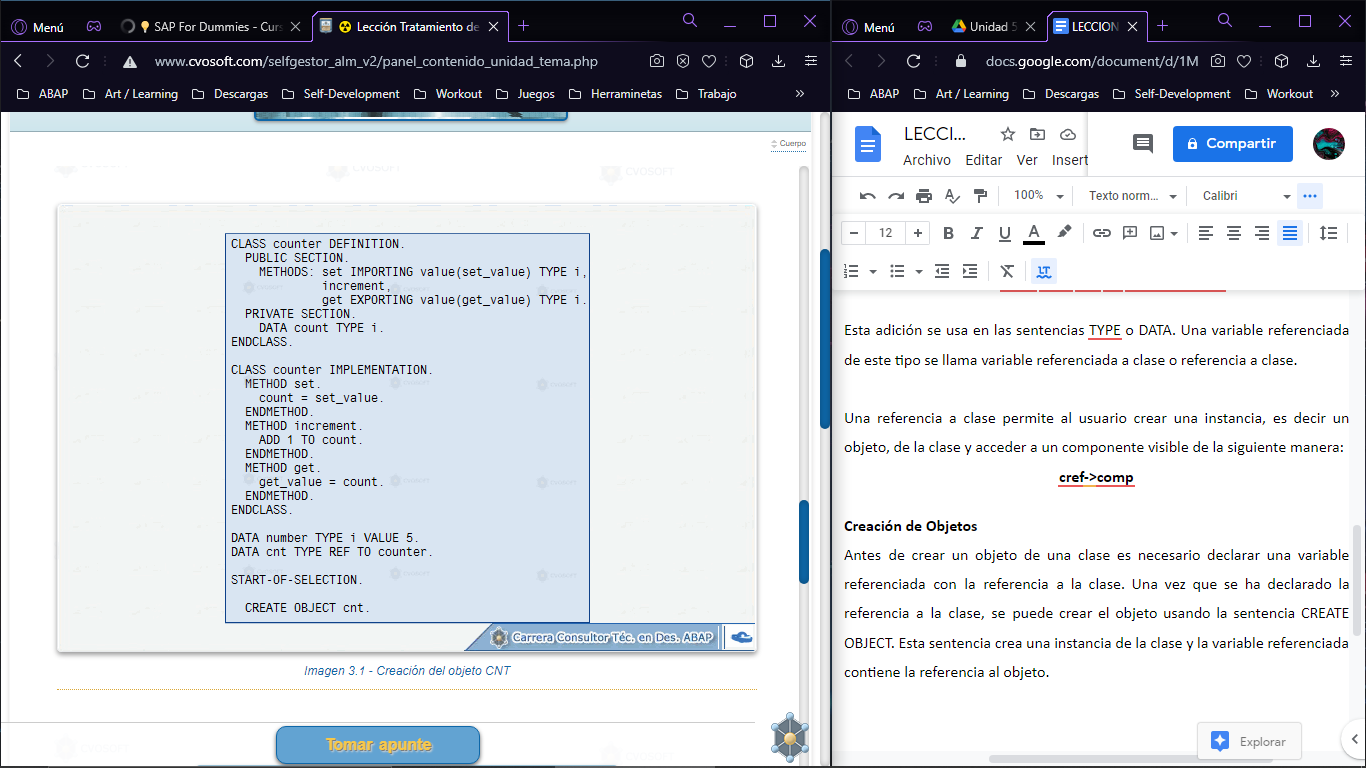
# Data: v\_alv TYPE REF TO cl\_gui\_alv\_grid.

Esta adición se usa en las sentencias TYPE o DATA. Una variable referenciada de este tipo se llama variable referenciada a clase o referencia a clase.

Una referencia a clase permite al usuario crear una instancia, es decir un objeto, de la clase y acceder a un componente visible de la siguiente manera:

# cref->comp

1. **Creación de Objetos**

Antes de crear un objeto de una clase es necesario declarar una variable referenciada con la referencia a la clase. Una vez que se ha declarado la referencia a la clase, se puede crear el objeto usando la sentencia CREATE OBJECT. Esta sentencia crea una instancia de la clase y la variable referenciada contiene la referencia al objeto. Ejemplo:

Audio Tip:

1. **¿Cómo acceder a los componentes de un Objeto?**

Los programas solo pueden acceder a los componentes de las instancias usando las referencias de las variables referenciadas. La sintaxis es la siguiente, siendo ref la variable referenciada:

* Para acceder al atributo sttr: ref->attr.
* Para llamar al método meth: CALL METHOD ref->meth.

Para los componentes estáticos o independientes de instancia, solo dependientes de clase, se puede usar tanto el nombre de la clase como la variable referenciada. También es posible acceder a los componentes estáticos de una clase antes de que un objeto de la clase haya sido creado.

La sintaxis, siendo class la clase es la siguiente:

* Para acceder al atributo estático attr: class->attr.
* Para llamar al método estático meth: CALL METHOD class->meth.

Dentro de una clase se puede acceder tambíen a los componentes individuales mediante la referencia a sí mismo ME:

* Para acceder al atributo attr en la propia clase: me->attr.
* Para llamar al método meth en la propia clase: CALL METHOD me->meth.

Audio Tip:

1. **Asignación de Referencias**

Se pueden asignar referencias a distintas variables referenciales usando la sentencia, MOVE. De esta manera se puede tener las referencias en varias variables referenciadas apuntando al mismo objeto.

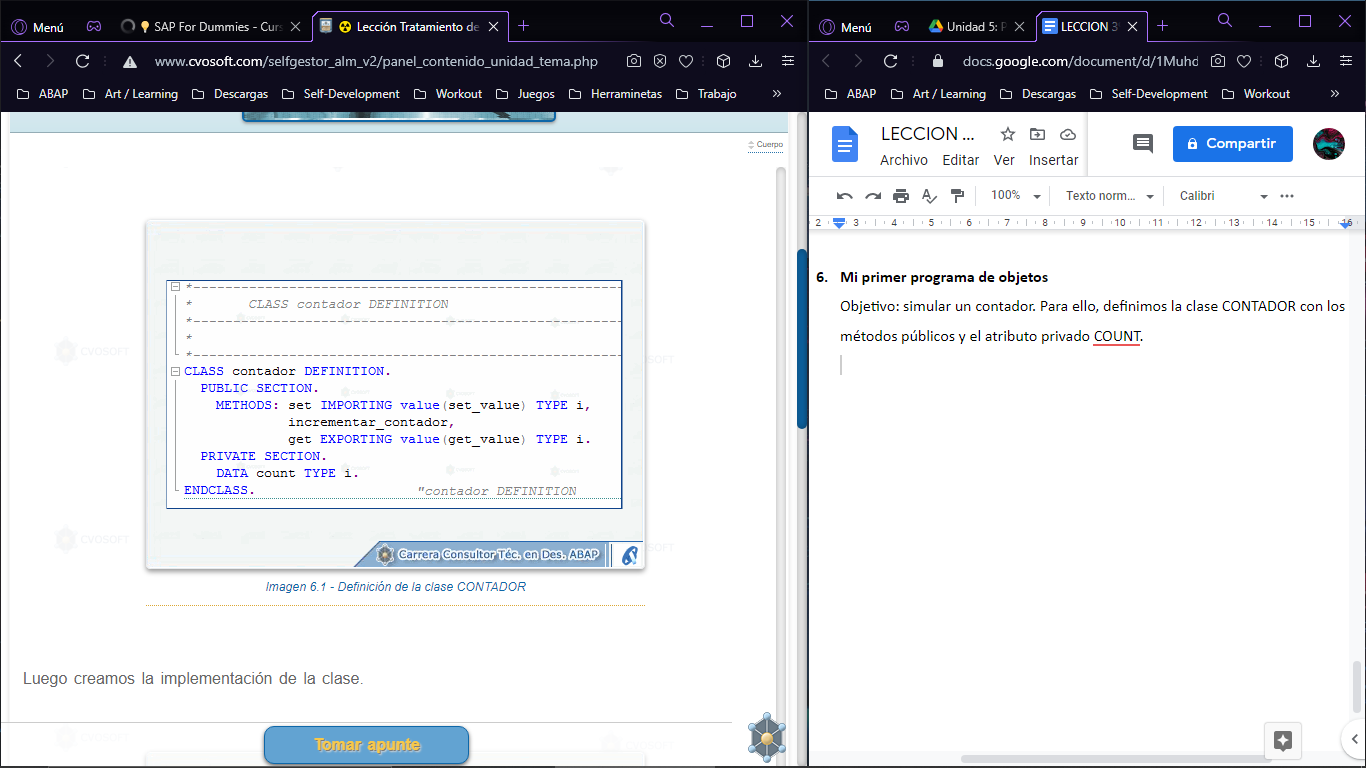
Cuando se asigna una referencia a una variable referenciada distinta, sus tipos deben ser compatibles. Cuando se usa la sentencia MOVE o el operador de asignación = para asignar variables referenciadas, el sistema debe ser capaz de reconocer en el chequeo de la sintaxis si la asignación va a ser posible. Esto mismo aplica cuando se pasan variables referenciadas como parámetros a procedimientos.

Si escribimos la sentencia cref1 = cref2, las dos referencias tienen que tener el mismo tipo, tienen que referirse a la misma clase, o la clase de cref1 tiene que ser la clase predefinida como vacía, o sea OBJETC.

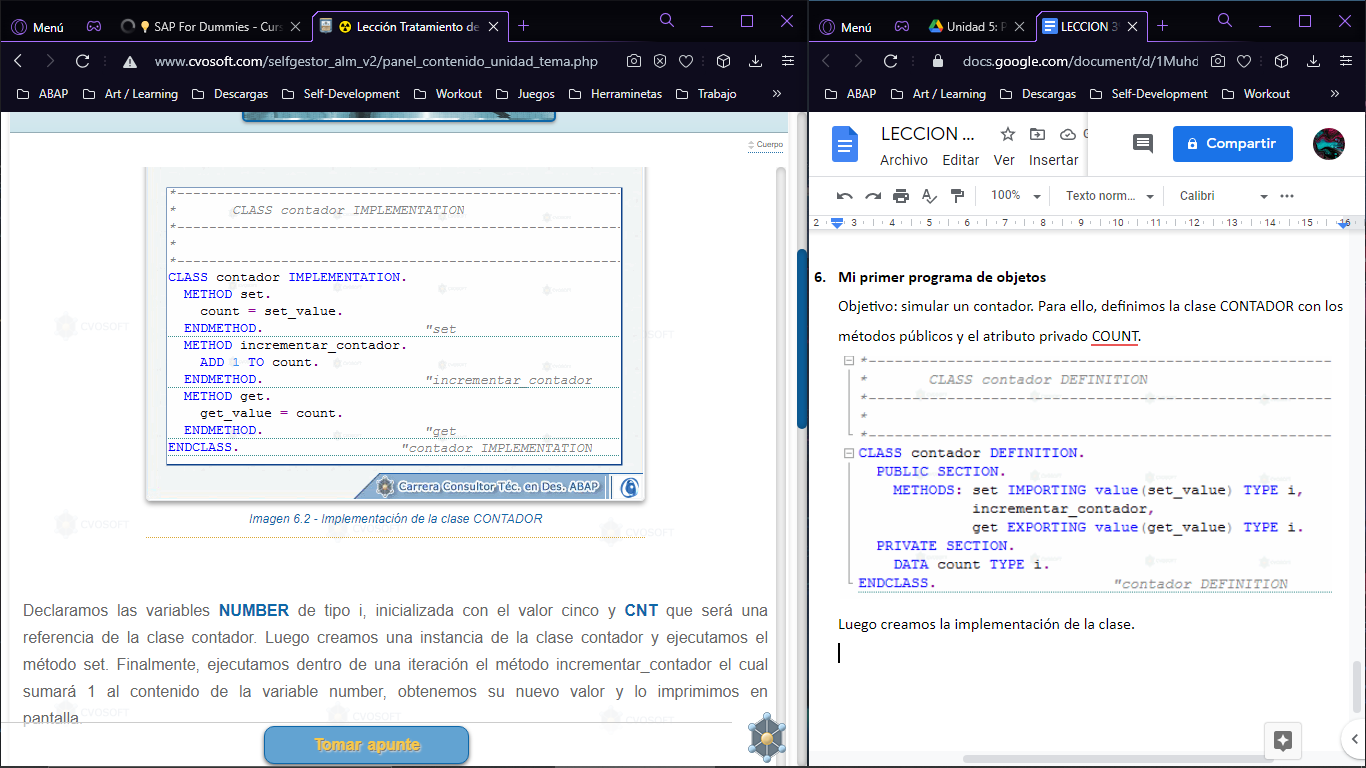
La clase OBJECT no tiene componentes y tiene la misma función para las variables referenciadas que el tipo de datos ANY para las variables normales. Las variables referenciadas con el tipo OBJETC pueden funcionar como contenedoras para pasar referencias. De cualquier manera, nunca pueden ser usadas para acceder a objetos.

1. **Mi primer programa de objetos**

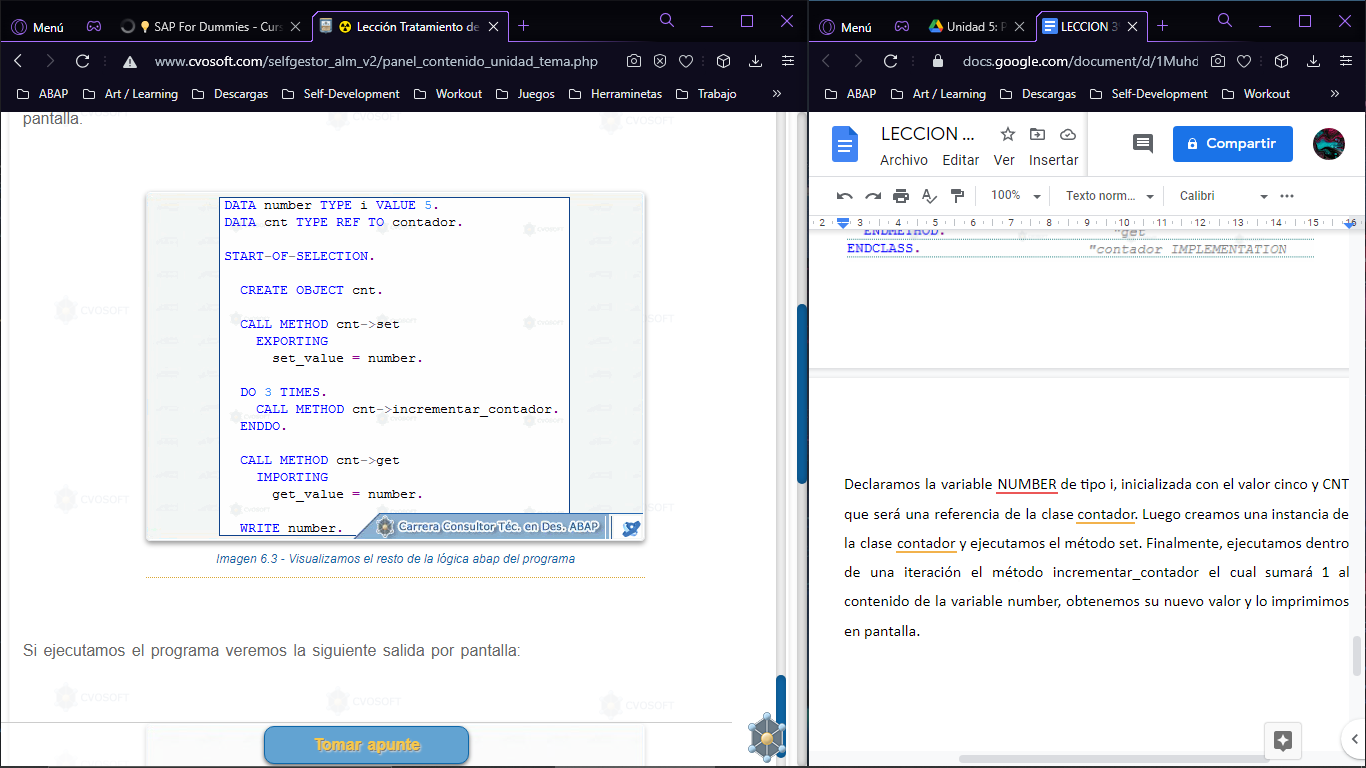
Objetivo: simular un contador. Para ello, definimos la clase CONTADOR con los métodos públicos y el atributo privado COUNT.



Luego creamos la implementación de la clase.



Declaramos la variable NUMBER de tipo i, inicializada con el valor cinco y CNT que será una referencia de la clase contador. Luego creamos una instancia de la clase contador y ejecutamos el método set. Finalmente, ejecutamos dentro de una iteración el método incrementar\_contador el cual sumará 1 al contenido de la variable number, obtenemos su nuevo valor y lo imprimimos en pantalla.



Salida por pantalla:

