



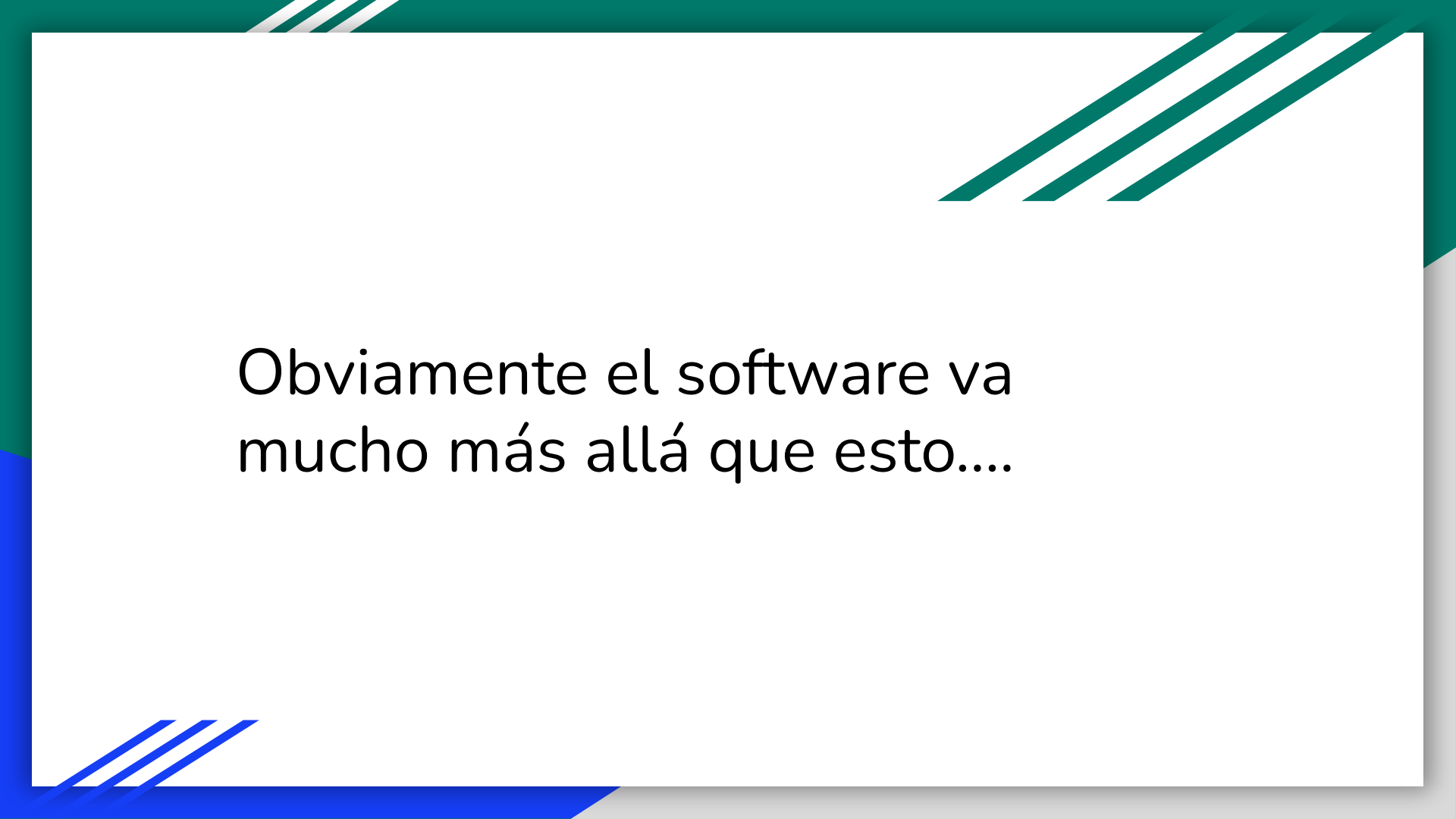
Buenas prácticas de programación



¿ Qué entendemos por software ?

Según la RAE

El **software** es un conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora.



Obviamente el software va
mucho más allá que esto....

El Software tiene 4 propiedades :

- ❏ Complejidad
- ❏ Conformidad
- ❏ Variabilidad
- ❏ Invisibilidad

El Software tiene 4 propiedades :

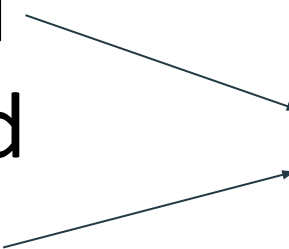
❏ Complejidad

❏ Conformidad

❏ Variabilidad

❏ Invisibilidad

Estas son las que nosotros, como programadores, podemos atacar



¿Qué podemos hacer como programadores?



“Any fool can write code that a computer can understand. Good programmers write code that **humans can understand.**” - Martin Fowler



*Cualquier tonto puede escribir código que una computadora pueda entender. Los buenos programadores escriben código que los **humanos puedan entender.***



Objetivos

Queremos que nuestro código sea:

- Fácil de entender: Esto significa que debe ser simple y legible. (Complejidad)
- Fácil de modificar: Como el software es algo que está cambiando constantemente tenemos que hacer que este proceso sea ágil y sencillo. (Variabilidad)



Entonces... ¿Cuáles son las
buenas prácticas de
programación?

Nombres descriptivos

Los nombres de variables y funciones deben ser:

- Autodescriptivos: El mismo nombre debe darme información de la variable o función.
- Fáciles de buscar
- Pronunciables: Evitar nombres como “X Æ A-12” o “Sztajnszrajber”.

Mala práctica

```
1 int x;  
2 int xx;  
3 char v[MAX];  
4
```

Buena práctica

```
1 int i;  
2 int contador_personas;  
3 char nombre_alumno[MAX_NOMBRE];  
4
```

Uso de constantes

- Usamos constantes para que cambiar valores de nuestro programa sea más fácil si llegan a cambiar en el futuro.
- Usamos constantes para que nuestro código sea más legible. Es más claro leer `VELOCIDAD_LUZ` que `299792458`

Convención: Los nombres de constantes siempre están en mayúsculas para que sean más distinguibles.

Evitar el uso de variables globales

Las variables globales son variables declaradas fuera de cualquier función, estas pueden ser accedidas desde cualquier lado del programa. El problema con las variables globales es que son muy difíciles de seguir y pueden cambiar en cualquier lugar, hacen que el trabajo de encontrar errores sea más largo y tedioso.

Nota: Como las constantes no cambian no sufren de estos problemas y podemos declararlas globalmente.

Indentación

Cada vez que entramos en un bloque nuevo (esto es cuando abrimos los corchetes) dejamos un espacio tabulado. Esto significa que cada bloque está a una altura diferente y es claro dónde empiezan y terminan las diferentes estructuras de control.

Mala práctica

```
1 int suma = 0;
2 while(suma < MAX_PERSONAS){
3     for(int i = 0 ; i < MAX_PERSONAS ; i++){
4         suma = suma + i;
5         printf("%d\n" , i);
6     }
7 }
```

Buena práctica

```
1 int suma = 0;
2 while(suma < MAX_PERSONAS){
3     for(int i = 0 ; i < MAX_PERSONAS ; i++){
4         suma = suma + i;
5         printf("%d\n" , i);
6     }
7 }
```

K.I.S.S (Keep It Simple, Stupid!)

- Muchas veces se nos ocurren formas super complejas de resolver un problema de forma más eficiente pero si sacrificamos la legibilidad y simpleza de nuestro código entonces no lo vale.
- Nuestro objetivo como buenos programadores es escribir código que los humanos puedan entender.
- **El código simple es más fácil de leer y de arreglar.**

Algunas más:

- Inicializar variables.
- Ser consistente con las convenciones que elegimos.
- Usar paréntesis para evitar ambigüedades.
- Agregar pre y post condiciones a nuestras funciones.
- ¡¡MODULARIZAR!!



Y todavía hay más...