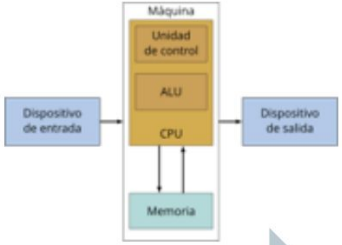
HISTORIA

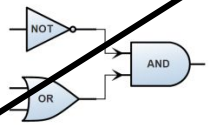
S IX. Álgebra. (Al-Juarismi)

1854. Álgebra de Boole.

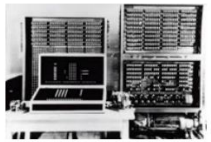
1941. Z3 (Konrad Zuse)



1837. Máquina Analítica. (Babbage, Lovelace)



1936. Máquina de Turing.

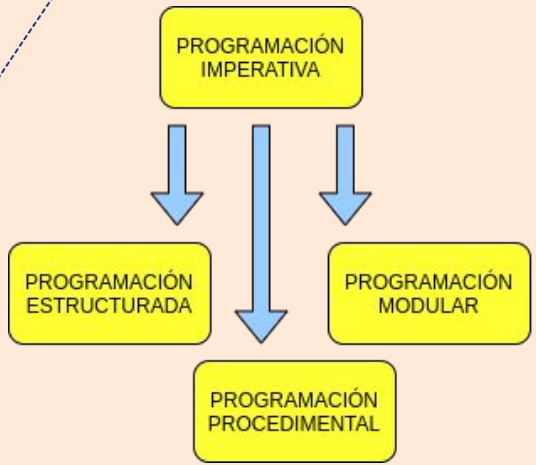


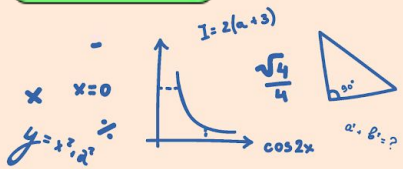
1945. Arquitectura de Von Neumann

Alonzo Church, Cálculo Lambda (1936)

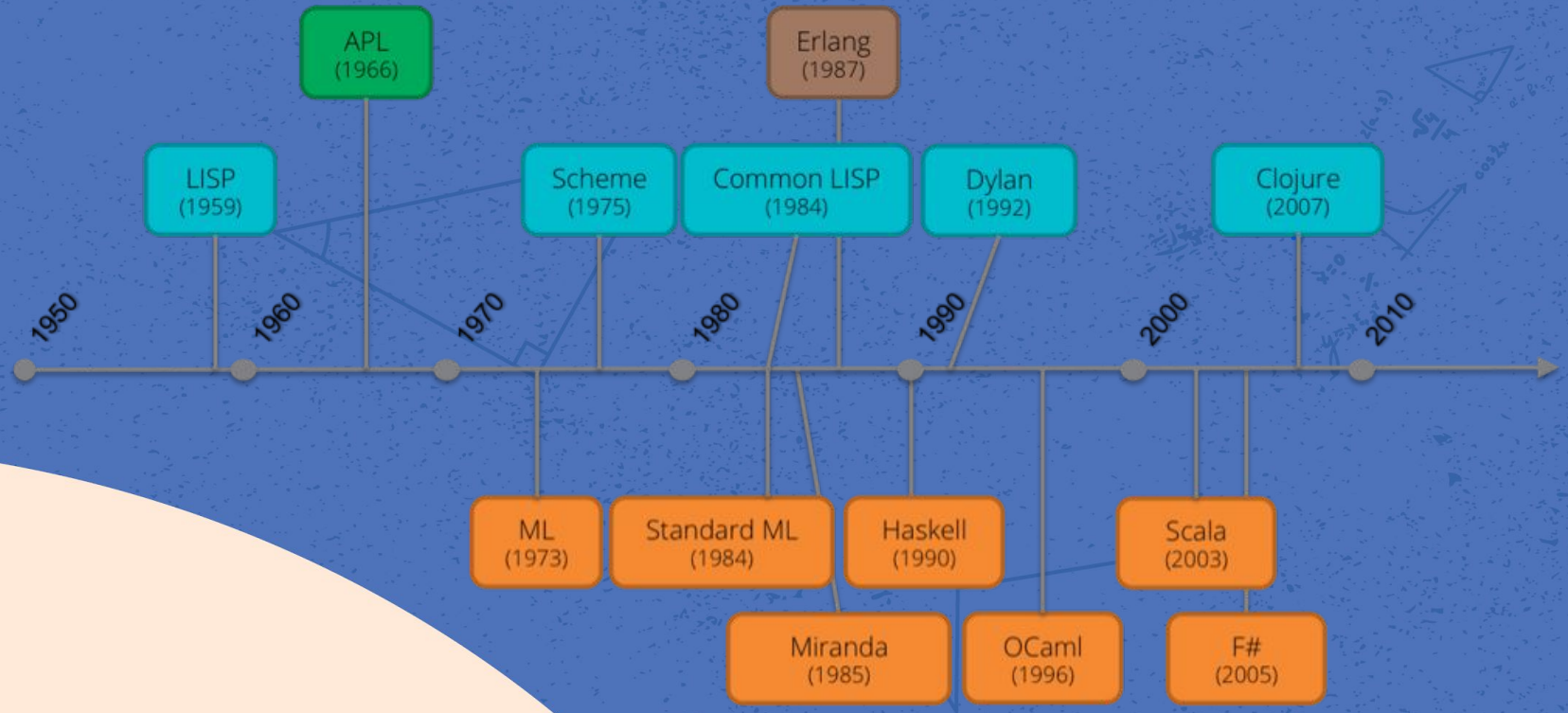
¿Qué es un paradigma de programación?

Es una forma de clasificación de los lenguajes de programación de acuerdo a las características que lo identifican.





HISTORIA DE LA PROGRAMACIÓN FUNCIONAL



EFECTOS COLATERALES

Una función tiene efectos colaterales si además de retornar un valor, modifica el estado de su entorno.

```
program ejemplo;
var x : Integer;
function opaca (y : Integer) : Integer;
begin
   x := x + y;
   opaca := x
end;
begin
    x := 0;
    x := opaca(1) + opaca(2);
end.
```

```
x       opaca(1)
0       0 + 1      -> 1
1       opaca(2)
3       1 + 2      -> 3
opaca(1) + opaca(2) -> 4
-----------------------------------
x       opaca(2)
0       0 + 2        -> 2
2       opaca(1)
3       2 + 1        -> 3
opaca(2) + opaca(1) -> 5
```

¿Cuánto vale x? ¿qué pasa si cambiamos el orden de los sumandos?

DECLARATIVO VS. IMPERATIVO

MATEFUN

```
factorial :: N -> N
factorial (n) = 1 si n == 0
                o n * factorial(n-1)
```

PASCAL

```
program factorial;
var num, fact, i: Integer;
begin
 readln (num);
 fact := 1;
 for i :=2 to  num do
    fact := fact * i;
 writeln('el factorial de ', num , ' es ', fact);
end.
```

DECLARATIVO VS. IMPERATIVO

MATEFUN

```
factorial :: N -> N
factorial (n) = 1 si n == 0
                o n * factorial(n-1)


fact :: N -> N
fact(n) = prodSec(rango((1,n,1))
```

PASCAL

```
program factorial;
var num, fact, i: Integer;
begin
 readln (num);
 fact := 1;
 for i :=2 to  num do
    fact := fact * i;
 writeln('el factorial de ', num , ' es ', fact);
end.
```

RELACIÓN CON MATEMÁTICA

Property. $\forall$ `n` $\in$ `N`, `fact(n) = factorial(n)`.

The property is proved using the principle of structural induction:

1) Base case: prove Property for `n` $= 0$
2) Inductive case: $\forall$ `n` ≥ 0, if `fact(n) = factorial(n)` then `fact (n+1) = factorial(n+1)`
3) If 1) and 2) then Property holds.

Base case:

$$
\begin{aligned}
& \texttt{fact}(0) \\
= \; & 1 && \{ \text{ def. } \texttt{fact} \} \\
= \; & \texttt{factorial}(0) && \{ \text{ def. } \texttt{factorial} \}
\end{aligned}
$$

Inductive case:

$$
\begin{aligned}
& \texttt{fact}(n+1) \\
= \; & (n+1) * \texttt{fact}(n+1-1) && \{ \text{ def. } \texttt{fact} \} \\
= \; & (n+1) * \texttt{fact}(n) && \{ \text{ arithmetic } \} \\
= \; & (n+1) * \texttt{factorial}(n) && \{ \text{ Ind. Hypothesis } \} \\
= \; & (n+1) * \texttt{prodSeq}(\texttt{range}(1, n, 1)) && \{ \text{ def. } \texttt{factorial} \} \\
= \; & \texttt{prodSeq}(\texttt{range}(1, n+1, 1)) && \{ \text{ lemma } \} \\
= \; & \texttt{factorial}(n+1) && \{ \text{ def. } \texttt{factorial} \}
\end{aligned}
$$