




Happy Path

 2025-03   7 min.

Como visualizan un algoritmo?

Como visualizan un algoritmo?



"Early Return"



Early Return

Un *Early Return* es un **guard clause** que, como el nombre lo indica, es un retorno temprano, antes de siquiera iniciar el procesamiento.

"Early Return"



"Guard Clause"



Guard Clause

Un guard clause es como un early return, pero en medio del procesamiento.

"Guard Clause"



"if" Branch



"if-else" Branch



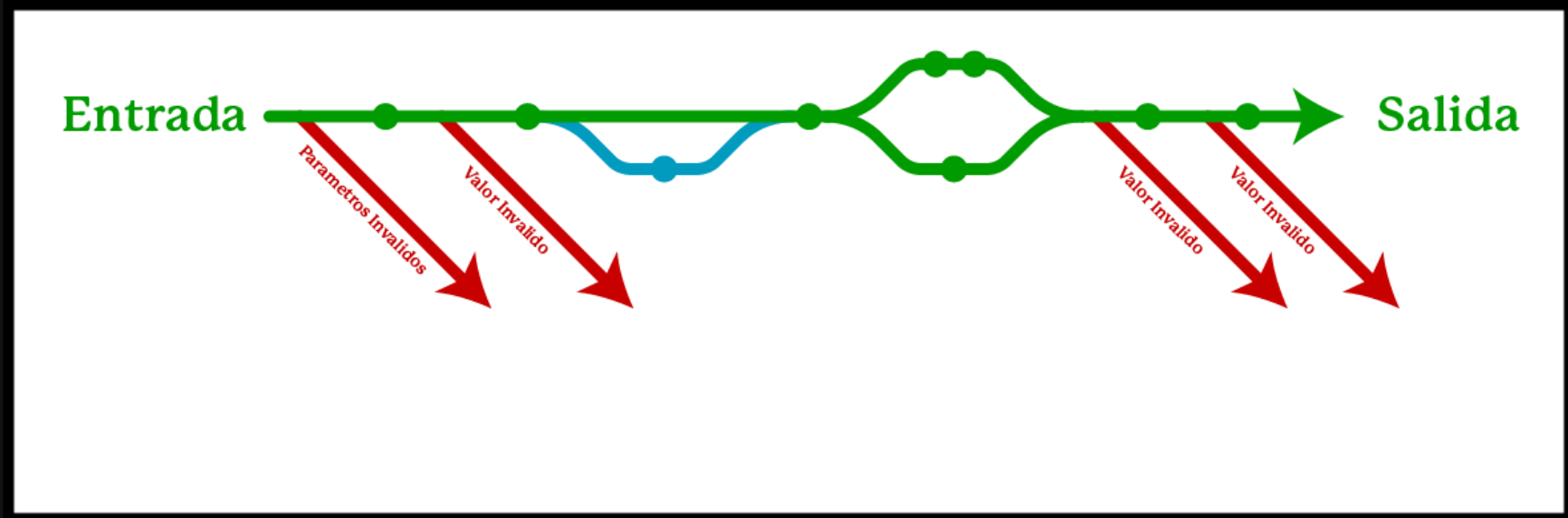
Guard Clause 2



Guard Clause 3



Happy Path



Lo que llamamos el "happy path" es el camino verde.

Es cuando todo va bien, todos los datos son correctos y el algoritmo va de inicio a fin de la manera mas directa.

Happy Path



Happy Path & Fail-fast System

Un sistema "fail-fast" (falla rapido) es un sistema que detiene o cancela la operación en vez de intentar continuar con la operación de manera posiblemente incorrecta.

Fail-fast System

"Si algo anda mal, cancela todo."

Fail-fast System

Entonces, un sistema con **Early Returns** y **Guard Clauses** es un sistema "Fail-fast".

Bouncer Pattern

Patrón de diseño donde se implementan **Early Returns** y **Guard Clauses** para evitar que se realice un proceso de manera incorrecta o datos incorrectos.

Saca su nombre de los "bouncers", el guardia de seguridad presente en las entradas a discotecas o clubs VIP, que solo permite la entrada a cierta gente.

Fail-fast System & Bouncer Pattern

Entonces un *fail-fast system* está usando el *bouncer pattern*.

Comúnmente en backend se utiliza el *bouncer pattern*, se construye un *fail-fast system*.

Ejemplo Muy Básico

```
function attemptLogin(username, password) {  
  
  // **espacio 1**  
  
  // codigo que busca el usuario en la base de datos  
  
  // **espacio 2**  
  
  // codigo que verifica la contraseña ingresada con la contraseña de la base de datos  
  
  // **espacio 3**  
  
  // codigo que responde con OK o Error si el login es exitoso o fallido.  
  
  // **espacio 4**  
  
}
```

1. Donde validamos que el usuario y contraseña pasados por parametro son validos? (Por ejemplo, no vacios.)
2. Donde validamos que el usuario existe en la base de datos?