AÑO 2026

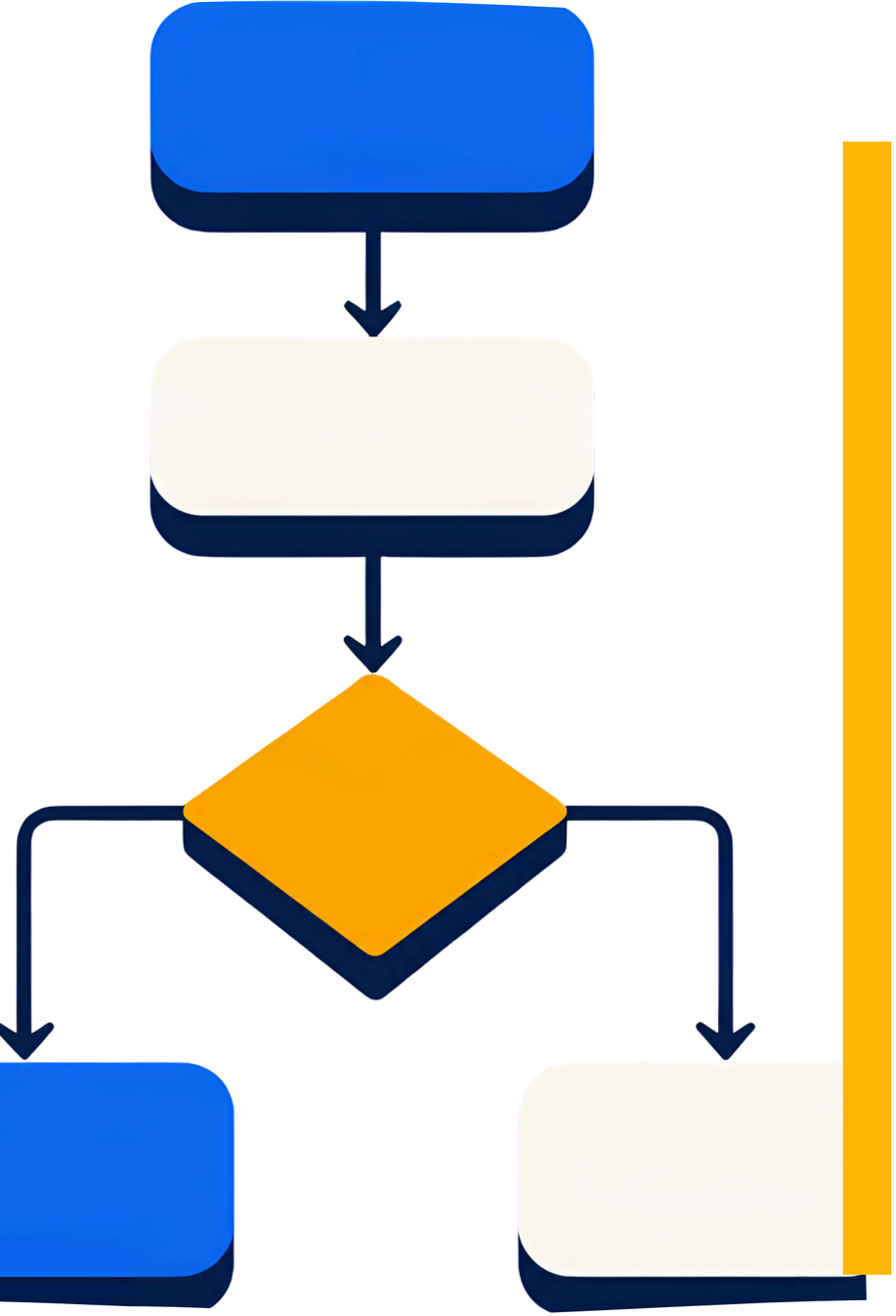
VERIFICACIÓN DE CONDICIONALES

Presentado por

Gabriela Del Mar Estupiñán Ramírez
Pool Alexander Delgado Palacio
Isabella Renteria Renteria

CONTENIDO

1. **Introducción**
2. **Definición y propósito**
3. **Conceptos fundamentales**
4. **¿Cómo funciona?**
5. **Relación con PRE y POST**
6. **Ejemplo paso a paso**
7. **Errores frecuentes**
8. **Proyecto integrado**
9. **Conclusión**



INTRODUCCIÓN

En un programa, los condicionales permiten tomar decisiones según una expresión sea verdadera o falsa. Cada resultado puede conducir a una ruta de ejecución diferente.

Por eso, verificar un condicional significa analizar las dos alternativas y comprobar que el comportamiento de cada una corresponda con lo que el programa debe hacer.

En esta exposición veremos qué significa verificar una decisión, cómo se relaciona con PRE y POST y cómo se analizan sus ramas.

¿QUÉ ES LA VERIFICACIÓN DE CONDICIONALES?

1 · DEFINICIÓN

La verificación de condicionales analiza las rutas producidas por una decisión para justificar que cumplen la especificación del programa.
Regla de Hoare: {P} C {Q} significa que, si P se cumple antes de ejecutar C y C termina, Q se cumple después.

RAMA VERDADERA

Se ejecuta cuando la condición se cumple.
{P ∧ B} C1 {Q}

RAMA FALSA

Se ejecuta cuando la condición no se cumple.
{P ∧ ¬B} C2 {Q}

La rama verdadera se analiza con P y B; la falsa, con P y ¬B. Ambas deben conducir a Q.

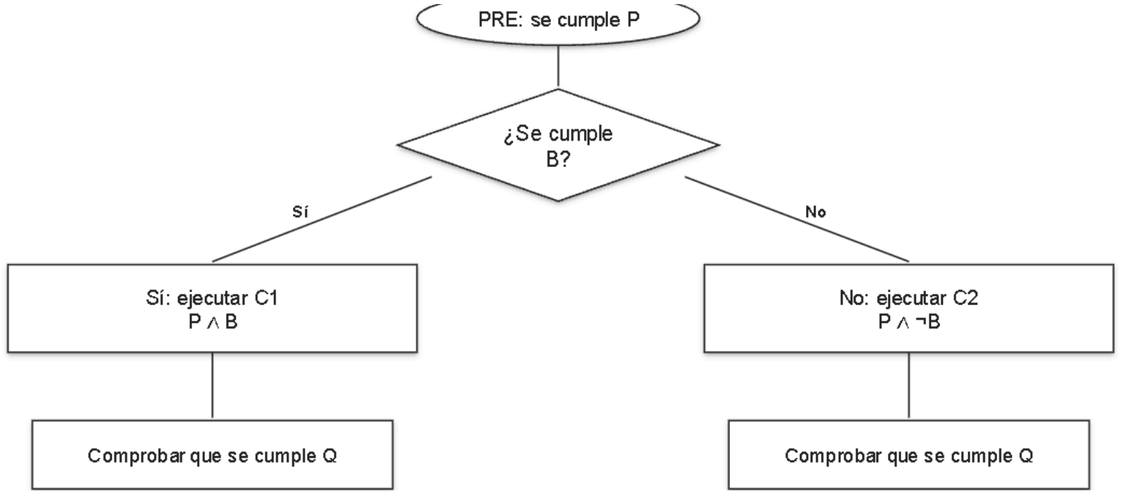
¿QUÉ ES LA VERIFICACIÓN DE CONDICIONALES?

2· PROPÓSITO

- El propósito es comprobar ambas rutas. Que una rama funcione no demuestra que todo el condicional sea correcto.

- La regla de Hoare expresa que la rama verdadera se analiza suponiendo P y B; la rama falsa, suponiendo P y ¬B. Las dos deben conducir a Q para justificar el condicional respecto de la especificación

$$\frac{\{P \wedge B\}\ C1\ \{Q\} \qquad \{P \wedge \neg B\}\ C2\ \{Q\}}{\{P\}\ \text{SI } B \text{ ENTONCES } C1 \text{ SINO } C2\ \{Q\}}$$



CONCEPTOS FUNDAMENTALES

RAMA → Cada alternativa de la decisión, verdadera o falsa.

CONDICIÓN → Expresión lógica que determina qué rama se ejecuta.

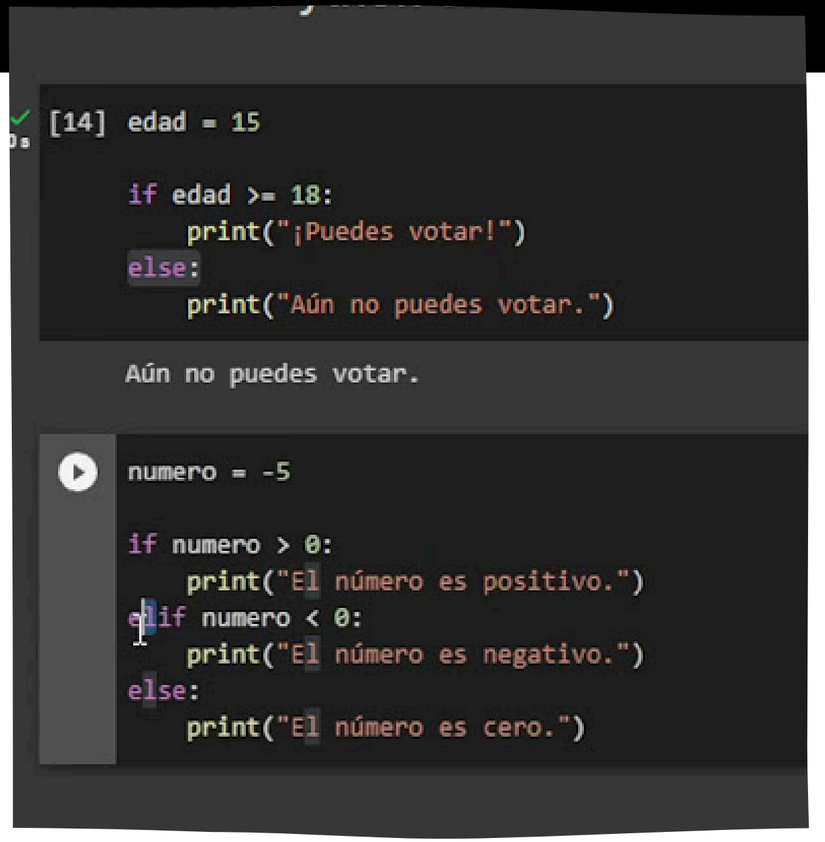
CONDICIONAL → Estructura que selecciona instrucciones según una condición.

PRECONDICIÓN → Afirmación que debe ser válida antes de ejecutar el código.

POSTCONDICIÓN → Afirmación que debe cumplirse al terminar la ejecución correcta.

ESPECIFICACIÓN → Descripción de las condiciones iniciales y del resultado esperado.

RUTA DE EJECUCIÓN → Secuencia de instrucciones que sigue el programa después de evaluar la condición.



```
[14] edad = 15

     if edad >= 18:
         print("¡Puedes votar!")
     else:
         print("Aún no puedes votar.")
```

```
Aún no puedes votar.
```

```
numero = -5

if numero > 0:
    print("El número es positivo.")
elif numero < 0:
    print("El número es negativo.")
else:
    print("El número es cero.")
```

¿CÓMO FUNCIONA?

IDENTIFICAR P

Establecer qué debe cumplirse antes de la decisión.

IDENTIFICAR B

Reconocer la condición que separa las rutas.

SEPARAR LOS CASOS:

Rama verdadera $P \wedge B$; rama falsa $P \wedge \neg B$.

COMPROBAR Q

Revisar si las instrucciones de cada rama producen el resultado esperado.

Se estudia cada alternativa por separado y luego se concluye si el condicional completo cumple la especificación. Este razonamiento por casos es la idea central de la regla de Hoare para condicionales

RELACIÓN CON PRE Y POST

Una operación se puede describir así:

PRE → operación → POST

La PRE indica qué debe cumplirse antes de ejecutar la operación. La POST describe qué debe cumplirse al terminar.

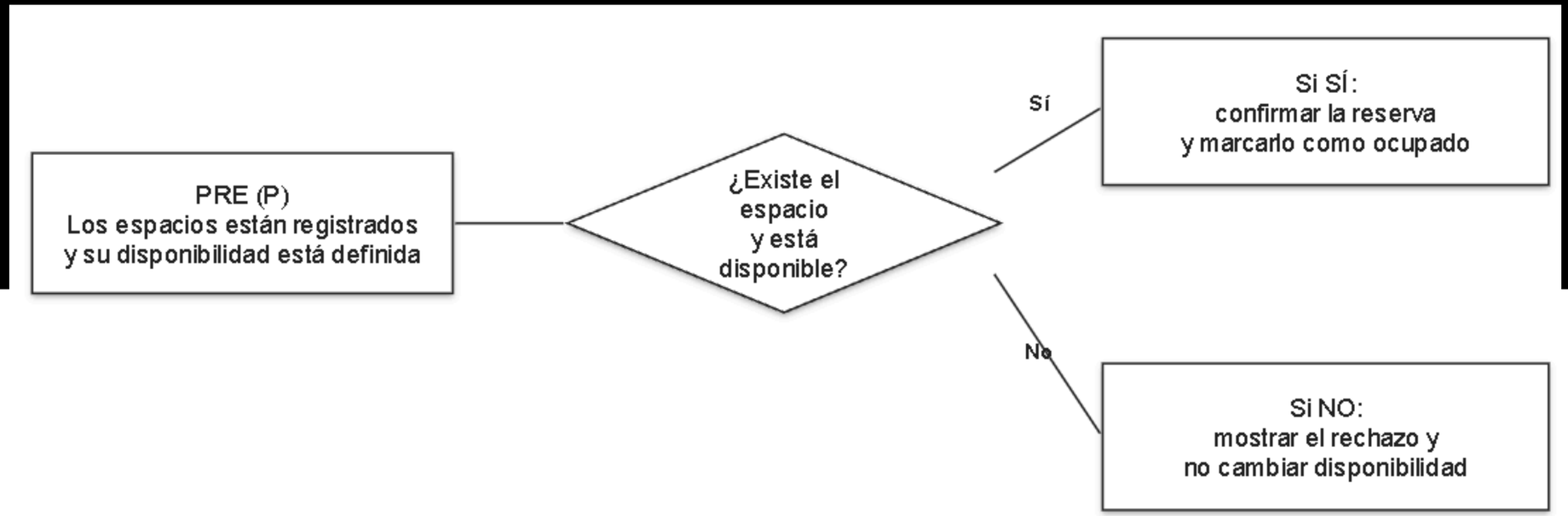
Cuando hay una decisión, el análisis se divide:

Rama verdadera: {P ∧ B} C1 {Q}
Rama falsa: {P ∧ ¬B} C2 {Q}

La precondición se combina con la condición correspondiente a cada rama. No basta con comprobar Q cuando B es verdadera: también hay que comprobarlo cuando B es falsa.

EJEMPLO, RESERVA DE ESPACIOS

Una universidad administra tres espacios para reservar: Sala 1, Sala 2 y Sala 3. Al iniciar el programa, la Sala 1 y la Sala 3 están disponibles y la Sala 2 está ocupada. El programa muestra la lista de espacios y le pide al usuario el nombre del espacio que desea reservar.



POST (Q): el resultado debe corresponder a la rama que se ejecutó.

EJEMPLO,RESERVA DE ESPACIOS

```
espacios = {
    "Sala 1": True,
    "Sala 2": False,
    "Sala 3": True
}
print("Espacios para reservar:")
for nombre in espacios:
     print("-", nombre)

espacio = input("¿Qué espacio desea reservar? ").strip()
if espacio in espacios and espacios[espacio]:
     espacios[espacio] = False
     print("Reserva confirmada")

else:
 print("No se puede reservar")
```

PRE: el diccionario está creado y sus valores representan la disponibilidad.
B: el espacio existe y está disponible.
El programa solicita un nombre y evalúa B.

RAMA	COMPROBACIÓN
Verdadera	Cambia el valor a False y confirma. Q: la sala queda ocupada.
Falsa	Muestra el aviso y no cambia el diccionario. Q: el diccionario no cambia.

ERRORES FRECUENTES

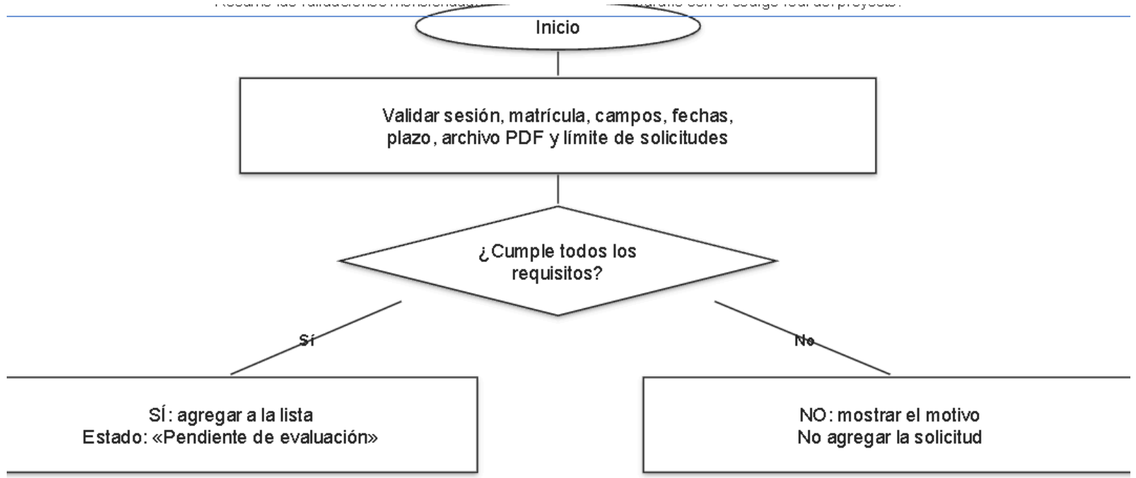
- VERIFICAR SOLO LA RAMA VERDADERA.
- OLVIDAR LA RAMA FALSA IMPLÍCITA (UN IF SIN ELSE).
- NEGAR MAL LA CONDICIÓN B CUANDO TIENE AND U OR.
- NO PROBAR LOS VALORES LÍMITE (> CONTRA >=).
- RAMA DE RECHAZO SIN RETURN, DE MODO QUE EL PROGRAMA SIGUE Y REGISTRA LA SOLICITUD.
- USAR UNA POSTCONDICIÓN QUE NO DESCRIBE AMBAS RAMAS.
- SUPONER UNA PRECONDICIÓN QUE EL PROGRAMA NO GARANTIZA.
- CREER QUE EJECUTAR UN CASO ES VERIFICAR.

PROYECTO INTEGRADO

Antes de registrar una solicitud, el condicional decide si cada validación se cumple.

PRE: los datos deben superar las validaciones del programa.

- Inicio de sesión y matrícula.
- Campos obligatorios y fechas.
- Máximo de cinco días hábiles.
- Archivo con extensión .pdf.
- No tener dos solicitudes para el mismo estudiante y asignatura.



PROYECTO INTEGRADO

```
if dias > 5:

 print("Solicitud rechazada: se superaron los cinco días hábiles.")

return

else:
 print("Plazo válido: continúa la verificación.")
```

SI SE ACEPTAN LAS VALIDACIONES

Se crea el diccionario nueva y se ejecuta lista.append(nueva).

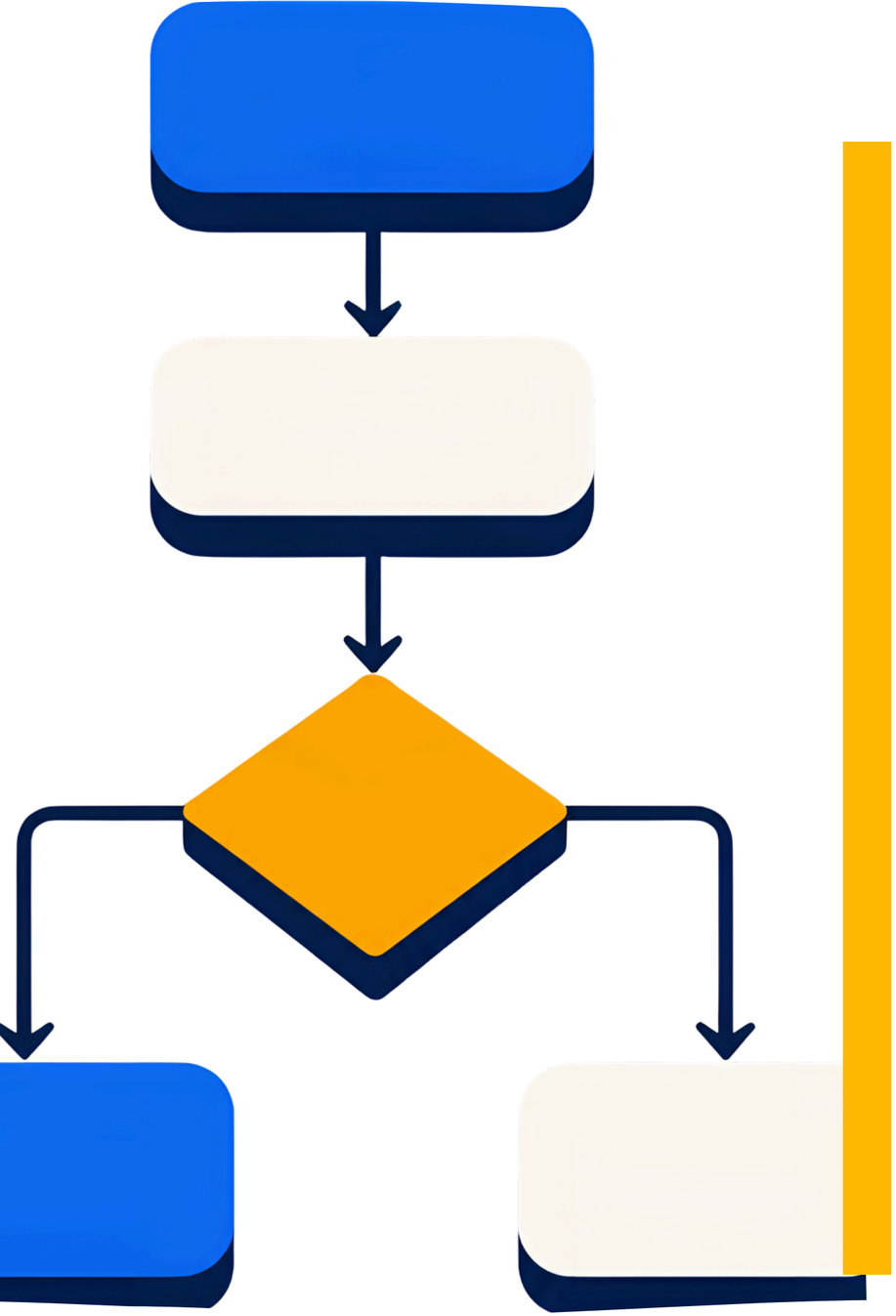
POST de aceptación: la solicitud queda en lista con estado “Pendiente de evaluación”.

SI SE RECHAZAN

Se informa el motivo y se sale de la función antes de agregar.

POST de rechazo: no se agrega una nueva solicitud a lista.

Si alguna validación falla, la función termina antes de llegar a esa instrucción.
Flujo de registro: validar, crear nueva, agregar a lista o informar el motivo.



CONCLUSIÓN

La verificación de condicionales permite analizar cada ruta de una decisión. La PRE indica qué debe cumplirse antes de ejecutar; la condición determina qué rama se sigue; y la POST define el resultado esperado al terminar. Para justificar el condicional, se revisan tanto la rama verdadera como la falsa. En el ejemplo de reserva, una ruta ocupa el espacio y confirma; la otra informa que no se puede reservar y no modifica su estado. En el proyecto de supletorios, las validaciones permiten registrar una solicitud o rechazarla antes de agregarla a la lista .

GRACIAS

POR SU ATENCIÓN