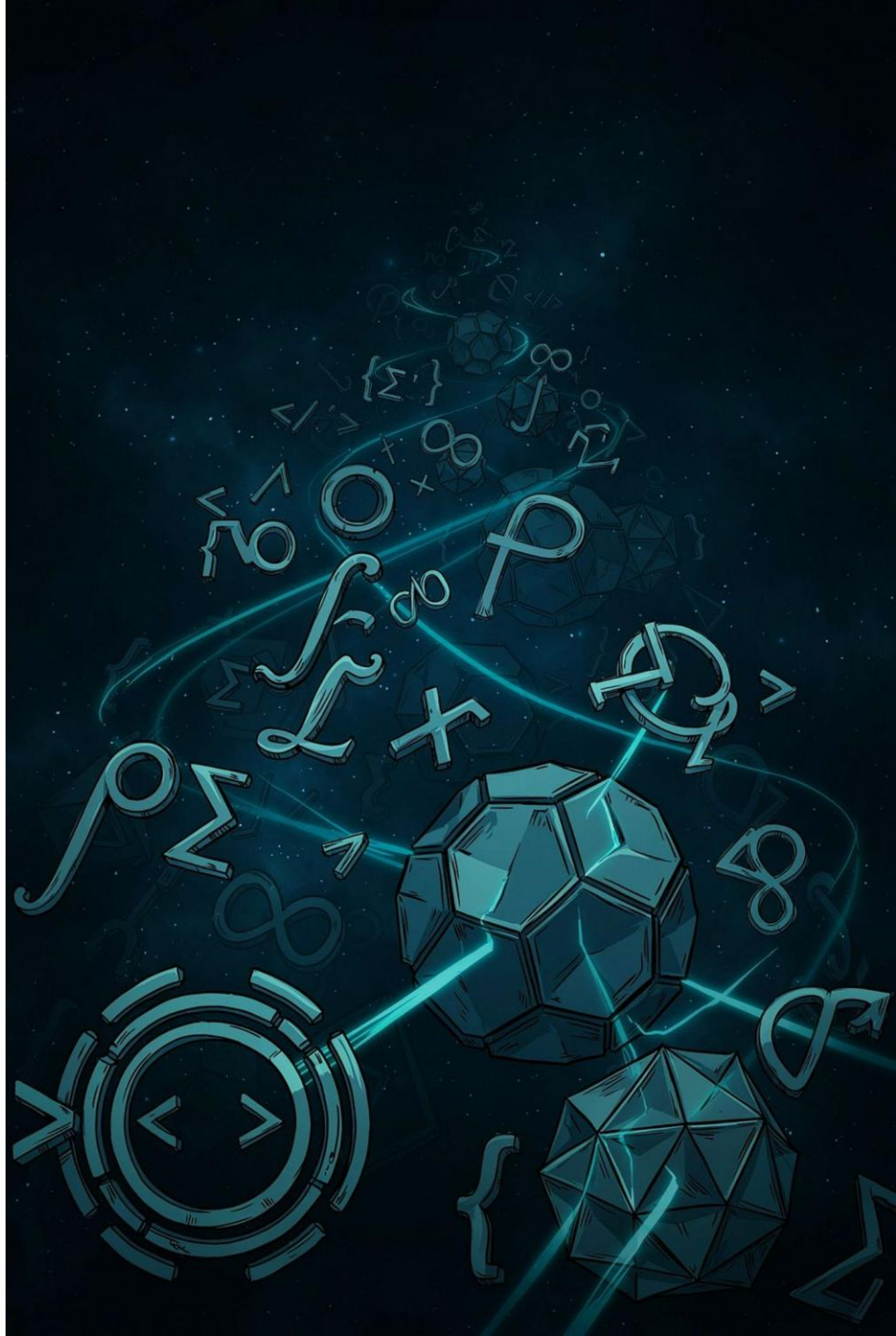
Verificación de Programas Secuenciales y Condicionales

Fundamentos de Verificación

DESARROLLO FORMAL DE SOFTWARE

INGENIERÍA DE SISTEMAS · UNIVERSIDAD DEL PACÍFICO



¿Qué conectan los dos temas?

Ambos tipos de programa comparten una misma pregunta esencial: **¿qué sabemos antes, qué ejecuta el programa y qué debe cumplirse después?**

Programa secuencial

Las instrucciones se ejecutan en un orden definido y cada una transforma el estado del programa.

Programa condicional

El resultado depende de una condición lógica y de la rama que se ejecute en cada caso.

Verificación común

En ambos casos comprobamos si las instrucciones preservan las propiedades especificadas.

by ykcaicedo



Verificación formal y pruebas no son lo mismo

Pruebas

Ejecutan el programa con casos concretos y comparan resultados. Pueden encontrar defectos, pero pasar algunos casos **no garantiza** que todos sean correctos.

Verificación formal

Razona con condiciones, aserciones y reglas lógicas para **justificar propiedades del programa** para todos los casos posibles.

La verificación también depende de que la especificación y las premisas sean adecuadas.

by ykcaicedo

PARTE 1

Verificación de Programas Secuenciales

01

Orden de ejecución

Las instrucciones se ejecutan una después de otra, en el orden escrito.

02

Transformación del estado

Se analiza cómo cambia el estado del programa tras cada instrucción.

03

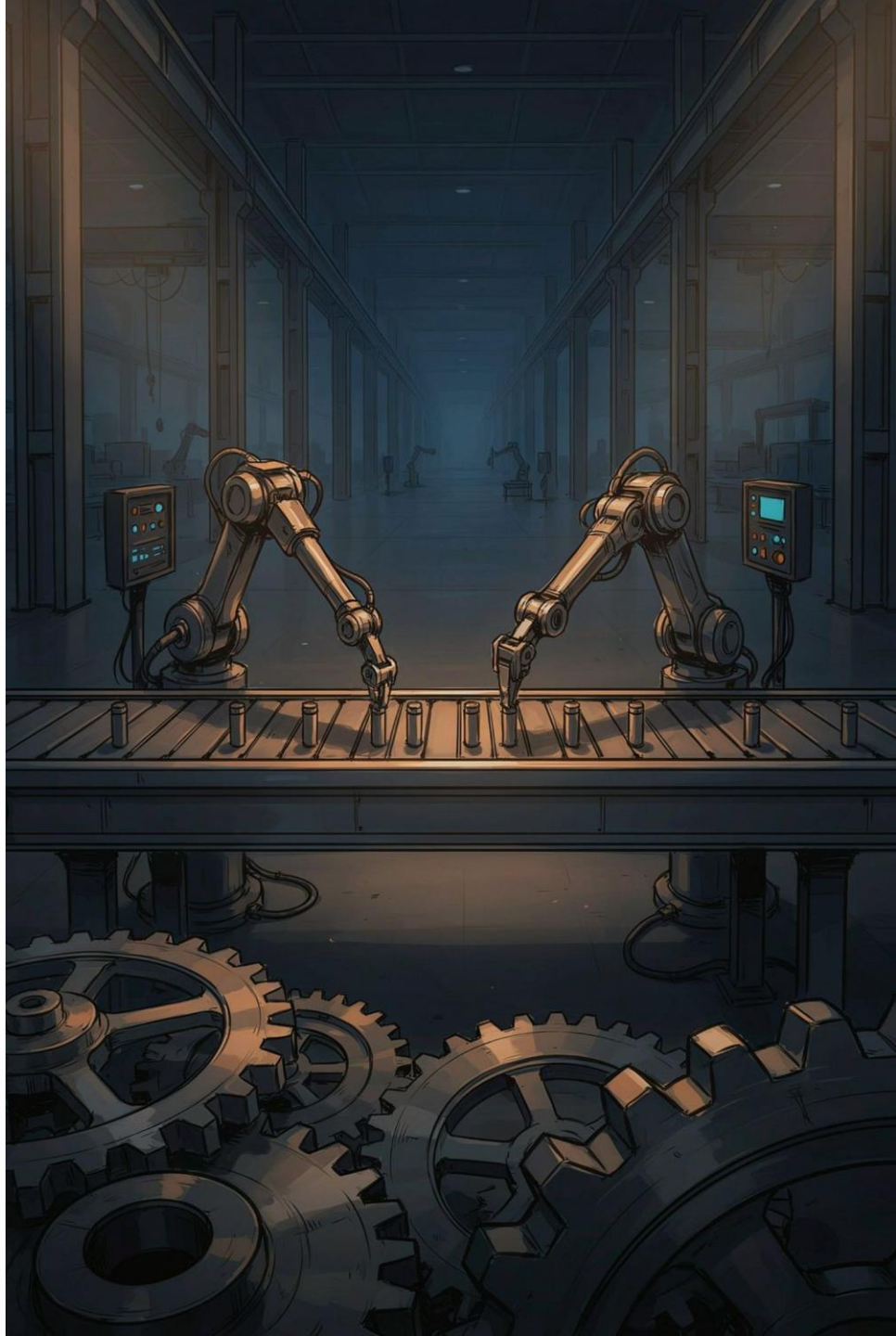
Relación pre–post

Se vinculan la precondición, las asignaciones y la postcondición esperada.

04

Estado acumulado

Una instrucción posterior utiliza el estado que dejaron las anteriores.



Ejemplo secuencial: calcular un subtotal

Precondición: $cantidad \geq 1 \land precioUnitario \geq 0$

```
1. subtotal := cantidad × precioUnitario
2. descuento := 0
3. total    := subtotal − descuento
```

Postcondición: $total = cantidad \times precioUnitario \land total \geq 0$

¿Qué justificamos?

Que la secuencia de tres asignaciones transforma el estado inicial —bajo la precondición dada— en un estado que satisface exactamente la postcondición especificada.

Paso 1

Calcula el valor bruto sin descuento.

Paso 2

Fija el descuento en cero explícitamente.

Paso 3

Deduce el descuento para obtener el total.

Seguimiento del estado paso a paso

Con **cantidad = 3** y **precioUnitario = 12 000**, la traza muestra cómo cada instrucción contribuye al resultado final.



Subtotal Descuento Total Postcondición

La traza confirma que el programa alcanza el estado final esperado partiendo de los datos iniciales definidos en la precondición.

Terna de Hoare para una secuencia

$$\{ P \}\ C\ \{ Q \}$$

P describe la condición inicial · **C** representa las instrucciones · **Q** expresa la propiedad esperada al terminar.

En una secuencia, la postcondición de una instrucción actúa como **condición intermedia** para razonar sobre la siguiente, encadenando el razonamiento de forma modular.

{ P }

Precondición: lo que sabemos antes de ejecutar.

C

Instrucciones: el código que transforma el estado.

{ Q }

Postcondición: la propiedad que debe cumplirse al terminar.

Hay que justificar que el razonamiento se sostiene bajo las condiciones definidas, no solo en casos concretos.

by ykcaicedo

PARTE 2

Verificación de Condicionales

Selección de rama

Un condicional selecciona una rama según el valor de una expresión lógica.

Análisis de ambas ramas

Se razona sobre el caso verdadero *y* el caso falso por separado.

Propiedad global

La postcondición debe justificarse para **todas** las ramas posibles.



Ejemplo condicional: validar una reserva

```
if (cuposSolicitados > 0
    && cuposSolicitados <= cuposDisponibles) {
    cuposRestantes = cuposDisponibles
                   - cuposSolicitados;
    estado = "ACEPTADA";
} else {
    estado = "RECHAZADA";
}
```

Objetivo de la verificación

Demostrar que, bajo cualquier entrada válida, el programa **nunca acepta más cupos de los disponibles** y que el estado resultante refleja correctamente la decisión tomada.

- Rama verdadera: descuenta cupos y marca como aceptada.
- Rama falsa: no modifica los cupos y rechaza la solicitud.

by ykcaicedo

Analizar las dos ramas



Condición verdadera

disponibles = 10, solicitados = 3 → cuposRestantes = 7, estado = "ACEPTADA". La invariante se mantiene: restantes ≥ 0.



Condición falsa — exceso

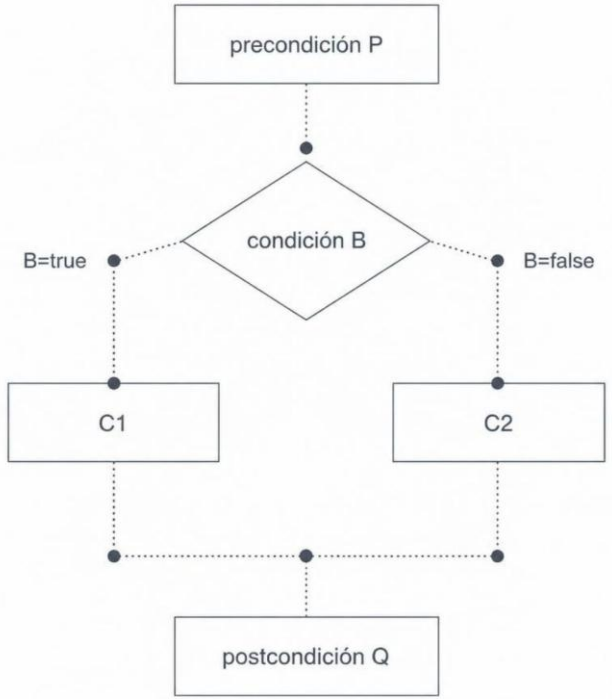
disponibles = 2, solicitados = 4 → estado = "RECHAZADA". Los cupos no se modifican.



Condición falsa — cantidad inválida

solicitados = 0 → estado = "RECHAZADA". La condición exige cantidad positiva.

Razonamiento de Hoare con condicionales



Regla del condicional

Para `if B then C1 else C2`, el razonamiento exige verificar **ambos caminos** de forma independiente:

Una PRE y una POST claras ayudan a detectar ramas incompletas o mal especificadas.

Si B es verdadera

C1 debe llevar al estado que satisface la postcondición Q.

Si B es falsa

C2 también debe llevar a Q, aunque recorra un camino distinto.

Errores frecuentes al verificar

Olvidar la rama falsa

Analizar solo el camino exitoso e ignorar los casos en que la condición es falsa.

Confundir prueba y demostración

Creer que pasar varios casos de prueba garantiza la corrección para todas las entradas.

Postcondición no derivable

Enunciar una postcondición que no se desprende lógicamente de las instrucciones del programa.

Traza incorrecta

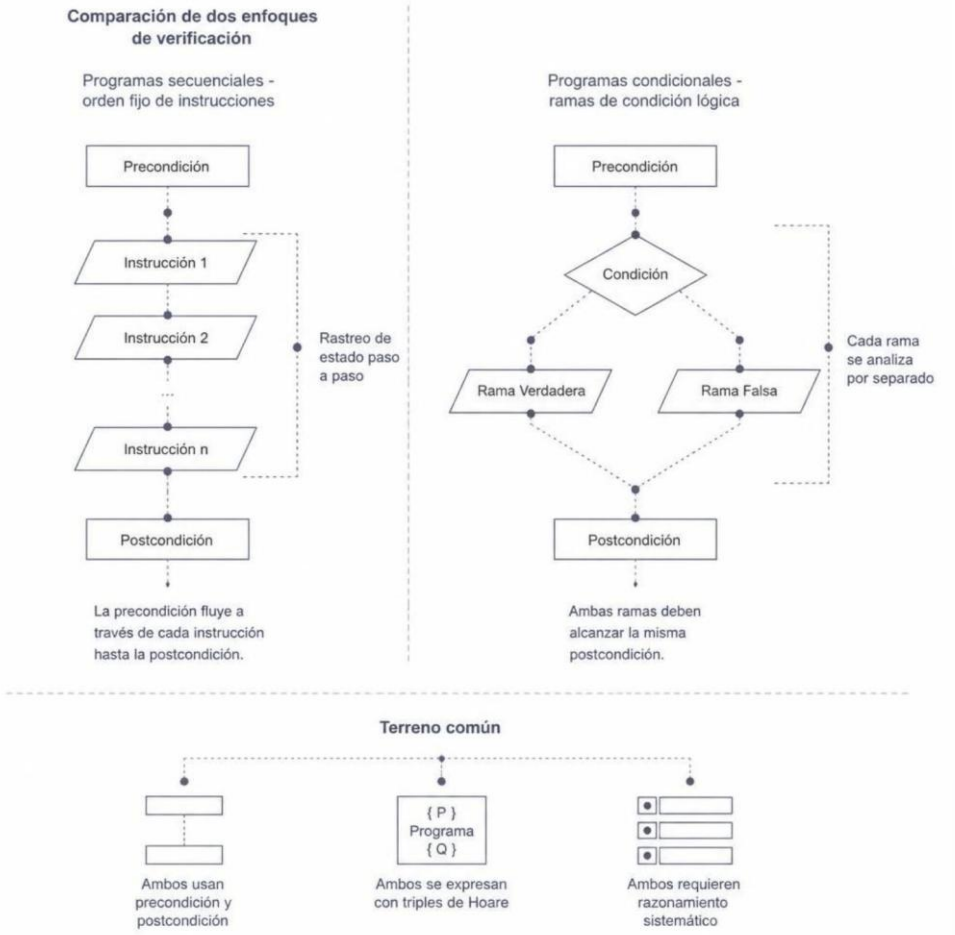
No actualizar correctamente los valores de las variables en cada paso de la traza.

Ignorar entradas límite

No contemplar entradas inválidas, valores nulos o casos en los extremos del dominio.

by ykcaicedo

Secuenciales y condicionales: comparación



Lo que comparten

Precondición explícita

Ambos exigen definir qué se asume antes de ejecutar el programa.

Postcondición verificable

Ambos requieren enunciar la propiedad que debe cumplirse al terminar.

Tripleta de Hoare

Ambos pueden expresarse y razonarse con el esquema `{P} C {Q}`.

> La verificación justifica propiedades; las pruebas aportan evidencia mediante ejecuciones concretas.

by ykcaicedo

PROYECTO INTEGRADO

Aplicación al Proyecto Integrado

01

Elegir una operación

Consultar disponibilidad, calcular costo o registrar una reserva.

02

Definir la precondición

¿Qué debe cumplirse antes de ejecutar la operación elegida?

03

Seguir las instrucciones o ramas

Trazar el estado paso a paso o analizar cada rama del condicional.

04

Formular la postcondición

¿Qué propiedad debe cumplirse al terminar la operación?

05

Revisar casos

Contemplar casos normales, límite e inválidos pertinentes al dominio.

Preguntas para comprobar comprensión

1

Secuencia vs. condicional

¿Qué diferencia fundamental existe entre la verificación de una secuencia y la de un condicional?

2

Ambas ramas

¿Por qué es imprescindible analizar la rama verdadera *y* la falsa en un condicional?

3

Valor de la traza

¿Qué información aporta una traza de ejecución que no da directamente el código?

4

Límites de las pruebas

¿Por qué una prueba exitosa no demuestra la corrección para todos los casos posibles?

5

Tripleta de Hoare

¿Cómo se conectan P, C y Q en `{P} C {Q}` y qué rol cumple cada parte?

by ykcaicedo

Cierre: una idea común

Programa secuencial

Justificar el **efecto acumulado** de las instrucciones sobre el estado del programa.

Programa condicional

Justificar el resultado de **cada rama** para garantizar la propiedad final en todos los casos.

Principio unificador

Partir de condiciones **explícitas** y comprobar sistemáticamente la propiedad final.

> Verificar es **razonar sistemáticamente** sobre la relación entre especificación y programa, no simplemente ejecutar y observar.

