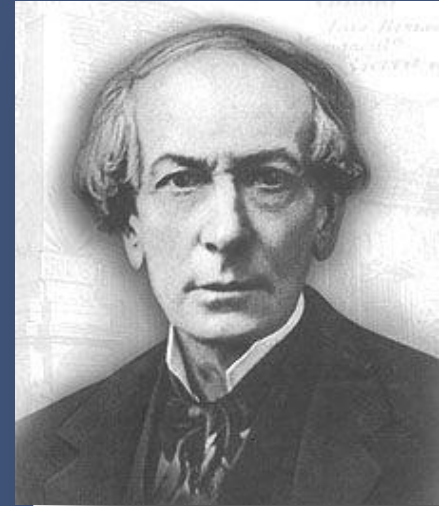


Sistema electrónico de enclavamiento para cambios de vía ferroviarios

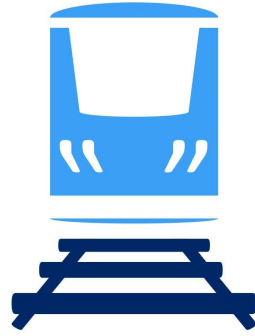


GICSAFe

“ Él hará la unidad de la República mejor que todos los Congresos. Estos podrán declararla una e indivisible pero sin el camino de fierro que acerque sus extremos remotos, quedará siempre divisible y dividida contra todos los Decretos Legislativos



Juan Bautista Alberdi (1810-1884).



GICSAFe

CONICET-GICSAFe

Grupo de Investigación y Control para la Seguridad y
Aplicaciones Ferroviarias



5% del PBI

Se moviliza por ferrocarril

+50 años

Antigüedad de las tecnologías

+5 millones de dólares

Costo de un sistema de enclavamiento

1

Introducción

Estado actual de la red ferroviaria



Red ferroviaria argentina

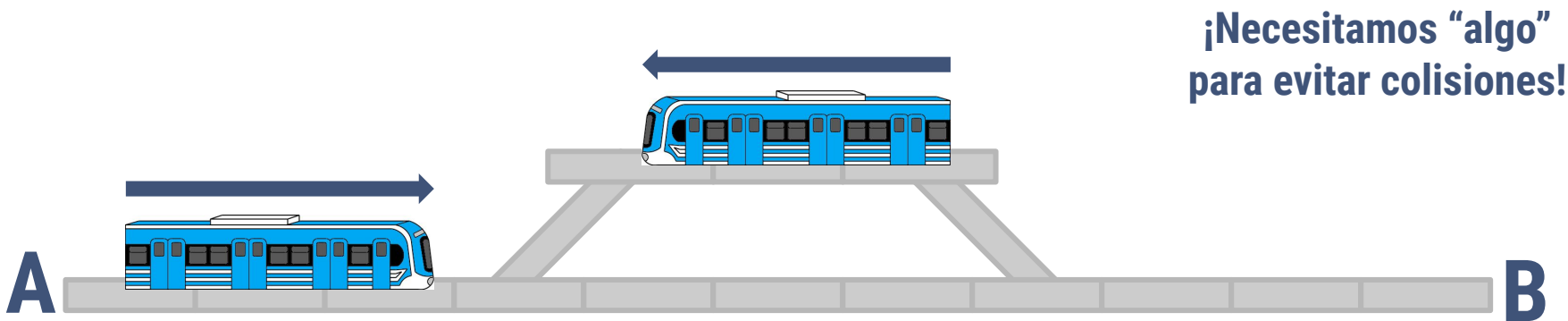


La red ferroviaria requiere diversas mejoras.

En particular, en los sistemas para evitar colisiones de trenes.



Red ferroviaria - Bypass



Circulación en ambas direcciones utilizando solo una vía.

Sirve para cubrir largas distancias con vías simples.

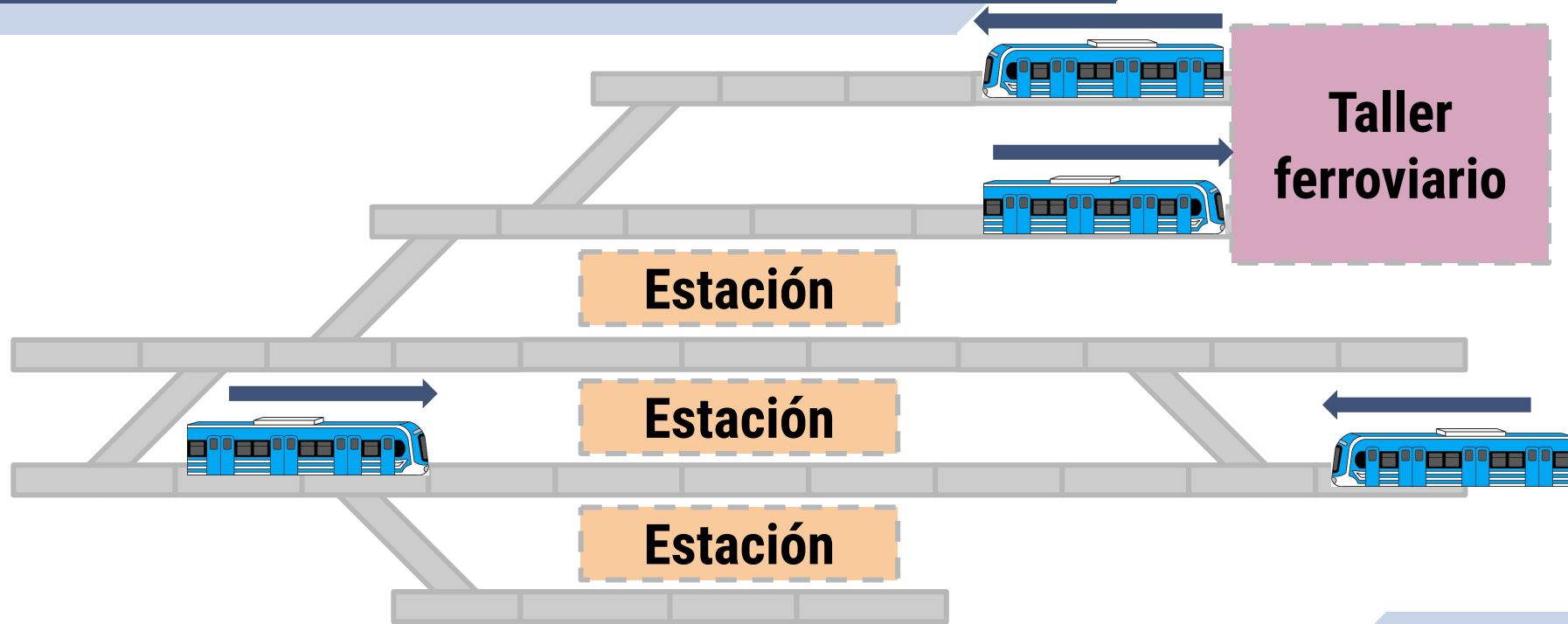


Estación ferroviaria típica





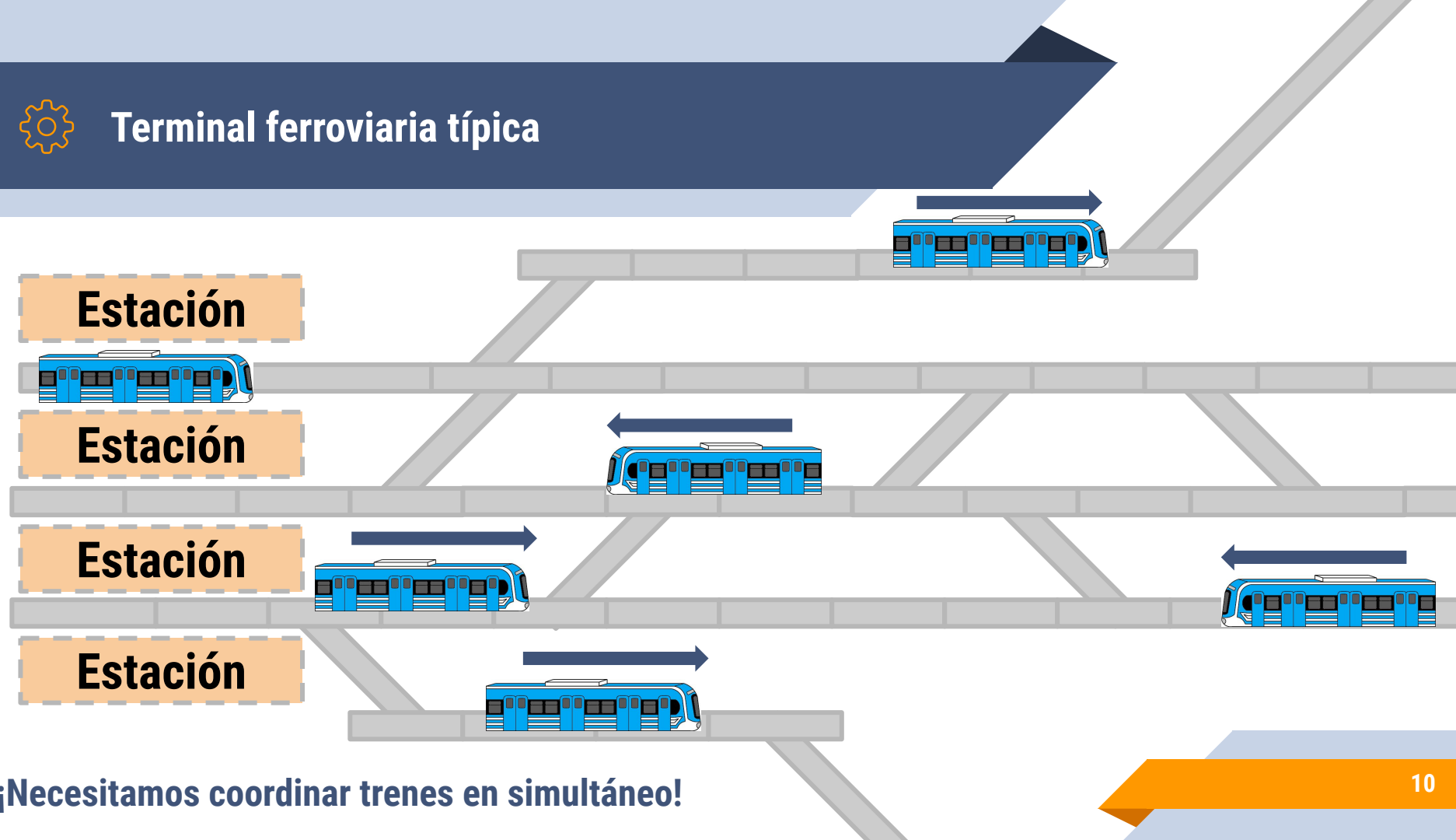
Hub ferroviario típico



¡Necesitamos coordinar trenes en simultáneo!



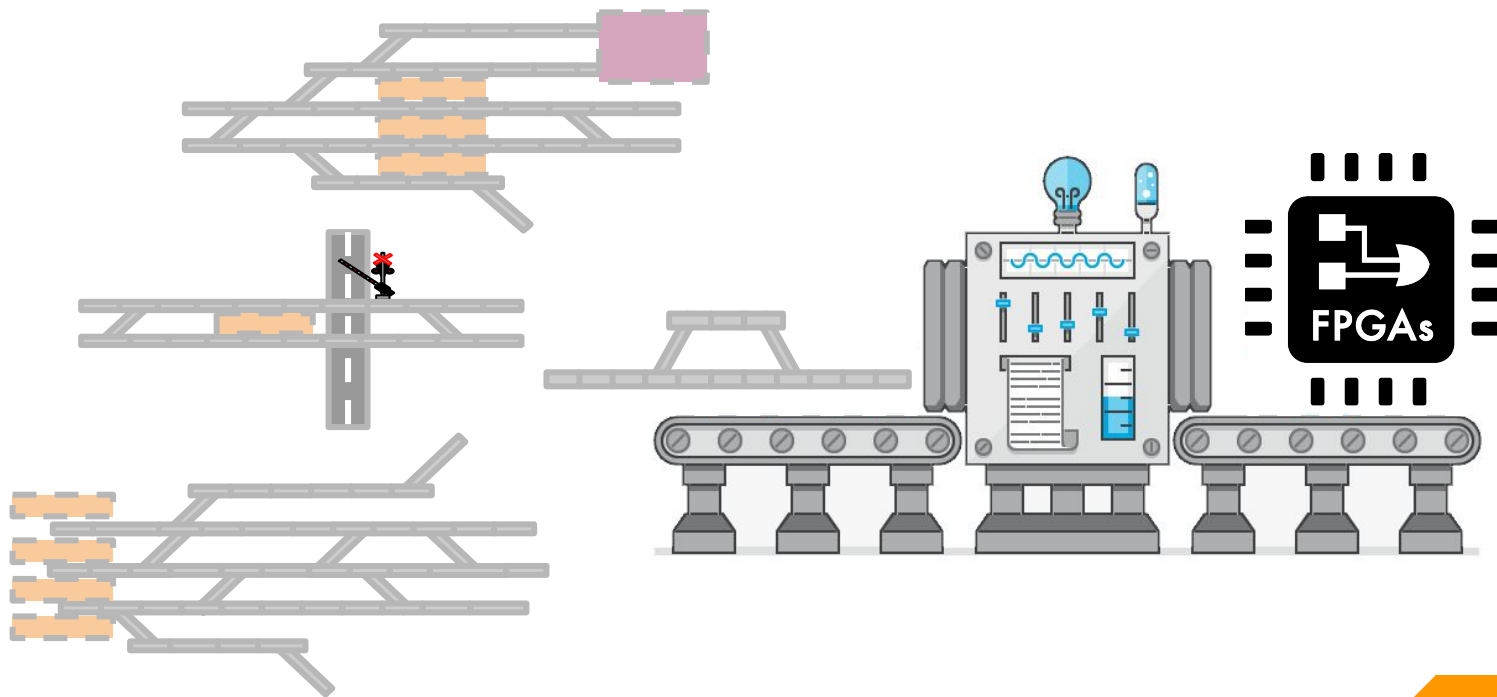
Terminal ferroviaria típica



¡Necesitamos coordinar trenes en simultáneo!



Necesitamos generar automáticamente la solución



2

Sistema ferroviario

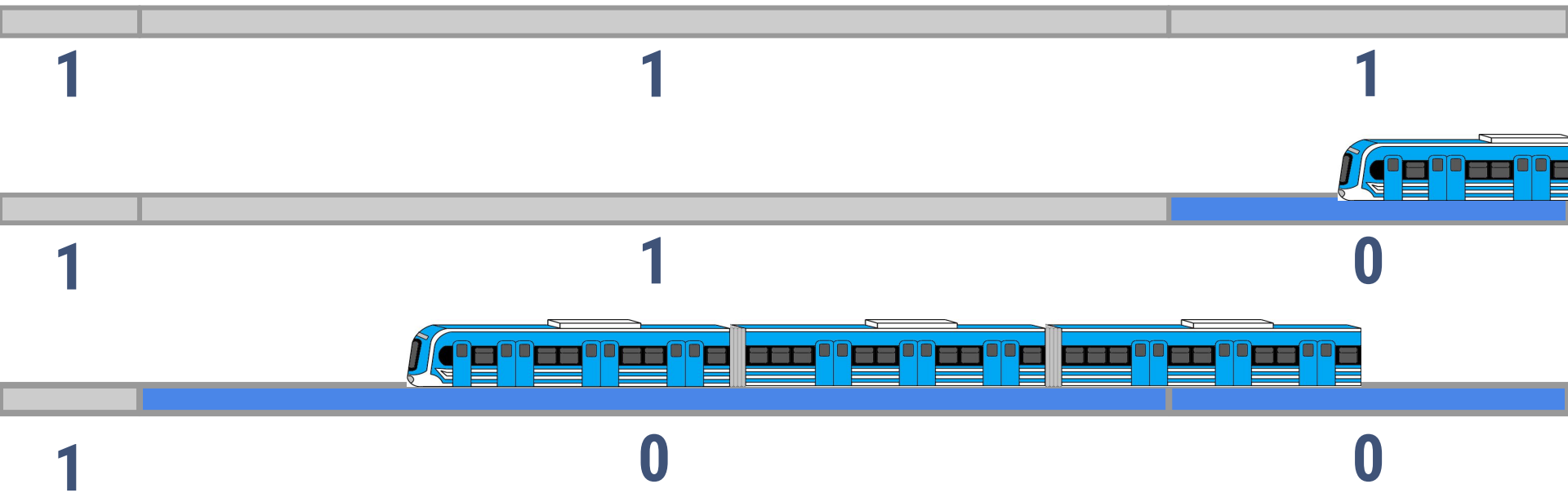
Componentes principales



Circuitos de vías

Ocupado (0:GND)

Libre (1:VCC)



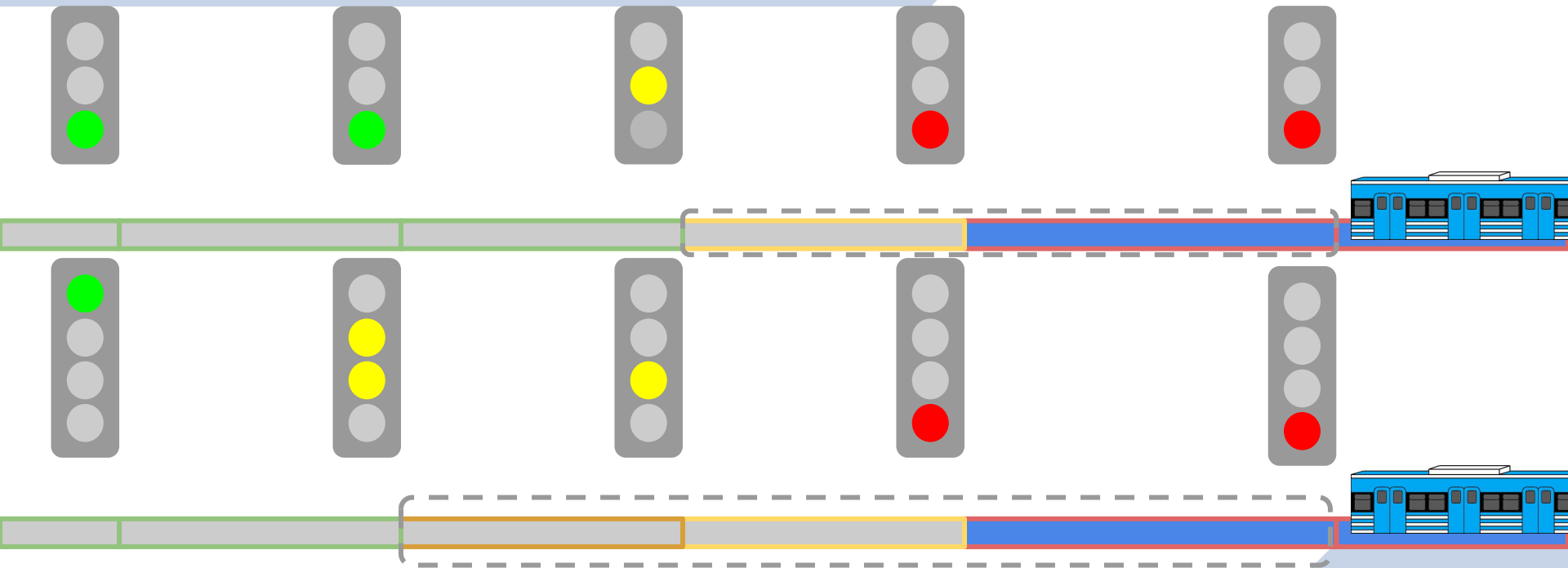
Usado para conocer dónde **PODRÍA** estar un tren.



Señales ferroviarias

No todas las secciones tienen.

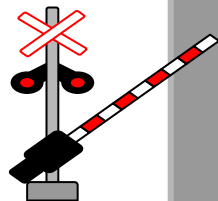
Algunas tienen más de una.



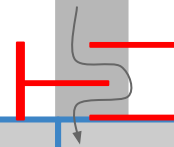
Usadas para **proteger** a la formación de **colisiones** o **descarrilamientos**.



Pasos a nivel



Cruce peatonal

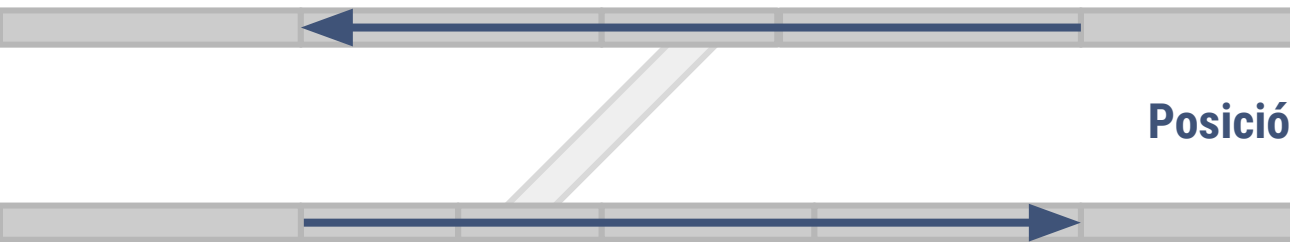


Cruce vehicular

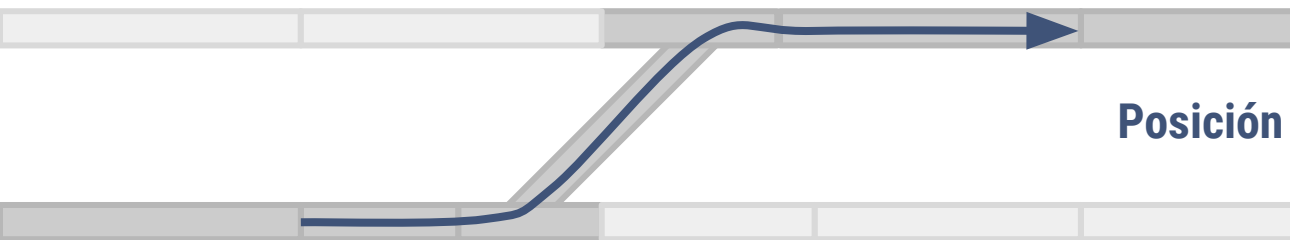


Cambios de vías

Permite acceder a diferentes vías.



Posición normal : circulación directa.



Posición reversa : circulación ramificada



Función del sistema de enclavamiento

- Mantener una **distancia de seguridad** entre los trenes.
- **Proteger a los trenes** en cruces y cambios.
- **Evitar descarrilamientos** asegurando los cambios.
- **Regular la circulación** de los trenes.
- Vinculan elementos de señalización.
- Lógica de **control vital**.
- Garantizar comportamiento **fail-safe**.



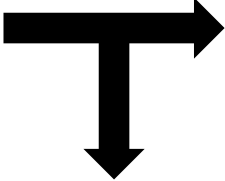
Los enclavamientos en Argentina tienen entre 40 y 100 años de uso.



Fail-Safe



Puedo hacer un sistema vital con componentes comerciales? **NO!**

Si ocurre una falla  **SIN** efectos **fatales**

Puede ser más **restrictivo**



Un sistema es **vital** si para **todos** sus modos de falla no existe ninguno fatal.



Mesa de mando



- Mover cambios.
- Comandar señales.
- Establecer/ cancelar rutas.
- Constituidos por HMI.
- Puede tener automatismos.
- NO es crítico.



Tecnologías de enclavamientos

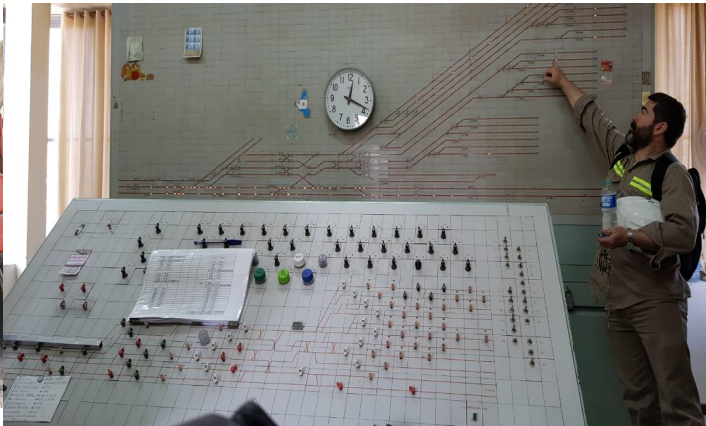


Clave para evitar
las colisiones

Mecánico

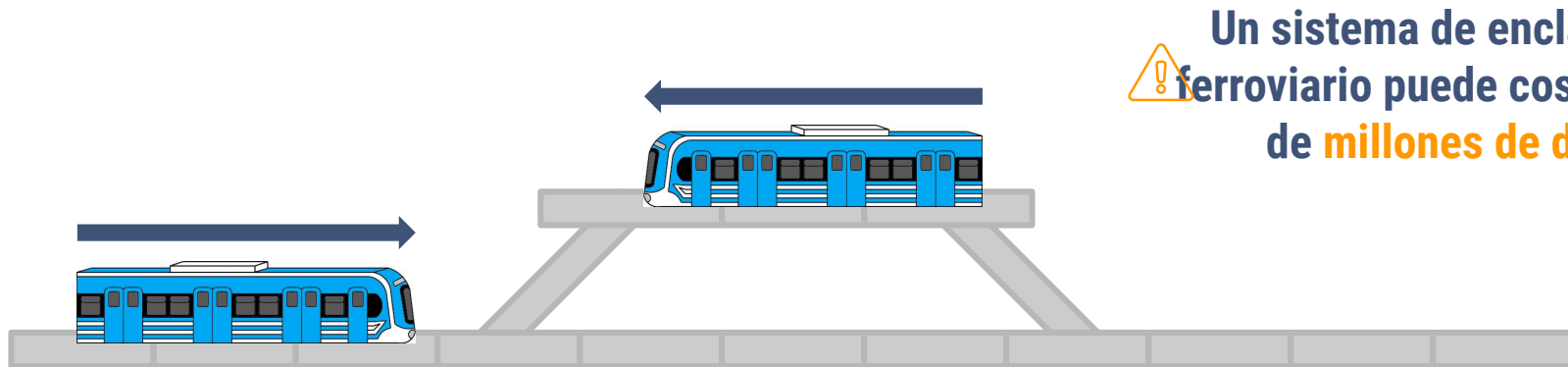
Electromecánico

Electrónico





Sistema de enclavamiento ferroviario



Un sistema de enclavamiento ferroviario puede costar decenas de **millones de dólares**.

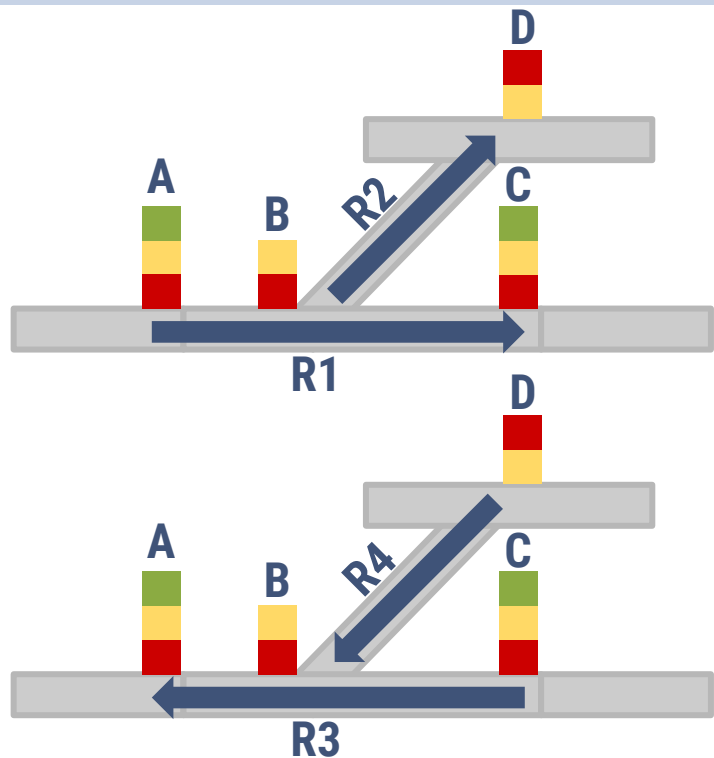
El sistema de enclavamiento tiene que **proteger** al tren de **colisiones** desde atrás, coordinar todo el **señalamiento** (señales ferroviarias, barreras, cambios de vías) y evitar **descarrilamientos**.



¿Qué es una tabla de enclavamientos?



Ruta: camino entre dos semáforos consecutivos.



	Señal inicial	Señal final	Cambio	Ruta bloqueante
R1	A	C	N	R2 R3 R4
R2	B	D	R	R1 R3 R4
R3	C	A	N	R1 R2 R4
R4	D	B	R	R1 R2 R3

3

Funcional vs Geográfico

CISC vs RISC: enfoques en enclavamientos



Modelado del sistema



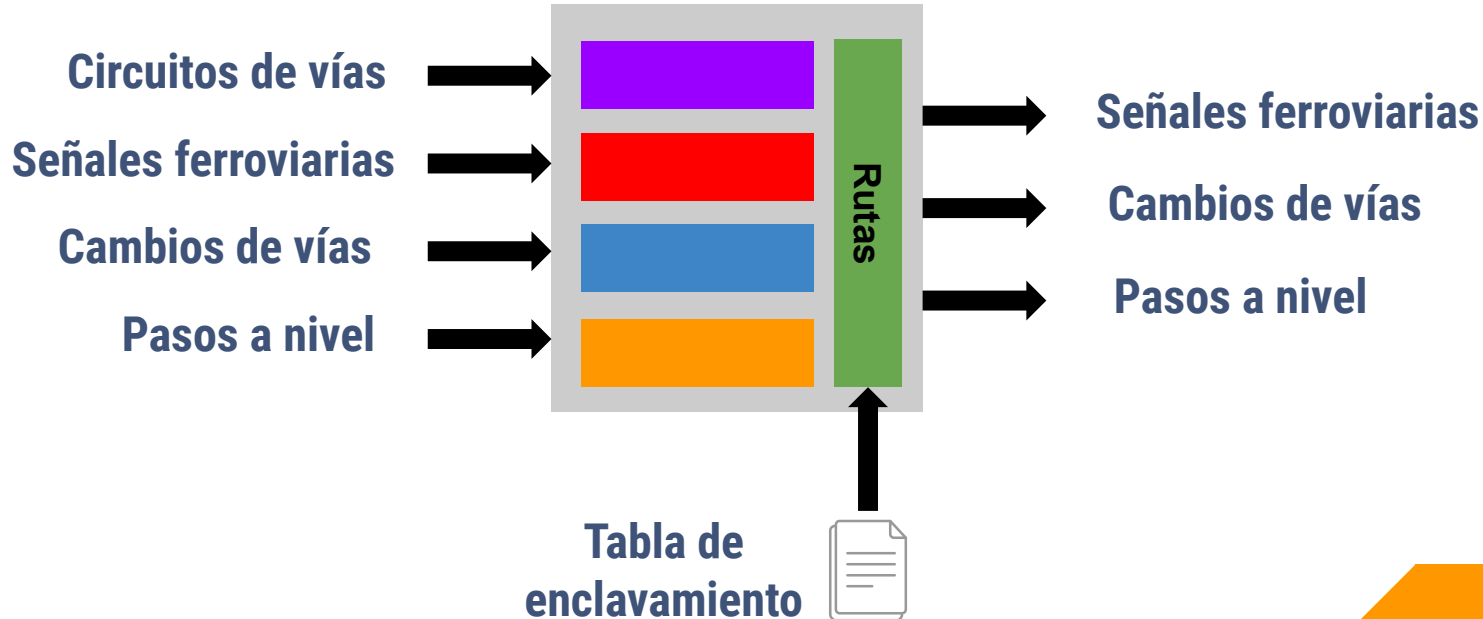
Los circuitos de vías son de **solo lectura**.





CISC: el enfoque funcional

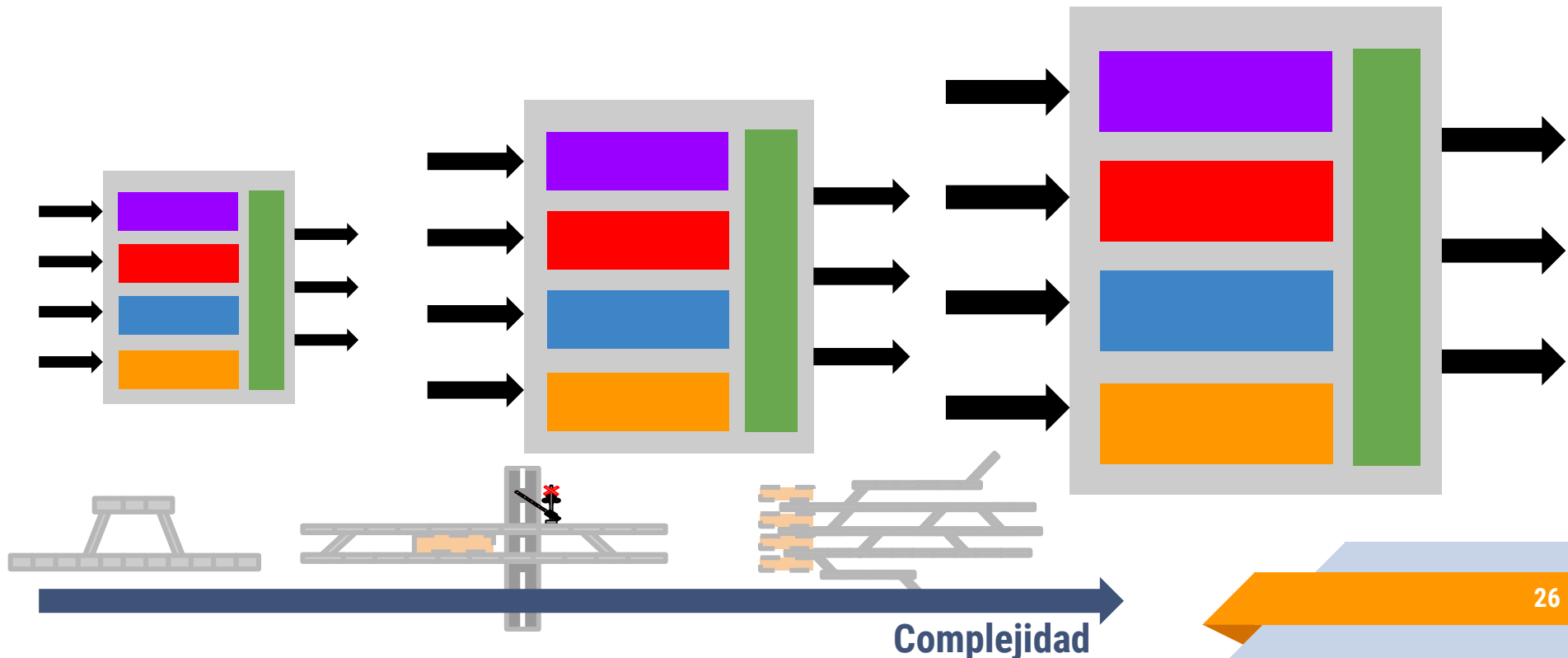
**“¡Una ruta para
dominarlas a todas!”**
J.R.R Tolkien





CISC: el enfoque funcional

Necesitamos
bloques **MÁS**
GRANDES!



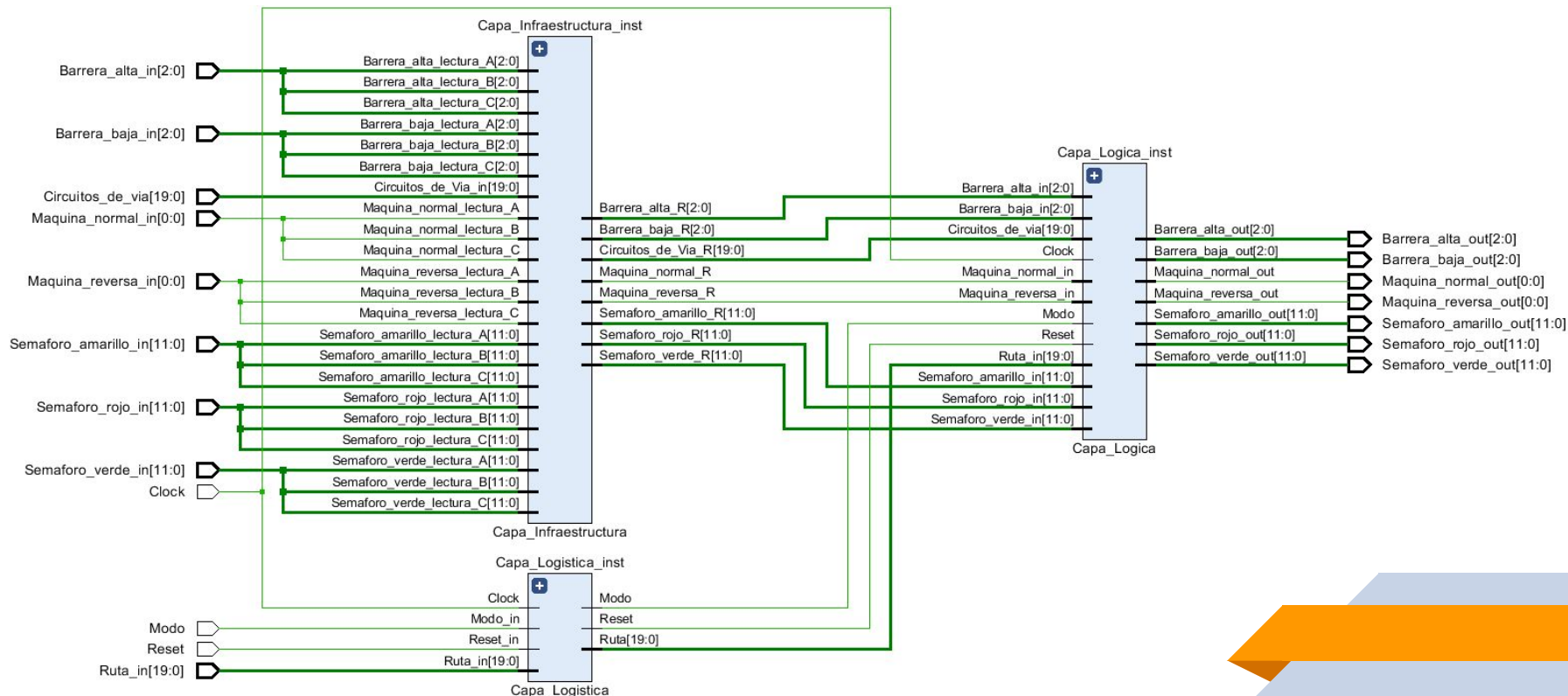


Enfoque funcional

No es escalable

No es mantenible

No es testeable

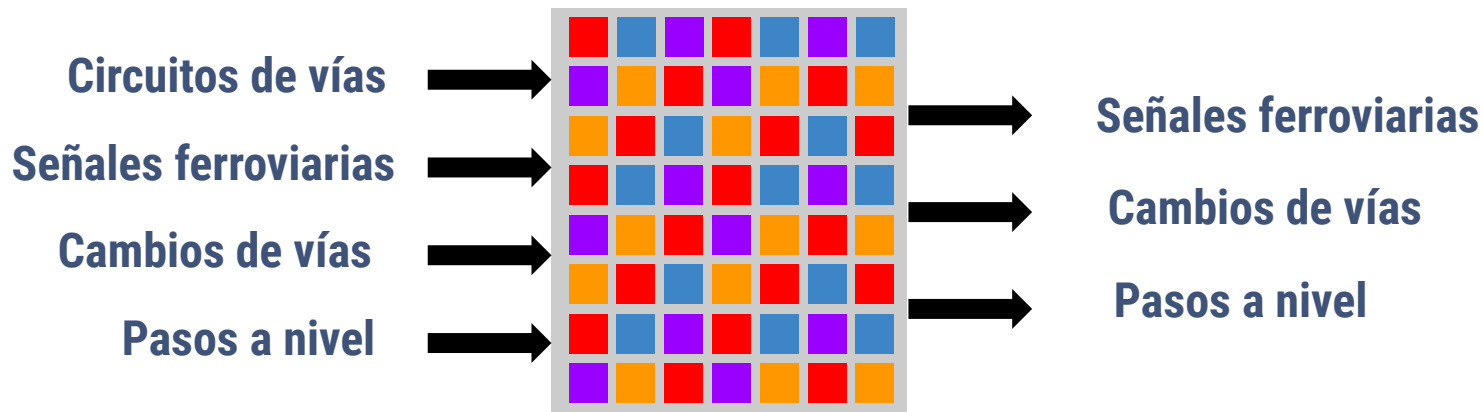




RISC: el enfoque geográfico

“¿Rutas? A donde vamos
no necesitamos rutas.”

Dr. Emmett Brown

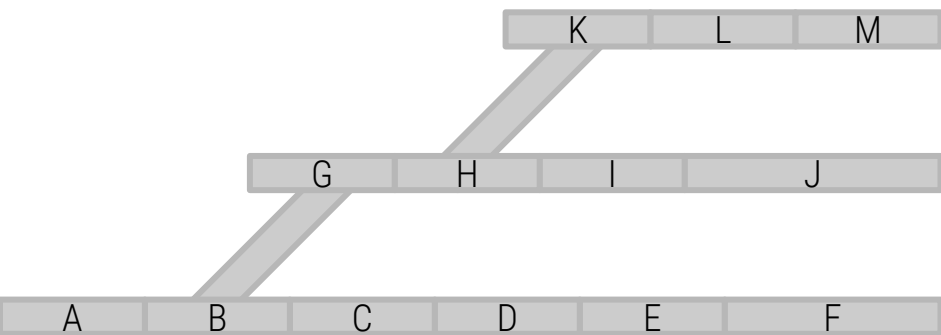


¿Puede un **auto** moverse sin un **camino**? SI

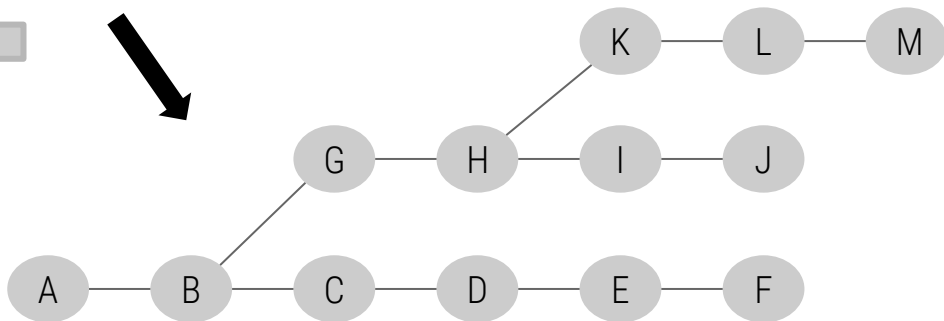
¿Puede un **tren** moverse sin una **vía**? NO



Topologías ferroviarias y redes de grafos



La información principal es la **conexión** entre las secciones

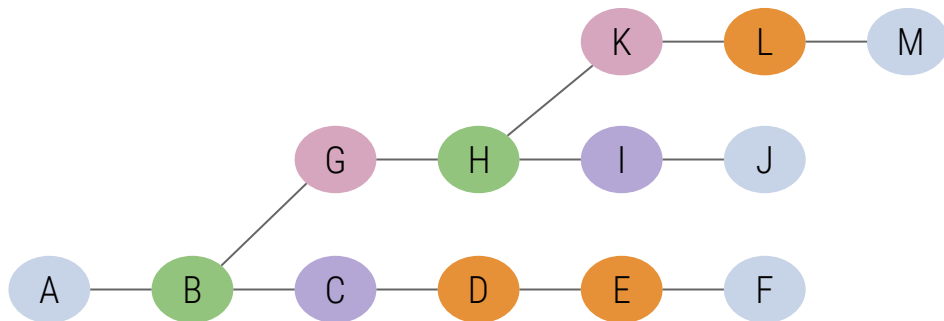




Analizador de redes ferroviarias



Criterio de clasificación desarrollado en el marco de este trabajo.



Nodo extremo



Nodo cambio raíz



Nodo cambio complemento rama



Nodo cambio complemento directo



Nodo simple

Podemos modelar cada bloque y sabemos cómo conectarlos ...

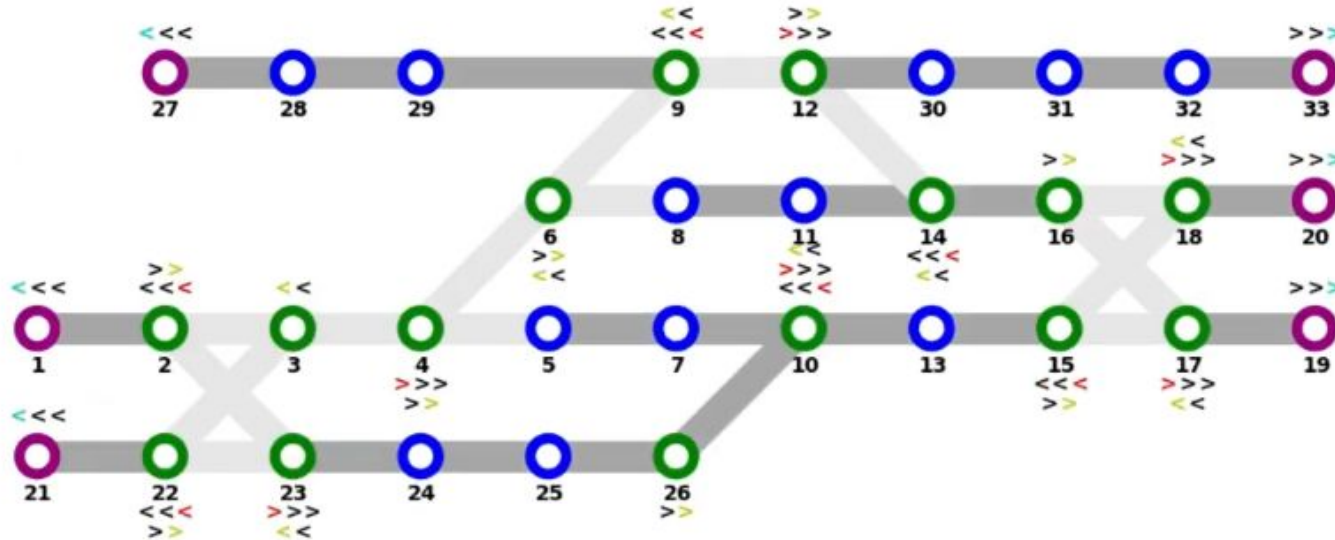
Es como jugar Lego!!



Analizador de redes ferroviarias



Implementado en Python en
el marco de este trabajo.



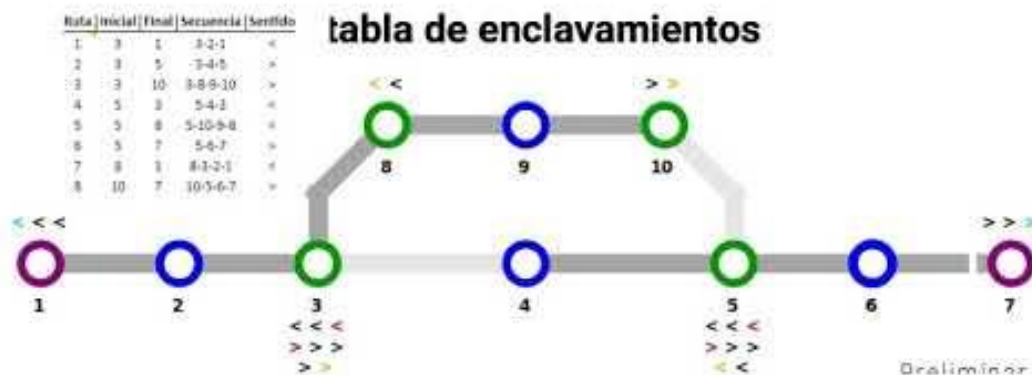
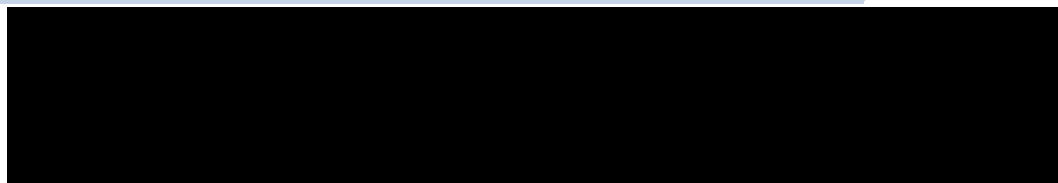
<https://www.youtube.com/watch?v=MFB-p1lhvYs>



Analizador de redes ferroviarias



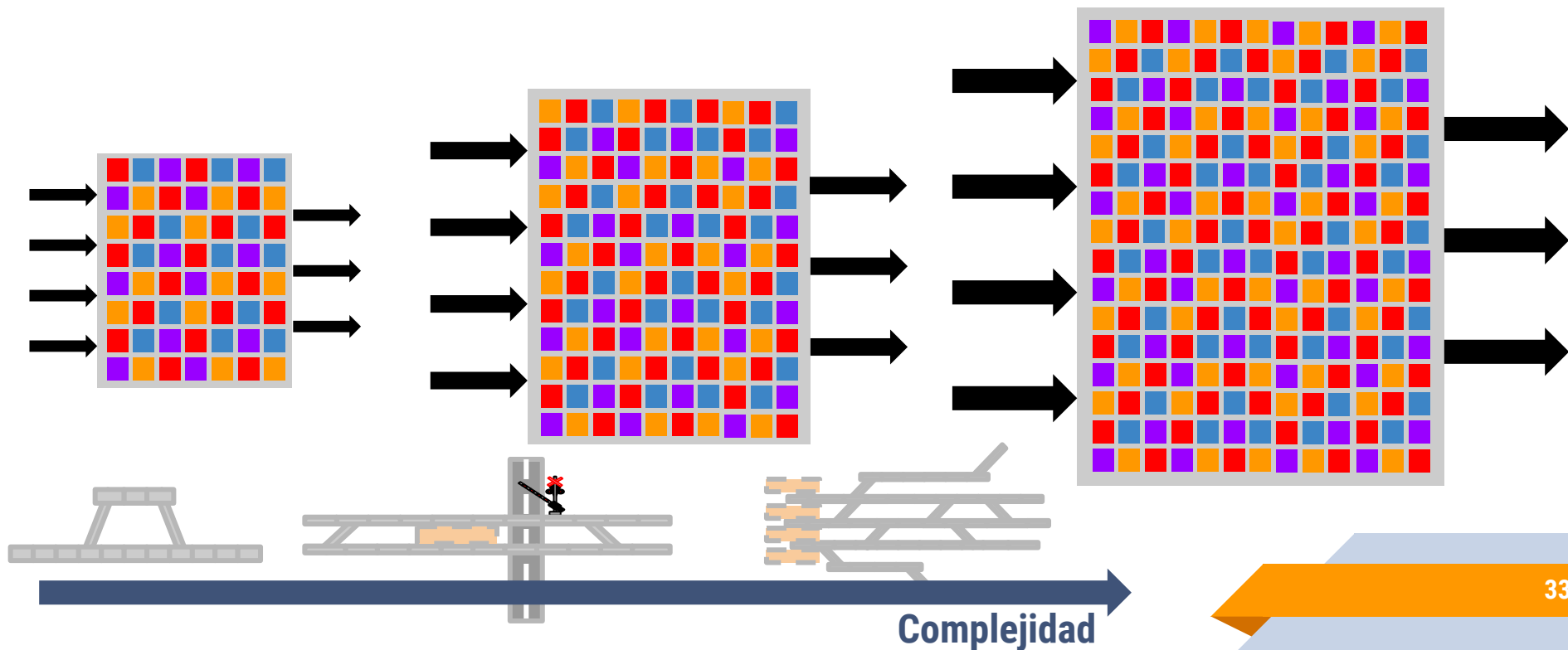
Implementado en Python en el marco de este trabajo.





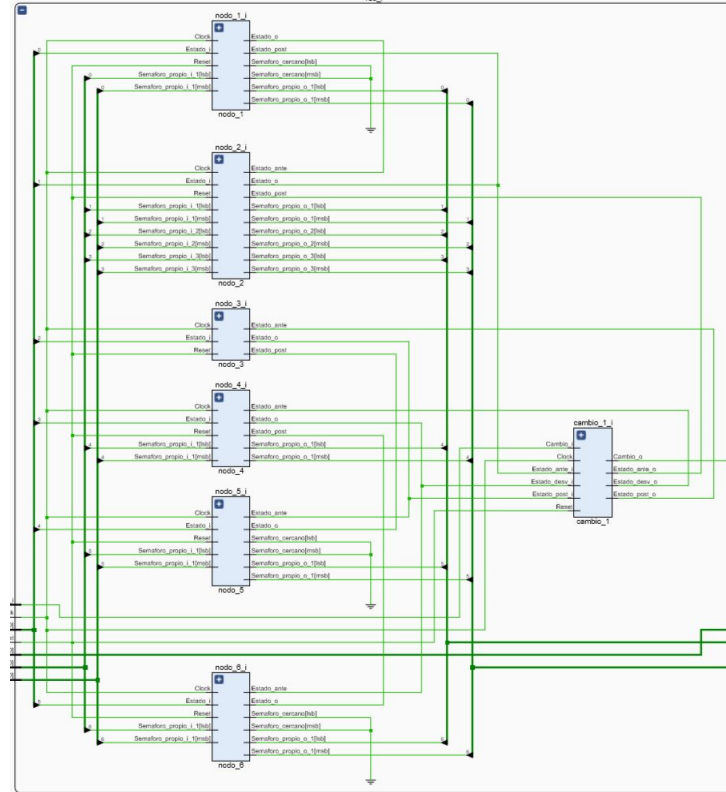
RISC: el enfoque geográfico

Necesitamos
MÁS bloques!





Grafo implementado en bloques



Cada uno de los nodos se procesa de forma **concurrente**.

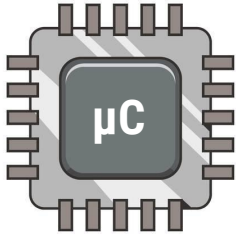


Tal como ocurre en la realidad.



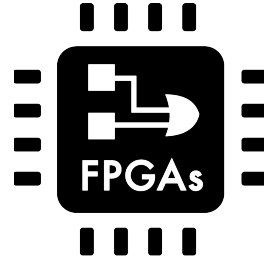
Criterio de selección de plataforma

Es esencial utilizar redundancias 2oo3



- Necesitamos más de 1 µC.
- Difícil de asegurar determinismo.
- Difícil de determinar la latencia.

¡La cantidad de componentes crece enormemente!

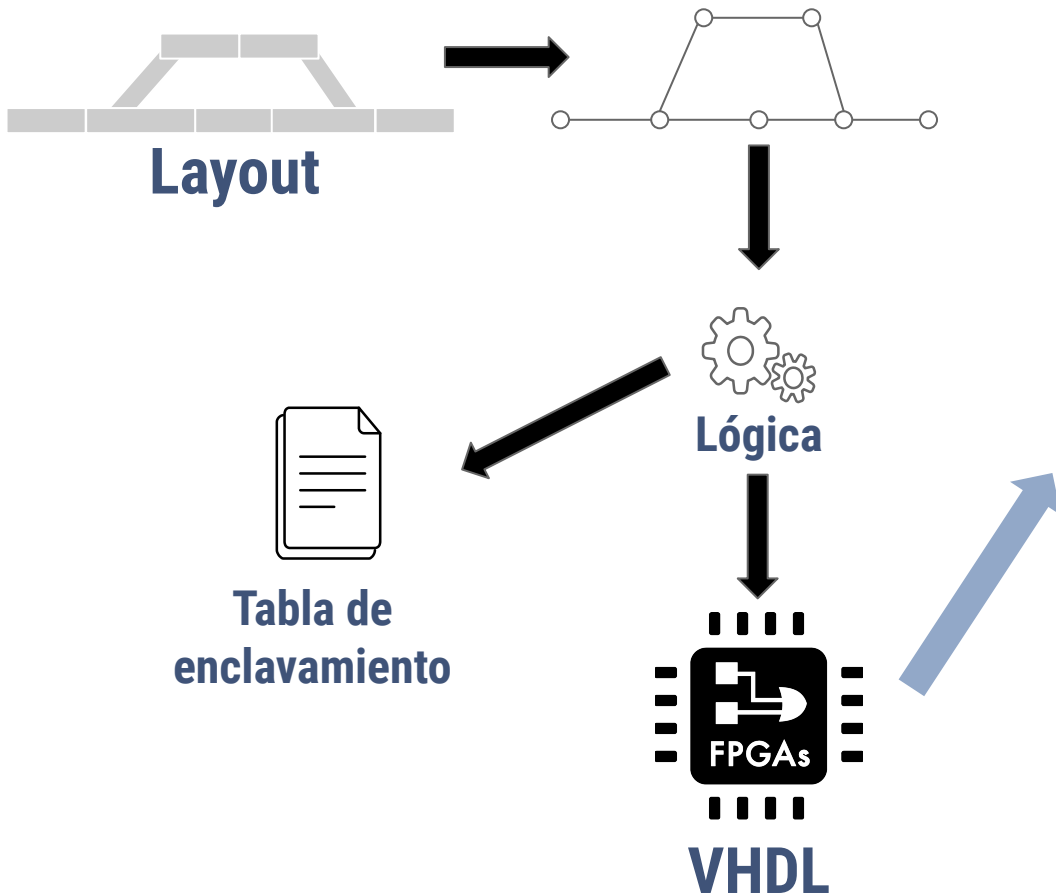


- Más conveniente para redundar.
- Determinístico, es HW puro.
- Mayor control de la latencia.
- Concurrencia real.

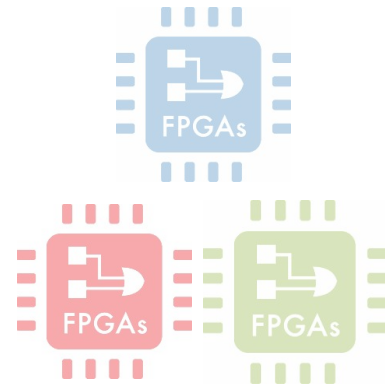


Enfoque geográfico - Consideraciones

- **Difícil** de implementar → Procesar el grafo puede ser complejo.
- **Modular** ↓
- **Completo** → Define **TODAS** las posibles circulaciones.
- Testing a priori → Test completos. **Certificado una vez.**
- Depende de la topología
- **Minimo uso de memoria** → Escalable.
- **Mayores** chances de ser sintetizable → Redundable.



Redundancia → **Disponibilidad**



(Safety Integrity Level)

→ **SIL 4**
Validación

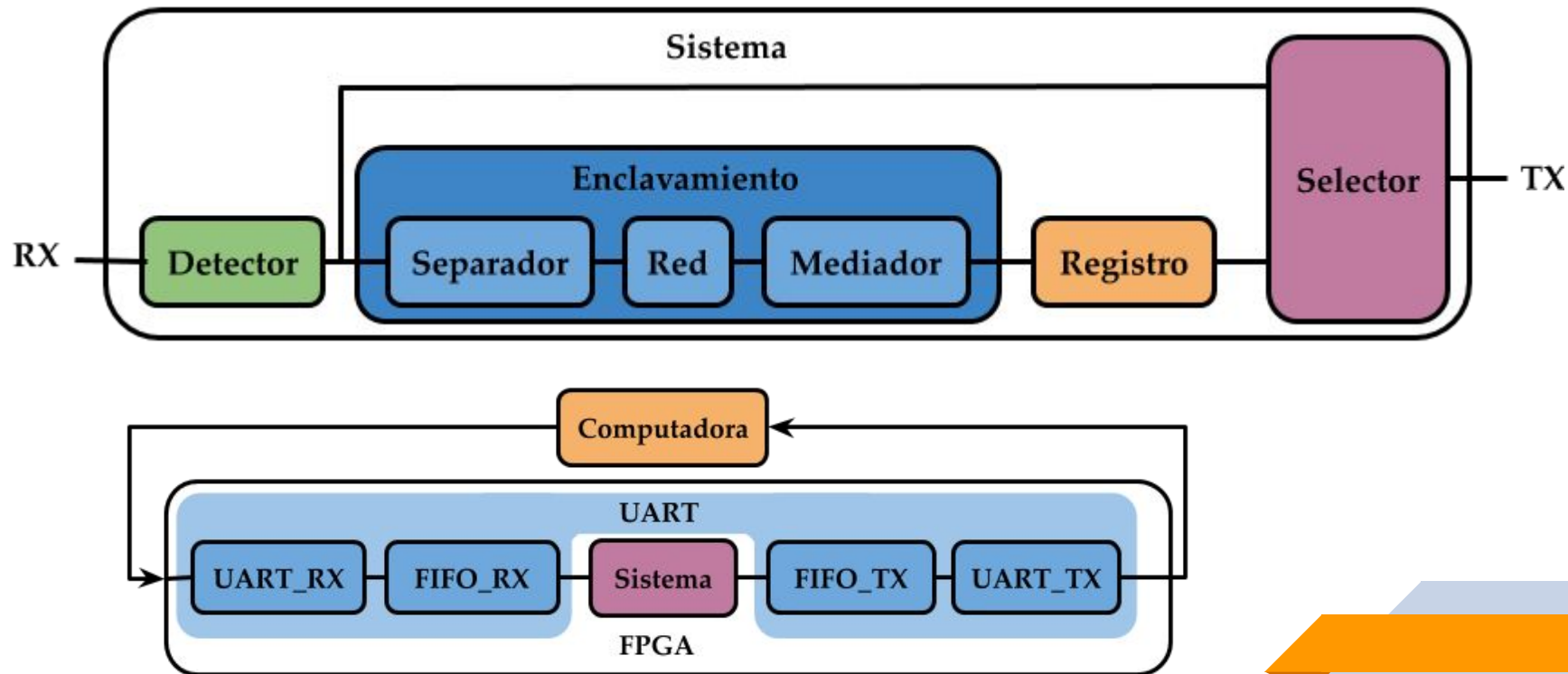
Diversidad → Mitigación de fallas de causa común



Enfoque geográfico - Arquitectura propuesta



Diseño propuesto en el marco de este trabajo.



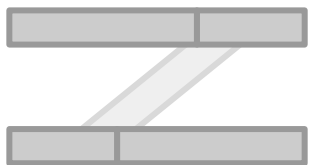
4

Conceptos accesorios

Funcionalidades del enclavamiento



Señales de la máquina de cambios



Comando: Normal

Indicación: Normal

Correspondencia: Normal



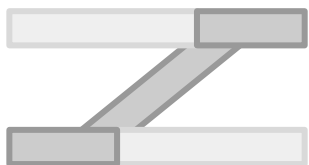
Lo que pido al sistema



Lo que leo del campo



Estado asumido



Comando: Normal

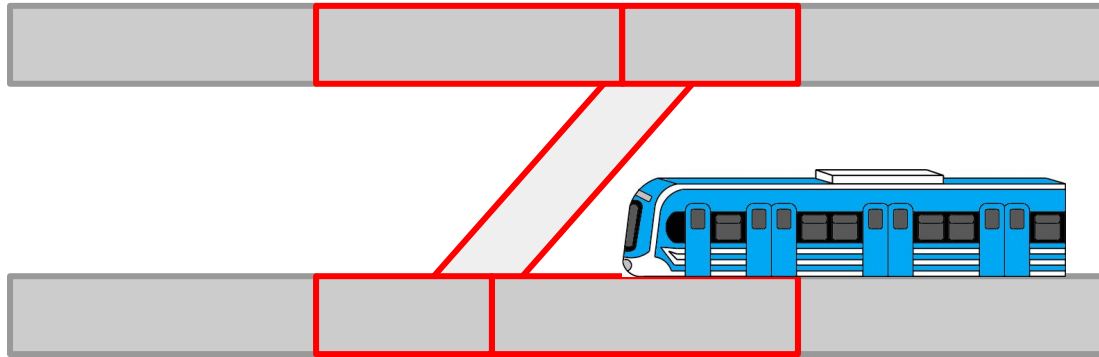
Indicación: Reverso

Correspondencia: Ninguna



Bloqueo de máquina de cambios por ocupación

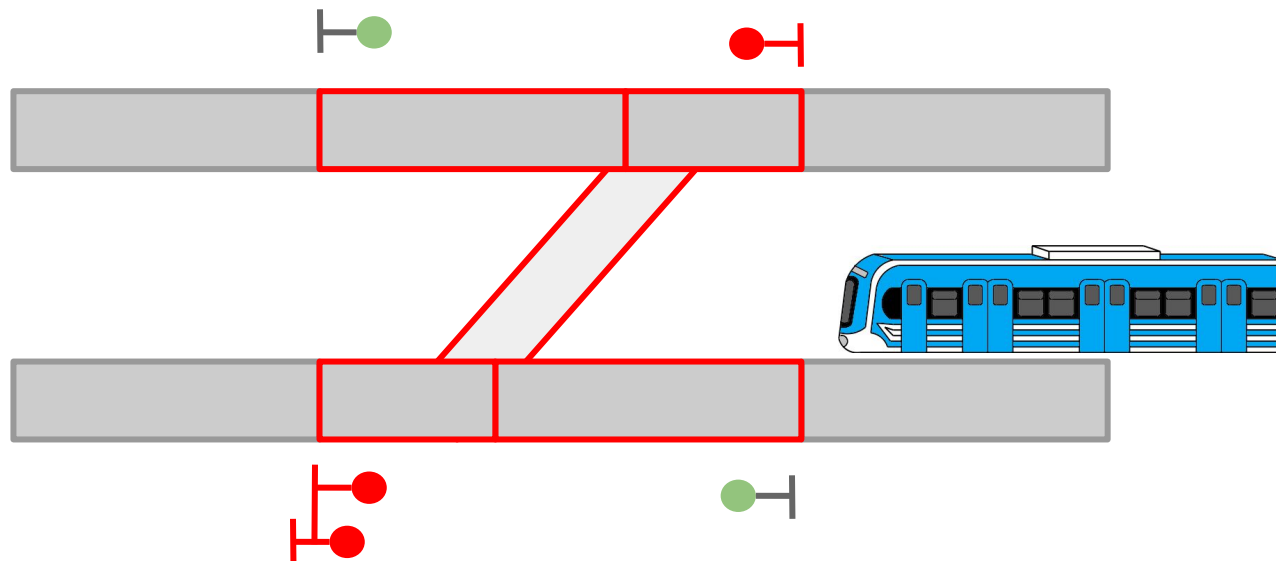
Inhibe la conmutación del **comando** si los sectores de detección que protegen la cambio se encuentran **ocupados**





Requerimiento de rutas y bloqueo de cambios en ruta

Inhibe la conmutación del **comando** de los cambios involucrados y protege las **señales conflictivas**.

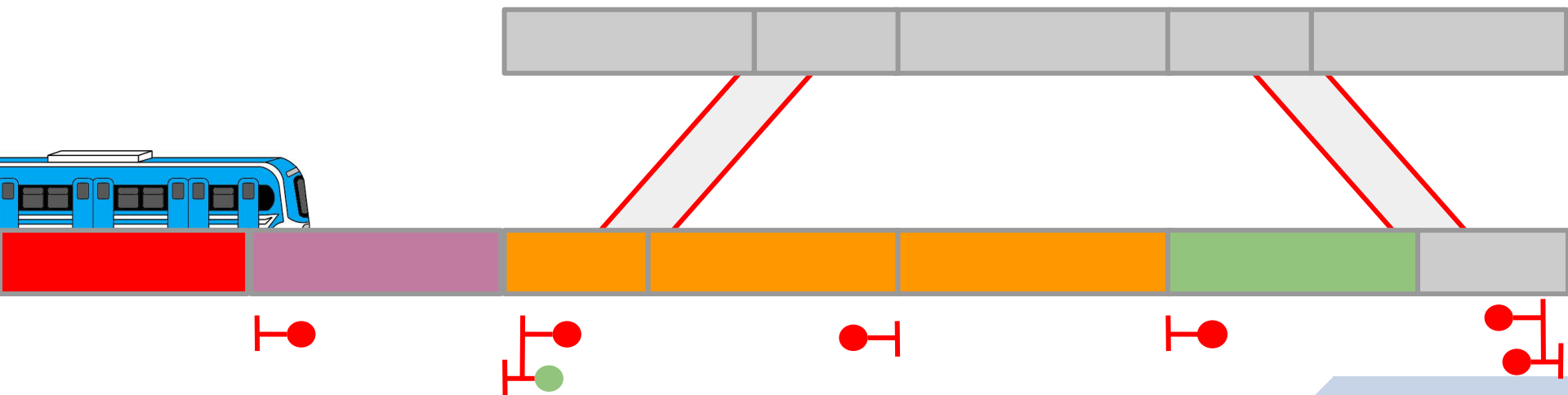


Rutas: Pedidas o canceladas desde la mesa de mando



Protección por aproximación

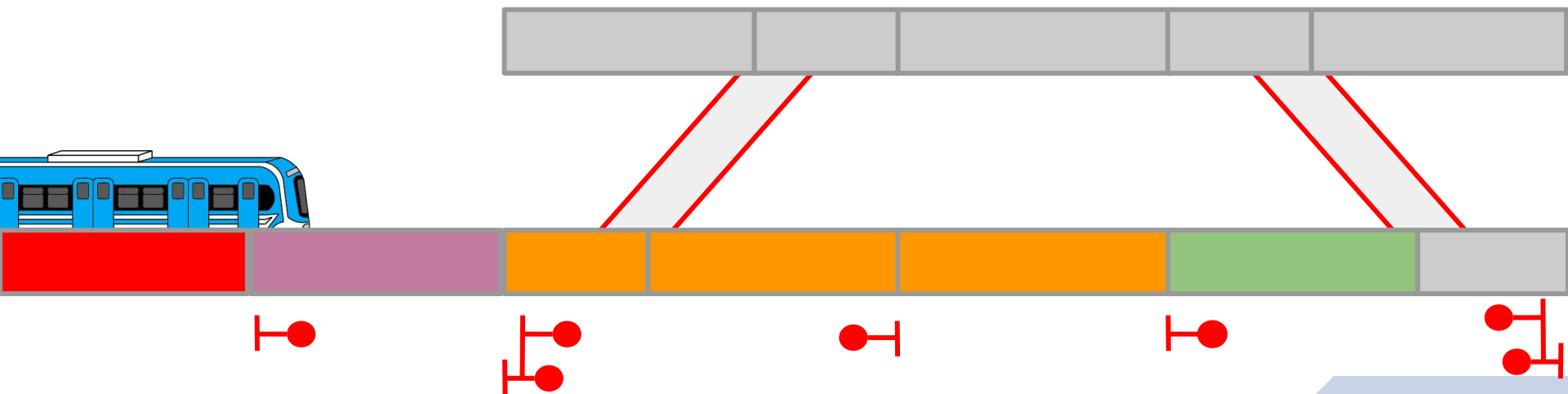
Si el **sector de aproximación** está ocupado y la **ruta establecida** es cancelada sin tiempo a frenar:





Protección por aproximación

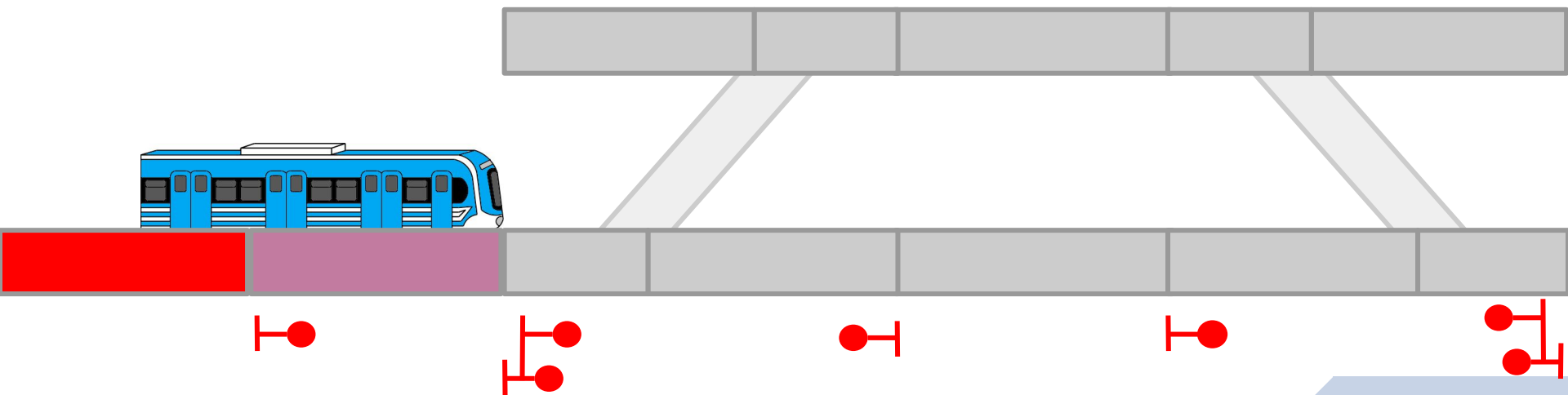
La señal pasa a **peligro**, pero la **ruta** se mantiene protegida durante un tiempo de seguridad.





Protección por aproximación

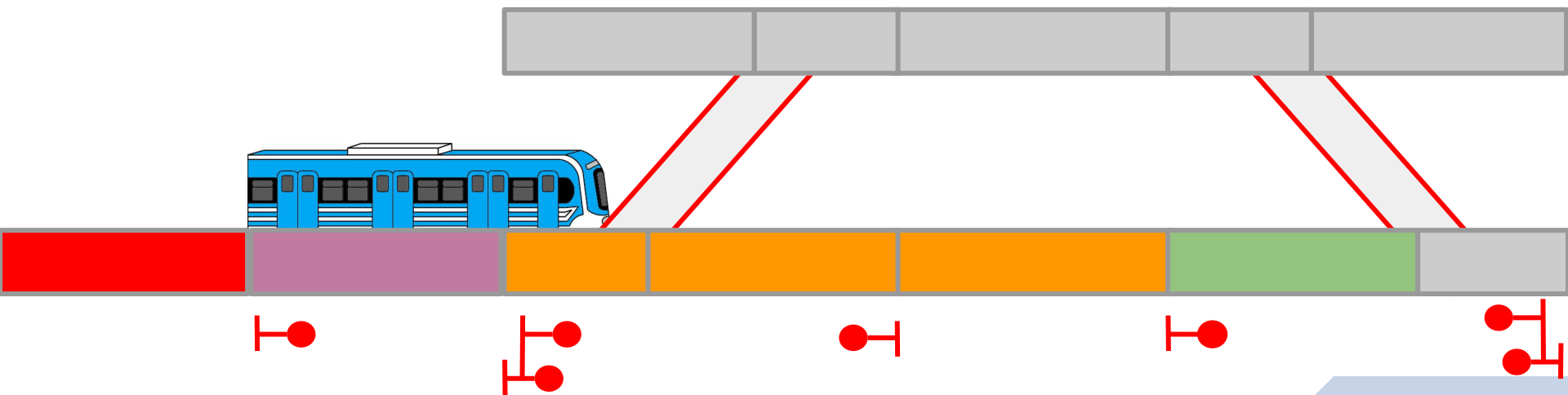
Si el tren se detiene en el **sector**, cuando transcurre el tiempo de protección se libera la ruta.





Protección por aproximación

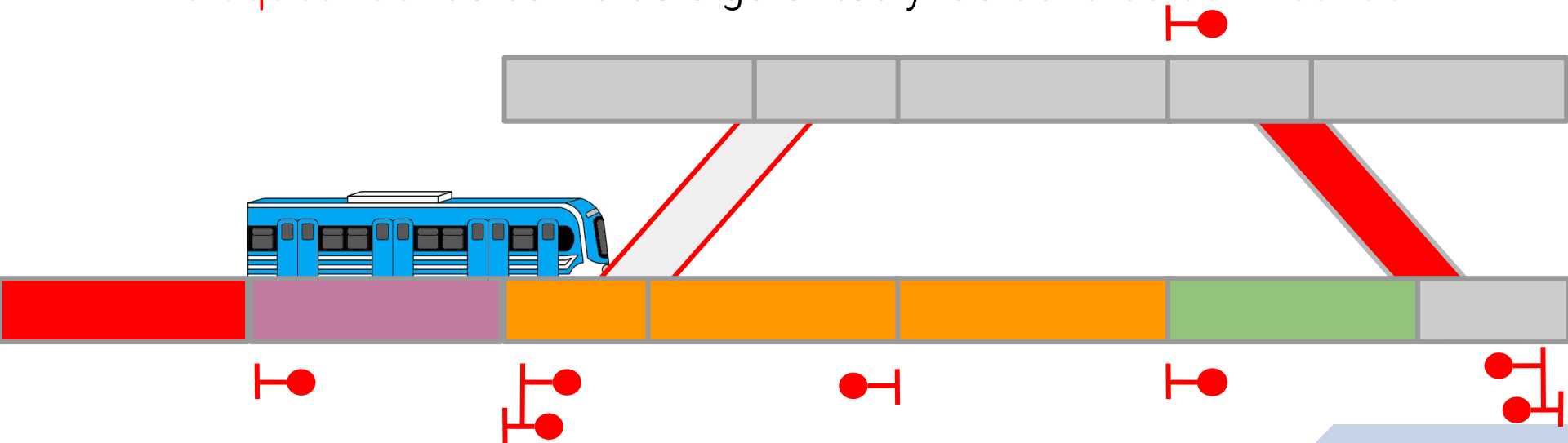
Pero si paso la **señal**, la **ruta** se mantiene protegida hasta que termine de circular la formación.





Protección por solape

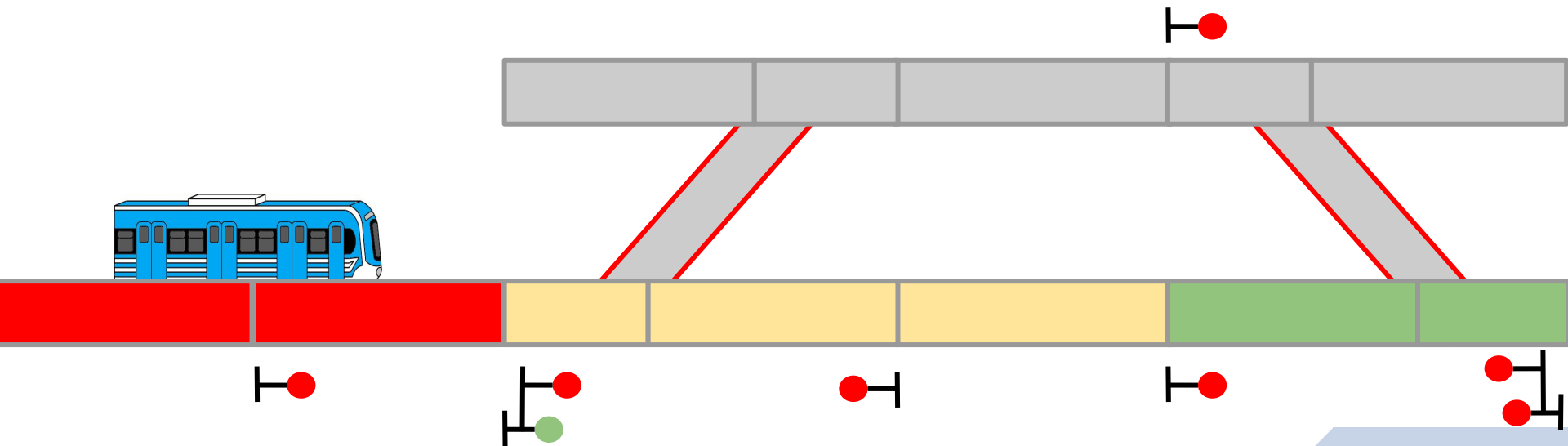
Protegen al tren de la eventualidad de no frenar en la señal de destino,
bloqueando los cambios siguientes y las **señales conflictivas**





Liberación secuencial

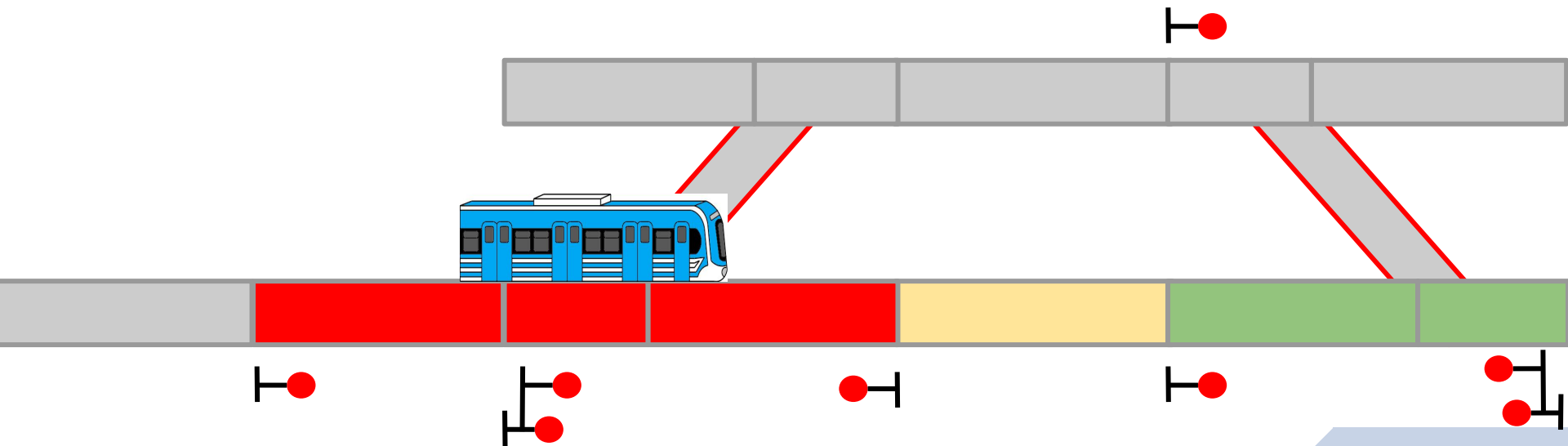
Al circular el tren, los cambios se van liberando tras su paso





Liberación secuencial

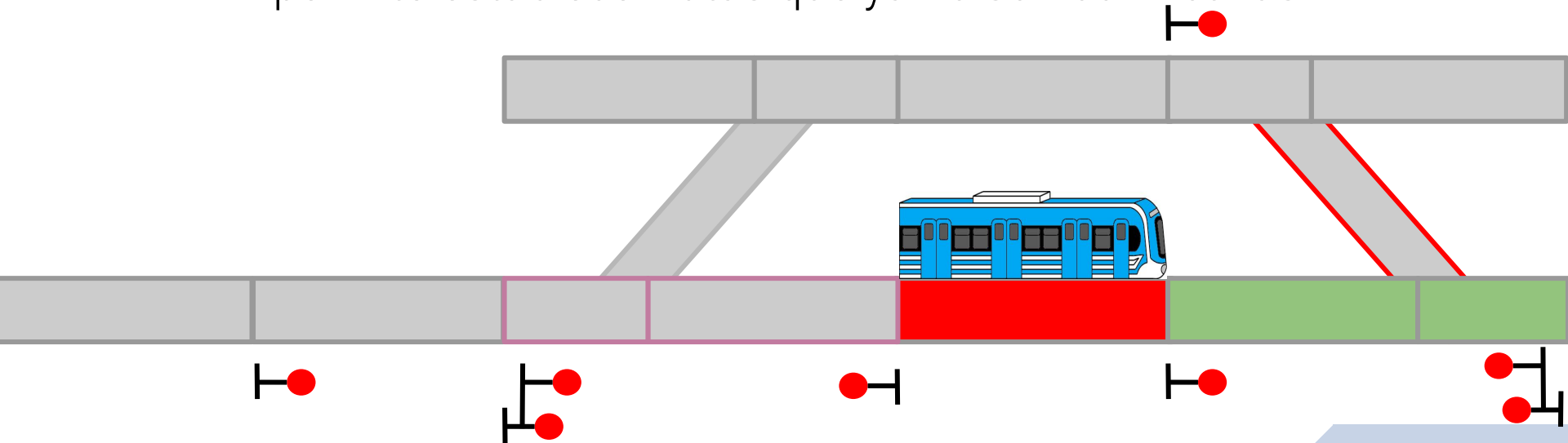
Al circular el tren, los cambios se van liberando tras su paso





Liberación secuencial

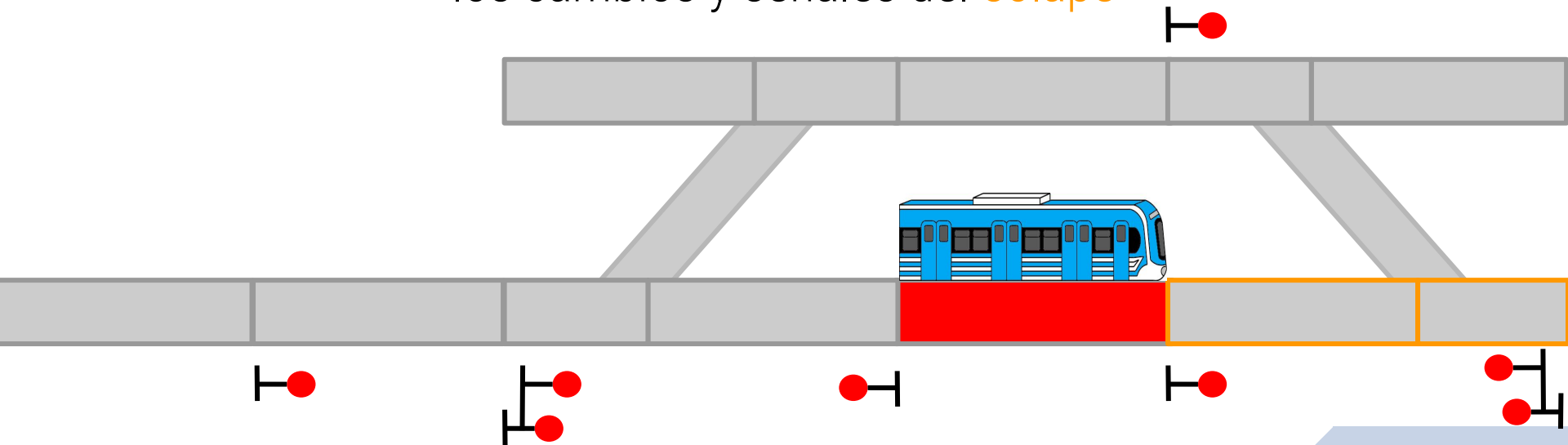
Al liberar los **sectores de protección**, los cambios se desbloquean y permite establecer rutas que ya no son conflictivas.





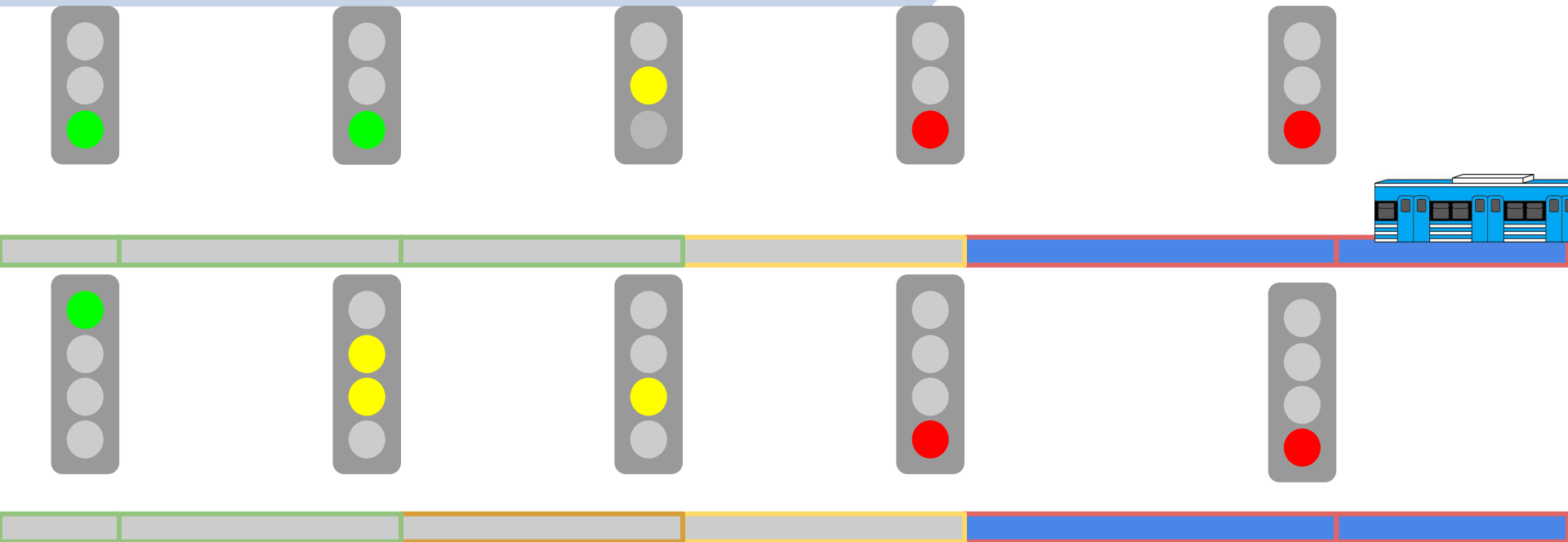
Liberación secuencial

Pasado un tiempo de seguridad, se asume que el tren se detuvo y se liberan los cambios y señales del **solape**





Modelado de semáforos



5

Plataforma a utilizar

FPGA

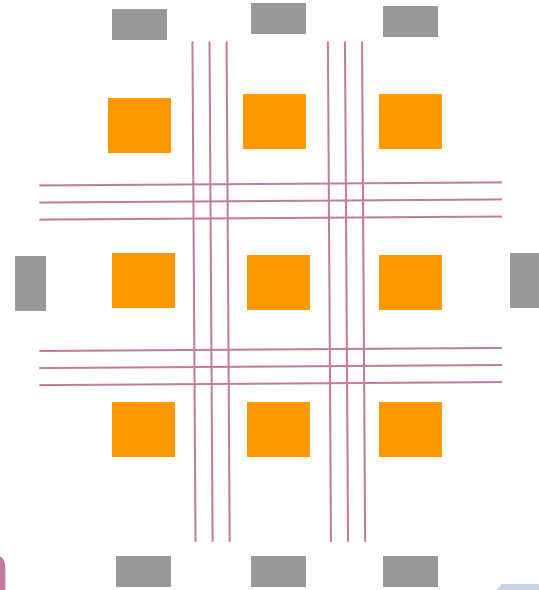


¿Que es una FPGA? (*Field-Programmable Gate Array*)

- *Matriz de compuertas programables*
- Dispositivo programable
- Conformado por **bloques lógicos**
- **Configurable** mediante lenguajes de descripción de hardware

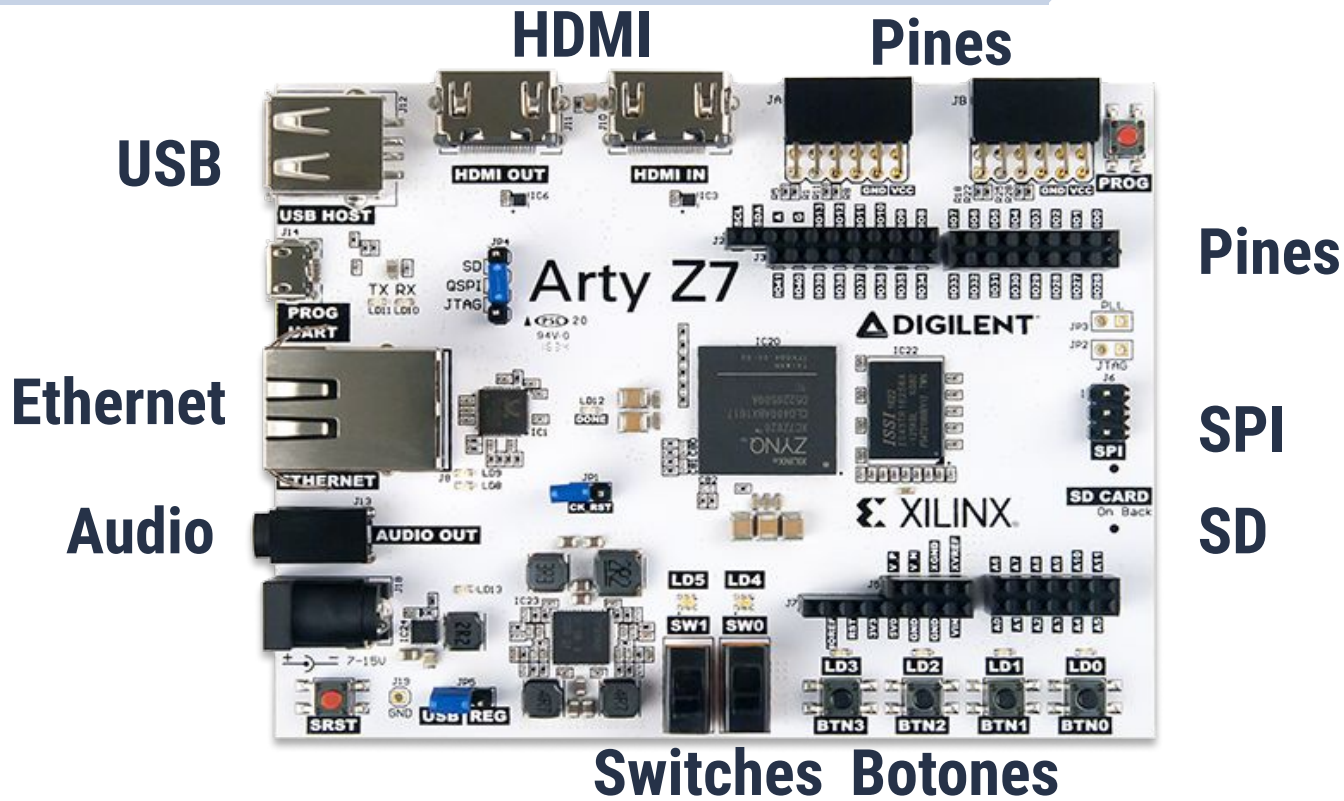
Canales de interconexión

Bloques de I/O





Arty Z7 10



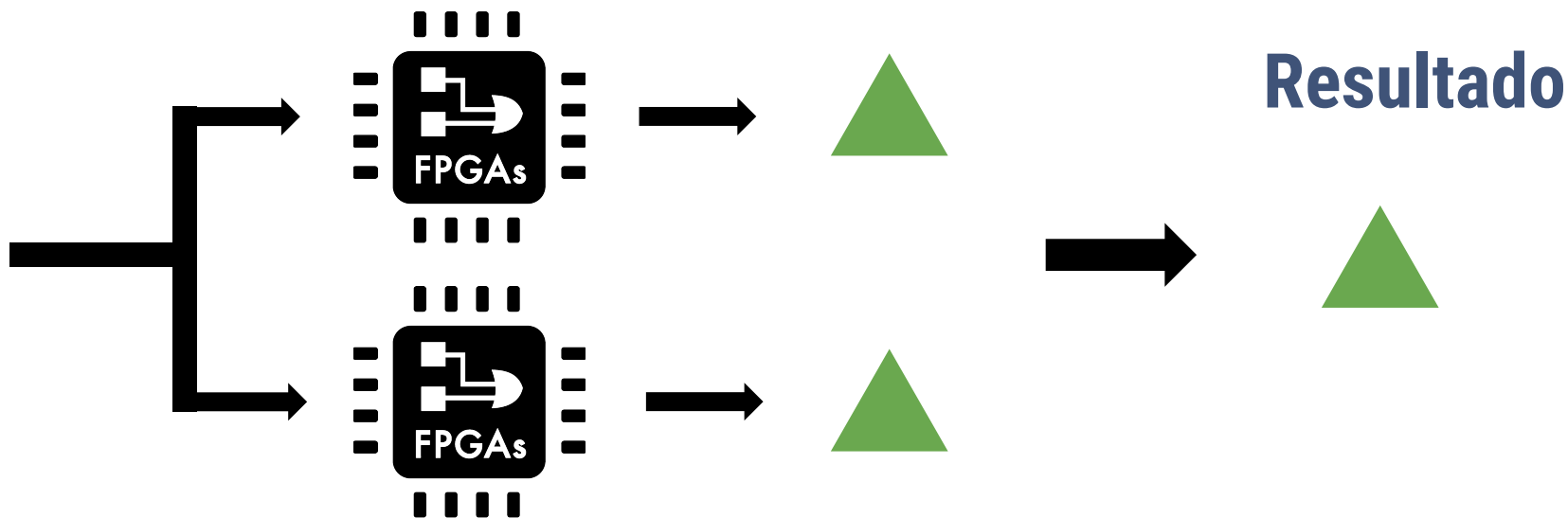
6

Estrategias a utilizar

Redundancia y diversidad

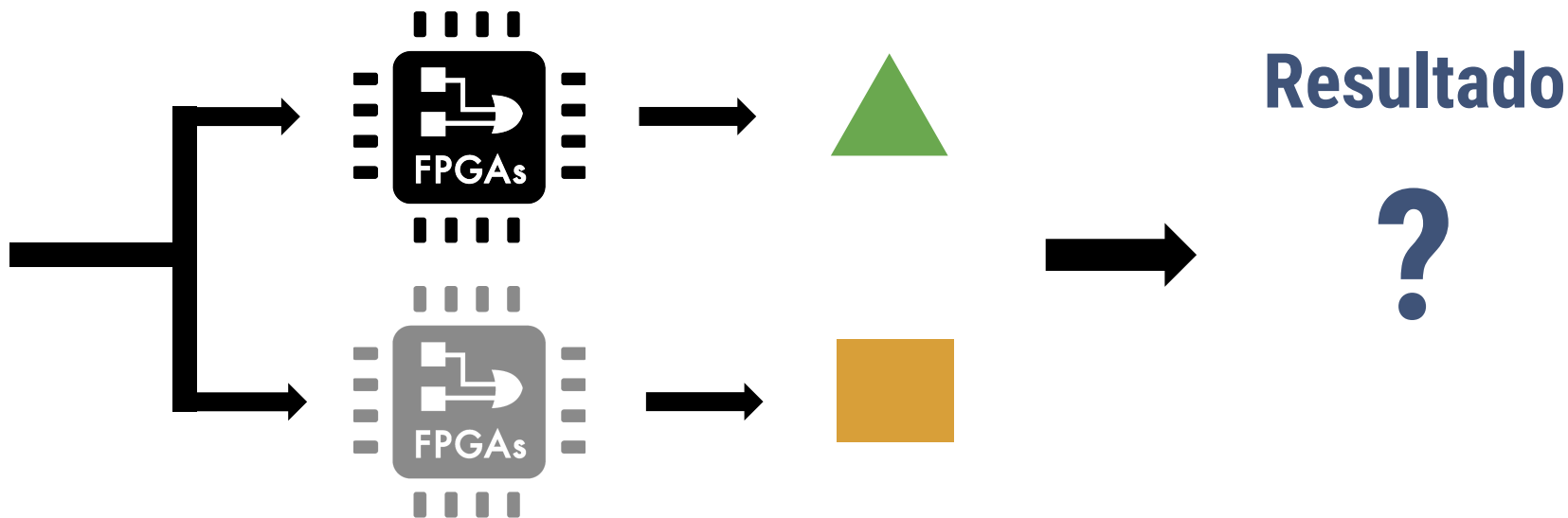


Redundancia (2oo2 -> coincidencia)



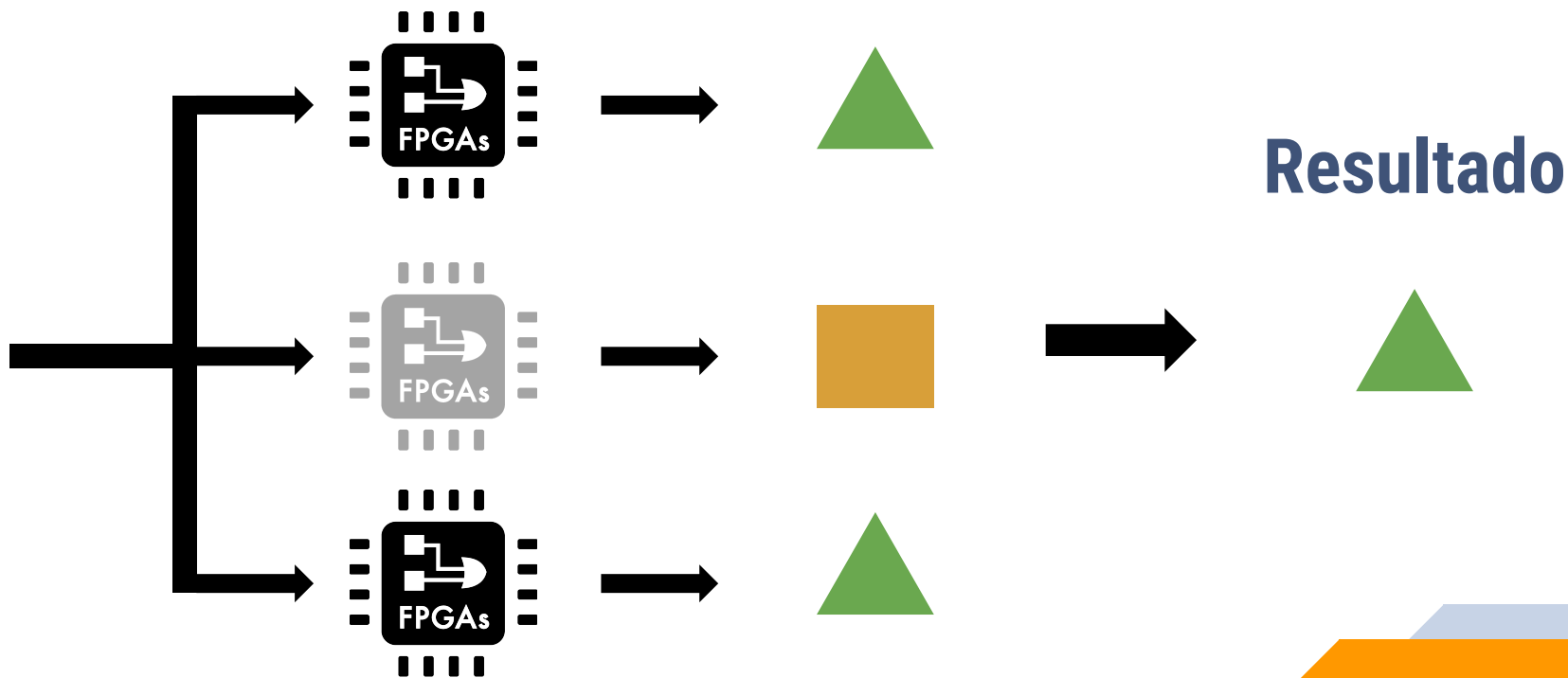


Redundancia (2oo2 -> discrepancia)



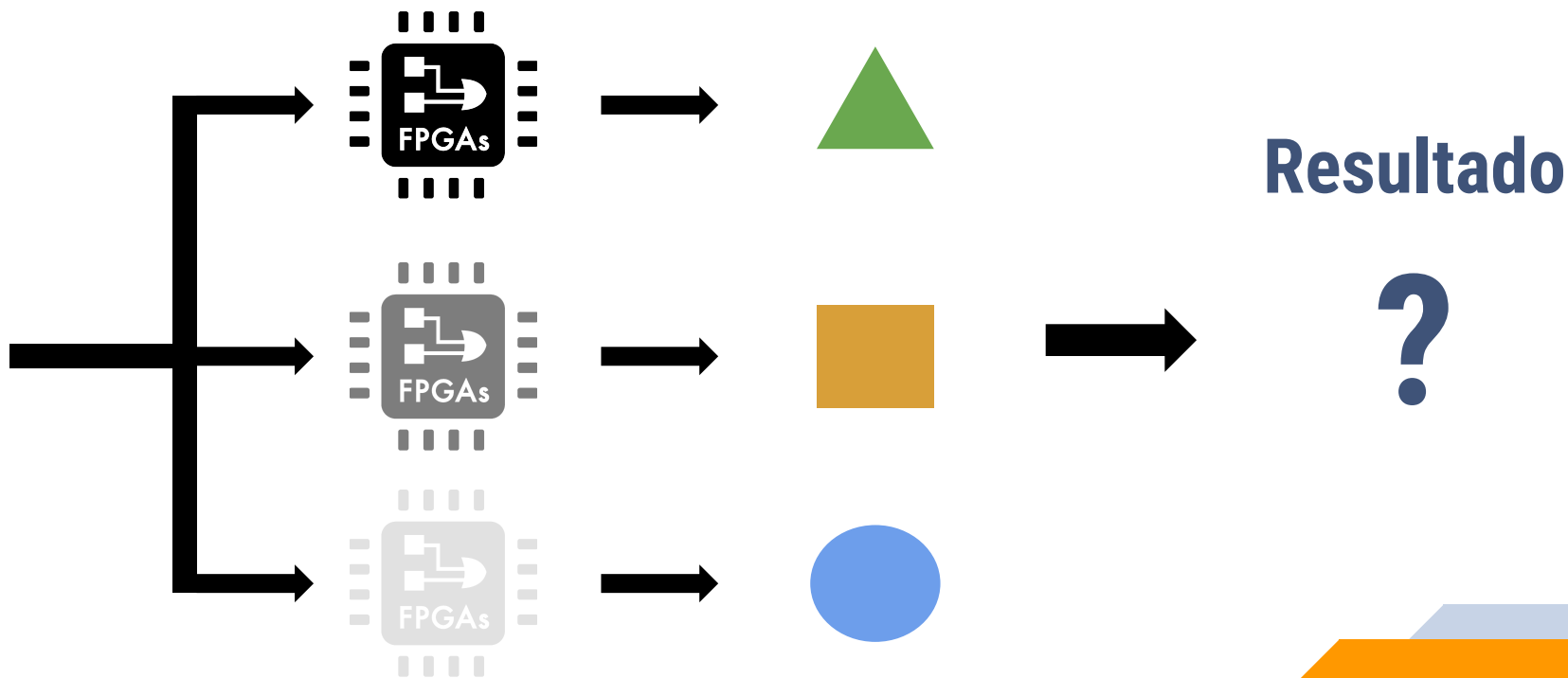


Redundancia (2oo3 -> votación)



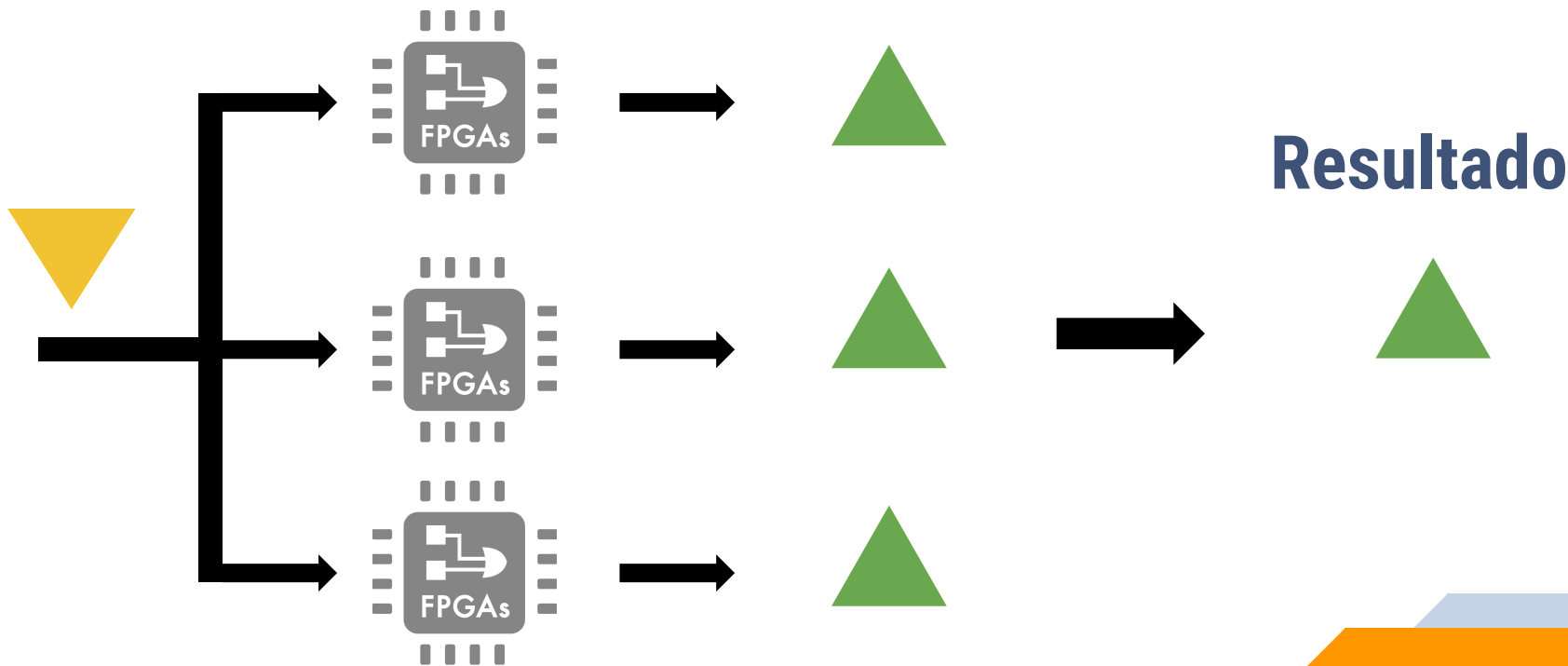


Redundancia (2oo3 -> discrepancia)



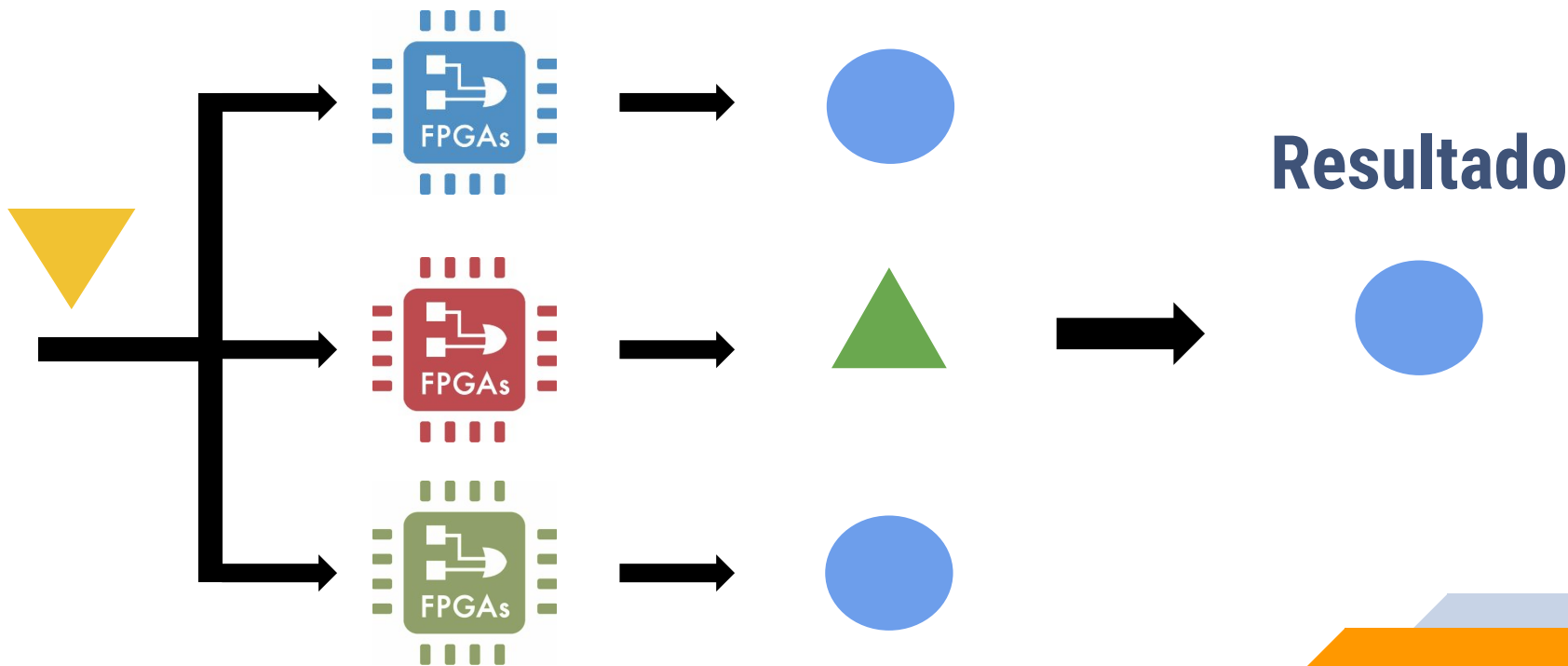


Diversidad (FPGAs del mismo fabricante)



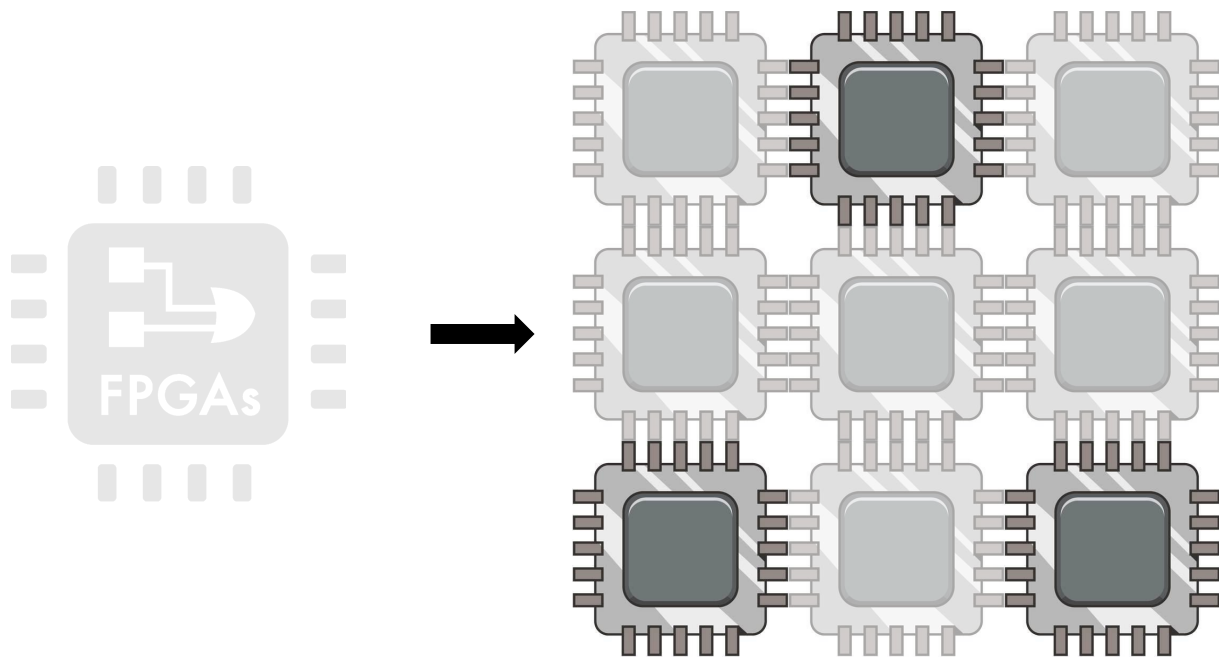


Diversidad (FPGAs diferente fabricante)





Redundancia (En diferentes partes de la FPGAs)



7

Estrategias de redundancia

Qué hacer con cada elemento



Single Event Upset (SEUs) Single Event Transients (SETs)

- SEUs : Alteran el **estado lógico** de las memorias estáticas.
- SETs : Causan **pulsos transitorios** en las rutas lógicas combinacionales.
- La funcionalidad de la FPGA depende de los datos.
- Triple Modular Redundancy (TMR)

SEUs **→** Que no se **acumulen**.

SETs **→** Que no se **propaguen**.





Throughput Logic

Módulos donde entrada y salida se encuentran conectadas **sin lazos**



No dependen de **estados anteriores**.



Por ejemplo : Sumadores

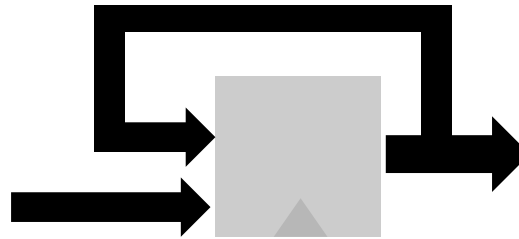


State-Machine Logic

Módulos donde la salida depende de la entrada y el **estado anterior**.



Si un estado es incorrecto, los próximos lo serán también



Por ejemplo : Acumuladores, contadores



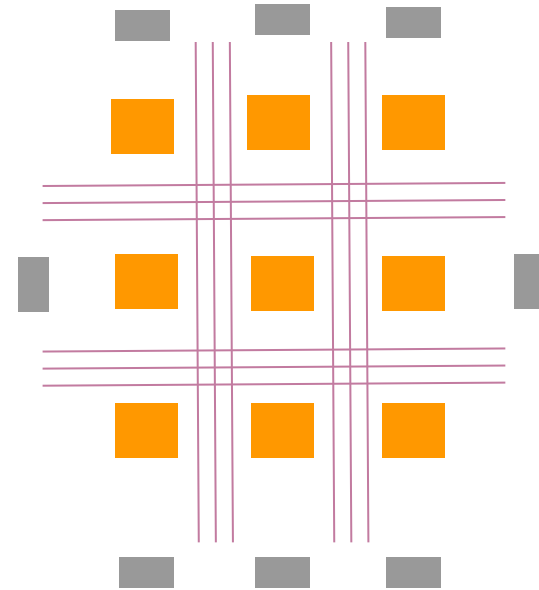
Input/Output Logic

Entradas y salidas de la FPGA misma



Finitas y sometidas a **ruido eléctrico** y **radiación**.

Bloques de I/O



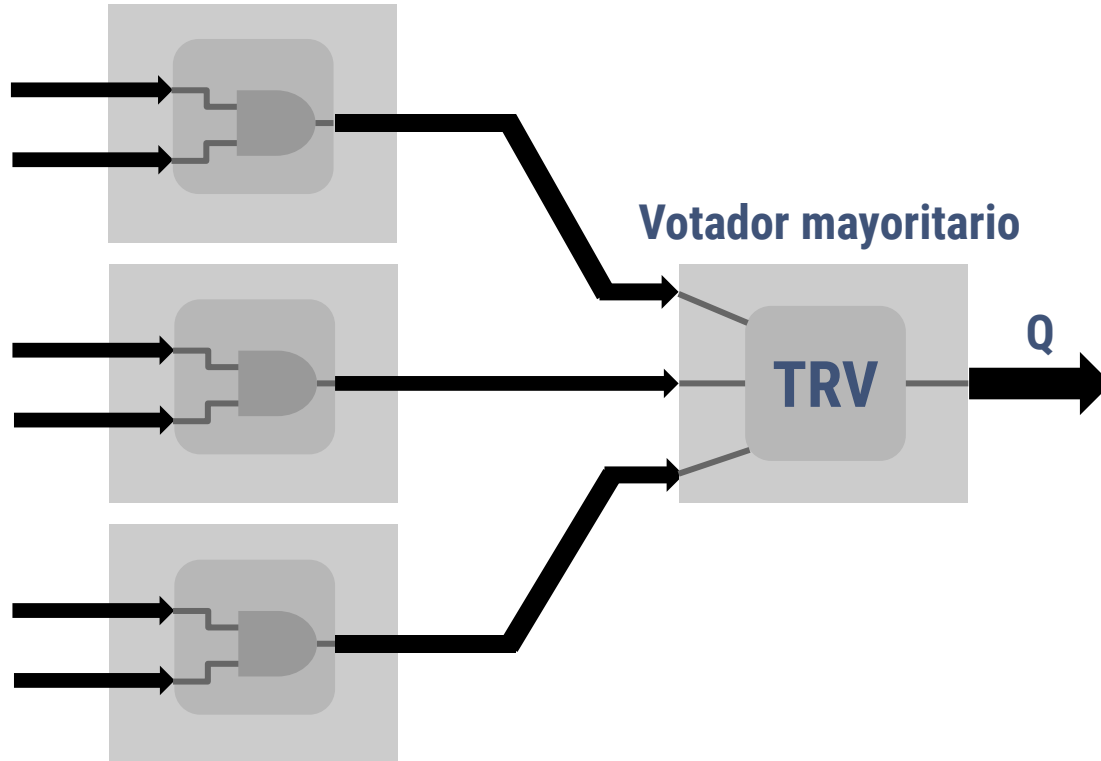


Special features (Block RAM, LUT, Shift-registers, DLLs)

- Block RAM : Guarda **gran cantidad de datos** de la FPGA.
- Look-Up Table (LUTs) : **Tablas de verdad** de la FPGA.
- Shift-Registers : **Registros** de desplazamiento.
- Delay Locked Loop (DLLs) : Usado para sincronismos y **clocks altos**.

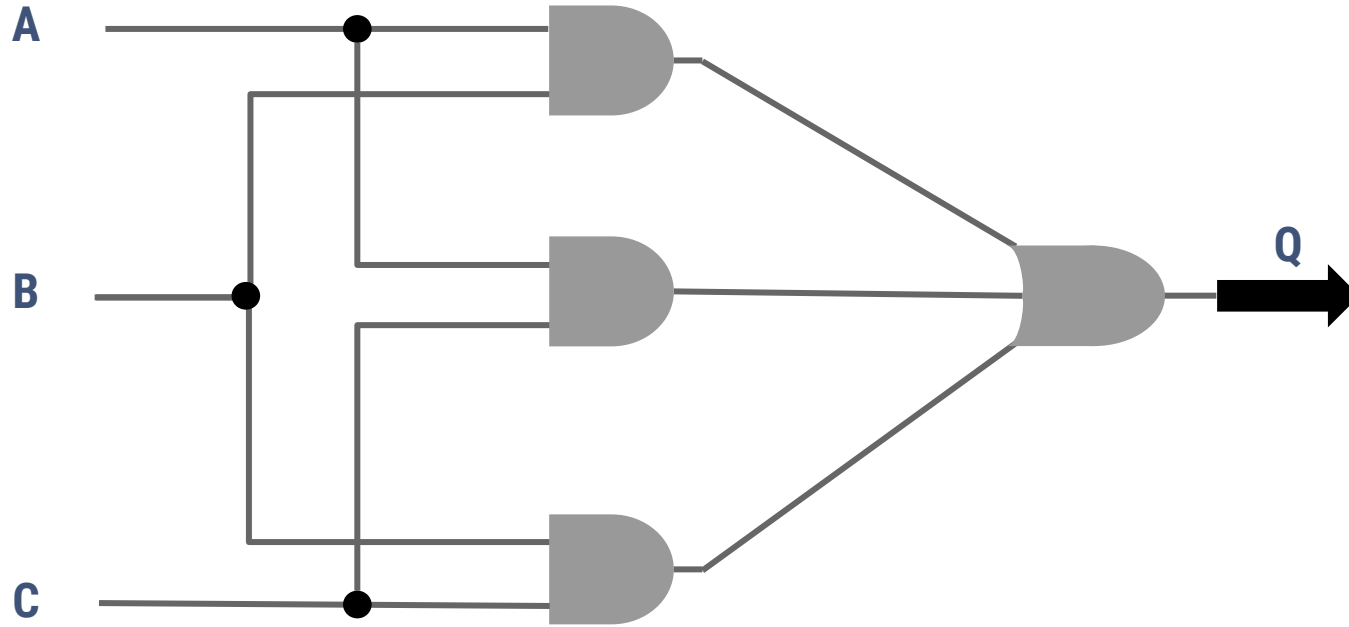


Triple Redundant Voter (TRV)





Votador mayoritario en LUT



A	B	C	Salida
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Si las LUTs no te sobran puedes implementarlos con 3-State Buffers (BUFT)

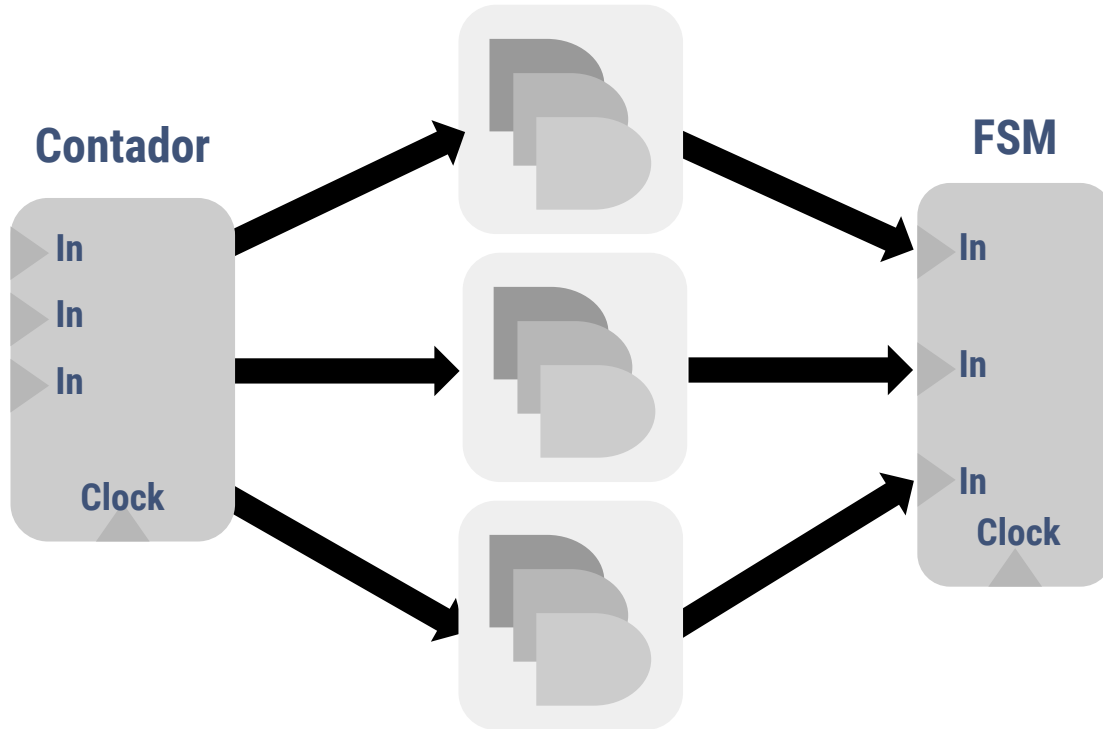


TMR for Throughput Logic Structures



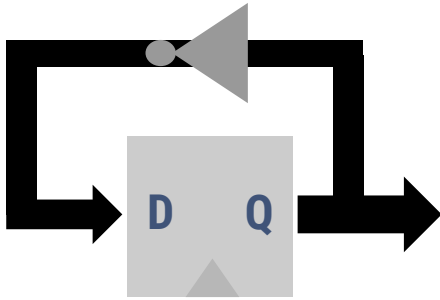


TMR for Throughput Logic Structures





TMR for simple State-Machines



Esperado



Obtengo



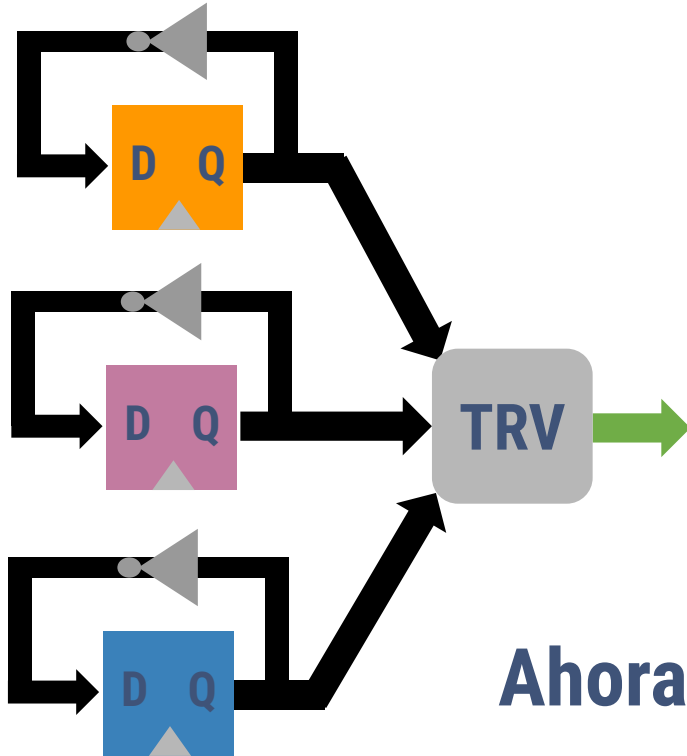
SEU



TMR for simple State-Machines

$P_i = 0.01\%$

$P_T = 0.02\%$



Q1



Q2



Q3



Obtengo



Esperado



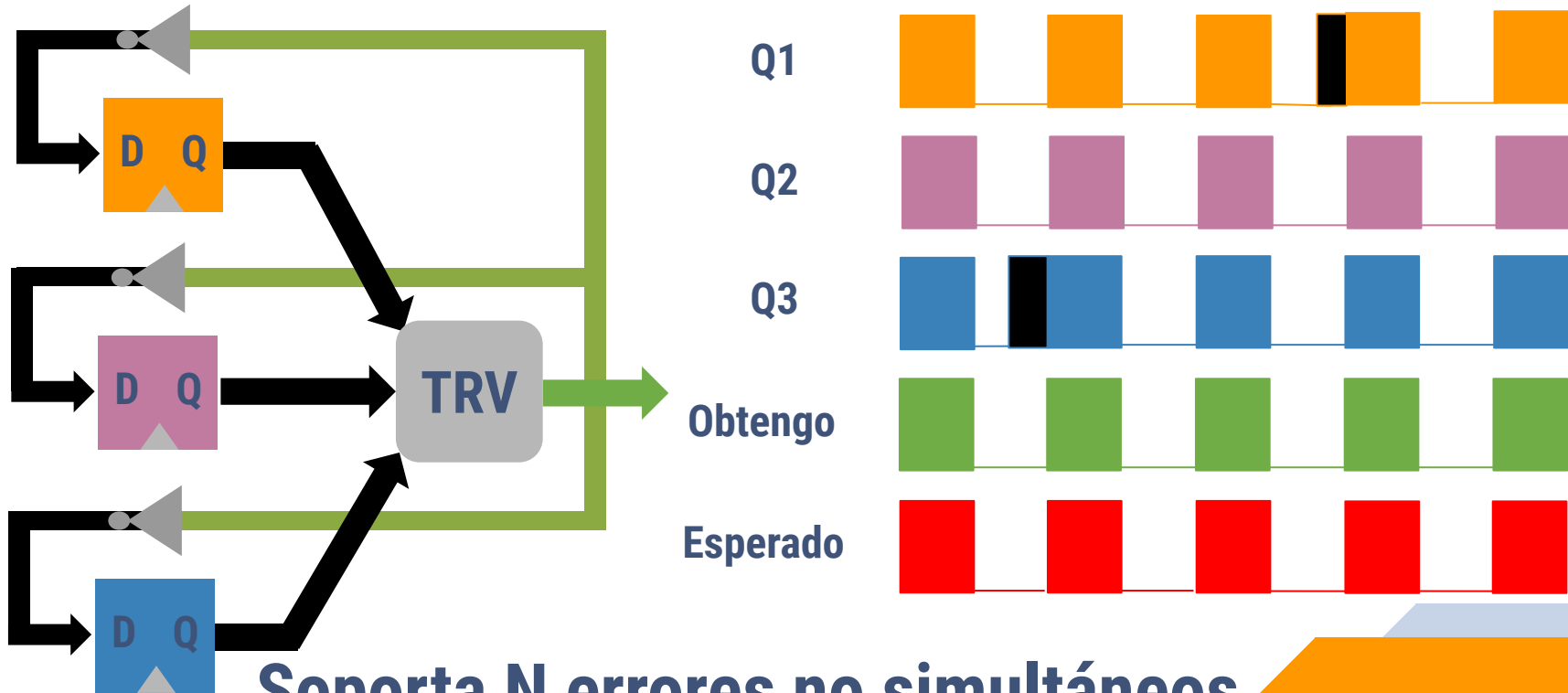
Ahora soporta hasta UN
error.



TMR for simple State-Machines

$$P_i = 0.01\%$$

$$P_T = P_i^2 = 0.0001\%$$



Soporta N errores no simultáneos.

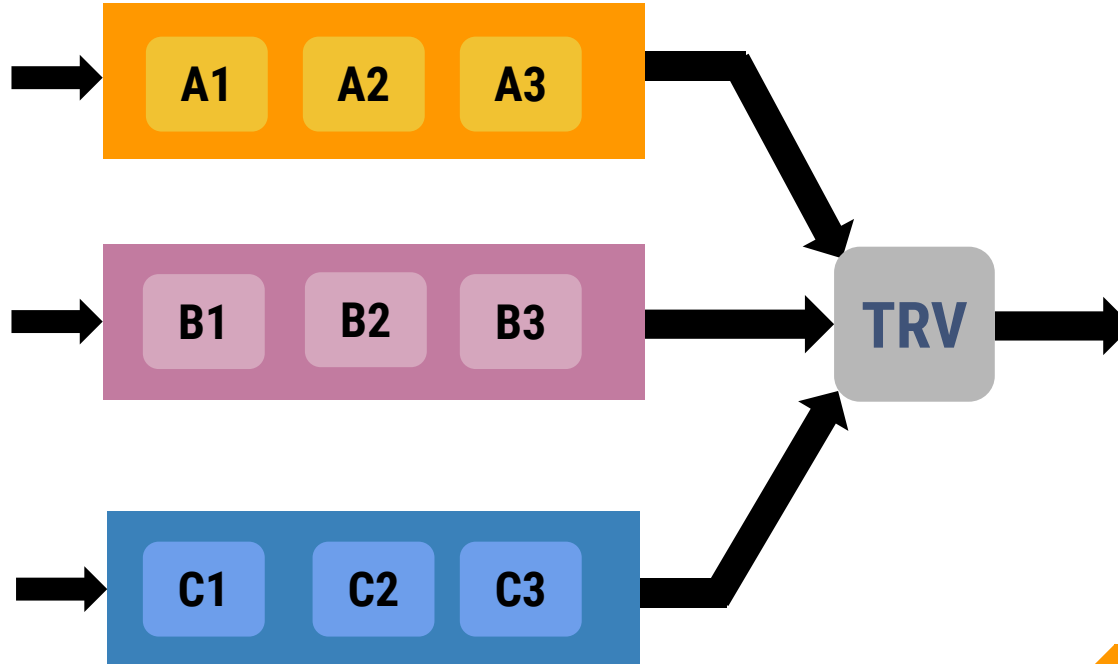


TMR for simple State-Machines

$$P_{i_sub} = 0.01\%$$

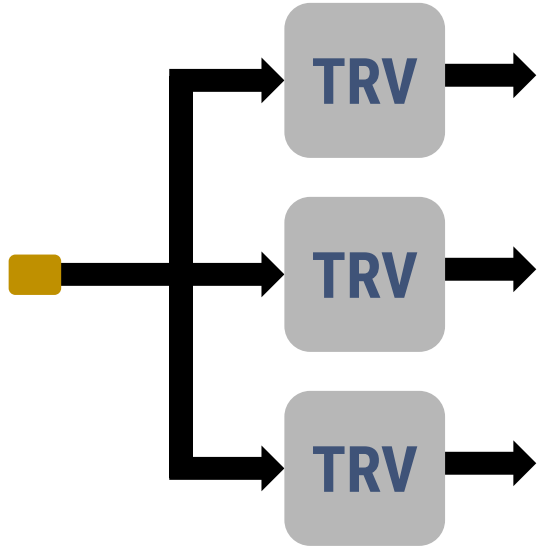
$$P_i = 0.0001\%$$

$$P_T = P_i^4 = 0.00000001\%$$

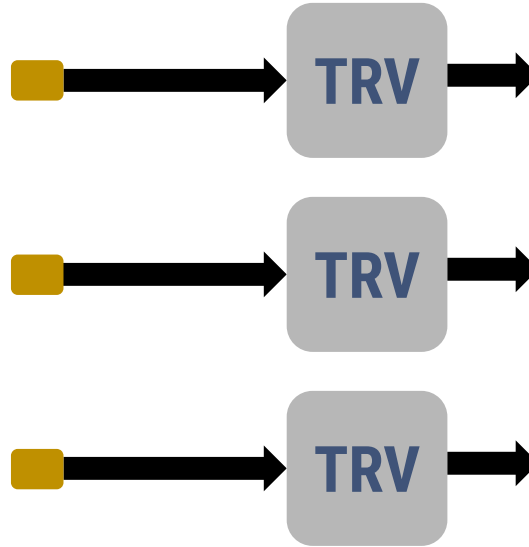




TMR for Input Logic



Leer el mismo Pin



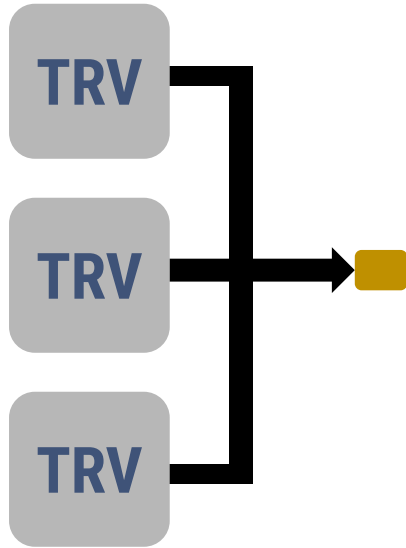
Leer el mismo dato
de diferentes Pines



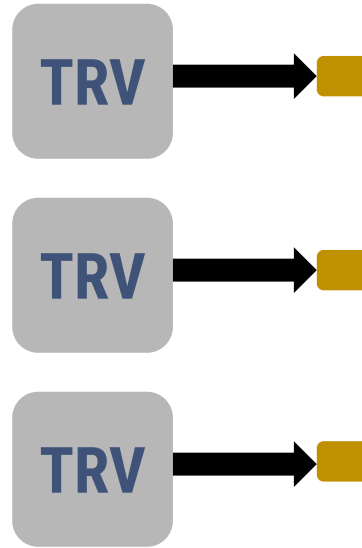
**¡Necesito el
TRIPLE de pines
de entrada!**



TMR for Output Logic



Escribir el mismo Pin



**Escribir el mismo dato
en diferentes Pines**



**¡Necesito el
TRIPLE de pines
de salida!**



TMR for Input/Output Block (IOB) Logic

¿A quien le sobran el **triple de pines de los requeridos?**



TMR for Special features

Block RAM y Clock Management



8

¿Mucho? ¿Poco? ¿Nada?

Medida de cuánto redundar y donde



Y Dios le pidió a Noé el desarrollador ...

¡Poneme 3 de cada especie!

X3 Máquinas de estado

X3 Pines de entrada

X3 Pines de salida

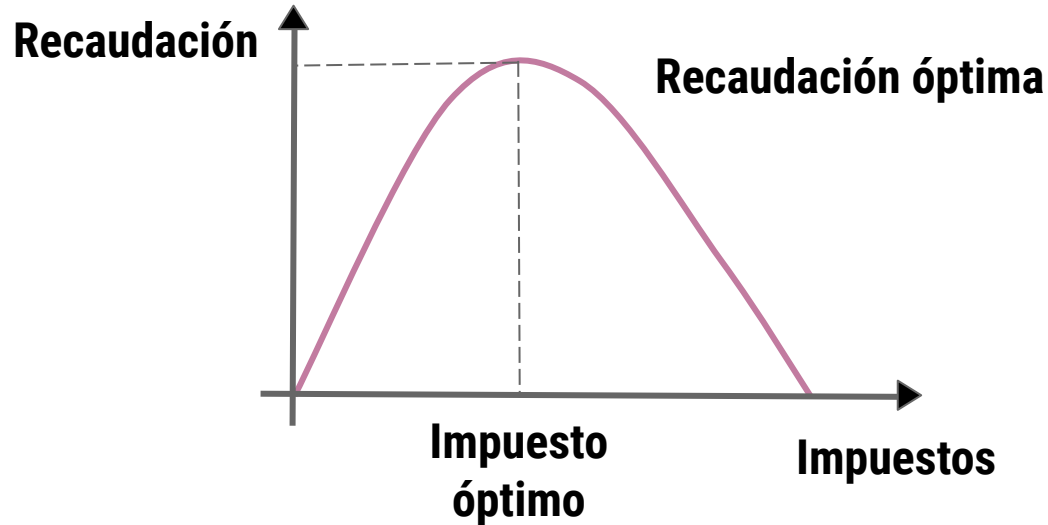
X3 Almacenamiento

X3 Procesamiento



¿Todo, un poquito o nada?

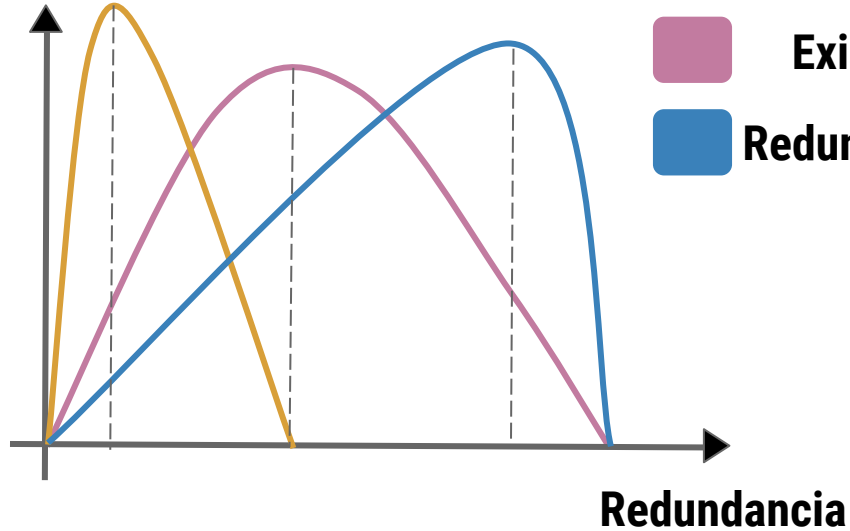
Curva de Laffer





¿Todo, un poquito o nada?

Disponibilidad /
Seguridad



- La redundancia es para los débiles
- Existe una estrategia intermedia
- Redundamentalistas fundamentalistas



Scrubbing

- Si un SEU ocurre en una LUT redundada, su efecto se mantiene hasta el **próximo ciclo**.
- La constante reconfiguración **evita la acumulación** de SEUs.
- A esta continua carga de datos se denomina **scrubbing** (Depuración).





Análisis de los SEUs con el enfoque TMR

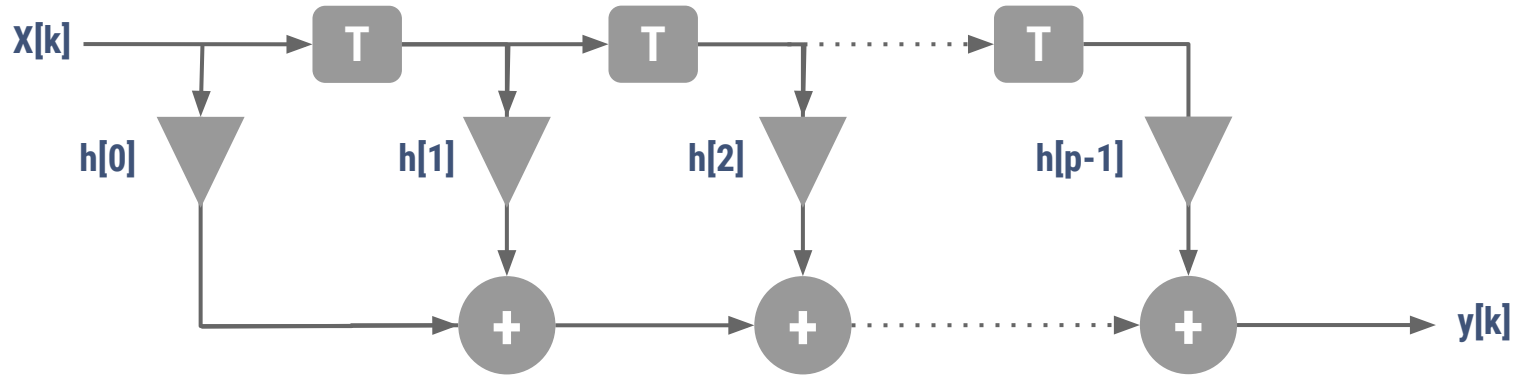
Localización	Perturbación	Consecuencia	Corrección
LUT	Modificación en la lógica	Error en una parte redundada, sin error en TMR.	Scrubbing
Ruteo	Conexión/desconexión entre dos o más señales del diseño	Error en una parte redundada, sin error en TMR. Error en más de un parte redundada, error en TMR.	Scrubbing
Lógica de conexiones	Conexión/desconexión entre dos o más señales del mismo CLB	Error en una parte redundada, sin error en TMR. Error en más de un parte redundada, error en TMR.	Scrubbing
Flip-Flops	Modificación en la lógica secuencial	Error en una parte redundada, sin error en TMR.	Modificar diseño



CLB : Complex Logic Blocks

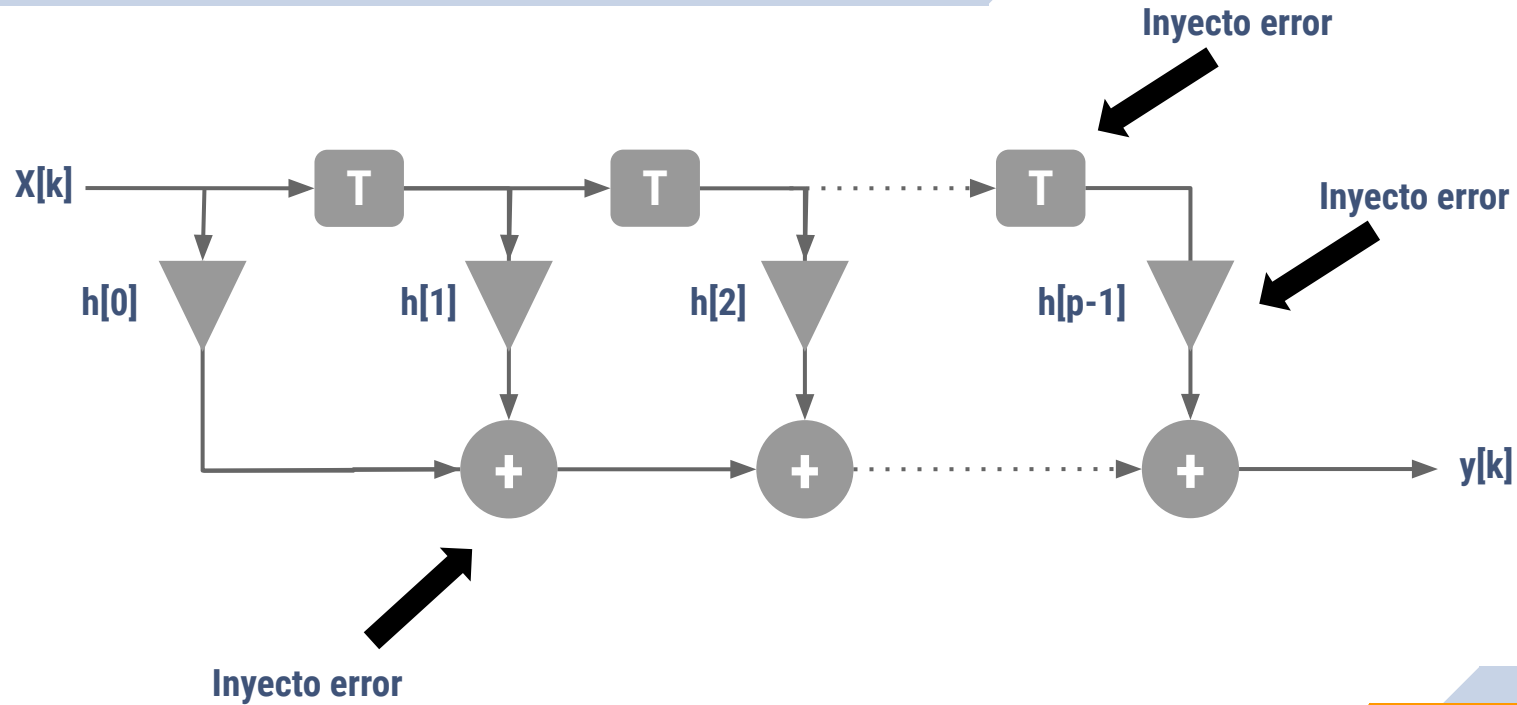


Filtro FIR



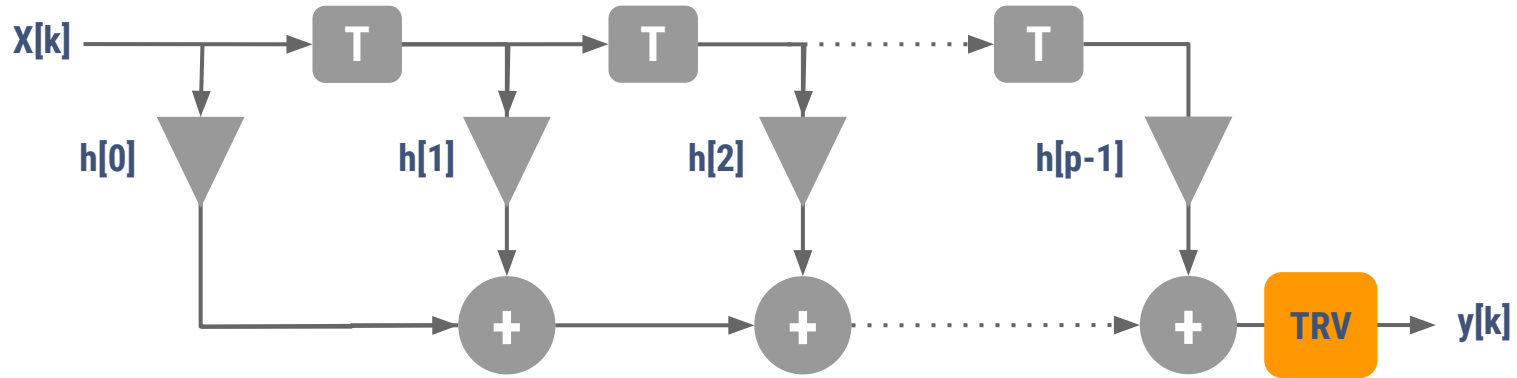


Fault injection experiments



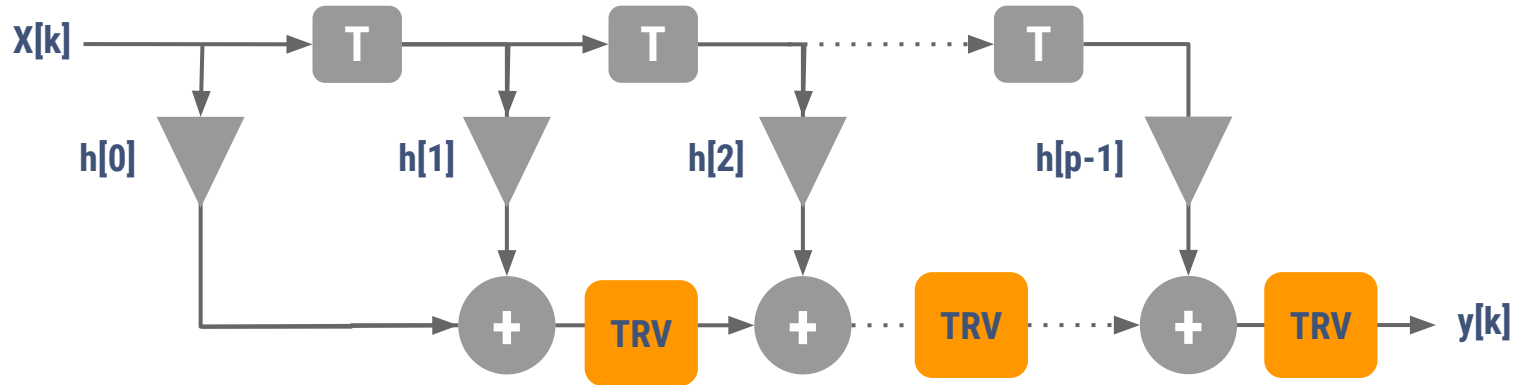


Filtro FIR con TMR (poco)



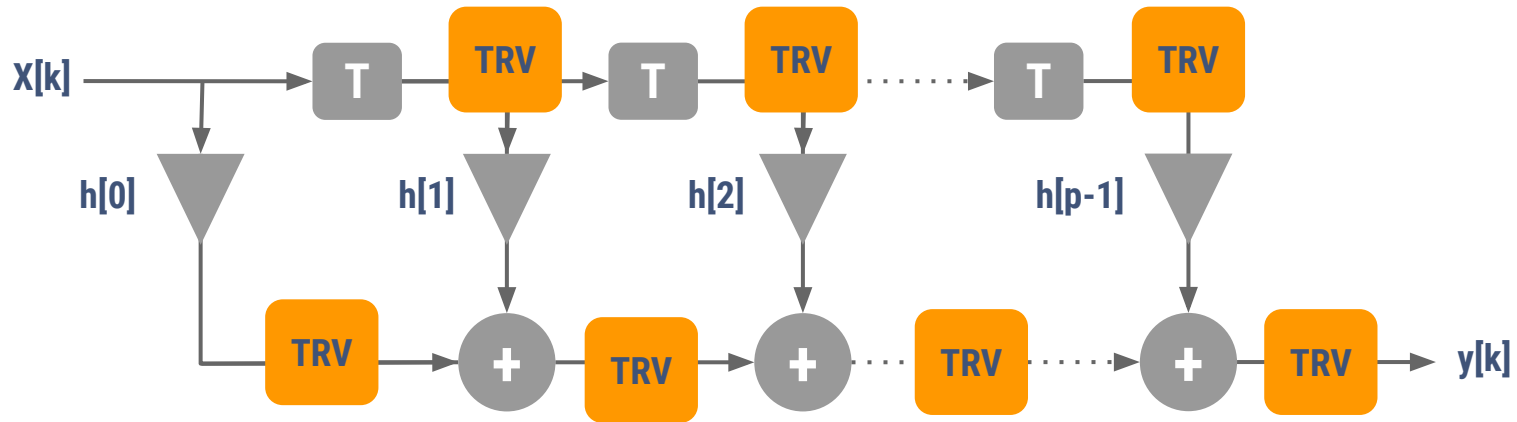


Filtro FIR con TMR (medio)





Filtro FIR con TMR (mucho)





Comparaciones

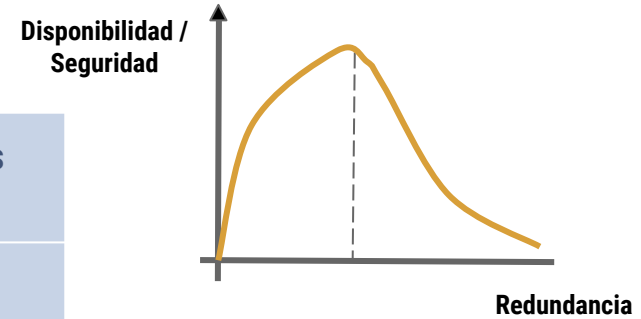
TMR	Area (#Slices)	#Routing bits	#LUTs bits	Performance
Nada	150	42.953	9600	154 MHz
Poco	X 3.7 498	X 3.2 151.994	X 3.7 31.872	153 MHz
Medio	504	161.568	32.256	137 MHz
Mucho	560	138.453	35.840	123 Mhz

- 21%



Comparaciones

TMR	# Errores inyectados	# Respuestas incorrectas	% Respuestas incorrectas
Nada	5.100	4.952	97.10 %
Poco	18.501	289	1.56 %
Medio	19.401	190	0.98 %
Mucho	17.515	706	4.03 %



← **¡El mejor!**

Hay que ser moderado al redundar, nunca demasiado

9

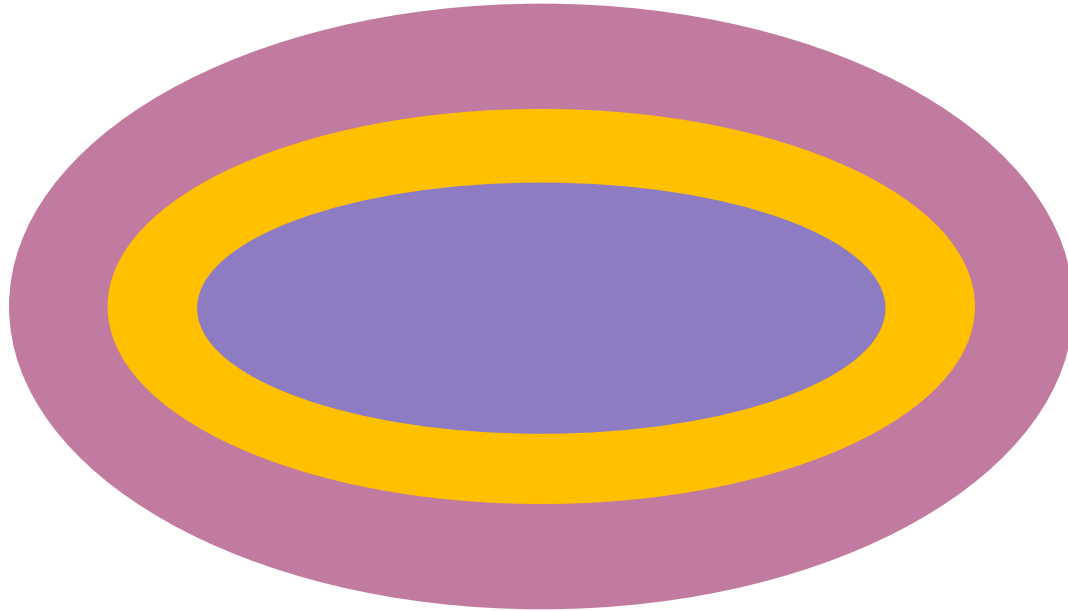
Implementación

VHDL



VHDL

VHSIC (*Very High Speed Integrated Circuit*) + HDL (*Hardware Description Language*).



Lo que quiero hacer



Compilable (lo que puedo)



Sintetizable (lo posible)



VHDL (código de ejemplo para biestable)

```
entity BIEST is  
  port(RST: in std_logic;  
        CLK: in std_logic;  
        D:  in std_logic;  
        Q:  out std_logic);  
end BIEST;
```

- Reset asincrono
- Reloj
- Dato de entrada
- Salida (dato guardado en el biestable)



Entidad: I/O y tipos

```
architecture D of BIEST is  
begin
```

```
  biest_D : process(RST, CLK)  
  begin  
    if RST = '1' then  
      Q <= '0';  
    elsif CLK'event and CLK = '1' then  
      Q <= D;  
    end if;  
  end process;  
end D;
```

- Reset asincrono
- Condicion de reloj activo por flanco de subida



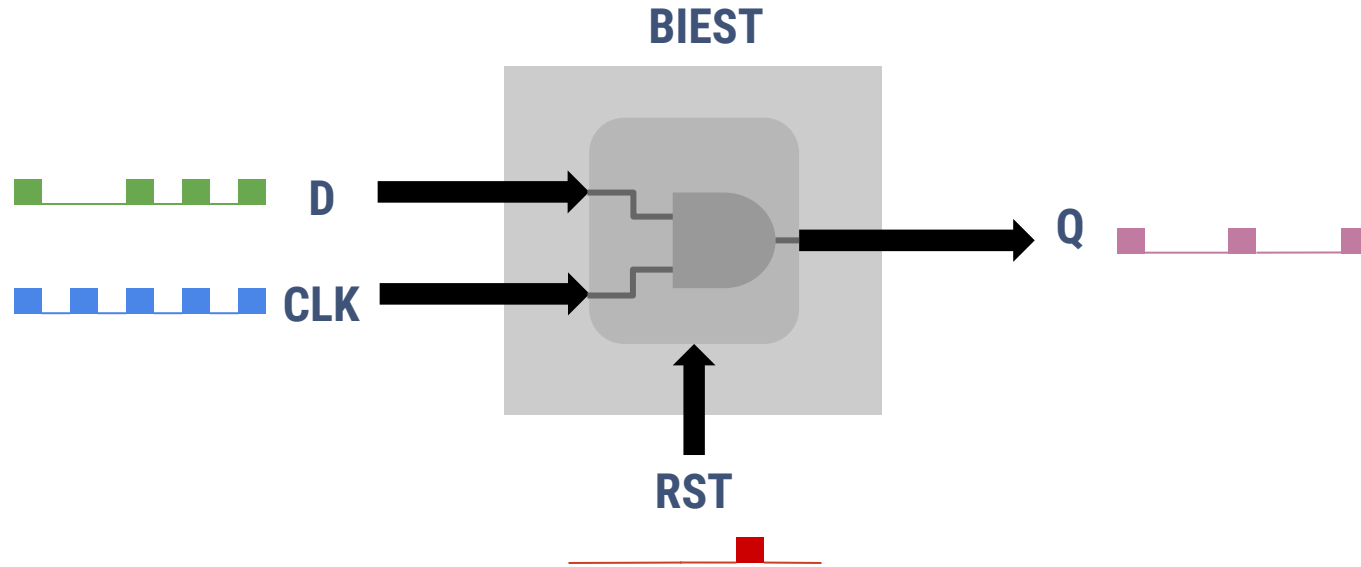
Arquitectura: Instanciación

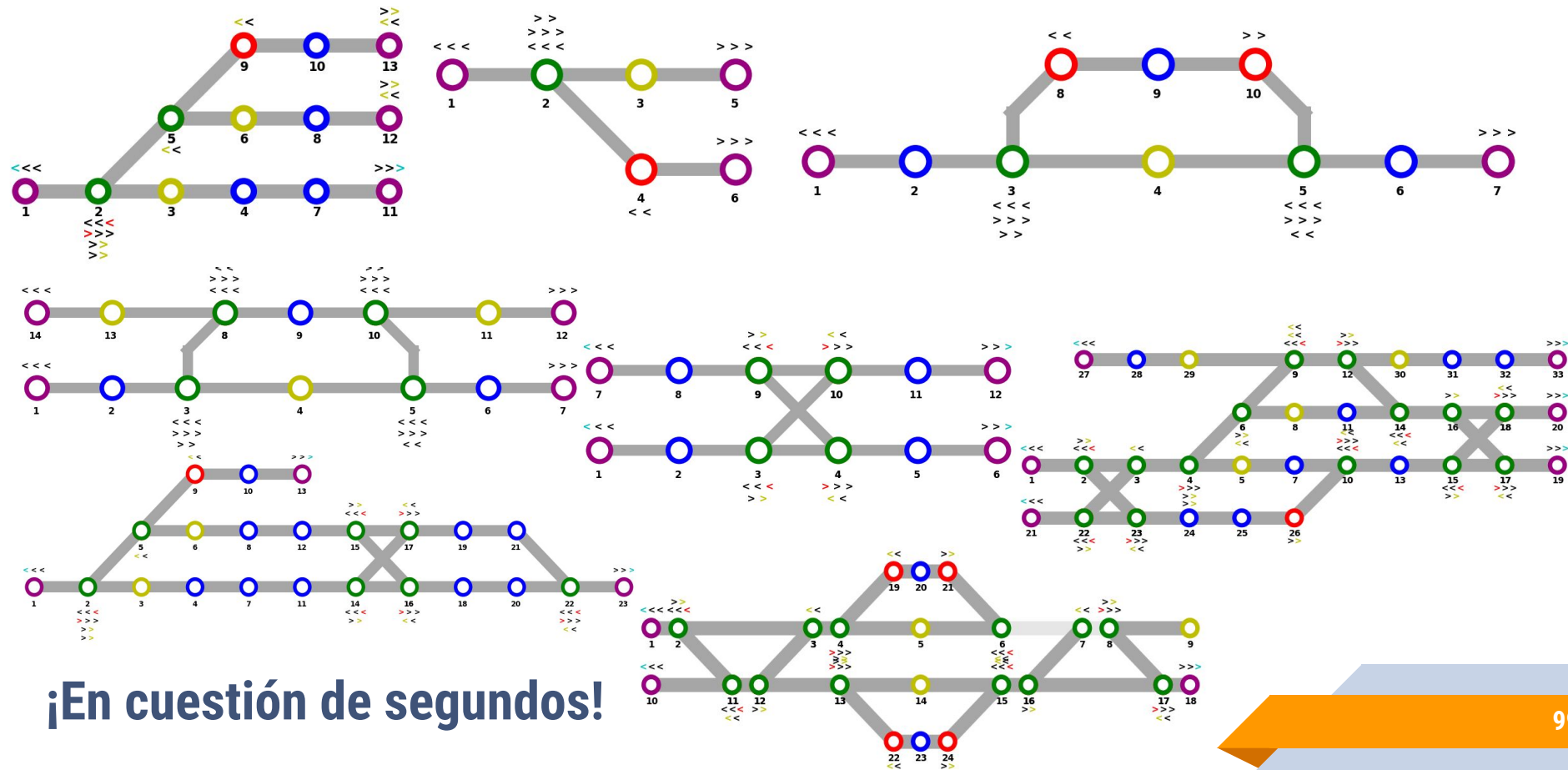


Proceso: Comportamiento



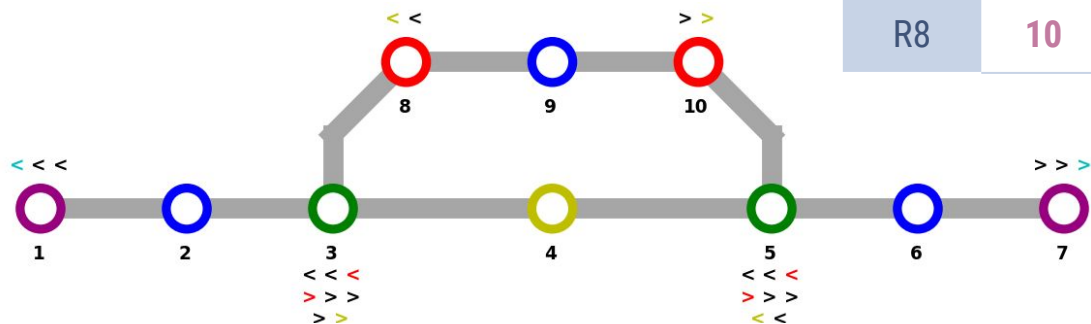
VHDL (Implementación de biestable)





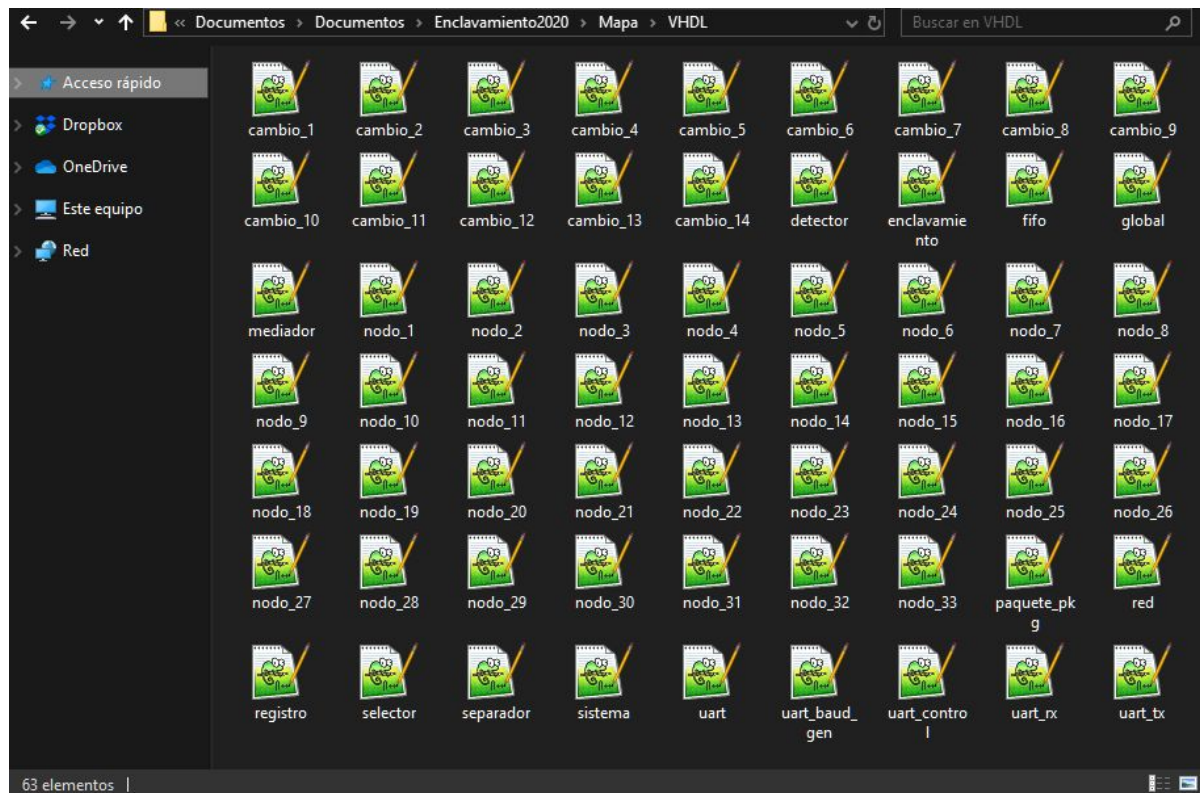
¡En cuestión de segundos!

La tabla es generada automáticamente contemplando **TODAS** las rutas soportadas por la red.

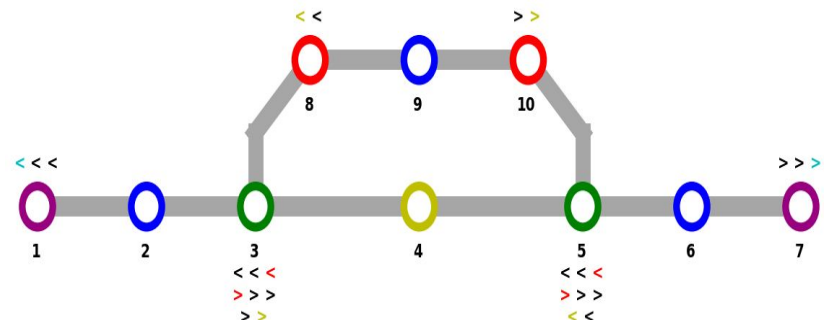
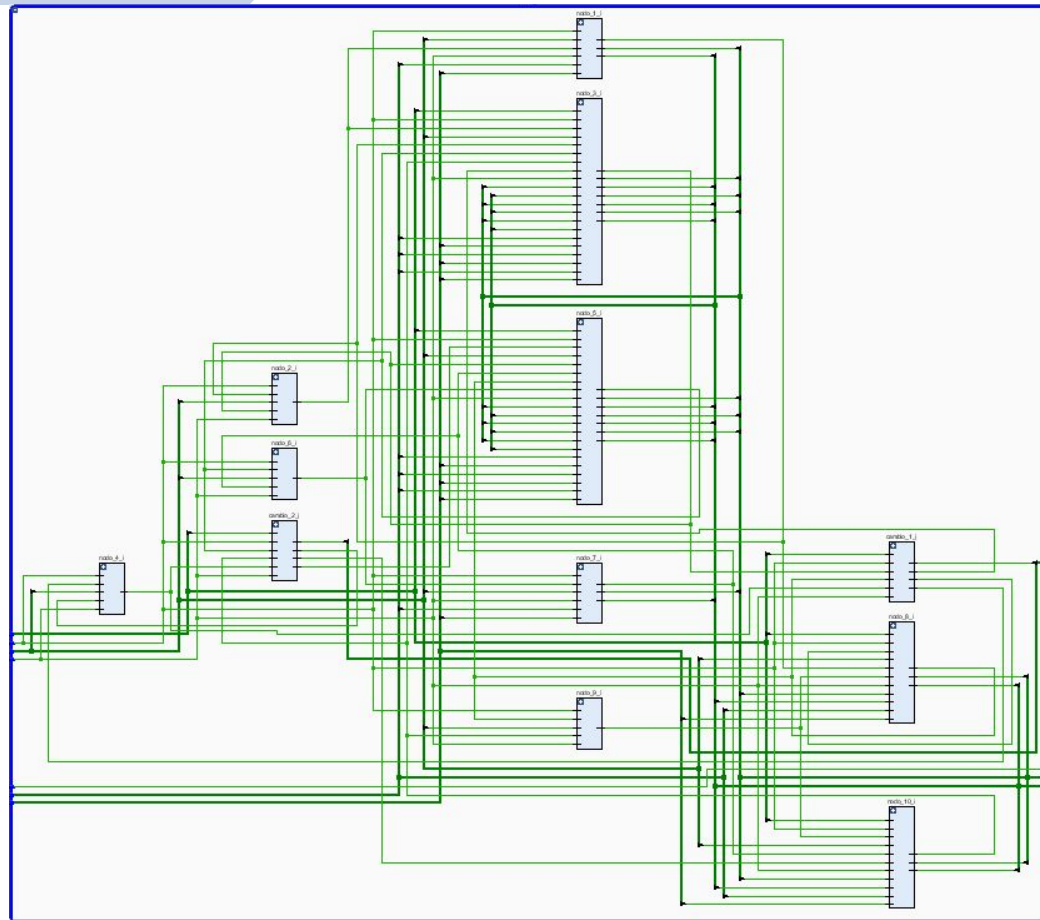


	Señal inicial	Señal final	Secuencia	Cambio	Sentido
R1	3	1	3-2-1	1-N	<
R2	3	5	3-4-5	1-N 2-N	>
R3	3	10	3-8-9-10	1-R	>
R4	5	3	5-4-3	1-N 2-N	<
R5	5	8	5-10-9-8	2-R	<
R6	5	7	5-6-7	2-N	>
R7	8	1	8-3-2-1	1-R	<
R8	10	7	10-5-6-7	2-R	>

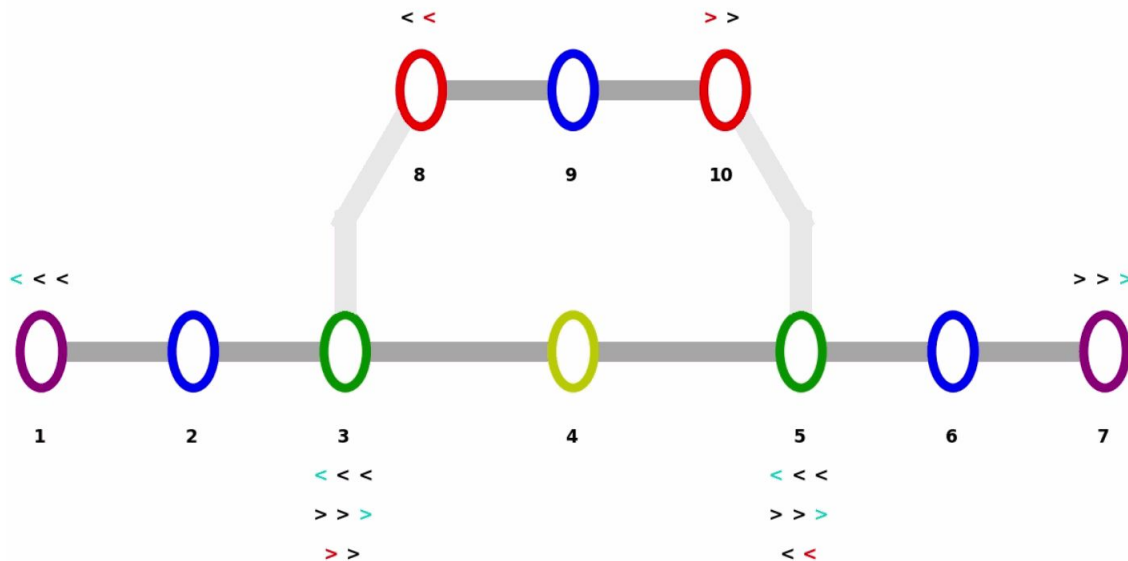
Todos los archivos VHDL necesarios son creados automáticamente.



Red generada automáticamente en base al grafo.

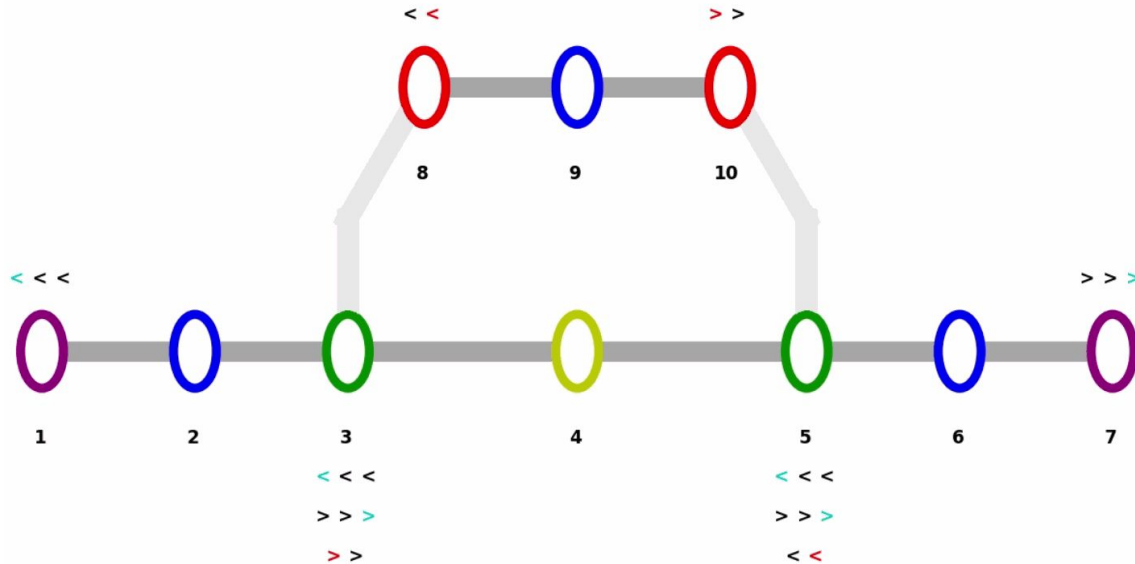


La FPGA devuelve el estado del señalamiento a la PC.
La PC muestra el estado en la interfaz diseñada.



Ejemplo de dos formaciones en sentido opuesto.

La FPGA devuelve el estado del señalamiento a la PC.
La PC muestra el estado en la interfaz diseñada.



Ejemplo de dos formaciones en el mismo sentido.

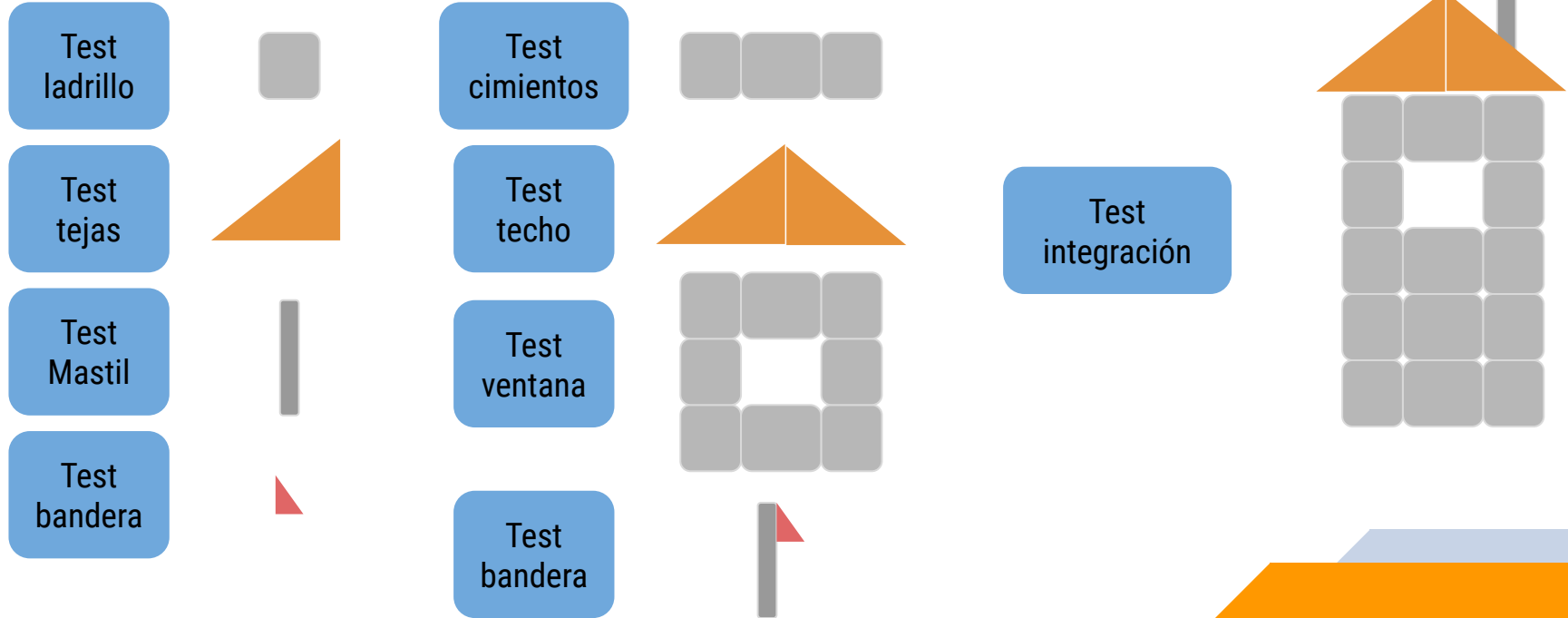
10

Testing

TDD y COCOTB



TDD (Desarrollo guiado por pruebas)





Testing en VHDL

Y es una única salida!

Lo que quiero obtener:

01001001000100100001010100101010101001010101011010011101001010101000100010

Lo que obtengo:

01001001000100100001010100101010101001010101011011011101001010101000100010

¿Y si fuese por más tiempo?

¿Y si agregan más entradas?

¿Y si las condiciones de error cambian?



Testing en VHDL

**Bender: Fry! Tuve una horrible pesadilla!
0s y 1s por todas partes! Y creo que vi un 2!**

Fry: Calma bender, el 2 no existe.

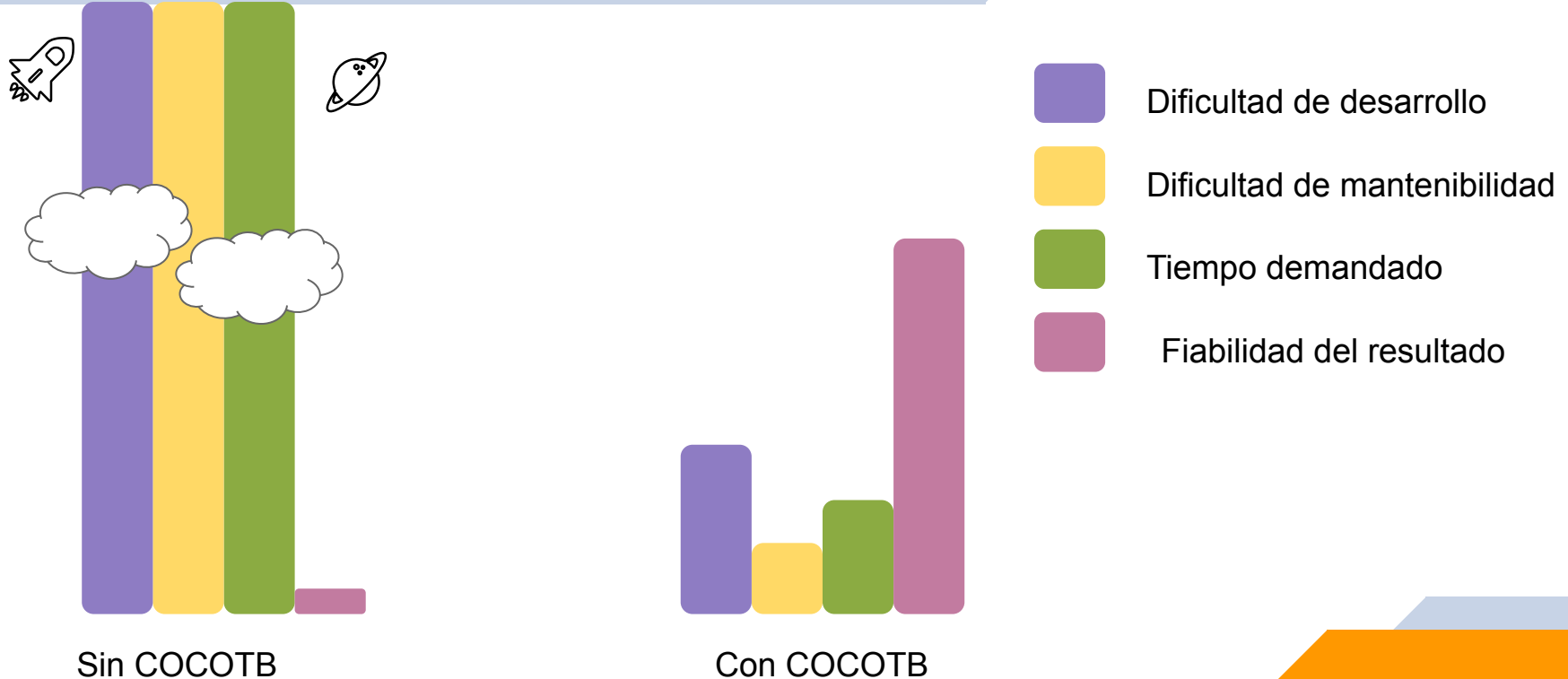


Sumale horas sin dormir y hasta puede aparecer un "2" sin que lo notes



Testing con COCOTB

Coroutine based COsimulation TestBench





¡Muchas gracias!

¿Alguna pregunta?