

SimSEE



# Diseño e Implementación

Marzo 2025  
IIE-FING-UdeLaR.

Ing. Ruben Chaer.  
[rchaer@simsee.org](mailto:rchaer@simsee.org)



## La necesidad de un modelo.

- La realidad nos resulta inalcanzable.
- Pensamos sobre modelos.



Simulador = Herramienta.  
¿Por qué y para qué?



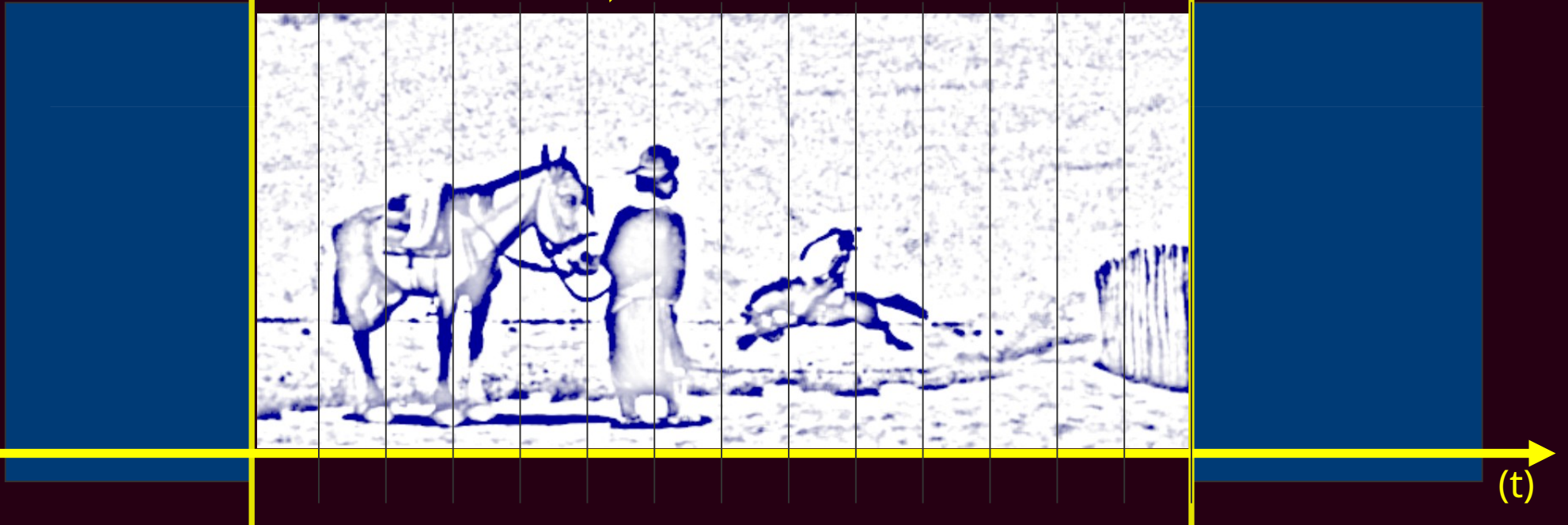
# Horizonte de y Paso de tiempo

$t_{inicial}$

$t_{final}$

Paso 1

Paso k



Horizonte

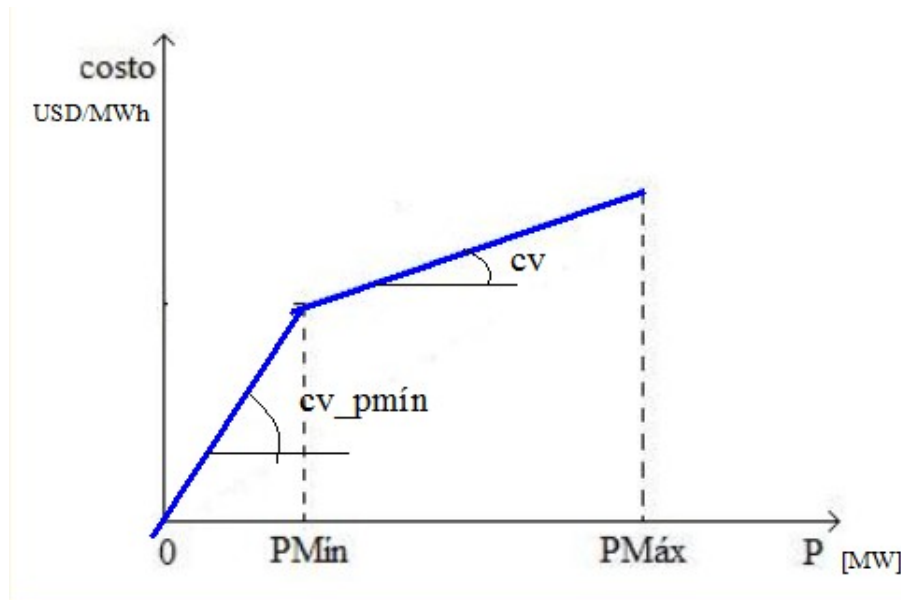
# Object Oriented Modeling.

## Modelado Orientado por los Objetos.

- Trauma de las correcciones.
- Es imposible no equivocarse, pero...



# Modelos



- Funcionalidad
- Calibración
- Mantenimiento
- Refinamiento

# Exactitud vs. Esfuerzo de cálculo.

En 1985, las máquinas tenían menos de 1MB de RAM y un micro-procesador, en 2022 es común laptops con 8 y 16GB y 8 procesadores y servidores de cálculo con 200GB de RAM y 48 procesadores.

En 1985, interconectar máquinas era costoso. En 2022 están todas interconectadas.

En 1985, los procesadores no llegaban a 100 MHz, 2022 superan los 3GHz y hay más de un procesador computadora.



# Lo más caro

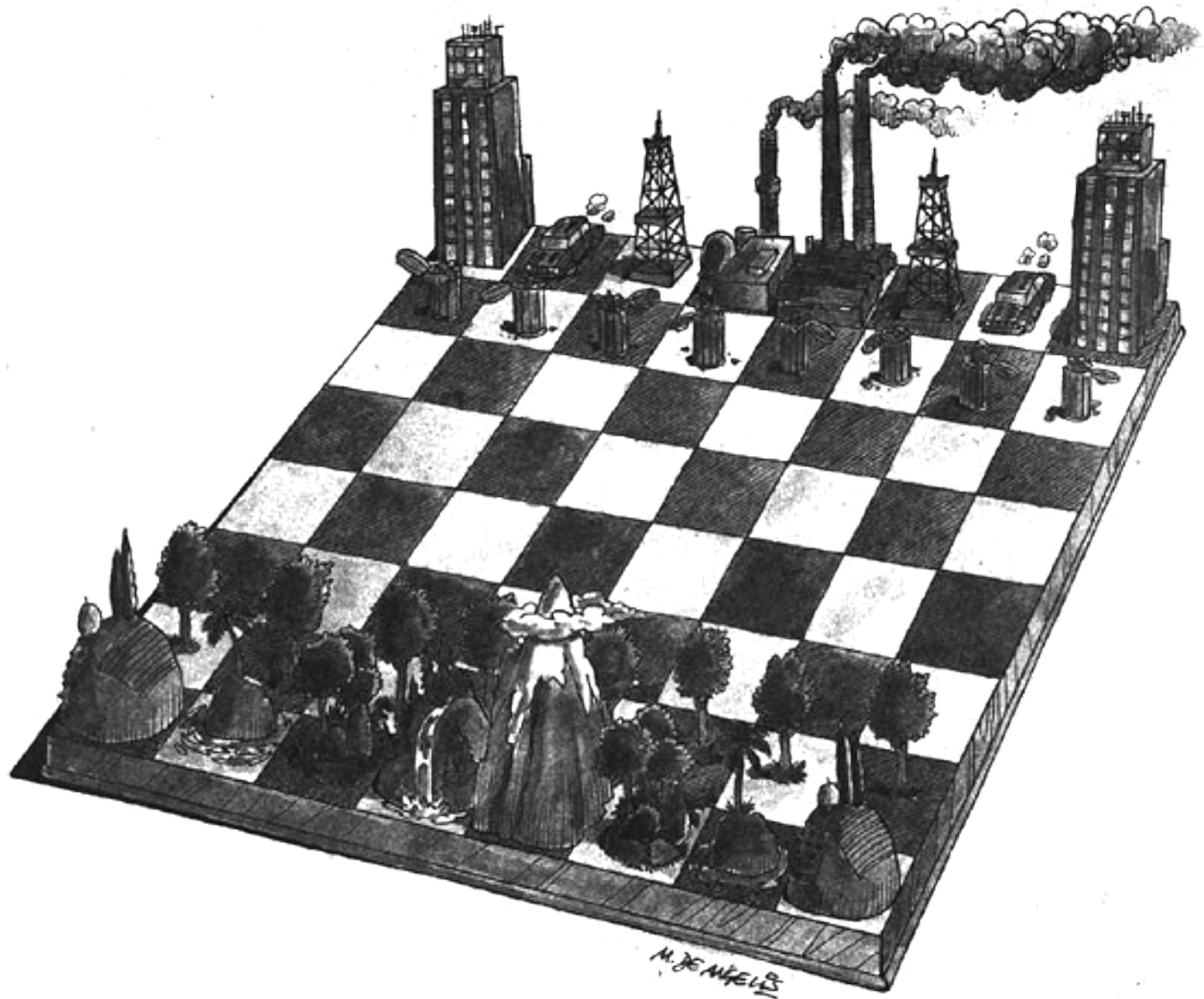




# Sala de Juego.



# Sala con Actores.



- Generadores
- Demandas
- Nodos
- Arcos



# Programación por Eventos

(mala traducción??? será SEÑALES, OCURRENCIAS???)

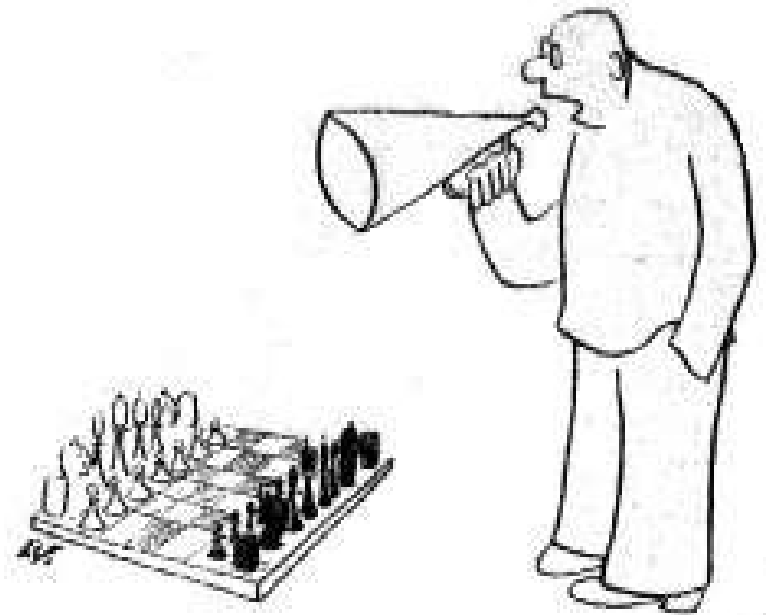
Repetir para cada paso:

<evSorteosDelPaso >

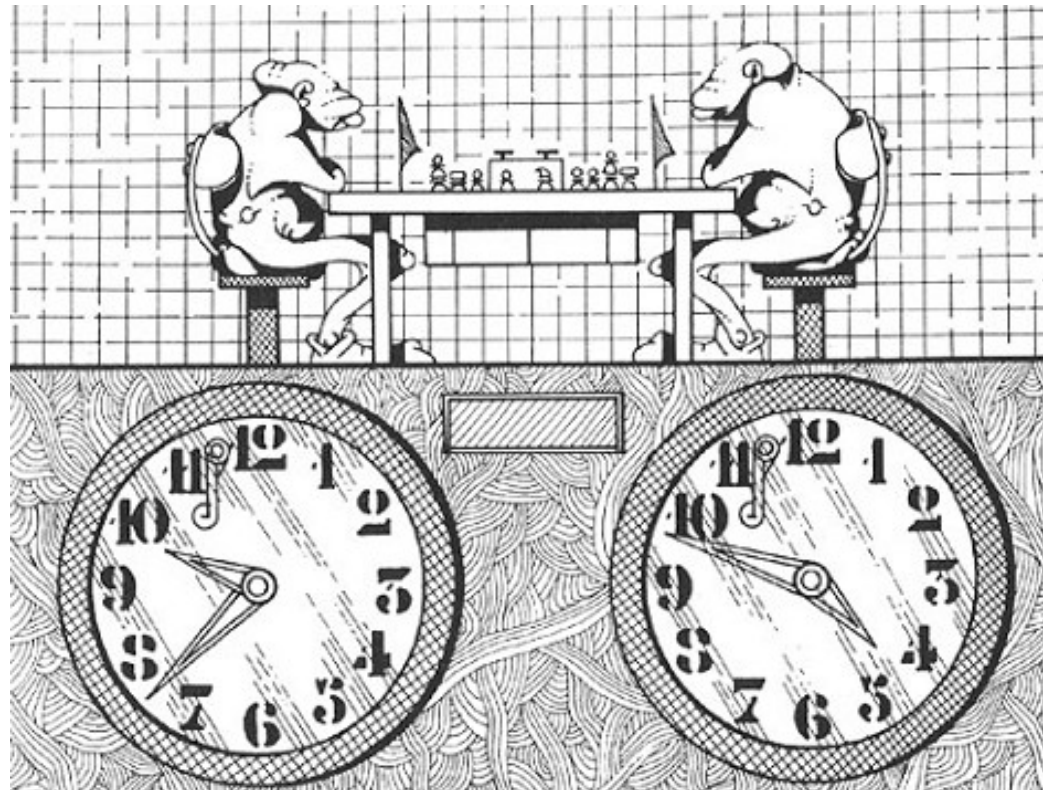
< evCargarProblema >

Resolver el despacho

< evLeerResultados >



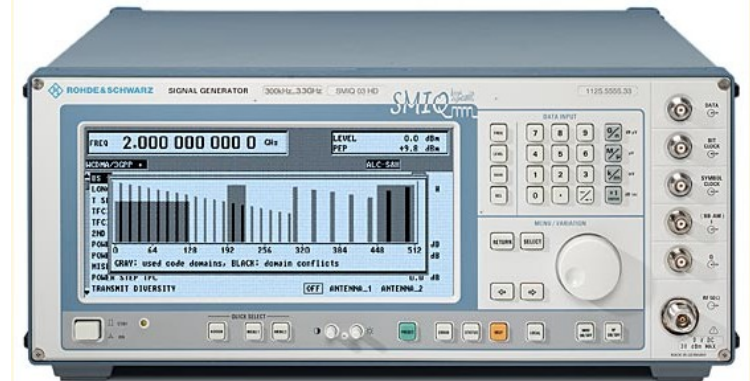
# Parámetros dinámicos.



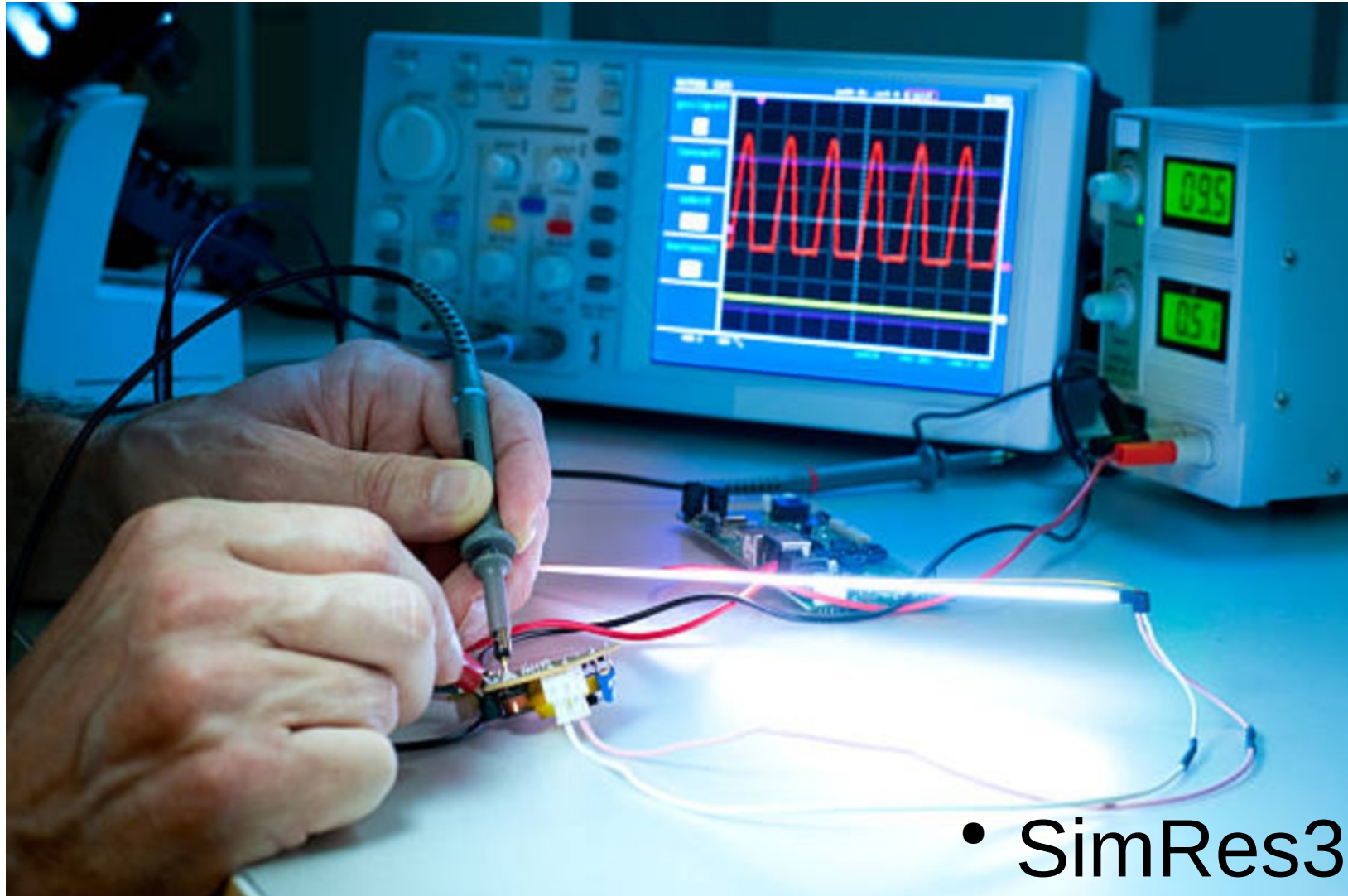
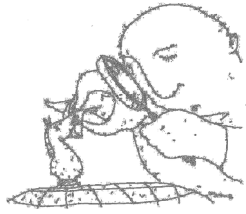
# Fuentes de valores



- Constante
- Uniforme
- Gaussiana
- Weibull
- Combinación
- Producto
- CEGH

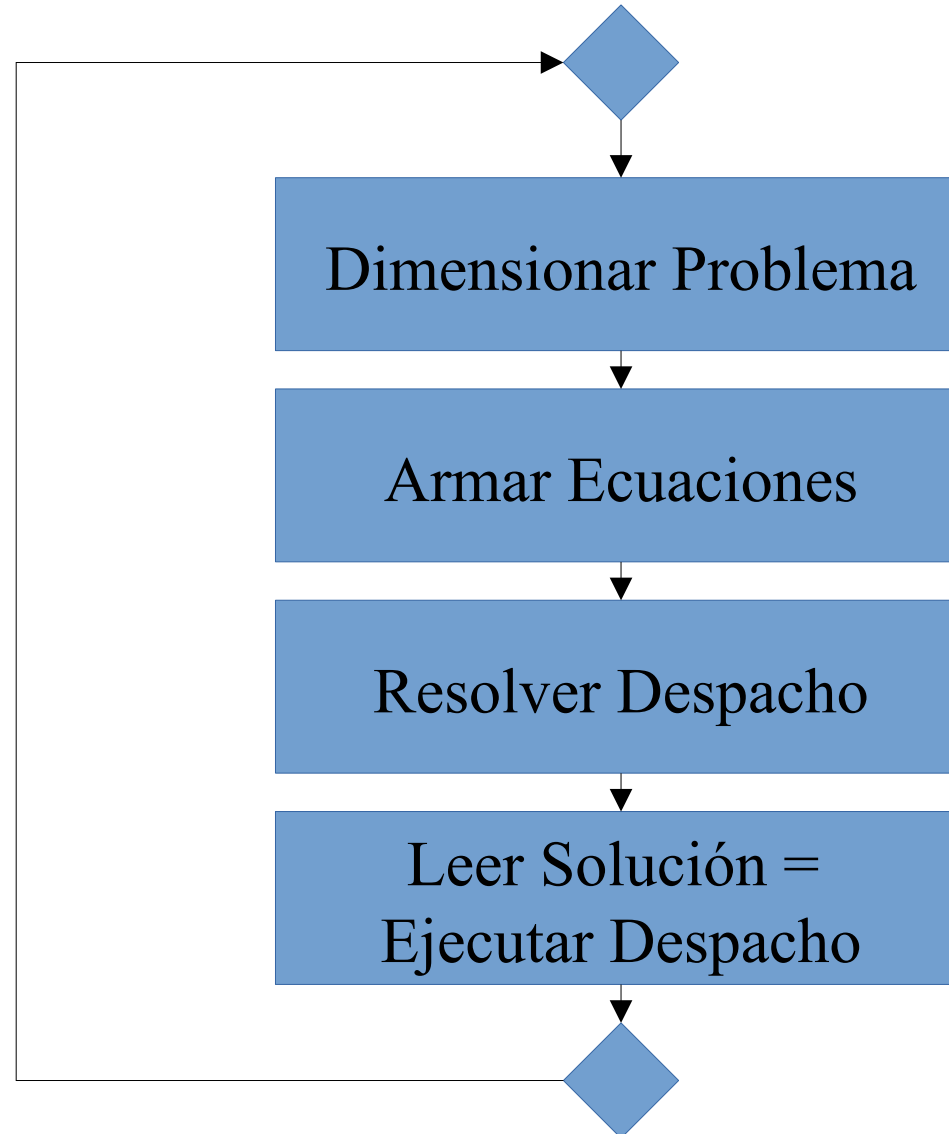


# Monitores



# Implementación de SimSEE

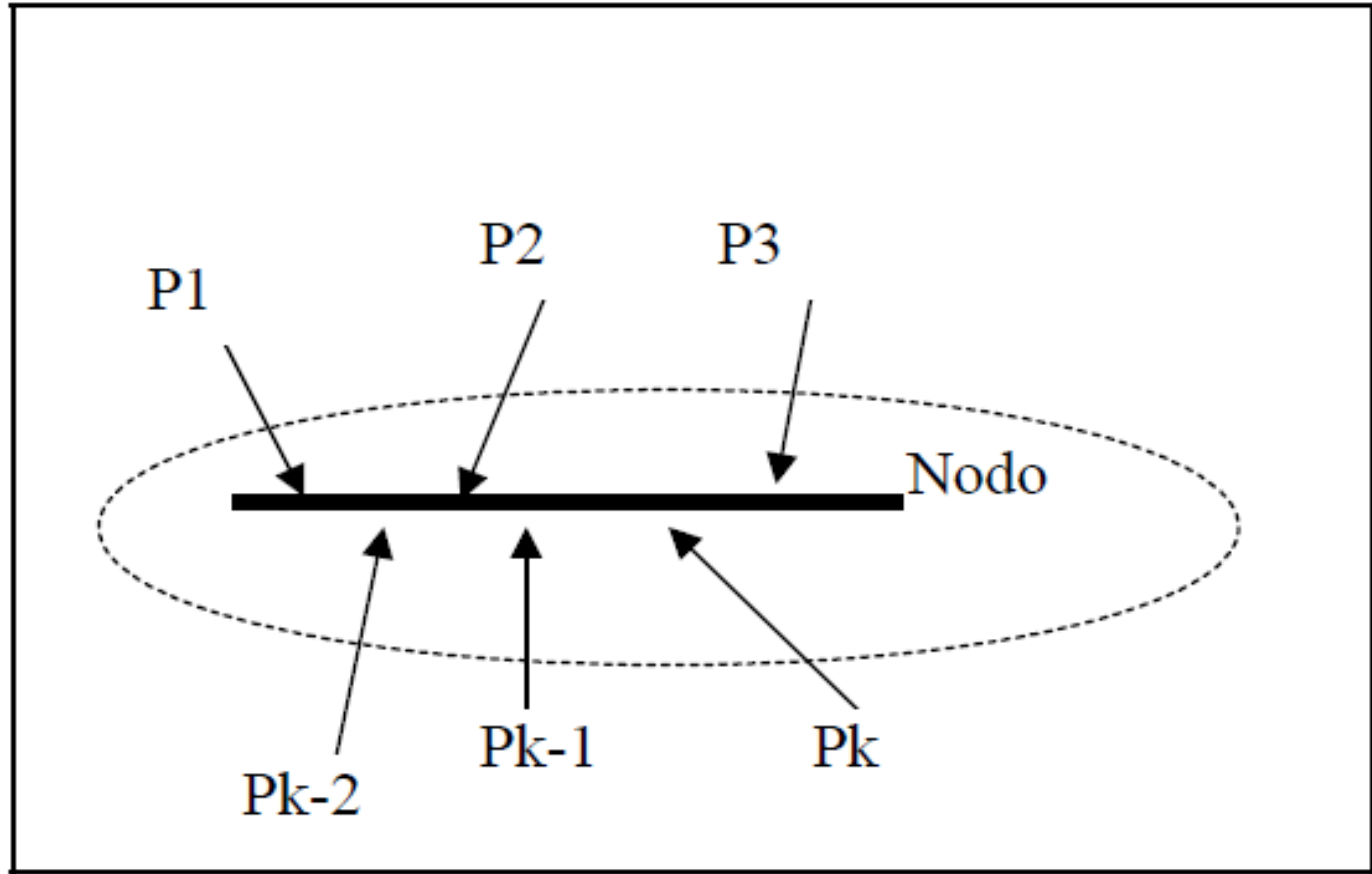
En cada paso de tiempo hay que:



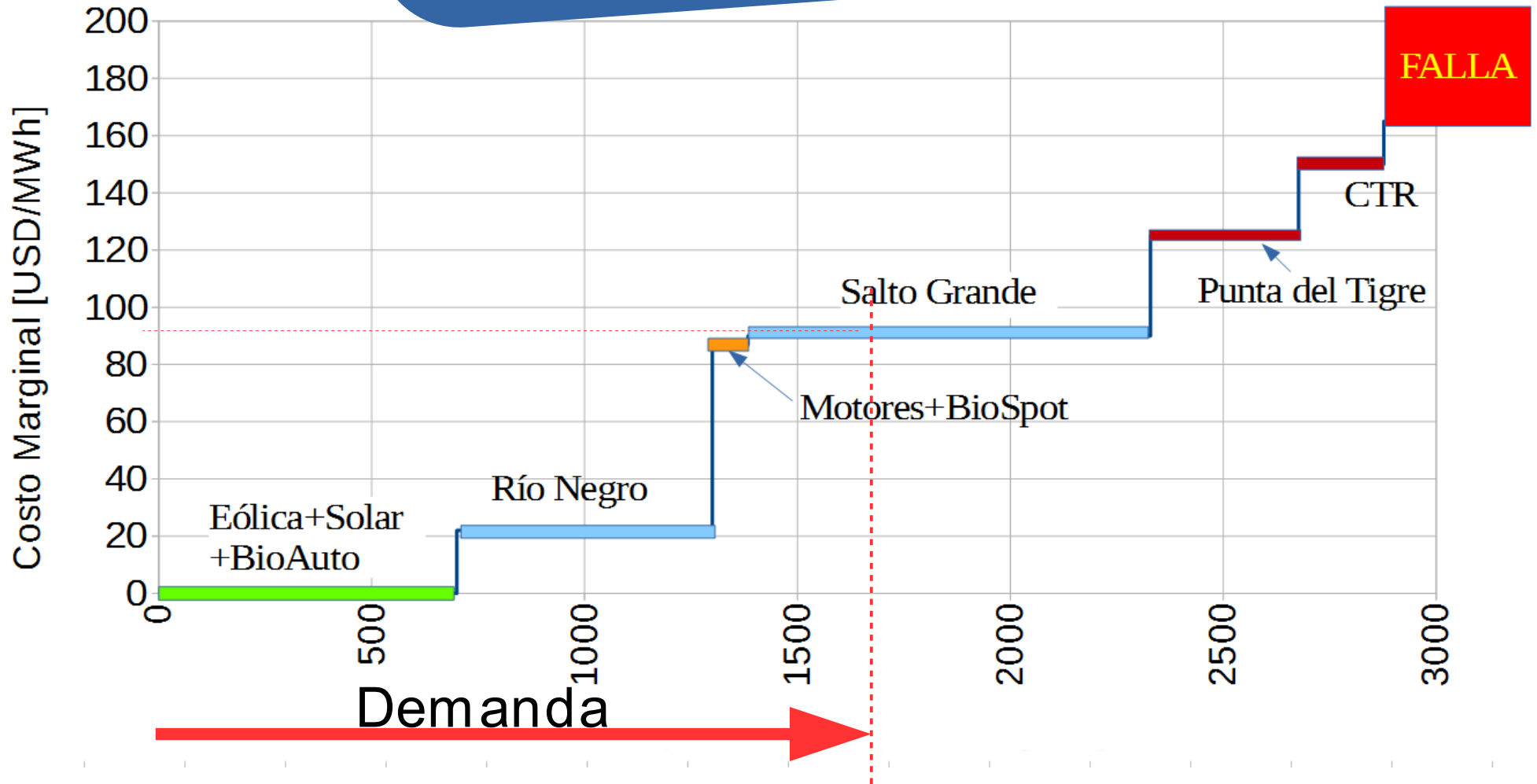


# Despacho y restricciones de Nodo

$$\sum_k P_k = 0 ;$$



# Orden de mérito



## ***Despacho como problema de optimización.***

$$\begin{aligned} & \text{mín} \left( \sum_k cv_k G_k \Delta t \right) \\ & @ \sum_h G_k - \sum_k D_h = 0 \end{aligned}$$

Donde:

- $G_k$  es la potencia inyectada por el generador  $k$  en MW.
- $D_h$  es la potencia retirada por la demanda  $h$  en el nodo en cuestión
- $cv_k$  es el costo de generación del generador  $k$  expresado en USD/MWh cuando entrega una potencia  $PG_k$  y
- $\Delta t$  es la duración del paso de tiempo en horas.

# Resolución de un paso

$$\underset{x}{\text{máx}} \left( - \sum_{j=1}^{j=nv} c_j \cdot u_j \right)$$

*sujeto a :*

$$RD_i) : \sum_{j=1}^{j=nv} a_{ij} \cdot u_j + b_i \geq 0 ; i = 1..NRD$$

$$RI_l) : \sum_{j=1}^{j=nv} a_{lj} \cdot u_j + b_l = 0 ; l = 1..NRI$$

# Tabla del problema.

		Variables de control					
		u1	u2	...	Unv	1	
Restricciones	R1	a11	a12	...	a1nv	b1	$\geq 0$
	R2	a21	a22	...	a2nv	b2	$= 0$
	...	...	...	...	...	...	...
	RM	am1	am2	...	amnv	bm	$\geq 0$
	-fc	-c1	-c2	...	-cnv		

$$m = \text{NRD} + \text{NRI}$$

Tipo de restricción (Desigualdad o Igualdad ).

Restricciones de caja.



# Implementación COLABORATIVA de los Actores para cargar un TSimplex.

# Dimensionado

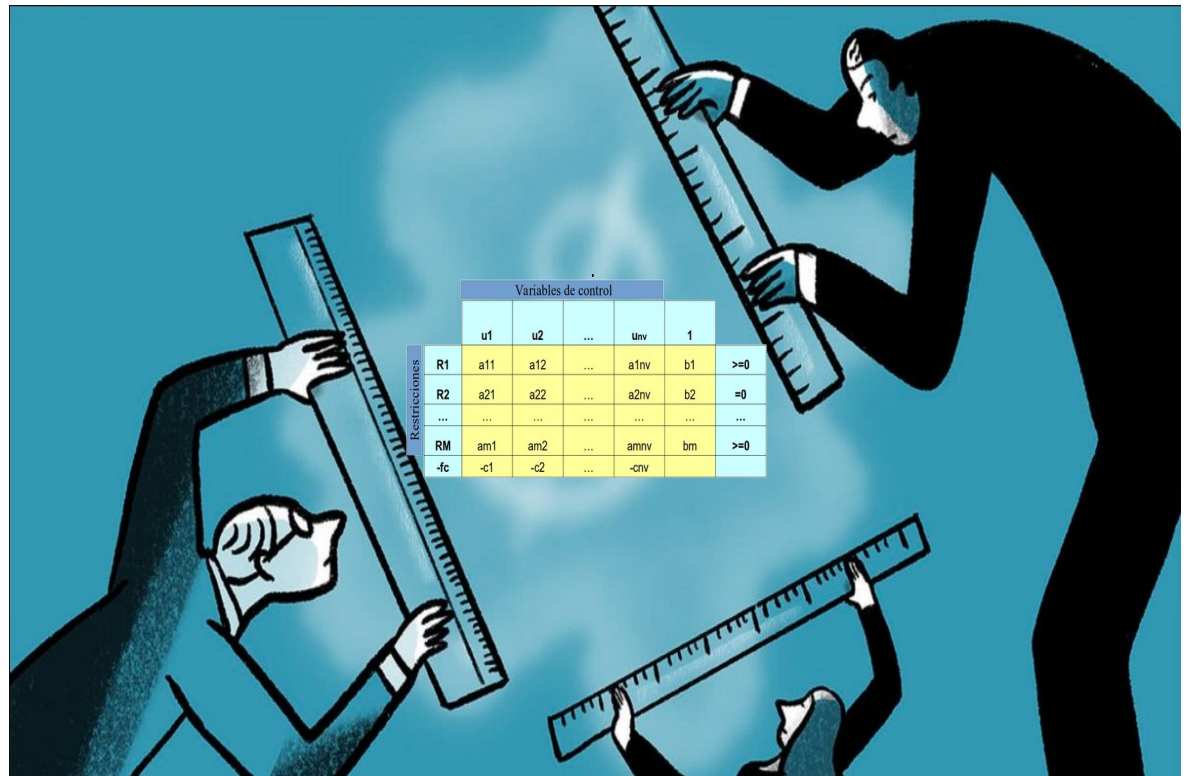


Consulta a todos los actores.

Variables de control “ $u$ ”, Cantidad de restricciones  
(y cantidad de variables enteras).

# Dimensionado del MIP Simplex

- Ponemos el índice de variables y de restricciones en UNO y llamamos a cada actor pasándole dichos índices para que cada actor guarde el valor de los mismos en el momento y que los incremente en la cantidad de variables y restricciones.
- De esa forma, además de tener la dimensión del problema, logramos que cada Actor sepa el lugar de inicio de sus variables y sus restricciones en la tabla del problema.





# El código

```
ivar:= 1; // índice de variables de control  
ires:= 1; // índice de restricciones adicionales  
ivae:= 1; // índice de variables enteras  
  
for k:= 0 to high(actores) do  
    actores[k].opt_nvers( ivar, ivae, ires );
```

# Recorrida de carga del MIPSimplex



		Variables de control					
		u1	u2	...	Unv	1	
Restricciones	R1	▶ a11	▶ a12	...	▶ a1nv	▶ b1	≥ 0
	R2	▶ a21	▶ a22	...	▶ a2nv	▶ b2	= 0
	...	...	...	...	...	...	...
	RM	▶ am1	▶ am2	...	▶ amnv	▶ bm	≥ 0
	-fc	▶ -c1	▶ -c2	...	▶ -cnv		

# Acceso a la matriz del Simplex

- s.pon\_e( kFil, jCol, val )
- s.acum\_e( kFil, jCol, val )

		Variables de control					
		u1	u2	...	Unv	1	
Restricciones	R1	a11	a12	...	a1nv	b1	$\geq 0$
	R2	a21	a22	...	a2nv	b2	$= 0$
	...	...	...	...	...	...	...
	RM	am1	am2	...	amnv	bm	$\geq 0$
	-fc	-c1	-c2	...	-cnv		

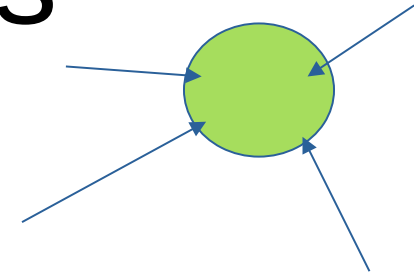
# El código de carga.

- `// creamos el Simplex  
spx:= TMIPSimplex.Create_init( ires, ivar, ivae-1 );`
- `// Los Actores cargan la matriz  
for k:= 0 to high( actores ) do  
    actores[k].opt_cargue( spx );`
- `// Los Actores cargan las restricciones de caja  
for k:= 0 to high( actores ) do  
    actores[k].opt_fijarRestriccionesDeCaja( spx );`

# Resolución del problema de despacho de la etapa

```
spx.resolver;
```

# TNodo opt\_nvers



- Variables de control = 0
- Restricciones = NPOSTES

```

procedure TNodo.opt_nvers( var ivar, ivae, ires: integer );
begin
  // el nodo no agreaga variables de control
  // no modificamos ivar, ni ivae y no necesitamos guardar sus valores

  // El NODO agrega una restricción por cada POSTE
  Self.ires:= ires; // guardamos el valor de ires
  ires:= ires + globs.NPostes; // incrementamos ires en la cantidad de POSTES
end;
  
```

		Variables de control					
		u1	u2	...	unv	1	
Restricciones	R1	a11	a12	...	a1nv	b1	$\geq 0$
	R2	a21	a22	...	a2nv	b2	$= 0$
	...	...	...	...	...	...	...
	RM	am1	am2	...	amnv	bm	$\geq 0$
	-fc	-c1	-c2	...	-cnv		

# TNodo opt\_cargue

Cargando el problema: El nodo no carga coeficientes en la matriz del problema, pero debe indicar que las restricciones de nodo son de igualdad.

```

.
. procedure TNodo.opt_cargue(s: TSimplex);
270 begin
.   // Nada que decir.
. end;

```

		Variables de control					
		u1	u2	...	Unv	1	
Restricciones	R1	a11	a12	...	a1nv	b1	>=0
	R2	a21	a22	...	a2nv	b2	=0
	...	...	...	...	...	...	...
	RM	am1	am2	...	amnv	bm	>=0
	-fc	-c1	-c2	...	-cnv		

# TNodo

## opt\_fijarRestriccionesDeCaja

```

.
. procedure TNodo.opt_fijarRestriccionesDeCaja(s: TSimplex);
275 var
.   ip: integer;
. begin
.   for ip := 0 to globs.NPostes - 1 do
.     s.FijarRestriccionIgualdad(ires + ip);//;
280 end;

```

		Variables de control					
		u1	u2	...	unv	1	
Restricciones	R1	a11	a12	...	a1nv	b1	$\geq 0$
	R2	a21	a22	...	a2nv	b2	$= 0$
	...	...	...	...	...	...	...
	RM	am1	am2	...	amnv	bm	$\geq 0$
	-fc	-c1	-c2	...	-cnv		



# TNodo opt\_LeerResultados

```

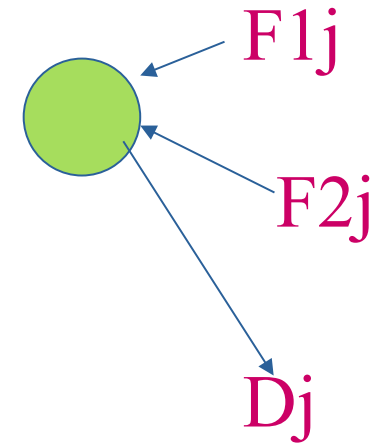
- procedure TNodo.opt_leerSolucion( s: TSimplex );
-   var
-     iposte: integer;
-
-   begin
-     // no tiene variables no tiene que leer las variables.
40  // lo que podemos leer es el multiplicador de Lagrange para tener
-     // el costo marginal del nodo.
-     for iposte:= 0 to globs.NPostes-1 do
-       cmarg[iposte]:= -s.ymult( ires+ iposte ) / globs.durpos[ iposte ];
-   end;
-

```

		Variables de control					
		u1	u2	...	Unv	1	
Restricciones	R1	a11	a12	...	a1nv	b1	>=0
	R2	a21	a22	...	a2nv	b2	=0
	...	...	...	...	...	...	...
	RM	am1	am2	...	amnv	bm	>=0
	-fc	-c1	-c2	...	-cnv		

# TDemanda opt\_nvers

- Variables de control =  $NF \cdot NPOSTES$
- Restricciones = 0



```

procedure TDemanda01.opt_nvers( var ivar, ivae, ires: integer );
begin
  Self.ivar:= ivar;
  // tengo para cada escalón de falla una variable por poste
  ivar:= ivar+ NEscalonesDeFalla*globs.NPostes ;
end;
  
```

		Variables de control					
		u1	u2	...	Unv	1	
Restricciones	R1	a11	a12	...	a1nv	b1	$\geq 0$
	R2	a21	a22	...	a2nv	b2	$= 0$
	...	...	...	...	...	...	...
	RM	am1	am2	...	amnv	bm	$\geq 0$
	-fc	-c1	-c2	...	-cnv		

# TDemanda opt\_cargue

Carga en la tabla, en cada restricción del nodo un 1 en la columna de las Fhj y Dj en el termino independiente.

```
procedure TDemanda01.opt_cargue( s: TSimplex );
```

```
var
```

```
  iposte, iescalon: integer;
```

```
  ibaseres: integer;
```

```
begin
```

```
  ibaseres:=nodo.i-res;
```

```
  // Carga el término constante de las restricciones de demanda
```

```
  for iposte:= 0 to high( PPa ) do
```

```
    s.acum_e( ibaseres + iposte, s.nc, -PPa[iposte] );
```

```
  // Ahora corresponde cargar las máquinas de falla
```

```
  for iescalon:= 0 to high( falla_profundidad ) do
```

```
    for iposte:= 0 to globs.NPostes-1 do
```

```
      begin
```

```
        // cargamos la potencia en la restricción del nodo
```

```
        s.pon_e( ibaseres+iposte, ivar+iescalon*globs.NPostes+iposte, 1 );
```

```
        // cargamos el coeficiente del costo en la función objetivo
```

```
        s.pon_e( s.nf, ivar+iescalon*globs.NPostes+iposte, -falla_costo[ iescalon ]* globs.durpos[ iposte ] );
```

```
      end;
```

```
end;
```

Variables de control

		u1	u2	...	unv	1	
Restricciones	R1	a11	a12	...	a1nv	b1	>=0
	R2	a21	a22	...	a2nv	b2	=0
	...	...	...	...	...	...	...
	RM	am1	am2	...	amnv	bm	>=0
	-fc	-c1	-c2	...	-cnv		

# TDemanda

## opt\_fijarRestriccionesDeCaja

```

10 procedure TDemanda01.opt_fijarRestriccionesDeCaja( s: TSimplex );
var
    iescalon, iposte: integer;
    Fpmax: NReal;
begin
    for iescalon:= 0 to high( falla_profundidad ) do
// Le fijamos la potencia máxima de cada escalón en cada poste
        for iposte:= 0 to globs.NPostes-1 do
            begin
                Fpmax:= PPa[iposte]* Self.falla_profundidad[iescalon];
                s.cota_sup_set( ivar+iescalon*globs.NPostes+iposte, Fpmax);
            end;
        end;
    end;
end;

```

		Variables de control					
		u1	u2	...	Unv	1	
Restricciones	R1	a11	a12	...	a1nv	b1	>=0
	R2	a21	a22	...	a2nv	b2	=0
	...	...	...	...	...	...	...
	RM	am1	am2	...	amnv	bm	>=0
	-fc	-c1	-c2	...	-cnv		

# TDemanda opt\_LeerSolucion

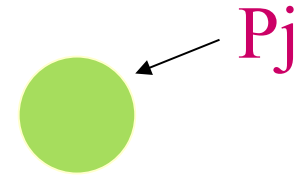
```

- procedure TDemanda01.opt_leerSolucion( s: TSimplex );
-   var
-     iescalon, iposte: integer;
00 begin
-   // recuperamos los valores de Potencia de cada escalón de falla en
-   // cada poste.
-   for iescalon:= 0 to high( falla_profundidad ) do
-     for iposte:= 0 to globs.NPostes-1 do
-       fallas[iescalon][iposte]:= s.xval( ivar+iescalon*globs.NPostes+iposte );
-   end;
-
-

```

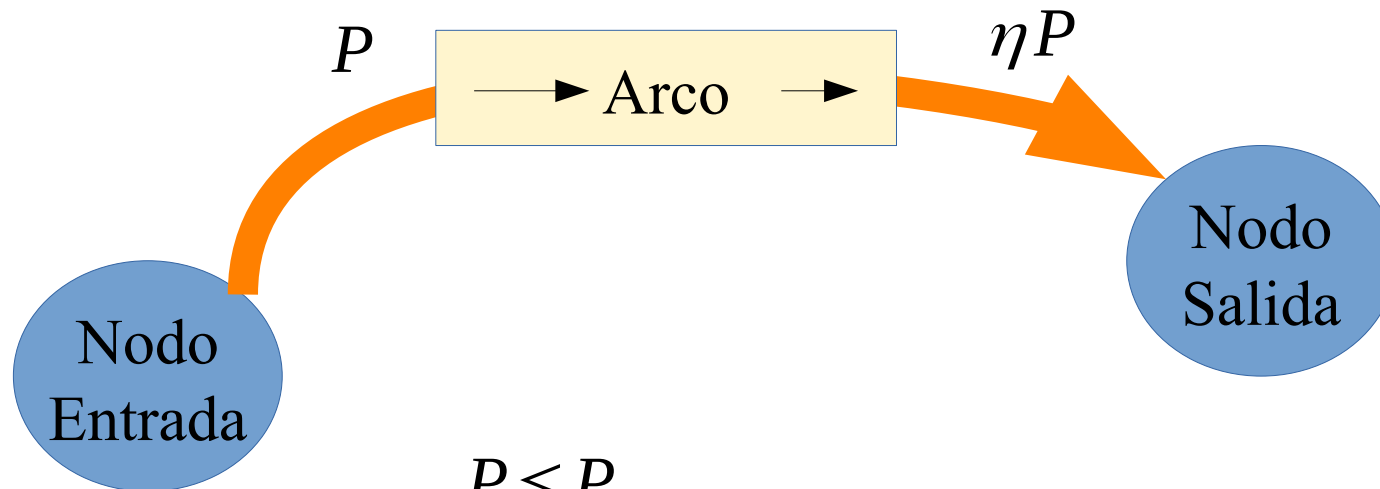
		Variables de control					
		u1	u2	...	unv	1	
Restricciones	R1	a11	a12	...	a1nv	b1	$\geq 0$
	R2	a21	a22	...	a2nv	b2	$= 0$
	...	...	...	...	...	...	...
	RM	am1	am2	...	amnv	bm	$\geq 0$
	-fc	-c1	-c2	...	-cnv		

# TGTer\_Basico



- Variables de control = NPOSTES
- Restricciones = 0
- Variables de estado = 0
- Carga en la tabla, en cada restricción del nodo un 1 en la columna de las  $P_j$  y en la fila de costo  $cv \cdot DT_j$  en cada columna de las  $P_j$ .

# TArco



$$P \leq P_{Máx}$$

$$Costo = cv_{Peaje} P \Delta T$$

# TArco-Parámetros

- Nombre
- Nodo de entrada
- Nodo de salida
- Cantidad de unidades (líneas)
- Parámetros dinámicos:
  - +Rendimiento
  - +Peaje [\$/MWh]
  - +Factor de disponibilidad
  - +Potencia máxima



# TArco-Variables y Restricciones

- Como variable de control tenemos la potencia por el arco en cada poste.
- La restricción es la del límite máximo de potencia.  $P_j < P_{\text{máx.}}$  (de caja, no adicional)
- El arco aparece como un generador para el nodo de partida de potencia  $-P$  y como un generador para el nodo de llegada de potencia  $\mu P$ .

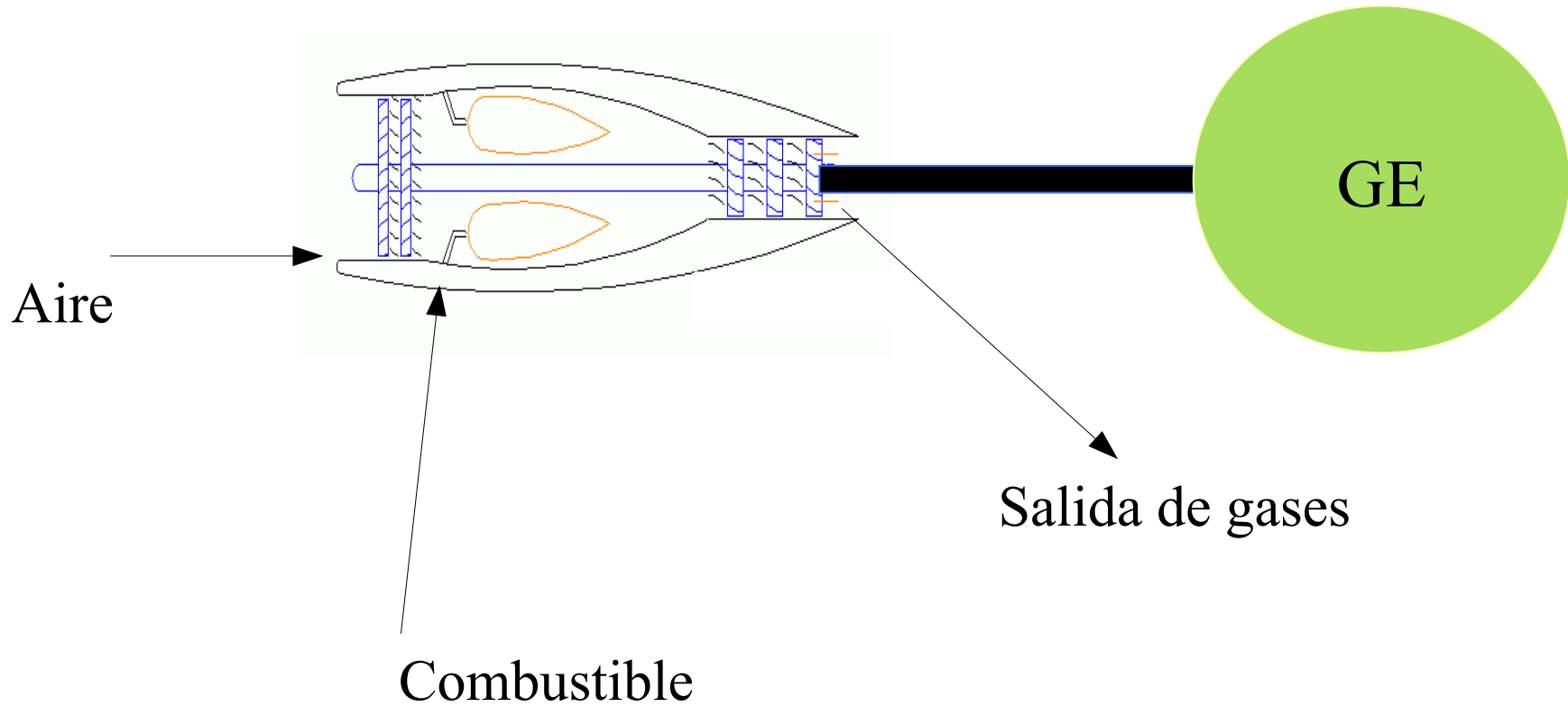
# TArco-Resultados del despacho

- potencia por poste
- multiplicador de Lagrange de la restricción.

# Centrales Térmicas

# Turbina aeroderivativa

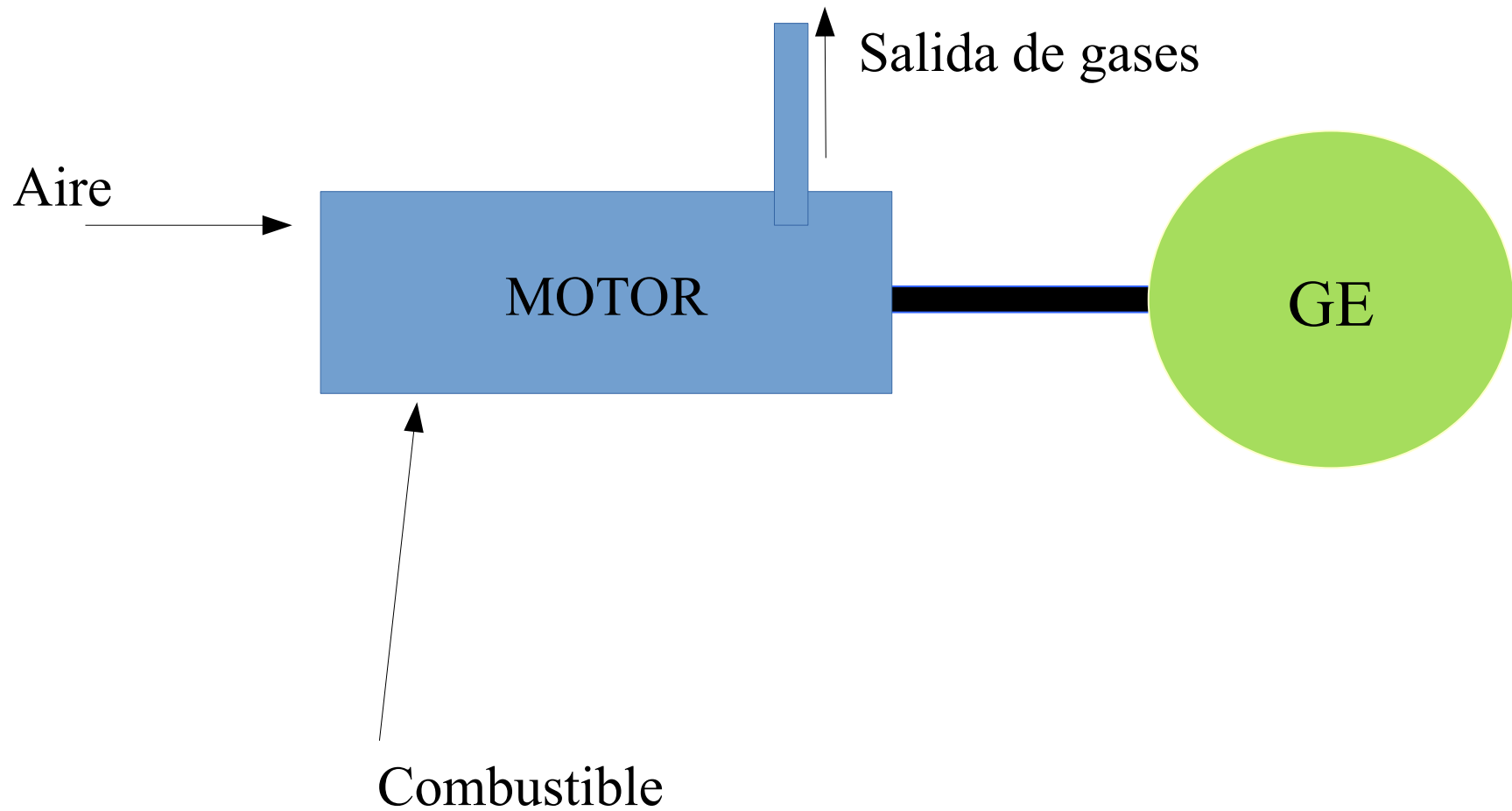
Arranque rápido, seguimiento de carga.



Gasoil / Gas Natural

# Moto-generador

Arranque rápido, seguimiento de carga.

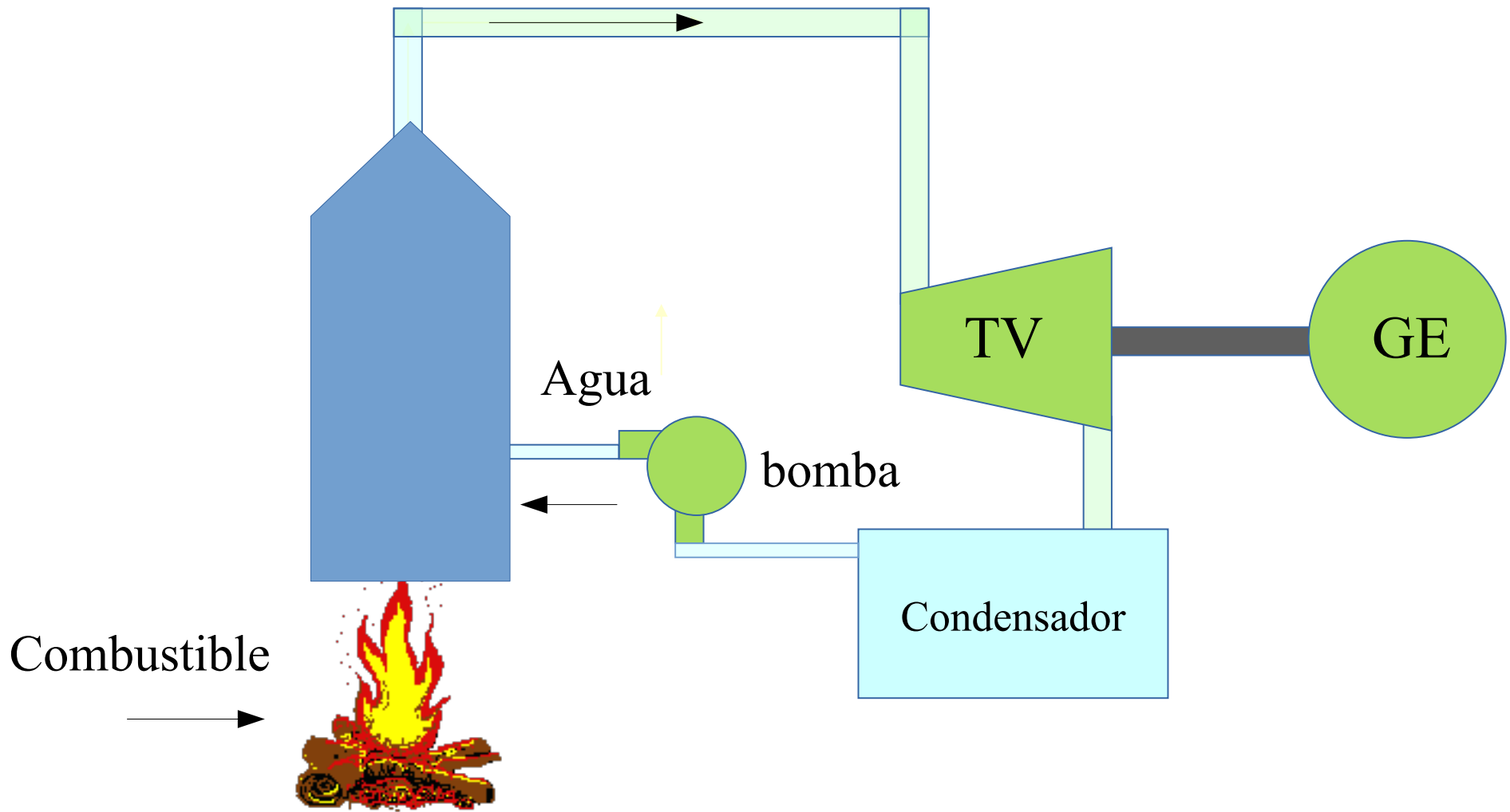


Gasoil / Gas Natural / Fuel-oil

# Turbo Vapor

Arranque lento, central de base.

Vapor a presión

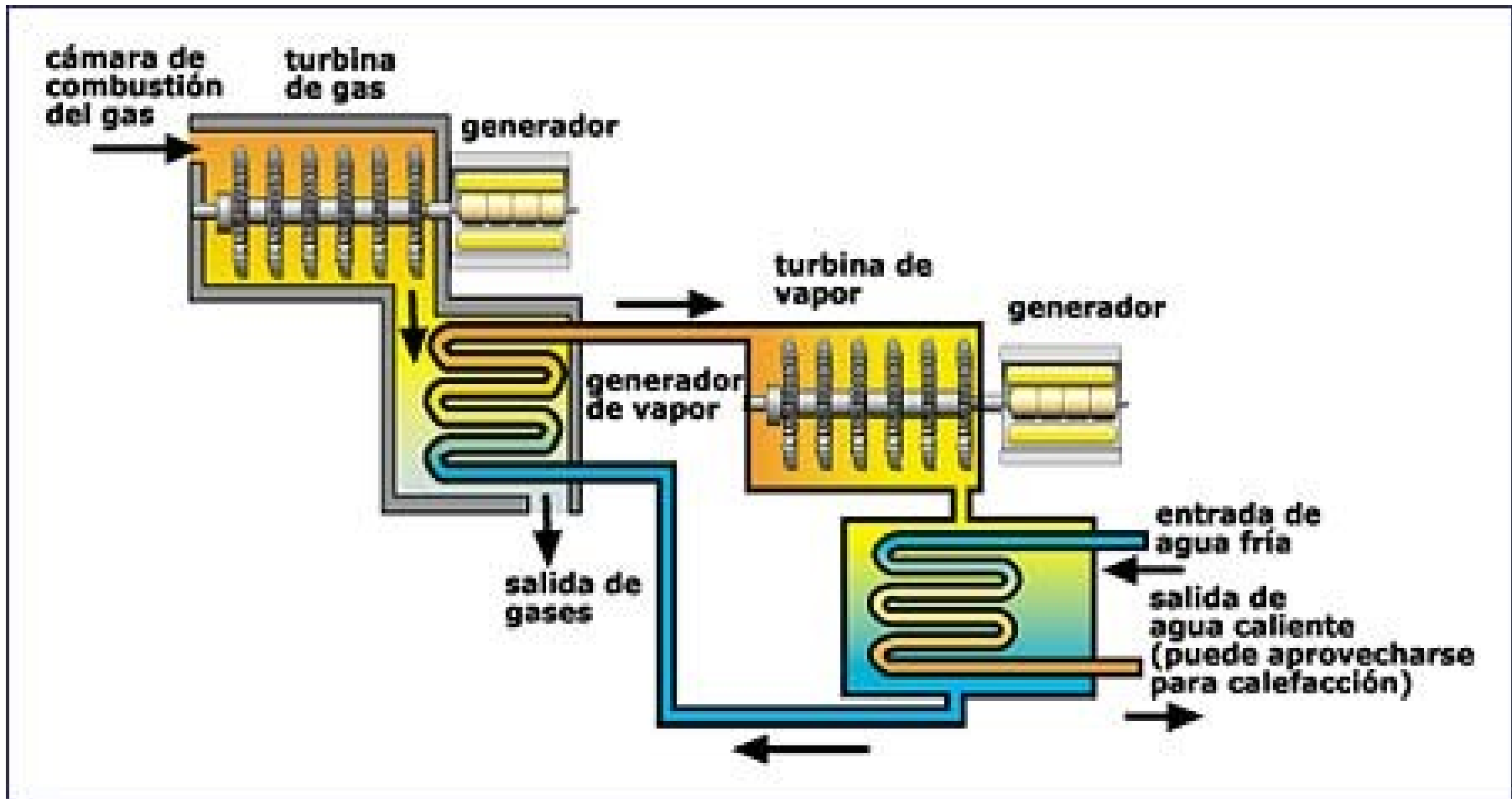


Combustible

Fuel Oil / Carbón / Nuclear / Biomasa

# Ciclo Combinado (2TG + 1TV)

Combinado: Arranque lento, central de base.  
 Los modernos son eficientes siguiendo la carga.  
 En ciclo abierto, son TGs.



Gasoil / Gas Natural

# Videos



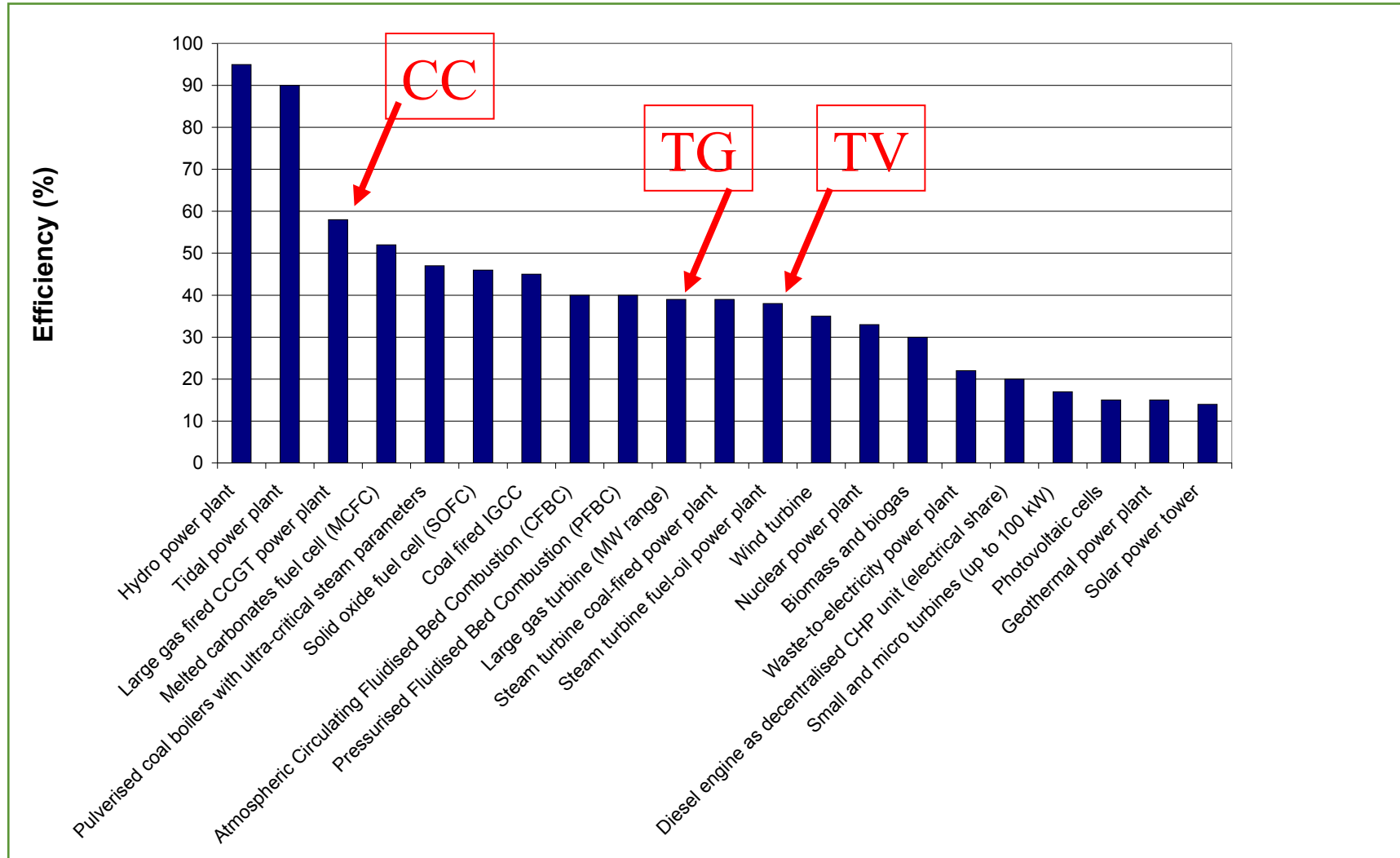
Turbina 1 <https://youtu.be/CgTCy71UkY0>

Turbina 2: <https://youtu.be/28O6scP21Dw>

Ciclo combinado: [https://youtu.be/0NFFvAUd\\_sc](https://youtu.be/0NFFvAUd_sc)



# Rendimientos comparados



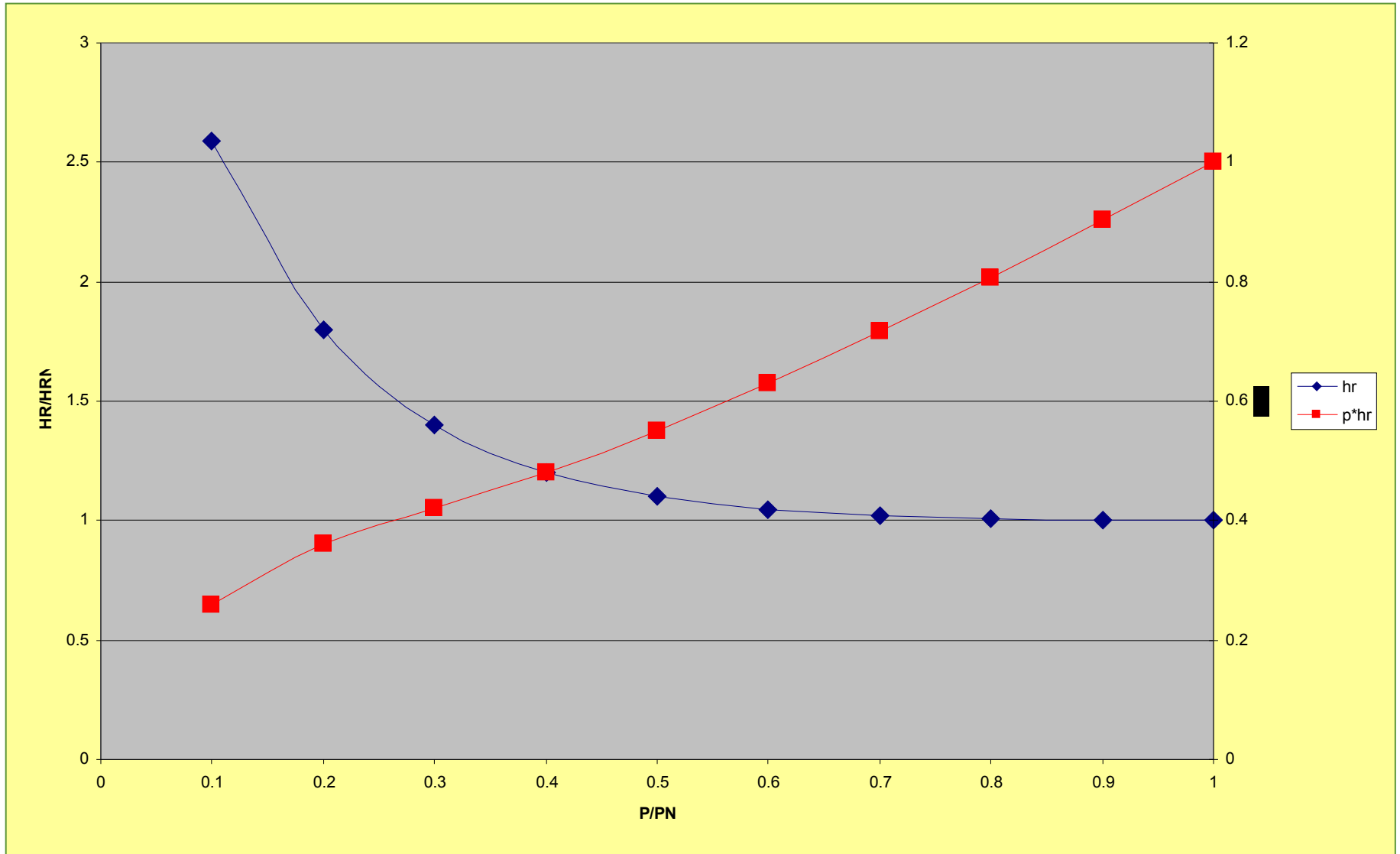
## Combustibles

FUENTE	DENSIDAD	PODER CALORIFICO INFERIOR		PODER CALORIFICO SUPERIOR		FACTOR DE CONVERSION A kep (sobre PCI)		
		Kg/lit	kcal/lit	kcal/kg	kcal/lit	kcal/kg	lit a kep	kg a kep
Carbón Mineral (Argentina) (*)	-	-	5900	-	6200	-	0,59	
Carbón Mineral (importado)	-	-	7200	-	7500	-	0,72	
Coque	-	-	6800	-	7500	-	0,68	
Petróleo Crudo	0,885	8850	10000	9293	10500	-	1	
Aeronaftas	0,709	7374	10400	8012	11300	0,7374	1,04	
Naftas	0,735	7607	10350	8232	11200	0,7607	1,035	
Kerosene y Comb Jets	0,808	8322	10300	8945	11070	0,8322	1,03	
Gas Oil	0,845	8619	10200	9211	10900	0,8619	1,02	
Diesel Oil	0,88	8800	10000	9416	10700	0,88	1	
Fuel Oil	0,945	9261	9800	9923	10500	0,9261	0,98	
Mezcla 70-30	0,91	8995	9885	9638	10591	0,8995	0,988	
Carbón Residual	1	-	7600	-	7900	-	0,76	
Coque de Carbón Residual	-	-	7200	-	7800	-	0,72	
Gas Residual de Petróleo	-	8500/m3	-	9000/m3	-	0.8500 de m3		
Gas Natural	-	8300/m3	-	9300/m3	-	0.8300 de m3		
Propano	0,508	5588	11000	6102	12013	-	1,1	
Butano	0,567	6180	10900	6735	11878	-	1,09	
Gas Licuado	0,537	-	10950	6418	11951	-	1,095	
Leña Blanda	-	-	1840	-	2940	-	0,184	
Leña Dura	-	-	2300	-	3500	-	0,23	
Carbón de Leña	-	-	6500	-	7500	-	0,65	
Marlo de Maíz	-	-	2300	-	3000	-	0,23	
Cáscara de Arroz	-	-	2300	-	3000	-	0,23	
Bagazo	-	-	1500	-	2000	-	0,15	
Aserrín	-	-	1800	-	1995	-	0,18	
Otros Residuos Vegetales	-	-	1760	-	2310	-	0,176	
Papeles	-	-	1620	-	1796	-	0,162	
Alcohol de Quemar	0,789	6080	-	6400	-	0,608		
Gas de Alto Horno de C de Leña	-	950/m3	-	1055/m3	-	0.095 de m3		
Gas de Alto Horno de C de Coque	-	800/m3	-	905/m3	-	0.080 de m3		
Etano	1,27	14413/m3	11350	15746	12399	1,4413	1,135	
Metanol	0,8	3818	4773	4345	5431	0,3818	0,477	
Etanol	0,794	5082	6400	5633	7092	0,5082	0,64	
Electricidad	-	-	860 kcal/kWh	-	-	-	-	0,086
Licor Negro	-	-	3600	-	-	-	0,36	
Cortezalchips de leña	-	-	4600	-	-	-	0,46	

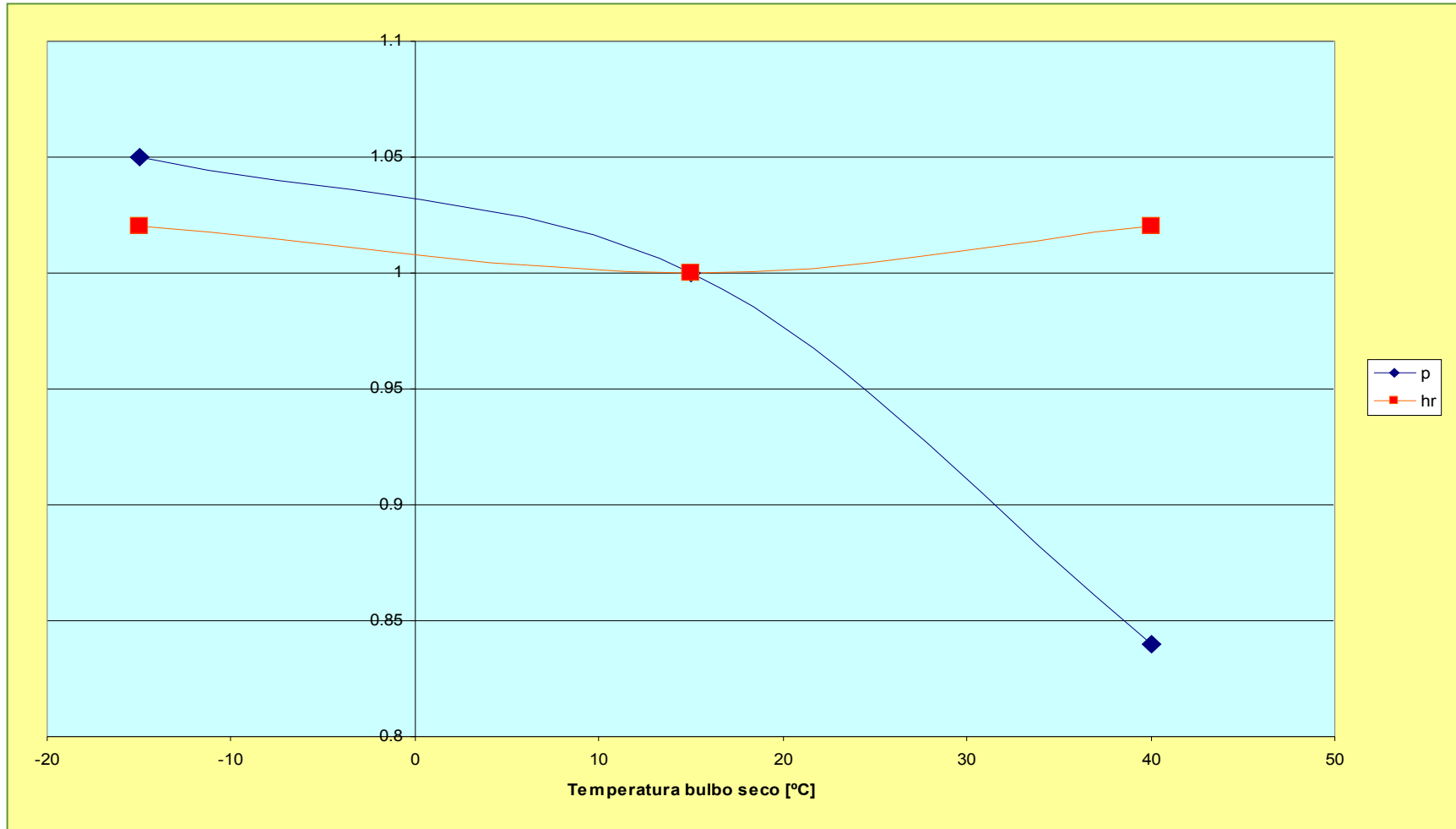
(\*) Sobre base húmeda

1 kep = kilo equivalente de petróleo = 10000 kcal

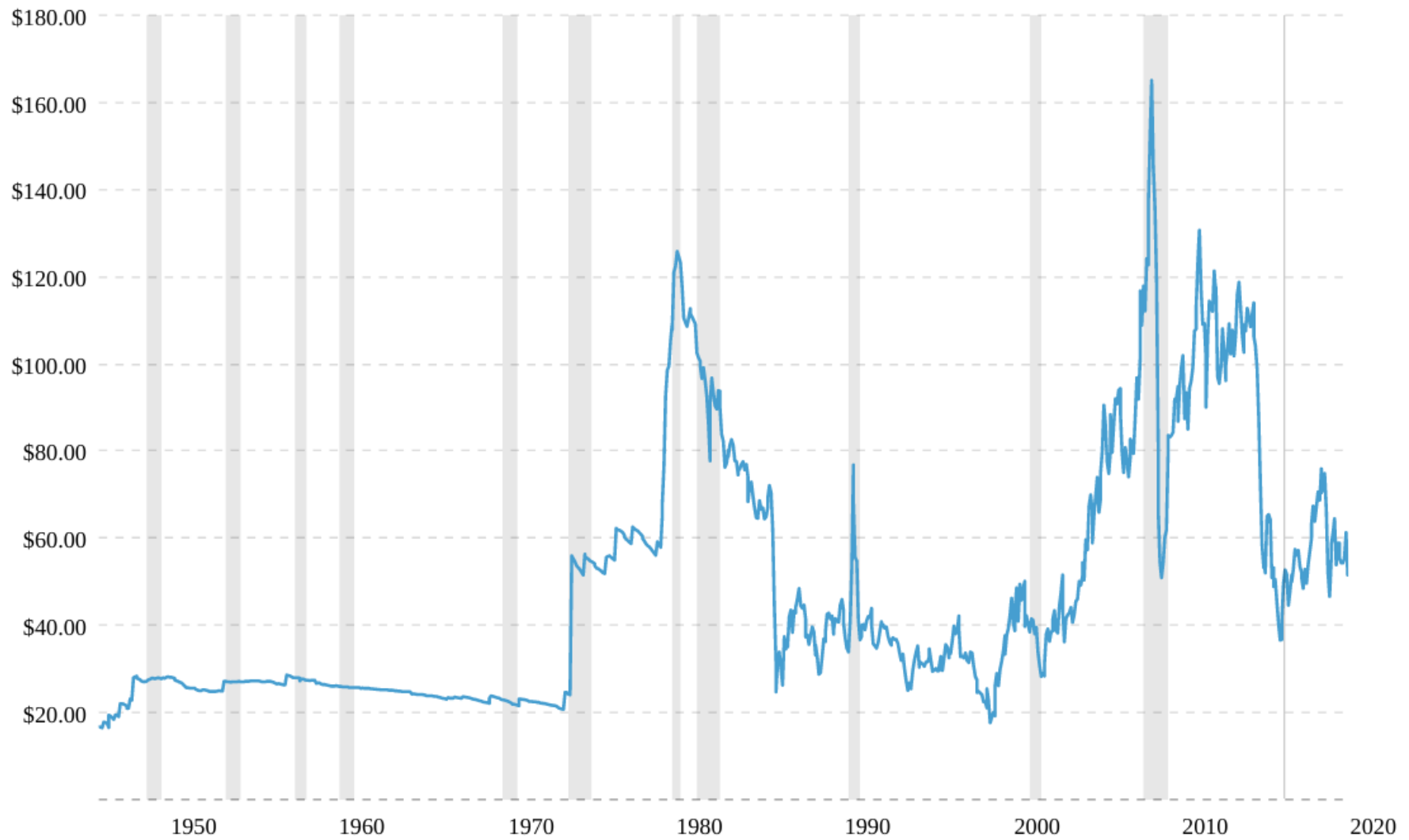
# TG Carga parcial



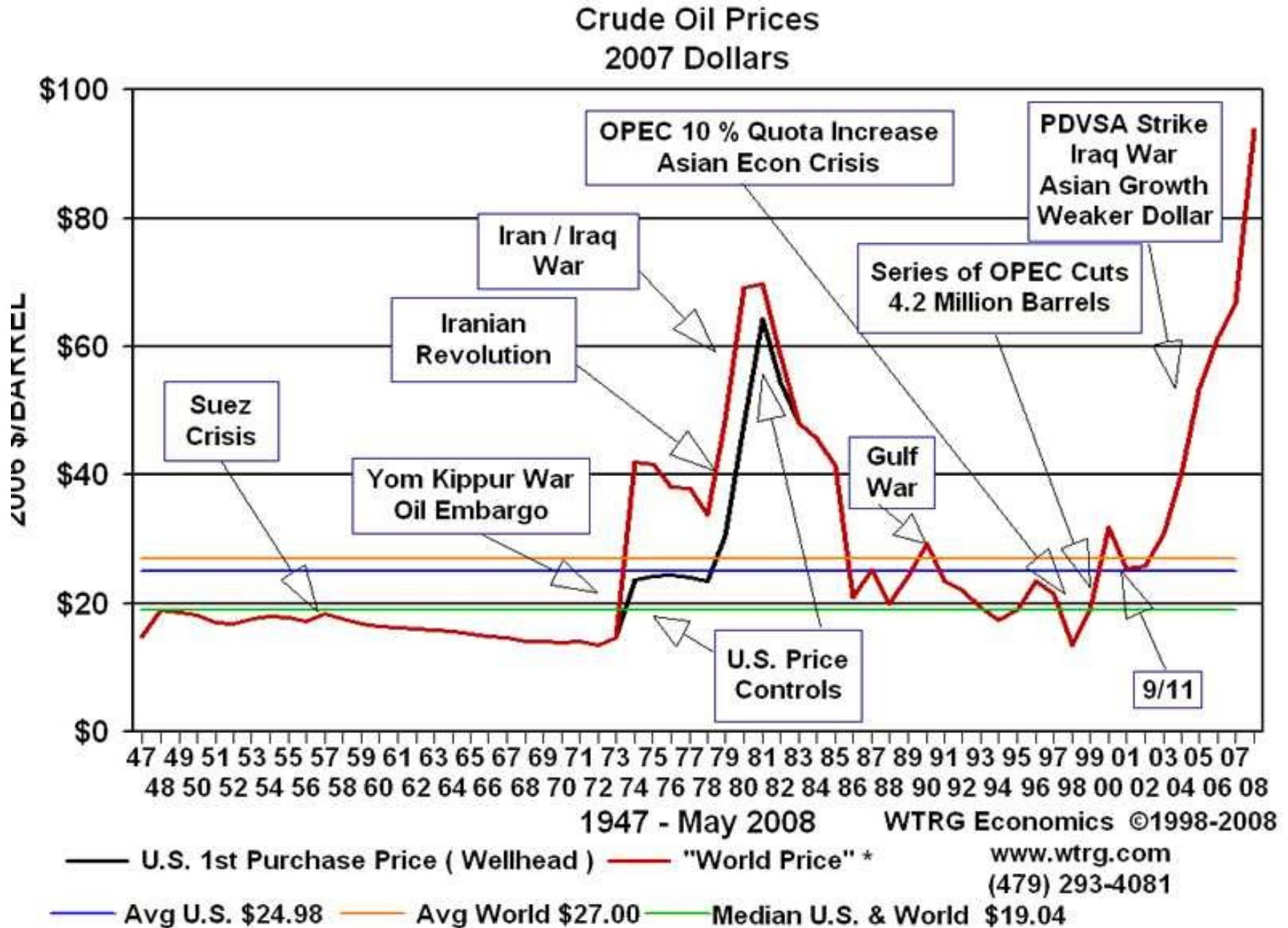
# TG Temp ambiente



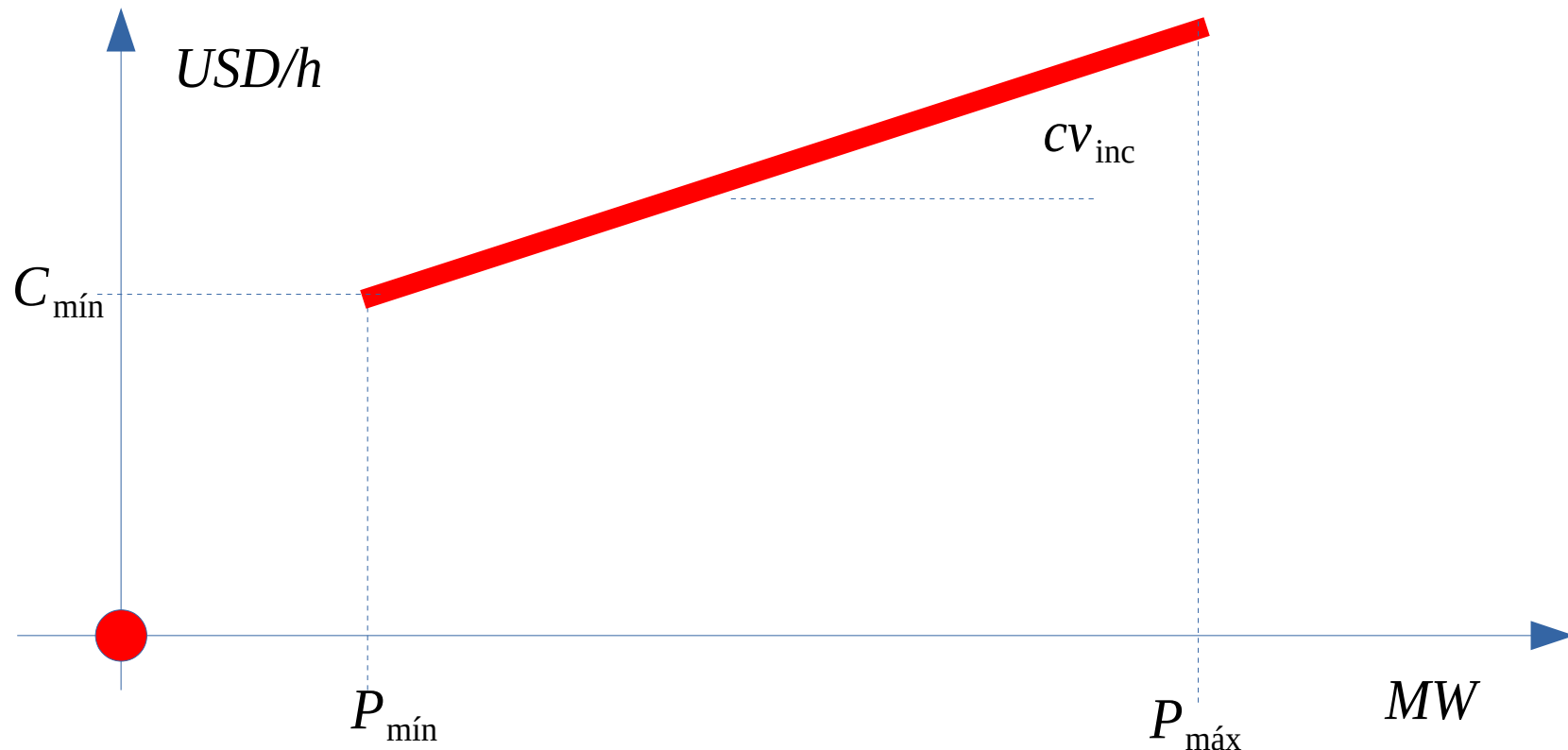
# WTI 1946-2020 (en USD/bbl de 2020)



# Precios del Petróleo 1998-2008.



# Modelo de máquina con mínimo técnico.



# Modelo de máquina con mínimo técnico. ON/OFF por paso

## VARIABLES DE CONTROL

$$A = \begin{cases} 1; ON \\ 0; OFF \end{cases}$$

$$B_j = (P_j - P_{mín})$$

$$0 \leq B_j \leq P_{máx} - P_{mín}$$

1+NPOSTES

## CARGADO DE LA MATRIZ

$$P_j = B_j + P_{mín} \cdot A$$

Aporte a las restricciones del nodo

$$B_j \leq (P_{máx} - P_{mín}) \cdot A$$

Restricciones adicionales (NPOSTES)

$$c = c_0 \cdot A + \sum_j c_v \cdot B_j$$

Aporte a la función de costo



# Modelo de máquina con mínimo técnico. ON/OFF por poste

## Variables de control

$$A_j = \begin{cases} 1; ON \\ 0; OFF \end{cases}$$

$$B_j = (P - P_{mín})$$

$$0 \leq B_j \leq P_{máx} - P_{mín}$$

2\*NPOSTES

## Cargado de la matriz

$$P_j = B_j + P_{mín} \cdot A_j$$

Aportes a las restricciones del nodo

$$B_j \leq (P_{máx} - P_{mín}) \cdot A_j$$

Restricciones adicionales (NPOSTES)

$$c = \sum_j co \cdot A_j + \sum_j cv \cdot B_j$$

Aporte a la función de costo

FIN