**MODELACIÓN MATEMÁTICA DEL DESPACHO ECONÓMICO CON RESTRICCIONES DE SEGURIDAD DE UN INTERVALO EN UN LENGUAJE DE PROGRAMACIÓN ALGEBRAICO (AMPL).**

GAR-MEM

Versión [1.0.0]

08-04-2019

**Revisión [0]**

Fecha de revisión: [08-04-2019]

**Elaborado por:**

Adriana González Urquiza (INEEL)

**Revisado por:**

Uriel Iram Lezama Lope (INEEL)

**Hoja de control de cambios**

|  |  |  |  |
| --- | --- | --- | --- |
| Número de versión | Fecha | Observaciones | Responsable |
| 1.0.0 | 08/04/2019 | Se genera documento |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

CONTENIDO

[1.0 INTRODUCCIÓN 5](#__RefHeading___Toc3307_1360854962)

[2.0 EL LENGUAJE DE PROGRAMACIÓN *AMPL* 5](#__RefHeading___Toc3309_1360854962)

[3.0 El DERS-I 5](#__RefHeading___Toc3311_1360854962)

[4.0 DESARROLLO 6](#__RefHeading___Toc13791_3162764200)

[4.1 Instrucciones de Instalación para Windows 6](#__RefHeading___Toc3315_1360854962)

[4.2 Escritura de modelos en AMPL 6](#__RefHeading___Toc3317_1360854962)

[4.3 Ejemplo en AMPL 6](#__RefHeading___Toc3298_3162764200)

[5.0 EL MODELO DEL DESPACHO ECONÓMICO CON RESTRICCIONES DE SEGURIDAD 11](#__RefHeading___Toc13793_3162764200)

[5.1 Observaciones sobre los códigos anteriores 17](#__RefHeading___Toc13795_3162764200)

[6.0 BIBLIOGRAFÍA 17](#__RefHeading___Toc3319_1360854962)

# INTRODUCCIÓN

Se presenta el procedimiento a seguir para la instalación del programa AMPL, y la forma de escribir modelos matemáticos de problemas de optimización; en particular, se detalla la elaboración del Despacho Económico con Restricciones de Seguridad de un intervalo (DERS-I) sin la parte de transmisión y pérdidas.

# EL LENGUAJE DE PROGRAMACIÓN *AMPL*

AMPL es un lenguaje de programación algebraica para describir y solucionar problemas de gran complejidad para computación matemática de gran escala. Es un lenguaje que se comunica con varios solvers tanto de código abierto como software comercial como CPLEX, FortMP, Gurobi, MINOS, IPOPT, SNOPT, KNITRO y LGO.

AMPL representa una gran ventaja a la hora de representar modelos matemáticos de optimización ya que la sintaxis del lenguaje tiene una sintaxis semejante a la notación matemática. Lo que permite una definición concisa y legible de problemas en el ámbito de optimización.

En AMPL se pueden programar diferentes tipos de problemas, entre ellos:

* Programación lineal y no lineal
* Programación cuadrática
* Programación entera mixta
* Programación no lineal entera mixta.

# El DERS-I

El Despacho Económico con Restricciones de Seguridad de un intervalo (DERS-I), es una de las funciones del Mercado de Tiempo Real. Tiene como objetivo determinar los puntos base económicos de las Unidades de Central Eléctrica que cuentan con infraestructura de Control Automático de Generación, así como los factores de participación de las Unidades que contribuyen a la regulación secundaria de frecuencia. Este modelo se ejecuta cada cinco minutos.

La demandad considerada por el DERS-I es la medición instantánea más reciente proveniente del estimador de estado. Por lo tanto, los puntos base calculados por el DERS-I son para dicho instante. Para el cálculo de los factores de participación económicos se considera un +-1% de cambio en la demanda nodal en los próximos 5 minutos.

La función objetivo del DERS-I es minimizar las ofertas de venta de energía de unidades térmicas, de rango continuo y discontinuo, las unidades hidroeléctricas y renovables intermitentes más los costos de penalizaciones por usar variables artificiales para evitar en el modelo un excedente de generación en los nodos del sistema, un corte de carga por demanda participante y violar límites en los grupos de ramas eléctricas.

# DESARROLLO

El programa está disponible en una versión libre para estudiantes, con la característica de que sólo soporta 500 variables y restricciones para problemas lineales, 300 para problemas no lineales y dependiendo el solver que se usa, puede ser menor el límite. Está disponible para los sistemas operativos: Windows, Linux y MacOS. Esta versión para estudiantes fue la utilizada para la elaboración de los modelos que se describen en las secciones siguientes de este documento. La versión del entorno de desarrollo utilizada para este trabajo es: AMPL IDE Version: 3.5.0.201802211231. Copyright (c) 2017 AMPL Optimization Inc. La versión de AMPL fue la

## Instrucciones de Instalación para Windows

Descargar la distribución [ampl.mswin32.zip](https://ampl.com/demo/ampl.mswin32.zip) para la versión de 32 bits o [ampl.mswin64.zip](https://ampl.com/demo/ampl.mswin64.zip) para la versión de 64 bits y para el sistema operativo Windows. Descomprimir los archivos contenidos en el archivo zip y extraer el la carpeta llamada ampl.mswin32 o ampl.mswin64 del archivo zip. Ésta será la carpeta AMPL, se puede renombrar y colocarlo en una localización conveniente.

Luego, para correr el programa, dentro de la carpeta AMPL, dar doble click en el archivo sw.exe, escribir ampl en la terminal que aparece. Entonces aparecerá la línea ampl: y se puede comenzar a escribir comandos de AMPL.

Opcionalmente se puede instalar un entorno de desarrollo (IDE) de ampl, llamado amplide; para instalarlo se deben descargar los archivos [amplide.mswin32.zip](https://ampl.com/demo/amplide.mswin32.zip) para la versión de 32 bits o [amplide.mswin64.zip](https://ampl.com/demo/amplide.mswin64.zip) para la versión de 64 bits. Descomprimir el archivo, extraer la carpeta llamada amplide.mswin32 (o amplide.mswin64). Para abrir la interfaz del entorno de desarrollo, dar doble click en el archivo amplide.exe que estará en la carpeta amplide.mswinX. Dentro de esta carpeta debe estar también el archivo ampl.exe.

## Escritura de modelos en AMPL

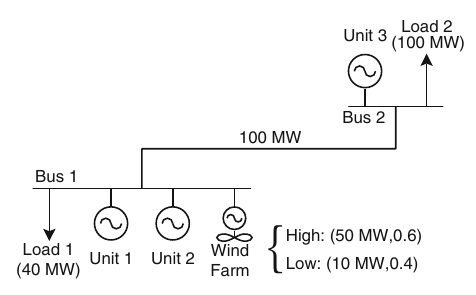
Para que un problema de optimización sea resuelto a través de alguno de los *solvers* que soporta AMPL, es necesario hacer un archivo con extensión .mod, y otro .dat. En el archivo .dat estarán definidos los conjuntos sobre los cuales corren las variables, y los parámetros. Se definen también los parámetros. Mientras que en el archivo .mod se menciona los conjuntos sobre los cuales se trabajará, se especifican los índices con los que estarán asociados los parámetros, posteriormente se definen las variables del modelo junto con el conjunto al que están asociados; finalmente se escribe la función objetivo del modelo, y después de ésta, las restricciones a las que está sujeto el problema.

## Ejemplo en AMPL

El siguiente ejemplo se presenta en el libro (4), página 65. Un sistema con dos nodos, como en la Ilustración 1, con un panorama estocástico ante la posibilidad de dos escenarios determinados por la potencia generada por el generador eólico.

El sistema contiene tres unidades térmicas, dos cargas y una planta eólica de 50MW, colocada en el primer nodo. La transmisión de la linea en el sistema tiene una reactancia por unidad de 0.13. Para un periodo dado en el futuro, el operador del sistema deberá determinar el despacho de energía y la capacidad de reserva que se necesita. La capacidad de reserva se requiere para cubrir la incertidumbre de la producción eólica, que está representada en dos escenarios, uno donde la generación es alta (de 50MW) y otro donde es baja (10MW), cada escenario con probabilidad 0.6 y 0.4, respectivamente. Las decisiones que el operador del sistema tiene que enfrentar son las siguientes:

1. Determinar los niveles de producción de las unidades térmicas y la cantidad y asignación de reservas para enfrentar la incertidumbre de la producción eólica de energía.
2. Desplegar reservas en la forma de energía de balance durante la operación de tiempo real del sistema de potencia para adaptar la producción real de potencia eólica.

  
Ilustración 1: Sistema de dos nodos, con tres unidades térmicas y una de viento.

Este modelo está detallado en la bibliografía, donde se definen los conjuntos de nodos, las unidades generadoras y los escenarios, así como el costo unitario por unidad, el costo de reserva de subida y de bajada, la generación máxima de cada unidad; la carga de cada nodo, la capacidad de transmisión de la línea, la penalización por pérdida de carga, y el pronóstico de generación por viento. Estos parámetros quedan establecidos en un archivo con extensión *dat.* A continuación se presenta el código del archivo correspondiente. Supongamos que lo guardamos con el nombre *dispatch.dat.*

**set** UNIDADES := U1 U2 U3 ;

**set** NODOS := N1 N2 ;

**set** ESCENARIOS := H L ;

**param**: MaxGen CostUnit CostResU CostResD :=

U1 50 10 16 15

U2 110 30 13 12

U3 100 35 10 9 ;

**param** CargaNodo :=

N1 40

N2 100 ;

**param** UbGen :

N1 N2 :=

U1 1 0

U2 1 0

U3 0 1 ;

**param** NodosAdyacentes :

N1 N2 :=

N1 0 1

N2 1 0 ;

**param** Reactancia :

N1 N2 :=

N1 0 0.13

N2 0.13 0 ;

**param** CapacTransmision :

N1 N2 :=

N1 0 100

N2 100 0 ;

**param** PerdidaCarga := 200 ;

**param** GenViento :=

N1 1

N2 0;

**param**: ProbEsce VientoMax :=

H 0.6 50

L 0.4 10 ;

**param** VientoPron :=

N1 40

N2 0 ;

Mientras, el modelo con los parámetros, las variables, la función objetivo y las restricciones quedan expresados en el siguiente código:

# conjuntos

**set** UNIDADES ;

**set** NODOS ;

**set** ESCENARIOS ;

# parametros

**param** MaxGen {UNIDADES};

**param** CostUnit {UNIDADES};

**param** CostResU {UNIDADES};

**param** CostResD {UNIDADES};

**param** UbGen {UNIDADES, NODOS};

**param** CargaNodo {NODOS};

**param** NodosAdyacentes {NODOS, NODOS};

**param** Reactancia {NODOS, NODOS};

**param** CapacTransmision {NODOS, NODOS};

**param** GenViento {NODOS};

**param** VientoPron {NODOS};

**param** ProbEsce {ESCENARIOS};

**param** VientoMax {ESCENARIOS};

**param** PerdidaCarga;

# variable no negativas

**var** P {UNIDADES} >= 0;

**var** Ru {UNIDADES} >= 0;

**var** Rd {UNIDADES} >= 0;

**var** ru {UNIDADES, ESCENARIOS} >= 0;

**var** rd {UNIDADES, ESCENARIOS} >= 0;

**var** Ws {ESCENARIOS} >= 0;

**var** Ls {NODOS, ESCENARIOS} >= 0;

**var** del0 {NODOS};

**var** del {NODOS, ESCENARIOS};

# funcion objetivo

**minimize** Costo\_Operacion : **sum** {u **in** UNIDADES} CostUnit[u]\*P[u] + **sum** {u **in** UNIDADES} CostResU[u]\*Ru[u] + **sum** {u **in** UNIDADES} CostResD[u]\*Rd[u] +

**sum** {s **in** ESCENARIOS} ProbEsce[s]\*(**sum** {u **in** UNIDADES} CostUnit[u]\*(ru[u,s]-rd[u,s]) + PerdidaCarga\***sum** {n **in** NODOS}Ls[n,s]);

# Restricciones de Operacion del Dia En Adelanto

**subject** **to** BalancePotencia {n **in** NODOS}: **sum** {u **in** UNIDADES} UbGen[u,n]\*P[u] + GenViento[n]\*VientoPron[n] - CargaNodo[n] == **sum** {m **in** NODOS} NodosAdyacentes[n,m]\*(del0[n]-del0[m])/0.13;

**subject** **to** NodoSetting: del0['N1'] == 0;

**subject** **to** TransmissionCapacity {n **in** NODOS}: **sum** {m **in** NODOS} NodosAdyacentes[n,m]\*(del0[n]-del0[m])/0.13 <= **sum** {m **in** NODOS} CapacTransmision[n,m];

**subject** **to** EnergiaReservaAjenaU {u **in** UNIDADES} : P[u] + Ru[u] <= MaxGen[u];

**subject** **to** EnergiaReservaAjenaD {u **in** UNIDADES} : P[u] - Rd[u] >= 0;

# Restricciones de Operacion

**subject** **to** BalanceDemandaGeneracion {n **in** NODOS, s **in** ESCENARIOS} : **sum** {u **in** UNIDADES} UbGen[u,n]\*(ru[u,s] - rd[u,s]) + Ls[n,s] + VientoMax[s] - VientoPron[n] - GenViento[n]\*Ws[s] == **sum** {m **in** NODOS} (del[n,s] - del0[n] + del0[m] - del[m,s])/0.13;

**subject** **to** PostBalanceTransCap {n **in** NODOS, s **in** ESCENARIOS}: **sum** {m **in** NODOS} NodosAdyacentes[n,m]\*(del[n,s]-del[m,s])/0.13 <= **sum** {m **in** NODOS} CapacTransmision[n,m];

**subject** **to** NodoSetting2 : del0['N1'] == 0;

**subject** **to** NodoSettingEsc {s **in** ESCENARIOS} : del['N1',s] == 0;

**subject** **to** WindProdCurtailed {s **in** ESCENARIOS} : Ws[s]<= VientoMax[s];

**subject** **to** BalanceCapReservaU {u **in** UNIDADES, s **in** ESCENARIOS} : ru[u,s] <= Ru[u];

**subject** **to** BalanceCapReservaD {u **in** UNIDADES, s **in** ESCENARIOS} : rd[u,s] <= Rd[u];

Esté código se guarda en un archivo de extensión *mod.* Supongamos que en este caso se llama *dispatch.mod.* Como se mencionó previamente, en el entorno de desarrollo de AMPL, se puede hacer la escritura de estos archivos. Ahí mismo se puede solicitar la solución al programa, mediante los siguientes comandos.

*ampl:* model dispatch.mod;

ampl: data dispatch.dat;

ampl: solve;

MINOS 5.51: optimal solution found.

17 iterations, objective 2130

Donde en la primera línea se solicita que lea el archivo .mod. Con la instrucción en la segunda línea lee el archivo .dat. Después de haber leído ambos archivos, el programa ya tiene los datos necesarios y el modelo para resolverlo; en la tercera línea se escribe el comando “solve;” para obtener la solución.

Enseguida se muestra en la interfaz la solución de este ejemplo. Indicando que se ha solucionado mediante el solver MINOS 5.51. Se encontró una solución óptima que arroja un costo de operación mínimo de $2130.

# EL MODELO DEL DESPACHO ECONÓMICO CON RESTRICCIONES DE SEGURIDAD

Con base en lo establecido en la Formulación Matemática del Modelo de Despacho Económico con Restricciones de Seguridad de Un Intervalo (2), se desarrollan los siguientes archivos que corresponden al modelo en el lenguaje AMPL. Primero el archivo de extensión dat:

##### CONJUNTOS DEL DERS-I

**set** SegOferta; # Segmentos de Ofertas/Requerimiento.

**set** RamasElect; # Grupos de ramas eléctricas.

**set** CentHidro; # Centrales Hidroeléctricas.

**set** PronDemanda; # Pronostico de demandas.

**set** ModoOperacionRD # Modos de operacion de unidades con rango discontinuo.

**set** NodosElec; # Nodos electricos del sistema.

**set** UnidadesDersiRC; # Unidades que participan en el DERS-I Rango Continuo.

**set** UnidadesDersiRD; # Unidades que participan en el DERS-I Rango Discontinuo.

**set** UnidadesDersiH; # Unidades que participan en el DERS-I Hidroelectricas.

**set** UnidadesDersiRE; # Unidades que participan en el DERS-I Renovables.

**set** UnidadesDersiNPR; # Unidades que participan en el DERS-I No Programables.

#set UnidadesDersi := UnidadesDersiRC union UnidadesDersiRD union UnidadesDersiH union UnidadesDersiRE union UnidadesDersiNPR;

##### CONSTANTES DEL DERS-I

**param** CD; # Costo de la energía no suministrada; en $/MWh.

**param** CS; # Penalizacion por exedente de generacion; en $/MWh.

**param** ContFluMax; # Limite maximo para el contraflujo en la rama electrica, en MW.

**param** FluMax; # Limite maximo para el flujo en la rama electrica, en MW.

**param** potenciaMinRD; # Potencia mnima operativa en Rango Discontinuo; en MW.

**param** potenciaMaxRD; # Potencia maxima operativa en Rango Discontinuo; en MW.

**param** potenciaMin; # Potencia minima operativa; en MW.

**param** potenciaMax; # Potencia maxima operativa; en MW.

**param** GB; # Potencia de venta de energia; en MW.

**param** GBRD;

**param** intercambioPot; # {IN}Intercambio de potencia activa, en MW.

**param** penVioFlujo; # Penalizacion por violacion en el flujo de los grupos de ramas electricas; en $/MW.

**param** costoOportH; # Costo de oportunidad para unidades hidroeléctricas basado en el calor del agua a corto plazo, en $/MWh.

**param** precioVentaEn; # Precio de oferta de venta de energía; en $/MWh.

**param** pronDemanda; # Pronóstico de demanda; en MW.

**param** resistenciaElec; # Resistencia electrica de la rama electrica.

**param** rampaBajada; # Rampa de bajada para operación; en MW/h.

**param** capGenRRegS; # Capacidad de generacion asignada para reserva de regulacion secundaria; en MW.

**param** capGenRRegSRD;

**param** capGenRRod10; # Capacidad de generacion asignada para reserva rodante de dies minutos, en MW.

**param** capGenRRod10RD;

**param** capGenRRodS; # Capacidad de generacion asignada para reserva rodante suplementaria; en MW.

**param** capGenRRodSRD;

**param** rampaRegSec; # Rampa para regulación secundaria en x minutos; en MW

**param** rampaSubida; # Rampa de subida para operacion; en MW/h

**param** sensFlujo; # Sensibilidad de flujo en la rama eléctrica con respecto a los cambios en las inyecciones de neta potencia en los nodos; adimensional.

**param** sensPerdida; # Sensibilidad de perdidas en el nodo con respecto a los cambios en las inyecciones netas de potencia en el nodo; adimensional.

Y a continuación el modelo matemático. Los números que se muestran entre paréntesis en las restricciones del modelo, corresponden al número de la ecuación en el manual Formulación Matemática del Modelo de Despacho Económico con Restricciones de Seguridad de Un Intervalo.

####### CONJUNTOS DEL DERS-I

# Nombre de Conjunto # {indice usado en la ecuacion} Descripcion del conjunto.

**set** SegOferta; # {b} Segmentos de Ofertas/Requerimiento.

**set** RamasElect; # {br} Grupos de ramas electricas.

**set** CentHidro; # {ci} Centrales Hidroelectricas.

**set** PronDemanda; # {d} Pronostico de demandas.

**set** ModoOperacionRD; # {m} Modos de operacion de unidades con rango discontinuo.

**set** NodosElec; # {n} Nodos electricos del sistema.

**set** UnidadesDersiRC; # {u} Unidades que participan en el DERS-I Rango Continuo.

**set** UnidadesDersiRD; # {u} Unidades que participan en el DERS-I Rango Discontinuo.

**set** UnidadesDersiH; # {u} Unidades que participan en el DERS-I Hidroelectricas.

**set** UnidadesDersiRE; # {u} Unidades que participan en el DERS-I Renovables.

**set** UnidadesDersiNPR; # {u} Unidades que participan en el DERS-I No Programables.

**set** UnidadesDersi := UnidadesDersiRC **union** UnidadesDersiRD **union** UnidadesDersiH **union** UnidadesDersiRE **union** UnidadesDersiNPR;

####### CONSTANTES DEL DERS-I

**param** CD; # {CD} Costo de la energia no suministrada; en $/MWh.

**param** CS; # {CS} Penalizacion por exedente de generacion; en $/MWh.

**param** ContFluMax {RamasElect};; # {\bar{fn}} Limite maximo para el contraflujo en la rama electrica, en MW.

**param** FluMax {RamasElect}; # {\bar{fp}} Limite maximo para el flujo en la rama electrica, en MW.

**param** potenciaMinRD {UnidadesDersiRD, ModoOperacionRD}; # {\underline{g}} Potencia minima operativa; en MW.

**param** potenciaMaxRD {UnidadesDersiRD, ModoOperacionRD}; # {\bar{g}} Potencia maxima operativa; en MW.

**param** potenciaMin {UnidadesDersi};

**param** potenciaMax {UnidadesDersi};

**param** GB {SegOferta, UnidadesDersi};

**param** GBRD {SegOferta, ModoOperacionRD, UnidadesDersiRD}; # {GB} Potencia de venta de energia; en MW.

**param** intercambioPot; # {IN} Intercambio de potencia activa, en MW.

**param** penVioFlujo; # {P^BR} Penalizacion por violacion en el flujo de los grupos de ramas electricas; en $/MW.

**param** costoOportH {UnidadesDersiH}; # {P^COP} Costo de oportunidad para unidades hidroelectricas basado en el calor del agua a corto plazo, en $/MWh.

**param** precioVentaEn {SegOferta, UnidadesDersi}; # {P^OVE} Precio de oferta de venta de energia; en $/MWh.

**param** pronDemanda {PronDemanda}; # {PD} Pronostico de demanda; en MW.

**param** resistenciaElec; # {R} Resistencia electrica de la rama electrica.

**param** rampaBajada; # {RB} Rampa de bajada para operación; en MW/h.

**param** capGenRRegS {UnidadesDersi};

**param** capGenRRegSRD {UnidadesDersi, ModoOperacionRD}; # {\hat{rre}} Capacidad de generacion asignada para reserva de regulacion secundaria; en MW.

**param** capGenRRod10 {UnidadesDersi};

**param** capGenRRod10RD {UnidadesDersi, ModoOperacionRD}; # {\hat{rro}^10} Capacidad de generacion asignada para reserva rodante de dies minutos, en MW.

**param** capGenRRodS {UnidadesDersi};

**param** capGenRRodSRD {UnidadesDersi, ModoOperacionRD}; # {\hat{rro}^S} Capacidad de generacion asignada para reserva rodante suplementaria; en MW.

**param** rampaRegSec; # {RR} Rampa para regulacion secundaria en x minutos; en MW

**param** rampaSubida; # {RS} Rampa de subida para operacion; en MW/h

**param** sensFlujo; # {sfbr} Sensibilidad de flujo en la rama electrica con respecto a los cambios en las inyecciones de neta potencia en los nodos; adimensional.

**param** sensPerdida; # {sper} Sensibilidad de perdidas en el nodo con respecto a los cambios en las inyecciones netas de potencia en el nodo; adimensional.

####### VARIABLES

**var** corteCarga {PronDemanda}; # {df} Corte de carga; en MW.

**var** excFlujoRamElec {RamasElect}; # {f+} Variable artificial de excedente de flujo en una rama electrica; en MW.

**var** excConFluRamElec {RamasElect}; # {f-} Variable artificial de excedente de contraflujo en una rama electrica; en MW.

**var** nivelGeneracion {UnidadesDersi}; # {g} Nivel de generacion total de venta de energia; en MW.

**var** genSegOferta {SegOferta, UnidadesDersi}; # {gb} Generacion del segmento de la oferta de venta de enrgia; en MW.

**var** artReqEneFija; # {gef} Variable artificial para satisfacer el requerimeinto de energia fija cuando hay un d�ficit o excendente; en MW/h

**var** inyecPotencia {NodosElec}; # {iny} Inyeccion neta de potencia en un nodo; en MW.

**var** perdidasIsla; # {Per} Perdidas en la isla electrica.

**var** excGeneracion {NodosElec}; # {Y} Excedente de generacion; en MW.

**minimize** Costo\_Total : (**sum** {u **in** UnidadesDersiRC} **sum** {b **in** SegOferta} precioVentaEn[b,u] \* genSegOferta[b,u]) +

(**sum** {u **in** UnidadesDersiRD} **sum** {b **in** SegOferta} precioVentaEn[b,u] \* genSegOferta[b,u]) +

(**sum** {u **in** UnidadesDersiH} costoOportH[u] \* nivelGeneracion[u]) +

(**sum** {u **in** UnidadesDersiRE} **sum** {b **in** SegOferta} precioVentaEn[b,u] \* genSegOferta[b,u]) + # las siguientes son costos de penalizaciones

(**sum** {n **in** NodosElec} CS \* excGeneracion[n]) + # penalizacion por excedente de generacion en los nodos

(**sum** {d **in** PronDemanda} CD \* corteCarga[d]) + # penalizacion por corte de carga por demanda participante

(**sum** {br **in** RamasElect} penVioFlujo \* (excFlujoRamElec[br] + excConFluRamElec[br])); # penalizacion por violar limites en los grupos de ramas electricas

################### RESTRICCIONES

#### UNIDADES CON RANGO DE OPERACION CONTINUO

## NIVEL DE GENERACION (ec 3)

**subject** **to** NivelGenRC {u **in** UnidadesDersiRC} : nivelGeneracion[u] = potenciaMin[u] + **sum**{b **in** SegOferta} genSegOferta[b,u];

## SEGMENTOS DE OFERTAS DE VENTA DE ENERGIA (ec 4)

**subject** **to** GenOfertaVentaRC {u **in** UnidadesDersiRC, b **in** SegOferta} : 0 <= genSegOferta[b,u] <= GB[b,u]; # Revisar que asi se escriba la doble desigualdad

## LIMITE MAXIMO DE GENERACION OPERATIVO (ec 5)

**subject** **to** LimMaxGenOpRC {u **in** UnidadesDersiRC} : nivelGeneracion[u] + capGenRRegS[u] + capGenRRod10[u] + capGenRRodS[u] <= potenciaMax[u];

## LIMITE MINIMO DE GENERACION OPERATIVO (ec 6)

**subject** **to** LimMinGenOpRC {u **in** UnidadesDersiRC} : nivelGeneracion[u] - capGenRRegS[u] >= potenciaMin[u];

######## UNIDADES CON RANGO DE OPERACION DISCONTINUO

## NIVEL DE GENERACION (ec 7)

**subject** **to** NivelGenRD {u **in** UnidadesDersiRD, m **in** ModoOperacionRD} : nivelGeneracion[u] = potenciaMinRD[u,m] + **sum**{b **in** SegOferta} genSegOferta[b,u]; # Preguntar qu� onda con los modos. Como estan en los archivos?

## SEGMENTOS DE OFERTAS DE VENTA DE ENERGIA (ec 8)

**subject** **to** GenOfertaVentaRD {u **in** UnidadesDersiRD, m **in** ModoOperacionRD, b **in** SegOferta} : 0 <= genSegOferta[b,u] <= GBRD[b,m,u];

## LIMITE MAXIMO DE GENERACION OPERATIVO (ec 9)

**subject** **to** LimMaxGenOpRD {u **in** UnidadesDersiRD, m **in** ModoOperacionRD} : nivelGeneracion[u] + capGenRRegSRD[u,m] + capGenRRod10RD[u,m] + capGenRRodSRD[u, m] <= potenciaMaxRD[u,m];

## LIMITE MINIMO DE GENERACION OPERATIVO (ec 10)

**subject** **to** LimMinGenOpRD {u **in** UnidadesDersiRD, m **in** ModoOperacionRD} : nivelGeneracion[u] - capGenRRegSRD[u,m] >= potenciaMinRD[u,m];

######## UNIDADES DE CENTRALES HIDROELECTRICAS

## OFERTA DE VENTA DE ENERGIA

# Unidades hidroeléctricas que pertenecen a un mismo embalse

# ofrecen un mismo precio de oportunidad para la generación

# de energia.

## LIMITE MAXIMO DE GENERACION OPERATIVO (ec 11)

**subject** **to** LimMaxGenOpH {u **in** UnidadesDersiH} : nivelGeneracion[u] + capGenRRod10[u] + capGenRRodS[u] + capGenRRegS[u] <= potenciaMax[u];

## LIMITE MINIMO DE GENERACION OPERATIVO (ec 12)

**subject** **to** LimMinGenOpH {u **in** UnidadesDersiH} : nivelGeneracion[u] - capGenRRegS[u] >= potenciaMin[u];

######## UNIDADES RENOVABLES INTERMITENTES

## NIVEL DE GENERACION (ec 13)

**subject** **to** NivelGenRE {u **in** UnidadesDersiRE} : nivelGeneracion[u] = **sum** {b **in** SegOferta} genSegOferta[b,u];

## SEGMENTOS DE OFERTA DE VENTA DE ENERGIA (ec 14)

**subject** **to** SegOfertasVentEner {u **in** UnidadesDersiRE, b **in** SegOferta} : 0 <= genSegOferta[b,u] <= GB[u,b];

## LIMITE MAXIMO DE GENERACION OPERATIVO (ec 15)

**subject** **to** LimMaxGenOpRE {u **in** UnidadesDersiRE} : nivelGeneracion[u] <= potenciaMax[u];

## LIMITE MINIMO DE GENERACION OPERATIVO (ec 16)

**subject** **to** LimMinGenOpRE {u **in** UnidadesDersiRE} : nivelGeneracion[u] >= potenciaMin[u];

######## DEMANDA

## NIVEL DE DEMANDA

# El pronostico de la demanda utilizada en el DERS-MI para los quince minutos es

# remplazado en el DERS-I por el valor de la demanda proveniente del estimador de estado.

#### RESTRICCIONES DEL SISTEMA ELECTRICO

## BALANCE DE POTENCIA (ec 18)

**subject** **to** BalancePotencia : **sum** {n **in** NodosElec} inyecPotencia[n] + perdidasIsla = 0;

## REPRESENTACION TANGENCIAL DE PERDIDAS DE TRANSMISION

## LIMITE SOBRE FLUJOS EN GRUPOS DE RAMAS ELECTRICAS

## COTAS SIMPLES

**subject** **to** FlujoRamas {br **in** RamasElect} : 0 <= excFlujoRamElec[br] <= FluMax[br]; # (ec 25)

**subject** **to** ContraFlujoRamas {br **in** RamasElect} : 0<= excConFluRamElec[br] <= ContFluMax[br]; # (ec 26)

# subject to X >= 0; # (ec 27)

**subject** **to** ExcGeneracion {n **in** NodosElec}: excGeneracion[n] >= 0; # (ec 28)

## 5.1 Observaciones sobre los códigos anteriores

* El código del DERS-I se hizo sin considerar las pérdidas de transmisión, ni los limites de flujo sobre las ramas eléctricas.
* El código se escribió sin acentos.
* Los nombres de los conjuntos, parámetros, variables y restricciones, se eligieron de forma que fueran explicativos. No corresponden a los nombres actuales del Modelo DERS-I que está programado en Fortran. En el archivo dersi.dat se explica a que parámetro o variable corresponde en la referencia (2) de la bibliografía.
* Las ecuaciones 25 y 26 del documento (4) hacen referencia a un conjunto de índices I, sin que se explique previamente de qué se trata ese conjunto.

# CONCLUSIONES

El modelo que se presenta del DERS-I está listo con los conjuntos, parámetros y variables que se muestran. Hace falta integrar al modelo la parte de transmisión y flujos sobre las ramas eléctricas.

Para implementar el modelo, de manera que sirva para el actual Mercado Eléctrico Mayorista, es hacer que lea archivos, o se conecte a una base de datos. Para lo que hay diferentes opciones. Una opción es a través de un lenguaje gestor del modelo, hacer que se conecte a una base de datos o que lea archivos; otra posibilidad es que suceda desde el mismo programa AMPL, como se describe en el Capítulo 10 del libro (5) señalado en la sección de bibliografía de este documento.

AMPL API, provee una librería orientada a objetos que se puede llamar para poder accesar a los modelos de AMPL y correr los comandos desde programas externos. Los resultados pueden ser intercambiados directamente entre las estructuras de datos en lenguajes externos y los modelos de AMPL. Los lenguajes para los que existe el API son C++, C#, Java, MATLAB, R y Python.

En particular, la API para Pyhton se llama amplpy. Se instala desde la interfaz de desarrollo de Python mediante los siguientes comandos:

python -m pip install pip --upgrade

python -m pip install amplpy.

Con esta herramienta es posible construir un programa del DERS-I con todas las necesidades del Mercado Eléctrico.

# **BIBLIOGRAFÍA**

1. Aquino Bolaños, Luisa, Ávila Rosales, Rene. **El Sistema del Mercado Eléctrico Mayorista en tiempo real.** Boletín del Instituto Nacional de Electricidad y Energías Límpas.
2. Alvarez Lopez, Juan; Ceciliano Meza, José Luis; De la Torre Sánchez, Armando; **Formulación Matemática del Modelo de Despacho Económico con Restricciones de Seguridad de Un Intervalo**. Julio 2015. GAR-DERS-I-FOR.
3. Página wed de AMPL: <https://ampl.com/>
4. Morales, J.M.; Conejo, A.J.; Madsen, H. et al. Integrating Renewables in Electricity Markets. Operational Problems. Springer, Vol. 205.
5. R. Fourer, D. M. Gay, B. W. Kernighan. **AMPL. A Modeling Language for Mathematical Programming.** Duxbury Thomson.

# 