

Nombres: José Wisner Urrego Portilla **Código:** 2151549
 Carlos Enrique Delgado Rico 2151547

Grupo: G52

Asignatura: Investigación de Operaciones 2

OBJETIVO DEL TRABAJO

Consolidar los conocimientos adquiridos hasta el momento en la aplicación de criterios de toma de decisiones en esquemas de rentabilidad o costos, por medio del desarrollo de un aplicativo informático que permita generar, a partir de una matriz de alternativas/estados, las mejores alternativas de acuerdo a cada criterios de decisión.

DESCRIPCIÓN DEL APLICATIVO INFORMÁTICO

El aplicativo fue desarrollado con base en tecnologías del momento ([HTML5](#), [CSS3](#), [Javascript](#)) que permitan su disponibilidad en web (lo que permite su uso en múltiples plataformas) y la capacidad de ser visualizado de forma cómoda en dispositivos móviles (diseño responsivo). Adicionalmente, la aplicación fue publicada en un servidor del servicio [Microsoft Azure](#) licenciado para estudiantes por [DreamSpark](#) para su disponibilidad 24x7x365. Para visitar el aplicativo [Pulse Aquí](#).

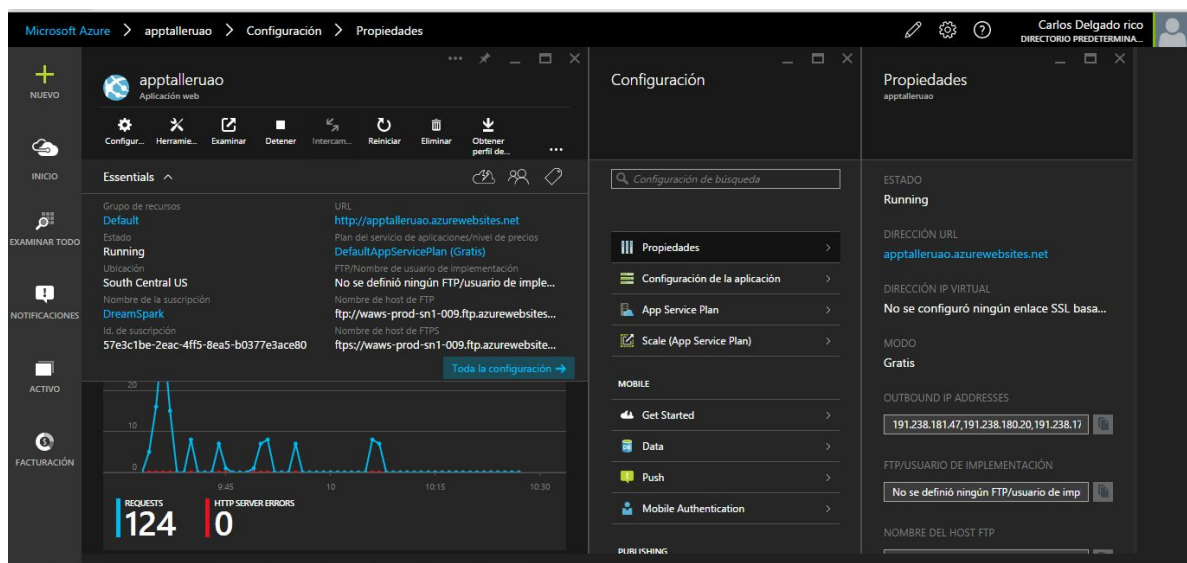


Imagen 1. Consola de administración del servicio de Windows Azure.

Reto Investigación de Operaciones 2									
Alternativas	4	Estados	4	Esquema	Rentabilidad	α	0.6	K	1
Generar Tabla Resolver Riesgo Resolver Incertidumbre									
Alternativas	Estado 1		Estado 2		Estado 3		Estado 4		
Alternativa 1	0,0		0,0		0,0		0,0		
Alternativa 2	0,0		0,0		0,0		0,0		
Alternativa 3	0,0		0,0		0,0		0,0		
Alternativa 4	0,0		0,0		0,0		0,0		
Probabilidades	0,0		0,0		0,0		0,0		

Resultados Riesgo	
Criterio	Alternativas
Valor Esperado	
Dispersión	
Max Posibilidad	

Resultados Incertidumbre	
Criterio	Alternativas
Wald	
Laplace	
Hurwicz	

Power By Jose Wisner Urrego and Carlos Enrique Delgado

Imagen 2. Vista de la aplicación ejecutándose en un navegador web en plataforma Windows 8

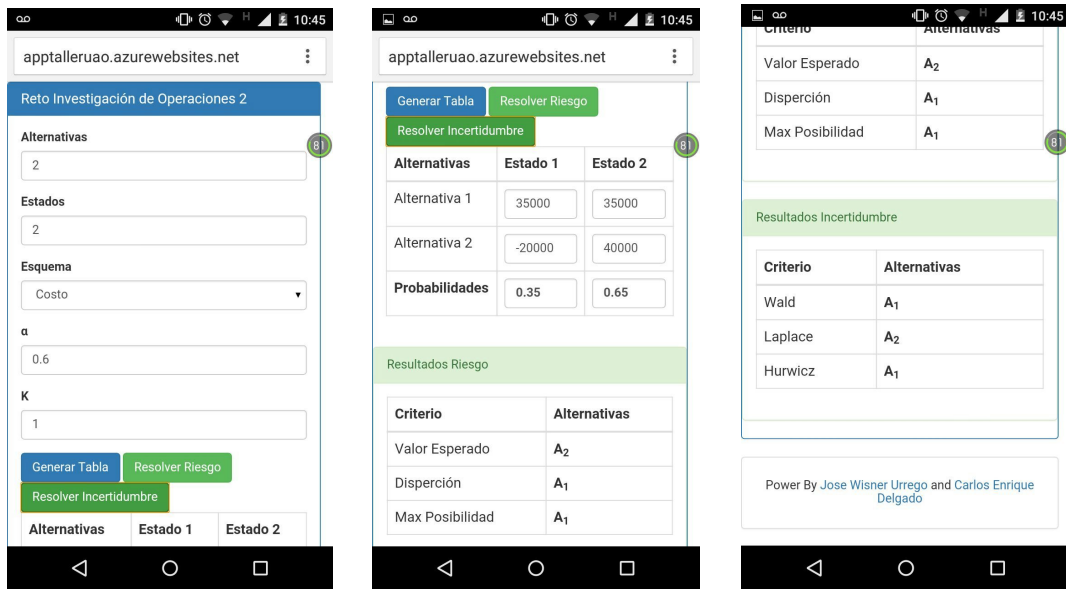


Imagen 3. Vista de la aplicación ejecutándose en un móvil con sistema operativo Android 5.1

EJEMPLO DE APLICACIÓN PARA EL DESARROLLO DEL RETO

Ejemplo 5 - Taller 1. Considere la siguiente Matriz de Pagos (Beneficios)

Alternativas	Estados de la naturaleza				
	E1	E2	E3	E4	E5
A1	15	10	0	-6	17
A2	3	14	8	9	2
A3	1	5	14	20	-3
A4	7	19	10	2	0
Probabilidades	0,05	0,20	0,15	0,25	0,35

1. Incertidumbre:

- Wald
- Laplace
- Hurwicz (Suponga Alpha = 0,60)

2. Riesgo:

- Valor Esperado
- Máxima probabilidad
- Dispersión (Suponga K = 1)

PASOS PARA LA SOLUCIÓN DEL EJEMPLO DE APLICACIÓN

PASO 1. PARAMETRIZACIÓN DEL APLICATIVO: Para poder hacer uso del aplicativo lo primero es realizar la parametrización inicial que comprende la siguiente información: número de alternativas, número de estados de la naturaleza, un valor supuesto para K, el esquema que se va a usar (costo o rentabilidad) y un valor supuesto para Alpha. Se puede observar cómo deben de ser diligenciados estos datos en la siguiente imagen:

Reto Investigación de Operaciones 2

Alternativas 4 Estados 5 Esquema Rentabilidad α 0.6 K 1

Generar Tabla Resolver Riesgo Resolver Incertidumbre

Alternativas	Estado 1	Estado 2	Estado 3	Estado 4
Alternativa 1	0	0	0	0
Alternativa 2	0	0	0	0
Alternativa 3	0	0	0	0
Alternativa 4	0	0	0	0
Probabilidades	0	0	0	0

Resultados Riesgo

Criterio	Alternativas
Valor Esperado	

Resultados Incertidumbre

Criterio	Alternativas
Wald	

Después de diligenciar se debe dar clic en el botón “Generar Tabla”, esto hará que la aplicación genere una matriz con el número de alternativas y estados digitados por el usuario. En la siguiente imagen se muestra como es la tabla resultante (con los datos del ejemplo de aplicación respectivamente diligenciados):

Reto Investigación de Operaciones 2 - Mozilla Firefox

Reto Investigación de Operaciones 2

Alternativas 4 Estados 5 Esquema Rentabilidad α 0.6 K 1

Generar Tabla Resolver Riesgo Resolver Incertidumbre

Alternativas	Estado 1	Estado 2	Estado 3	Estado 4	Estado 5
Alternativa 1	15	10	0	-6	17
Alternativa 2	3	14	8	9	2
Alternativa 3	1	5	14	20	-4
Alternativa 4	7	19	10	2	0
Probabilidades	0,05	0,20	0,15	0.25	0.35

Resultados Riesgo

Criterio	Alternativas
Valor Esperado	

Resultados Incertidumbre

Criterio	Alternativas
Wald	

PASO 2. ELEGIR EL TIPO DE CRITERIOS QUE SE DESEAN EVALUAR: A continuación, el usuario deberá elegir el tipo de criterios que desea evaluar, ya sea criterios de incertidumbre, riesgo o ambos; se debe tener en cuenta que para poder aplicar los criterios de riesgo la aplicación exige que sean diligenciados las probabilidades respectivas. Para elegir los criterios simplemente se debe presionar el botón dependiendo del criterio como se ven en las siguientes imágenes:

Reto Investigación de Operaciones 2

Alternativas: 4 Estados: 5 Esquema: Rentabilidad α : 0.6 K: 1

Generar Tabla Resolver Riesgo Resolver Incertidumbre

Alternativas	Estado 1	Estado 2	Estado 3	Estado 4	Estado 5
Alternativa 1	15	10	0	-6	17
Alternativa 2	3	14	8	9	2
Alternativa 3	1	5	14	20	-4
Alternativa 4	7	19	10	2	0
Probabilidades	0,05	0,20	0,15	0,25	0,35

Resultados Riesgo

Criterio	Alternativas
Valor Esperado	

Resultados Incertidumbre

Criterio	Alternativas
Wald	

Después de hacer clic en alguna de las opciones, los resultados se verán reflejados en la parte inferior de la aplicación en sus respectivas tablas “Resultados Riesgo” y “Resultados Incertidumbre”.

Reto Investigación de Operaciones 2

Alternativa 2: 3, 14, 8, 9, 2

Alternativa 3: 1, 5, 14, 20, -4

Alternativa 4: 7, 19, 10, 2, 0

Probabilidades: 0,05, 0,20, 0,15, 0,25, 0,35

Resultados Riesgo

Criterio	Alternativas
Valor Esperado	A ₁
Dispersión	A ₂
Max Posibilidad	A ₁

Resultados Incertidumbre

Criterio	Alternativas
Wald	A ₂
Laplace	A ₄
Hurwicz	A ₄

Power By Jose Wisner Urrego and Carlos Enrique Delgado

El aplicativo mostrará la mejor alternativa dependiendo del criterio que se use (en caso de empate se mostrarán las mejores alternativas), la toma de decisiones de cuál será la alternativa elegida depende del usuario quien debe usar la herramienta como insumo y no como un determinante de qué alternativa debe tomar, pues lo ultimo tambien depende de si el problema se desea abordar desde enfoque es pesimista, optimista o conservador.

La aplicación de los criterios de toma de decisión al problema y a su implementación en el código de programación serán explicados a continuación:

I. CRITERIO DE WALD

```
1 function wald(param_json){
2   var matriz = param_json.matriz;
3   var esquema = param_json.esquema;
4   var minimos = [];
5   var minimo = 0;
6   var maximos = [];
7   var maximo = 0;
8   if(esquema == 'rentabilidad'){ // se valida si el problema se debe trabajar sobre un esquema de rentabilidad, entonces
9     for(var i in matriz){ // se recorre la matriz
10      minimos.push(Math.min.apply(Math, matriz[i])); // se obtienen los mínimos de cada alternativa
11    }
12    maximo = Math.max.apply(Math, minimos); // después de haber obtenido los mínimos de cada i, se procede a obtener el máximo
13    var proc = contar_repetidos(minimos, maximo); // se usa el método "contar_repetidos" para saber si hay algún empate en las alternativas
14    var mensaje = "";
15    for(var i in proc.posiciones){
16      mensaje += "<b>A<sub>" + (parseInt(proc.posiciones[i]) + 1) + "</sub></b> "; // se construye el mensaje de retorno con la(s) mejor(es) alternativas
17    }
18    return mensaje; // se retorna el mensaje construido anteriormente
19  } else if(esquema == 'costo'){ // se valida si el problema se debe trabajar sobre un esquema de rentabilidad, entonces
20    for(var i in matriz){ // se obtienen los máximos de cada alternativa
21      maximos.push(Math.max.apply(Math, matriz[i]));
22    }
23    minimo = Math.min.apply(Math, maximos); // después de haber obtenido los máximos de cada i, se procede a obtener el máximo
24    var proc = contar_repetidos(maximos, minimo); // se usa el método "contar_repetidos" para saber si hay algún empate en las alternativas
25    var mensaje = "";
26    for(var i in proc.posiciones){
27      mensaje += "<b>A<sub>" + (parseInt(proc.posiciones[i]) + 1) + "</sub></b> "; // se construye el mensaje de retorno con la(s) mejor(es) alternativas
28    }
29    return mensaje; // se retorna el mensaje construido anteriormente
30  }
31 }
32 }
```

Para el **criterio de Wald** primero se debe diferenciar el esquema sobre el que se trabajará el problema, ya sea costos o rentabilidad; en el caso de que se desee trabajar bajo un esquema de rentabilidad, primero se deben obtener los mínimos de cada alternativa y luego de estos se debe tomar el(los) máximo(s). En el caso de un esquema de costos, primero se deben obtener los máximos y posteriormente, de estos, obtener el(los) mínimo(s).

II. CRITERIO DE LAPLACE

```
53 function laplace(param_json){
54   var matriz = param_json.matriz;
55   var esquema = param_json.esquema;
56   var n = matriz[0].length;
57   var probabilidad = 1/n;
58   var matriz_sum = [];
59   for(var j in matriz){ // se realiza el recorrido de la matriz para realizar una sumatoria
60     var sum_tmp = 0;
61     for(var i in matriz[j]){ // se recorren los valores de cada alternativa
62       sum_tmp += probabilidad * matriz[j][i]; // se realiza la sumatoria de los resultados del producto entre probabilidad y el valor de la alternativa
63     }
64     matriz_sum.push(sum_tmp); // se crea un nuevo vector con los valores de cada sumatoria
65   }
66   if(esquema == 'rentabilidad'){ // se valida si se trabajará un esquema de valores de rentabilidad
67     var maximo = Math.max.apply(Math, matriz_sum); // se obtiene el máximo valor de los resultados presentes en el vector que contiene la sumatoria
68     var proc = contar_repetidos(matriz_sum, maximo); // se usa el método "contar_repetidos" para saber si hay algún empate en las alternativas
69     var mensaje = "";
70     for(var i in proc.posiciones){
71       mensaje += "<b>A<sub>" + (parseInt(proc.posiciones[i]) + 1) + "</sub></b> "; // se construye el mensaje de retorno con la(s) mejor(es) alternativas
72     }
73     return mensaje; // se retorna el mensaje construido anteriormente
74   } else if(esquema == 'costo'){ // se valida si se trabajará un esquema de valores de costos
75     var minimo = Math.min.apply(Math, matriz_sum); // se obtiene el mínimo valor de los resultados presentes en el vector que contiene la sumatoria
76     var proc = contar_repetidos(matriz_sum, minimo); // se construye el mensaje de retorno con la(s) mejor(es) alternativas
77     var mensaje = "";
78     for(var i in proc.posiciones){
79       mensaje += "<b>A<sub>" + (parseInt(proc.posiciones[i]) + 1) + "</sub></b> "; // se construye el mensaje de retorno con la(s) mejor(es) alternativas
80     }
81     return mensaje; // se retorna el mensaje construido anteriormente
82   }
83 }
84 }
```

Para el **criterio de Laplace** se debe diferenciar el esquema sobre el que se trabajará el problema, ya sea de costos o rentabilidad, en el caso de que se desee trabajar bajo un esquema de rentabilidad, se realiza la sumatoria de los resultados del producto entre la probabilidad de cada estado de la naturaleza con el valor de la alternativa en este estado, estos resultados conformarán una nueva columna (vector en el código de programación); después, se debe elegir el máximo valor de los que conforman la columna con los resultados de las sumatorias. En caso de trabajar bajo un esquema de costos, el proceso es el mismo pero en vez de elegir el máximo se debe elegir mínimo valor de los que conforman la columna con los valores de las sumatorias.

III. CRITERIO DE HURWICZ

```
33 ▼ function hurwicz(param_json){
34   var matriz = param_json.matriz;
35   var esquema = param_json.esquema;
36   var alpha = param_json.alpha;
37   var vector_pond = [];
38   ▼ for(var j in matriz){
39     var maximo = Math.max.apply(Math, matriz[j]); // se obtiene el máximo valor de cada alternativa
40     var minimo = Math.min.apply(Math, matriz[j]); // se obtiene el mínimo valor de cada alternativa
41     var sum_pond_tmp = (alpha * maximo) + ((1 - alpha) * minimo); // se realiza el poderado de cada alternativa, se realiza el producto entre el máximo y el alpha, y el producto
42                                     // entre el mínimo y 1 - alpha
43     vector_pond.push(sum_pond_tmp); // se crea un vector con los resultados de cada ponderado
44   }
45   var maximo_hwc = Math.max.apply(Math, vector_pond); // se obtiene el máximo valor de los que se encuentran en el vector de los ponderados
46   var proc = contar_repetidos(vector_pond, maximo_hwc); // se usa el método "contar_repetidos" para saber si hay algún empate en las alternativas
47   var mensaje = "";
48   ▼ for(var i in proc.posiciones){
49     mensaje += "<b>A<sub>" + (parseInt(proc.posiciones[i]) + 1) + "</sub></b> "; // se construye el mensaje de retorno con la(s) mejor(es) alternativas
50   }
51   return mensaje; // se retorna el mensaje construido anteriormente
52 }
53
```

Para el **criterio de Hurmicz** se debe establecer un coeficiente de optimismo “alpha”, para obtener la alternativa óptima primero se obtienen los máximos y los mínimos de cada alternativa y posteriormente se realiza el producto entre el alpha y el máximo, sumándolo con el producto entre el mínimo y (1 - alpha); estos valores confirman una nueva columna (vector), de la cual se debe obtener el máximo valor que corresponderá a la mejor alternativa.

IV. VALOR ESPERADO

```
1
2 /*Función que obtiene matriz con los valores ingresados y resuelve por el criterio de valor esperado */
3 ▼ function valor_esperado(param_json){
4   var matriz = param_json.matriz; //Obtiene matriz con valores
5   var esquema = param_json.esquema; //Obtiene Esquema (rentabilidad o costo)
6   var v_probabilidades = param_json.probabilidades; //Obtiene Matriz de Probabilidades
7
8   var maximo = 0;
9   var minimo = 0;
10  var v_esperado = [];
11
12  //recorre la matriz por filas
13  ▼ for(var i in matriz){
14    var vn_valor = 0;
15    var va_filas = matriz[i]; //genera array con los valores de la fila
16    //recorre cada fila para generar el valor esperado
17    ▼ for(var j in va_filas){
18      vn_valor += (va_filas[j] * v_probabilidades[j]); //Calcula valor esperado
19    }
20    //almacena valor esperado
21    v_esperado.push(vn_valor);
22  } //for(var i in matriz)
23
24  ▼ if(esquema == 'rentabilidad'){
25
26    maximo = Math.max.apply(Math, v_esperado); //Metodo que devuelve el valor Maximo de un array de datos (valores esperados)
27    var proc = contar_repetidos(v_esperado, maximo); //Función que obtiene alternativas donde exista empaque en la decisión
28    var mensaje = "";
29    ▼ for(var i in proc.posiciones){
30      mensaje += "<b>A<sub>" + (parseInt(proc.posiciones[i]) + 1) + "</sub></b> ";
31    }
32
33    } //if(esquema == 'rentabilidad')
34  ▼ else if(esquema == 'costo'){
35
36    minimo = Math.min.apply(Math, v_esperado); //Metodo que devuelve el valor Minimo de un array de datos (valores esperados)
37    var proc = contar_repetidos(v_esperado, minimo); //Función que obtiene alternativas donde exista empaque en la decisión
38    var mensaje = "";
39    ▼ for(var i in proc.posiciones){
40      mensaje += "<b>A<sub>" + (parseInt(proc.posiciones[i]) + 1) + "</sub></b> ";
41    }
42
43    }
44    return mensaje; //retorna mensaje con resultados de la decisión
45  } //function valor_esperado(param_json)
46
```

En el código anterior se describe en los comentarios cada uno de los pasos necesarios para la aplicación del criterio del **valor esperado**.

V. DISPERSIÓN

```
47 /*Función que obtiene matriz con los valores ingresados y resuelve por el criterio de Dispersión */
48 ▼ function dispersion(param_json){
49     var matriz = param_json.matriz; //Obtiene matriz con valores
50     var esquema = param_json.esquema; //Obtiene Esquema(rentabilidad o costo)
51     var v_probabilidades = param_json.probabilidades; //Obtiene Matriz de Probabilidades
52     var v_k = param_json.v_k; //Obtiene valor de K=1
53
54     var maximo = 0;
55     var v_esperado = [];
56
57     //recorre la matriz por filas
58     for(var i in matriz){
59         var vn_valor = 0;
60         var va_filas = matriz[i]; //genera array con los valores de la fila
61         //recorre cada filas para generar el valor esperado
62         for(var j in va_filas){
63             vn_valor += (va_filas[j] * v_probabilidades[j]); //Calcula valor esperado
64         }
65         //almacena valor esperado
66         v_esperado.push(vn_valor);
67     } //for(var i in matriz)
68
69     var va_vmc = [];
70     //recorre la matriz por filas
71     for(var i in matriz){
72         var vn_vmc = 0;
73         var va_filas = matriz[i]; //genera array con los valores de la fila
74         //recorre cada filas para generar la varianza
75         for(var j in va_filas){
76             var va_varianza;
77             vn_vmc += (Math.pow((va_filas[j] - v_esperado[i]),2) * v_probabilidades[j]); //Calcula Varianza
78         }
79         //almacena valor esperado
80         va_vmc.push((v_esperado[i] - (v_k * Math.sqrt(vn_vmc)))); //Calcula el VMC
81     } //for(var i in matriz)
82
83     maximo = Math.max.apply(Math, va_vmc); //Metodo que devuelve el valor Maximo de un array de datos(VMC)
84     var proc = contar_repetidos(va_vmc, maximo); //Función que obtiene alternativas donde exista empaque en la desición
85     var mensaje = "";
86     for(var i in proc.posiciones){
87         mensaje += "<b>A<sub>" + (parseInt(proc.posiciones[i]) + 1) + "</sub></b> ";
88     }
89
90     return mensaje; //retorna mensaje con resultados de la desición
91 } //function valor_esperado(param_json)
92
```

En el código anterior se describe en los comentarios cada uno de los pasos necesarios para la aplicación del criterio de **Dispersión**.

VI. MÁXIMA PROBABILIDAD

```
93 /*Función que obtiene matriz con los valores ingresados y resuelve por el criterio de Maxima Probabilidad */
94 ▼ function maxima_probabilidad(param_json){
95     var matriz = param_json.matriz; //Obtiene matriz con valores
96     var esquema = param_json.esquema; //Obtiene Esquema(rentabilidad o costo)
97     var v_probabilidades = param_json.probabilidades; //Obtiene Matriz de Probabilidades
98     var maximo = 0;
99     maximo = Math.max.apply(Math, v_probabilidades); //Metodo que devuelve el valor Maximo de un array de datos(Probabilidades)
100     var proc = contar_repetidos(v_probabilidades, maximo); //Función que obtiene probabilidades donde exista empaque en el estados
101     var mensaje = "";
102     //Busca los maximos o minimos valores en J
103     for(var i in proc.posiciones){
104         if(esquema == 'rentabilidad'){
105             var va_valoresest = [];
106             //generar array de datos en forma de {Xj}
107             for (var mi in matriz){
108                 var va_filas = matriz[mi];
109                 va_valoresest.push(va_filas[parseInt(proc.posiciones[i])]);
110             }
111             var maximo1 = Math.max.apply(Math, va_valoresest); //Metodo que devuelve el valor Maximo de un array de datos(Max{Xj})
112             var proc1 = contar_repetidos(va_valoresest, maximo1); //Función que obtiene alternativas donde exista empaque en la desición
113             //var mensaje = "";
114             for(var i in proc1.posiciones){
115                 mensaje += "<b>A<sub>" + (parseInt(proc1.posiciones[i]) + 1) + "</sub></b> ";
116             }
117         } else if(esquema == 'costo'){
118             var va_valoresest = []; //generar array de datos en forma de {Xj}
119             for (var mi in matriz){
120                 var va_filas = matriz[mi];
121                 va_valoresest.push(va_filas[parseInt(proc.posiciones[i])]);
122             }
123             var minimo = Math.min.apply(Math, va_valoresest); //Metodo que devuelve el valor Minimo de un array de datos(Max{Xj})
124             var proc1 = contar_repetidos(va_valoresest, minimo); //Función que obtiene alternativas donde exista empaque en la desición
125             for(var i in proc1.posiciones){
126                 mensaje += "<b>A<sub>" + (parseInt(proc1.posiciones[i]) + 1) + "</sub></b> ";
127             }
128         } //else if(esquema == 'costo')
129     } //for(var i in proc.posiciones)
130     return mensaje;
131 } //function maxima_posibilidad()
```

En el código anterior se describe en los comentarios cada uno de los pasos necesarios para la aplicación del criterio de ***máxima probabilidad***.