

PLANEACIÓN PARA EL DESARROLLO DE UN PROGRAMA INFORMÁTICO PARA DESBALANCEAR MATRICES DE DISEÑO EN DISEÑOS FACTORIALES 2^k

PLANNING FOR THE DEVELOPMENT OF A COMPUTER PROGRAM TO UNBALANCE DESIGN MATRICES IN FACTORY DESIGNS 2^k

Martin Saldaña Ruiz

Tecnológico Nacional de México / IT de Celaya, México
isc.saldana.martin@gmail.com

Ricardo Ramírez Tapia

Tecnológico Nacional de México / IT de Celaya, México
raigchardo12@gmail.com

Armando Javier Ríos Lira

Tecnológico Nacional de México / IT de Celaya, México
armando.rios@itcelaya.edu.mx

Manuel Darío Hernández Ripalda

Tecnológico Nacional de México / IT de Celaya, México
dario.hernandez@itcelaya.edu.mx

Recepción: 4/mayo/2020

Aceptación: 22/septiembre/2020

Resumen

Generalmente los diseños de experimentos son desarrollados a partir de matrices de diseño balanceadas ya que de acuerdo con la literatura se sabe que el desbalance en las matrices de diseño ocasiona efectos negativos y cierto grado de error en los diferentes términos del modelo. Debido a que los efectos negativos, así como el grado de error que ocasiona el desbalance, no han sido estudiados a fondo ni cuantificados. El propósito de esta investigación es plantear el desarrollo de un programa informático que ayude a desbalancear en tres distintos grados a las matrices de diseño utilizadas en los diseños experimentales 2^k . Para posteriormente identificar y cuantificar las consecuencias del desbalance. Las categorías de desbalance que dicho programa manejará son: bajo, medio y alto. Estas categorías serán determinadas a partir de los resultados obtenidos después de aplicar el

método GBM por sus siglas en inglés "General Balance Metric". El cual únicamente nos indica el grado de balance de una determinada matriz de diseño.

Palabras Clave: Matrices de diseño, diseños experimentales 2^k , General Balance Metric, balance.

Abstract

Generally, the experimental designs are developed from balanced design matrices since, according to the literature, it is known that the imbalance in the design matrices causes negative effects and a certain degree of error in the different terms of the model. Since the negative effects, as well as the degree of error caused by the imbalance, have not been thoroughly studied or quantified. The purpose of this research is to propose the development of a computer program that helps to unbalance the design matrices used in experimental designs 2^k in three different degrees. To later identify and quantify the consequences of the imbalance. The imbalance categories that said program will handle are: low, medium and high. These categories will be determined from the results obtained after the application of the GBM method for its acronym in English "General Balance Metric". Which only tells us the degree of balance of a certain design matrix.

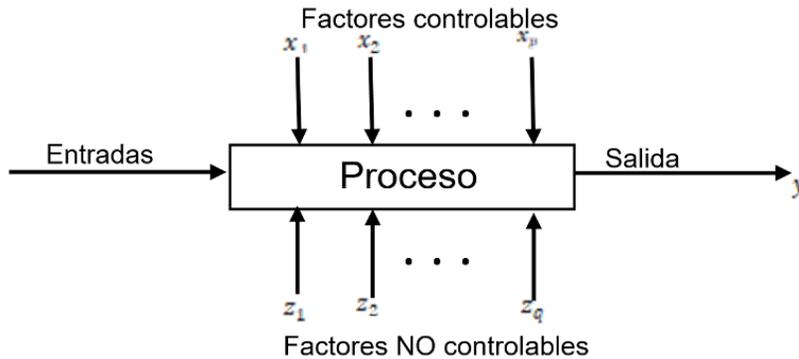
Keywords: *Balanced design matrices, experimental designs 2^k , General Balance Metric, experimental design, Balance.*

1. Introducción

Un experimento puede definirse como una prueba o una serie de pruebas en las que se hacen cambios deliberados en las variables de entrada de un proceso para observar las razones de los cambios en las variables de salida [Montgomery, 2017]. En figura 1 se muestra la representación de un proceso donde las entradas se ven afectadas tanto por factores controlables (x_p) como no controlables (z_q), teniendo estos un impacto en la salida o variable de respuesta (y).

En el área de diseño de experimentos los diseños factoriales son frecuentemente los más utilizados debido a que trabajan en cada ensayo o replica completa, todas

las posibles combinaciones de los niveles de los factores. Además, son comúnmente usados como filtro para encontrar a los factores que tienen mayor significancia o influencia sobre la variable respuesta.



Fuente: Elaboración propia.

Figura 1 Modelo general de un proceso o sistema.

Dentro de los diseños factoriales se encuentran los 2^k y los 3^k (donde k representa la cantidad de factores a evaluar y la base la cantidad de niveles que posee el factor) así como los diseños factoriales fraccionados de dos niveles, los diseños factoriales de tres niveles y los de niveles mixtos. Sin embargo, esta investigación únicamente se centrará en los diseños factoriales 2^k . El diseño factorial 2^k trabaja únicamente con dos niveles por cada factor, denominados comúnmente como alto y bajo, representados con 1 y -1 respectivamente. Este diseño tiene la restricción de que k tiene que ser mayor o igual a dos, estos diseños pueden emplearse para representar tanto factores cualitativos como cuantitativos [Montgomery, 2017]. La matriz de diseño es un arreglo ortogonal, en el cual se representan los niveles de cada factor, donde la cantidad de corridas o tratamientos está definida por el número de factores, esto se ilustra en la tabla 1, la cual muestra la representación gráfica de un diseño factorial 2^3 .

El balance es una propiedad importante en el diseño experimental, se dice que un diseño está balanceado cuando en cada columna aparece cada posible nivel de factor el mismo número de veces [Ríos,2009]. Además, los experimentos balanceados tienen dos grandes ventajas cuando se trata de análisis. Primero, cada combinación experimental se estima con la misma precisión.

Tabla 1 Ejemplo de matriz de diseño, de un factorial 2^3 .

Corrida	A	B	C
1	-1	-1	-1
2	1	-1	-1
3	-1	1	-1
4	1	1	-1
5	-1	-1	1
6	1	-1	1
7	-1	1	1
8	1	1	1

Fuente: Elaboración propia.

En segundo lugar, en un experimento factorial balanceado, el efecto de cada factor puede evaluarse de forma independiente, respecto de los otros factores [Van Belle & F. Kerr, 2012]. El balance en las matrices de diseño nos asegura que exista ortogonalidad entre los factores y la intersección. Existen dos formas de medir el balance en las matrices de diseño, una de estas es con el estadístico ji-cuadrado, en el cual el valor que devuelva debe de ser igual a cero para poder decir que la matriz de diseño está balanceada.

El otro método es el GBM "General Balance Metric" este método devuelve el grado de balance de una determinada matriz de diseño. Para la realización del método de desbalance, se utilizará el método de GBM. De acuerdo con [Guo, Simpson, & Pignatiello, 2009] el GBM puede ser calculado para una matriz d de tamaño $n \times k$, donde n es el número de filas y k es el número de columnas (ecuaciones 1, 2 y 3), $dt(t = 1, \dots, k)$ indica las columnas de interacciones para los factores desde 1 hasta k y d^1 representa la matriz de efectos principales.

$$GBM = (H^1, H^2, \dots, H^3, \dots H^k) \quad (1)$$

$$H^t = \sum_{j=1}^{\binom{k}{t}} H_j^t = \sum_{j=1}^{\binom{k}{t}} \sum_{r=1}^{l_j^t} \left(c_{rj}^t - \frac{n}{l_j^t} \right)^2 \quad (2)$$

$$H_j^t = \sum_{r=1}^{l_j^t} \left(c_{rj}^t - \frac{n}{l_j^t} \right)^2 \quad (3)$$

Donde:

c_{rj}^t = número de veces que un nivel r aparece en la columna j

l_j^t = número de niveles que la columna j contiene.

El cálculo del GBM proviene de la diferencia del número de veces que un nivel debería aparecer respecto al número de veces que aparece, estos cálculos se hacen para la submatriz de efectos principales, y para las submatrices de interacciones de dos y de tres factores. Finalmente se obtiene un vector como el de la ecuación 3, donde H^1 indica el balance para los efectos principales, H^2 para interacciones de dos factores y H^3 para interacciones de tres factores. Cuando un diseño está balanceado para efectos principales, para interacciones de dos factores y para interacciones de tres factores, el GBM es un vector de ceros (0,0,0) [Naranjo Palacios, Pantoja Pacheco, Ríos Lira, & Tapia Esquivias, 2020].

Además, de la propiedad de balance, también existe la propiedad de ortogonalidad, la cual asegura que las diferentes clases de efectos a los que está sujeto el material experimental se podrán estimar de forma directa y sin confusiones Yates [1933]. En otras palabras, la ortogonalidad asegura que los efectos puedan ser estimados de forma independiente [Pantoja, Ríos, & Tapia Esquivias, 2019]. Existen varios métodos para medir la ortogonalidad:

- Producto Punto
- Correlación
- Coeficientes de correlación
- Factor de inflación de varianza

El método para medir la ortogonalidad que se implementará será el producto punto, esto por su practicidad. Si el producto punto entre dos vectores $x = (x_1, x_2 \dots x_n)$ y $y = (y_1, y_2, \dots y_n)$ representados en la ecuación 4, da como resultado 0, se dice que los vectores son ortogonales [Murray, 2004].

$$x \cdot y = x_1y_1 + x_2y_2 + \dots x_ny_n \quad (4)$$

Donde: x_ny_n = representa el producto punto de los factores x_n e y_n .

Aunque es el método más práctico y sencillo de implementar, este solo sirve para determinar si dos vectores son o no ortogonales, es decir, este método no cuantifica los grados de ortogonalidad entre dos vectores, pero debido a que solo es necesario verificar si los factores son ortogonales, el producto punto resulta suficiente.

Algoritmo y programación

Se denomina algoritmo al conjunto de pasos ordenados y finitos que permiten resolver un problema o tarea específica. Los algoritmos son independientes del lenguaje de programación y de la computadora que se vaya a emplear para ejecutarlo. Todo algoritmo debe ser: finito, preciso, definido y general. La representación de los algoritmos se da a través de los diagramas de flujo o de pseudocódigo [Corona & Ancona, 2011]. Un diagrama de flujo es una representación gráfica de un algoritmo, esta representación, se vale de símbolos; los cuales indican diferentes procesos a realizar [Joyanes, 2008]. El pseudocódigo, es la técnica que permite expresar la solución de un problema mediante un algoritmo escrito en palabras normales de un idioma (por ejemplo, español), utilizando palabras imperativas. Es común encontrar en pseudocódigo palabras como: Inicie, lea, imprima, sume, divida, calcule, finalice. No hay un léxico obligado para el pseudocódigo, pero con el uso frecuente se ha establecido algunos estándares [Joyanes, 2008]. El lenguaje de programación es la combinación de símbolos y reglas que permiten la elaboración de programas con los cuales la computadora puede realizar tareas o resolver problemas de manera eficiente [Corona & Ancona, 2011]. Existen diferentes técnicas de programación, las más utilizadas son:

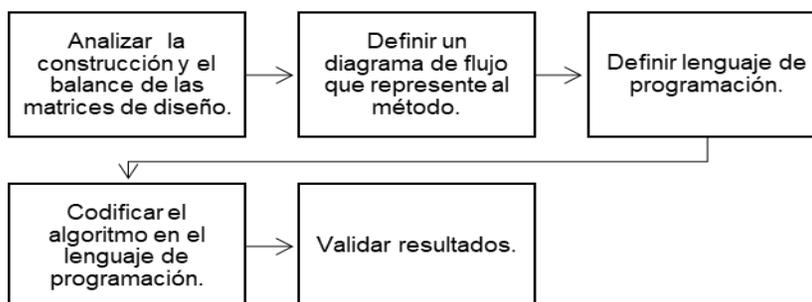
- Programación descendente
- Estructura de datos
- Estructura de control
- Programación modular [Corona & Ancona, 2011].

La planeación es la acción de la elaboración de estrategias que permiten alcanzar una meta ya establecida. Partiendo de este concepto y del enfoque de esta investigación, se elaborará la planificación pertinente para el desarrollo de un programa que creé matrices de diseño con tres distintos grados de balance para los diseños factoriales desde 2^3 hasta 2^9 . Las matrices de diseño generadas a partir del programa desarrollado se utilizarán en un futuro como caso de estudio para analizar y cuantificar los efectos del desbalance en la estimación de los coeficientes de regresión y del error que esto pueda generar.

2. Métodos

Modelo de Experimentación Analítico

Con base en una secuencia lógica. La planeación para el desarrollo del programa se dividirá en cinco etapas, las cuales se muestran en la figura 2.



Fuente: Elaboración propia.

Figura 2 Planificación para el desarrollo del programa.

Etapa 1. Analizar la construcción y el balance de las matrices de diseño

Los diseños factoriales con los cuales se trabajó fueron de 2^3 hasta 2^9 . La cantidad de interacciones con los que estos diseños trabajan va desde dos factores hasta interacciones de nueve factores. Se dice que la interacción existe cuando el efecto de una variable independiente X_1 en una variable dependiente (Y) depende del valor de otra variable independiente X_2 [Jaccard, 1998]. En la tabla 2 que se presenta a continuación, se muestran los diseños factoriales analizados en este trabajo, así como los grados de interacciones que maneja cada diseño factorial. Además del total de interacciones que usa cada diseño factorial.

Tabla 2 Grados de interacciones utilizadas por los diseños factoriales.

Diseños Factoriales	Interacciones entre Factores									Total
	1°	2°	3°	4°	5°	6°	7°	8°	9°	
2^3	3	3	1							7
2^4	4	6	4	1						15
2^5	5	10	10	5	1					31
2^6	6	15	20	15	6	1				63
2^7	7	21	35	35	21	7	1			127
2^8	8	28	56	70	56	28	8	1		255
2^9	9	36	84	126	126	84	36	9	1	511

Fuente: Elaboración propia.

Se hizo una revisión de cada diseño factorial, tomando en cuenta la cantidad de corridas que manejan (tabla 3) así como de las interacciones a utilizar (Tabla 4). A partir de esta revisión se comenzó por tomar una columna o factor pivote, en el cual se ubicaron y seleccionaron los niveles que aparecían en alto de dicha columna pivote, a partir de esto se comenzó a multiplicar la matriz de diseño por un vector de -1 (niveles bajo). De forma empírica la cantidad de -1 del vector multiplicador fue aumentando. Esto ocasionó que la matriz de diseño se comenzará a desbalancear, posteriormente se calculó el balance de la matriz de diseño, a través del GBM.

Tabla 3 Grados de interacciones utilizadas.

Diseños Factoriales	Interacciones entre Factores									Total
	1°	2°	3°	4°	5°	6°	7°	8°	9°	
2 ³	3	3	1	-	-	-	-	-	-	7
2 ⁴	4	3	1	-	-	-	-	-	-	8
2 ⁵	5	4	1	-	-	-	-	-	-	10
2 ⁶	6	7	1	-	-	-	-	-	-	14
2 ⁷	7	11	1	-	-	-	-	-	-	19
2 ⁸	8	14	1	-	-	-	-	-	-	23
2 ⁹	9	18	1	-	-	-	-	-	-	28

Fuente: Elaboración propia.

Tabla 4 Interacciones con los que cada diseño factorial trabajará.

Diseño	Factores e interacciones																											
2 ³	A	B	C	AB	AC	BC	ABC																					
2 ⁴	A	B	C	D	AB	AC	AD	ACD																				
2 ⁵	A	B	C	D	E	AD	AE	BC	BD	ADE																		
2 ⁶	A	B	C	D	E	F	AC	AD	AE	AF	BC	BE	BF	AEF														
2 ⁷	A	B	C	D	E	F	G	AE	AF	AG	BC	BD	BE	CD	CE	DF	DG	EG	AFG									
2 ⁸	A	B	C	D	E	F	G	H	AF	AG	AH	BE	BF	BG	CF	CG	CH	DE	DF	DH	EF	EG	AGH					
2 ⁹	A	B	C	D	E	F	G	H	I	AG	AH	AI	BE	BF	BG	CG	CH	CI	DE	DF	DG	EG	EH	EI	FG	FH	FI	AHI

Fuente: Elaboración propia.

Con la información recabada, se creó la tabla 5, en la cual se presentan los diseños factoriales desde 2³ hasta 2⁹, la cantidad de corridas de cada diseño, cantidad de niveles bajos posibles, así como cantidad de niveles altos posibles y primordialmente el porcentaje de niveles bajos que debe de tener cada modelo para tener un grado de desbalance, bajo, medio o alto, de forma correspondiente. Además del cálculo del GBM para cada diseño y cada grado de desbalance.

Tabla 5 Diseños factoriales, con grado bajo, medio y alto de desbalance.

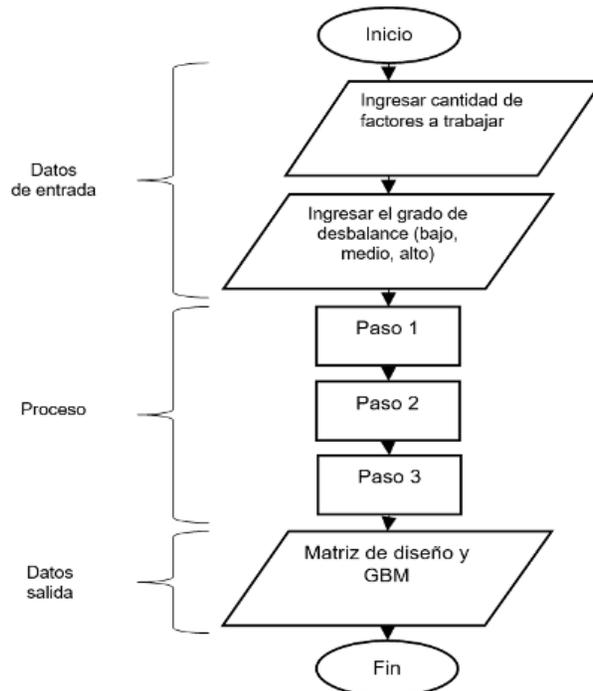
Diseño	Total de corridas	Grado de desbalance	Nivel Bajo	Nivel Alto	%Niv. Bajo	GBM
2 ³	8	Nulo	4	4	50%	0,0,0
		Bajo	5	3	63%	6,0,2
		Medio	6	2	75%	16,0,0
		Alto	7	1	88%	22,0,2
2 ⁴	16	Nulo	8	8	50%	0,0,0
		Bajo	10	6	63%	24,0,8
		Medio	11	5	69%	40,0,2
		Alto	13	3	81%	72,0,2
2 ⁵	32	Nulo	16	16	50%	0,0,0
		Bajo	21	11	66%	122,0,18
		Medio	25	7	78%	266,0,2
		Alto	29	3	91%	378,0,18
2 ⁶	64	Nulo	32	32	50%	0,0,0
		Bajo	39	25	61%	300,0,98
		Medio	46	18	72%	808,0,8
		Alto	53	11	83%	1196,0,50
2 ⁷	128	Nulo	64	64	50%	0,0,0
		Bajo	79	49	62%	1358,0,450
		Medio	94	34	73%	3632,0,8
		Alto	109	19	85%	5150,0,338
2 ⁸	256	Nulo	128	128	50%	0,0,0
		Bajo	161	95	63%	6288,0,1922
		Medio	195	61	76%	16496,0,18
		Alto	229	27	89%	23440,0,1458
2 ⁹	512	Nulo	256	256	50%	0,0,0
		Bajo	323	189	63%	25474,0,7442
		Medio	391	121	76%	66130,0,98
		Alto	459	53	90%	94210,0,5618

Fuente: Elaboración propia.

Etapa 2. Definir un diagrama de flujo que represente al método

La representación del método a través de un diagrama de flujo facilitará su futura codificación en algún lenguaje de programación. En la figura 3 está representado el diagrama de flujo del método. El flujo de información comenzará cuando el usuario ingrese la cantidad de factores con los que va a trabajar, esto pueden ser desde tres hasta nueve. Como segundo dato de entrada, el usuario deberá de ingresar el grado de desbalance con que se va a trabajar, teniendo como opción bajo, medio u alto. Una vez ingresados los datos de entrada, se comenzará con el paso 1 del proceso, en el cual se generarán las interacciones de la matriz de diseño, y se desbalanceará de acuerdo con el grado de desbalance previamente seleccionado. El paso 2 del proceso, consiste en calcular el GBM de la matriz de diseño creada

en el paso 1. El paso 3 del proceso, verificará a través del producto punto, la ortogonalidad de la matriz de diseño desarrollada. Como datos de salida, tendremos la matriz de diseño desbalanceada, así como el GBM de dicha matriz.



Fuente: Elaboración propia.

Figura 3 Diagrama de flujo.

Etapa 3. Definir lenguaje de programación

Se comenzará por seleccionar el entorno de programación, el cual será MATLAB, esto debido a que es un software especializado en operaciones matriciales, lo cual será de mucha ayuda para realizar la codificación. MATLAB maneja su propio lenguaje de programación el cual es llamado lenguaje M. Este lenguaje es interpretado, y puede ejecutarse tanto en el entorno interactivo, así como a través de un archivo script. MATLAB está disponible para los sistemas operativos Unix, Windows, macOS y GNU/Linux.

Etapa 4. Codificar el algoritmo en el lenguaje de programación

Una vez seleccionado el entorno de desarrollo, así como el lenguaje de programación, se procederá a la codificación, para lo cual el diagrama de flujo

fungirá un papel importante. La técnica de programación que se utilizará será la programación modular, esto por su practicidad. Ya que resulta más fácil el estar trabajando con módulos.

Etapa 5. Validar resultados

Es de vital importancia validar los datos de salida que arroje el programa. En esta etapa, se propone llevar a cabo la elaboración de matrices de diseño en la hoja de cálculo Excel, con el fin de comparar los resultados obtenidos a través del software y los desarrollados en Excel. De esta forma se podrán depurar cualquier error de lógica cometido en el programa.

3. Resultados

En el presente artículo únicamente se muestra la planeación para el desarrollo de un programa informático para generar matrices de diseño con tres distintos grados de desbalance para los diseños factoriales 2^k .

La primera etapa sirvió para crear una estrategia con la cual se analizó la construcción y el balance de las matrices de diseño, acotando las interacciones de los distintos diseños factoriales con los que se trabajó. En la segunda etapa, se creó un diagrama de flujo donde se representó de forma gráfica el método para llevar a cabo el programa. El diagrama de flujo está separado en tres secciones las cuales son datos de entrada, proceso y datos de salida.

En la primera sección el usuario ingresará la cantidad de factores con los cuales trabajará, así como el grado de desbalance deseado (bajo, medio u alto).

En la segunda sección se realizarán distintas operaciones matriciales para generar las interacciones de la matriz de diseño para después calcular el GBM y por último verificar su ortogonalidad.

En la tercera sección el programa imprimirá en pantalla la matriz de diseño desbalanceada, así como el GBM que esta posea. En las figuras 2, 3 y 4 se muestra la interfaz del programa, tras la ejecución de un diseño 2^3 , con los distintos grados de desbalance que maneja el programa (bajo, medio y alto), la interacción con el usuario es a través de command window de MATLAB.

Ingrese el la cantidad de factores a trabajar: 3
Ingresa el grado de desbalance ninguno=0 bajo=1, medio=2, alto=3: 1
La matriz cuenta con los factores A,B,C y las interacciones AB,AC,BC,ABC respectivamente

MatrizD =

-1	-1	-1	1	1	1	-1
-1	1	1	-1	-1	1	-1
-1	1	-1	-1	1	-1	1
1	1	-1	1	-1	-1	-1
-1	-1	1	1	-1	-1	1
1	-1	1	-1	1	-1	-1
-1	1	1	-1	-1	1	-1
1	1	1	1	1	1	1

El GBM de la matriz es:d1 = 6 d2 = 0 d3 = 2
La matriz de diseño es ortogonal

Fuente: Elaboración propia.

Figura 2 Interfaz del programa, diseño 2^3 , desbalance bajo.

Ingrese el la cantidad de factores a trabajar: 3
Ingresa el grado de desbalance ninguno=0 bajo=1, medio=2, alto=3: 2
La matriz cuenta con los factores A,B,C y las interacciones AB,AC,BC,ABC respectivamente

MatrizD =

-1	-1	-1	1	1	1	-1
-1	1	1	-1	-1	1	-1
-1	1	-1	-1	1	-1	1
-1	-1	1	1	-1	-1	1
-1	-1	1	1	-1	-1	1
1	-1	1	-1	1	-1	-1
-1	1	1	-1	-1	1	-1
1	1	1	1	1	1	1

El GBM de la matriz es: d1 = 16 d2 = 0 d3 = 0
La matriz de diseño es ortogonal

Fuente: Elaboración propia.

Figura 3 Interfaz del programa, diseño 2^3 , desbalance medio.

Ingrese el la cantidad de factores a trabajar: 3
Ingresa el grado de desbalance ninguno=0 bajo=1, medio=2, alto=3: 3
La matriz cuenta con los factores A,B,C y las interacciones AB,AC,BC,ABC respectivamente

MatrizD =

-1	-1	-1	1	1	1	-1
-1	1	1	-1	-1	1	-1
-1	1	-1	-1	1	-1	1
-1	-1	1	1	-1	-1	1
-1	-1	1	1	-1	-1	1
-1	1	-1	-1	1	-1	1
-1	1	1	-1	-1	1	-1
1	1	1	1	1	1	1

El GBM de la matriz es: d1 = 22 d2 = 0 d3 = 2
La matriz de diseño es ortogonal

Fuente: Elaboración propia.

Figura 4 Interfaz del programa, diseño 2^3 , desbalance alto.

En la tercera etapa se seleccionó el entorno de desarrollo, siendo MATLAB el software seleccionado, esto debido a la practicidad del programa para el manejo de matrices, la selección del entorno de desarrollo lleva implícita la selección del lenguaje de programación, que en este caso sería el lenguaje M. En la cuarta etapa se llevó a cabo la codificación, la cual se realizó a partir del diagrama de flujo. La estrategia de programación que se utilizó fue la programación modular, debido a la flexibilidad que esta presenta.

En la quinta etapa, se realizó la validación de los resultados, la cual se llevó a cabo comparando la salida del programa con los resultados de una plantilla elaborada en la hoja de cálculo Excel, esto permitió tener un punto de comparación para poder realizar las correcciones pertinentes, en el caso de que sean necesarias. Con este estudio se desarrollaron matrices de diseño con tres distintos grados de desbalance, que posteriormente formarán parte de otra investigación donde se analizará el impacto del desbalance en el cálculo de los coeficientes de regresión, así como el error que este pudiera generar.

4. Discusión

En el área de diseño de experimentos los diseños factoriales son frecuentemente los más utilizados debido a que trabajan en cada ensayo o replica completa, todas las posibles combinaciones de los niveles de los factores. Además, son comúnmente usados como filtro para encontrar a los factores que tienen mayor significancia o influencia sobre la variable respuesta. La matriz de diseño que es manejada en los diseños factoriales, tiene que cubrir ciertas características, siendo la propiedad de balance una de las primordiales. De acuerdo a la literatura, se sabe que la falta de balance ocasiona cierto error en la estimación de los regresores, además de otros efectos negativos, los cuales no han sido estudiados ni cuantificados a fondo. Debido a esto se considera importante desarrollar un programa informático el cual desbalancee las matrices de diseño en tres distintos grados de desbalance, para posteriormente trabajar con las matrices desbalanceadas y estudiar y cuantificar las repercusiones que conlleva el desbalance. La planeación presentada en este documento muestra las diferentes

etapas para el desarrollo de un programa informático que desbalancee las matrices de diseño en tres diferentes grados. El análisis y la experimentación empírica, permitió el acomodo de los pasos para el desarrollo del programa informático de manera secuencial lógica, que posteriormente se utilizó para la elaboración de un diagrama de flujo. El diagrama de flujo, es una herramienta utilizada ampliamente como paso previo a la programación, no siendo este caso la excepción. Tras el desarrollo del diagrama de flujo se procedió a la selección de un entorno de programación siendo MATLAB el entorno seleccionado y por ende el uso del lenguaje de programación M. La técnica de programación con la que se trabajó para la codificación fue la programación modular. Como último paso se procedió a la validación de los resultados, la cual se hizo a través de una plantilla creada en la hoja de cálculo Excel. El hecho de haber desarrollado este programa informático facilitará al experimentador, conocer el grado de balance con el que se está trabajando y esto permitirá un mejor análisis en cómo afecta el desbalance en los resultados finales.

5. Bibliografía y Referencias

- [1] Corona, M. A., & Ancona, M. A. (2011). Diseño de algoritmos y su codificación en el lenguaje C. México: McGraw-Hill/Interamericana Editores.
- [2] Guo, Y., Simpson, J. R., & Pignatiello, J.J. (2009). The general balance metric for mixed-level fractional factorial designs. *Quality and Reliability Engineering International*, 25(3), 335-344. Doi:10.1002/qre.982
- [3] Jaccard, J. J. (1998). Interaction effects in factorial analysis of variance. SAGE Publications.
- [4] Murray, M. A. (2004). Sobre el producto cruz en espacios vectoriales n-dimensionales. *Ingeniería. Investigación y Tecnología*, 27-47: www.redalyc.org/pdf/404/40450103.pdf
- [5] Joyanes, L. (2008). Fundamentos de programación. Madrid: McGraw-Hill/Interamericana de España.
- [6] Montgomery, D. (2017). Design and Analysis of Experiments. New York: John Wiley & Sons.

- [7] Naranjos Palacios, F., Pantoja Pacheco, Y. V., Ríos Lira, A. J., & Tapia Esquivias, M. (2020). Diseños ortogonales de Taguchi fraccionados. *Ingeniería Investigación y Tecnología*, 21(2), 1-12. Doi: 10.22201/ii.25940732e.2020.21n2.011
- [8] Pantoja, Y. V., Ríos, A. J., & Tapia Esquivias, M. (2019). A method for construction of mixed-level fractional designs. *Quality and Reliability Engineering International*, 35. Doi: 10.1002/qre.2466
- [9] Ríos, A. (2009). Sequential Experimentation, A new experimentation approach for resolution III, mixed-level and robust designs. VFM Verlag Dr. Müller.
- [10] Salmorán López, E., Ríos Lira, A. J., & Pantoja Pacheco, Y. V. (2019). Planeación para la generación de un programa para construir estructuras de alias para diseños factoriales fraccionados de niveles mixtos. *Pistas Educativas*, 41(133), 580-594.
- [11] Van Belle, G., & F. Kerr, K. (2012). Design and analysis of experiments in the health sciences. John Wiley & Sons.
- [12] Yates, F. (1933). The Principles of Orthogonality and Confounding in Replicated Experiments. Cambridge University Press, 108-145. Doi: 10.1017/S0021859600052916.