Universidad Nacional Autónoma de México

FACULTAD DE CIENCIAS

Estudio de Métodos para la Obtención de Dígitos de Control

T E S I S

QUE PARA OBTENER EL TITULO DE:

A C T U A R I O

P R E S E N T A:

María Elena Escalante Pliego

MEXICO, D. F. 1982

C O N T E N I D O

INTODUCCION.

En la actualidad se ha hecho cada vez mas necesaria la creación de una clave para identificar a cada uno de los elementos que integran los bancos de información que se ván a manejar en un sistema determinado.

Como ejemplos podrían citarse algunos conocidos en nuestro medio, como son: los números de cuenta de los sistemas bancarios, el número de cuenta de la UNAM, Registro Federal de Causantes.

Dichas claves se han generado para tener un más fácil y eficiente manejo de la información; por ello, es escencial que al hacer uso de éstas, no se cometan errores.

Se hace referencia a ésto, pues será un error bastante grave si, por ejemplo, se le descontara o incrementara la cuenta bancaria de un cuenta-habiente en lugar de la que debió de modificarse, pues, a causa de este error, habría reclamaciones posteriores por parte de la persona afectada, debiéndosele reponer su dinero, sin saber, tal vez, dónde fue abonado erroneamente.

En el caso de un sistema escolar, en el que se manejan a los alumnos por medio de una clave individual (por ejemplo número de cuenta de la UNAM), son de imaginarse los problemas que se acarrearian si se le acreditara una o varias materias a otra persona, que no fuera la indicada, por un error en el manejo de dichas claves individuales. Podría suceder que la persona afectada tuviera que volver a cursar dichas materias, mientras que a la que se le acreditaron erroneamente ya no las cursaría. En forma similar se podrían listar las consecuencias de muchos otros errores.

Estudiando estos graves problemas, han surgido algunas técnicas para detectar el mayor número de estos errores y prevenir así sus consecuencias.

Se ha encontrado que este problema puede asociarse con el que surge al momento de la transmición de la información en una computadora, que comenzó a estudiarse a partir de los trabajos de Claude E. Shannon en el año de 1948

Sus trabajos se enfocan a la teoría de la transmición de información, y a la teoría de la codificación de ésta. Estos marcan el principio de una serie de estudios[19,20,21,22] acerca del diseño de esquemas eficientes por medio de los cuales la información pueda ser codificada para transmisiones dignas de confianza a través de los canales de una computadora, los cuales son corrompidos por ruido. El punto escencial a lo largo de los estudios realizados desde entonces [19,20,21,22], ha sido el encontrar un diseño de codificación y decodificación que proporcione como resultado la recepción de información correcta, y que al mismo tiempo sea un sistema fácil de implementar.

Para entender mejor ésto es necesario hacer un paréntesis para entender que codificación se entiende como el convertir cierta información a un código que conste de letras y/o símbolos definidos con anterioridad; y decodificación sería el regresar esta clave a la forma que en un principio se tenía (información fuente).

Supóngase que se quiere transmitir una secuencia de dígitos binarios a lo largo de un canal con ruido, y que al hacerlo el resultado de mandar un 1 sea efectivamente un 1. Es decir, que a lo largo de la transmisión no cambie la información. Como no se es capáz de prevenir que los canales manden ruido, por lo tanto, el problema radica en encontrar un diseño de códigos y de una técnica de detección se errores sobre estos códigos. La idea es la siguiente:

Se toma una serie de k dígitos (mensaje que se quiere transmitir), se le anexan r dígitos de chequeo, y se transmite el bloque completo n = k + r dígitos por canal.

El inconveniente que puede surgir aquí es que los bloques transmitidos contengan poca información y sean demaciado grandes debido a que se les haya anexado una gran cantidad de dígitos de control, provocando ésto que si se trata de almacenar dichos bloques ocupen un mayor espacio, o si se quiere transmitirlos, por ser de un gran tamaño en comparación con la información en si, tome mas tiempo, volviendo mas lento el proceso.

Como puede verse, hay que encontrar un número de dígitos de control r que sea el más eficiente, aplicados a un sistema que también lo sea.

Un sistema que se usa con frecuencia es el de añadir un solo dígito al mensaje que se está transmitiendo (por ejemplo: bit de paridad), resultando así una gran cantidad de información a transmitir con un mínimo de dígitos añadidos a ésta como control. El problema resulta al momento de decodificar, siendo no muy eficiente ya que en caso de que haya ocurrido algún error, no puede detectarse dónde ocurrió éste y por lo tanto sería muy difícil el corregirlo.

Con el desarrollo de los sistemas de información se empezaron a estudiar teorías para la detección de errores, pero ahora no solo en lo relacionado a la transmisión de información en una computadora, sino en el manejo de claves como las que se mencionaron en un principio, enfocándose éstos al diseño de una técnica que detecte la mayor cantidad de errores y no a la corrección automática de éstos.

Estas técnicas se han ido denominando como el estudio de dígitos de control, y consisten en añadir uno o varios caracteres o dígitos a la clave que se mencionó antes, obteniéndose éste de acuerdo a alguna combinación de los dígitos originales, con el fin de permitir verificar si ésta ha sido escrita correctamente o no.

Con el desarrollo de la comunicación entre las máquinas y el establecimiento de las redes de computadoras, se introduce la necesidad del control de los mensajes (paquetes) que se intercambian entre las diversas máquinas de una red. El tamaño de estos paquetes puede llegar a implicar una mas difícil detección de errores en la transmisión, lo que vá a dar lugar al desarrollo de técnicas más sofisticadas (usando polinomios), que son en cierta forma una generalización de las técnicas estudiadas aquí.

Estas técnicas encuentran dígitos de control a través de cierta función o manipulación (i.e.:operaciones aritméticas) de los dígitos que componen la clave.

Las operaciones que se emplean en algunos de los métodos estudiados son, por un lado el multiplicar cada uno de los dígitos que componen la clave por ciertos valores llamados "pesos" y, por otro, el aplicar la función modulo N.

En base a esto, se han ido desarrollando a lo largo del tiempo distintas técnicas cuyo propósito es el detectar la mayor cantidad de los errores que suceden con más frecuencia.

Uno de los métodos estudiados es aquel en el que se emplea la función módulo 11 pues, dependiendo de los "pesos", tendrá buenos porcentajes de detección de errores. Este método lo analiza Beckley[4] enfocando su atención a ciertos tipos de error; el método que el propone genera 10 distintos posibles dígitos (1,2,3,4,5, 6,7,8,9,10). No estudia la forma de que cada uno de estos dígitos sea de un solo caracter (el dígito 10 realmente tiene 2).

Mas adelante, Wild[18] estudia en forma mas generalizada un sistema en base N indicando las características que deben cumplir los "pesos" optimizar el proceso de detección de errores en este sistema.

De aquí en adelante, los siguientes autores de este tema se han dedicado a perfeccionar basicamente el sistema módulo 11. Algunos proponen distintas soluciones para substituir el dígito(s) 10 como por la letra A, omitiendo las claves que lo generen [9], o repitiendo el cálculo del dígito con una serie de "pesos" distinta que no generará una vez mas ese dígito [15]. Y así sucesivamente tratando de mejorar cada vez mas este método [1,7,8,10, 16].

De la misma forma se han estudiado sistemas módulo 10 [11,12, 13], pues, como se vera después, también pueden obtenerse buenos porcentajes de detección con estos métodos.

De todo lo analizado, no queda claro el por qué algunos autores han escogido módulo 11, que aparentemente es un número arbitrario, y cuál es el impacto de utilizar la base decimal clásica a cualquier otra base.

Reflexionando en el trabajo que se ha desarrollado sobre este tema, ha resultado necesario el realizar un estudio de los métodos para la creación de los dígitos de control, en el cual se ha efectuado un análisis y un mecanismo de comparación de éstos, al mismo tiempo que se han podido visualizar las ventajas y desventajas que cada uno lleva consigo.

En el Capítulo 1, se presenta una explicación del por qué pueden suceder errores al operar con claves numéricas, que será basicamente el tipo de claves con que se trabajará, compuestas por dígitos decimales, exeptuándose el caso de claves alfabéticas que se estudiará por separado (ver Capítulo 6).

En el mismo capítulo, se hará un análisis de los tipos de errores que pueden ocurrir, haciéndose una clasificación con respecto a los más frecuentes.

En el Capítulo 2 se propondrá una función para calcular dígitos de control basada en Módulo 10, por lo que dada la forma en que está definida, y que N = 10, proporcionará un solo dígito de control para cada clave, y buenos porcentajes de captación de errores.

Al mismo tiempo, se presentará un mecanismo a base de tablas, para observar y explicar más facilmente los cambios que produce un error en la clave sobre la generación de los dígitos de control, obteniendose una serie de reglas que deben de cumplir los dígitos que se encuentran en las tablas antes mencionadas, para que ciertos errores siempre sean detectados.

En el Capítulo 3 se verá por qué no es conveniente usar un sistema de dígitos de control módulo N, para N menor que 10, cuando la clave está formada por dígitos decimales.

En el Capítulo 4, se expondrán algunos sistemas para N = 11, que siendo muy eficientes, tendrán el inconveniente de la cantidad de espacio necesario para el dígito verificador. Se ha tratado de solucionar este problema; en este mismo capítulo se presentarán algunas posibles soluciones a esto.

En el Capítulo 5 se presentan en forma más generalizada las características que deben de cumplir los sistemas en base N para N mayor que 11 de tal modo que capten el mayor número de errores, y dadas estas, los porcentajes de captación de errores dependiendo del valor de N.

En el Capítulo 6 se verá un sistema de dígitos de control para claves alfabéticas como un ejemplo de una extensión posible a lo que se expone en los capítulos anteriores a éste. Como se podrá notar, este sistema va a tener buenos porcentajes de captación de errores.

Y finalmente en el Capítulo 7, se presentan las conclusiones del presente estudio, con una tabla de porcentajes de captación de errores para cada uno de los métodos expuestos.

CAPITULO 1. Descripción del Problema.

En nuestros días, cada vez es mayor la cantidad de información que debe manejarse, y la necesidad de simplificar su identificación mediante una clave numérica.

Se puede tomar como ejemplo una fábrica de herramientas. Entre otras cosas, esta fábrica debe manejar información concerniente a las personas que trabajan en ella, acerca de los productos que fabrica, la materia prima, las órdenes de producción, etcétera.

Supóngase que dicha fábrica cuenta con un archivo en el cual tiene todos los datos necesarios por cada trabajador. En un momento dado quiere consultar un registro de este archivo. Si es un ser humano el que va a realizar la búsqueda, puede ser que resulte más fácil el hacerlo, si va checando por el nombre de la persona; por el contrario, si va a ser un sistema computarizado el que lo haga, resultará mucho más fácil el encontrarlo, si cada persona tiene una clave, y por ésta se hace la búsqueda, a que si se hace por el nombre. Esto es porque, en el segundo caso, puede por ejemplo suceder que dos personas se llamen exactamente igual (cosa no imposible).

En este caso, cómo se sabrá a cuál de éstas se está haciendo referencia ?

Otro problema que puede haber, es el caso de las abreviaciones y de los espacios en blanco, pues un sistema computarizado buscará a partir de cierta máscara, y si alguna vez se escribe ésta distinta, ya sea por abreviar un nombre o por dejar espacios de más, no se encontrará el registro buscado.

Puede ocurrir también que, cuando sea dado el nombre de la persona, no sea proporcionado completo, por ejemplo:

Nombre Proporcionado:	PEREZ ARENAS JUAN
Nombre Completo:	PEREZ ARENAS JOSE JUAN

y de ésta forma, no encontrarlo en el archivo.

Por éstos motivos, es cada vez más común el que se maneje esta información por medio de claves, que casi siempre serán más cortas que el identificador antes señalado, siendo otro punto a favor de éstas.

En el ejemplo dado, se puede ver que de igual forma que a los empleados les es asignada una clave (casi siempre el RFC, ya que es único para cada persona cuando se le anexa la clave de homonimia), a los productos fabricados también puede asignárseles una, dependiendo por ejemplo del departamento que los fabrica, etcétera.

Se ha puesto un ejemplo de este tipo, para hacer notar que no solo se le puede asignar una clave a cada individuo, sino también, a información

de distinta índole.

Pero, al manejar las claves que se han mencionado, pueden cometerse errores. Por ejemplo, si es una persona la que las está dictando, y otra escribiendo, puede suceder que ciertos números no los entienda correctamente ésta última, y por lo tanto los escriba erroneamente. De igual forma, pueden ocurrir errores si estas claves son leídas de un documento escrito en forma manuscrita no muy clara. O también porque sencillamente la persona que escribe u ocupe dichas claves se equivoque.

Pensando en esto, y considerando los graves problemas que pueden acarrear estos errores, han surgido los Sistemas de Digitos de Control.

Las técnicas para encontrarlos, que se veran más adelante, obtienen dichos dígitos en función de los que forman la clave.

Como ejemplos de estos tipos de función, pueden verse los siguientes:

1.- Suma de dígitos.

Clave: 731254

Dígito: 7+3+1+2+5+4 = 22

Método: Sumar todos los dígitos que componen la clave.

Nueva clave: 731254-22

2.- Suma Módulo 10.

Clave: 51348

Dígito: (5+1+3+4+8) mod 10 = 1

Método: Sumar todos los valores que forman la clave, y al resultado aplicarle la función módulo 10.

Nueva clave: 51348-1

3.- Suma con pesos Módulo 11.

Clave: 51348

Dígito: [(5+3+8)2 + (1+4)5] mod 11 = 7

Método: Sumar los elementos en posiciones nones de la clave, y multiplicar el resultado por 2, a esto sumarle el resultado de sumar los elementos en posiciones pares multiplicados por 5, y al resultado de todo aplicarle la función módulo 11.

Nueva clave: 51348-7

De ésta forma podrían definirse muchas funciones distintas para calcular dígitos de control, pero lo importante es encontrar una que detecte la mayor cantidad de los errores que se cometen.

Para poder identificar la bondad de un método dado, sería necesario el identificar primero los errores que ocurren, y su frecuencia.

Tras un estudio [4,7,12,18] de los errores que pueden ocurrir al trabajar con claves numéricas, se ha llegado a una clasificación que se da en seguida:

1.- Errores de Transcripción.

2.- Errores de Transposición.

3.- Errores de Corrimiento.

4.- Errores por la introducción de caracteres no numéricos a la clave.

5.- Otros errores o errores aleatorios.

Lo cual se explica a continuación.

1.Errores de Transcripción.

Son definidos como el error que se comete cuando uno o varios dígitos de la clave numérica con que se está trabajando, son cambiados por otros dígitos distintos.

Un error de transcripción sencillo es aquel que envuelve sólo un dígito de la clave.

Un error de transcripción múltiple es aquel que envuelve dos o más dígitos de ésta. En este último grupo están incluidos aquellos errores que se producen al cambiar dos o más dígitos consecutivos iguales por otro.

Como ejemplos de este tipo de errores se tienen los siguientes:

Ejemplos:

1.- Clave correcta: 7312165
Clave incorrecta: 7317165
(error de transcripción sencillo).

2.- Clave correcta: 7917819
Clave incorrecta: 7912879

(error de transcripción múltiple).

3.- Clave correcta: 7622514

Clave incorrecta: 7677514

(error de transcripción múltiple en dígitos iguales y consecutivos).

2.Errores de Transposición.

Estos errores son definidos como aquellos que se cometen al intercambiar cualesquiera dos dígitos de una clave.

Puede existir el caso en el que la transposición sea hecha entre dos dígitos que se encuentran juntos, o entre dos que no.

Ejemplos:

1.- Clave correcta: 7124358

Clave incorrecta: 7123458

(error de transposición entre dos dígitos consecutivos).

2.- Clave correcta: 7312165

Clave incorrecta: 7316125

(error de transposición entre dos dígitos no consecutivos).

3.Errores de Corrimiento.

Estos pueden definirse como aquellos errores que se cometen por la inserción de ceros, o de otro digito, en una clave numérica provocando un corrimiento hacia la derecha o la izquierda en el caso de que no se cometa una inserción, sino una supresión.

Ejemplos:

1.- Clave correcta: 56006
Clave incorrecta: 560006
(error de corrimiento por la inserción de ceros).

2.- Clave correcta: 391118
Clave incorrecta: 3918
(error de corrimiento por la supresión de unos).

4.Errores por la introducción de caracteres no numéricos en la clave.

Estos errores se cometen cuando en lugar de un dígito es introducida una letra o un caracter especial en la clave numérica.

Ejemplos:

1.- Clave correcta: 7329981

Clave incorrecta: 7329A81

Nota: Los errores por la introducción de caracteres no numéricos en la clave, no se estudiarán aquí, pues, dependiendo de la instalación y del lenguaje de programación con que se trabaje, pueden ser detectados con cierta facilidad.

5.Otros errores o errores aleatorios.

Dentro de esta clasificación se encuentran todos los errores que resultan de una combinación de los anteriores, y aquellos errores que ocurran al azar.

Como ejemplos se tienen:

1.- Clave correcta: 7125173

Clave incorrecta: 7215178

(error de transposición y error de

transcripción).

2.- Clave correcta: 7125173

Clave incorrecta: 7215713

(dos errores de transposición sencilla).

3.- Clave correcta: 7125173

clave incorrecta: 2175371

(dos errores de transposición múltiple).

De esta misma forma, podrían haber muchas más clasificaciones, como por ejemplo: un error de transcripción múltiple de dos digitos iguales no consecutivos (7312165 a 7342465).

El objetivo de hacer una clasificación de los errores que pueden ocurrir, no es el listar todos los casos posibles, pues pueden resultar bastantes combinaciones, sino clasificar por un lado los que suceden con mayor frecuencia, y así entonces tener un mecanismo para evaluar los diversos métodos de acuerdo a la facilidad que estos nos den para detectar la mayor cantidad de estos errores de acuerdo a dichas frecuencias.

Según pruebas que se han hecho [4,13], se ha podido detectar que los errores que más suceden son los de transcripción sencilla, aunque la frecuencia en que los errores se cometen depende mucho de si las claves numéricas son manejadas en forma oral, si son leídas de un documento escrito en forma manuscrita para después copiarse, y así sucesivamente.

El resultado de una evaluación de los errores más comunes y su frecuencia realizado por Beckley [4] se da a continuación:

Errores de Transcripción Sencilla	:	86%
Errores de Transposición Sencilla	:	8%
Errores de Doble Transposición, Corrimiento, Otros y Aleatorios	:	6%

Esto nos hace concluir que el tipo de errores en los cuales se debe fijar la atención, son los de transcripción, los de transposición como segundo en importancia, sin olvidar los demás que en adelante serán nombrados como aleatorios.

Con estos porcentajes puede establecerse una métrica mediante la cual se pueda hacer una comparación entre distintos métodos. Por ejemplo, si se tiene un método que detecte errores con los siguientes porcentajes: 100%, 90%, 90% para cada tipo de errores respectivamente, resultaría que, en total, éste tendrá un porcentaje de detección de 98.6%, y, de acuerdo a esta métrica, resultará un mejor método que otro que detecte errores con los siguientes porcentajes: 97%, 100%, 100% respectivamente para cada tipo de error, pues ésta tiene en total un porcentaje de detección del 97.42%.

Concluyendo, se van a clasificar los errores en:

1.- Transcripción.

2.- Transposición.

3.- Aleatorios.

donde, en la categoría de aleatorios se incluirán a todos los errores dobles, los de corrimiento, y en general, a todos los no incluídos en las primeras dos clasificaciones.

Nota: Más adelante (Capítulo 2) se regresará a estudiar los errores de transposición dentro de una categoría separada a la de los errores aleatorios.

Hasta éste momento se han identificado los errores más comunes, por lo tanto, puede buscarse ya un método que, en base a esto, dé una mejor solución al problema.

Se han buscado varios métodos para la obtención de los dígitos de los dígitos de control [3,4,7,8,9,15,18]. Uno de ellos es el sistema módulo 11 [4,7,9,15,18]. Este método, como se verá después, tiene porcentajes muy altos de captación de errores, pero tiene el inconveniente de que llega a generar 11 posibles dígitos distintos de control (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10), por lo que surge el problema de cómo almacenar al dígito 10.

Pensando en esto, han surgido algunos sistemas módulo 10.

En particular, el método que se presentará en el siguiente capítulo, genera dígitos del 0 al 9 por ser un sistema módulo 10, y sus porcentajes de captación de errores son bastante buenos. Posteriormente, en el Capítulo 4, se regresará al sistema módulo 11 en el que se expondrán algunas soluciones al dígito de control 10.

CAPITULO 2. Discusión del Caso Módulo 10.

Para atacar el problema de la detección de los errores planteados, se procederá de la siguiente forma:

la clave numérica se representará como "c", que estará dada como:

$$c : N1\ N2\ N3\ \ldots\ Nk$$

donde Ni para toda i en el intervalo [1,k], es un dígito en el sistema decimal (i.e. base 10).

A esta clave se le aplicará una cierta función "f", por medio de la cual se encontrará el dígito verificador "d", que será anexado a la clave original, dando como resultado la clave:

$$c' : N1\ N2\ N3\ \ldots\ Nk\ \ d$$

que será con la que se deberá trabajar para checar si la clave original ha sido escrita correctamente o no. Esto se hará aplicándole la función "f" a "c", y verificando si el dígito generado es igual al que tenga c' ; en la medida en que los errores sean detectados, la función habrá logrado su objetivo.

Es importante hacer notar que un error cometido en el digito de control, se encontrará de forma similar a como uno en cualquier digito de "c".

El problema ahora es encontrar una función "f" tal que detecte el mayor número posible de los errores de transcripción, transposición y aleatorios de acuerdo a lo establecido en el capitulo anterior.

Para lograr el objetivo, y sea más claro el por qué se llegará a la función resultado, se empezará con una función muy sencilla para poco a poco irle poniendo restricciones, hasta llegar a la propuesta de este capitulo.

Se analizará ahora la función f1 que está definida de tal forma que trata por igual a todos los elementos de "c", independientemente de su posición dentro de ésta.

Por ejemplo:

$$\text{Sea} \quad f1(c) = f1(N1\ N2\ N3\ \ldots\ Nk)$$
$$= (N1 + N2 + N3 + \ldots + Nk) \bmod 10$$

Si se observa esta función, se notará que:

$$f1(N1 + N2 + N3 + \dots + Nk) = f1(N1 + N3 + N2 + \dots + Nk)$$

ya que

$$(N1 + N2 + N3 + \dots + Nk) \bmod 10 = (N1 + N3 + N2 + \dots + Nk) \bmod 10$$

Por ejemplo:

$$f1(5932) = 19 \bmod 10 = 9$$

$$f1(9532) = 19 \bmod 10 = 9$$

implica que $f1(5932) = f1(9532)$

Así puede concluirse que cualquier función del tipo de "f1" que tome por igual a cada elemento de "c", traerá el problema de que todo error de transposición cometido no podrá detectarse, sin embargo los de transcripción si. Por ello se puede decir que, para los datos de Beckley,

este método detecta (100%, 0%, 90%), resultando ser el 91.4% como porcentaje total de detección, lo cual, como se verá, no es satisfactorio.

Ahora bien, considérese una función f2, tal que trate en distinta forma a los elementos consecutivos de la clave 'c', es decir, que trate distintamente a los elementos de posición par, de los elementos de posición non de ésta.

Como ejemplo de este tipo de funciones será la función "f2" definida como:

$$f2(c) = f2(N1\ N2\ \dots\ Nk)$$

$$= [\ P\ (N1 + N3 + \dots\ ) + K\ (N2 + N4 + \dots\ )\ ] \bmod 10$$

$$= [\ P\ (N1 + N3 + \dots\ ) \bmod 10 + K\ (N2 + N4 + \dots\ ) \bmod 10\ ] \bmod 10$$

$$= [\ PN1 + KN2 + PN3 + KN4 + \dots\ ] \bmod 10$$

Para que "f2" no sea del tipo de la función "f1" (es decir, que detecte el mayor número de errores de transposición de dos dígitos consecutivos), P y K deben tener valores distintos.

Ahora es necesario definir dichos valores para P y K. Para ello se analizará primero que es lo que ocurre en la clave si un error de transcripción sucede, es decir, se observará cómo afecta este cambio al resultado de calcular el dígito de control "d" sobre la clave correcta y la erronea, para así encontrar aquellos valores de P y K que detecten el máximo número de estos errores.

Supóngase que el término que fue modificado fue el Ni, y que el valor que fue colocado en su lugar fue "x", entonces la diferencia que habrá al calcular el dígito de control será:

$$d = f2(c) = [\ P\ [\ N1 + N3 + \dots\ ] + K\ [\ N2 + N4 + \dots\ ]\ ] \bmod 10$$

$$d' = f2(c') = [\ P\ [\ N1 + \dots + x + \dots] + K\ [\ N2 + N4 + \dots\ ]\ ] \bmod 10$$

en caso de que i fuera un número non.

$$d = [\ PN1 + KN2 + PN3 + KN4 + \dots\ ] \bmod 10$$

$$d' = [\ PN1 + KN2 + PN3 + KN4 + \ldots + Px + \ldots\ ] \bmod 10$$

$$d - d' = P\ [\ Ni - x\ ] \bmod 10$$

Ahora bien, se quiere que P por cualquier valor Ni dé un resultado distinto de multiplicar P por otro valor, es decir, que si se comete un error por cambiar un número por otro, éste sea detectado. Lo mismo se quiere para K, puesto que la detección de errores se quiere tanto en las posiciones pares como en las nones.

Entonces, cómo deben ser P y K para que al multiplicarse por dígitos distintos den como resultado valores distintos módulo 10 ?

Para ejemplificar algunos valores "buenos" y otros "malos", se verá el proceso con los siguientes valores para P y K:

$P = 2$

$K = 5$

Clave correcta: c: 7312165

Según la formula dada anteriormente para calcular el dígito de control por medio de f2, se tiene:

$$d = f2(c) = f2(7312165)$$

$$= [\ 2\ (7 + 1 + 1 + 5) + 5\ (3 + 2 + 6)]\ \bmod 10$$

$$= [\ 2\ (14) + 5\ (11)\ ]\ \bmod 10$$

$$= [\ 28 + 55\ ] \bmod 10$$

$$= [83] \bmod 10$$

implica que $d = 3$

Entonces, se le añadirá este dígito a la clave "c", para tener:

7312165-3

Ahora, supóngase que esa clave es erroneamente escrita de la forma:

$$c' = 7352165\text{-}3$$

donde el dígito erróneo es el tercero de izquierda a derecha,

Para checar si está bien o mal escrita, se debe calcular el dígito de control de esta última, sin tomar en cuenta el dígito de control que trae, y entonces, checar si estos dos son iguales.

Es decir:

$$d' = f2(c') = f2(7352165)$$

$$= [\ 2(7+5+1+5) + 5(3+2+6)\ ] \bmod 10$$

$$= [\ 36 + 55\ ] \bmod 10 = [91] \bmod 10$$

implica que $d' = 1$ y por lo tanto d es distinta de d'

ya que 3 es distinto de 1

De ello se ve que este error si fue detectado, y por lo tanto se podría concluir erroneamente que los valores 2 y 5 son "buenos" para encontrar estos errores, lo cual no es cierto, ya que en el ejemplo el dígito fue cambiado de un valor 1 a uno 5, y era multiplicado por P = 2. Se verá ahora qué pasaría si el error hubiera sido el transcribir ese mismo dígito con valor 1 a un valor 6.

c" : 7362165

d" = f2(7362165) = [93] mod 10 = 3

implica que d = d", no logrando así los fines buscados.

Para entender mejor esto y visualizar con qué cambios de dígitos no se detectarán los errores de transcripción con los valores dados de P y K, se hará uso de una tabla de 10 X 10, en la cual las columnas denotarán la suma de los valores de los dígitos que se encuentran en posición non módulo 10, y los renglones, la suma de los valores de los dígitos que se encuentran en posición par módulo 10.

		SUMA POSICIONES NONES MOD 10									
		1	2	3	4	5	6	7	8	9	0
SUMA POSICIONES PARES MOD 10	1										
	2										
	3										
	4										
	5										
	6										
	7										
	8										
	9										
	0										

TABLA 1

Por lo tanto, el cuadrito (x,y) será accesado cuando la suma de los valores de los digitos en posición par mod 10 sea "x", y la suma de los valores de los digitos en posición non mod 10 sea "y".

El valor que se va a tener en el cuadrito (x,y), será el del digito de control que se encontrará al aplicarar la función f2 con P = 2

y K = 5 a los valores "x" y "y".

En el ejemplo anterior, los valores "x" y "y" para la clave correcta fueron:

$$c: 7312165$$

Suma posiciones pares = $(3+2+6) \bmod 10 = 11$ módulo $10 = 1 = x$

Suma posiciones nones = $(7+1+1+5) \bmod 10 = 14$ módulo $10 = 4 = y$

$$(x,y) = (1,4)$$

$$d = 3$$

Por lo que el cuadrito (1,4) deberá tener el valor 3. De esta forma se genera la siguiente tabla (sin olvidar que es para valores P=2 y K=5).

SUMA POSICIONES NONES MOD 10

SUMA POSICIONES PARES MOD 10	1	2	3	4	5	6	7	8	9	0
1	7	9	1	3	5	7	9	1	3	5
2	2	4	6	8	0	2	4	6	8	0
3	7	9	1	3	5	7	9	1	3	5
4	2	4	6	8	0	2	4	6	8	0
5	7	9	1	3	5	7	9	1	3	5
6	2	4	6	8	0	2	4	6	8	0
7	7	9	1	3	5	7	9	1	3	5
8	2	4	6	8	0	2	4	6	8	0
9	7	9	1	3	5	7	9	1	3	5
0	2	4	6	8	0	2	4	6	8	0

TABLA 2

Se seguirá analizando este ejemplo para observar como cambian los valores "x" y "y" cuando sucede un error, y cómo desplazarse en la tabla anterior al suceder esto.

Para c': 7352165 se encontró que la suma de los dígitos en posición par módulo 10 es : x'= 1, la suma de los dígitos en posición non módulo 10 es : y'= 8 y que d'= 1.

Regresando a la tabla puede observarse que para el valor (x' y'

) = (1,8) el valor en la tabla es efectivamente 1, lo cual no es raro pues la tabla ha sido generada de igual forma que los digitos de control. Lo importante aquí es observar que al ocurrir un error de transcripción, lo que sucedió fue que el desplazamiento se hizo (en este ejemplo) sobre el mismo renglón de "d", es decir, sobre el valor x = x'pero lo que cambió fue la columna, es decir de "y" a "y'".

La pregunta sería: por qué no cambió el valor de "x" y si el de "y" ? Y por qué el desplaziamiente fue de 4 casillas hacia la derecha?

No cambió el valor de "x", pues el error ocurrió en una posición non, por lo que la suma de las posiciones pares (x) no tenia por que alterarse. Cambió el valor "y", pues ésta es la suma de las posiciones nones que es donde hubo un cambio.

Regresando la atención a la cantidad que al principio de la discusión se hizo notar:

$$F\ (Ni - x) \bmod 10$$

Puede observase que para el ejemplo dado se tiene:

$$2\ [1 - 5] \bmod 10$$

$$= 2[-4] \bmod 10 = -8 \bmod 10 = 2$$

que es efectivamente la diferencia entre "d" y "d'".

El que el desplazamiento se dé por las columnas o por los renglones dependerá de que el cambio o error ocurra en las posiciones nones o en las pares.

Es fácil de notar que no pueden cambiar al mismo tiempo "x" y "y" con un error de transcripción sencillo, en el cual solo cambia el valor de un solo dígito.

Viendo esto serán de interés aquellos valores de P y K que den como resultado el que no se repitan los valores ni en cada columna, ni en cada renglón, para que el valor del dígito de control cambie, y así sea detectado el error.

Regresando la atención a la Tabla 2, se puede ver que es "mala", pues muchísimos errores no serán detectados ya que cada valor en cada renglón se repite dos veces, y en cada columna solo hay dos valores distintos, por lo que cada uno se repite 5 veces.

Tal es el caso, en el mismo ejemplo, de lo que sucede con c" : 7362165, en la que:

$$x" = 11 \bmod 10 = 1$$

$$y" = 19 \bmod 10 = 9$$

en el cual no es detectado el error, pues:

$$(x,y) = (1,4) = 3 = (1,9) = (x',y')$$

lo mismo sucederá con:

$$7 = (1,1) = (3,1) = (5,1) = \ldots$$

y así sucesivamente si se observa la tabla.

De este caso donde se utilizaron los valores 2 y 5 para P y K respectivamente, se puede concluir que aquellos valores que no sean primos relativos de 10 [14], no sirven en el propósito de detectar los errores de transcripción. Entonces, el conjunto de valores: { 1, 3, 7, 9 } serán los únicos que servirán para ser tomados como valores de P y K dadas las premisas del problema.

Se tomarán ahora los valores P = 3 y K = 7 y se formará la Tabla 3 de la misma forma que se creó la Tabla 2, para ver si no se repiten los valores en las columnas y en los renglones.

SUMA POSICIONES NONES MOD 10

SUMA POSICIONES PARES MOD 10	1	2	3	4	5	6	7	8	9	0
1	0	3	6	9	2	5	8	1	4	7
2	7	0	3	6	9	2	5	8	1	4
3	4	7	0	3	6	9	2	5	8	1
4	1	4	7	0	3	6	9	2	5	8
5	8	1	4	7	0	3	6	9	2	5
6	5	8	1	4	7	0	3	6	9	2
7	2	5	8	1	4	7	0	3	6	9
8	9	2	5	8	1	4	7	0	3	6
9	6	9	2	5	8	1	4	7	0	3
0	3	6	9	2	5	8	1	4	7	0

TABLA 3

Es fácil detectar que ningún digito de control se repite en todos los renglones y en todas las columnas. Por lo que todos los errores de transcripción se detectarán para los valores P = 3 y K = 7.

Así se ha solucionado el problema de detectar todos los errores de transcripción que era muy importante, pues son los que suceden con mayor frecuencia. Ahora sería bueno observar con el mismo sistema, cuántos errores de transposición se detectan y cómo utilizar la misma tabla en este estudio para que sea igual de fácil el darse cuenta de esto visual y practicamente.

Qué sucede cuando un error de transposición de dos digitos consecutivos ocurre?

Supóngase que los términos Ni y Ni+1 son intercambiados; por lo que, si se tenía:

$$d = [PN1 + KN2 + PN3 + \dots + PNi + KNi+1 + \dots] \bmod 10$$

$$y\ d' = [PN1 + KN2 + PN3 + \dots + PNi+1 + KNi + \dots] \bmod 10$$

como resultado de intercambiar Ni y Ni+1, entonces la diferencia entre el valor real y el erroneo será:

$$(d - d') \bmod 10 = ((Ni - Ni+1)P + (Ni+1 - Ni)K) \bmod 10$$

$$= ((Ni - Ni+1)\ (P - K)) \bmod 10$$

Cómo deben ser los valores de P y K para que este valor sea distinto de cero y por lo tanto los errores de transposición sean detectados ?

Se verá esto con un ejemplo:

Clave correcta : c : 7132165

Suma digitos en posición par : x : 09 mod 10 = 9

Suma digitos en posición non : y : 16 mod 10 = 6

$$d = (x,y) = (9,6) = 1$$

Supóngase que ocurre un error de transposición, o sea que "c" se convierte por ejemplo en c': 7123165.

Cuál será el digito de control para c'con los valores P = 3 y K = 7 ? Basta con que se calcule la suma de los valores de los digitos de las posiciones pares y la suma de los valores de los digitos de las posiciones nones para entonces buscar el valor del digito que le corresponde en la Tabla 3.

c' : 7123165

Suma digitos en posición par mod 10 : $x = (6+3+1) \bmod 10 = 0$

Suma digitos en posición non mod 10 : $y = (5+1+2+7) \bmod 10 = 5$

$$d' = (x',y') = 5$$

Si se busca el valor que le corresponde a (x'y')para d' se encontrará que d = 1 distinto de d'= 5.

Por lo que este error si es detectado para los valores dados de P y K.

Se verá ahora en la Tabla 4 cómo es el traslado cuando un error de transposición ocurre.

SUMA POSICIONES NONES MOD 10

SUMA POSICIONES PARES MOD 10

	1	2	3	4	5	6	7	8	9	0
1										
2										
3										
4										
5										
6										
7										
8										
9						1				
0					5					

TABLA 4

Con el valor correcto para la clave "c" se encuentran los valores: $(x,y) = (9,6) = 1$.

Al trasponer los dos digitos ya indicados antes, se observó un traslado que puede verse en la Tabla 4.

$$d' = (x', y') = (0,5) = 5$$

Según la figura, y los valores obtenidos de x, y, $x'y'$ pueden verse dos cosas:

Tanto el valor de "x" como el de "y" fueron modificados, es decir, al ocurrir un error de transposición de dos dígitos consecutivos, se modifica la suma de los dígitos en posición non, y la suma de los dígitos en posición par, lo cual era de esperarse.

Es importante notar que estos valores se incrementan o decrementan en la misma cantidad, por lo que si ocurre un desplazamiento de un renglón hacia abajo, al mismo tiempo se hace una columna hacia la izquierda.

La suma de los valores nones se decrementó en 1, por lo que la suma de los valores pares se incrementó en 1. De aquí puede observarse que cada vez que una transposición decremente en N (donde N es un número entero del intervalo [1,9]) la suma de los valores que se encuentran en posición non módulo 10, esto provocará un incremento de N a la suma de los valores que se encuentran en posición par módulo 10. Esto mismo sucederá a la inversa.

Se verá ahora un ejemplo en el que la suma de los valores de los dígitos de las posiciones nones se ha incrementado, para observar esto en la tabla.

Clave correcta : c : 5526351

$x = 16 \bmod 10 = 6$

$y = 11 \bmod 10 = 1$

$d = (x,y) = (6,1) = 5$

Supóngase que el error de transposición ocurre entre los dígitos "6" y "3" generándose la clave: 5523651, por lo que:

Clave incorrecta: c' : 5523651

x' : 13 mod 10 = 3

y' : 14 mod 10 = 4

d' : (x',y') = (3,4) = 3

Puede observarse ahora el traslado ocurrido en "x" y "y" a "x'" y "y'" en la Tabla 5 que se ilustra a continuación:

SUMA POSICIONES NONES MOD 10

SUMA POSICIONES PARES MOD 10	1	2	3	4	5	6	7	8	9	0
1										
2										
3				3						
4				¦						
5				¦						
6	5	-	-	-						
7										
8										
9										
0										

TABLA 5

Como se ve, debido a que el valor de "x" se decrementó en 3, ello implicó que el valor de "y" se incrementara en esos 3 exactamente.

Puede entonces observarse que cuando se comete un error de transposición (error del tipo II) el dígito que le corresponderá a la clave erronea será d'= (x'y'), donde x'= (x+a) mod 10 y y'= (y-a) mod 10, donde "a" es un entero positivo o negativo distinto de cero. Es decir, un error de este tipo implica un movimiento dentro de la diagonal derecha módulo 10. Puede observarse en la figura siguiente las 10 diagonales derechas que pueden existir en la tabla.

SUMA POSICIONES NONES MOD 10

SUMA POSICIONES PARES MOD 10	D2	D3	D4	D5	D6	D7	D8	D9	D0	D1	
1											D1
2											D2
3											D3
4											D4
5											D5
6											D6
7											D7
8											D8
9											D9
0											D0

TABLA 6

Estas diagonales pueden ser visualizadas de mejor forma si se escribe la Tabla 6 dos veces consecutivas, una debajo de la otra, pues así la diagonal no tiene que ser interrumpida, esto es:

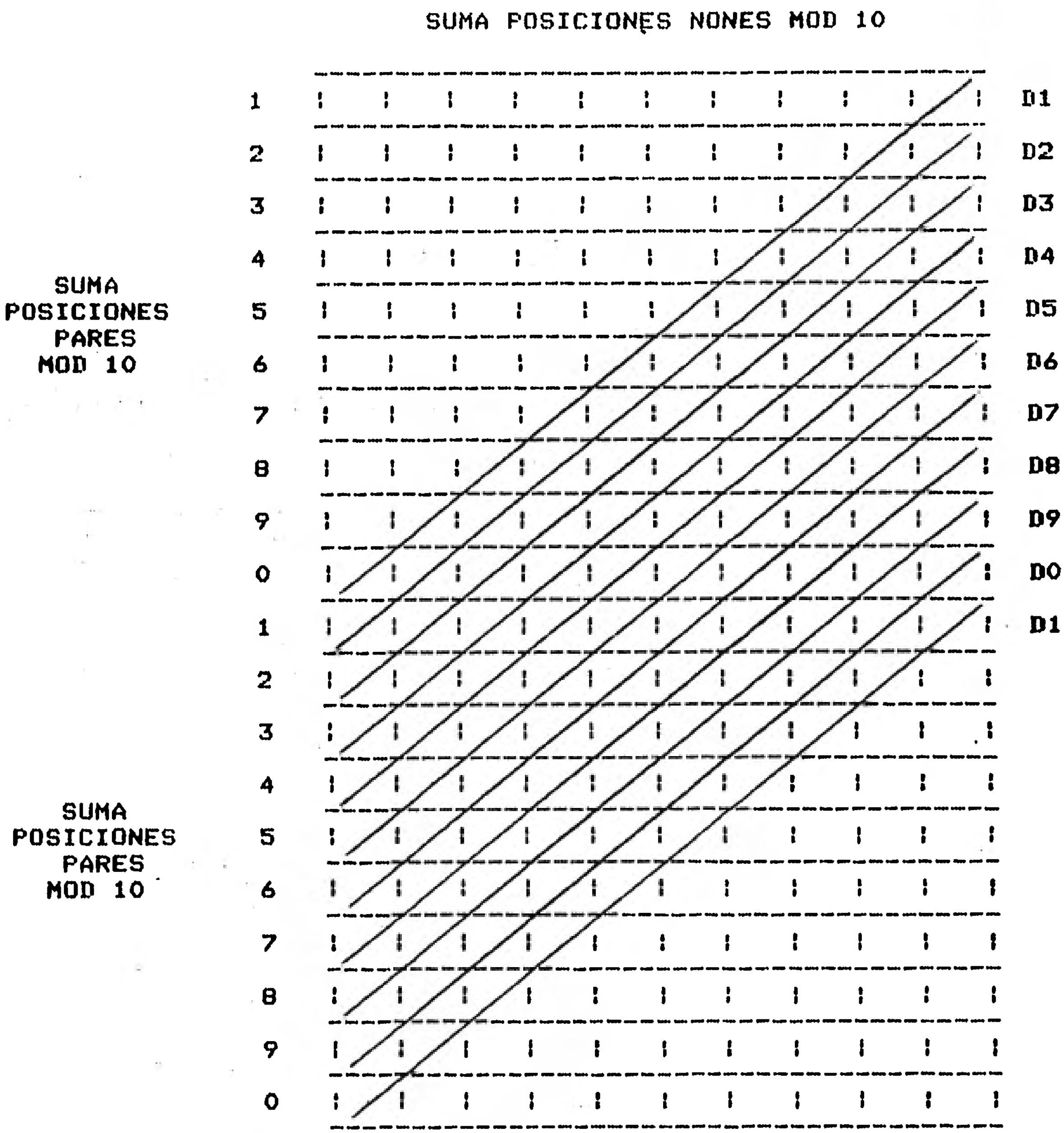
SUMA POSICIONES NONES MOD 10
SUMA POSICIONES PARES MOD 10
SUMA POSICIONES PARES MOD 10
1
2
3
4
5
6
7
8
9
0
1
2
3
4
5
6
7
8
9
0
D1
D2
D3
D4
D5
D6
D7
D8
D9
D0
D1

TABLA 7

Ahora bien, tal como se hizo en el caso de los errores del tipo I (errores de transcripción), en el cual se buscó una función "f" tal que al ser aplicada a la clave diera un dígito distinto de aplicar esa misma función a la misma clave con un error de transposición, se observó que esto era lo mismo que encontrar una cierta tabla tal que un mismo número no se repitiera ni en cada columna, ni en cada renglón. Esta tabla se encontró (Tabla 3) al trabajar con la función "f" definida como:

$$f : f(c) = [\ P(N1+N3+...) + K(N2+N4+...)\ ] \bmod 10$$

$$\text{con } P = 3 \text{ , } K = 7$$

Lo que ahora se quisiera encontrar, para poder detectar todos los errores del tipo II, sería una tabla tal que un mismo número no se repita ni en cada renglón, ni en cada columna, ni en cada diagonal derecha.

Esta tabla no es posible de encontrar cuando se está trabajando con una función módulo M para M par.(Tanto este caso, como en particular el caso M = 10 son demostrados en el Apéndice 1 y 2).

El caso que se trata aquí es M = 10, por lo tanto no se podrá encontrar una tabla tal que detecte todos los errores del tipo I y todos los errores del tipo II.

Por esta limitación con el presente método, serán detectados el 100% de los errores de transcripción, el 88.8% de los errores de transposición, y el 90% de los aleatorios. El 11.1% de los errores de transposición no detectados, es debido a aquellos casos en que sucede un error de este tipo y la diferencia entre las dos cifras transpuestas es de 5 módulo 10; en cada 1 de 9 casos esto sucede. Es un 90% el porcentaje de los errores aleatorios detectados pues, dada una clave cualesquiera, cuál es la probabilidad de que si se le añade un digito al azar, sea exactamente el que le correspondería a ésta si se le aplica la función dada anteriormente?

Esta probabilidad es de 1/10 pues son 10 los distintos digitos posibles, y cada uno tiene la misma probabilidad de suceder. De aquí que sea el 90% el porcentaje de errores aleatorios que se captarán (ver Tabla 3).

De usarse los resultados de Beckley mostrados anteriormente (Cap 1), este método detectará el 98.51% total de errores.

A continuación se presentará una pequeña ampliación de este método, en la cual se estudiará un tipo de error, que es el de transposición entre dos digitos no consecutivos, que en el estudio anterior quedó englobado dentro de la categoría de los aleatorios.

Se hace esto, pues existen ciertas claves formadas por ejemplo de la siguiente manera:

01 05 03

donde cada pareja de dígitos va a ser similar a las demás, tal vez porque cada una de éstas hace referencia a ciertas cosas determinadas:

01 05 03

depto sub oficina

depto

En este tipo de claves va a ser más frecuente la ocurrencia de un error de transposición de la forma:

01 03 05

es decir, un error de transposición de dos dígitos no consecutivos.

Si se tratara de claves de este estilo, y se ocupara un sistema de dígitos de control con dos pesos, uno para las posiciones pares, y otro para las nones; estos errores no serían detectados, siendo probable que sucedieran.

Para esto se propondrá que en lugar de tener una función que tome por separado solo las posiciones pares y nones, se tenga uno que tome tres tipos de posiciones.

Es decir, si se tiene la clave: 7312165 se numerarán los dígitos que la componen del 1 al 3 de la siguiente forma:

CLAVE : 7 3 1 2 1 6 5

POSICIONES : 1 3 2 1 3 2 1

de esta forma, los valores: 7, 2, y 5 ocupan posiciones "1"; los valores: 1, y 6 ocupan posiciones "2"; y los valores: 3, y 1 ocupan posiciones "3".

El dígito verificador será encontrado de la siguiente forma:

d = {P[suma pos."1"] + K[suma pos."2"] + R[suma pos."3"]}mod 10

Como se vió anteriormente, los valores que podían tomar P y K, para que todos los errores de transcripción fueran detectados, eran del conjunto: {1, 3, 7, 9} por ser primos relativos de 10; ahora bien, de este mismo conjunto de valores pueden tomarse tres para P, K y R y garantizarán el 100% de detección de estos errores, pues siguen el mismo razonamiento.

Sea : P = 3, K = 7, R = 9 .

Si se substituyen estos valores en la fórmula anterior, se tendrá:

d = {3[suma pos."1"] + 7[suma pos."2"] + 9[suma pos."3"]}mod 10

EJemplo:

Clave : 7312165

suma pos."1" mod 10 = 14 mod 10 = 4

suma pos."2" mod 10 = 07 mod 10 = 7

suma pos."3" mod 10 = 04 mod 10 = 4

por lo tanto: $d = [3(4) + 7(7) + 9(4)] \bmod 10$

$= [12 + 49 + 36] \bmod 10$

implica que: $d = 7$

Al igual que en el caso de dos pesos, se presentará a continuación una serie de tablas en las que las columnas representarán la suma de los valores en posición "1", los renglones representarán la suma de los valores en posición "2", y la profundidad representará la suma de los valores en posición "3".

Este punto de la profundidad debe visualizase de la siguiente forma: cuando la suma de los valores en posición "3" es igual al 1, recurrir a la tabla 8; cuando es 2, recurrir a la tabla 9; y así sucesivamente hasta la tabla 17.

Estas tablas se dan a continuación:

SUMA POSICIONES "1" MOD 10

SUMA POSICIONES "2" MOD 10

	1	2	3	4	5	6	7	8	9	0
1	9	2	5	8	1	4	7	0	3	6
2	6	9	2	5	8	1	4	7	0	3
3	3	6	9	2	5	8	1	4	7	0
4	0	3	6	9	2	5	8	1	4	7
5	7	0	3	6	9	2	5	8	1	4
6	4	7	0	3	6	9	2	5	8	1
7	1	4	7	0	3	6	9	2	5	8
8	8	1	4	7	0	3	6	9	2	5
9	5	8	1	4	7	0	3	6	9	2
0	2	5	8	1	4	7	0	3	6	9

SUMA POSIC "3" MOD 10 = 1

TABLA 8

SUMA POSICIONES "1" MOD 10

SUMA POSICIONES "2" MOD 10

	1	2	3	4	5	6	7	8	9	0
1	8	1	4	7	0	3	6	9	2	5
2	5	8	1	4	7	0	3	6	9	2
3	2	5	8	1	4	7	0	3	6	9
4	9	2	5	8	1	4	7	0	3	6
5	6	9	2	5	8	1	4	7	0	3
6	3	6	9	2	5	8	1	4	7	0
7	0	3	6	9	2	5	8	1	4	7
8	7	0	3	6	9	2	5	8	1	4
9	4	7	0	3	6	9	2	5	8	1
0	1	4	7	0	3	6	9	2	5	8

SUMA POSIC "3" MOD 10 = 2

TABLA 9

SUMA POSICIONES "1" MOD 10

SUMA POSICIONES "2" MOD 10

	1	2	3	4	5	6	7	8	9	0
1	7	0	3	6	9	2	5	8	1	4
2	4	7	0	3	6	9	2	5	8	1
3	1	4	7	0	3	6	9	2	5	8
4	8	1	4	7	0	3	6	9	2	5
5	5	8	1	4	7	0	3	6	9	2
6	2	5	8	1	4	7	0	3	6	9
7	9	2	5	8	1	4	7	0	3	6
8	6	9	2	5	8	1	4	7	0	3
9	3	6	9	2	5	8	1	4	7	0
0	0	3	6	9	2	5	8	1	4	7

SUMA POSIC "3" MOD 10 = 3

TABLA 10

SUMA POSICIONES "1" MOD 10

SUMA POSICIONES "2" MOD 10

	1	2	3	4	5	6	7	8	9	0
1	6	9	2	5	8	1	4	7	0	3
2	3	6	9	2	5	8	1	4	7	0
3	0	3	6	9	2	5	8	1	4	7
4	7	0	3	6	9	2	5	8	1	4
5	4	7	0	3	6	9	2	5	8	1
6	1	4	7	0	3	6	9	2	5	8
7	8	1	4	7	0	3	6	9	2	5
8	5	8	1	4	7	0	3	6	9	2
9	2	5	8	1	4	7	0	3	6	9
0	9	2	5	8	1	4	7	0	3	6

SUMA POSIC "3" MOD 10 = 4

TABLA 11

SUMA POSICIONES "1" MOD 10

SUMA POSICIONES "2" MOD 10

SUMA POSIC "3" MOD 10 = 5

	1	2	3	4	5	6	7	8	9	0
1	5	8	1	4	7	0	3	6	9	2
2	2	5	8	1	4	7	0	3	6	9
3	9	2	5	8	1	4	7	0	3	6
4	6	9	2	5	8	1	4	7	0	3
5	3	6	9	2	5	8	1	4	7	0
6	0	3	6	9	2	5	8	1	4	7
7	7	0	3	6	9	2	5	8	1	4
8	4	7	0	3	6	9	2	5	8	1
9	1	4	7	0	3	6	9	2	5	8
0	8	1	4	7	0	3	6	9	2	5

TABLA 12

SUMA POSICIONES "1" MOD 10

SUMA POSICIONES "2" MOD 10

SUMA POSIC "3" MOD 10 = 6

	1	2	3	4	5	6	7	8	9	0
1	4	7	0	3	6	9	2	5	8	1
2	1	4	7	0	3	6	9	2	5	8
3	8	1	4	7	0	3	6	9	2	5
4	5	8	1	4	7	0	3	6	9	2
5	2	5	8	1	4	7	0	3	6	9
6	9	2	5	8	1	4	7	0	3	6
7	6	9	2	5	8	1	4	7	0	3
8	3	6	9	2	5	8	1	4	7	0
9	0	3	6	9	2	5	8	1	4	7
0	7	0	3	5	9	2	5	8	1	4

TABLA 13

SUMA POSICIONES "1" MOD 10

SUMA POSICIONES "2" MOD 10

	1	2	3	4	5	6	7	8	9	0
1	3	6	9	2	5	8	1	4	7	0
2	0	3	6	9	2	5	8	1	4	7
3	7	0	3	6	9	2	5	8	1	4
4	4	7	0	3	6	9	2	5	8	1
5	1	4	7	0	3	6	9	2	5	8
6	8	1	4	7	0	3	6	9	2	5
7	5	8	1	4	7	0	3	6	9	2
8	2	5	8	1	4	7	0	3	6	9
9	9	2	5	8	1	4	7	0	3	6
0	6	9	2	5	8	1	4	7	0	3

SUMA POSIC "3" MOD 10 = 7

TABLA 14

SUMA POSICIONES "1" MOD 10

SUMA POSICIONES "2" MOD 10

	1	2	3	4	5	6	7	8	9	0
1	2	5	8	1	4	7	0	3	6	9
2	9	2	5	8	1	4	7	0	3	6
3	6	9	2	5	8	1	4	7	0	3
4	3	6	9	2	5	8	1	4	7	0
5	0	3	6	9	2	5	8	1	4	7
6	7	0	3	6	9	2	5	8	1	4
7	4	7	0	3	6	9	2	5	8	1
8	1	4	7	0	3	6	9	2	5	8
9	8	1	4	7	0	3	6	9	2	5
0	5	8	1	4	7	0	3	6	9	2

SUMA POSIC "3" MOD 10 = 8

TABLA 15

SUMA POSICIONES "1" MOD 10

SUMA POSICIONES "2" MOD 10 (rows); SUMA POSIC "3" MOD 10 = 9

	1	2	3	4	5	6	7	8	9	0
1	1	4	7	0	3	6	9	2	5	8
2	8	1	4	7	0	3	6	9	2	5
3	5	8	1	4	7	0	3	6	9	2
4	2	5	8	1	4	7	0	3	6	9
5	9	2	5	8	1	4	7	0	3	6
6	6	9	2	5	8	1	4	7	0	3
7	3	6	9	2	5	8	1	4	7	0
8	0	3	6	9	2	5	8	1	4	7
9	7	0	3	5	9	2	5	8	1	4
0	4	7	0	3	6	9	2	5	8	1

TABLA 16

SUMA POSICIONES "1" MOD 10

SUMA POSICIONES "2" MOD 10 (rows); SUMA POSIC "3" MOD 10 = 10

	1	2	3	4	5	6	7	8	9	0
1	0	3	6	9	2	5	8	1	4	7
2	7	0	3	6	9	2	5	8	1	4
3	4	7	0	3	6	9	2	5	8	1
4	1	4	7	0	3	6	9	2	5	8
5	8	1	4	7	0	3	6	9	2	5
6	5	8	1	4	7	0	3	6	9	2
7	2	5	8	1	4	7	0	3	6	9
8	9	2	5	8	1	4	7	0	3	6
9	6	9	2	5	8	1	4	7	0	3
0	3	6	9	2	5	8	1	4	7	0

TABLA 17

Estas tablas fueron generadas con la fórmula antes mencionada para generar el dígito de control, y tienen todos los valores posibles que pueden tomar la suma de los valores en posición "1" mod 10, la suma de los valores en posición "2" mod 10, y la suma de los valores en posición "3" mod 10.

Para hacer esto más claro, se verá a continuación un ejemplo.

Sea la clave:

7 3 1 2 1 6 5

1 3 2 1 3 2 1

y sean:

x : Suma posic "1" mod 10

y : Suma posic "2" mod 10

z : Suma posic "3" mod 10

entonces:

$x = 7 + 2 + 5 = 14 \bmod 10 = 4$

$y = 1 + 6 = 7 \bmod 10 = 7$

$z = 3 + 1 = 4 \bmod 10 = 4$

$(x, y, z) = (4, 7, 4)$

Este valor puede encontrarse en la tabla 11, y da el valor de 7 para "d", que es el mismo que se había encontrado anteriormente.

Se analizará ahora qué es lo que ha pasado con la detección de los errores.

Errores de Transcripción Sencillos.

Como se vió anteriormente en la relación que se hizo de la ocurrencia de estos errores con la tabla generada con los dos pesos P y K, era necesario que ningún valor se repitiera ni en cada columna, ni en cada renglón, para que estos errores se detectaran en un 100%. La tabla que se generó tenía esta característica, por lo que estos errores en este método se detectaban en un 100%.

Con la ampliación que se propone aquí de éste, en la cual se usaron tres pesos distintos en lugar de dos, va a ser necesario que ningún valor se repita ni en cada columna, ni en cada renglón, ni en cada cuadrito hacia su profundidad. Es decir, si se ha dicho que se tiene:

(x, y, z)

donde

x: suma valores en posición "1"

y: suma valores en posición "2"

z: suma valores en posición "3"

Se quiere que al fijar dos valores cualesquiera del intervalo [0,9] a cualquier combinación de dos de estas variables, y hacer que la otra recorra todos los valores dentro de ese mismo intervalo, los cuadritos accesados tengan, cada uno, valores distintos.

Esto si sucede en las tablas dadas anteriormente (ver tablas 8 - 17) y puede verificarse facilmente.

Esto era de esperarse, pues los valores usados como pesos (3, 7, 9) son todos primos relativos de 10.

De esta forma puede decirse que con este **método** de tres pesos, se detectan el 100% de los errores del tipo I.

Errores de Transposición Sencilla.

De igual forma, se ha visto que para que un error de este tipo sea detectado, es necesario que ningún valor se repita en las diagonales derechas que forman la tabla dada. Se llegó a que como se trataba con una función módulo N para N par esto no iba a suceder, y por ello se tendría uno que conformar con detectar el 88.8% de los errores de este tipo.

En el caso que se trata ahora, al ocurrir un error de transposición sencillo, será de alguna de las siguientes formas:

Entre un digito de una posición "1" y una "2"
Entre un digito de una posición "2" y una "3"
Entre un digito de una posición "3" y una "1"

Como se ve, en cualquiera de estas tres formas se tendrá el mismo caso que con el método anterior, pues unicamente se implican en el error dos pesos. Ahora bien, como los pesos tomados son como ya se ha dicho primos relativos de 10, y como la función módulo N es para N par, entonces se puede concluir que con este método se captarán el mismo 88.8% de los errores del tipo II.

Errores Aleatorios.

Dentro de esta clasificación se englobaron a todos los errores que no eran del tipo I y II, pues como se vió, no era importante tratarlos en un estudio separado, ya que ocurren con muy poca frecuencia en comparación con los otros.

Como se hizo la aclaración en el método anterior (dos pesos), los errores de transposición de dos digitos no consecutivos que ocurren al ser intercambiados dos digitos que se encuentran ambos en posición par o non, no son detectados.

Con el presente método (tres pesos distintos), los errores de transposición de dos digitos consecutivos (como ya se vió) serán detectados en un 88.8%, los errores de transposición que se encuentran separados por uno solo también serán detectados en un 88.8%.

Generalizando, los errores de transposición que no serán detectados serán aquellos que engloben en el intercambio dos digitos que se encuentren ambos en posición "1", "2" o "3".

Por esto, resultaría conveniente para detectar el mayor número posible de errores de transposición no sencilla el tener el mayor número posible de pesos para asignarle a la clave.

Por estar trabajando módulo 10, solo se cuenta con cuatro posibles valores que son: 1, 3, 7 y 9; ya que se quiere que los porcentajes de detección de los errores del tipo I y II no sean modificados por aumentar un porcentaje de detección de un tipo de errores que suceden con menor frecuencia que estos.

De esta forma se podría tener como función para encontrar el dígito de control, la siguiente:

d = (P suma posic "1" + K suma posic "2" +
R suma posic "3" + S suma posic "4") mod 10

Se propone que el valor que se le asigne a P no sea ni 1 ni 9, porque dado que la numeración de las posiciones se hace de derecha a izquierda, a P le corresponde el valor de la extrema derecha, entonces si P valiera 1, sucedería que a ciertas claves consecutivas les corresponderían dígitos consecutivos, (para el valor 9 sería igual pero en forma descendente), cosa que no es muy conveniente pues puede acarrear desiciones erroneas para la asignación del valor del dígito de control por parte de personas que no conoscan la mecánica de la formación de éste, y que por otra parte tienen que manejar dichas claves.

Una posible combinación de los valores que pueden tomar, será:

$$P = 3$$

$$K = 9$$

$$R = 7$$

$$S = 1$$

En general, se sugiere ocupar el sistema de dígitos de control estudiado con cuatro pesos distintos, salvo que por alguna naturaleza intrínseca de las claves, la posibilidad de errores de triple transposición sea mayor que la de doble transposición o cuádruple, o por causas similares.

CAPITULO 3. Discusión del Caso Módulo Menor que 10.

En el presente capítulo se tratará el caso módulo N menor que 10, pretendiendo analizar la conveniencia de su uso, en vez del módulo 10.

Para este efecto, se definirá formalmente cómo está compuesta una clave y qué nomenclatura se usará.

Sea una clave "c" representada como:

$$c:\ N1\ N2\ N3\ \ldots\ Nk$$

Se cuenta en el método propuesto con una función para encontrar los dígitos de control, de la siguiente forma:

$$f(c) = [\ P(N1+N3+\ldots)+K(N2+N4+\ldots)\ ] \bmod N$$

donde Ni es un número entero del intervalo [0,9], para toda "i" del intervalo [1,k], P y K menores que N, y N menor que 10.

Si un error del tipo | ocurre, entonces la diferencia entre la cantidad real y la erronea será de:

$$P\ (\ Ni - x\ )\ \bmod\ N$$

Suponiendo que el error ocurrió en el lugar "i", para "i" non. (Sería K(NJ - x) si se tratara de un lugar par, pero como el razonamiento es el mismo, se ilustrará unicamente el caso en que "i" es non).

Se quiere que esta diferencia sea detectada, es decir, que esta cantidad módulo N no sea igual a cero.

Para que sea detectada, tinen que pasar:

1) (Ni - x) no sea divisible por N,

2) P no sea divisible por N,

o 3) P(Ni - x) no sea divisible por N.

1) Como x,Ni están en el intervalo [0,9] y "x" es distinta de "Ni", ello implica que (Ni-x) mod N va a estar en el intervalo [1,9] mod 10, por lo que si N es menor que 10, resultará que para ciertas combinaciones de Ni y de "x", esta diferencia módulo N dará el valor 0(cero), que multiplicado por cualquier valor dará 0(cero), y consecuentemente, la cantidad P(Ni-x) dará 0(cero).

No es necesario ya analizar los puntos 2 y 3, pues ya en el punto 1 se encontró que un gran número de errores del tipo I no serán detectados.

Por esto, un sistema módulo N, con N menor que 10, no funciona cuando el código está formado por valores decimales, y será abandonado para efectos futuros de esta tesis. Posibles ampliaciones a ésta, podrían regresar a extender este estudio a módulo N menor que 10 y bases menores a 10, donde la conclusión anterior no es válida.

CAPITULO 4. Sistemas Módulo 11. Ventajas y Desventajas.

Los sistemas de dígitos de control módulo 11, como su nombre lo indica, emplean la función módulo 11 para generar los dígitos de control; por esto, los dígitos generados serán cualesquiera del conjunto: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10}.

A continuación se hará un estudio empleando la función f = 7x+3y donde "x" representa la suma de los valores que se encuentran en posición par, y "y" la suma de los valores que se encuentran en posición non, módulo 11, substituyendo cada vez que se presente el valor 10 por la letra A, con la finalidad de aprovechar el método gráfico utilizado en el Capítulo 2 para evaluar la eficiencia de este método.

La tabla ilustrada a continuación ha sido generada de igual manera a la enunciada en el capítulo 2, con la diferencia de que ésta tiene 11 columnas y 11 renglones puesto que ahora se está trabajando módulo 11.

SUMA POSICIONES NONES MOD 10

SUMA POSICIONES PARES MOD 10	1	2	3	4	5	6	7	8	9	10	0
1	A	2	5	8	0	3	6	9	1	4	7
2	6	9	1	4	7	A	2	5	8	0	3
3	2	5	8	0	3	6	9	1	4	7	A
4	9	1	4	7	A	2	5	8	0	3	6
5	5	8	0	3	6	9	1	4	7	A	2
6	1	4	7	A	2	5	8	0	3	6	9
7	8	0	3	6	9	1	4	7	A	2	5
8	4	7	A	2	5	8	0	3	6	9	1
9	0	3	6	9	1	4	7	A	2	5	8
10	7	A	2	5	8	0	3	6	9	1	4
0	3	6	9	1	4	7	A	2	5	8	0

TABLA 18

Si se observa esta tabla con detenimiento, podrá verse que no hay dígito que se repita en culquier columna, cualquier renglón y cualquier diagonal derecha.

Ello implica que si se usara el sistema propuesto módulo 11, serían detectados el 100% de los errores de transcripción, el 100% de los errores de transposición, y el 91% de los aleatorios. El porcentaje total de detección según los datos de Beckley[4] resultará ser del 99,4%.

Este sistema es bastante eficiente como puede verse, el problema que se ha encontrado en el, es cuando el digito de control generado es 10, pues éste necesita de dos lugares para su almacenamiento.

Una de las soluciones propuestas es, como ya se ha incluído en la Tabla 18, la de substituír el valor 10 por la letra A. Esto puede ocacionar serios problemas en ciertos sistemas en los que se tenga la premisa de que todos los digitos de la clave, incluyendo el de control, deben ser numéricos, pues éste será rechazado constantemente. De igual forma podrá darse el caso de que las personas que tuvieran que operar con esta clave, les resultara raro ver una letra incluída ahí, y entonces por voluntad propia, la substituyeran por un digito que juzgaran fuera el conveniente(por ejemplo: 4 en vez de A).

Una solución al problema de este digito es la de descartar todas aquellas claves cuyo digito resultará ser el 10. Esto generaría una gran cantidad de claves que tendrían que excluirse, y podría provocar el perder secuencialidad en ciertos sistemas que la necesitan. En caso de que sea posible omitir dichas claves, esto resultará ser una buena solución.

A continuación se presentará un sistema de digitos de control módulo 11, con un tratamiento distinto para cada una de las posiciones de los dígitos; se incluirá una propuesta distinta para solucionar el problema del dígito de control 10.

Dado que como se trata ahora módulo 11, se tienen como primos relativos de éste todos los valores del 1 al 10; de esta forma, entre más pesos se ocupen, más errores de transposición múltiple serán detectados.

Sea una clave cualesquiera formada así:

c: N1 N2 N3 ... Nk

A cada uno de los dígitos que componen esta clave le será asignado un cierto peso (factor). De esta forma, podrá encontrarse la siguiente suma:

SUMATORIA de i=1 hasta i=k de (Wi Ni)

ésta será dividida entre 11, y el residuo de esta división será usado como el o los dígitos de control.

Ahora bien, surge la pregunta siguiente: Qué valores deben tomar las Wi ?

La gran mayoría de los usuarios [4] ha optado por tomar los valores del 1 al 10 en forma decreciente como se ilustra a continuación:

N10	N9	N8	N7	N6	N5	N4	N3	N2	N1
10	9	8	7	6	5	4	3	2	1

Pero según unos estudios que se han realizado [4, 7, 9, 15, 18], se han propuesto ciertos órdenes para estos pesos con el fin de incrementar la detección de ciertos errores que en el presente estudio se han clasificado dentro de los aleatorios.

La serie propuesta es:

W10	W9	W8	W7	W6	W5	W4	W3	W2	W1
9	10	7	8	4	6	3	5	2	1

El sistema propuesto con estos pesos dará los siguientes porcentajes de detección de errores:

Errores de Transcripción	:	100%
Errores de Transposición Sencilla y Múltiple	:	100%
Errores Aleatorios	:	91%

Como se ve, este método alcanza muy altos porcentajes de detección de errores, lo único que puede llegar a ser un inconveniente es el caso del dígito de control 10, pero si puede tomarse alguna de las soluciones propuestas anteriormente para este caso, resultará beneficioso el uso de éste.

Continuando con el estudio de este dígito de control (10) para los sistemas módulo 11, se presentará a continuación otro método módulo 11, con una alternativa diferente que algunos autores [9, 15] han propuesto.

Este método consiste en lo siguiente, la primera parte resultará similar a la dada anteriormente:

1.Se toma un código c representado como c: N1 N2 N3 ... Nk.

2.Se selecciona la serie de pesos. (Esta puede ser la dada anteriormente por las ventajas que proporciona).

3.Se calcula S que será la sumatoria desde que i=1 hasta 9 de NiWi, donde Wi representa los pesos.

4.Se calcula R, donde R es el residuo obtenido de dividir S/11.

5.Se calcula el dígito de control mediante la operación d= (11 - R). *

* Este último paso es realizado por los autores antes señalados, aunque para efectos de detección de errores no proporcionará una mayor o menor detección, puesto que todo lo que hace es intercambiar los valores resultantes de la siguiente forma: 0 a 11, 1 a 10, 2 a 9, 3 a 8, 4 a 7, 5 a 6, 6 a 5, 7 a 4, 8 a 3, 9 a 2, 10 a 1, 11

a 0.

De esta forma será encontrado el dígito de control. Como se puede apreciar, estos estarán en el intervalo [1,11]. Como el propósito es que solamente sea generado un solo dígito, entonces a los dígitos 10 y 11 se les dará un tratamiento especial que se expone a continuación.

Cuando el dígito calculado sea el 11, se puede notar que éste y el dígito 0 resultarán ser el mismo valor si a ambos se les aplica la función módulo 11, por esto, cuando el dígito 11 sea el generado como resultado de la función, se reemplazará por el dígito 0.

Para el caso del dígito 10, se verá lo que algunos autores han propuesto como una alternativa más [9].

El método consiste en que cuando dicho dígito sea generado, no se termine ahí el proceso, sino que se le aplique una vez más la función a la clave en cuestión con una serie de pesos distinta, que garantice que en esta ocasión el dígito generado será otro distinto al 10.

Para esto, después de varias propuestas [9, 15, 7] con respecto a los distintos valores que deberían de tomar las dos series de pesos, se llegó a la siguiente, que cumple con las características dadas en el modelo anterior, y con la cualidad de que si el dígito

10 se genera con la primera serie, no resultará así con la segunda:

$$W: \quad 9 \quad 10 \quad 7 \quad 8 \quad 4 \quad 5 \quad 3 \quad 6 \quad 2$$

$$W': \quad 2 \quad 10 \quad 4 \quad 3 \quad 7 \quad 6 \quad 8 \quad 5 \quad 9$$

De esta forma se ha resuelto el inconveniente ya enunciado ampliamente del digito 10, pero los problemas que se han originado con este nuevo planteamiento son que ahora no se captarán el 100% de los errores de transcripción ni de los de transposición.

Esto podrá verse más claramente en el ejemplo que se expone a continuación en el cual no es captado un error de transposición con este nuevo método.

Clave correcta: $c = 38$

Por medio de este sistema y usando los pesos correspondientes de W, se tiene:

$$S = 6(3) + 2(8) = 18 + 16 = 34$$

por lo tanto $R = 1$

ello implica que $d = (11-1) = 10$

como el digito generado es 10, se repetirá el procedimiento con la siguiente serie, de donde:

$$S = 5(3) + 9(8) = 15 + 72 = 87$$

por lo tanto $R = 10$

ello implica que $d = (11-10) = 1$

$d = 1$

Ahora bien, después del error de transposición:

Clave errónea : $c' = 83$

por lo tanto:

$$S = 6(8) + 2(3) = 48 + 6 = 54$$

por lo tanto $R = 10$

ello implica que: $d = (11-10) = 1$

$d = 1$

como se ve, el dígito generado para ambas claves por este método resultó ser el mismo. De ahí que este error de transposición no sea detectado.

Hasta este momento pueden sacarse las siguientes conclusiones:

Se han evaluado varios sistemas de dígitos de control módulo 11, y se ha detectado la problemática que existe con éste, que básicamente radica en la generación del dígito de control 10 que tiene dos cifras; para esto se han analizado una serie de soluciones donde lo más conveniente resulta ser la de tomar una letra para substituir ese dígito o ignorar (es decir no ocupar) las claves que generen dicho dígito (en caso de que alguna de estas dos opciones puedan tomarse), pero si no pueden tomarse ninguna de las dos, lo más conveniente será dejar a un lado el sistema módulo 11 y tomar el sistema de dígitos de control módulo 10 explicado anteriormente.

CAPITULO 5. Sistemas Módulo N para N Mayor Que 11.

Los sistemas módulo N mayor a 11 han sido estudiados bajo el hecho de que no importará que como resultado de estos sea generado un dígito de control de más de una cifra.

En el caso estudiado en el capítulo anterior (donde N = 11), se trató de abolir toda opción en la que se generara un dígito de más de una cifra, puesto que en algunos sistemas de información esto no es posible, no es conveniente, o causará un mayor costo con una utilidad marginal.

Para aquellos sistemas de información en los cuales es posible tener más de un dígito de control, y se quiere un método con mejores porcentajes para obtenerlo, lo que se presentará aquí resultará de gran utilidad.

Como se verá, no es el sistema módulo 11 visto anteriormente el que mejores porcentajes de captación de errores tiene, sino los que se verán a continuación.(Sin olvidar que estos sistemas generan un dígito de control de más de una cifra pues la función módulo que se ocupa es para N mayor que 11.).

Nota. No hay que olvidar que el estudio que se está haciendo es para claves formadas por dígitos decimales.

La forma por medio de la cual estos métodos encuentran los dígitos de control es mediante el uso del residuo, o de un derivado de éste, de dividir la clave numérica, o de un número derivado de ésta según una serie de reglas predeterminadas, entre un entero N (donde N es mayor que 11).

Como base de análisis se tiene:

$$\text{Clave} : c = N1\ N2\ N3\ \ldots\ Nk$$

$$\text{Pesos} : w = W1\ W2\ W3\ \ldots\ Wk$$

Los dígitos se generarán del residuo de la operación: la sumatoria desde que i=1 hasta que i=k de (Wi Ni) mod N.

Lo que ahora ha de determinarse son los valores que deben tomar las Wi's y N.

Si se estudia esto en base a los errores que pueden ocurrir con mayor frecuencia, se ha obtenido que[18]:

1.-Para que un error de transcripción sencillo sea detectado, debe suceder que (Ni - x), Wi o (Ni - x)Wi no sean divisibles por N.

Como (Ni - x) está en el intervalo [1,9], entonces N debe ser mayor a 9. Ninguno de los pesos debe ser igual a N. Si todos los pesos son asignados menores a N esto se cumplirá. Para que el producto (Ni - x)Wi no sea divisible por N debe suceder que todas las Wi sean primos relativos de N, y esto se cumplirá, y se tendrán más valores con que contar si N es un número primo.

2.-Para que un error de transcripción múltiple sea detectado, será necesario que la suma de dos o más pesos consecutivos no sea igual a N o a un múltiplo de éste.

Para que un error de transposición sea detectado, la diferencia entre cualesquiera dos pesos no debe ser igual a N. Si todos los pesos son menores a N, y ninguna pareja de ellos son iguales, esto se conseguirá. Por ello N debe ser un número primo.

En resumen si: - N es mayor que 9 - todos los pesos Wi son menores a N - la suma de dos o más pesos consecutivos no es igual a N o a un múltiplo de N - Wi es distinto de Wj para toda i,j en el intervalo [1,k] e i es distinta de j - N es un número primo

Entonces los errores de transcripción sencilla, transcripción múltiple de dos o más dígitos iguales, y los errores de transposición podrán ser cubiertos en un 100% por este sistema.

El porcentaje de errores aleatorios será de aproximadamente 100(N-1)/N %.

Como ejemplos de algunos sistemas que cumplen con estas propiedades se tienen los siguientes:

1) N = 11 (usando los pesos recomendados por Beckley)

W1	W2	W3	W4	W5	W6	W7	W8	W9	W10
1	2	5	3	6	4	8	7	10	9

N es mayor que 9

Wi es menor a N para toda i del intervalo [1,10]

Wi es distinto de Wj para toda i y j del mismo intervalo

N es un número primo

En este sistema son captados el 100% de los errores de transcripción sencilla y de transposición sencilla. Los errores de múltiple transcripción de dos o más digitos iguales consecutivos serán detectados en un poco menos del 100%, pues algunos pesos consecutivos sumas N o un múltiplo de éste.

2) N = 97

Pesos:	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10
	34	81	76	27	90	9	30	3	10	1

en el cual:

N es mayor que 9

Wi es menor que N para toda i del intervalo [1,10]

No existen dos o mas pesos consecutivos que sumen N o un múltiplo de éste.

Wi es distinto de Wj para toda i y j del intervalo [1,10]

En este sistema se detectarán el 100% de los siguientes tipos de error: transcripción sencila, múltiple, transcripción de dos o más dígitos consecutivos iguales, transposición. Los errores aleatorios serán detectados en aproximadamente un 99%.

Para los datos de Beckley[4], este sistema dará un porcentaje de detección de 99.94%.

CAPITULO 6. Sistema de Control Para Claves Alfabéticas.

Hasta este momento se han estudiado sistemas de dígitos de control para claves numéricas en sistema decimal. Ahora se presentará un breve estudio de dígitos de control para claves formadas por letras en vez de números.

Como se verá a continuación, el caso que se expone en este capítulo ha sido desarrollado tomando como base la misma metodología usada en capítulos anteriores. Lo que muestra una forma en que el estudio de esos capítulos puede extenderse a otros tipos de claves formadas no solo por dígitos en sistema decimal.

Para estudiar las claves alfabéticas se hará uso de cierta nomenclatura que se da a continuación.

Supóngase que se tiene una clave alfabética que está representada de la forma:

$$\text{Clave} : C : L1\ L2\ L3\ \ldots\ Lk$$

donde el conjunto de valores que puede tomar Li son del conjunto:

M = { A, B, C, D, E, F, G, H, I,

J, K, L, M, N, O, P, K, R,

S, T, U, V, W, X, Y, Z }

Como se ve, Li puede tomar 26 valores distintos. Si se descartan las letras : I, O y S ya que pueden ser confundidas con los dígitos : 1, 0 y 5 respectivamente[17], y con el fin de que el total de letras que pueda tomar Li sea un número primo, se tiene el siguiente conjunto:

L = { A, B, C, D, E, F, G, H,

J, K, L, M, N, P, Q, R,

T, U, V, W, X, Y, Z }

por lo tanto Li puede tomar 23 valores(letras) distintos.

Entonces, si se siguen las reglas dadas en el capítulo anterior para sistemas módulo N donde N era mayor que 11, se dará un sistema tan eficiente como los presentados ahí.

Para esto, a cada letra del conjunto L le será asignado un valor entero del intervalo [0,22]. Para que sea fácil de recordar, estos valores serán asignados de acuerdo a la siguiente tabla:

A = 0	N = 12
B = 1	P = 13

C = 2	Q = 14
D = 3	R = 15
E = 4	T = 16
F = 5	U = 17
G = 6	V = 18
H = 7	W = 19
J = 8	X = 20
K = 9	Y = 21
L = 10	Z = 22
M = 11	

TABLA 19

Se analizarán ahora los errores.

Si un error de transcripción sucede, para que este sea detectado debe suceder que :

$(L_i - x)\,W_i$ no sea múltiplo o igual a 23

$(L_i - x)$ está en el intervalo $[1,22]$

Si se toman pesos entre el 1 y el 22 inclusive, no resultarán valores múltiplos a 23, y por lo tanto se captarán todos los errores de transcripción.

En el caso que se está tratando, N debe ser mayor no unicamente que 9, sino también que 22, pues hasta este valor puede tomar la diferencia (Li - x).

Si se toman como "pesos" la siguiente serie con valores comprendidos entre 1 y 14:

	W12	W11	W10	W9	W8	W7	W6	W5	W4	W3	W2	W1
W:	13	3	4	9	11	5	6	2	12	7	8	1

se puede observar que:

1) Todos los pesos son distintos.
2) Todos son menores a N = 23.
3) Si se suman dos o más pesos consecutivos, no dará como resultado N, ni un múltiplo de éste.

Este último punto puede verificarse:

N = 23, 46, 69, 92, 115, ...

Cualquier combinación que se tome de W (de pesos consecutivos) nunca dará alguno de estos valores. A continuación se ven algunos casos.

13	03	04	09	11	05	06	02	12	08	07	01

81	68	65	61	52	41	36	30	28	16	08	01
80	67	64	60	51	40	35	29	27	15	07	
72	59	56	52	43	32	27	21	19	07		
65	52	49	45	36	25	20	14	12			
53	40	37	33	24	13	08	02				
51	38	35	31	22	11	06					
45	32	29	25	16	05						
40	27	24	20	11							
29	16	13	09								
20	07	04									

13	16	20	29	40	45	51	53	65	72	80	81
	03	07	16	27	32	38	40	52	59	67	68
		04	13	24	29	35	37	49	56	64	65
			09	20	25	31	33	45	52	60	61
				11	16	22	24	36	43	51	52
					05	11	13	25	32	40	41
						06	08	20	27	35	36
							02	14	21	29	30

12	19	27	28
	07	15	16

TABLA 20

El primer renglón de esta tabla representa la serie de pesos. Para poder entender dicha tabla, basta con empezar en el renglón 2, de derecha a izquierda, y se verá que cada valor de este renglón es el resultado de sumar el valor que se encuentra a la derecha de éste, más el que se encuentra arriba de este mismo.

En el tercer renglón es el mismo mecanismo, pero se empieza con el segundo dígito, para así ir analizando todas las combinaciones.

Como se ve, en las combinaciones antes presentadas, en ningun caso se presenta algún valor igual a N = 23, o a algún múltiplo de éste.

De esta misma forma se sacaron los demás casos en que podría ser posible esto, llegándose al mismo resultado.

Así se puede decir que con esta serie de pesos los errores de transcripción múltiple podrán ser detectados ya que cumple las condiciones generales dadas en el capítulo anterior.

Según se encontró en los casos módulo N, para que un error de transposición fuera detectado, era necesario que la diferencia entre cualesquiera dos pesos no fuera igual a N. Esto se consigue con tal que: Wi sea menor que N para toda i = 1,2,...,k y que Wi sea distinto de Wj para toda i y j del conjunto {1,2,3,...,k}; lo cual sucede en la serie propuesta.

En resumen, con el sistema propuesto los errores de transcripción sencilla, transcripción múltiple de dos o más dígitos iguales, y los errores de transposición serán cubiertos en un 100% por este sistema.

El porcentaje de errores aleatorios será aproximadamente de: 100(22)/23 = 95.65 % (según fórmula dada anteriormente).

Según los datos de Beckley[4], este sistema detectará un porcentaje total de 99.739 %.

El valor que resultará de calcular el dígito de control, siempre convirtiendo las letras en su valor numérico correspondiente dado en la tabla 19, será convertido en la letra que corresponda a ese valor.

Lo que se ha hecho en este capítulo ha sido cambiar la base del álgebra con que se estaba trabajando en este estudio, y se ha observado que la misma metodología es aplicable. Se partió del alfabeto, y se convirtió en un problema módulo 23, por el contrario, se hubiese optado por el módulo 26, como este valor es par al igual que en el caso módulo 10, no sería posible alcanzar el 100% de la detección en los errores de transposición (Ver demostración en el Apéndice 2), obteniéndose pues resultados similares a los obtenidos anteriormente.

CAPITULO 7. Conclusiones.

Se ha expuesto en el presente trabajo la necesidad de contar con un buen sistema de dígitos de control, que estará determinado, en cierta forma, por su mayor capacidad para detectar los errores que con mayor frecuencia se cometen, y también, por la cantidad de dígitos de control que puedan ser añadidos a la clave que se trate.

Con estos fines, se dió una clasificación de los errores con sus correspondientes porcentajes de ocurrencia según pruebas que se han hecho[4]; en base a esto, se expusieron varios sistemas en bases distintas con un análisis de los porcentajes de error que se detectaban, y una explicación clara de cómo encontrar el o los dígitos de control en cada caso.

La clasificación dada, con los porcentajes de ocurrencia en cada caso[4], es la siguiente:

Errores del tipo I

Errores de Transcripción Sencilla 86 %

Errores del tipo II

Errores de Transposición Sencilla 8 %

Errores del tipo III

Errores Aleatorios y Otros 6 %

Para la mejor detección de estos errores se estudiaron varios métodos, los cuales manejan la función módulo N con distintos valores para ésta, dando como resultado distintos posibles dígitos, con las conclusiones que se dan a continuación.

Dependiendo del número de dígitos que el sistema de dígitos de control genere, se tendrán alguno de los siguientes casos: 1.Un solo caracter numérico siempre. -El sistema con mejores porcentajes que cumpla con estas características será: Sistema con N = 11 con los pesos siguientes:

W : 9 10 7 8 4 5 3 6 2

y, en cada caso que resulte el dígito 10, descartar dichas claves.

Este sistema es el aconsejable cuando sea posible y se quiera descartar todas las claves que generen el dígito 10.

En caso contrario:

-Sistema con N = 10, con la siguiente función para encontrar el digito de control:

$$f(c) = [\, P(N1 + N5 + \ldots) + K(N2 + N6 + \ldots) + R(N3 + N7 + \ldots) + S(N4 + N8 + \ldots) \,] \bmod 10$$

Con los valores: $P = 3$

$K = 9$

$R = 7$

$S = 1$

2.Dos caracteres numéricos.

-Sistema en base 97 con los siguientes pesos (por ejemplo):

W : 34 81 76 27 90 9 30 3 10 1

Los porcentajes de captación de errores de cada método pueden verse en la siguiente tabla, así como el porcentaje total tomando en cuenta los datos de Beckley[4].

	ERROR TRANSC.S	ERROR TRANSP.S	ERROR TRANSP.M	ERRORES ALEATORIOS	TOTAL
SIST. MODULO 10	100%	88.8%	*	90%	98.51
SIST. MODULO 11	100%	100%	100%	90.90%	99.4
SIST. MODULO 97	100%	100%	100%	99%	99.94
SIST. PARA CLAVES ALFABETICAS MOD 23	100%	100%	100%	95.65%	99.74

TABLA 21

Como puede verse en esta tabla, el sistema de digitos de control módulo 97 alcanza un porcentaje total de detección(según los datos de Beckley[4]) del 99.94 % que es el más alto de los tres primeros, que son los que se refieren a claves numéricas en sistema decimal.

De ello puede concluirse que si se necesita un sistema que detecte el mayor número posible de errores, y que pueden tenerse dígitos de control de dos cifras, este es el adecuado.

En el caso de que solo se quiera una cifra como digito de control, los sistemas módulo 10 y 11 tienen muy buen porcentaje total de detección (según datos de Beckley[4]).

Este mismo estudio que se ha realizado, podría ser aplicado en forma similar para claves cuyos digitos no sean decimales forsosamente, sino en base binaria, octal, y así sucesivamente. Con ciertos cambios en los valores de los pesos, dependiendo de los valores con que se cuente por la base con que se esté trabajando, y ocupando de estos, los que sean primos relativos de ésta, realizándose todo esto en forma paralela y muy similar a lo aquí expuesto.

✱ Nota: Para el sistema módulo 10, estos errores se englobaron dentro se la categoría de los aleatorios; es por eso que el porcentaje de captación no aparece aquí.

APENDICE 1.

DEMOSTRACION DE POR QUE EN UNA TABLA DE NXN PARA N = 10, NO ES POSIBLE PONER UN DIGITO DISTINTO EN CADA COLUMNA, CADA RENGLON, Y CADA DIAGONAL DERECHA.

Sea un digito representado como:

$$N1\ N2\ N3\ \ldots\ Nk$$

con sus respectivos pesos a cada posición:

$$W1\ W2\ W3\ \ldots\ Wk$$

El digito verificador resultará entonces de la fórmula: la sumatoria desde que i = 1 hasta que i = k del producto WiNi módulo M.

En el método propuesto se tenian unicamente dos pesos distintos, uno para las posiciones nones y otro para las pares, por lo que la expresión anterior se convertirá en:

$$Eij = [\ (N1 + N3 + \ldots)\ P1 + (N2 + N4 + \ldots)\ P2\ ]\ \mathrm{mod}\ M$$

Errores de Transcripción.

Qué pasa si un error de transcripción sucede ?

Supóngase que fue el término Ni el que fue modificado, entonces si el nuevo término es x, se tendrá que la diferencia entre la cantidad real y la errónea será de:

$$P\ (\ Ni - x\ )\ \mathrm{mod}\ M$$

Se quiere que esta diferencia sea detectada, y para que eso suceda esta cantidad no debe ser igual a 0 (cero) módulo M.

Para que esto no suceda, tiene que pasar que:

1.- (Ni-x) no sea divisible por M,

2.- P no sea divisible por M,

o 3.- P(Ni-x) no sea divisible por M.

Se analizará ahora que pasa para M = 10.

1.Como Ni, x están en el conjunto { 0, 1, 2, ..., 9 } ello implica que (Ni-x) está en el intervalo [1,9]. El valor 0 (cero) se excluye del intervalo pues Ni no puede ser igual a x ya que en ese caso no habría error. Como M es mayor a (Ni-x), entonces esto no puede ser divisible por M.

2.Si P es menor a M, entonces P(Ni-x) no es divisible.

3.Ya se vió que (Ni-x) puede tomar valores del intervalo [1,9]. Los valores de P que pueden ocasionar que la diferencia no sea detectada, son todos los pares menores a 10 y el valor 5 [14], es decir, serían los valores: 0, 2, 4, 5, 6, 8.

$$\text{Como} \quad (2X1) \bmod 10 = 2 \bmod 10 = 2$$

$$\text{y} \quad (2X6) \bmod 10 = 12 \bmod 10 = 2$$

Entonces, para poder detectar P(Ni-x) P debe pertenecer al conjunto {1, 3, 7, 9}.

En conclusión, si P está en el conjunto {1, 3, 7, 9}, entonces los errores de transcripción serán detectados en su totalidad.

Errores de Transposición de dos dígitos consecutivos.

Supóngase que son los términos Ni y Ni+1 los que son intercambiados, entonces, si se tenía:

$$Ni\ P1 + Ni{+}1\ P2 + k$$

el número erróneo será:

$$Ni{+}1\ P2 + Ni\ P1 + k$$

La diferencia entre el valor real y el erroneo será:

$$(\ Ni - Ni{+}1\ )\ P2 + (\ Ni{+}1 - Ni\ )\ P1$$

$$= (\ Ni - Ni{+}1\ )\ (\ P2 - P1\ )$$

Se quiere que este valor sea detectado, o sea, que esta cantidad no sea igual a cero módulo 10. Para que sea detectado, debe pasar que:

1.- (Ni - Ni+1) no sea divisible por 10,

2.- (P2 - P1) no sea divisible por 10,

o 3.- (Ni - Ni+1)(P2 - P1) no sea divisible por 10.

Como se vió en el caso de los errores de transcripción, Ni y Ni+1 pertenecen al conjunto { 0, 1, 2, ..., 9}, ello implica que (Ni - Ni+1) están el el intervalo [1,9] mod 10, ello implica que (Ni - Ni+1) no es divisible por 10.

2.Como fue concluido en el caso de los errores de transcripción, Pi está en el conjunto {1, 3, 7, 9}, ello implica que Pi sea menor que 10, por lo tanto no es divisible por 10.

3.Se ha visto que (Ni - Ni+1) está en el intervalo [1,9]. Pi puede tomar cualquier valor del conjunto { 1, 3, 7, 9}. Si P1 y P2 fueran iguales, la diferencia (P2 - P1) sería cero, por lo que no se detectaría ningún error de transposición; entonces supóngase que son distintas. Si se observa esta diferencia, podrá verse que siempre dará como resultado un número par cualesquiera (dependiendo de la elección de P2 y P1) del conjunto (2, 4, 6, 8}. Entonces, al hacer el producto de (Ni - Ni+1)(P2 - P1) habrá dos valores para los que

(P2 - P1) dará dos dígitos de control iguales usando módulo 10.

Esto es, pues por ejemplo:

$$(4 \times 1) \bmod 10 = 4 \bmod 10 = 4 \text{ módulo } 10$$

$$(4 \times 6) \bmod 10 = 24 \bmod 10 = 4 \text{ módulo } 10$$

y eso sucede para cualquier valor del conjunto {2, 4, 6, 8}, por lo que no pueden ser detectados todos los errores de este tipo.

APENDICE 2.

DEMOSTRACION DE POR QUE EN UNA TABLA DE NXN PARA N PAR, NO ES POSIBLE PONER UN DIGITO DISTINTO EN CADA COLUMNA, CADA RENGLON Y CADA DIAGONAL DERECHA.

Sea un códiso representado como:

$$N1 \quad N2 \quad N3 \ \ldots \ Nk$$

con sus respectivos pesos a cada posición:

$$W1 \quad W2 \quad W3 \ \ldots \ Wk$$

El dígito verificador resultará de calcular la sumatoria desde que i = 1 hasta que i = k a (Wi Ni) mod M'.

Como M'es par, en adelante se denotará como M'= 2M donde M es cualquier valor de los enteros positivos. Entonces ahora la fórmula para calcular el dígito verificador será el resultado de la sumatoria desde que i = 1 hasta que i = k de (Wi Ni) mod 2M.

En el método propuesto en este estudio se tienen solo dos pesos distintos por lo que la fórmula se convertirá en:

$$d = [\ (N1 + N3 + \ldots)P1 + (N2 + N4 + \ldots)P2\ ] \bmod 2M$$

Errores de Transcripción.

Si un error de transcripción sucede en el término N_i, al modificarse por el valor x, se tendrá que la diferencia entre la cantidad real y la erronea será de:

$$P \, (N_i - x) \bmod 2M$$

donde P representa ya sea a P_1 o a P_2 dependiendo del lugar donde ocurra el error.

Se quiere que esta diferencia sea detectada, para que esto suceda, esta cantidad no deberá ser igual a 0 (cero) módulo 2M.

Para que esto suceda tiene que pasar que:

1.- $(N_i - x)$ no sea divisible por M,

2.- P no sea divisible por M,

o 3.- $P(N_i - x)$ no sea divisible por M.

1.Como N_i y x pertenecen al conjunto $\{ 0, 1, \ldots, 2M-1\}$ ello implica que $(N_i - x)$ esté en el intervalo $[1,2M-1]$. Basta con que 2M sea mayor que $(N_i - x)$ para que este no sea divisible por 2M, y esto sucede ya que 2M es mayor que 2M-1.

2.Basta con que P sea menor a 2M para que P no sea divisible por 2M. Ello implica que 2M es mayor a P, ello implica que M es mayor que P/2.

3.Ya se vio que $(N_i - x)$ puede tomar valores entre $[1,2M-1]$; los valores de P que pueden ocacionar que el producto P(Ni - x) sea divisible por 2M son, según residuos completos[14]: 0, 2, 4, 6, 8, 10, ..., 2M-2 y M también se descartaría, quedándose con el conjunto para escojer P y K de todos los valores del intervalo [1,2M-1] menos los siguientes: { 2, 4, 6, .., 2M-2, M }; de esta forma, si 0 es menor o igual a P que es menor o igual a 2M-1 y P es distinta de 2, 4, 6, ..., 2M-2,M ; entonces todos los errores de transcripción podrán ser detectados.

Errores de Transposición de dos digitos consecutivos.

Supóngase que son los términos Ni y Ni+1 los que serán intercambiados, entonces, si se tenía:

$$(Ni\ P2 + Ni{+}1\ P1 + k) \bmod 2M$$

con el intercambio se tendrá:

$$(Ni{+}1\ P2 + Ni\ P1 + k) \bmod 2M$$

la diferencia entre el valor real y el erroneo será de:

$$(Ni - Ni{+}1)(P2 - P1) \bmod 2M$$

Se quiere que esta diferencia sea detectada, es decir, que esta cantidad no sea igual a 0 (cero) módulo 2M.

Para que sea detectada, debe pasar que:

1.- $(N_i - N_{i+1})$ no sea divisible por 2M,

2.- $(P2 - P1)$ no sea divisible por 2M,

o 3.- $(N_i - N_{i+1})(P2 - P1)$ no sea divisible por 2M.

1.Como se vió en el caso de los errores de transcripción, N_i y N_{i+1} están en el intervalo $[0,2M-1]$, ello implica que la diferencia $(N_i - N_{i+1})$ está en el intervalo $[1,2M-1]$. Basta con que 2M sea mayor que $(N_i - N_{i+1})$ para que éste no sea divisible por 2M. Ello implica que 2M es mayor que 2M-1.

2.Como fue concluido en el caso de los errores de transcripción, P_i pertenece al conjunto $\{1, 3, 5, 7, 9, \ldots, 2M - 1\} - \{M\}$, por lo tanto $(P2 - P1)$ es menor que $2M - 1$, ello implica que como $P2 - P1$ es menor a 2M, $(P2 - P1)$ no es divisible por 2M.

3.Si P1 y P2 fueran iguales, la diferencia $(P2 - P1)$ sería 0(cero), por lo que no se detectaría ningún error de transposición; por lo tanto P1 y P2 deben ser distintas.

Si se observan los valores que puede tomar P_i, se ve que la diferencia $(P2 - P1)$ dará como resultado siempre un número par del conjunto $\{2, 4, 6, 8, \ldots, 2M - 2\}$ dependiendo de la elección de P1 y P2.

Por lo que el producto:

$$(Ni - Ni+1)(P2 - P1)$$

puede escribirse como:

$$(Ni - Ni+1)\ 2P'$$

$$\text{donde } 2P' = P2 - P1$$

No se quiere que sea divisible por 2M, se puede decir como:

$$(Ni - Ni+1)\ 2P' \bmod 2M$$

no se quiere sea cero.

Si por ejemplo se tuviera el caso en que Ni - Ni+1 = M, que es completamente posible pues Ni y Ni+1 pueden tomar cualquier valor del intervalo [0,2M-1], se verá lo siguiente:

$$(Ni - Ni+1)\ 2P' \bmod 2M$$

se transformará en:

$$M\ 2\ P' \bmod 2M$$

$$2M\ P' \bmod 2M$$

que como se ve dará como resultado 0(cero), pues es un valor P'multiplicado por 2M.

De aquí se concluye que no es posible encontrar una tabla de MXM con M par tal que no se repita ningún número ni en cada renglón, ni en cada columna, ni en cada diagonal derecha.

BIBLIOGRAFIA.

1. ANDREW A. M. A variant of modulus-11 checking. The Computer Bulletin. Vol. 14, 1970.

2. ANDREW A. M. Checking. The Computer Bulletin. Vol 14, Agosto 1970.

3. ANDREW A.M. Decimal numbers with two check digits. The Computer Bulletin. Vol 16, 1972.

4. BECKLEY D.F. An optimum system with modulus 11. The Computer Bulletin. Diciembre, 1967.

5. BECKLEY D.F. Check digit verification. Data Processing. Julio-Agosto, 1966.

6. BELL. Decimal numbers. The Computer Bulletin. 1972, Vol 16.

7. BRIGGS. Modulus 11 check digit systems. The Computer Bulletin. Vol 14, No. 8, 1970.

8. BRIGGS. Weights for modulus 97 systems. The Computer Bulletin. Vol 15, 1971.

9. CAMPBELL D. V. A. A modulus 11 check digit system for a given system of codes. The Computer Bulletin. Vol 14, 1970.

10. CAMPBELL D. V. A. Check digits. The Computer Bulletin. Vol 14, 1970.

11. CARBAJAL Raúl, GRAPA Enrique. Dígitos de Control. Comunicaciones Técnicas. UNAM 1973.

12. CARVAJAL Raúl. Modulus N check digits and the chromatic number problem. Comunicaciones Técnicas. UNAM 1974.

13. GRAPA Enrique. Tesis de Licenciatura.

14. NIVEN. An introduction to the teory of numbers. 1960.

15. REID C. J. Modulus 11 check digits. The Computer Bulletin. Vol 14, 1970.

16. RICHARDSON M. Check digits. The Computer Bulletin. Vol 14, 1970.

17. ROWLANDSON. Check digits. The Computer Bulletin. Vol 15, 1971.

18. WILD. The teory of modulus N check digit systems. The Computer Bulletin. Diciembre 1968.

19. RICHARD W. Hamming. Coding and Information Theory. 1980.

20. BERLECKAMP, E. R. Algebraic Coding Theory. New York. 1968.

21. GALLAGER, Robert G. Low-Density Parity-Check codes. 1963.

22. PETERSON and WELDON. Error-Correcting Codes. 1961.