



UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO

FACULTAD DE ESTUDIOS SUPERIORES
CUAUTITLAN

DESARROLLO Y APLICACION DE UN
SISTEMA INFORMATICO DE ESTADISTICA.

T E S I S

PARA OBTENER EL TITULO DE:
LICENCIADA EN INFORMATICA
P R E S E N T A N:
VERONICA / FLORES CASTILLO
MARIA GUADALUPE YAÑEZ MEDINA

ASESOR Y COASESOR: ING. JORGE ALTAMIRA IBARRA.
ING. ARMANDO AGUILAR MARQUEZ.

TESIS CON
FALLA DE ORIGEN

CUAUTITLAN, EDO. DE MEXICO.

1997



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

PAGINACION VARIA

COMPLETA LA INFORMACION



UNIVERSIDAD NACIONAL
AVENIDA DE
MEXICO

FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLAN
UNIDAD DE LA ADMINISTRACION ESCOLAR
DEPARTAMENTO DE EXAMENES PROFESIONALES

ASUNTO: VOTOS APROBATORIOS

DR. JAINE KELLER TORRES
DIRECTOR DE LA FES-CUAUTITLAN
P R E S E N T E .

AT'N: Ing. Rafael Rodríguez Cobalico
Jefe del Departamento de Exámenes
Profesionales de la F.E.S. - C.

Con base en el art. 28 del Reglamento General de Exámenes, nos permitimos comunicar a usted que revisamos la TESIS TITULADA:
Desarrollo y aplicación de un sistema informático de estadística

que presenta la pasante: Verónica Flores Castillo
con número de cuentas: 8814336-1 para obtener el TITULO de:
Licenciada en Informática

Considerando que dicha tesis reúne los requisitos necesarios para ser discutida en el EXAMEN PROFESIONAL correspondiente, otorgamos nuestro VOTO APROBATORIO.

A T E N T A M E N T E .
"POR MI RAZA HABLARA EL ESPIRITU"
Cuautilán Izcalli, Edo. de Mex., a 27 de Octubre de 1996

PRESIDENTE	Ing. Juan R. Garibay Bernádez	
VOCAL	Lic. Artemio Hernández Rodríguez	
SECRETARIO	Ing. Jorge Altamira Ibarra	
PRIMER SUPLENTE	Ing. Rogelio Ramos Carranza	
SEGUNDO SUPLENTE	Lic. Héctor Hernández Guzmán	



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLÁN
UNIDAD DE LA ADMINISTRACIÓN ESCOLAR
DEPARTAMENTO DE EXÁMENES PROFESIONALES

ASUNTO: VOTOS APROBATORIOS

DR. JAIME KELLER TORRES
DIRECTOR DE LA FES-CUAUTITLÁN
P R E S E N T E .

AT'NI: Ing. Rafael Rodríguez Ceballos
Jefe del Departamento de Exámenes
Profesionales de la F.E.S. - C.

Con base en el art. 28 del Reglamento General de Exámenes, nos permitimos comunicar a usted que revisamos la TESIS TITULADA:
Desarrollo y aplicación de un sistema informático de estadística

que presenta la pasante: María Guadalupe Yáñez Medina
con número de cuentas: 8829485-B para obtener el TÍTULO de:
Licenciada en Informática

Considerando que dicha tesis reúne los requisitos necesarios para ser discutida en el EXÁMEN PROFESIONAL correspondiente, otorgamos nuestro VOTO APROBATORIO.

A T E N T A M E N T E .
"POR MI RAZA HABLARA EL ESPIRITU"
Cuautitlán Izcalli, Edo. de Mex., a 27 de Octubre de 1996

PRESIDENTE Ing. Juan R. Garibay Bernádez

VOCAL Lic. Artemio Hernández Rodríguez

SECRETARIO Ing. Jorge Altamira Ibarra

PRIMER SUPLENTE Ing. Rogelio Ramos Carranza

SEGUNDO SUPLENTE Lic. Héctor Hernández Guzmán

AGRADECIMIENTOS DE VERONICA

*Primeramente doy Gracias a **DIOS**,
por ser mi guía y compañía en este y en todos los caminos de mi existencia.
Mi eterno amor y agradecimiento por no haberme abandonado y darme
esta oportunidad.*

*Con todo mi cariño y respeto a mi **MAMA**,
ya que sin tu paciencia y tus esfuerzos esta meta no hubiera sido posible, tu
enorme deseo de buscar lo mejor para mí fue el principal motivo para
comenzar, seguir y terminar una carrera profesional, así como la presente
tesis. Te estoy infinitamente agradecida por todo el apoyo brindado, por los
sacrificios que esto representó, por el tiempo dedicado y sobre todo por la
confianza depositada en mí.
Por todo esto y mucho más eres merecedora no sólo de este testimonio,
sino de mi devoción para toda la vida.*

*A mis **HERMANOS**,
por estar a mi lado y compartir este logro, por todo su tiempo,
comprensión y apoyo que me brindaron, todo ello necesario para seguir
adelante.*

A mis **SOBRINOS,**

por el cariño y alegría que me brindan.

A mis **CUÑADOS,**

quienes forman parte de mi familia.

A **JAVIER,**

*por toda tu comprensión y apoyo en los momentos difíciles y porque juntos
lograremos todos nuestros sueños.*

A **LUPITA,**

*por todo este tiempo que hemos dedicado en la realización de este trabajo,
por todos tus conocimientos compartidos, pero sobre todo por tu amistad
incondicional.*

A **LA FAMILIA YAÑEZ MEDINA,**

*gracias por todo el apoyo y facilidades proporcionadas para la realización
de este proyecto.*

*A mi asesor de tesis **ING. JORGE ALTAMIRA IBARRA,**
con admiración y respeto, una inmensa gratitud por todo el tiempo, apoyo,
dedicación y paciencia en la realización de esta tesis. Un enorme
agradecimiento por confiar y creer en este trabajo y en nosotras.*

*Al coasesor de tesis **ING. ARMANDO MARQUEZ,**
agradeciendo inmensamente su tiempo, apoyo y paciencia que dedicó en la
realización y revisión de esta tesis.*

A los profesores:

ING. JUAN R. GARIBAY BERMUDEZ
LIC. ARTEMIO HERNANDEZ RODRIGUEZ
ING. ROGELIO RAMOS CARRANZA
LIC. HECTOR HERNANDEZ GUZMAN

gracias por el tiempo dedicado en la revisión de esta tesis.

AGRADECIMIENTOS DE GUADALUPE

Hoy, quiero compartir este logro con todas las personas que me han brindado su apoyo y confianza de manera incondicional.

DIOS:

Gracias por la oportunidad que me das día con día de seguir adelante.

MAMA:

Siempre estaré agradecida por el amor que nos brindas y por el sacrificio, lucha y entrega que has mostrado para sacarnos adelante.

PAPA:

Gracias por el cariño que me has dado. Hoy te dedico este trabajo esperando que sea un aliciente para que sigas luchando como hasta ahora.

HERMANOS:

Por su cariño, alegría, comprensión, enseñanzas y apoyo que en todo momento me han dado, gracias.

AMIGOS:

Gracias por su apoyo, su alegría y por los momentos que juntos compartimos.

VICENTE Y FERNANDO:

Les agradezco y valoro su amistad, sus críticas, su interés en este proyecto, su apoyo incondicional y sobre todo la confianza que me tienen.

VERONICA:

Agradezco y correspondo la amistad que me brindas, gracias por compartir este proyecto conmigo.

NUESTRO ASESOR: ING. JORGE ALTAMIRA IBARRA:

Mi gratitud por el tiempo y entusiasmo mostrado en la revisión de cada una de las pruebas del programa y borradores de la tesis, así como por sus enseñanzas, consejos y motivación que nos brinda.

NUESTRO COASESOR: ING. ARMANDO AGUILAR MARQUEZ:

- Gracias por las recomendaciones señaladas, el interés y estímulo para llevar adelante este proyecto*

PROFESORES:

Mi cariño y gratitud por el invaluable tesoro que entusiastamente nos ofrecen, por el interés mostrado por cada uno de sus alumnos, y por su dedicación y estímulos brindados.

Un agradecimiento especial a profesores, amigos y compañeros que nos apoyaron durante la carrera y en la realización de esta tesis.

INDICE

INTRODUCCION 1

CAPITULO I.- ESTADISTICA: ASPECTOS GENERALES

1.1 Antecedentes (reseña histórica)..... 4
 1.2 Objeto de la estadística..... 8
 1.2.1 Definición..... 8
 1.2.2 Objetivos 11
 1.2.3 Limitaciones 11
 1.2.4 Relación con el método científico 12
 1.2.5 Método estadístico..... 13
 1.2.5.1 Estadística descriptiva 13
 1.2.5.2 Inferencia estadística 14
 1.3 Importancia de la estadística 14
 1.3.1 Campo de aplicación 15
 1.3.2 Fenómenos de estudio..... 17

CAPITULO II.- ANALISIS Y DISEÑO DE SISTEMAS DE INFORMACIÓN

2.1 Análisis y diseño de sistemas de información 20
 2.2 Categorías de los sistemas de información 21
 2.2.1 Sistemas para el procesamiento de transacciones 21
 2.2.2 Sistemas informáticos para la administración..... 22
 2.2.3 Sistemas de apoyo para la toma de decisiones 22
 2.2.4 Sistemas expertos..... 22
 2.3 Ciclo de vida de los sistemas..... 23
 2.3.1 Enfoques del ciclo de vida para el desarrollo de sistemas 23
 2.3.1.1 Ciclo de vida lineal o tradicional 24
 2.3.1.2 Ciclo de vida estructurado..... 31
 2.3.1.3 Ciclo de vida por prototipos 35

CAPITULO III.- ANALISIS DEL SISTEMA DE ESTADISTICA PROPUESTO

3.1 Análisis del sistema de Estadística propuesto.....	41
3.2 Estudio de factibilidad.....	41
3.2.1 Determinación de la oportunidad.....	41
3.2.2 Identificación de las deficiencias actuales.....	42
3.2.2.1 Índices de reprobación.....	42
3.2.2.2 Características de los sistemas estadísticos comerciales.....	46
3.2.2.2.1 SAS.....	46
3.2.2.2.2 SPSS.....	48
3.2.2.2.3 Statgraphics.....	50
3.2.2.2.4 Stata.....	51
3.2.3 Determinación de requerimientos.....	53
3.2.4 Establecimiento de los objetivos del nuevo sistema.....	54
3.3 Análisis del nuevo sistema.....	55
3.3.1 Diagramas de flujo de datos.....	56
3.3.2 Tablas de decisión.....	60

CAPITULO IV.- DISEÑO DEL SISTEMA DE ESTADISTICA PROPUESTO

4.1 Diseño del sistema de Estadística propuesto.....	64
4.2 Diseño lógico.....	64
4.3 Diseño físico.....	65
4.3.1 Diagrama jerárquico funcional.....	65
4.3.2 Diseño de la interfaz.....	68
4.3.3 Especificación de programas.....	74
4.4 Implementación.....	86

CONCLUSIONES	88
---------------------------	----

BIBLIOGRAFIA	90
---------------------------	----

ANEXOS

- A.- MANUAL DE OPERACION.
- B.- GRAFICAS QUE GENERA EL SISTEMA.
- C.- FORMULAS UTILIZADAS.
- D.- CODIGO FUENTE DEL SISTEMA.



INTRODUCCION

INTRODUCCION

La Informática estudia los métodos de transformación de la información y los mecanismos que permiten realizar dicha transformación, a fin de aplicar las técnicas en las diversas áreas del conocimiento y generar información confiable, rápida, eficiente y suficiente. El desarrollo de la Informática va ligado al de los ordenadores, lo cual se refleja en la gran evolución de sistemas que son ejecutados en computadoras muy versátiles y poderosas.

Lo anterior ha abierto grandes posibilidades de desarrollo de aplicaciones en todos los campos de la actividad humana, y en particular para los procesos de enseñanza-aprendizaje.

Por otro lado, la importancia de aplicar la Estadística en los diversos campos de la actividad humana, radica en el riesgo que implica tomar una decisión en cada uno de ellos. Por ejemplo: El Gobierno se basa en el análisis obtenido de los censos que se realizan cada década, para conocer las tendencias de ciertas características de la población y así determinar las políticas de acción que apoyen o frenen el comportamiento observado, por ello es necesario contar con herramientas o alternativas que nos permitan obtener una mayor comprensión de los métodos estadísticos, para aplicarlos adecuadamente.

Una de estas herramientas la constituyen los sistemas aplicados a la Estadística (actualmente existen varios en el mercado), los cuales proporcionan una gran cantidad de procedimientos para el análisis de datos: dichos sistemas generalmente están orientados a investigadores expertos, lo que refleja la necesidad y oportunidad de crear un sistema amigable, que apoye a los usuarios implicados en el estudio y aplicación de la Estadística.

La aplicación de un sistema de información en la realización del análisis estadístico, le permite al usuario: darle prioridad a la planeación de la investigación; practicar con un mayor número de ejercicios; destinar mayor tiempo a la comprensión de los métodos estadísticos; disminuir errores en los cálculos, entre otros.

La presente tesis pretende ilustrar la importancia de aplicar la Estadística en las diversas ramas del conocimiento, mediante el diseño e implementación de un sistema informático que constituya una alternativa de apoyo en el estudio de la misma.

En el primer capítulo se hace una reseña histórica del desarrollo que ha tenido la Estadística, de sus objetivos, alcances y limitaciones, de su relación con el método científico, y de la importancia de aplicarla en las diversas ramas del conocimiento.

El segundo capítulo describe de forma general el desarrollo de sistemas de información, los tipos de sistemas y su aplicación, así como una descripción breve de las características de algunas metodologías existentes para el análisis y diseño de sistemas.

En el tercer capítulo presentamos las actividades comprendidas en el análisis del sistema; entre ellas están: un estudio de las características principales de algunos de los sistemas comerciales más conocidos, una investigación realizada acerca de los índices de reprobación en materias de Estadística que se imparte en la FES-Cuautitlán, así como la determinación, con base en los resultados de los estudios anteriores, de los requerimientos y objetivos del sistema que proponemos, además del análisis de los flujos de datos en los procesos estadísticos que conformarán el sistema.

Finalmente, en el cuarto capítulo, se presenta el diseño del sistema, el cual describe a través de los diagramas jerárquico funcional, de diseño de la interfaz y de especificación de programas, los procesos implementados en el sistema para que éste pueda cubrir los objetivos y requerimientos planteados en el análisis.



CAPITULO I
ESTADISTICA:
ASPECTOS GENERALES

1.1 ANTECEDENTES (RESEÑA HISTÓRICA)

Los orígenes de la Estadística datan mucho antes del siglo XVII; en todas partes del mundo se dedicaban al recuento de población, de nacimientos, matrimonios y defunciones, así como al registro de todo lo referente a la agricultura, caza, etc. Estos orígenes están ligados con la administración pública, ya que los recuentos que se practicaban eran con el objeto de apreciar el poderío militar, así como la recaudación de impuestos.

A mediados del siglo XVII comienza a precisarse un concepto de Estadística, entendiéndose ésta como "la descripción cuantitativa de cuanto concierne y caracteriza al Estado". En Alemania, Herman Conring hace evolucionar la Estadística, dándole una orientación que se conoció como **Estadística Universitaria**. A él le continuaron Schmeitzel y Godofredo Achenwall, entre otros; a éste último se le debe que la Estadística se convirtiera en una disciplina académica y fuera implantada en diversas universidades.

En esa misma época, en Inglaterra, John Graunt sienta las bases de la **Estadística Investigadora** mediante la realización de observaciones sobre la mortalidad en la ciudad de Londres, según las estaciones, las profesiones, etc., así como el crecimiento de la población urbana a expensas de la del campo y otros fenómenos sociales.

De esta manera, mientras que en Alemania se le daba a la Estadística una orientación hacia la descripción, en Inglaterra se le dirigía hacia la investigación.

En el siglo XIX, Adolfo Quetelet, llamado padre de la **Estadística Moderna**, se basó en las regularidades descubiertas por Graunt (y comprobadas por Süßmilch) como

¹ García Pérez, Andrés. *Elementos del Método Estadístico*, p. 9.

manifestaciones de leyes naturales, para construir un sistema de física social. Originó el método estadístico mediante la investigación numérica apoyada en el cálculo de probabilidades. Sus sucesores fueron Francis Galton y Karl Pearson, entre otros, quienes dieron un gran impulso al desarrollo de la estadística.

*Fue Gustavo Rumelin quien resolvió los desacuerdos existentes entre los que conceptuaban que el objeto de la Estadística consistía en la descripción de lo más característico de un Estado, y los que la tomaban como la investigación de los fenómenos de la sociedad humana. Rumelin superó dichas interpretaciones opuestas y confusiones que en torno a ella se producía, dividiendo la aplicación de una y otra disciplina de la parte correspondiente a la **Técnica Metodológica**. Quedaron de esta forma determinadas: 1) La Estadística Metodológica; 2) la Estadística Social; y 3) la Estadística Descriptiva. Sin embargo, con el tiempo esta división llegó a ser deficiente e inadecuada su aplicación.*

En la siguiente tabla se presentan algunos precursores de la Estadística, así como su contribución en el campo de la misma.

PRECURSORES DE LA ESTADÍSTICA

PRECURSOR	PROFESIÓN	CONTRIBUCIONES IMPORTANTES	OBRAS PUBLICADAS
GIROLAMO CARDANO ITALIANO (1501-1576)	Físico, astrologo, filósofo, matemático y médico	Se le atribuye la discusión sobre probabilidad. Autor de la fórmula de la resolución de la ecuación de tercer grado. Inició la teoría de las ecuaciones.	Manual "Liber de Ludo Aleae" (Manual para tirar los dados) Su trabajo más notable es "Ars Magna", en el cual se presentan las raíces negativas de una ecuación y algunos cálculos con números imaginarios.
PÉRE DE FERMAT FRANCS (1601-1665)	Matemático	Es considerado padre de la teoría moderna de los números y de las probabilidades. Creador, a la vez que Descartes, de la Geometría Analítica. Precursor del Cálculo Diferencial. Desarrollo, junto con Pascal, la Teoría de las Probabilidades a mediados del siglo XVII.	Autor del llamado "Último Teorema". Sus escritos principales fueron publicados en 1679 con el título de "Varia Opera Mathematica".
HERMAN CONRING ALEMAN (1606-1681)	Estadista	Elaboró un sistema del cual arranca la orientación que se conoció como Estadística Universitaria.	
JAKOB BERNOULLI BELGA (1654-1705)	Matemático	Fue uno de los primeros en estudiar los principios y aplicaciones del cálculo de probabilidades matemáticas; en su libro apareció la proposición de la Ley de los Grandes Números. Perfeccionó el Cálculo Diferencial e Integral creado por Leibnitz.	Su libro fue "Ars Conjectandi", publicado en 1713, en el cual sentó las bases del cálculo de probabilidades.
GODOFREDO ACHENWALL ALEMAN (1719-1772)	Estadista	Seguidor de Conring, se le debe que la Estadística quedase adoptada en las universidades.	Su obra fue "Abriss der neuesten Staatswissenschaft der vornehmsten europäischen Reiche und Republicken", publicada en 1749.
JOHN GRAUNT INGLES (1620-1674)	Estadista	Sentó las bases de la Estadística Investigadora.	Autor de trabajos estadísticos sobre la población londinense.
LAMBERTO ADOLFO QUETELET BELGA (1796-1874)	Astrónomo y estadista	Se le debe la Estadística Moderna. Creó la Antropometría, la Estadística Criminal, la Estadística Moral, la Psicometría, la Mecánica Estadística y la Econometría.	

PRECURSORES DE LA ESTADÍSTICA

PRECURSOR	PROFESIÓN	CONTRIBUCIONES IMPORTANTES	OBRAS PUBLICADAS
PAFNUTI LVOVICH CHEBYSHEV RUSO (1821-1894)	Matemático	Contribuyó a la teoría de los números primos, a la probabilidad, al análisis matemático, a la matemática aplicada y a la teoría de la probabilidad de Chebyshev.	
KARL PEARSON INGLES (1857-1936)	Matemático, biólogo y estadista	Se dedicó a la investigación estadística rigurosa. Sentó las bases para la aplicación de la estadística en los problemas biológicos.	Fundó la revista "Biometrika". Aportaciones matemáticas a la teoría de la evolución, publicada en 1893.
W. S. GOSSET INGLES (1876-1937)		Contribuyó con la Teoría de Student.	Sus escritos aparecieron en "Biometrika", en 1908.
RONALD LYMER FISHER INGLES (1890-1962)	Especialista en genética y estadística	Previsó los métodos estadísticos para interpretar datos cuantitativos. Desarrollo aplicaciones de la distribución F. Impulsó el uso de los procedimientos estadísticos en muchos campos, particularmente en la agricultura, biología y genética.	Su libro fue "Statistical Methods for Research Workers", apareció en 1925.
J. NEYMAN		Presentó junto con Karl Pearson una teoría sobre la verificación o prueba de hipótesis estadísticas en 1936 y 1938.	
ANDREI NIKOLAEVIC- HKOL MOGOROV RUSO (1903-)	Matemático	Presentó el primer tratamiento axiomático de la Teoría de la Probabilidad, basado en la teoría de la medida.	Lo presentó en la monografía "Grundbegriffe der Wahrscheinlichkeitsrechnung", en 1933.
WILLIAM FELLER (1906-)	Estadista	Contribuyó a la Teoría de la Probabilidad con su trabajo sobre el límite central y las cadenas de Markov.	Introdujo un nuevo tratamiento en su libro "An introduction to probability and its applications", Vol. 1, 1950, que contiene muchos ejemplos que explican nuevas aplicaciones a los fenómenos biológicos, físicos y estadísticos.
JOHN VON NEUMANN HUNGARO (1909-1957)	Matemático, físico y economista	Llevó a cabo la demostración del teorema minimax, base fundamental de la teoría de juegos. También fue el pionero de la teoría de computadoras.	

1.2 OBJETO DE LA ESTADÍSTICA

Para el estudio de los fenómenos, es necesario primeramente determinar la población, la cual estará delimitada por todos los elementos que cumplan con ciertas características. Posteriormente se realiza la recopilación, ordenación y clasificación de los datos de una muestra representativa de dicha población.

El resumen de los datos observados refleja el comportamiento de la muestra, mas no el de la población, por lo que para generalizar sobre el comportamiento de ésta, es necesario aplicar la inferencia estadística.

Es importante hacer notar que la correcta interpretación de resultados dependerá de la aplicación y comprensión de los métodos estadísticos.

1.2.1 DEFINICION

La palabra estadística proviene del vocablo latino status, el cual tiene las acepciones de estado (situación) o Estado (entidad política), o bien la situación del Estado. Es Achenwall quien deriva esta denominación del significado status = Estado.

Desde los orígenes de la Estadística se ha suscitado un sinnfin de definiciones, las cuales pueden ser consideradas desde dos puntos de vista:

- 1) Como rama de una ciencia, y que sirve para profundizar el estudio de alguno de los fenómenos que aquélla investiga; o*

2) Como una técnica especial independiente de ciencia alguna; en este sentido recibe el nombre de Metodología Estadística (técnica especial que estudia los fenómenos en masa, considerándolos como el conjunto de sucesos particulares e individuales).²

A continuación se menciona algunas de las definiciones más representativas que en torno a la Estadística se ha hecho:

La Estadística es:

"El tratamiento con métodos para obtener conclusiones a partir de los resultados de los experimentos o procesos". (Fraser)

"Un cuerpo de técnicas o metodología, que se ha desarrollado para la recopilación, presentación y análisis de datos cuantitativos, y al uso de tales datos para tomar decisiones".³

"Una ciencia pura y aplicada, que crea, desarrolla y aplica técnicas de modo que pueda evaluarse la incertidumbre de inferencias inductivas".⁴

"La descripción de las características de la sociedad humana con base en las observaciones metodológicas de enumeraciones de fenómenos similares". (Rumelin)⁵

"El arte de describir todos los objetos en razón de sus cualidades y, en rigor del término, es una lógica descriptiva. Es un razonado conocimiento de las normas generales para investigar, de las fuentes a qué recurrir, de los síntomas para reconocer,

² García Pérez, Andrés. Op. cit. p. 20.

³ Neier, John. *Fundamentos de Estadística y la Economía*, p. 19.

⁴ Robert Steel. *Biostatística: Principios y Procedimientos*, p. 2.

⁵ García Pérez, Andrés. Op. cit. p. 17.

de los principios para juzgar, de los usos a que sirven los elementos relativos al estado de las naciones". (Melchor Gioja)⁶

"Toda información relativa a estados o procesos, en la que se considera como homogéneos ciertos casos, con abstracción de sus elementos, diferenciándolos, emumerándolos y reduciéndolos a grupos".⁷

"Una rama de las matemáticas que trata de la recopilación, el análisis, la interpretación y la representación de una gran cantidad de datos numéricos". (New Collegiate Dictionary of Webster)⁸

"El conocimiento relacionado con el tomar decisiones en situaciones de incertidumbre". (Freund)⁹

Según Wundt, "la Estadística se debe considerar simplemente como un método; su designación como ciencia es, en realidad, superflua, ya que no existe sector alguno de aplicación del método estadístico que no pueda ser suficientemente definido y denominado a otras notas distintivas perfectamente reales."¹⁰

Si revisamos las definiciones mencionadas, podemos observar que todas tienen elementos en común, que son:

a) Cada definición implica una recopilación de datos teniendo como objetivo la inferencia.

⁶ García Pérez, Andrés. Op. cit. p. 16.

⁷ Schott, Sigmund. *Estadística*, p. 12

⁸ Mendenhall, William. *Estadística Matemática con Aplicaciones*, p. 1.

⁹ *Ibid.*, p. 2

¹⁰ Schott, Sigmund. Op. cit. p. 26.

b) Cada una requiere una selección de datos existente o conceptual, con el propósito de hacer inferencias con respecto a las características del conjunto completo.

De este punto podemos concluir que: La Estadística es un conjunto de métodos para la recopilación, organización, representación y análisis de un conjunto de datos representativos de una población, con el objeto de inferir sobre el comportamiento de la misma, para ayudar en el proceso de toma de decisiones.

1.2.2 OBJETIVO DE LA ESTADISTICA

El objetivo de la Estadística es: hacer inferencias con respecto a una población, a partir de la información contenida en una muestra.

1.2.3 LIMITACIONES DE LA ESTADISTICA

Existen aún muchas concepciones erróneas acerca de la naturaleza de la Estadística, en estado de rápido desarrollo. La Estadística:

- *No es un método con el que uno puede probar casi todo aquello que desea probar.*
- *No hay algo en los métodos estadísticos en sí mismos que sea capaz de evitar que el individuo superficial o intelectualmente poco escrupuloso saque sus propias conclusiones, a pesar de los datos.*
- *La Estadística no es sencillamente una colección de hechos.*
- *No constituye un sustituto del pensamiento abstracto teórico o del examen minucioso de los casos excepcionales.*
- *No sólo es aplicable en presencia de un gran número de casos, o que no puedan emplearse en los estudios de exploración.*

- *No es un sustituto de la medida o de la preparación cuidadosa de una cédula de investigación, o de otros instrumentos para la recolección de datos.*

1.2.4 RELACION CON EL METODO CIENTIFICO

Con el método científico se trata de fundamentar el conocimiento en términos de la realidad física que se propone explicar.

La Estadística es un instrumento aplicable en el método científico, para el cual fue desarrollada. Su aplicación particular está en los muchos aspectos del diseño de un experimento, desde el plan inicial para la recolección de los datos y el análisis de los resultados a partir de los datos resumidos, hasta la evaluación de la incertidumbre de toda inferencia extraída de ellos.

A continuación se presenta el papel de la Estadística en los aspectos más importantes del método científico:

1. *Formulación de hipótesis: Sugerir la hipótesis apropiada, que posteriormente deberá ser comprobada o rechazada.*
2. *Obtención de datos: El papel de la Estadística aquí es crucial, pues se trata de adquirir información de manera que: a) los datos sean relevantes al problema; y b) las conclusiones que de ellos se extraiga tengan cierto grado de fiabilidad. La cantidad de información necesaria, la forma de recolección y las técnicas para adquirirla son todos problemas del dominio de los métodos estadísticos. Las ramas de la Estadística que tratan estos problemas reciben el nombre de Diseño de Experimentos y Muestreo Estadístico.*

3. *Confrontación de la información obtenida de las consecuencias de las hipótesis postuladas: Una vez obtenidos los datos, el papel de la estadística se vuelve más importante, puesto que llega el momento de analizarlos; el primer paso consiste en la organización de los mismos, en su representación gráfica y en su descripción, resaltando los aspectos más característicos (Estadística Descriptiva). El segundo paso es realizar una generalización sobre el análisis de la información y examinar su compatibilidad con las consecuencias de la hipótesis que se sustenta. Puesto que la información será incompleta o reflejará la variabilidad del fenómeno que se observa, la generalización irá acompañada de un grado de incertidumbre que es cuantificable mediante principios probabilísticos (Inferencia Estadística o Estadística Inductiva).*

1.2.5 METODO ESTADISTICO

El método estadístico tiene dos principales funciones, que son:

- 1) Estadística Descriptiva*
- 2) Técnicas de inferencia o Estadística Inductiva.*

1.2.5.1 ESTADISTICA DESCRIPTIVA

Los primeros métodos estadísticos consistían esencialmente en la organización de datos, en su representación gráfica y en el cálculo de cantidades ejemplificantes del conjunto, con el objeto de que los aspectos sobresalientes del mismo fueran más rápida y fácilmente aprehensibles. Es a esta parte lo que se conoce como Estadística Descriptiva.

1.2.5.2 INFERENCIA ESTADÍSTICA

*Los métodos antes mencionados se vuelven de utilidad limitada debido a que la información de que se dispone representa sólo una fracción de la población, y así cualquier conclusión a la que se llegue contendrá elementos de incertidumbre. Esta incertidumbre es inherente a cualquier proceso en que se extienden conclusiones hacia un conjunto mayor que aquel sobre el que se tiene información. El método de razonamiento que nos conduce a esta extensión es conocido como **inductivo**.*

El papel de la Estadística en el proceso inductivo es cuantificar la incertidumbre, que es inseparable de las conclusiones obtenidas por su intermedio. Esta cuantificación se logra mediante el uso de los conceptos y técnicas de probabilidad. Estas técnicas, junto con el proceso inductivo, integran la rama de la Estadística Inductiva o Inferencia Estadística. Esta es la función más importante de la Estadística.

1.3 IMPORTANCIA DE LA ESTADÍSTICA

La Estadística es una disciplina que en la actualidad se encuentra inmersa en muchos aspectos de la vida moderna. Los medios de comunicación como son radio, televisión, periódicos, etc., siempre están informando acerca de las estadísticas de juegos, concursos, enfermedades, clima, etc.

Por lo anterior y con la generalización de las investigaciones interdisciplinarias, ningún estudioso de algún equipo puede ignorar los conocimientos fundamentales de la Estadística. Esta proporciona los métodos más apropiados para organizar e interpretar los datos que se obtiene en censos, muestreos, encuestas y registros de todo tipo.

1.3.1 CAMPO DE APLICACION DE LA ESTADISTICA

La aplicación de los métodos estadísticos es tan extensa, y a la vez la influencia de la Estadística en nuestra vida es tan grande, que por eso se le considera un valioso auxiliar en los más diversos campos del conocimiento y en las más variadas actividades de las ciencias fundamentales y aplicadas. La Estadística es aplicable a la Astronomía, Psicología, Medicina, Demografía, Economía, Política, Administración, Geología, Biología, Sociología, etc., es decir, en todo sector en que las decisiones deben ser tomadas con base en el análisis de muestras de datos o informes, por la imposibilidad o dificultad que implica considerarlos a todos.

En la aplicación de la Estadística los principios son generales, aun cuando las técnicas pueden diferir y la necesidad de información estadística crece a medida que se incrementa su empleo. A continuación damos ejemplos de la forma en que la Estadística es aplicada en algunos campos del conocimiento:

CIENCIAS NATURALES: Los servicios que presta el estudio de la naturaleza suelen ser de exploración, es decir, que la Estadística se emplea en las etapas preliminares de la investigación científica para ceder su sitio más tarde a la definición matemática.

CIENCIAS SOCIALES: En la investigación básica y aplicada, en los desarrollos tecnológicos que recurren al apoyo de las Ciencias Sociales, en el ámbito de la interdisciplina en las fases de investigación, sistematización y exposición de la aplicación del método científico; en todo ello es imprescindible el lenguaje y el contenido de la Estadística.

AGRICULTURA: Este campo aplica la Estadística para ayudarse en los planos y en los análisis de los experimentos agrícolas.

GOBIERNO. *El uso de la Estadística en este campo es de gran utilidad; su uso obedece al estudio y análisis para ayudar a la política del Gobierno, estudios de impuestos, gastos públicos, fondos para la asistencia pública, desempleo, etc.*

MERCADOTECNIA. *Se usa los métodos estadísticos para el control de calidad de los productos que se va a lanzar al mercado; en el estudio del mercadeo, en el análisis de la eficacia de la publicidad, etc.*

ECONOMIA: *La Estadística constituye la columna vertebral de la teoría económica. Se utiliza en la descripción de fenómenos económicos, en la estimación de las relaciones económicas, en la verificación de las teorías económicas, en la predicción y previsión de las variables económicas. La teoría económica debe formularse de manera que se tenga relaciones para estimar e hipótesis para verificar, de lo contrario la Estadística sólo sería un recurso descriptivo. Los economistas tienen que probar, en la construcción de modelos matemáticos de sus actividades económicas, la validez de sus teorías por medio de métodos estadísticos.*

ADMINISTRACION: *El creciente uso de la Estadística en la administración, es parte de la tendencia hacia basar las decisiones administrativas en un fundamento tan objetivo y científico como resulte. Con el paso del tiempo se ha introducido métodos estadísticos a más fases de la administración, con el objeto de mejorar su eficacia.*

BIOESTADISTICA Y MEDICINA: *El conocimiento de los métodos estadísticos permite determinar cuándo una variación es normal y cuándo pasa a ser patológica, dependiendo del método utilizado, además de ayudar a realizar un análisis crítico de las ventajas de la aplicación de un nuevo sistema con respecto a los ya utilizados. Por lo tanto, quienes estudian estas materias deben tener un conocimiento de la Estadística que les permita tener acceso a publicaciones de investigaciones realizadas, y aprovechar los resultados obtenidos por otros investigadores; estar en condiciones de*

transmitir la seguridad y confianza en los pronósticos realizados, mismos que dependen de su capacidad de interpretación; y contar con el criterio y las bases que les permitan ser selectivos en toda la información que perciben.

CIENCIAS DE LA CONDUCTA: *Los métodos estadísticos son de particular importancia en las Ciencias de la Conducta, debido a la variabilidad de la conducta humana. Grupos de personas deben ser estudiados para evitar el peligro de generalizar sobre la conducta de una persona a otra.*

1.3.2 FENOMENOS DE ESTUDIO

Los fenómenos que deben ser tratados estadísticamente, pueden dividirse en tres grandes grupos:

PRIMERO: *Los que no pueden ser retenidos por la observación objetiva, a saber:*

- a) Los de carácter colectivo, es decir, los que abarcan un gran número de casos particulares (ejemplos: población de un país, los salarios, etc.).*
- b) Fenómenos que se producen o se manifiestan muy separados entre sí, (los temblores, epidemias, etc.).*
- c) Los que se presentan a nuestra observación con igual frecuencia, pero impresionando más unos que otros, por lo que se tiene un concepto errado de las veces que se producen (nacimientos de hombres y mujeres, ganancias y pérdidas de un jugador, etc.).*
- d) Fenómenos que se presentan con determinada intensidad en una pequeña parte del colectivo y se cree que con esa misma intensidad existen en la totalidad.*

SEGUNDO: Fenómenos de los cuales se posee cierta apreciación cualitativa, pero desconociendo su intensidad cuantitativa (ejemplo: es conocido el hecho de que en la mayoría de los matrimonios el hombre es de mayor edad y de mayor estatura que la mujer, pero para obtener un concepto cuantitativo de estos fenómenos, es necesaria la intervención de la Estadística).

TERCERO: Fenómenos que pueden ser apreciados cuantitativamente, pero cometiendo errores graves (ejemplo: la longitud de un cuerpo puede apreciarse cuantitativamente, mas con una aproximación muy burda).¹¹

¹¹ García Pérez, Andrés. Op. cit. p. 27



CAPITULO II

ANALISIS Y DISEÑO DE SISTEMAS DE INFORMACION

2.1 ANALISIS Y DISEÑO DE SISTEMAS DE INFORMACION

Un sistema es un conjunto de componentes que se interrelacionan entre si para lograr un objetivo común. Un sistema informático es aquel que utiliza ordenadores para almacenar y procesar los datos, y poner la información a disposición de los usuarios.

La información puede llegar a ser el elemento decisivo que, en un momento dado, determine el éxito o el fracaso de un negocio, por lo que el análisis y diseño de sistemas de información es considerado como una actividad importante.

En general, el desarrollo de sistemas está formado por dos grandes componentes: el análisis de sistemas y el diseño de sistemas. El análisis de sistemas es el proceso de clasificación e interpretación de hechos que permite comprender el funcionamiento del sistema actual, hacer el diagnóstico de problemas y recomendar mejoras en el empleo de la información. El diseño de sistemas, por otra parte, es el proceso de planificar, reemplazar o complementar un sistema organizacional existente.

El análisis y diseño de sistemas de información pretende estudiar sistemáticamente la operación de ingreso, flujo de los datos y salida de la información, en un contexto particular que tiene lugar cuando se construye un sistema de información o cuando se modifica uno ya existente. En suma, el análisis y diseño de sistemas de información sirve para estudiar una situación dada, con el propósito de fomentar mejoras proponiendo diferentes alternativas, una de ellas el uso de sistemas de información computarizados.

El propósito de usar algún método para el análisis y diseño de sistemas de información, es estructurar el costoso esfuerzo de la implantación de sistemas de información, lo cual permita evitar la creación de sistemas que sean insatisfactorios en el cumplimiento de

las necesidades para el cual fue desarrollado. Para ello se apoya en un cuerpo extenso de metodologías y un conjunto bien determinado de herramientas y técnicas.

Actualmente el análisis de sistemas de información comienza con la construcción de modelos de los sistemas existentes. Estos modelos describen los procesos, datos y flujos del sistema. Mediante la construcción de modelos y la descripción de procesos, se busca ganar precisión y evitar ambigüedades, generalmente encontradas en la descripción del sistema en lenguaje natural. El diseño comienza con la definición del modelo del nuevo sistema, la cual muestra qué debe hacer el sistema para satisfacer los requerimientos. El modelo se va detallando hasta llegar al modelo final.

2.2 CATEGORIAS DE SISTEMAS DE INFORMACION

El analista de sistemas desarrolla diferentes tipos de sistemas de información, de acuerdo a los propósitos que cada situación requiere y dependiendo de las diversas necesidades del entorno.

De acuerdo a las categorías de sistemas de información, se tiene:

2.2.1 SISTEMAS PARA EL PROCESAMIENTO DE TRANSACCIONES O DE DATOS

Una transacción es cualquier suceso o actividad que afecta a toda la organización. Los sistemas de procesamiento de transacciones son los más importantes dentro de una organización; éstos tienen como finalidad mejorar las actividades rutinarias de una empresa y de las que depende toda la organización. Estos sistemas procesan grandes volúmenes de información generada de la función administrativa, sustituyendo procesos manuales por otros basados en un ordenador. Tratan con procesos de rutina bien estructurados, por ejemplo la nómina, el control de inventarios, etc.

2.2.2 SISTEMAS INFORMATICOS PARA LA ADMINISTRACION (MIS)

Estos soportan situaciones de decisión estructurada, en un espectro amplio de tareas de las organizaciones -más aún que los sistemas de procesamiento de transacciones-, incluyendo el análisis y la toma de decisiones. Los usuarios de estos sistemas utilizan una base de datos compartida para tener acceso a la información, la cual almacena tanto datos como modelos que le ayudan en la interpretación y el uso de la información para la toma de decisiones.

2.2.3 SISTEMA DE APOYO PARA LA TOMA DE DECISIONES (DSS; Decision Support System)

Los sistemas para el soporte de decisiones proporcionan información a los usuarios que deben tomar decisiones complejas, en circunstancias donde no existen procedimientos ni es posible identificar con anticipación todos los factores implicados en la decisión. Estos sistemas son diseñados para asistir en la toma de decisiones que se presentan sólo una vez o escasamente.

2.2.4 SISTEMAS EXPERTOS

El objetivo de estos sistemas es desarrollar máquinas que cuenten con un desempeño inteligente. Sus áreas de investigación son la comprensión del lenguaje natural y la habilidad para interiorizar racionalmente. Un sistema experto captura y utiliza, para resolver un problema particular, el conocimiento de un experto, siendo éste quien selecciona la mejor solución. El elemento básico de un sistema experto es la base del conocimiento.

2.3 CICLO DE VIDA DEL DESARROLLO DE SISTEMAS (SDLC; System Development Life Cycle)

Es la serie de etapas para el análisis y diseño, que enmarca el desarrollo de sistemas. Los objetivos de contar con un ciclo de vida, son:

- *Mejorar el desarrollo, especificando las tareas de cada etapa o fase.*
- *Mantener una congruencia entre los sistemas que se encuentren en desarrollo.*
- *Proporcionar puntos de control y revisión.*
- *Disminuir costos de desarrollo de sistemas.*
- *Prever posibles fracasos al crear sistemas insatisfactorios.*

2.3.1 ENFOQUES DEL CICLO DE VIDA PARA EL DESARROLLO DE SISTEMAS

Existen tres enfoques del ciclo de vida para el desarrollo de sistemas:

1. *Ciclo de vida lineal o tradicional*
2. *Ciclo de vida estructurado*
3. *Ciclo de vida por prototipos*

A continuación se describe cada una de las etapas de estos ciclos. Cabe mencionar que los diversos autores consultados difieren tanto en el número de fases como en las actividades a realizar en cada una de ellas, por lo que, de la información encontrada, proponemos las siguientes etapas en cada uno de ellos:

2.3.1.1 CICLO DE VIDA LINEAL O TRADICIONAL.

Los aspectos que caracterizan al ciclo de vida lineal para el desarrollo de sistemas son:

Una fuerte tendencia a la implantación ascendente del sistema, es decir, las fases sucesivas van elaborando el sistema incrementando el detalle; cada fase define una solución parcial y llama a la siguiente para aumentar el detalle.

Una progresión lineal de las fases; esto es, que ninguna fase puede empezar sino hasta que haya concluido la anterior, ni puede repetirse. Al final de cada etapa se genera un informe de las actividades de la misma, se elabora el plan de acción y se determina los requerimientos para la siguiente fase.

A continuación se especifica las fases que componen al ciclo de vida lineal o tradicional:

1.- Estudio o investigación preliminar

Durante la fase de estudio preliminar se determina si el problema de sistemas u oportunidad de mejora que se ha descubierto, es tan serio como para ameritar los gastos de una investigación de sistemas.

2.- Análisis

El propósito del análisis de sistemas es establecer en detalle las especificaciones de un nuevo sistema de información propuesto, o la modificación del ya existente.

En esta fase se presenta la propuesta del estudio del sistema, que es el documento inicial de la planeación del proyecto, donde se especifica los alcances, limitaciones y

recursos necesarios; se realiza la investigación detallada del sistema actual que permita evaluar su desempeño; la especificación del sistema trata de llegar a un acuerdo acerca del problema actual en el sistema; la revisión de la propuesta del estudio del sistema, es la propuesta modificada como hallazgos en la investigación; y el análisis del sistema determina lo que debe realizar el nuevo sistema, a través del informe de especificación de requerimientos del sistema.

3.- Diseño

El objetivo de la fase de diseño de un sistema de información, es producir los detalles que establecen la forma en la que el sistema cumplirá con los requerimientos identificados durante la fase de análisis.

En la búsqueda de soluciones que satisfagan potencialmente los objetivos perseguidos, se les realiza evaluaciones a cada una de las alternativas diseñadas, con el fin de poder seleccionar la más adecuada. Para cada una de las alternativas se efectúa un estudio de factibilidad, con el fin de descartar aquellos proyectos que sean inconsistentes con los objetivos que se persigue, que sean técnicamente imposibles, o que no sean redituables desde el punto de vista económico. El estudio de factibilidad consiste en evaluar:

- La factibilidad económica, donde se realiza un estudio de costo-beneficio que incluye: hardware, software, operación, mano de obra, material, etc.*
- La factibilidad técnica, que investiga si se cuenta con el equipo necesario y el personal con la experiencia requerida.*
- La factibilidad operativa, la cual se refiere al pronóstico de si, una vez instalado, el sistema llegará a utilizarse como se prevé debe hacerlo.*

Una vez hechas las evaluaciones, las alternativas son presentadas junto con las recomendaciones por parte del analista, para su elección. Posteriormente se realiza la

diagramación de las especificaciones de la opción seleccionada, y de los programas o de las modificaciones a los mismos.

4.- Desarrollo del software

En esta fase, también llamada diseño físico, el analista transmite al programador los requerimientos de programación, especificaciones funcionales y de diseño, para desarrollar todo el software original que sea necesario, o adquirir el software desarrollado por terceros.

Aquí los programadores son los responsables de la documentación de los programas y de proporcionar una explicación de cómo y por qué ciertos procedimientos se codifican en determinada forma.

5.- Prueba del sistema

Antes de ser puesto en marcha el sistema de información, debe ser evaluado su desempeño por el programador en compañía del analista, aplicando primero datos ficticios y después reales. Si el sistema se desarrolló con el fin de sustituir a otro, deberán aplicarse las pruebas en paralelo, las cuales consisten en poner en práctica ambos sistemas al mismo tiempo.

La finalidad de las pruebas es identificar posibles fallas, verificar que el sistema desarrollado en la fase de programación cumpla con las especificaciones, así como determinar su confiabilidad y eficiencia, lo que permitirá disminuir el costo de desarrollo.

6.- Implantación del sistema

Se llega a esta fase una vez que las pruebas realizadas al nuevo sistema son satisfactorias. La implantación es el proceso de verificar e instalar el nuevo sistema, establecer y probar el equipo, capacitar a los usuarios, generar todos los archivos de datos necesarios para utilizar la aplicación, así como actualizar y generar toda la documentación del sistema.

La documentación es esencial para probar el programa y llevar a cabo el mantenimiento una vez que la aplicación se encuentre instalada.

7.- Auditoría o evaluación posterior

La auditoría posterior debe realizarse cuando un sistema ha sido liberado de todos los errores; los usuarios están entrenados en la operación del sistema, y todas las partes asociadas con él amplían sus ventajas y desventajas.

La auditoría posterior evalúa si el sistema satisface las especificaciones requeridas y qué tan eficientemente se realizaron las actividades de la investigación de sistemas.

Los propósitos de la auditoría de sistemas son determinar:

- *Si el sistema resuelve el o los problemas que debe solucionar;*
- *hasta qué punto el sistema satisface las necesidades que se especifican en el informe de requerimientos del sistema;*
- *si existen nuevas oportunidades para mejorar el sistema;*
- *si el sistema se terminó a tiempo y con el presupuesto asignado, y si no, por qué;*

- *la evaluación del desempeño de los miembros del equipo de investigación de sistemas.*

8.- Mantenimiento

Durante la vida práctica de un sistema, puede generarse ligeras modificaciones que implican una pequeña variación en el sistema actual a nivel de programas o procedimientos; estos cambios deben ser aplicados en un tiempo breve, sin necesidad de reestructurar o modificar significativamente el sistema. Los cambios deberán registrarse en la documentación.

El mantenimiento durante la vida de trabajo del sistema, es necesario para adaptarlo a cualquier variación de su entorno de trabajo. Siempre son detectados errores que se puede corregir. Los responsables del sistema de información deben planificar los recursos necesarios para realizar estas funciones de mantenimiento.

El siguiente cuadro muestra las actividades, salidas y herramientas utilizadas en cada fase del ciclo de vida lineal o tradicional.

FASES Y ACTIVIDADES EN EL CICLO DE VIDA LINEAL

FASE	ACTIVIDADES	SALIDAS	HERRAMIENTAS
1. INVESTIGACION PRELIMINAR	<ul style="list-style-type: none"> Determinación del problema u oportunidad de mejora. Aprobación de la investigación. Establecer los alcances y límites del proyecto. Estimación de costo, tiempo necesario para terminarlo y necesidades del personal. 	Informe preliminar.	Entrevistas. Diagramas de flujo. Tablas de organización. Manuales operativos. Descripción de puestos. Cuestionarios. Observación.
2. ANALISIS	<ul style="list-style-type: none"> Elaborar la propuesta del estudio del sistema. La investigación del sistema. La especificación del problema. La revisión de la propuesta del estudio del sistema. El análisis de sistema. Especificaciones de requerimientos del sistema. 	Propuesta del estudio del sistema. Informe de especificaciones de requerimientos del sistema. Diseño conceptual.	Lenguaje estructurado, tablas y árboles de decisiones. Proceso de intercambio y la aplicación de métodos ponderados. Entrevistas. Diagramas de flujo. Tablas de organización. Manuales operativos. Descripción de puestos. Cuestionarios. Documentación de sistemas. Revisión de formas. Seguimiento de documentos. Observación directa. Medición del trabajo. Examen de otros sistemas.
3. DISEÑO	<ul style="list-style-type: none"> Diseño de alternativas. Estudio de factibilidad para cada una. Evaluación de las alternativas. Presentar estudios y recomendación. Diseño del sistema de información, según la alternativa seleccionada. Especificación de los programas o de las modificaciones a los mismos. Selección del equipo necesario. 	Macro-diseño de cada alternativa. Diseño de: <ul style="list-style-type: none"> Sistema de archivos o bases de datos; sistema de acceso o de seguridad; diagramas de flujo de sistema; procedimientos para la captura; salidas por pantalla e impresora; formas; interfaz. 	Diseño de formas.

FASES Y ACTIVIDADES EN EL CICLO DE VIDA LINEAL

FASE	ACTIVIDADES	SALIDAS	HERRAMIENTAS
4. DESARROLLO	Desarrollo de los programas de aplicación.	Codificación de los programas. Configuración del equipo. Programa fuente. Programa objeto.	
5. IMPLANTACION	<ul style="list-style-type: none"> • Educación y capacitación al personal. • Programación o planeación de la transición. • Preparación del lugar. • Instalación y prueba del equipo. • Instalación y prueba del software comprado. • Conversión de archivos. • Uso del nuevo sistema. • Prueba final y aceptación. • Realizar la documentación de: 	<p>Sistema en operación.</p> <ul style="list-style-type: none"> • Información histórica del sistema (solicitud, planteamientos, diagramas de procesamiento, etc.). • Manuales de operación. • Manuales de procedimientos. • Instructivos de recolección y llenado de formas. • Carpeta de programas. 	
6. AUDITORIA O EVALUACION	<p>Determinar si:</p> <ul style="list-style-type: none"> • el sistema resuelve el o los problemas que trata de resolver. • hasta que punto el sistema satisfase las necesidades que se especifican en el informe de requerimientos del sistema. • después de la experiencia con el sistema, si existen nuevas oportunidades para mejorarlo. • el sistema se terminó a tiempo y con el presupuesto asignado, y si no, por qué. • evaluar el desempeño de los miembros del equipo de investigación de sistemas. 	Resultados de la evaluación.	
7. MANTENIMIENTO	<ul style="list-style-type: none"> • Realizar modificaciones menores. • Reflejar las actualizaciones en la documentación del sistema. 	Programa fuente. Programa objeto. Actualización de la documentación.	

2.3.1.2 CICLO DE VIDA ESTRUCTURADO

El ciclo de vida estructurado, a diferencia del ciclo de vida lineal, permite que más de una actividad se lleve a cabo a la vez. No implica que se deba terminar una actividad para poder comenzar la siguiente.

Este ciclo no se considera secuencial, es por lo mismo que se prefiere llamar actividades y no fases del ciclo de vida estructurado, entendiéndose como fase "un periodo particular en un proyecto, en el cual se estaba desarrollando una y sólo una actividad".¹

A continuación se especifica las actividades que componen el ciclo de vida estructurado:

1.- Estudio de factibilidad o viabilidad

Conocido también como estudio inicial, es la actividad donde se determina a los usuarios implicados en el sistema. La finalidad es el estudio del sistema actual, para determinar las deficiencias actuales. Una vez que éstas fueron identificadas, se establece los objetivos y requerimientos para el nuevo sistema.

Dentro de esta actividad está comprendido un estudio de la viabilidad técnica, operacional y económica, con el objetivo de verificar si el proyecto es factible de ser efectuado.

Al término de esta fase se entrega un informe especificando una solución conceptual, así como una aproximación del costo-beneficio y el tiempo requerido para el desarrollo del sistema.

¹ Yourdon, Edward. *Análisis Estructurado Moderno*. p. 104.

2.- Análisis

El propósito de esta actividad es un estudio de forma detallada del sistema actual, diseñando un modelo del mismo, con la finalidad de representar y comprender mejor los componentes del sistema.

En esta actividad también se plantea la solución, creando el modelo esencial que representa la descripción formal de lo que el nuevo sistema debe hacer; ésta deberá presentarse en un informe junto con los presupuestos de costo-beneficio en forma detallada.

3.- Diseño

El diseño de sistemas se compone de dos partes: diseño lógico y diseño físico. El primero de ellos tiene como objetivo obtener los datos necesarios y diseñar los procesos adecuados para el nuevo sistema. El diseño físico se refiere a la definición de programas y las interfaces del mismo, las cuales lograrán cubrir los objetivos del sistema.

4.- Implantación

En esta actividad se logra la codificación e integración del sistema, es decir, llevar a cabo la programación del sistema con las especificaciones realizadas en la actividad de diseño.

5.- Pruebas de aceptación

Una vez que se haya codificado e integrado el sistema, se debe verificar que éste sea aceptado por el usuario y que cumpla con los objetivos especificados.

6.- Garantía de calidad

Es conocida como prueba final o pruebas de aceptación; ésta se refiere a la revisión del sistema verificando que cumpla con un nivel apropiado de calidad, prueba que se realiza tanto en el análisis como en el diseño del sistema.

7.- Conversión

Una vez que el sistema haya sido probado y aprobado, se procede a la conversión de los ficheros y base de datos.

8.- Instalación

Es la actividad que pone en marcha el sistema, dejándolo listo para ser operado por el usuario.

9.- Revisión postimplantación

Las actividades no terminan con la instalación del sistema; la actividad que le sigue a ésta, es la revisión postimplantación, que se refiere a la verificación constante del sistema para comprobar que está funcionando adecuadamente o poder hacer las correcciones correspondientes en el momento en que sea detectada alguna falla.

El siguiente cuadro muestra las actividades, salidas y herramientas utilizadas en cada fase del ciclo de vida estructurado.

FASES Y ACTIVIDADES EN EL CICLO DE VIDA ESTRUCTURADO

ACTIVIDADES	TAREAS	SALIDAS	HERRAMIENTAS
1. ESTUDIO DE FACTIBILIDAD	<ul style="list-style-type: none"> • Determinar si es factible automatizar el sistema. • Determinar el problema o las deficiencias del sistema actual • Determinación de requerimientos. • Establecer objetivos para un sistema nuevo. • Preparar un plan como guía del proyecto 	<p>Informe de estimación de costos y beneficios esperados del sistema Propuesta del tiempo requerido Asignación del responsable del proyecto</p>	<p>Entrevistas Cuestionarios Documentos Manuales Programas existentes. Diagrama de flujo de datos general</p>
2. ANÁLISIS	<ul style="list-style-type: none"> • Estudio de las entidades externas que interactúan con el sistema. • Analizar los flujos de datos entre las entidades y el sistema. • Estudiar la operación física del sistema. • Modelo esencial del nuevo sistema. 	<p>Descripción formal de lo que el nuevo sistema debe hacer (modelo detallado del sistema). Informe de las funciones, estructura de datos y flujos de información del sistema.</p>	<p>Diagramas de flujo de datos. Diagrama de estructura de datos. Diccionario de datos. Texto estructurado. Tablas de decisión y árboles de decisión. Diagramas de Entidad-relación. Diagrama de transición de estados</p>
3. DISEÑO	<p>DISEÑO LÓGICO</p> <ul style="list-style-type: none"> • Obtener datos y diseñar procesos para el nuevo sistema. <p>DISEÑO FÍSICO</p> <ul style="list-style-type: none"> • Dividir el sistema en módulos, subsistemas y submódulos. • Determinar los procesos que serán llevados en forma manual y cuales en ordenador • Elegir dispositivos físicos para almacenar los datos • Desarrollar la interfaz entre el usuario y el ordenador. • Crear planes de prueba. • Crear especificaciones de programas. • Elaborar plan para efectuar el desarrollo e implantación del nuevo sistema. • Revisar y estimar costos de operación del nuevo sistema. 	<p>Especificación del sistema lógico.</p>	<p>Diagrama jerárquico funcional Especificación de programas. Diseño de la interfaz.</p>
4. IMPLANTACION	<ul style="list-style-type: none"> • Escoger software para la programación del sistema. • Codificación e integración de módulos. 	<p>Sistema codificado e integrado</p>	<p>Programación estructurada. Implementación descendente.</p>
5. PRUEBAS DE ACEPTACION	<ul style="list-style-type: none"> • Probar programas. • Realizar pruebas al sistema. 	<p>Datos obtenidos de las pruebas de aceptación.</p>	

FASES Y ACTIVIDADES EN EL CICLO DE VIDA ESTRUCTURADO

ACTIVIDADES	TAREAS	SALIDAS	HERRAMIENTAS
6. GARANTIA DE CALIDAD	<ul style="list-style-type: none"> • Verificar que el sistema tenga un nivel apropiado de calidad. 	Análisis de alta calidad. Diseño de alta calidad. Código de alta calidad. Aceptación del sistema.	
7. CONVERSION	<ul style="list-style-type: none"> • Preparar manuales del usuario. • Realizar conversión de ficheros. 	Bases de datos convertidas. Sistema aceptado en marcha. Manual del usuario.	
8. INSTALACION	<ul style="list-style-type: none"> • Puesta en marcha del sistema. 	Sistema aceptado en marcha.	
9. REVISION POSTIMPLANTACION	<ul style="list-style-type: none"> • Empezar monitorización inicial del sistema. 	Reporte de la monitorización.	

2.3.1.3 CICLO DE VIDA POR PROTOTIPOS

Boar lo describe como:

“Una alternativa de enfoque para la definición de los requerimientos, consiste en capturar un conjunto inicial de necesidades e implantarlo rápidamente con la intención declarada de expandirlo y refinarlo iterativamente al ir aumentando la comprensión que del sistema tienen el usuario y quien lo desarrolla. La definición del sistema se realiza mediante el descubrimiento evolutivo y gradual, y no a través de la previsión omnisciente...². Este tipo de enfoque se llama ‘de prototipos’. También se conoce como modelado del sistema o desarrollo heurístico. Ofrece una alternativa y es practicable a los métodos de especificación para tratar

² Del latín *omnis*, todo, y *sciens*, que sabe, adj. Que conoce todas las cosas, atributo sólo de Dios.

*mejor la incertidumbre, la ambigüedad y la volubilidad de los proyectos reales”.*³

Este enfoque se basa en la creación de una serie de modelos de un sistema, llamados “prototipos”, los cuales determinan cada vez con mayor detalle el sistema final. Estos modelos son realizados con la finalidad de ilustrar la viabilidad, identificar los requerimientos, probar ideas y suposiciones relacionadas con un nuevo sistema, y dirigirse hacia su desarrollo.

El principio fundamental del desarrollo del prototipos es el siguiente:

*“Los usuarios pueden señalar las características que les agradaría o no tener, junto con los problemas que presenta un sistema que existe y funciona, con mayor facilidad que si se les pidiese que las describieran en forma teórica o por escrito. El uso y la experiencia producen comentarios más significativos que el análisis de diagramas y las propuestas por escrito”.*⁴

Las etapas que involucra la creación de un sistema mediante el ciclo de vida por prototipos son:

1.- Factibilidad

Determinar si el proyecto es factible: técnica, operativa y económicamente.

³ Yourdon, Edward. Op. cit. p. 108

⁴ Senn, James A. *Análisis y Diseño de Sistemas*. p. 21.

2.- Determinar si es un buen candidato

Estimar si el ciclo de vida por prototipos es el adecuado para el problema y circunstancias que se tiene. Los analistas encuentran que el uso de los prototipos es adecuado cuando:

- *Los encargados de diseñar e implantar sistemas nunca han desarrollado uno con las características del sistema propuesto;*
- *sólo se conoce una parte de las características esenciales del sistema;*
- *existen situaciones de riesgo y costos elevados;*
- *el diseño propuesto es novedoso y aún no ha sido probado;*
- *el usuario no puede o no está dispuesto a examinar modelos abstractos en papel -tales como diagramas de flujo de datos-, ni a preespecificar sus requerimientos, por lo que sólo se los puede determinar mediante un proceso de ensayo y error;*
- *se requiere rapidez en el desarrollo, más que eficiencia en el procesamiento;*
- *el sistema no requiere de la especificación de una gran cantidad de detalles algorítmicos, ni de muchas especificaciones de procesos para describir los algoritmos con los cuales se obtiene los resultados.*

3.- Identificación de requerimientos

El analista debe identificar los requerimientos de información que el usuario conoce, así como las características necesarias del sistema.

4.- Desarrollo de un prototipo funcional

Con base en los requerimientos de la fase anterior, se desarrolla un prototipo que los cumpla, utilizando herramientas para el desarrollo de software.

5.- Revisión del prototipo y obtención de refinamientos

El prototipo del sistema es revisado a través de la información y experiencia del usuario, para identificar las posibles fallas, nuevas necesidades o requerimientos, cambios y mejoras al prototipo. Esta información le servirá al analista y diseñador para refinar el prototipo.

6.- Realizar prototipos evolutivos

Revisar y realizar las modificaciones al prototipo, con base en los nuevos requerimientos obtenidos.

Las etapas de la tercera a la sexta, son repetidas tantas veces como sea necesario, hasta obtener los requerimientos esenciales del diseño.

Cuando el analista y el usuario deciden que ya cuentan con la suficiente información proveniente del proceso de construcción del prototipo, determinan satisfacer los requerimientos ya identificados. En general, se opta por una de las siguientes opciones: por volver a desarrollar el prototipo; implantar el prototipo como sistema terminado; abandonar el proyecto por considerarlo insuficiente para satisfacer los objetivos; o bien, iniciar la construcción de otra serie de prototipos con un enfoque distinto.

La siguiente tabla nos muestra las actividades y herramientas utilizadas en cada fase del ciclo de vida de prototipos.

FASES Y ACTIVIDADES EN EL CICLO DE VIDA POR PROTOTIPOS

FASE	ACTIVIDADES	SALIDAS	HERRAMIENTAS
1. FACTIBILIDAD	Determinar si el problema es adecuado para ser resuelto por el método por prototipos.	Análisis costo-beneficio del proyecto.	Herramientas para el desarrollo de software. Reutilización de código. Un diccionario de datos integrado.
2. DETERMINAR SI ES UN BUEN CANDIDATO	Evaluar las características del problema y condiciones existentes.	Determinación de si proseguir o seleccionar otro método.	Un generador de pantallas. Un generador de reportes no guiado por procedimientos. Un lenguaje de programación de cuarta generación.
3. IDENTIFICAR REQUERIMIENTOS	Identificación de requerimientos.	Reporte de requerimientos.	Un lenguaje de consultas no guiado por procedimientos. Medios poderosos de administración de bases de datos.
4. DESARROLLO DE UN PROTOTIPO FUNCIONAL.	Diseño y programación del prototipo.	Prototipo funcional.	
5. REVISION DEL PROTOTIPO Y OBTENCION DE REFINAMIENTOS	Evaluar el prototipo. Obtener nuevos requerimientos, mejoras o cambios.	Mejoras al prototipo.	
6. REALIZAR PROTOTIPOS EVOLUTIVOS	Modificación del prototipo con base en los resultados obtenidos en la fase anterior.	Prototipo mejorado.	



CAPITULO IV

ANALISIS DEL SISTEMA DE ESTADISTICA PROPUESTO

3.1 ANALISIS DEL SISTEMA DE ESTADISTICA PROPUESTO

El caso práctico de la presente, consiste en el desarrollo de un sistema informático aplicado a la Estadística, cuyo análisis se expone a continuación, dejando para el siguiente capítulo el diseño. El desarrollo se realizó siguiendo las actividades del ciclo de vida estructurado, por ser éste el que se consideró apropiado al tipo de problema presentado.

3.2 ESTUDIO DE FACTIBILIDAD

3.2.1 DETERMINACION DE LA OPORTUNIDAD

La aplicación de un sistema en el estudio de la Estadística, proporciona: a) al alumno, la posibilidad de practicar en él, adquirir experiencia en la manipulación de datos mediante medios electrónicos, así como manejar sistemas que le permitirán por analogía manipular otros; y b) al profesor, la ventaja de poder proponer diseños experimentales como alternativa, que estén orientados al ámbito de estudio de la carrera en la que imparte la materia, y haga de ello un curso más participativo y práctico.

Por otra parte, en el ambiente laboral, por la premura con que se debe tomar las decisiones, es de vital importancia contar con un auxiliar que agilice dichos procedimientos; esto se logra si se cuenta con un sistema estadístico amigable que constituya realmente una ayuda, es decir, que el usuario no se tenga que preocupar por el manejo del mismo, sino que éste sea más intuitivo; de otro modo, demoraría más el proceso de obtención de resultados, que si se desarrollara manualmente.

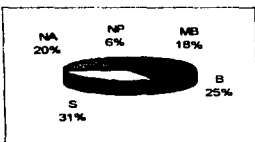
El uso de un sistema estadístico apoya al usuario en el análisis estadístico mediante la generación de los cálculos, lo que le permite dar prioridad y más tiempo al planteamiento del problema, al diseño del muestreo, al estudio del procedimiento a aplicar, y a la correcta interpretación de resultados, para tomar una decisión de manera rápida y eficiente, optimizando con ello recursos tanto humanos como financieros. De esta manera el usuario, al delegar la tarea de realizar los cálculos, reduce la probabilidad de generar resultados erróneos, además de poder visualizar gráficamente los fenómenos de estudio.

3.2.2 IDENTIFICACION DE LAS DEFICIENCIAS ACTUALES

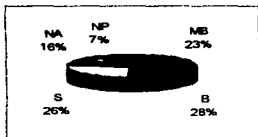
Una vez identificada la oportunidad de crear una herramienta de estudio, como son los paquetes o sistemas aplicados a la materia de interés, nos dedicamos a recopilar los índices de reprobación comprendidos en los periodos 91-1 al 95-2 de las materias de Estadística de las diferentes carreras que se imparten en la Facultad de Estudios Superiores Cuautitlán, a fin de contar con información que nos muestre el comportamiento de los resultados obtenidos por los estudiantes en el estudio de dichas materias. También realizamos un estudio de las características de los sistemas estadísticos existentes, lo cual nos permitirá determinar los requerimientos y objetivos que deberá cumplir el nuevo sistema.

3.2.2.1 INDICES DE REPROBACION

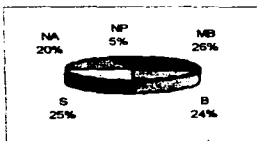
A continuación se presenta las gráficas que muestran los índices de reprobación por materia, resultantes de la información recopilada.



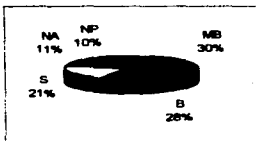
Materia: Estadística Descriptiva
Carrera: Licenciatura en Administración
Licenciatura en Contaduría



Materia: Taller de Estadística
Descriptiva
Carrera: Licenciatura en Administración
Licenciatura en Contaduría

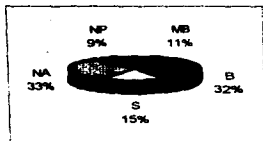


Materia: Inferencia Estadística
Carrera: Licenciatura en Administración
Licenciatura en Contaduría

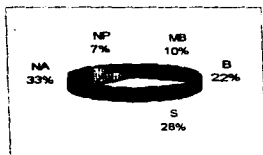


Materia: Taller de Inferencia Estadística
Carrera: Licenciatura en Administración
Licenciatura en Contaduría

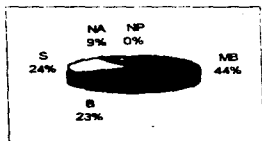
Capítulo III: Análisis del sistema de estadística propuesto



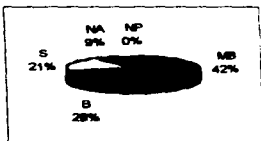
Materia: Probabilidad y Estadística
Carrera: Licenciatura en Administración
Licenciatura en Contaduría



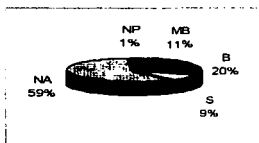
Materia: Estadística III
Carrera: Ingeniería Agrícola



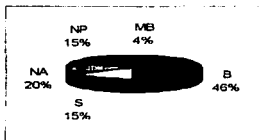
Materia: Estadística I
Carrera: Licenciatura en Informática



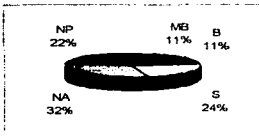
Materia: Estadística II
Carrera: Licenciatura en Informática



*Materia: Estadística Aplicada I
Carrera: Ingeniería Mecánica y
Eléctrica*



*Materia: Estadística Aplicada II
Carrera: Ingeniería Mecánica y
Eléctrica*



*Materia: Bioestadística
Carrera: Medicina Veterinaria y
Zootecnia*

Si agrupamos los índices de reprobación por carrera, vemos que su comportamiento es similar entre las materias, a excepción de las asignaturas de la carrera de IME, donde se presenta mucha variación entre ellos. También podemos observar que las materias prácticas, como son los talleres que se imparte en las carreras de Administración y Contaduría, presentan menor índice de reprobación en relación a las materias teóricas de las mismas carreras. Por otra parte, las carreras que registran un mayor índice de reprobación en materias de Estadística son: Medicina Veterinaria y Zootecnia, e Ingeniería Agrícola.

3.2.2.2 CARACTERÍSTICAS DE LOS SISTEMAS ESTADÍSTICOS COMERCIALES

Existen en el mercado algunos sistemas de Estadística que facilitan el uso de la mayoría de los procedimientos estadísticos. Algunos de los más conocidos son: SAS, SPSS, Statgraphics y Stata, entre otros, algunos de los cuales corren en varias plataformas.

A continuación se describe las características de cada uno de los sistemas antes mencionados, con el fin de conocer su alcance, orientación, ámbito de aplicación, ventajas y limitaciones, lo que nos permitirá determinar las necesidades actuales.

3.2.2.2.1 SAS (Sistema para Análisis Estadístico)

Sistema computacional que se terminó de desarrollar en 1972 con el trabajo de Bary Goodnight en la Universidad Estatal de Carolina del Norte. Es considerado como un lenguaje de computación (que trabaja en forma de intérprete), el cual cubre la mayor parte de las necesidades de procesamiento electrónico de información estadística.

Está dirigido a investigadores: usuarios de los procedimientos estadísticos, quienes crearán sus propios programas codificando la rutina que consideren apropiada para el análisis. Por lo anterior es que se requiere que tenga cierta experiencia en términos y procedimientos estadísticos, así como de programación no estructurada.

Para su utilización, el usuario debe codificar su información experimental e ir especificando línea tras línea las operaciones a realizar, tales como de dónde debe leer los datos, definir las variables, funciones y procedimientos que afectarán a los datos, etc.

Como lenguaje de programación, cuenta con procedimientos estadísticos, funciones, expresiones, enunciados de control y macros, los que determinan su condición.

Las instrucciones de programación permiten al usuario de SAS trabajar sus datos para realizar operaciones como:

- a) Transformar valores.*
- b) Crear nuevas variables.*
- c) Acumular totales.*
- d) Perforar tarjetas de datos.*
- e) Seleccionar ciertas observaciones o variables a procesar.*

La gama de procedimientos permite realizar estadísticos descriptivos, análisis de regresión simple y múltiple mediante el método de mínimos cuadrados, prueba de hipótesis, el coeficiente de correlación, permutaciones, entre otros.

Características:

- La salida de la computadora se compone de dos partes: la primera, lista todas la instrucciones con notas acerca de los conjuntos de datos generados; y la segunda, contiene la salida producida por los procedimientos SAS.*
- Permite especificar el encabezado a imprimir (máximo 10 líneas).*
- Reconoce los valores perdidos que se presentan en las colecciones de datos y los manipula de manera especial para cada procedimiento.*
- Sólo ejecuta los enunciados que no se ven afectados por algún error (por lo que se considera como intérprete).*
- El contenido de un conjunto de datos SAS puede generarse por medio de instrucciones de programación.*

- *Permite ordenar, clasificar, crear subgrupos, concatenar e intercalar conjuntos de datos.*

3.2.2.2.2 SPSS (Statistical Package for the Social Sciences)

Es un paquete estadístico para el análisis de datos; originalmente su diseño estuvo orientado hacia las aplicaciones de las Ciencias Sociales, y aunque su vocabulario lo muestra, SPSS puede ser utilizado por investigadores de cualquier disciplina.

SPSS operaba primeramente en grandes sistemas computacionales, lo cual limitaba su acceso. En 1984, SPSS Inc. introdujo SPSS PC[®] para computadoras personales, corriendo bajo el sistema operativo MS-DOS; posteriormente se desarrolló el mismo sistema, pero en diversas plataformas como Unix, OS 2 y Macintosh (1990), y en varios idiomas como en francés, inglés, italiano, japonés y español. En 1992, SPSS llegó a ser el mejor software estadístico desarrollado para Windows.

SPSS es un conjunto de comandos para desarrollar los siguientes tipos de análisis: distribución de frecuencias, correlación, regresión, análisis de varianza, modelos lineales generales, tablas de contingencia, análisis del factor y discriminantes, estadísticas no paramétricas, series de tiempo, gráficas, entre otros. Además, cuenta con funciones matemáticas muy variadas como son: numéricas, generadoras de números aleatorios, logarítmicas, trigonométricas, etc.

En este paquete se puede trabajar mediante el sistema de menús (realizando un programa), línea de comandos (introduciendo los comandos uno por uno en el prompt) o bien de forma interactiva alternado las dos opciones anteriores (lo que le permitirá dar entrada a respuestas en el prompt y obtener ayuda).

Para que el usuario pueda hacer un correcto manejo del paquete, es necesario que primeramente conozca la función de cada uno de los comandos que el sistema proporciona, así como de cada opción que los acompaña, y posteriormente desarrolle la rutina con los comandos adecuados, que le permitan analizar los datos.

SPSS es muy similar al paquete estadístico SAS, por lo que, si el usuario cuenta con un entendimiento básico de los comandos de SPSS, podrá con mínimas dificultades preparar programas SAS para desarrollar operaciones similares, y viceversa: basta con aprender la sintaxis particular y las diferencias entre ambos. Incluso se puede hacer uso de los mismos datos.

Características:

- *Proporciona un sistema de ayuda que muestra información relevante acerca del comando sobre el que se solicita, y un editor.*
- *Suministra un subsistema llamado CMS Batch, que es otra alternativa para ejecutar un programa SPSS, cuando éste requiere memoria mayor a 3 Megabytes o el tiempo de ejecución es largo.*
- *Cuenta con un sistema de archivos SPSS-X, que constituye un eficiente camino para salvar datos que han sido accedidos por múltiples programas SPSS.*
- *Permite la alimentación de datos mediante archivos realizados con editores externos (importación de datos) o con el editor propio (el cual permite exportar datos).*
- *Realiza la transformación de variables, recodificación, generación de nuevas variables, selección de casos, selección de muestras aleatorias, así como el manejo de valores faltantes.*
- *Da facilidades para manipular grupos de datos como son: ordenación, concatenación, combinación e intercalación.*

3.2.2.2.3 STATGRAPHICS

Statgraphics es un paquete de software que integra la captura y manipulación de datos a una amplia gama de procedimientos de análisis estadístico, cálculos matemáticos y generación de gráficas de alta resolución.

El paquete cuenta con dos tipos de procedimientos: los propios del sistema y los estadísticos. Los procedimientos de sistema le permiten al usuario realizar tareas no analíticas, como formar archivos de datos, crear presentaciones, modificar el ambiente del sistema, etc. Los procedimientos estadísticos permiten analizar datos y producir resultados en formatos de texto y gráficos.

Cuenta con procedimientos estadísticos como son: métodos descriptivos, estimación, muestreo, prueba de hipótesis, análisis de datos exploratorios, análisis de varianza, análisis de regresión, series de tiempo, y procedimientos avanzados como: análisis de datos categóricos, métodos multivariable, métodos no paramétricos, tiempo de vida de un análisis de datos, diseño experimental, funciones matemáticas y otras funciones.

Requerimientos:

Es necesario que el usuario cuente con conocimientos sobre Estadística que le permitan seleccionar las rutinas de análisis e interpretar adecuadamente los datos.

Para su manipulación, el usuario deberá capturar los datos en un archivo mediante el editor integrado; posteriormente podrá seleccionar los procedimientos estadísticos de su interés, los cuales le solicitarán la entrada de datos específicos, entre ellos el nombre del archivo donde se encuentran los datos sobre los cuales va a operar.

Características:

- *Provee al usuario de las herramientas necesarias para correr procesos estadísticos de principio a fin, con amplia variedad de opciones para cada proceso.*
- *Ofrece diversas alternativas para especificar los datos, por ejemplo combinar dos o más archivos de datos.*
- *Los resultados emitidos pueden ser editados mediante un procesador de palabras, antes de ser grabados en un archivo o impresos.*
- *Permite producir y editar gráficos de alta resolución.*
- *Soporta una amplia variedad de impresoras de matriz, laser y de burbuja, así como graficadores de pluma.*
- *Permite la importación de datos capturados en editores u hojas de cálculo.*

3.2.2.2.4 STATA

Es un paquete estadístico que está diseñado para soportar y desarrollar tutoriales interactivos para estudiantes. Tiene varias características de construcción que hacen posible programar sesiones interactivas.

Su poder, portabilidad y programabilidad, hacen de Stata una muy buena, pero no única, plataforma para desarrollar software (tutoriales interactivos) para un curso singular.

Para que un instructor pueda crear tutoriales, necesita conocer lo suficientemente bien los comandos de Stata.

Algunos de los tutoriales que se han escrito con ayuda de Stata, son: Un ejercicio para demostrar el experimento de Monte Carlo, un tutorial para demostrar el Teorema del Límite Central, uno para practicar los cálculos de los significados condicionales y varianzas, otro para ayudar a los estudiantes a minimizar la suma de residuos de cuadrados en una regresión, así como uno para practicar el desempeño de los tests de hipótesis y uno para aprender a usar los comandos del mismo Stata.

Características:

- *Stata corre en varias plataformas, entre ellas: DOS, Windows, OS-2, Macintosh, UNIX.*
- *Cuenta con características de programación estructurada.*
- *Incluye tutoriales.*
- *Crea programas completamente interactivos.*
- *Permite la programación de exámenes a estudiantes, para checar su desempeño durante y después de la utilización del tutorial.*
- *Hace factible la repetición de problemas de práctica.*
- *Los estudiantes pueden incluso correr el tutorial en su casa, sin necesidad de instructor.*
- *Los tutoriales creados, son programas Stata.*
- *Algunos programas despliegan simplemente instrucciones y explicaciones en la pantalla, otros realizan problemas prácticos o permiten a los estudiantes introducir datos.*

3.2.3 DETERMINACION DE REQUERIMIENTOS

Con base en los resultados de los estudios antes mencionados, nos damos cuenta de que el comportamiento de los índices de reprobación es muy variable en cada materia, lo que nos indica que son muchos los factores que intervienen en el aprovechamiento del alumno.

Por otra parte, mediante el análisis de los sistemas más comerciales, encontramos las limitantes con que un usuario se enfrenta al querer utilizar algún paquete de Estadística, por las características de los mismos, entre las cuales están: existen pocos paquetes comercializados, en la mayoría de los casos, desarrollados en idioma inglés; están orientados a usuarios expertos en el uso de los métodos estadísticos y con ciertas habilidades en la programación y el manejo de datos; el precio de la licencia de uso es muy alto; se requiere que el usuario cuente con equipo disponible, con ciertas características y capacidades para su instalación y operación; tener conocimientos para poder instalarlo; y para aquellos usuarios que empiezan su aprendizaje en computación, generalmente es necesario tomar un curso para poder manejarlo.

Por lo anterior, creemos que es necesario contar con herramientas orientadas especialmente a los estudiantes, que les permitan tener acceso a nuevos caminos para adquirir los conocimientos y la práctica. El sistema que se propone deberá, para cumplir con las necesidades detectadas, tener las siguientes características:

- *Proporcionar una interfaz amigable con el usuario.*
- *Generar los procedimientos estadísticos más utilizados.*
- *Mostrar en pantalla las fórmulas utilizadas en el desarrollo del procedimiento.*
- *Contar con un sistema de ayuda.*

- *Proporcionar un sistema de mensajes de ayuda y alerta en línea.*
- *Necesitar un mínimo de requerimientos de equipo para su funcionamiento.*
- *Producir las salidas necesarias, como son fórmulas, gráficas, tablas y datos, tanto en pantalla como en impresora.*
- *Buscar el valor de tablas, en las tablas de distribución de la probabilidad t , z y χ^2 cuadrada.*

3.2.4 ESTABLECIMIENTO DE LOS OBJETIVOS DEL NUEVO SISTEMA

El sistema de Estadística que se propone, estará orientado principalmente a los estudiantes que cuentan con conocimientos básicos de Estadística; no obstante, también puede servirles a todos aquellos usuarios que tengan conocimientos más avanzados de la Estadística y que constantemente la estén utilizando. El sistema tiene como objetivos los siguientes:

- *Servir como herramienta de apoyo al estudiante, permitiéndole la práctica constante de ejercicios vistos en clase y estimulándolo a practicar con problemas aplicables en su área de interés.*
- *Incrementar la confianza en el tratamiento de datos a través de medios computarizados, motivando al usuario a buscar nuevas herramientas de este tipo que lo apoyen en otras áreas de estudio o aplicación.*
- *Aligerar la carga de trabajo en el proceso de la obtención de resultados estadísticos.*
- *Apoyar la rápida toma de decisiones donde éstas deban ser establecidas en el menor tiempo posible, mediante la consecución de resultados rápidos, confiables y precisos.*
- *Minimizar la cantidad de errores obtenidos al realizar los cálculos manualmente.*

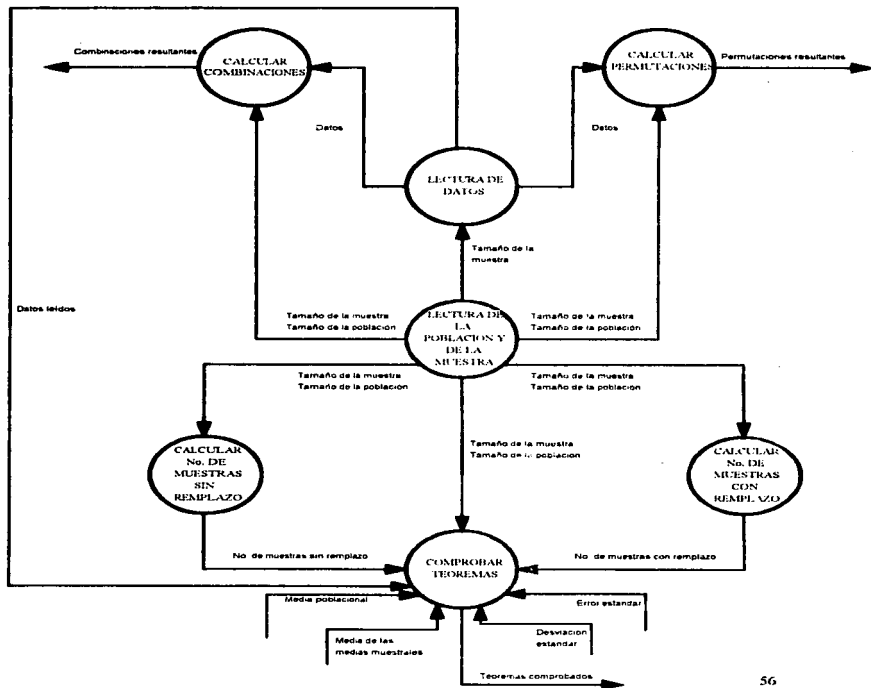
3.3 ANALISIS DEL NUEVO SISTEMA

Para que el diseño e implementación del sistema cumpla con los objetivos anteriores, es necesario identificar y realizar un análisis de los procedimientos que serán incluidos en el sistema; para ello nos apoyamos en los diagramas de flujo de datos, que muestran gráficamente el recorrido y obtención de datos a través de los procesos. Otra herramienta en esta fase la constituyen las tablas de decisión, utilizadas en aquellos casos donde existen varias condiciones, acciones y alternativas, para identificar claramente los cursos de acción a seguir.

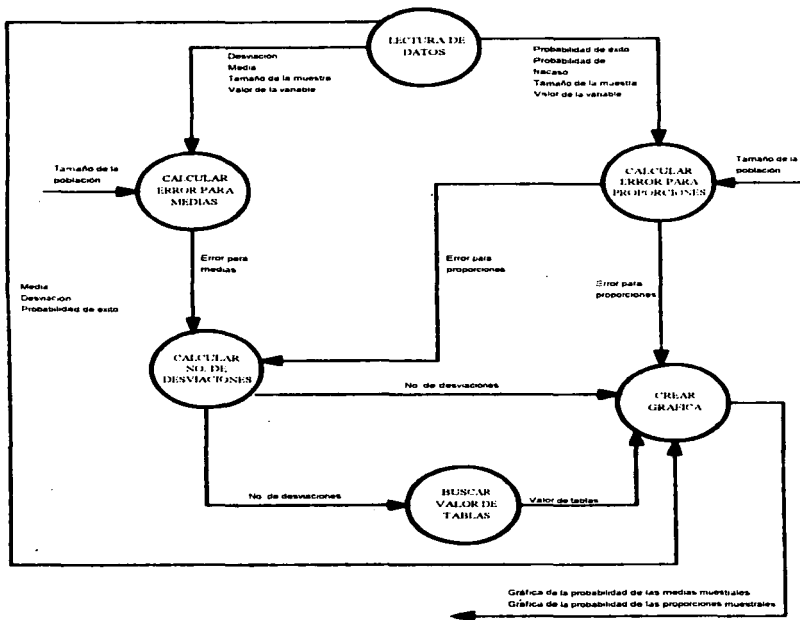
A continuación se muestra los diagramas de flujo de datos y tablas de decisión de algunos de los procesos a implementar.

3.3.1 DIAGRAMAS DE FLUJO DE DATOS

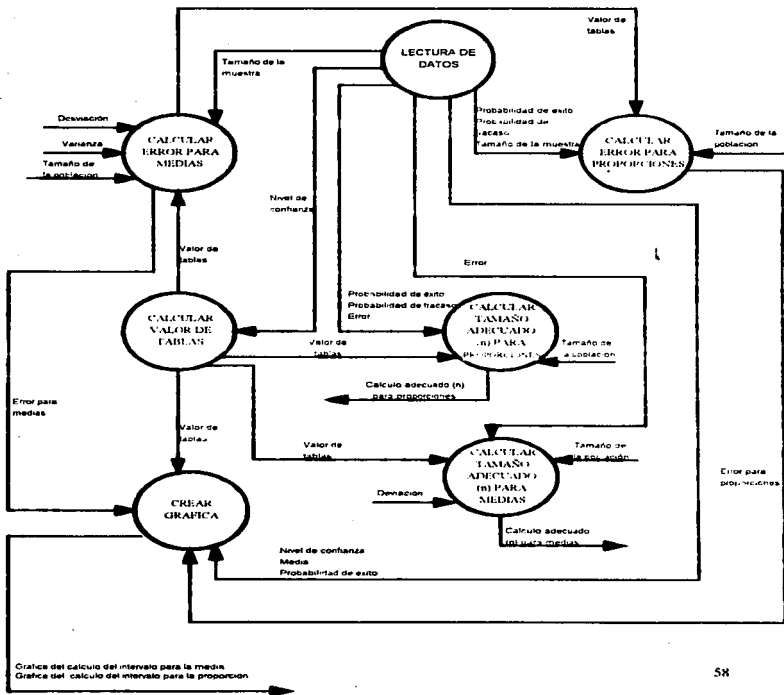
Módulo: Muestreo



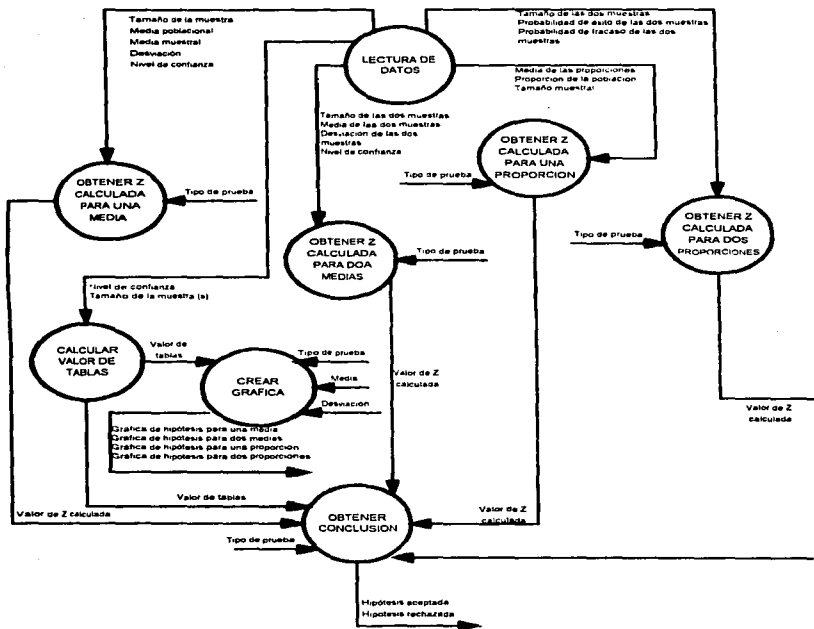
Módulo: Distribución para medias y proporciones



Módulo: Estimación



Módulo: Prueba de hipótesis



3.3.2 TABLAS DE DECISION

Sistema: Sistema Estadístico

Usuario:

Módulo: Distribución

Tabla de decisiones								
CONDICIONES	REGLAS							
	1	2	3	4	5	6	7	8
Poblacion finita	S	S	S	N	N	N	N	N
Número de desviaciones ± 3	S	S	N	S	S	S	S	N
Distribuida normalmente	-	-	-	S	S	S	N	N
$N > 30$	S	N	-	S	N	S	N	-
ACCIONES								
Aplicar teorema del limite central							X	
Seleccionar tipo de prueba	X	X	X	X	X	X	X	X
Leer dato mayor o menor	X	X	X	X	X	X	X	X
Calcular el número de desviaciones	X	X	X	X	X	X	X	X
Calcular error	X	X						
Calcular error infinito					X	X	X	
Buscar el valor de tablas en z	X			X	X	X	X	
Buscar el valor de tablas en t	X	X			X	X	X	
Calcular la probabilidad	X	X		X	X	X	X	
Desplegar datos, fórmula y resultados	X	X		X	X	X	X	
Desplegar mensaje de error				X				X

Sistema: Sistema Estadístico

Usuario: _____

Modulo: Estimación

Tabla de decisiones

CONDICIONES	REGLAS															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Población finita	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Desviación desconocida	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
$n \leq 30$	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Datos sueltos	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Factor mayor a 0.05	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
ACCIONES																
Leer N	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Leer los datos sueltos	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Calcular la desviación, varianza y media	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Leer el valor de la desviación y media	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Leer varianza y media	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Leer valor de significancia	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Buscar el valor en la tabla z	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Buscar el valor en la tabla t	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Calcular el factor n/N	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Calcular el error aplicando el factor de corrección	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Calcular error	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Calcular el intervalo	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Desplegar datos, fórmula y gráfica	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Sistema: Sistema Estadístico

Usuario:

Modulo: Prueba de hipótesis

Tabla de decisiones						
CONDICIONES	REGLAS					
	1	2	3	4	5	6
Muestras grandes	S	S	S	S	N	N
Muestras dependientes	S	S	S	S	S	S
Datos sueltos	S	S	N	Z	N	N
$n1 \text{ o } n2 > 30$	S	N	S	Z	N	N
ACCIONES						
Leer el número de datos	X	X				
Leer el tamaño de la muestra (n)					X	X
Leer los datos de cada muestra					X	X
Calcular las diferencias y las diferencias cuadradas					X	X
Leer la suma de las diferencias y diferencias cuadradas						X
Leer los datos	X	X				
Calcular media y desviación estándar	X	X				
Leer los datos de las dos muestras			X	X		
Leer el nivel de confianza	X	X	X	X	X	X
Seleccionar tipo de prueba	X	X	X	X	X	X
Calcular el valor de t_{α}	X	X	X	X	X	X
Buscar el valor en tabla Z	X	X	X	X	X	X
Buscar el valor en tabla t	X	X	X	X	X	X
Calcular Z_c o t_c	X	X	X	X	X	X
Desplegar los datos, hipótesis y gráfica	X	X	X	X	X	X
Evaluar la hipótesis y mostrar conclusión	X	X	X	X	X	X

CAPITULO IV

**DISEÑO DEL SISTEMA DE
ESTADÍSTICA PROPUESTO**

4.1 DISEÑO DEL SISTEMA DE ESTADÍSTICA PROPUESTO

A partir de la especificación de los flujos de datos resultante de la fase de análisis, la cual nos muestra en forma general los procesos que habrán de ser implementados, ahora detallaremos cada uno de ellos. En el diseño lógico se da una descripción general de los módulos que conforman el sistema y en el diseño físico se hace una definición clara de los programas e interfaz, a través del empleo de herramientas como son: el diagrama jerárquico funcional, la especificación de programas y el diseño de pantallas, las cuales facilitan la codificación y prueba del sistema.

4.2 DISEÑO LÓGICO

El sistema de Estadística que se propone, está dividido en dos partes: la primera, titulada Estadística I, la cual abarca procedimientos comprendidos en la Estadística Descriptiva, entre ellos: la organización de los datos en una tabla de distribución de frecuencias, la representación gráfica de las observaciones, el cálculo de las medidas de tendencia central y medidas de dispersión, además de la obtención de las probabilidades de las distribuciones Normal, Binomial y Poisson.

La segunda parte, titulada Estadística II, abarca métodos de Inferencia Estadística como son: muestreo con remplazo y sin remplazo, distribución de la probabilidad, estimación del intervalo de confianza y del tamaño adecuado de la muestra (n), prueba de hipótesis, análisis de la varianza, regresión y correlación simple.

4.3. DISEÑO FÍSICO

4.3.1 DIAGRAMA JERÁRQUICO FUNCIONAL

El diagrama jerárquico funcional muestra la forma en que el problema es dividido en módulos para su implementación; en este caso, los módulos están delimitados por los procedimientos estadísticos.

DIAGRAMA JERARQUICO FUNCIONAL

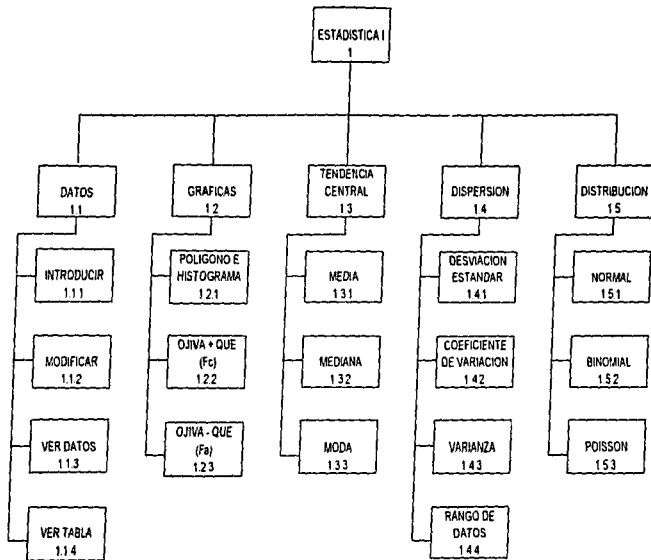
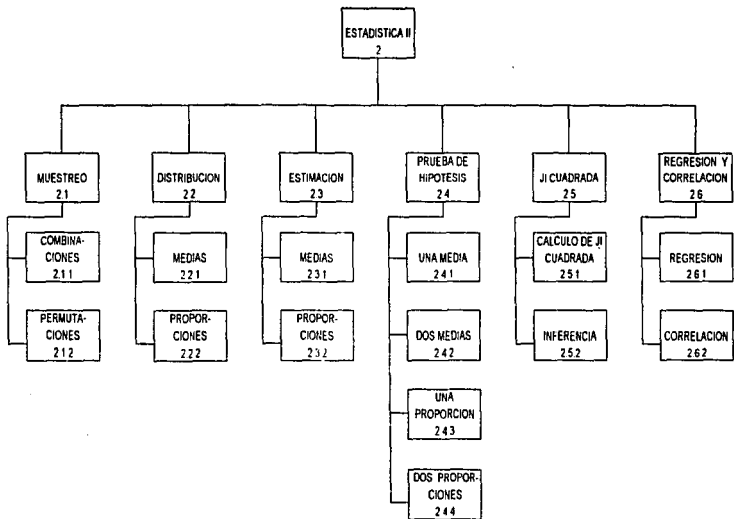


DIAGRAMA JERARQUICO FUNCIONAL



4.3.2 DISEÑO DE LA INTERFAZ

El diseño de la interfaz facilita al desarrollador del sistema la programación del mismo; mediante el diseño de las pantallas se muestra los elementos y contenidos de cada una de ellas, tales como: título, etiquetas, mensajes, espacios para la lectura de datos, y en algunos casos una plantilla que establece el tipo de datos que el usuario deberá introducir al ejecutar el programa. El uso de esta herramienta evita perder el tiempo y minimiza el número de errores en la codificación del sistema. También se determina y especifica los dispositivos que serán utilizados para dar entrada a los datos.

En el sistema que proponemos, la comunicación sistema-usuario es a través de menús, submenús, teclas de función, líneas de estado y de mensajes, etc., donde el usuario selecciona a través de las flechas de dirección el procedimiento de su interés, e introduce los datos que éste le va solicitando en un espacio delimitado por una celda.

A continuación se muestra algunas de las pantallas utilizadas en la implantación del sistema.

DISEÑO DE PANTALLA

No. DE PANTALLA:

3 MENU PRINCIPAL DE ESTADISTICA I

PROGRAMA:

SISTEMA ESTADISTICO

The screen design is a grid-based menu for a statistics system. The grid is approximately 28 columns wide and 24 rows high. The text is arranged as follows:

- Row 1:** A long line of small, illegible characters, likely a header or separator.
- Row 2:** "UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO" centered.
- Row 3:** "ESTADISTICA I" centered.
- Row 4:** A blank row.
- Row 5:** A blank row.
- Row 6:** A blank row.
- Row 7:** A blank row.
- Row 8:** Five main menu items: "DATOS", "GRAFICAS", "TECU CENT", "DISPERSION", and "DISTRIBUCION", spaced across the width.
- Row 9:** A blank row.
- Row 10:** A sub-menu for "DATOS" with options: "INTRODUCIR", "MODIFICAR", "VER DATOS", and "VER TABLA".
- Row 11:** A sub-menu for "TECU CENT" with options: "MEDIA", "MEDIANA", and "MODA".
- Row 12:** A sub-menu for "DISTRIBUCION" with options: "NORMAL", "BINOMIAL", and "POISSON".
- Row 13:** A blank row.
- Row 14:** A blank row.
- Row 15:** A blank row.
- Row 16:** A blank row.
- Row 17:** A blank row.
- Row 18:** A blank row.
- Row 19:** A blank row.
- Row 20:** A blank row.
- Row 21:** A blank row.
- Row 22:** A blank row.
- Row 23:** A blank row.
- Row 24:** A footer line with the text "Selecciones ESC. Estadística P. Ayuda" on the right side.

DISEÑO DE PANTALLA

No. DE PANTALLA: 4 MENÚ PRINCIPAL DE ESTADISTICA I

PROGRAMA: SISTEMA ESTADISTICO

1	UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO				21	
2	ESTADISTICA I				22	
3					23	
4					24	
5					25	
6					26	
7					27	
8	DATOS	GRAFICAS	TEND. CENT.	DISPERSION	DISTRIBUCION	28
9						29
10	POLIGONO E HISTOGRAMA			DESV. ESTANDAR		30
11	OJIVA MAS QUE (F _x)			COEFICIENTE DE VAR.		31
12	OJIVA MENOS QUE (F _x)			VARIANZA		32
13				RANGO DE DATOS		33
14						34
15						35
16						36
17						37
18						38
19						39
20						40
21						41
22						42
23						43
24						44
25						45
26						46
27						47
28						48
29						49
30						50
31						51
32						52
33						53
34						54
35						55
36						56
37						57
38						58
39						59
40						60
41						61
42						62
43						63
44						64
45						65
46						66
47						67
48						68
49						69
50						70
51						71
52						72
53						73
54						74
55						75
56						76
57						77
58						78
59						79
60						80
61						81
62						82
63						83
64						84
65						85
66						86
67						87
68						88
69						89
70						90
71						91
72						92
73						93
74						94
75						95
76						96
77						97
78						98
79						99
80						100

DISEÑO DE PANTALLA

No. DE PANTALLA: 7 MODIFICACION DE LOS DATOS

PROGRAMA: SISTEMA ESTADISTICO

1	MODIFICACION DE DATOS																									27
2																										28
3	ALGUN DATO																									29
4	INCREMENTAR X																									30
5	DECREMENTAR X																									31
6	TIPO DE DATOS																									32
7																										33
8																										34
9																										35
10																										36
11																										37
12																										38
13																										39
14																										40
15																										41
16																										42
17																										43
18																										44
19																										45
20																										46
21																										47
22																										48
23																										49
24																										50
25																										51
26																										52
27	Nuevo valor de x																									53
28																										54
29																										55
30																										56
31																										57
32																										58
33																										59
34																										60
35																										61
36																										62
37																										63
38																										64
39																										65
40																										66
41																										67
42																										68
43																										69
44																										70
45																										71
46																										72
47																										73
48																										74
49																										75
50																										76
51																										77
52																										78
53																										79
54																										80
55																										81
56																										82
57																										83
58																										84
59																										85
60																										86
61																										87
62																										88
63																										89
64																										90
65																										91
66																										92
67																										93
68																										94
69																										95
70																										96
71																										97
72																										98
73																										99
74																										100

DISEÑO DE PANTALLA

No. DE PANTALLA: _____

20 PRUEBA DE HIPOTESIS PARA DOS MEDIAS

PROGRAMA: _____

SISTEMA ESTADISTICO

1	HIPOTESIS PARA DOS MEDIAS		
2			
3			
4	Datos de la primera muestra:		
5			
6	Tamaño de la muestra (n)	10	
7			
8	Media muestral (x)	10.0	
9			
10	Desviación (S)	1.0	
11			
12			
13	Datos de la segunda muestra:		
14			
15	Tamaño de la muestra (n)	10	
16			
17	Media muestral (x)	10.0	
18			
19	Desviación (S)	1.0	
20			
21			
22			
23			
24			
25			
26			
27			
28			
29			
30	Conti/Menu/ESC Regresar		

4.2.3 ESPECIFICACION DE PROGRAMAS

En los diagramas de flujo de datos y jerárquico funcional, se menciona los procesos que componen el sistema de Estadística, pero aún no se ha dado una especificación detallada de lo que cada uno de ellos debe realizar para cumplir con los fines establecidos.

La especificación de programas permite describir lógicamente las acciones a ejecutar cuando la evaluación de una condición es verdadera, indicar la entrada de los datos necesarios, así como las salidas resultado del procesamiento de los datos. Estas especificaciones son utilizadas para la programación, depuración, prueba, evaluación, documentación y mantenimiento del sistema.

A continuación se muestra la especificación detallada de algunos de los procesos más representativos que se encuentran en el sistema.

Sistema: Sistema Estadístico
 Usuario:

Modulo: 2.1 Muestreo

Especificación de programas

<p>Opcion</p>	<p>2.1 Muestreo Presentar en pantalla las opciones Con remplazo Sin remplazo</p> <p>Seleccionar una opcion de las anteriores</p>		
<p>Tamaño de la población Datos de la población</p>	<p>1 Con remplazo (Permutaciones)</p> <p>1.1 Desplegar teoremas 1.2 Leer tamaño de la población (N) 1.3 Leer los N datos 1.4 Calcular el numero de permutaciones 1.5 Generar las permutaciones 1.6 Calcular la media poblacional 1.7 Calcular la media de cada permutacion 1.8 Calcular la media - media elevada de cada permutacion 1.9 Calcular la desviacion y el error estándar 1.10 Desplegar permutaciones 1.11 Comprobar y desplegar el primer teorema 1.12 Comprobar y desplegar el segundo teorema</p>	<p>Permutaciones Comprobacion de los teoremas</p>	<p>Se realizan con un tamaño de muestra de n= 2.</p>
<p>Tamaño de la población Datos de la población Tamaño de la muestra</p>	<p>2 Sin remplazo (Combinaciones)</p> <p>1.1 Desplegar teoremas 1.2 Leer tamaño de la población (N) 1.3 Leer los N datos 1.4 Leer tamaño de la muestra (n) 1.5 Calcular el numero de combinaciones 1.6 Generar las combinaciones 1.7 Calcular la media poblacional 1.8 Calcular la media de cada combinacion 1.9 Calcular la media - media elevada de cada combinacion 1.10 Calcular la desviacion y el error estándar 1.11 Desplegar combinaciones 1.12 Comprobar y desplegar el primer teorema 1.13 Comprobar y desplegar el segundo teorema</p>	<p>Combinaciones Comprobacion de los teoremas</p>	

Sistema: Sistema Estadístico

Usuario: _____

Modulo: 2.2 Distribución

Especificación de programas				
Entradas	Procesos	Salidas	Observaciones	
Datos de la muestra	<p>2.2 Distribución de la probabilidad</p> <p>Presentar en pantalla las opciones.</p> <p>En medias</p> <p>En proporciones</p> <p>1. Distribución en medias</p> <p>1.1 Leer datos (media, desviación, tamaño de la muestra)</p> <p>Si son poblaciones finitas</p> <p>1.2 Leer el tamaño de la población</p> <p>1.3 Seleccionar el tipo de prueba:</p> <ul style="list-style-type: none"> - Inferior a un valor - Superior a un valor - De la media a un valor - Entre dos valores <p>1.4 Leer el dato mayor o menor que la media</p> <p>1.5 Calcular el número de desviaciones</p> <p>Si el número de desviaciones es ≤ 3</p> <p>1.6 Calcular el error</p> <p>1.7 Buscar el valor de tablas</p> <p>1.8 Calcular la probabilidad</p> <p>1.9 Desplegar los datos, fórmula y resultados</p> <p>1.10 Graficar</p> <p>En caso contrario</p> <p>1.6 Desplegar mensaje de error</p> <p>Si son poblaciones infinitas</p> <p>Si está distribuida normalmente</p> <p>1.2 Seleccionar el tipo de prueba.</p> <ul style="list-style-type: none"> - Inferior a un valor - Superior a un valor - De la media a un valor - Entre dos valores <p>1.3 Leer el dato mayor o menor que la media</p> <p>1.4 Calcular el error infinito</p> <p>1.5 Calcular el número de desviaciones</p>	<p>Grafica de la distribución de la probabilidad en medias finitas</p> <p>Mensaje de error</p>		
Tamaño de N Opción				
Valor menor o mayor a la media				
Opción				
Valor menor o mayor a la media				

Capítulo IV: Diseño del sistema de Estadística propuesto

Sistema: Sistema Estadístico

Usuario:

Modulo: 2.2 Distribucion

Especificación de programas			
Entradas	Procesos	Salidas	Observaciones
Opción Dato mayor o menor que la media	<p>Si el número de desviaciones es ≤ 3</p> <p>1.6 Calcular el error</p> <p>1.7 Buscar el valor de tablas</p> <p>1.8 Calcular la probabilidad</p> <p>1.9 Desplegar los datos, fórmula y resultados</p> <p>1.10 Graficar</p> <p>En caso contrario</p> <p>1.6 Enviar mensaje de error</p> <p>Si no está distribuida normalmente</p> <p>Si $n < 30$</p> <p>1.2 Aplicar el teorema del límite central</p> <p>1.3 Seleccionar el tipo de prueba</p> <ul style="list-style-type: none"> - Inferior a un valor - Superior a un valor - De la media a un valor - Entre dos valores <p>1.4 Leer el dato mayor o menor que la media</p> <p>1.5 Calcular el error infinito</p> <p>1.6 Calcular el número de desviaciones</p> <p>Si el número de desviaciones es ≤ 3</p> <p>1.7 Calcular el error</p> <p>1.8 Buscar el valor de tablas</p> <p>1.9 Calcular la probabilidad</p> <p>1.10 Desplegar los datos, fórmula y resultados</p> <p>1.11 Graficar</p> <p>En caso contrario</p> <p>1.7 Enviar mensaje de error</p> <p>En caso contrario</p> <p>1.2 No se pued. hacer nada</p>	<p>Grafica de la distribución de la probabilidad en medias infinitas, no distribuida normalmente</p> <p>Mensaje de error</p> <p>Grafica de la distribución de la probabilidad en medias infinitas, distribuida normalmente</p> <p>Mensaje de error</p>	

Sistema: Sistema Estadístico

Usuario:

Módulo: 2.2 Distribución

Especificación de programas			
Entradas	Procesos	Salidas	Observaciones
Datos de la muestra	<p>2. Distribución en proporciones</p> <p>2.1 Leer datos (probabilidad de éxito, probabilidad de fracaso, tamaño de la muestra)</p> <p>2.2 Determinar si es finita o infinita la muestra Si es finita</p> <p>2.3 Leer el tamaño de la población</p> <p>2.4 Seleccionar el tipo de prueba</p> <ul style="list-style-type: none"> - Proporción ... - Proporción ... - Proporción ... - Proporción ... - Proporción entre dos valores <p>2.5 Leer el dato mayor o menor que la media</p> <p>2.6 Calcular el factor de corrección Si el factor es < 0.05</p> <p>2.7 Aplicación del error infinito En caso contrario</p> <p>2.7 Aplicación del error finito Si el número de desviaciones es ≤ 3</p> <p>2.8 Calcular el error</p> <p>2.9 Buscar el valor de tablas</p> <p>2.10 Calcular la probabilidad</p> <p>2.11 Desplegar los datos, fórmula y resultados</p> <p>2.12 Graficar</p> <p>En caso contrario</p> <p>2.8 Desplegar mensaje de error</p>	<p>Grafica de la distribución de la probabilidad en proporciones finitas</p> <p>Mensaje de error</p>	
Tamaño de N Opción	<p>Si son poblaciones infinitas</p> <p>2.3 Seleccionar el tipo de prueba:</p> <ul style="list-style-type: none"> - Proporción ... - Proporción ... - Proporción ... - Proporción ... - Proporción entre dos valores 		
Dato mayor o menor	<p>2.4 Leer el dato mayor o menor que la media</p> <p>2.5 Aplicación del error infinito</p>		
Opción			
Dato mayor o menor			

Sistema: Sistema Estadístico

Usuario: _____

Modulo: 2.2 Distribución

Especificación de programas

Entradas	Procesos	Salidas	Observaciones
	<p>Si el número de desviaciones es ≥ 3</p> <ul style="list-style-type: none"> 2.6 Calcular el error 2.7 Buscar el valor de tablas 2.8 Calcular la probabilidad 2.9 Desplegar los datos, fórmula y resultados 2.10 Graficar <p>En caso contrario</p> <ul style="list-style-type: none"> 2.6 Desplegar mensaje de error 	<p>Gráficas de la distribución de la probabilidad en proporciones infinitas</p> <p>Mensaje de error</p>	

Sistema Sistema Estadístico
 Usuario _____

Modulo 2.3 Estimacion

Especificación de programas			
Entradas	Procesos	Salidas	Observaciones
	<p>2.3 Estimacion</p> <p>Presentar en pantalla las opciones.</p> <p>Calculo del intervalo en medias</p> <p>Tamaño adecuado de n en medias</p> <p>Calculo del intervalo en proporciones</p> <p>Tamaño adecuado de n en proporciones</p>		
Opcion	<p>Seleccionar una de las opciones anteriores</p>		
Tamaño de la muestra	<p>1. Calculo del intervalo en medias</p> <p>1.1 Leer n y N</p> <p>Si la poblacion es finita</p> <p>1.2 Leer N</p> <p>Si la desviacion es desconocida</p> <p>Si $n > 30$</p> <p>Si son datos sueltos</p>		
Tamaño de la poblacion	<p>1.3 Leer los datos sueltos</p> <p>1.4 Calcular la desviacion, varianza y media</p> <p>En caso contrario, si son datos agrupados</p> <p>1.3 Leer la varianza y media</p>		
Datos sueltos	<p>1.5 Leer el nivel de significancia</p> <p>1.6 Buscar el valor en la tabla t de student</p> <p>En caso contrario, $n > 30$</p> <p>1.3 Leer la varianza y media</p> <p>1.5 Leer el nivel de significancia</p> <p>1.6 Buscar el valor en la tabla z</p> <p>Si la poblacion es finita</p> <p>1.7 Calcular el factor n/N</p> <p>Si factor $> .05$ y poblacion finita</p> <p>1.8 Calcular el error aplicando el factor de correccion</p> <p>En caso contrario</p> <p>1.8 Calcular el error</p> <p>1.9 Calcular el intervalo</p> <p>1.10 Graficar</p>		
Datos			
Nivel de significancia			
Datos			
Nivel de significancia			
		<p>Grafica del intervalo de confianza en medias finitas e infinitas cuando la desviacion es desconocida.</p>	

Sistema: Sistema Estadístico

Usuario:

Modulo: 2.3 Estimación

Especificación de programas			
Entradas	Procesos	Salidas	Observaciones
<p>Media</p> <p>Nivel de significancia</p>	<p>En caso contrario, si la desviación es conocida</p> <p>1.3 Leer el valor de la desviación y media</p> <p>1.4 Leer el nivel de significancia</p> <p>1.5 Buscar el valor de tablas de Z</p> <p>1.6 Calcular el factor n/N</p> <p>Si factor = 0.05 o población infinita</p> <p>1.7 Calcular el error</p> <p>En caso contrario</p> <p>1.7 Calcular el error, aplicando el factor de corrección</p> <p>1.8 Calcular el intervalo en medias</p> <p>1.9 Graficar</p>	<p>Gráfica del intervalo en medias, cuando la desviación es conocida, para medias finitas e infinitas</p>	
<p>Error y desviación</p>	<p>2 Cálculo del tamaño adecuado de n en medias</p> <p>2.1 Leer el error y la desviación</p> <p>Si la población es finita</p> <p>2.2 Leer el tamaño de la población (N)</p> <p>2.3 Leer el nivel de significancia</p> <p>2.4 Buscar el valor en la tabla z</p> <p>Si la población es infinita</p> <p>2.5 Cálculo de n, aplicando fórmula infinita</p> <p>En caso contrario</p> <p>2.5 Cálculo de n, aplicando fórmula finita</p> <p>2.6 Desplegar resultados</p>	<p>Tamaño adecuado de n en medias</p>	
<p>Probabilidad de éxito y n</p>	<p>3 Cálculo del intervalo en proporciones</p> <p>3.1 Leer probabilidad de éxito (p), y n</p> <p>3.2 Calcular probabilidad de fracaso (q)</p> <p>Si la población es finita</p> <p>3.3 Leer el tamaño de la población (N)</p> <p>3.4 Calcular el factor n/N</p> <p>3.5 Leer el nivel de significancia</p> <p>Si n > 30</p> <p>3.6 Buscar el valor de tablas en la tabla z</p> <p>En caso contrario</p> <p>3.6 Buscar el valor de tablas en la tabla t</p>		
<p>Tamaño de la población</p> <p>Nivel de significancia</p>			

Capítulo IV: Diseño del sistema de Estadística propuesto

Sistema: **Sistema Estadístico**

Usuario: _____

Modulo: **2.3 Estimación**

Especificación de programas

Entradas	Procesos	Salidas	Observaciones
<p>Error y probabilidad de éxito Tamaño de la población Nivel de significancia</p>	<p>Si factor = 0.05 o la población es infinita 3.7 Calcular el error para poblaciones infinitas En caso contrario 3.7 Calcular el error aplicando el factor de corrección 3.8 Calcular el intervalo de confianza en proporciones 3.9 Graficar</p> <p>4. Cálculo del tamaño adecuado de n en proporciones: 4.1 Leer el error y la probabilidad de éxito (p) Si la población es finita 4.2 Leer el tamaño de la población (N) 4.3 Leer el nivel de significancia 4.4 Buscar el valor de tabla de z Si la población es infinita 4.5 Cálculo de n, aplicando fórmula infinita En caso contrario 4.5 Cálculo de n, aplicando fórmula finita 4.6 Desplegar resultados</p>	<p>Cálculo del intervalo para la proporción finita e infinita</p> <p>Tamaño adecuado de n en proporciones</p>	

Sistema: Sistema Estadístico

Usuario:

Módulo: 2.4 Prueba de hipótesis

Especificación de programas

Entradas	Procesos	Salidas	Observaciones
	<p>2.4 Prueba de hipótesis</p> <p>Presentar las opciones:</p> <ul style="list-style-type: none"> Para una media Para dos medias Para una proporción Para dos proporciones <p>1 Prueba de hipótesis para una media</p> <ul style="list-style-type: none"> 1.1 Preguntar si está distribuida la media o no 1.2 Leer datos (tamaño de la muestra, media, media muestral y desviación) 1.3 Leer el nivel de confianza 1.4 Seleccionar tipo de prueba (unilateral por la derecha, por la izquierda o bilateral) 1.5 Calcular el valor de Z_c Si $n > 30$ y no está distribuida normalmente <ul style="list-style-type: none"> 1.6 Buscar el valor de confianza en la tabla t con $n-1$ grados de libertad En caso contrario: <ul style="list-style-type: none"> 1.6 Buscar el valor de confianza en la tabla z 1.7 Desplegar datos, hipótesis y fórmulas 1.8 Displayar la grafica 1.9 Evaluar el resultado y determinar la conclusión <p>2 Prueba de hipótesis para dos muestras</p> <p>Si son muestras grandes e independientes</p> <p>Si son datos sueltos</p> <ul style="list-style-type: none"> 2.1 Leer el número de datos 2.2 Leer los datos 2.3 Calcular media y desviación estándar <p>En caso contrario</p> <ul style="list-style-type: none"> 2.1 Leer los datos de las dos muestras 2.4 Leer el nivel de confianza 2.5 Seleccionar tipo de prueba (unilateral por la derecha, por la izquierda o bilateral) 2.6 Calcular el valor de alfa Si $n1 > n2 > 30$ <ul style="list-style-type: none"> 2.7 Buscar el valor de tablas en la tabla z En caso contrario: <ul style="list-style-type: none"> 2.7 Buscar el valor de tablas en la tabla t con $n1 + n2 - 2$ grados de libertad 		
Datos			
Nivel de confianza			
Opción			
Numero de datos			
Datos			
Datos			
Nivel de confianza			
Opción			
		Grafica de la prueba de hipótesis para una media	

Capítulo IV: Diseño del sistema de Estadística propuesto

Sistema: Sistema Estadístico

Usuario:

Modulo: 2.4 Prueba de hipótesis

Especificación de programas			
Entradas	Procesos	Salidas	Observaciones
	2.8 Calcular Z_0 o t_0 2.9 Desplegar los datos, hipótesis y fórmulas 2.10 Graficar 2.11 Evaluar las hipótesis y mostrar la conclusión	Gráfica de la prueba de hipótesis para medias grandes o independientes	
Tamaño de las muestras	En caso de ser muestras dependientes 2.1 Leer n Si son datos sueltos		
Datos de cada muestra	2.2 Leer los datos de cada muestra 2.3 Calcular las diferencias y las diferencias cuadradas		
Datos de las muestras	En caso contrario 2.2 Leer la sumatoria de los diferencias y diferencias cuadradas		
Nivel de confianza Opcion	2.4 Leer en nivel de confianza 2.5 Seleccionar tipo de prueba (unilateral por la derecha, por la izquierda o bilateral) 2.6 Calcular el valor de alfa 2.7 Buscar el valor de tablas en la tabla t con $n-1$ grados de libertad		
	2.8 Calcular t_0 2.9 Desplegar los datos, hipótesis y fórmulas 2.10 Graficar 2.11 Evaluar las hipótesis y mostrar la conclusión	Gráfica de la prueba de hipótesis para dos medias dependientes	
Datos	3. Prueba de hipótesis para una proporción 3.1 Leer datos (tamaño de la muestra, proporción de la población y media de la proporción)		
Nivel de confianza Opcion	3.2 Leer el valor de confianza 3.3 Seleccionar tipo de prueba (unilateral por la derecha, por la izquierda o bilateral) 3.4 Calcular el valor de alfa, dependiendo del tipo de prueba Si $n > 30$ 3.5 Buscar el valor de tablas en la tabla t con $n-1$ grados de libertad En caso contrario 3.5 Buscar el valor de tablas en la tabla z 3.6 Calcular Z_0 o t_0		

4.4 IMPLEMENTACION DEL SISTEMA

Una vez especificados los procesos y diseñada la interfaz del sistema, el siguiente paso es la programación y realización de pruebas al sistema, que es el resultado de esta fase y debe cumplir con los requerimientos detallados en el análisis.

El lenguaje de programación utilizado para la codificación del sistema fue C++, debido a que cuenta con las estructuras de control necesarias para la programación estructurada, además de permitir la manipulación de la pantalla para la graficación y desplegado de las fórmulas. También se utilizó la programación orientada a objetos, para facilitar la codificación de las gráficas, tablas y menús del sistema.



CONCLUSIONS

CONCLUSIONES

La Estadística es importante porque nos permite analizar el comportamiento de ciertas características de una población, y así disminuir el grado de incertidumbre en la toma de decisiones, al inferir con los resultados obtenidos en el estudio de una muestra. Prueba de esta importancia es que se encuentra inmersa en la mayoría de las actividades humanas, aplicándola en muchas situaciones, aun sin darnos cuenta.

Por lo anterior, es imprescindible el aprendizaje de la misma por los estudiantes, si no a un nivel experto, sí al nivel que se requiere en su ámbito profesional.

Por otra parte, la Informática ha tenido gran auge debido al sinnúmero de aplicaciones que se ha implementado para apoyar en tareas complejas y donde se requiere rapidez y precisión, por ejemplo: la realización del análisis estadístico, el diseño asistido por computadora, sistemas de consultas especializadas, sistemas integrales de información, entre otros, mediante la aplicación de sistemas computarizados. La utilización de tales sistemas proporciona muchas ventajas al usuario, entre las que se encuentran: exactitud, oportunidad, confiabilidad, eficiencia y presentación.

Para el desarrollo de sistemas es necesario tener una clara concepción de la problemática que se presenta, dado que es la base para la selección de la metodología que guiará el desarrollo del proyecto.

En cuanto a la selección de la metodología adecuada, dependerá de la consideración de factores tanto internos como externos. Entre los primeros tenemos: la orientación y alcance del sistema requerido, los objetivos organizacionales y planes de crecimiento, el volumen de información, el número de transacciones, los recursos económicos, técnicos y humanos disponibles, etc. Entre los factores externos encontramos: la situación económica, los avances tecnológicos, la disponibilidad de personal capacitado, entre otros.

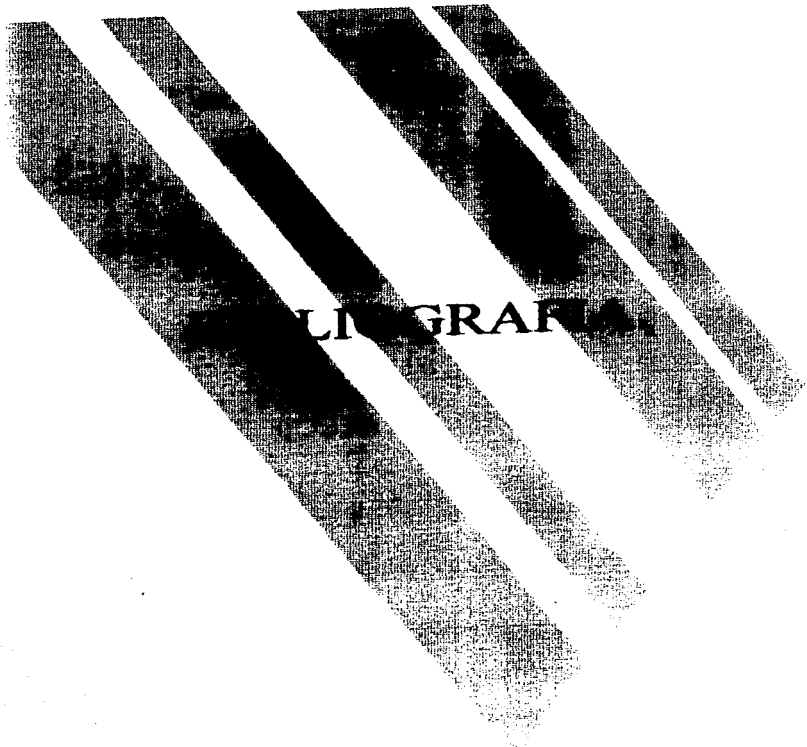
El uso de una metodología para el desarrollo de sistemas, beneficia tanto a los desarrolladores como a los interesados, entre ellos los futuros usuarios, ya que al planear adecuadamente el proyecto y tener el control del mismo, se evita esfuerzos infructuosos, gastos innecesarios y pérdida de tiempo al volver a diseñar el sistema cuando se detecta alguna falla en una etapa de desarrollo avanzado.

En general, el desarrollo de sistemas se divide en dos grandes fases, que son: Análisis y Diseño de Sistemas. La importancia de la primera, radica en la recopilación de los datos necesarios para identificar las necesidades existentes y determinar las especificaciones para el nuevo sistema, lo que permitirá crear varias alternativas que cumplan con dichas especificaciones.

El análisis se apoya en el uso de herramientas gráficas, las cuales ayudan en gran medida al buen entendimiento del sistema; cada una de ellas se concentra en una parte específica del mismo.

En el diseño es necesario crear un modelo acorde al tipo de usuario que lo operará, al equipo con que se cuenta, a la capacidad de respuesta que se requiere, al sistema de seguridad que debe existir, entre otros factores; asimismo, deberá basarse en la determinación de requerimientos resultante del análisis.

Por último, agregaremos que en el desarrollo del sistema presentado en este trabajo, fue utilizada la metodología del Ciclo de Vida Estructurado, por considerarla la más adecuada al problema que se presentó, el cual se prestó para ser dividido en módulos y desarrollar éstos de manera independiente, llevándolos a un mismo tiempo en diversas actividades sin que esto afectara al sistema global. No hay que olvidar que es precisamente esta metodología la que nos permite la división modular, y llevar diversas actividades paralelamente.



LITOGRAFIA

BIBLIOGRAFIA

Arnold, Robert R.; Sistema Moderno de Procesamiento de Datos. Ed. Limusa. Octava reimpresión. México, 1984.

Bancroft, Huldán; Introducción a la Bioestadística. Ed. Universitaria de Buenos Aires. Primera edición. Argentina, 1960. 246 p.

Blalock, Hubert M.; Estadística Social. Ed. Fondo de Cultura Económica. México, 1966. 509 p.

Don Yeates, Alan Daniels; Análisis Básico de Sistemas. Ed. Paraninfo. España, 1991.

García Pérez, Andrés; Elementos del Método Estadístico. Ed. UNAM. Cuarta edición. México, 1966. 503 p.

Hayslett, H. T.; Estadística Simplificada. Ed. Compañía General de Ediciones. Primera edición. México, 1973. 208 p.

Hoel, Paul G.; Jessen, Raymond J.; Estadística Básica para Negocios y Economía. Ed. Ceesa. México, 1983. 598 p.

Infante Gil, Saíd; Zárate de Lara y P., Guillermo; Métodos Estadísticos. Ed. Trillas. Primera edición. México, 1984. 643 p.

Kendall, E. Kenneth; Kendall, E. July; Análisis y Diseño de Sistemas. Ed. Prentice Hall Hispanoamericana. Primera edición. México, 1991. 881 p.

Levin, Richard Y.; Estadística para Administradores. Ed. Prentice Hall Hispanoamericana. Segunda edición. México, 1988. 940 p.

Lucas, Henry C.; Sistemas de Información. Ed. Paraninfo. Segunda edición. Madrid.

Márquez Vite, Juan Manuel; Sistema de Información por Computadora. Ed. Trillas. Segunda edición. México, 1990. 218 p.

Mendenhall, William; Scheaffer, Richard L.; Wackerly, Dennis D.; Estadística Matemática con Aplicaciones. Ed. Iberoamérica. Tercera edición. México, 1986. 751 p.

Neter, John; Fundamentos de Estadística y Economía. Ed. Ceesa. México, 1962. 951 p.

Ojguín Quiñones, Fernando; Estadística Descriptiva Aplicada a las Ciencias Sociales. Ed. UNAM. México, 1984. 451 p.

Scheaffer, Richard L.; McClave, James T.; Probabilidad y Estadística para Ingeniería. Ed. Iberoamérica. Primera edición. México, 1993. 643 p.

Schott, Sigmund; Estadística. Ed. Labor. Barcelona, 1928. 206 p.

Scott, George M.; Principios de Sistemas de Información. Ed. McGraw-Hill. México, 1990. 646 p.

Senn, A. James; Análisis y Diseño de Sistemas de Información. Ed. McGraw-Hill. Segunda edición. México, 1994. 942 p.

Bibliografía

Serrano Lomelin, Jesús A.; Vargas Ajuria, Ivonne; **Introducción al Manejo del Paquete Estadístico Statgraphics**. Servicios Editoriales de la Facultad de Ciencias UNAM. Primera edición. México, 1992. 47 p.

Steel, Robert G.; Torne, James H.; **Bioestadística: Principios y Procedimientos**. Ed. McGraw-Hill. Segunda edición. México, 1985. 622 p.

Villameva, Julio E.; **Computadoras y Procesamiento de Datos**. Serie General de la Organización de los Estados Americanos, Washington, D. C., Banco Mundial, 1987.

William, F.; **Razonamiento Estadístico**. Ed. Interamericana. Segunda edición. México, 1982. 189 p.

Willoughby, Atephen S.; **Probabilidad y Estadística**. Ed. Publicaciones Cultura. Primera edición. México, 1978. 215 p.

Yourdon, Edward; **Análisis Estructurado Moderno**. Ed. Prentice Hall Hispanoamericana. Primera edición. México, 1993. 735 p.

Young, Robert K.; Veldman, Donald J.; **Introducción a la Estadística Aplicada a las Ciencias de la Conducta**. Ed. Trillas. México, 1968. 453 p.



ANEXO A
MANUAL DE OPERACION

23

MANUAL DE OPERACION

CONTENIDO

<i>Introducción</i>	2
<i>Configuración necesaria</i>	2
<i>Requerimientos para su funcionamiento</i>	3
<i>Inicio del sistema</i>	3
<i>Elementos de una pantalla</i>	3
<i>Teclas de función</i>	4
<i>Estadística I</i>	5
<i>Datos</i>	5
<i>Gráficas</i>	9
<i>Medidas de tendencia central</i>	10
<i>Medidas de dispersión</i>	10
<i>Distribución</i>	11
<i>Estadística II</i>	13
<i>Muestreo</i>	13
<i>Distribución</i>	14
<i>Estimación</i>	15
<i>Hipótesis</i>	16
<i>Ji cuadrada</i>	19
<i>Regresión</i>	20

MANUAL DE OPERACION

INTRODUCCION

Bienvenido al sistema de Estadística, esperamos que éste resulte una herramienta para todas aquellas personas involucradas de alguna manera con los métodos estadísticos. Este sistema está diseñado para ayudarle a la manipulación de los datos de manera eficaz, fácil y sencilla, logrando con ello la obtención de resultados rápidos y confiables.

Si cuenta con conocimientos básicos de Estadística, no le será difícil el manejo del sistema, además de que la sencilla interfaz con el usuario logra llevarlo de la mano para la resolución de un problema.

Este sistema permite la solución de problemas del dominio de la Estadística Descriptiva e Inferencial. Los procedimientos que se sigue para la consecución de un resultado serán presentados en la pantalla, incluyendo la fórmula y gráfica en caso de haberla, con la opción de poder imprimir la salida; además proporciona ayuda sobre los temas principales.

CONFIGURACION NECESARIA

Para la visualización de los caracteres ASCII, es necesario que el teclado se encuentre configurado con la tabla de códigos en inglés (437). El comando MODE configura el teclado con la tabla de códigos seleccionada; este comando deberá encontrarse en el archivo AUTOEXEC.BAT. De ser necesario, incluya en el archivo las siguientes líneas:

```
MODE CON CODEPAGE PREPARE = (437), C: DOSEGA.CPI)
MODE CON CODEPAGE SELECT = 437
```

REQUERIMIENTOS PARA SU FUNCIONAMIENTO

Los requerimientos para que el sistema pueda funcionar son:

- *Una computadora 80286 o superior.*
- *Monitor VGA color o blanco y negro.*
- *Memoria de 2 MB (Megabytes).*
- *Capacidad en disco de menos de 1 MB, lo que permite su ejecución en un disco flexible de 3 1/2 de alta densidad, o en disco duro.*
- *Impresora de matriz (no indispensable)*

INICIO DEL SISTEMA

Para poder dar inicio al sistema, siga los siguientes pasos:

1. *Inserte el disquete que contiene el sistema.*
2. *Sitúese en la unidad donde se encuentre su disquete.*
3. *En la línea de comandos, teclee:*

A>: ESTADIST ←

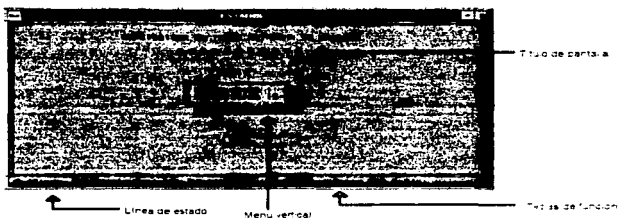
Una vez que haya hecho lo anterior, el sistema dará inicio, mostrando la presentación del mismo; pulse la tecla ENTER hasta que aparezca la Pantalla No. 1, esta pantalla es la principal del sistema, en la cual están contenidas dos opciones tituladas ESTADÍSTICA I y ESTADÍSTICA II.

ELEMENTOS DE UNA PANTALLA

Los elementos de una pantalla (ver pantalla No. 1) se mencionan a continuación:

- *Un título referente al método estadístico en el cual se encuentra.*
- *Un menú desplegable vertical y u horizontal.*

- Una línea de estado, la cual describe el proceso que se va a llevar a cabo al seleccionar la opción que está siendo resaltada en el menú, así como mensajes de error.
- Teclas de función



Pantalla No. 1: Pantalla principal que contiene las dos opciones del sistema.

TECLAS DE FUNCION

- | | |
|---------------------|--------------------------------------------------------------------------------------|
| ENTER (Seleccionar) | Da inicio a la opción del menú que se encuentre resaltada. |
| ENTER (Continuar) | Continúa con el procedimiento o acepta la entrada de datos. |
| ESC (Salir) | Abandona el sistema definitivamente desde donde se encuentre el usuario. |
| ESC (Regresar) | Retorna a la pantalla que antecede a la actual. |
| F1 (Ayuda) | Muestra información sobre los temas del menú que se muestra en pantalla. |
| F2 (Imprimir) | Manda a la impresora información o gráficas que se encuentran en la pantalla actual. |
| F3 (Ver tabla) | Visualiza la Tabla 'z', la Tabla 't', ó la Tabla 'ji cuadrada'. |

ESTADISTICA I

Utilice esta opción cuando necesite trabajar con los métodos de Estadística Descriptiva; esta parte del sistema permite capturar y organizar los datos en una tabla de distribución de frecuencias, representar gráficamente la distribución de los datos, calcular las medidas de tendencia central y de dispersión, así como el cálculo de las distribuciones de probabilidad Normal, Binomial y de Poisson.

A continuación se describe cada una de las opciones comprendidas en el menú de Estadística I.

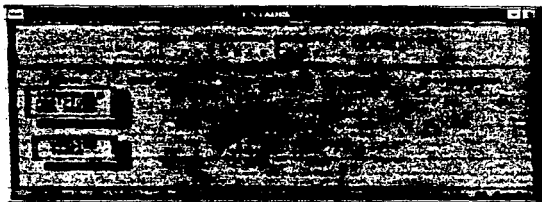
DATOS

Este submenú permite la introducción y manipulación de los datos; la lectura de los datos puede hacerse de dos formas, una de ellas es leer cada una de las observaciones, y la otra es la lectura de la tabla de distribución de frecuencias. Una vez capturados los datos por cualquiera de las dos formas, éstos serán guardados en un archivo para su uso posterior; los datos pueden ser modificados, desplegados en pantalla, impresos o utilizados por otras opciones del menú de Estadística I.

Introducir datos sueltos

Para capturar datos sueltos: teclee el número de observaciones, cada uno de los datos; para modificar alguno, selecciónelo con las flechas de dirección y presione la tecla ENTER cuando haya terminado, los datos serán guardados automáticamente en un archivo nombrado "DATOS.DAT" y usted regresará al menú de Estadística I.

El proceso de selección de las opciones se muestra en la Pantalla No. 2.



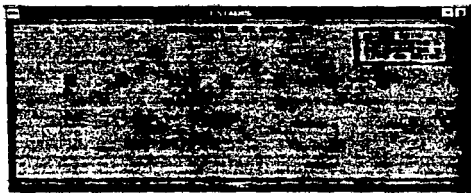
Pantalla No. 2: Proceso de selección de la opción para introducir datos sueltos.

Modificar datos sueltos

Esta opción permite cambiar el valor de alguno de los datos, los cuales ya debieron haber sido capturados.

Para modificar datos sueltos, después de seleccionar la opción, los datos serán desplegados en un formato como se muestra en la Pantalla No. 3; en la misma, se muestra un submenú con las siguientes opciones, elija la más adecuada:

- ***Algún dato.*** Seleccione el dato y prestone ENTER para editarlo.
- ***Incrementar n.*** Le solicita el nuevo valor de n, deberá capturar los datos faltantes.
- ***Decrementar n.*** Solicita el nuevo valor de n, se elimina los datos restantes.
- ***Tipo de datos.*** Le permite alternar entre considerar los datos como sueltos o agrupados. El tipo de datos afecta a las fórmulas empleadas para el cálculo de las medidas de tendencia central, medidas de distribución y gráficas.



Este formato se utiliza para introducir, modificar y visualizar datos sueltos. Para introducir y visualizar los datos no se presenta el menú que aparece del lado derecho en la pantalla.

Pantalla No. 3: Formato que permite la modificación de datos sueltos

Visualizar datos sueltos

Esta opción despliega en pantalla los últimos datos que fueron capturados, y con los cuales se está trabajando.

Leer la tabla de distribución de frecuencias

Como ya se mencionó, otra forma de leer los datos es a través de la captura de la tabla de distribución de frecuencias (ver Pantalla No. 4). Una vez leída, la tabla es guardada en un archivo en disco llamado "TABLA.DAT", para su uso posterior. Para introducir datos directamente a una tabla de distribución de frecuencias, después de seleccionar esta opción teclee: el número de intervalos, el límite inferior, la amplitud entre clases y la frecuencia de cada clase.

Modificar tabla de distribución de frecuencias

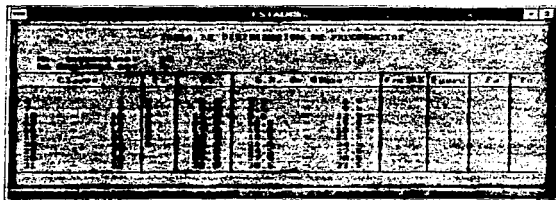
*La opción **Modificar** del menú **Datos**, permite cambiar un elemento o un dato de la tabla de distribución de frecuencias, pero esto sólo es posible si no fueron leídos los datos sueltos, sino la tabla de distribución de frecuencias; de no ser así, puede modificar el tipo de datos en la opción **Modificar** del submenú **Datos**.*

En esta opción, la tabla de distribución de frecuencias será desplegada en una pantalla; en la misma, aparece un submenú con las siguientes opciones, elija la más adecuada:

- **Intervalos.** Le solicita el nuevo número de intervalos; si éste es mayor que el anterior, deberá capturar la frecuencia de cada clase que se incrementó, en caso contrario las últimas clases se pierden.
- **Amplitud.** Le pide el nuevo valor de la amplitud de clases; el programa modifica automáticamente los límites de cada una de las clases.
- **Límite inferior.** Deberá introducir el nuevo valor del límite inferior de la primera clase.
- **Frecuencias.** Le permite modificar la frecuencia de cada una de las clases.

Visualizar tabla de distribución de frecuencias

La opción Visualizar del menú Datos permite ver en pantalla la tabla de distribución de frecuencias con la que se está trabajando; esto sólo será posible si se está trabajando con datos sueltos considerados como agrupados, o si la entrada de los datos se hizo directamente a una tabla de distribución de frecuencias.



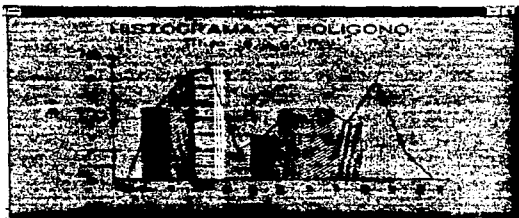
The screenshot shows a window titled 'ESTADÍSTICA' with a menu bar containing 'Datos', 'Gráficos', 'Análisis', 'Ayuda', and 'Salir'. Below the menu bar, there is a title bar for the active window: 'Tabla de Distribución de Frecuencias de la Población'. The main area of the window displays a table with the following columns: 'Clase', 'Frecuencia', 'Frecuencia Relativa', 'Frecuencia Acumulada', 'Frecuencia Relativa Acumulada', and 'Porcentaje'. The table contains several rows of data, with the first row showing a class interval of 1-2, a frequency of 10, and a relative frequency of 0.2.

Clase	Frecuencia	Frecuencia Relativa	Frecuencia Acumulada	Frecuencia Relativa Acumulada	Porcentaje
1-2	10	0.2	10	0.2	20%
2-3	15	0.3	25	0.5	30%
3-4	20	0.4	45	0.9	40%
4-5	15	0.3	60	1.2	30%
5-6	10	0.2	70	1.4	20%

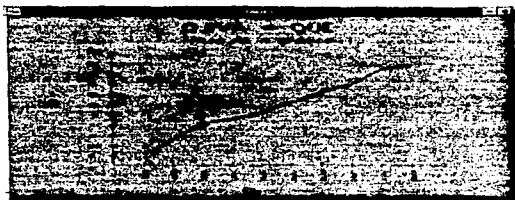
Pantalla No. 4: Lectura de datos directamente a una tabla de distribución de frecuencias.

GRAFICAS

El submenú *Gráficas* crea las gráficas correspondientes a los datos con los cuales se está trabajando, que son: el polígono de frecuencias e histograma (ver Pantalla No. 5), la ojiva + que y la ojiva - que (ver Pantalla No. 6); para ello es necesario que los datos ya hayan sido capturados con la opción **Introducir** del submenú **Datos** y que hayan sido considerados como agrupados, o bien que ya haya sido capturada la tabla de distribución de frecuencias, de lo contrario el submenú *Gráficas* no mostrará ninguna de las opciones contenidas en el mismo. Podrá poner un subtítulo, un título para el eje y un título para el eje x, en cada una de las gráficas.



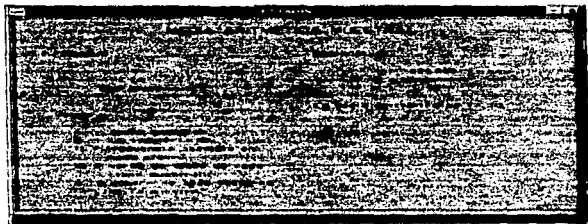
Pantalla No. 5: Gráfica del histograma y polígono de frecuencias



Pantalla No. 6: Gráfica de la Ojiva - Que

MEDIDAS DE TENDENCIA CENTRAL

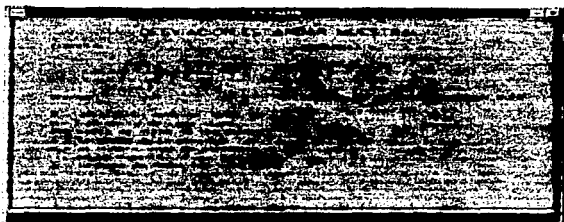
Las opciones del submenú *Tend_Cent* permiten el cálculo de las medidas de tendencia central: *Media*, *Mediana* y *Moda*, tanto para datos agrupados como para datos sueltos; para ello utiliza la tabla o los últimos datos capturados. Los resultados obtenidos serán desplegados en una pantalla similar a la Pantalla No. 7, junto con la fórmula utilizada. El empleo de fórmulas dependerá de cómo fueron considerados los datos, si sueltos o agrupados.



Pantalla No. 7: Formato que muestra el resultado de las medidas de tendencia central para datos sueltos y datos agrupados

MEDIDAS DE DISPERSION

Las opciones de este submenú *Dispersión* permiten el cálculo de las medidas de dispersión: *Desviación estándar*, *Coefficiente de variación*, *Varianza* y *Rango de datos*, tanto para datos sueltos como para datos agrupados. Los resultados obtenidos serán desplegados en una pantalla similar a la Pantalla No. 8 junto, con la fórmula empleada.



Pantalla No. 8. Formato que muestra el resultado de las medidas de dispersión para datos sueltos y datos agrupados.

DISTRIBUCION

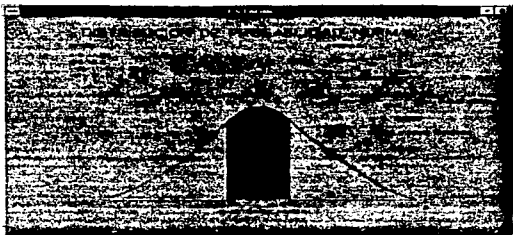
Las opciones que se encuentran en el submenú Distribución permiten calcular la distribución de la probabilidad mediante la distribución Normal, Binomial y Poisson. En el caso de la primera, si el número de desviaciones se encuentra en el rango de ± 3 desviaciones, será graficada.

Distribución Normal

Para obtener la distribución normal, introduzca los datos que se le pide: media (de 0 a 32000) y desviación estándar (de 0 a 32000); posteriormente se despliega un submenú con las siguientes opciones:

- *Menor que la media*
- *Entre dos valores*
- *Mayor que la media*
- *De la media a un valor*

Elija la más adecuada, dependiendo del planteamiento del problema. La gráfica resultante se muestra en la Pantalla No. 9.



Pantalla No. 9: Gráfica de la distribución normal.

Distribución

Para obtener la distribución, introduzca los datos que se le pide: número de ensayos, probabilidad de éxito y número de éxitos.

Distribución Poisson

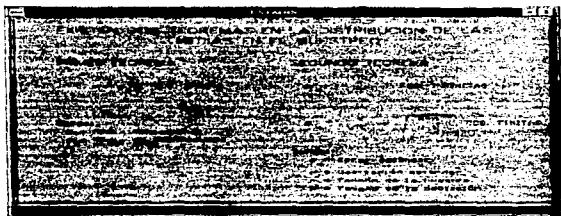
Para obtener la Distribución de Poisson, teclee: número de éxitos y la media.

ESTADÍSTICA II

La opción de Estadística II es la segunda parte del sistema, utilicela si va a trabajar con muestreo, distribuciones de probabilidad en medias y proporciones, estimación del intervalo de confianza y o tamaño adecuado de n tanto para medias como para proporciones, prueba de hipótesis, ji cuadrada, análisis de variancia, regresión mediante el método de mínimos cuadrados y correlación entre dos variables, es decir, cuando necesite inferir sobre el comportamiento de la población, analizando los resultados obtenidos del muestreo aplicado.

MUESTREO

Las opciones del submenú Muestreo le ayudan a comprobar los dos teoremas que existen en la distribución de las medias en el muestreo; éstos se muestran en la Pantalla No. 10, el muestreo puede hacerse con remplazo (permutaciones) o sin remplazo (combinaciones), en ambos casos las combinaciones y permutaciones resultantes del muestreo son desplegadas en pantalla.



Pantalla No. 10: Teoremas de la distribución de las medias en el muestreo

Muestreo sin remplazo (combinaciones)

Para obtener el cálculo de muestreo sin remplazo, introduzca: Tamaño de la Población, cada uno de los datos y el tamaño del que desea que se realice el muestreo.

Muestreo con remplazo (permutaciones)

Para realizar el cálculo de muestreo con remplazo, capture: Tamaño de la Población, y el tamaño en que se realizará el muestreo.

DISTRIBUCION

*El menú **Distribución** permite calcular la distribución de la probabilidad en las medias y en las proporciones, y graficar los resultados obtenidos, con opción a ser impresos.*

Es importante aclarar que si la desviación estándar resultante no se encuentra en el intervalo de ± 3 desviaciones, la gráfica no se visualizará.

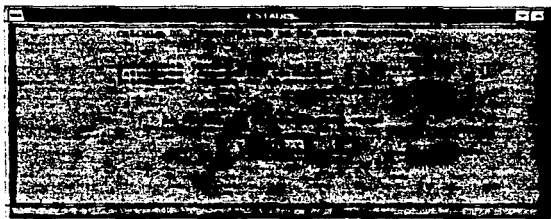
Distribución de probabilidad de la media muestral

Para el cálculo de la probabilidad de la media muestral, son necesarios los siguientes datos: la media poblacional, la desviación estándar y el tamaño de la muestra.

El formato para la lectura de datos se muestra en la Pantalla No. 11; después de capturar los datos, se presenta un submenú con las siguientes opciones:

- *Inferior a un valor*
- *Superior a un valor*
- *Entre dos valores*
- *De la media a un valor*

Elija la opción que considere más adecuada.



Pantalla No. 11: Formato para la lectura de datos para el cálculo de la probabilidad de la media muestral.

Distribución de la probabilidad de la proporción muestral

Para el cálculo de la probabilidad de la proporción muestral, debe introducir: la probabilidad de éxito, el tamaño de la muestra y el tamaño de la población, si esta es finita. Posteriormente, del submenú que contiene las siguientes opciones:

- *Proporción = ...*
- *Proporción < ...*
- *Proporción > ...*
- *Proporción = ...*
- *Entre dos valores;*

elija el tipo de prueba adecuado a su problema.

ESTIMACION

Las opciones del submenú Estimación permiten realizar estimaciones del intervalo de confianza y del tamaño adecuado de n tanto para medias como para proporciones. En el caso del intervalo de confianza, se puede observar la gráfica con los resultados obtenidos.

Estimación del intervalo de confianza para medias

Para obtener la estimación del intervalo de confianza para medias, introduzca: el tamaño de la muestra, el tamaño de la población, la desviación y la media, y seleccione el nivel de confianza apropiado.

Estimación del intervalo de confianza para proporciones

Para obtener la estimación del intervalo de confianza para proporciones, teclee: la probabilidad de éxito, el tamaño de la muestra, el tamaño de la población y el nivel de confianza.

Estimación del tamaño adecuado de n para medias

Para estimar el tamaño adecuado de n para medias, capture: el valor del error, la desviación, el tamaño de la población y el nivel de confianza.

Estimación del tamaño adecuado de n para proporciones

Para estimar el tamaño adecuado de la muestra para proporciones, escriba: el valor del error, la probabilidad de éxito, el tamaño de la población y el nivel de confianza.

HIPOTESIS

Las opciones del submenú Hipótesis permiten la lectura de datos para realizar cálculos de hipótesis tanto para medias como para proporciones. Se puede probar hipótesis para una media, dos medias independientes o apareadas, una proporción y dos proporciones.

Para todos los casos deberá elegir el nivel de confianza, así como seleccionar el tipo de prueba que le interesa, ya sea:

- *Que la media es $>$... Cuando es unilateral por la izquierda*
- *Que la media es $<$... Cuando es unilateral por la derecha*
- *Que la media es $=$ a ... Cuando es bilateral*

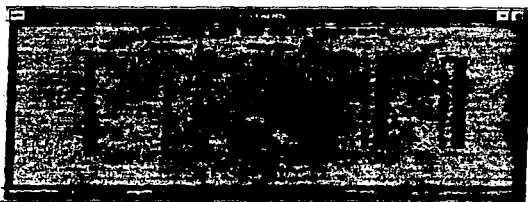
Cabe señalar que si la dispersión se encuentra en un intervalo de ± 3 desviaciones respecto a la media, podrá apreciar la gráfica de la prueba y la sustitución de la fórmula empleada.

Prueba de hipótesis de una media distribuida normalmente

Para obtener cálculo de hipótesis de una media distribuida normalmente, se requiere de los siguientes datos: tamaño de la muestra, media poblacional, media muestral, desviación y nivel de confianza, además de determinar el tipo de prueba.

Prueba de hipótesis para dos medias dependientes

Se utiliza este tipo de prueba cuando necesitamos hacer una comparación entre los resultados obtenidos en dos muestras del mismo tamaño y cuyo tamaño es menor o igual a 30. Si se cuenta con los datos sueltos, debe introducirlos (la lectura de los datos puede apreciarse en la Pantalla No. 12), la computadora se encargará de calcular la media y desviación estándar; en caso contrario, si los datos ya han sido agrupados, teclee la media y la desviación. Posteriormente deberá determinar el nivel de confianza que considera conveniente y seleccionar el tipo de prueba.



Pantalla No. 12: Formato para lectura de datos de hipótesis para dos medias dependientes de datos sueltos

Prueba de hipótesis para dos medias independientes

Para probar una hipótesis con dos medias independientes necesita capturar los datos de ambas muestras para que la computadora calcule la media y la desviación de cada una de ellas, en caso de conocer la media y la desviación deberá introducirlas, posteriormente indique el nivel de confianza y seleccione el tipo de prueba.

Prueba de hipótesis para la proporción

Para realizar la prueba de hipótesis de una proporción de la población, es necesario introducir el tamaño de la muestra, la proporción de la población, la media de las proporciones, así como el nivel de confianza, y seleccionar el tipo de prueba que se desea obtener (bilateral, unilateral por la derecha o unilateral por la izquierda).

Prueba de hipótesis para dos proporciones

Si lo que necesitamos es realizar la prueba comparando la medias de dos proporciones, se requiere introducir el tamaño de la muestra, la proporción de cada una de las muestras, el nivel de confianza y seleccionar el tipo de prueba.

Ji CUADRADA

El submenú Ji Cuadrada permite realizar cálculos del estadístico Ji Cuadrada y el Análisis de variancia, los cuales incluyen temas como calcular Ji Cuadrada como una prueba de la independencia, realizar inferencias sobre una variancia de la población, y calcular pruebas de uno y dos extremos de variancia. Los resultados obtenidos de dichos cálculos y la fórmula utilizada se presentarán en pantalla.

Ji cuadrada como una prueba de independencia

Para calcular el valor de Ji Cuadrada como una prueba de la independencia, teclee: el número de renglones y columnas que contiene la tabla de contingencia, posteriormente se introduce cada uno de los datos de la tabla, a continuación se presenta la tabla con el cálculo del Estadístico Ji Cuadrada, así como la gráfica de la Ji Cuadrada.

Inferencias sobre una variancia de la población

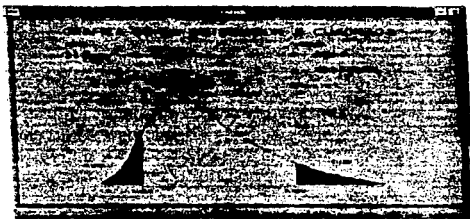
Para obtener el cálculo de inferencias sobre una variancia de la población, capture los datos que se le pide: número de datos con los que va a trabajar y cada uno de los datos; los resultados y la gráfica se despliegan en pantalla.

Prueba de un extremo de la variancia

En el cálculo de prueba de un extremo de la variancia, se necesita los siguientes datos: tamaño de la muestra, desviación poblacional, desviación muestral y nivel de significancia.

Prueba de dos extremos para la variancia

Para realizar el cálculo de prueba de dos extremos para la variancia, teclee: el tamaño de la muestra, la desviación poblacional, la desviación muestral y el nivel de significancia. La gráfica de la Ji Cuadrada para prueba de dos extremos para la variancia, es como la que se muestra en la Pantalla No. 13.



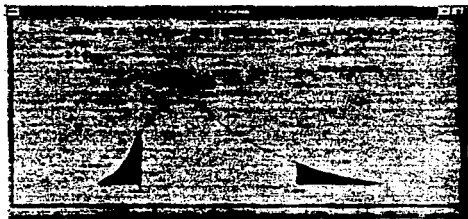
Pantalla No. 13: Gráfica de la Ji Cuadrada para la prueba para dos extremos para la variancia.

REGRESION

Las opciones del submenú Regresión permiten la lectura de datos para realizar el análisis de regresión mediante el método de mínimos cuadrados y correlación entre dos variables; deberá capturar los puntos observados, necesarios para calcular la línea de regresión, crear la gráfica de dispersión de los datos, calcular a Y b, calcular el error estándar de la estimación, así como el grado de correlación existente.

Prueba de dos extremos para la variancia

Para realizar el cálculo de prueba de dos extremos para la variancia, teclee: el tamaño de la muestra, la desviación poblacional, la desviación muestral y el nivel de significancia. La gráfica de la Ji Cuadrada para prueba de dos extremos para la variancia, es como la que se muestra en la Pantalla No. 13.



Pantalla No. 13: Gráfica de la Ji Cuadrada para la prueba para dos extremos para la variancia.

REGRESION

Las opciones del submenú Regresión permiten la lectura de datos para realizar el análisis de regresión mediante el método de mínimos cuadrados y correlación entre dos variables; deberá capturar los puntos observados, necesarios para calcular la línea de regresión, crear la gráfica de dispersión de los datos, calcular a y b , calcular el error estándar de la estimación, así como el grado de correlación existente.

Introducir puntos para el cálculo de la línea de regresión

Para la graficación de la línea de regresión, es necesario capturar el número de puntos y cada uno de ellos. La lectura de los puntos es: valor en x y valor en y para cada uno de los puntos; si desea interrumpir la lectura, preste la tecla ESC.

Tabla de datos del cálculo de la línea de regresión

Esta opción muestra la tabla con los cálculos necesarios para estimar el valor de a (pendiente) y b (ordenada al origen).

Gráfica de dispersión de datos

Para visualizar la gráfica de la dispersión de los datos y la línea del mejor ajuste: teclee el valor de la variable independiente, en caso de que desee estimar su posición; en caso contrario, teclee cero, así como los títulos y subtítulos para la gráfica, mismos que son opcionales; sólo teclee ENTER en cada uno de ellos para poder observar la gráfica.

Esta gráfica sólo puede ser observada si ya se ha introducido los puntos.

Cálculo de a y b

Seleccione esta opción si desea observar el cálculo de a y b.

Error estándar de la estimación

Si los datos ya han sido tecleados y selecciona esta opción, podrá obtener el cálculo del error estándar de la estimación.

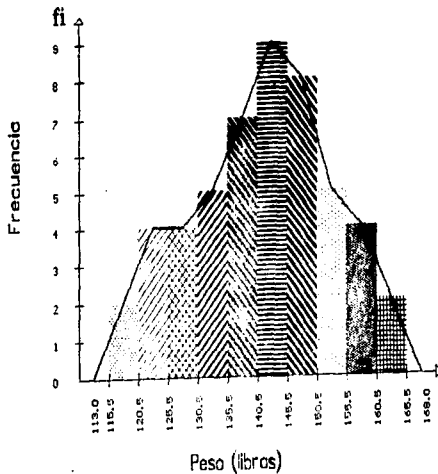


ANEXO

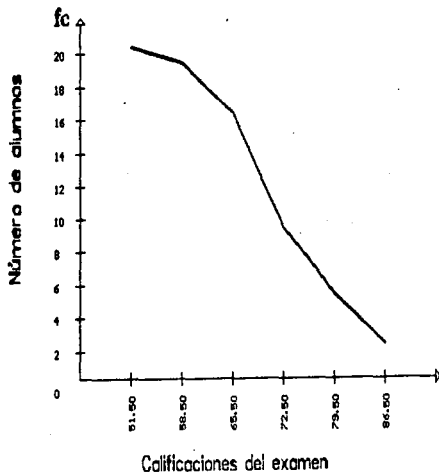
**GRÁFICAS QUE GENERAN
UN SISTEMA**

HISTOGRAMA Y POLIGONO

Pesos de 50 varones



OJIVA + QUE CALIFICACIONES



DISTRIBUCION DE PROBABILIDAD NORMAL

DATOS:

$$\mu = 5.0$$

$$\sigma = 0.4$$

$$\text{valor}(x_1) = 5.5$$

$$\text{valor}(x_2) = 6.2$$

FÓRMULA:

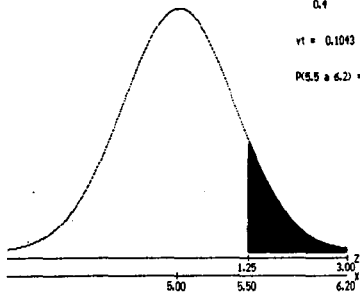
$$Z = \frac{x - \mu}{\sigma}$$

$$Z^1 = \frac{5.5 - 5.0}{0.4} = 1.25 \text{ desv.}$$

$$Z^2 = \frac{6.2 - 5.0}{0.4} = 3.00 \text{ desv.}$$

$$vt = 0.1043$$

$$P(5.5 \leq 6.2) = 0.1043$$



PROBABILIDAD DE LAS MEDIAS MUESTRALES

DATOS:

$\mu = 52.0$
 $\sigma = 5.0$
valor = 51.0
error = 0.205

FINITAS

FORMULA:

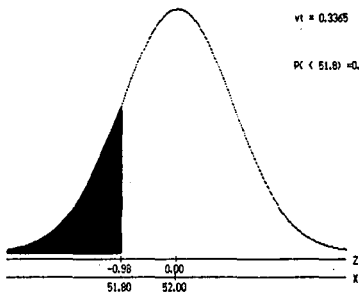
$$Z = \frac{\bar{x} - \mu}{\sigma}$$

$$Z = \frac{51.0 - 52.0}{0.205}$$

$$Z = -0.98 \text{ desviaciones}$$

$$vt = 0.3365$$

$$PK < 51.0 = 0.1635$$



CALCULO DEL INTERVALO PARA LA MEDIA FINITA

DATOS:

n = 36
 $\sigma = 80.000$
 $\mu = 810.000$
 $N_{L,C} = 90$
 $H = 50$

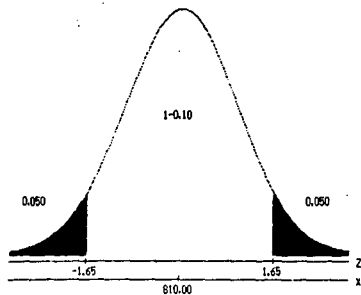
DESVIACION CONOCIDA

FORMULA:

$$e = \frac{\sigma}{\sqrt{n}} \cdot \sqrt{\frac{H-n}{H-1}} \cdot z$$

$$e = \frac{80.00}{\sqrt{36}} \cdot \sqrt{\frac{50-36}{50-1}} \cdot 1.650$$

$$e = 11.75919$$



$$798.2405 < \mu < 821.7595$$

CALCULO DEL INTERVALO PARA LA PROPORCION FINITA

DA-TOS:

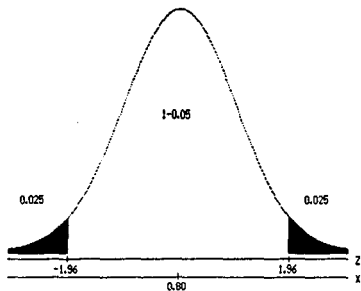
$p = 0.800$
 $q = 0.200$
 $n = 200$
 $N = 500$
 $N_c = 95 \%$

FORMULA:

$$e = \sqrt{\frac{p \cdot q}{n}} \cdot \sqrt{\frac{N-n}{N-1}} \cdot z$$

$$e = \sqrt{\frac{(0.800)(0.200)}{200}} \cdot \sqrt{\frac{500-200}{500-1}} \cdot 1.96$$

$$e = 0.04298$$



$$0.7570 <= P <= 0.8430$$

PRUEBA DE HIPOTESIS PARA DOS MEDIAS

Datos:

$n1 = 40$
 $n2 = 40$
 $\bar{x}1 = 75.000$
 $\bar{x}2 = 72.000$
 $S1 = 9.200$
 $S2 = 0.050$
 $\alpha = 0.10$

Hipótesis:

$H0: \mu1 = \mu2$
 $H1: \mu1 > \mu2$ Rechazada

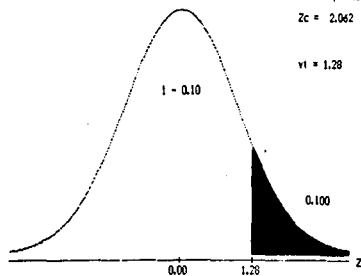
Prueba: Unilateral por la derecha

$$Zc = \frac{\bar{x}1 - \bar{x}2}{\sqrt{\frac{(S1)^2}{n1} + \frac{(S2)^2}{n2}}}$$

$$Zc = \frac{75.000 - 72.000}{\sqrt{\frac{(9.20)^2}{40} + \frac{(0.05)^2}{40}}}$$

$$Zc = 2.042$$

$$v1 = 1.28$$



PRUEBA DE HIPOTESIS PARA LA PROPORCION

Datos:

$n = 400$
 $\hat{p} = 0.260$
 $p = 0.320$
 $\alpha = 0.10$

Hipótesis:

$H_0: \hat{p} = 0.260$
 $H_1: \hat{p} \neq 0.260$ Rechazado

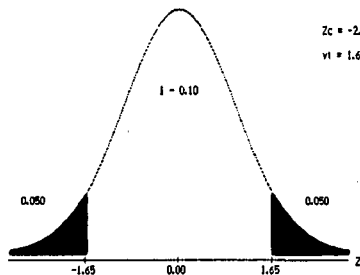
Prueba: Bilateral

$$Z_c = \frac{\hat{p} - p}{\sqrt{\frac{p \cdot q}{n}}}$$

$$Z_c = \frac{0.26 - 0.32}{\sqrt{\frac{(0.32)(0.68)}{400}}}$$

$$Z_c = -2.5725$$

$$v_t = 1.65$$



GRAFICA DE LA DISTRIBUCION JI CUADRADA

FORMULA

$$\chi^2 = \frac{(fo - fe)^2}{fe}$$

$$\chi^2 = 31.968$$

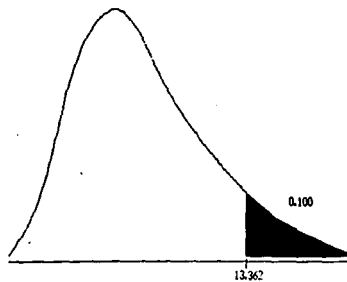
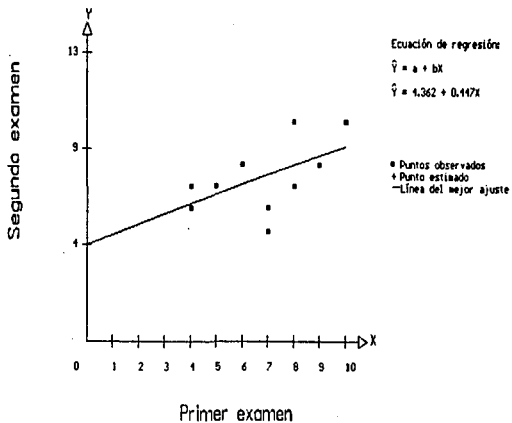
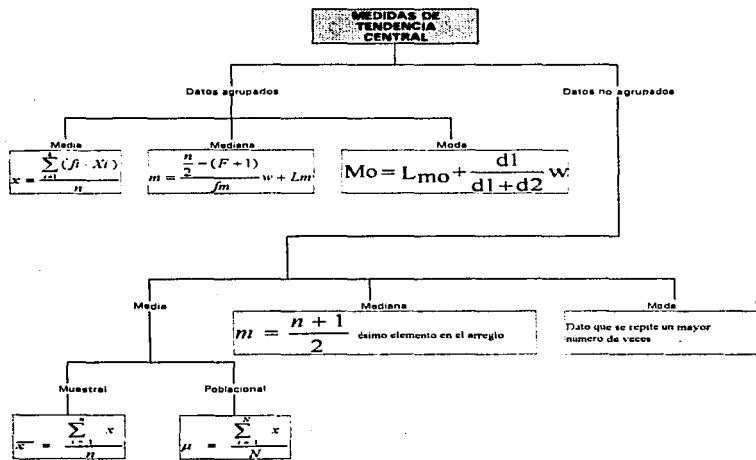


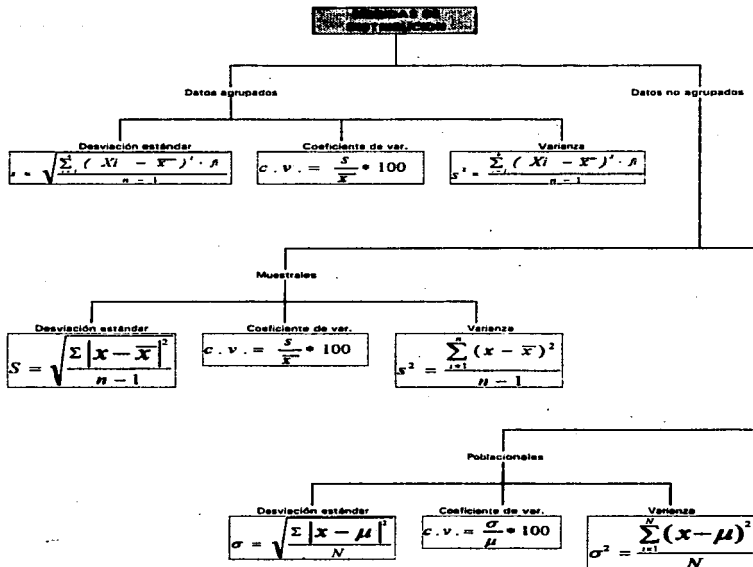
DIAGRAMA DE DISPERSION CALIFICACIONES

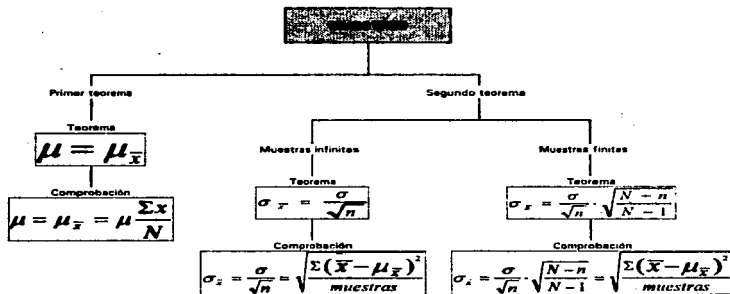
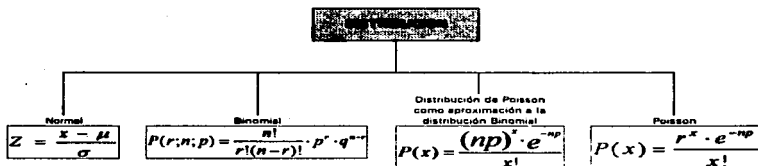


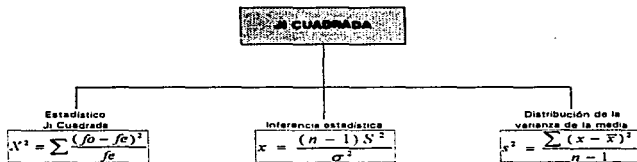
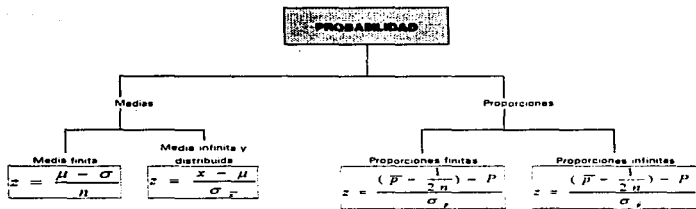
ANEXO C

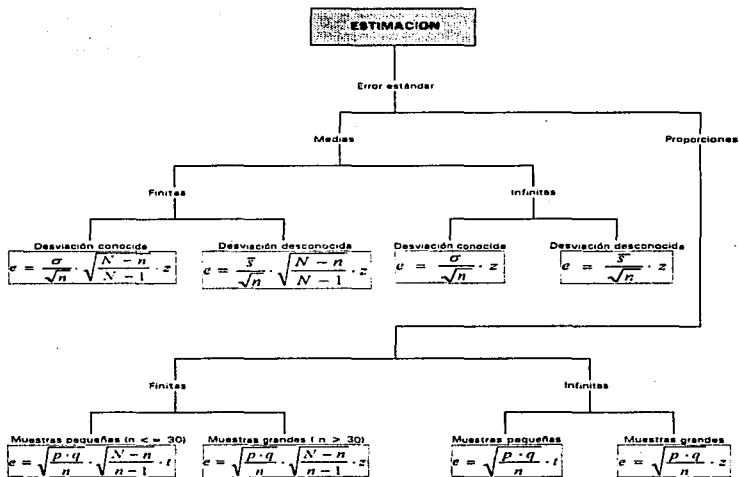
FORMULAS UTILIZADAS

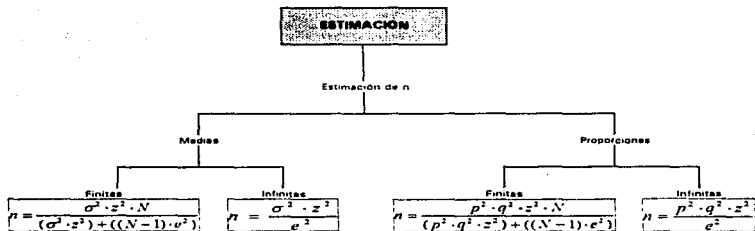


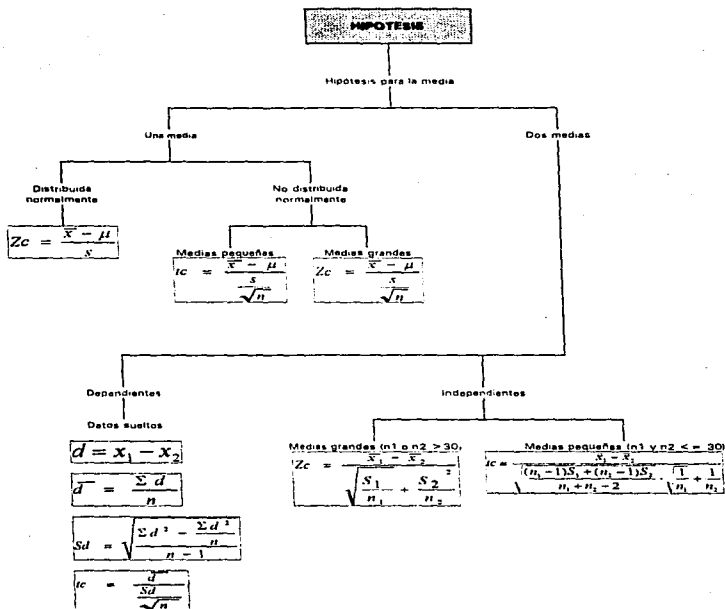


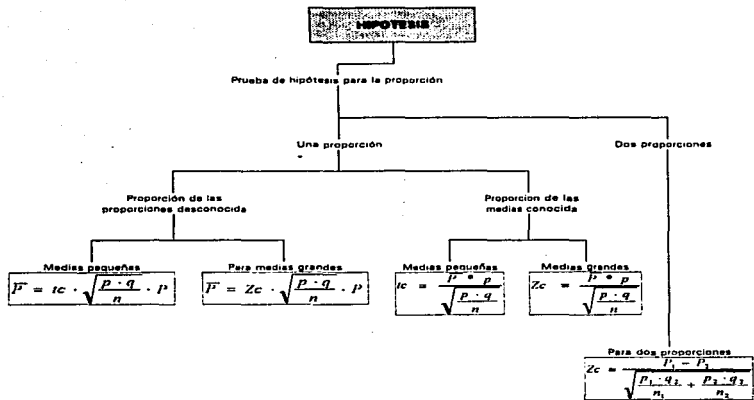


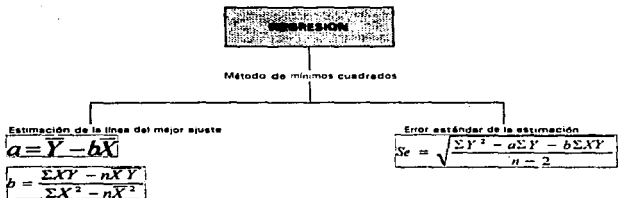












ANEXO D

**COMUNO FUENTE DEL
ESTERIA**

PROGRAMA DE ESTADISTICA
DESCRIPTIVA

```

#define NLISTV 6
#define MAX100
#define NLISTN 9
#define NOMP 34
#define "alfo h"
#include <dos h>
#include <allo h>
#include <strng h>
#include <con h>
#include <graph h>
#include <math h>
#include <manan h>
#include <time h>
#include <ctype h>
#include <std h>
#include <memsh h>
#include <su h>
#include <prev h>

struct tab {
float lmanf,
float lrmup,
float lrmvmp,
float lrmvmp,
int fl,
int fe,
float %d,
float fp,
float fr,
} ar[10];

float mem[10];
int colormem;
int datos = 1;
int tipo = 3;
int mas = 1;

FILE *fb, *arch;

void main()
{
int opc = 0;
int nam = 0;
int nader = 0;
float d,d1,amp,msrv,meda,
prn, invrtvmanf(1,1,80,24,0),
prn, colvrtvmanf(15,1,12,1,13),
vman, invrtvmanf(1,25,80,25,0),
vman, colvrtvmanf(1,3,4,3,0),
msrvcolmem(),
for (int i=1; i<=10; i++) mem[i] = 0;
// leer archvms
if (arch=fopen("memansun.dat","r+b")) != NULL {
if (fread(mem,sizeof(mem),1,arch)
==BUSCAME);
datos = TUPDATOS(),
if (datos == 1) {
tipo = mem[0]+1;
BUSCAR(&A&D),
msrv = INTERVALOS(n,&amp;D,d),
} else { msrv = mem[0];
tipo = 2; }
} else {
mas = 1;
} fclose(arch);
if (mas)
if (datos == 1 && tipo == 2)
if (fopen("Tabla.dat","r+b")) != NULL {
if (fread(ar,sizeof(ar),1,fp) == 1)
break(ar,sizeof(ar),1,fp);
else
tipo = 3;
fclose(fp);
} else tipo = 3;
else
if (datos == 2)
if (fopen("Tabla.dat","r+b")) != NULL {
if (fread(ar,sizeof(ar),1,fp) == 1)
break(ar,sizeof(ar),1,fp);
else
mas = 1;
fclose(fp);
} else tipo = 0;
INIGRAF(CV%I),
PREDESTA(),
CERRARGRAF(CV%),
textbackgrnd(BLUE),
clear(),
prn dtabar(),
do {
memsh(mem,sizeof),
memsh(invrtvmanf(2,9,78,9)),
memsh(colvrtvmanf(15,1,1,3)),
memsh(msrvcolmem(0,5,1,"DATOS "
"Captura, modificación de los datos "*,0-2),
"Polígono e histograma y ejivas "*,1-2),
else
memsh(msrvcolmem(1,17,1,"GRAFICAS "
"Polígono e histograma y ejivas "*,2-2),
memsh(msrvcolmem(2,1,1,"TEND-CIENT "
"Medias, medianas y moda "*,2-2),
memsh(msrvcolmem(3,46,1,"DISTRIBUCION "
"Distribucion unimodal, bimodal, etc "*,3-2),
memsh(msrvcolmem(4,6,1,"DISTRIBUCION "
"Normal, Bivariada y de Poisson "*,4-2),
memsh(0) colvrtvmanf(15,1,1,3,4,"*****"),
if (datos==1) // datos leidos
if (tipo==1) // datos sueltos
memsh(0) memsh(3,3),
memsh(0) invrtvmanf(3,10,16,14,2),
} else {
memsh(0) memsh(4,5),
memsh(0) invrtvmanf(3,10,16,15,2),
} memsh(0) msrvcolmem(5,3,2,"INTRODUCIR "
" Lectura de los datos o la tabla. "*,5-1),
memsh(0) msrvcolmem(6,3,"MODIFICAR "
" Edición de los datos, tipo y número "*,2-1),
memsh(0) msrvcolmem(7,3,4,"VER DATOS "
" Muestra los datos que fueron leidos "*,2-1),
if (tipo==2)
memsh(0) msrvcolmem(8,3,5,"VER TABLA "
" Tabla de distribución de frecuencias "*,2-1),
} else // tipo leida
memsh(0) memsh(3,5),
memsh(0) invrtvmanf(3,10,16,14,2),
memsh(0) msrvcolmem(5,3,2,"INTRODUCIR "
" Lectura de los datos o la tabla. "*,5-1),

```


ANEXO D: Código fuente del sistema

```

DESVIA, EST(n,media,interv.) } break.
case 1536 MEDIA(n,media,n). break.
DESVIA, EST(n,media). break.
case 1527 MEDIA(n,media,n). break.
DESVIA, EST(n,media). break.
case 1601 if (dacion == 2) {
suma = 0.
for(i=0; i<num(0); i++) suma += ar[i] * Xi * ar[i] / fi.
media = suma/n.
COEFICIENTE(n,media,num(0)).
} else {
MEDIA(n,media,n).
COEFICIENTE(n,media,interv.).
} break.
case 1626 MEDIA(n,media,n). break.
COEFICIENTE(n,media,interv.). break.
case 1627 MEDIA(n,media,n). break.
COEFICIENTE(n,media). break.
case 1700 if (dacion == 1) {
MEDIA(n,media,n).
VARIANZA(n,interv,media.) }
else {
suma = 0.
if (dacion == 2) {
for(i=0; i<núm(0); i++) suma += ar[i] * Xi * ar[i] / fi.
media = suma/n.
}
VARIANZA(n,num(1), num(0), media).
} break.
case 1726 MEDIA(n,media,n). break.
VARIANZA(n,n,media). break.
case 1727 MEDIA(n,media,n). break.
VARIANZA(n,media). break.
case 1800 DISPERSION(n). break.
case 1900 NORMAL(n). break.
case 2000 BINOMIAL(n). break.
case 2100 POISSON(n). break.
case -3 INFORMATO() break.
}
if (opc == 522 && opc == 800 && cas) {
n = BUSCAN(n).
if (dacion == 1) {
BUSCAN(n,media,n).
interv = INTERVALOS(n,damp,D,d).
} else interv = num(0).
}
if (opc == 900 && opc == 1800) {
setbackgromod(BLUB).
clear().
prin delugar().
} else prin limpiar().
} while (dacion == 0).
TIERRAGRAFICOS().
writeln(1,80, 23).
setbackgromod(10).
clear().
}
// FUNCIONES DE LECTURA Y MODIFICACION DE DATOS
// Lee un conjunto de datos y crea la tabla de distribución
// de frecuencias, según los gráficos en archivo en disco en CREARARCHIVOS()
en n, key, sum = 0.
num(0) = numnum(2,29).
num(1) = numnum(33, 10,47,13,2).
num(2) = numnum(19,1,1,3,4).
num(3) = numnum(29,2,2) SURTOS """,2,2).
num(4) = numnum(30,3,3) AGRUPADOS """,2,2).
prin activar(0).
if (arch = fopen("Numeros.dat","w+b")) != NULL)
if (fwrite(num,num,sizeof(num),1,arch) == 1)
ERROR ("ERROR El archivo 'Numeros.dat' ya
existe. No continue se borra n". Los datos 'Numeros.dat'
del ran = LEER (52,12,"Introduzca otro q o n").
if (ran = 'N')
return 1.
else prin limpiar().
}
limdad(22).
getkey(30, 2). oprant("LECTURA DE LOS DATOS").
getkey(24,13). oprant(" Cuantos datos va a introducir ").
if (key == -1 or (2,5,8,11,3,3,10,INT) == '3') {
if (key = 'ESC') { clear(arch). return 1. }
prin limpiar().
clear(arch).
for (i=0; i = 10). i++) num(i) = 0.
getkey(30, 2). oprant("LECTURA DE LOS DATOS").
vnum oprant("Introduzca el dato y presione ENTER").
limdad(22).
key = LEERDATOS(n,0,0).
if (key = 'ESC') { clear(arch). return 1. }
num(i) =
vnum en (suma).
prin limpiar().
setback(12).
getkey (21, 4). oprant("Los datos deben ser considerados como ").
un, err, (-1, 1).
en el tipo "sci", sub(am). //Guarda el tipo de datos 2: numerales, 3: potenciales
un, err(6, 7).
if (del_tipo == -1) {
if (del_tipo == -29) tipo = 3.
else tipo = 2.
num(i+1) = tipo.
dacion = num(n-2) - 1.
if (num(n+1) == 2) //datos agrupados
if (num(2)) return 1.
CREARTABLA(n).
grafar(2).
}
}
}
}
// Permite modificar el conjunto de datos
en MODIFICADATOS()
en n, key, opcion = 0.
if (dacion == 1) return 1.
n = BUSCAN(n).
limdad(12).
do {
prin limpiar().
num(0) = numnum(4,29).
num(1) = numnum(61,2,77,7,0).
num(2) = numnum(15,1,1,3,4).
num(3) = numnum(29,3,2) ALGUN DATO ".
num(4) = numnum(30,3,3) "INCREMENTAR n".
num(5) = numnum(30,3,3) "TIPO DE DATOS".
num(6) = numnum(31,3,4) "DECREMENTAR n".
num(7) = numnum(32,3,5) "TIPO DE DATOS".
num(8) = numnum(33,3,5) "TIPO DE DATOS".
num(9) = numnum(34,3,5) "TIPO DE DATOS".
} while (opcion != 0).
DISEP_DATOS(n).
getkey(29, 2). oprant("MODIFICACION DE LOS DATOS").
un, err(-1,-1).
opc = leer, sub(am).
un, err(6, 7).
prin activar(0).
setback(Cope) 1.
case 29. LEERDATOS(n,0,0,1).
break.
case 30. getkey (18, 23). oprant("Nuevo valor de n: del").
en n = leer(key,76,23,2,AMAX,INT).
getkey (18,23). oprant(" ").
if (ran = 0)

```

```

LEERDATOS(mul,m),
num(n) = n = num,
num(n-1) = tipo,
num(n-2) = datos,
break,
case 31 gotoxy(58,23); printf("Nuevo valor de n: %d\n",
    num);
    scanf("%d",&n);
    gotoxy(58,23); printf("
");
    if (n != num) {
        for (int i=num, <n-3, i--> num[i] = 0,
            num[i]=num,
            num[i-1]= tipo,
            num[i-2]= datos,
            i=num,
        }
        break;
case 32: num[n] = multarm(2,29);
    num[n-1] = desconvirted(53.8,76,11,0);
    num[n-2] = colvertarm(15,1,1,3,4);
    num[n-3] = invleemarm(29,3,2 - SUELTOS * * - 2, -2);
    num[n-4] = invleemarm(30,3,3-AGRUPADOS * * - 2, -2);
    int_cve(-1,-1); //Cambia el tipo de datos 0:monstrales,
    tipo = leer_sub(m); // publicacionales
    int_cve(6,7);
    printf("activar(0);
    if (tipo == 29) tipo = 3,
    else tipo = 2,
    num[n+1] = tipo,
    datos = num[n-2] + 1,
    break;
}
while (tipo != -2);
if (num[n+1] == 2) //datos agrupados
    if (cve(2)) return 1,
    macul(marm),
    CLEARTABLA(n),
    grabar(2);
}
grabar(1);
return 0;
}
// Muestra los datos en pantalla de texto
void DIMP_DATOS(int n)
{
    int row, col,dec;
    gotoxy(29,2);
    dec = BUSCADOR(n),
    printf("DATOS QUE FUERON LEIDOS",
    row=8,
    col=20;
    tamaño(15);
    for (int i=0, i++, i++) {
        col=leer(m, col,rm),
        gotoxy(col,rm);
        switch (dec) {
            case 0: printf("%d", num[i]); break;
            case 1: printf("%f", num[i]); break;
            case 2: printf("%d", num[i]); break;
            case 3: printf("%d", num[i]); break;
        }
    }
}
// Muestra los datos en pantalla grafica
void DIMP_DATOSG(int n)
{
    int row, col,dec;
    clear(30);
    if (dec)
        dec = BUSCADOR(n);
    else dec = num[3];
    for (i=0, row = 100, col=340; i++, row++)
        if ((i % 6) == 0) (col=340, row=-20);
        switch (dec) {
            case 0: printf("%d", num[i]); break;
            case 1: printf("%f", num[i]); break;
            case 2: printf("%d", num[i]); break;
            case 3: printf("%d", num[i]); break;
        }
        cout<<endl;
        col+=45;
    }
}
// Muestra los datos en pantalla
void DISPLAY(int n)
{
    tamaño("27");
    printf("activar(0);
    dec = BUSCADOR(n);
    DIMP_DATOS(n);
    int key;
    while (key != ESC || key != ENTER)
        if (key == F1 || key == F2)
            DIMP_DATOSG(n);
        }
    while (key != ESC || key != ENTER);
}
// Imprime en la impresora disponible, el conjunto de datos capturado
void IMP_DATOS(int n, int dec)
{
    int col = 8,
    clear(40);
    float num = num[0];
    for (int i=1; i++, i++)
        if (num < num[i]) num = num[i];
        switch (dec) {
            case 0: printf("%d", num); break;
            case 1: printf("%f", num); break;
            case 2: printf("%d", num); break;
            case 3: printf("%d", num); break;
        }
        int num = num(n)-5;
        printf("%d", num);
        for (i=0, col = 0; i++, col+=num) {
            switch (dec) {
                case 0: printf("%d", num[i]); break;
                case 1: printf("%f", num[i]); break;
                case 2: printf("%d", num[i]); break;
                case 3: printf("%d", num[i]); break;
            }
            dec (int i=leemarm(n), i=conv; i++) s[i] = *;
            s[i] = "0";
            if ((col+num) > 80) {
                printf("%d", num);
                col = 0;
            }
        }
        printf("%d", num);
        printf("%d", num);
    }
}
// Lee un conjunto de datos n, a partir del momento hasta cuando
int LEERDATOS(int n, int num, int num)
{
    int col, row,
    int key = ENTER,
    col = 20 + (((num % 7) - 1) * 9);
    row = 0 + col (num % 7);
    while (num % col < 0) {
        if (num % col < 0) {
            key = num % col + 8;
            key = leer (dec,num,num,col,row,0,0,5432,0,1,1);
        }
    }
}

```


Anexo D: Código fuente del sistema

```

for (i=0, i:=num[0], i++) {
  genkey(21,12+);
  cprand("ud",art[1] fi);
  vsum avsum("Frecuencia absoluta de cada clase ");
  LIMPRES("LIMITE INFERIOR: num[0], interv-1);
  for (i=0, i:=i, i:=interv, i++) num[5] := art[1] fi;
  vsum avsum();
} else { // decremento de intervacion
  for (i, interv, i:=num[0], i++) {
    art[1] interv := art[1] interv - 0;
    art[1] interv := art[1] interv - 0;
    art[1] fi := art[1] fi - 0;
    art[1] %6 := art[1] fi - art[1] fi - 0;
  }
  for (i:=0, i:=num[0], i:=interv, i++) {
    genkey(22,12+);
    cprand("ud",art[1] fi);
    num[5] := art[1] fi;
  }
  num[0] = interv;
  FRECUENCIAS(num[5],interv);
}
break;
case 30 amp := num[1];
genkey (60,23); cprand("Amplitud % 3F",num[1]);
if (key = int (d:=amp, %23.0,0.01),3,2000,FLOAT) == ESC)
  return;
genkey (60,23); cprand(" " " ");
if (amp := num[1] && key := ESC) {
  num[3] = amp;
  num[3] := BULSCABDC(3);
  num[4] := DIF(num[3]);
  LIMITES(num[0], num[1], num[3], num[4]);
}
break;
case 31 float lim, inf;
vsum avsum("Límites inferior de la clase ");
if (key = int (d:=lim, %23.0,0.01,32000,FLOAT) == ESC) return;
vsum avsum();
if (lim := int (art[0] interv && key := ESC) {
  art[0] interv := num[3] := lim, inf;
  num[3] := BULSCABDC(3);
  num[4] := DIF(num[3]);
  LIMITES(num[0], num[1], num[3], num[4]);
}
break;
case 32 int n=0;
vsum avsum("Frecuencia absoluta de cada clase ");
key = LIMPRES("FRECUENCIAS: num[0], 0, num[0]-1);
for (i=0, i:=num[0], i++) n := art[1] fi;
vsum avsum();
FRECUENCIAS(num[0], num[0]);
break;
}
while (key != -2);
grbar(3);
return num[5];
}
//Fin de tabla
void CREAR_TABLA(n)
{
  float d, D, amp, def;
  int interv;
  char s[10];
  BULSCABR(d,d,d,1,n);
  interv = INTERVALOS(n, d, amp, D, d);
  ut d:= BULSCABDC(n);
  avsum (d:=d) {
    case 0 d:= 1;
    if (amp:=1) amp=1;
    cprand(" % 3F", amp); amp := int(d); break;
    case 1 d:= 1; cprand(" % 3F", amp); amp := int(d); break;
    case 2 d:= 0; cprand(" % 3F", amp); amp := int(d); break;
    case 3 d:= 0; cprand(" % 3F", amp); amp := int(d);
  }
}
}
LIMITES(d,d,D, amp, interv, n, def);
//Calcula los límites y marca de clase de la tabla de distribución
void LIMITES(int interv, float amp, int dec, float d)
{
  art[0] interv = art[0] interv + amp-def;
  for (int :=0, i:=interv, i++) {
    if (i=0) art[1] interv := art[1] interv + amp;
    art[1] interv := art[1] interv + amp-def;
    art[1] %6 := (art[1] interv - art[1] interv)/2;
    art[1] interv := art[1] interv+(float)d/2;
    art[1] interv := art[1] interv+(float)d/2;
  }
  int ren = 12;
  textcolor(15);
  for (i=0, i:=interv, i++) {
    avsum (d:=d) {
      case 0 genkey (3,ren+);
      cprand(" % 3F", %6 OF art[1] interv,art[1] interv);
      genkey (26,ren+); cprand(" %6 3F",art[1] %6);
      genkey (36,ren+);
      cprand(" % 8 3F", %8 3F, art[1] interv,art[1] interv);
      break;
      case 1 genkey (3,ren+);
      cprand(" % 3F", %6 3F, art[1] interv,art[1] interv);
      genkey (27,ren+); cprand(" %6 3F",art[1] %6);
      genkey (36,ren+);
      cprand(" % 8 3F", %8 3F, art[1] interv,art[1] interv);
      break;
      case 2 genkey (3,ren+);
      cprand(" % 2F", %6 2F, art[1] interv,art[1] interv);
      genkey (26,ren+); cprand(" %6 3F",art[1] %6);
      genkey (35,ren+);
      cprand(" % 9 3F", %9 3F, art[1] interv,art[1] interv);
      break;
      case 3 genkey (2,ren+);
      cprand(" % 7 3F", %6 3F, art[1] interv,art[1] interv);
      genkey (26,ren+); cprand(" %6 3F",art[1] %6);
      genkey (34,ren+);
      cprand(" % 10 4E", %6 4E, art[1] interv,art[1] interv);
    }
  }
}
// Cálculo y escribe los frecuencias complementarias.
// porcentual, acumulada y relativa
void FRECUENCIAS(int n, int interv)
{
  art[0] fi = art[0] fi; art[0] %6 = n;
  for (int :=interv+1, i:=interv, i++) {
    if (i=0) {
      art[1] (sum:=sum+1) fi := art[1] fi;
      art[1] (cum:=cum+1) fi := art[1] fi;
    }
    art[1] fi := (float) art[1] fi / (float) n;
    art[1] (p:=float) art[1] fi * 100;
    genkey(56,12+); cprand(" % 6F", art[1] fi); art[1] fi;
    genkey(66,12+); cprand(" % 6F", art[1] fi); art[1] fi;
    genkey(69,12+); cprand(" % 6F", art[1] fi); art[1] fi;
    genkey(73,12+); cprand(" % 6F", art[1] fi);
  }
}
// Despliega la tabla de distribución
void DESP_TABLA(int n, int cprand)
{
  int d,dec;
  float d, D, interv, amp;
  interv:=("D");
  amp:=("amp");
  prin(interv);
  if (d:=n=1)
    BULSCABR(d,d,d,n);
}

```

```

interv=INTERVALOS(n, &serv, D, d).
dec = BURCADRC(n).
} else {interv=mas(n),
      dec = mas(n),
      }
TABLA(interv),
textcolor(12),
gotomy(23,4), sprntf("TABLA DE DISTRIBUCION DE FRECUENCIAS"),
textcolor(13),
for (h=0, k=interv, k++)
  switch (dec)
  case 0 gotomy(12,h),
        sprntf("%d 0%",&h,ar[h]) lmasf,ar[h] lmasap, break.
  case 1 gotomy(12,h),
        sprntf("%d 1%",&h,ar[h]) lmasf,ar[h] lmasap, break.
  case 2 gotomy(12,h),
        sprntf("%d 2%",&h,ar[h]) lmasf,ar[h] lmasap, break.
  case 3 gotomy(12,h),
        sprntf("%d 3%",&h,ar[h]) lmasf,ar[h] lmasap, break.
  }
gotomy(22,12+h), sprntf("%3d",ar[h] f),
vswitch(dec)
case 0 gotomy(27,12+h), sprntf("%6d 1%",ar[h] %), break.
case 1 gotomy(27,12+h), sprntf("%6d 2%",ar[h] %), break.
case 2 gotomy(27,12+h), sprntf("%6d 3%",ar[h] %), break.
case 3 gotomy(27,12+h), sprntf("%6d 4%",ar[h] %),
}
switch (dec) {
case 0 gotomy(36,12+h),
      sprntf("%8.1f,%8.1f",ar[h] lmasf, ar[h] lmasap), break.
case 1 gotomy(36,12+h),
      sprntf("%8.2f,%8.2f",ar[h] lmasf, ar[h] lmasap), break.
case 2 gotomy(36,12+h),
      sprntf("%8.3f,%8.3f",ar[h] lmasf, ar[h] lmasap), break.
case 3 gotomy(36,12+h),
      sprntf("%8.4f,%8.4f",ar[h] lmasf, ar[h] lmasap),
}
gotomy(56,12+h), sprntf("%4d", ar[h] f),
gotomy(64,12+h), sprntf("%6.0f%*", ar[h] f),
gotomy(69,12+h), sprntf("%6d", ar[h] f),
gotomy(75,12+h), sprntf("%6d", ar[h] f),
}
if (cpms) {
  int boy;
  do {
    if (boy = realmax("D2?")) == F2)
      IMP_TABLA(interv),
    } while (boy != ESC ||& boy != ENTER),
}
// Crea en pantalla la tabla, según el número de intervalos
// y TABLA(interv)
{
  int lmas;
  textcolor(12),
  gotomy(2,9),
  sprntf(" Clase f Xi L.R. de Clase Sides Space Ss S*"),
  textcolor(4),
  for (y = 1; y<=D, y++) {gotomy(1,8), sprntf("-"),
}
if (interv < 10) mas = 4;
else mas = 3;
for (i=0, i=mas+mas, i++) {
  gotomy(20,i+9), sprntf("), gotomy(25,i+9), sprntf("),
  gotomy(33,i+9), sprntf("), gotomy(53,i+9), sprntf("),
  gotomy(62,i+9), sprntf("), gotomy(74,i+9), sprntf("),
  }
for (y = 1, y<=D, y++) {gotomy(1,10), sprntf("-"),
for (y = 1, y<=D, y++) {gotomy(2,interv+mas+9), sprntf("-"),
}
// Imprime la tabla de distribución de frecuencias
void IMP_TABLA(int interv)
{
  int col, row,
  char a[2],
  int are,
  if (interv < 10) are = 4;
  else are = 3;
  sprntf("\n",stdgen),
  sprntf("TABLA DE DISTRIBUCION DE FRECUENCIAS",stdgen),
  sprntf("\n",stdgen),
  for (row = 1, row < 10+mas+mas, row++) {
    for (col = 1, col<=D, col++) {
      gotomy(col,row,col,row, &a[0]),
      a[1] = "0",
      if (a[0] == "1" & a[1] == "1" & a[0] == "A")
        a[0] = "1",
      else
        if (a[0] == "1" & a[1] == "1" & a[0] == "A")
          a[0] = "1",
        }
      sprntf(a,col,row),
    }
    sprntf("\n",stdgen),
  }
  // Calcula los límites reales, marcas de clase y frecuencias
  void LIMITESS(flon D,flon D,flon amp, int n, flon dec)
  {
    int i,j;
    // límites inferior y superior
    ar[0] lmasinf=0,
    ar[0] lmasup=(ar[0] lmasinf+amp-(flon)dec),
    for(i=1;i<mas;i++)
      ar[i] lmasinf = ar[i-1] lmasinf + amp,
      ar[i] lmasup = ar[i] lmasinf + amp + dec,
    }
    ar[interv-1] lmasup=(flon)D,
    // límites reales y marca de clase
    for(i=0;i<mas;i++) {
      ar[i] lmasinf = ar[i] lmasinf - (dec/2),
      ar[i] lmasup = ar[i] lmasup + (dec/2),
      ar[i] % = (ar[i] lmasup+ar[i] lmasinf)/(flon)D,
    }
    // frecuencia absoluta
    for(i=0;i<mas;i++) ar[i] f=0, // inicialización
    for(i=0,i<mas,i++)
      for(j=0;j<=D;j++)
        ar[i] f += ar[j] lmasf && (mas(i)-<j) lmasap,
        ar[i] f = int(X) f + 1;
    // frecuencia normalizada y complementaria
    ar[0] f = ar[0] f, ar[0] % = 1,
    for(i=1;i<mas;i++)
      ar[i] f = ar[i-1] f + ar[i] f,
      ar[i] % = (ar[X] ar[i-1] f - ar[i] f),
    // frecuencia porcentual, relativa
    for(i=0,i<mas;i++){
      ar[i] % = (flon) ar[i] f / (flon)%,
      ar[i] f = (flon) ar[i] f * 100,
    }
  }
  // Lee las frecuencias de la tabla de distribución
  // en LERRFRECUENCIAS en a, sin comas, sin espacios
  int boy=ENTER,

```

Anexo D. Código fuente del sistema

```

while (numfin < n) {
  if (numact = numfin) {
    key = potencia1(col,ren,3);
    if (key != ENTER) an[numact] fi = leer(key, col,ren,3,0,200,INT);
  }
  else if (numact = numfin) {
    an[numact] fi = leer(key, col,ren,3,0,200,INT);
  }
  switch (key) {
    case ENTER: if (numact == numfin &&& numfin < n)
      numact++,
      numact++,
      ren++;
      break;
    case ESC: return ESC;
    case DOWN: if (numact < numfin) // flecha abajo
      numact++, ren--, |
      else (numact = 0, ren = 12);
      break;
    case UP: if (numact = 1) // flecha arriba
      numact--, ren--;
      else ( numact = n, ren = 12+n. )
      break;
    case END: numact = numfin,
      ren = 12+n,
      break;
    case HOME: numact = 0,
      ren = 12,
      break;
  }
  return key;
}

// Busca el dato menor y mayor del conjunto de datos
void BUSCAR(float *d, float *D, int n)
{
  int i;
  *d=num(0);
  *D=num(1);
  for (i=2; i<=n; i++)
    if (d[i]< *d) *D=*d;
    if (d[i]> *D) *D=d[i];
  }

// Cálculo el numero de intervalos
int INTERVALOS(int n, float *amp, float D, float d)
{
  float interval;
  float intervalo;
  intervalo = log10(n) * 3.322 + 1;
  interval = ceil(intervalo);
  *amp=(D-d)/interval;
  return (interval);
}

// Formatea un dato segun el número de decimales
void Formateado(int dec, float dato, char s[])
{
  switch (dec) {
    case 0: sprintf(s, "%i", dato); break;
    case 1: sprintf(s, "%1f", dato); break;
    case 2: sprintf(s, "%2f", dato); break;
    case 3: sprintf(s, "%3f", dato); break;
    case 4: sprintf(s, "%4f", dato); break;
  }
}

// PLANIFICADOR DE LAS GRAFICAS
void PLANIFICAR(int n, int interval)
{
  int i,tem,marca,ren;
  float factor_norm,dec;
  int espacado,x1,y1,x2,y2;
  char s[25];
  char color[12];
  int estilo[12];
  int dec;

  if (datos==1)
    dec = BUSCARDEC(n);
  else dec = num(1);
  switch (dec) {
    case 0: dir="0 5"; break;
    case 1: dir="0 0 5"; break;
    case 2: dir="0 0 0 5"; break;
    case 3: dir="0 0 0 0 5"; break;
  }

  if (EYES) i=-1;
  INICIALIZAR(i);
  randomize();

  // Busca las frecuencias mínima y máxima
  min=an[0] fi;
  for (i=1;i<=n; i, interval, i++) {
    if (an[i] fi < min) min = an[i] fi;
    if (an[i] fi > max) max = an[i] fi;
  }
  if (max==0), min=1;

  DETERMINAR(colora, interval);

  // Establece los estilos de las barras
  int estilo=1;
  for (i=0, i=interval, i++) {
    estilo[i] = 0;
    do {
      estilo = random(10)+2;
      for (int k=0; k<=i, k++)
        if (estilo[i] == estilo)
          estilo = 0;
        if (estilo)
          estilo[i] = estilo;
    } while (estilo[i] != 0);
  }

  CLASES (i,marca,dir,tem,ren);

  // Cálculo el factor de normalización y la escala
  factor_norm = 270 / (float)max;
  espacado = aD*(interval-2);

  TITULOS("HISTOGRAMA Y POLIGONO",14,10,4);
  BARRAS;
  autozalyts("TRIPLE_X_FONT,HORIZ_DIR,2);
  autozalyts("100,80,71");
  GRAFAR(100,100,130+espacado*(interval-2),390);

  // Dibuje las barras y líneas
  x1=130+espacado/2; y1=390;
  for (i=0, i=interval, i++) {
    width=ceil(estilo[i]*colores[i]);
    bar(130+((i-1)*espacado),x1,X189 - factor_norm*an[i] fi);
    130+((i-1)*espacado)*espacado, 390;
    x2=130+((i-1)*espacado)*espacado/2;
    y2=(int)(390 - factor_norm*an[i] fi);
    autozalyts(5);
    line(x1,y1-1,x2,y2-1);
    line(x1,y1-2,x2,y2-2);
    x1=x2; y1=y2;
  }
  line(x1,y1,350-(espacado/2),390);
  line(x1,y1-1,350-(espacado/2),390-1);
}

```

```

// Dibuja marcas horizontales y verticales
for(i=0, i=nterov, i++)
  line(130+(i*1)*espacado,385,130+(i+1)*espacado,395),
  for (j=1, j=marcas, j++)
    line(125,395+(factor_norm*min*j),125,395+(factor_norm*min*j+1)),
  dif = (arr[0] limsup - arr[0] liminf)/(float)2,
}

// Escribe los datos horizontales
settextstyle(SMALL, FONT, VERT, DIR_4),
formatdate(doc, arr[1] liminf, dif, a),
outtextxy(130+(espacado*2),5,403,a),
formatdate(doc, arr[1] limsup, dif, a),
outtextxy(130+(espacado*2),5,403,a),

for(i=0, i=nterov, i++)
  formatdate(doc, arr[1] limsup, a),
  outtextxy(130+(i+2)*espacado,5,403,a),
}

formatdate(doc, arr[nterov+1] limsup+dif, a),
outtextxy(550+(espacado*2), 403,a),
formatdate(doc, arr[nterov+1] limsup+dif, a),
outtextxy(550+(espacado*2), 403,a),

// Escribe los datos verticales
settextstyle(SMALL, FONT, HORIZ, DIR_4),
for (i=0, i=nterov, i++)
  sprintf(s, "%d", i*min),
  outtextxy(100,385+(i*factor_norm*min)a),
  lineadoe("027", 2,0),
  if (realtime("027") == F2)
    printimage(O, getmaxx()-1, getmaxy()-20),
    CLEARGRAPHICS(),
}

// Crea la grava, que es de frecuencia acumulada
void CHIVABAS(N) int nterov
{
  int i, tam, min, clase,
  float factor_norm,
  int espacado, x1, y1, x2, y2,
  char colores[12],

  if (i=BSK(1) + -1) {
    INIGRAFICS(),
    clase=0,
    tam=arr[nterov-1] fa,
    min=arr[0] fa,
    CLASIFIC(&clase, &tam, &min)

    // Calcula el factor de normalizacion y la escala
    factor_norm = ((float)20)/(float) tam,
    espacado = 4200/nterov+1,

    TITLE,OS("CHIVA - QUE",1,4,10,4),
    BSK(2),
    settextstyle(TRIPLEX_FONT, HORIZ, DIR_2),
    outtextxy(100,80,"fa"),
    TIT_ CHIV(espacado, nterov, clase, factor_norm, min),
    x1 = 130+espacado, y1=(int)(390-factor_norm*arr[0] fa),
    DETERCOLOR(colores, nterov),

    // Dibuja las líneas
    for(i=1, i=nterov, i++)
      x2=130+(i+1)*espacado,
      y2=(int)(390-factor_norm*arr[i] fa),
      outtextxy(colores[i],
      linef(x1, y1, x2, y2),
      linef(x1, y1+1, y2+1,
      x1=x2, y1=y2,
      lineadoe("027", 2,0),

    // Muestra y Datos Verticales
    settextstyle(SMALL, FONT, HORIZ, DIR_4),
    outtextxy(100,395,"f"),
    autoeof(15),
    for (i=1, i=nterov, i++) {
      sprintf(s, "%d", i*min),
      outtextxy(100,395+(factor_norm*min*j)a),
      line(125,395+(factor_norm*min*j),125,395+(factor_norm*min*j+1)),
    }

    if (realtime("027") + -12)
      printimage(O, getmaxx()-1, getmaxy()-20),
      CLEARGRAPHICS(),
    }

// Crea la grava, que es de frecuencia complementaria
void CHIVABAS(int nterov)
{
  int i, tam, min, clase, color,
  float factor_norm,
  int espacado, x1, y1, x2, y2,
  char colores[12],

  if (i=BSK(1) + -1) {
    INIGRAFICS(),
    clase=0,
    tam=arr[0] fa,
    min=arr[nterov-1] fa,

    // Determina el numero de marcas verticales
    CLASIFIC(&clase, &tam, &min)

    // Calcula el factor de normalizacion y la escala
    factor_norm = ((float)20)/(float) tam,
    espacado = 4200/nterov+1,

    TITLE,OS("CHIVA + QUE",1,4,10,4),
    BSK(2),
    settextstyle(TRIPLEX_FONT, HORIZ, DIR_2),
    outtextxy(100,80,"fa"),
    TIT_ CHIV(espacado, nterov, clase, factor_norm, min),
    x1 = 130+espacado, y1=(int)(390-factor_norm*arr[0] fa),
    DETERCOLOR(colores, nterov),

    // Dibuja las líneas
    for(i=1, i=nterov, i++)
      x2=130+(i+1)*espacado,
      y2=(int)(390-factor_norm*arr[i] fa),
      outtextxy(colores[i],
      linef(x1, y1, x2, y2),
      linef(x1, y1+1, y2+1),
      x1=x2, y1=y2,
      lineadoe("027", 2,0),
      if (realtime("027") == F2)
        printimage(O, getmaxx()-1, getmaxy()-20),
        CLEARGRAPHICS(),
    }

// Establece las marcas y datos verticales y horizontales de las gravas
void TIT_ CHIV(int espacado, int nterov, int clase, float factor_norm, int min)
{
  int i, color,
  char c[20],

  int n = BUSCAN(),
  if (clase==1)
    dec = BUSCADEBC(n),
    else dec = num[3],
  GRAF(130,100,130+espacado*(nterov+1),390),

  // Muestra horizontales
  for(i=0, i=nterov, i++)
    line(130+(i+1)*espacado,385,130+(i+1)*espacado,395),

  // Muestra y Datos Verticales
  settextstyle(SMALL, FONT, HORIZ, DIR_4),
  outtextxy(100,395,"f"),
  autoeof(15),
  for (i=1, i=nterov, i++) {
    sprintf(s, "%d", i*min),
    outtextxy(100,395+(factor_norm*min*j)a),
    line(125,395+(factor_norm*min*j),125,395+(factor_norm*min*j+1)),
  }
}

```


Anexo D: Código fuente del sistema

```

// Datos horizontales
setstyle(BMALL_FONT, VERT_DIR,4),
tot(*t, i, unvar, i++)
for(m=1; m<=n; m++) {
  sprinf("%2f", arf[i] / lenvarf),
  outstrxy(130*(i+1))"espaciado=5,403,s),
}

// Gráficas los ejes x e y de la gráfica
void GRAF(un xl, un y1, un x2, un y2)
{
  len(x1,y1,x1,y2),
  len(x1,y2,x2,y2),
  len(x1,y1,x1+y1),
  len(x1+y1,x1,y1-y1),
  len(x1+y1,x1,y1+y1),
  len(x2,y2,x2+y2,y2),
  len(x2+y2,x2,y2-y2),
  len(x2+y2,x2,y2+y2),
}

// Determina el número y rango de intervalos
// verticales y horizontales de las gráficas
void CLASES(un *clase, un *tam, un *mun)
{
  float cla,
  *clase="array("mun),
  if (*clase*"mun) < *mun) {
    *clase+=1,
    *tam="clase("mun),
    cla = ((float)*clase,
    while(cla < 10) {
      cla=cla*2,
      *tam="mun*2,
      cla=cla,
      *clase="clase,
      *tam="clase("mun),
    }
}

// Determina los colores de las barras y líneas de las gráficas
void DEYNHCOLOR(char colores[], un unvar)
{
  un color=1,
  for (un i=0, i<=unvar, i++) {
    colores[i] = 0,
    do {
      color = random(14)+1,
      while (color==0 || color == 6),
      for (un k=0, k < 14, k++)
        if (color%k == color)
          color = 0,
          if (color)
            colores[k] = color,
            while (color%k != 0),
    }
}

// FUNCIONES DE LAS MEDIDAS DE TENDENCIA CENTRAL

// Medida aritmética para datos sueltos muestrales
void MEDIAARIT_3 (un n)
{
  float media, suma=0,
  char s(40),
}

// Cálculos
for (un i=0, i<=n, i++)
  suma += unvar[i],
  media = suma/n,

// Impresión de resultados
INFORMACION(),
MARCO(),
}

```

```

//medias("0,3),
TITULO("MEDIA ARITMETICA DE LA POBLACION ",14,20,3),
setcolor(15),
outstrxy(50,60,"FORMULA "),
outstrxy(50,60,"DESARROLLO "),
suma(145,105,1),
setstyle(BMALL_FONT,HORIZ_DIR,5),
outstrxy(100,120,"n = -----"),
outstrxy(160,140,"787"),
outstrxy(50,180,"Medida "),
outstrxy(50,220,"n = medida de la población "),
len(50,243,55,243),
len(50,253,55,253),
outstrxy(50,280,"n = número de valores de"),
outstrxy(50,260,"n = total de las observaciones"),
outstrxy(50,280,"N = número de elementos en"),
outstrxy(50,300,"la población"),
sprinf("%2f", 3F, suma),
un largo = lastword(s),
outstrxy(450,100,s),
outstrxy(410,120,"n ="),
len(440,128,440+larg*20,128),
sprinf("%2f", 3F, n), outstrxy(450+(larg*lastword(s))/2,140,s),
sprinf("%2f", 3F, media),
outstrxy(340,180,s),
return("02"),
CIERRAGRAFICOS(),
}

//Medida aritmética para datos sueltos muestrales
void MEDIAARIT_4 (un n)
{
  float media, suma=0,
  char s(40),
}

// Cálculos
for (un i=0, i<=n, i++)
  suma += unvar[i],
  media = suma/n,

//Impresión de resultados
INFORMACION(),
MARCO(),
}

//medias("0,3),
TITULO("MEDIA ARITMETICA MUESTRAL ",14,20,3),
setcolor(15),
outstrxy(50,60,"FORMULA "),
outstrxy(50,60,"DESARROLLO "),
suma(145,105),
outstrxy(170,105,"n"),
outstrxy(160,140,"n = -----"),
len(100,143,105,143),
outstrxy(160,180,"7"),
outstrxy(50,220,"Medida "),
outstrxy(50,240,"n = medida de la muestra"),
len(50,243,55,243),
len(50,243,55,243),
len(50,263,55,263),
len(50,273,55,273),
outstrxy(50,280,"n = suma de valores de todos"),
outstrxy(50,280,"las observaciones"),
outstrxy(50,280,"n = número de elementos de"),
outstrxy(50,300,"la muestra"),
sprinf("%2f", 3F, suma),
outstrxy(410,120,"n ="),
un largo = lastword(s),
len(440,128,440+larg*20,128),
sprinf("%2f", 3F, n), outstrxy(450+(larg*lastword(s))/2,140,s),
sprinf("%2f", 3F, media),
outstrxy(340,180,s),
return("02"),
CIERRAGRAFICOS(),
}

```

Anexo D: Código fuente del sistema

```

//Mediana aritm. para datos agrupados
void MEDIANARIT (int n, int interv)
{
    int i;
    float media, suma;
    char s[35];

    // Cálculo
    for(i=0; suma=0, i<interv, i++) suma+=arr[i]*Xi+arr[i]*fi;
    media = suma/n;

    // Impresión de resultados
    INICIARPROC(S);
    MARCOCK;
    lreswarr("O", 2);
    TTULOCS("MEDIANA ARITMETICA MUESTRAL", 14, 20, 2);
    asterisk(15);
    outstrarr(50, 60, "FORMULA ");
    outstrarr(340, 60, "TESARRCULO ");
    suma(135, 119);
    outstrarr(100, 110, "f(x)P");
    outstrarr(100, 140, "x -----");
    lres(100, 143, 103, 143);
    outstrarr(100, 160, "n");
    outstrarr(50, 204, "n de elementos");
    outstrarr(70, 240, "% = media muestral ");
    lres(70, 243, 75, 243);
    lres(70, 263, 75, 263);
    lres(70, 273, 75, 273);
    outstrarr(70, 284, "% = La suma de ");
    outstrarr(70, 280, "fi = frecuencia absoluta en");
    outstrarr(100, 300, "cada clase ");
    outstrarr(70, 320, "% = suma de clase de cada");
    outstrarr(100, 340, "clase muestral ");
    outstrarr(70, 360, "% = numero de observaciones");
    outstrarr(100, 380, "en la muestra ");
    aprnt(s, "% 3P, suma);
    int larg = testwddh(s);
    outstrarr(450, 100, s);
    outstrarr(910, 120, "x");
    lres(410, 123, 415, 123);
    lres(440, 128, 440, 128);
    aprnt(s, "% 4P, n);
    outstrarr(450+larg, 100, s);
    outstrarr(70, 280, "fi = frecuencia de cada clase");
    outstrarr(340, 80, "n");
    realms("O");
    CERRARGRAFICOS;
}

//Mediana para datos sueltos
void MEDIANAS(n, int n)
{
    int temp;
    float a;
    char s[40];

    a=(n+1)/2;
    INICIARPROC(S);
    MARCOCK;
    lreswarr("O", 2);
    TTULOCS("MEDIANA PARA DATOS NO AGRUPADOS", 14, 20, 2);
    asterisk(15);
    outstrarr(50, 80, "FORMULA ");
    outstrarr(150, 130, "n-1");
    outstrarr(70, 140, "n de elementos = ----- elemento");
    outstrarr(100, 150, "2");
    outstrarr(70, 160, "n un arreglo de datos ");
    outstrarr(50, 220, "Dónde ");
    outstrarr(70, 204, "n = numero de elementos del");
    outstrarr(100, 280, "arreglo ");
    ORDENAR(n);
    temp="";
    DESP_LINTECOS(n);
    if (temp=="")
        aprnt(s, "La mediana es % 2P, num(a-1));
    else
        aprnt(s, "La mediana es % 2P, (num(a-1)+num(a))/2);
}

outstrarr(50, 130, a);
lres(1);
realms("O");
CERRARGRAFICOS(S);
}

//Mediana para datos agrupados
void MEDIANAS(n, float arrp, int interv)
{
    int i, el;
    float media, suma=0;
    char s[45];

    a=n/2;
    //Encuase la clase modal
    for (int i=0; i< interv, i++)
        sum += arr[i]*fi;
    if (sum == a)
        el = i;
        break;
}

// Suma de las frecuencias hasta antes de la clase modal
suma = 0;
for (int i=0; i< i-1; i++) sum += arr[i]*fi;

media=(a-(sum+1)*arr[i]*fi) / arrp + arr[i]*larrf;

// Impresión de resultados
INICIARPROC(S);
MARCOCK;
lreswarr("O", 2);
TTULOCS("MEDIANA", 14, 20, 2);
asterisk(15);
outstrarr(50, 60, "FORMULA ");
outstrarr(340, 60, "TESARRCULO ");
outstrarr(137, 60, "n");
outstrarr(137, 110, "2");
lres(130, 108, 156, 108);
outstrarr(70, 100, "F = 1");
outstrarr(90, 120, "x ----- w*Lm");
outstrarr(170, 135, "m");
outstrarr(50, 160, "Dónde ");
outstrarr(50, 200, "m = mediana de la muestra ");
outstrarr(50, 218, "n = numero total de elementos");
outstrarr(80, 230, "de la distribución ");
outstrarr(50, 254, "F = suma de las frecuencias");
outstrarr(80, 272, "de clase hasta la mediana");
outstrarr(80, 290, "pero sin incluirla");
outstrarr(50, 308, "Fm = frecuencia de la mediana de");
outstrarr(80, 326, "clase ");
outstrarr(50, 344, "w = amplitud de intervalo de las");
outstrarr(80, 362, "clases ");
outstrarr(50, 380, "Lm = límite inferior del intervalo");
outstrarr(80, 398, "de mediana de clase ");
aprnt(s, "% 4P, n);
outstrarr(377, 90, s);
int larg1 = testwddh(s);
lres(372, 108, 375, 108);
outstrarr(377+larg1/2, 5, 310, "2");

aprnt(s, "% 3P, suma);
int larg2 = testwddh(s);
aprnt(s, "m = ");
lres(370, 120, 370, 120);
aprnt(s, "% 2P + % 3P, arrp, arr[i]*larrf);
outstrarr(370+larg1+larg2, 5, 120, a);
outstrarr(370+(larg1+larg2), testwddh(s)+2, 135, a);
aprnt(s, "La mediana de los datos es: % 2P, mediana);
realms("O");
CERRARGRAFICOS(S);
}

```

Anexo D: Código fuente del sistema

```

// Muestra para datos sueltos
void M3DIA(int n)
{
    int i;
    float max1, max2,
    float ac1, Mo1, Me2,
    char s[40];

    // Cálculos
    ORDINA(n);
    act = max[0];
    max1 = max[1] + Mo1 + Me2 + 1,
    i = 0;
    while (i < n) {
        cont = 0;
        while (sum[i] == max1) { i++; cont++; }
        if (cont == max1)
            if (i % 2 == 1) {
                max2 = max1;
                Mo1 = cont;
                Me1 = max[i+1];
            } else {
                max1 = max;
                Mo1 = max[i+1];
            }
        act = max[i];
    }

    // Impresión de resultados
    INGRAPICOSO;
    MARCOGO;
    limando("0.2");
    TITULO3("MODA PARA DATOS NO AGRUPADOS", 14, 30, 2);
    intitol(15);
    outentroy( 80, 100, "Uso limitado de la moda en", );
    outentroy( 80, 120, "datos no agrupados.");
    DISP_DATOS(n);
    if (max1 == max2)
        printf("La moda por consenso es: %2f y %2f", Mo1, Me1);
    else
        printf("La moda por consenso es: %2f", Mo1);
    outentroy(340, 200, a);
    lim("02");
    CIERRAGRAFICOSO;
}

// Muestra para datos agrupados
void MODAIA(int n, int w)
{
    int d1, d2,
    float h1o,
    char s[45];

    // Datos la clase modal
    for (int i = 0; i < n; i++)
        if (ar[i] != 0) {
            el = i;
            mayor = ar[i] * fi;
            if (el == 1) d1 = ar[el] * fi - ar[el-1] * fi;
            else d1 = 0;
            if (el < n-1) d2 = ar[el] * fi - ar[el+1] * fi;
            else d2 = ar[el] * fi;
            h1o = (ar[el] * limanf) / (((float) ar[el] * (d1 + d2)) * (ar[el] * limanf + ar[el+1] * limanf));
        }

    // Impresión de fórmula y resultados
    INGRAPICOSO;
    MARCOGO;
    limando("0.2");
    TITULO3("MODA PARA DATOS AGRUPADOS", 14, 30, 2);
    intitol(15);
    outentroy( 30, 60, "FÓRMULA ");
    outentroy( 340, 60, "DESARROLLO ");
    outentroy( 195, 105, "d1 ");
    outentroy( 190, 120, "Mo = L + (mayor * w)");

```

```

outentroy( 150, 135, "Mo d1 + d2");
outentroy( 50, 160, "Donde ");
outentroy( 50, 200, "L Mo = límite inferior de la clase");
outentroy( 85, 218, "clase modal ");
outentroy( 50, 230, "d1 = frecuencia de la clase modal");
outentroy( 85, 234, "suma de la frecuencia de la ");
outentroy( 85, 272, "clase que se encuentra antes.");
outentroy( 85, 290, "suma de las frecuencias de la clase");
outentroy( 50, 308, "d2 = frecuencia de la clase modal");
outentroy( 85, 326, "suma de las frecuencias de la clase");
outentroy( 85, 344, "que se haya inmediatamente");
outentroy( 85, 362, "después de ella.");
outentroy( 50, 380, "w = amplitud del intervalo de la");
outentroy( 85, 398, "clase modal ");
printf("Mo = %2f", ar[el] * limanf);
int largo = strlen(s);
outentroy(350, 120, a);
printf("Mo = %2f", ar[el] * d1, d2);
int largo2 = strlen(s) + 20;
lim("350+largo, 120, 350+largo+largo2, 120");
outentroy(360+largo, 135, a);
printf("Mo = %2f", ar[el] * d1);
outentroy(350+largo+(largo2+strlen(s)) * 2, 105, a);
printf("Mo = %2f", ar[el] * limanf + ar[el] * limanf);
outentroy(360+largo+largo2+10, 120, a);
printf("La moda muestral es %2f", Mo);
outentroy(340, 200, a);
return("02");
CIERRAGRAFICOSO;
}

```

/// FUNCIONES DE DISPERSION

```

// Diagrama de un conjunto de datos
void DISPERSION(int n)
{
    float d, d;
    char s[40];

```

```

    BUSCAR(d, d, d, n);
    INGRAPICOSO;
    MARCOGO;
    limando("0.2");
    TITULO3("DISPERSION DE LOS DATOS", 14, 30, 2);
    intitol(15);
    outentroy( 50, 60, "FÓRMULA ");
    outentroy(340, 60, "DESARROLLO ");
    outentroy(100, 100, "Amplitud Real = D - d");
    outentroy( 50, 130, "Donde ");
    outentroy( 70, 160, "D = Valor del dato mayor.");
    outentroy( 70, 180, "d = Valor del dato menor.");
    printf("D = %2f", d);
    printf("d = %2f", d);
    outentroy(340, 100, a);
    printf("Amplitud Real = %2f", D - d);
    outentroy(340, 180, a);
    return("02");
    CIERRAGRAFICOSO;
}

```

```

// Varianza para datos agrupados
void VARIANZAIA(int n, int w, float medio)
{
    int i;
    float suma2 = 0, varianza = 0;
    char s[45];

```

```

// Cálculos
for (int i = 0; i < n; i++)
    suma2 += (ar[i] - medio) * (ar[i] - medio) * ar[i] * fi;
varianza = suma2 / (n - 1);

// Impresión de la fórmula y resultados
INGRAPICOSO;
MARCOGO;
limando("0.2");
TITULO3("VARIANZA MUESTRAL", 14, 30, 2);

```

Anexo D: Código fuente del sistema

```

sumas(n)(15),
outstatoy(50,60,"FORMULA"),
outstatoy(140,60,"DESARROLLO"),
sumas(140,90),
outstatoy(170,100,"(xi - a)2"),
outstatoy(100,120,"S2"),
outstatoy(170,140,"n-1"),
outstatoy(50,160,"Dmedia"),
outstatoy(50,200,"S2"= Varianza de la muestra"),
outstatoy(50,220,"Xo"= media de las clases"),
outstatoy(50,240,"N"= media de la muestra"),
outstatoy(50,260,"n"= numero total de elementos"),
outstatoy(50,280,"de la muestra"),
outstatoy(50,300,"n"= frecuencia de cada clase"),
outstatoy(50,320,"n"= suma de todos los valores),
sumas(50,320),
outstatoy(80,340,"(Xi - a)3"),
aproxif("a"= 2o,sumatoria),
outstatoy(140,120,"Sy ="),
linea(180,128,"800"=largor=20,128),
aproxif("a"=2o,1),
outstatoy(100,"(largor-textwidth(s))/2,140,a),
aproxif("La varianza muestral es %3 2o,varianza),
outstatoy(140,180,a),
realineo("02"),
CERRARGRAFICOS),
}

// Varianza poblacional
void VARIANZAJ(int n, float media)
{
float sumatoria=0,
float varianza=0,
char s[45];

// Cálculos
for (int i=0; i < n; i++)
sumatoria += ((arr[i]-media)*(arr[i]-media)),
varianza = sumatoria/n,

// Impresión de resultados
INICIARFIN(S),
MARCOS(),
lineado("0",2),
TITULO("VARIANZA POBLACIONAL",14,20,2),
outstatoy(50,60,"FORMULA"),
outstatoy(140,60,"DESARROLLO"),
sumas(140,90),
outstatoy(168,100,"(a - m)2"),
outstatoy(100,120,"y = -----"),
linea(100,127,110,127),
outstatoy(185,140,"N"),
outstatoy(50,180,"Dmedia"),
outstatoy(50,220,"S2"= Varianza de la población"),
linea(56,227,60,227),
outstatoy(50,240,"n"= el elemento u observación"),
outstatoy(50,260,"a"= media de la población"),
outstatoy(50,280,"N"= numero total de elementos"),
outstatoy(80,300,"de la población"),
outstatoy(50,320,"n"= suma de todos los valores),
sumas(50,320),
outstatoy(80,340,"(a - m)3"),
aproxif("a"= 2o,sumatoria),
outstatoy(100,160,a),
int largor = textwidth(s),
outstatoy(140,120,"Sy ="),
linea(145,127,136,127),
linea(180,127,180"=largor=20,127),
aproxif("a"=2o,n),
outstatoy(100,"(largor-textwidth(s))/2,140,a),
aproxif("La varianza poblacional es %3 2o,varianza),
outstatoy(140,180,a),
realineo("02"),
}

// Desviación estándar para datos agrupados
void DESVIA_EST(int n,float media, float error)
{
float desviand, varianza,
char s[45],
float sumatoria=0,

// Cálculos
for (int i=0; i < n; i++)
sumatoria += ((arr[i]-media)*(arr[i]-media)*arr[i]),
varianza = sumatoria/(n-1),
desviand = sqrt(varianza),

// Impresión de resultados
INICIARFIN(S),
MARCOS(),
lineado("0",2),
TITULO("DESVIACION ESTANDAR MUESTRAL",14,20,2),
sumas(50,60,"FORMULA"),
outstatoy(140,60,"DESARROLLO"),
outstatoy(143,108,"(Xi - a)3"),
sumas(143,108),
linea(203,110,208,110),
}

```

Anexo D: Código fuente del sistema

```

nuestro(100,120,"S = -----"),
nuestro(170,132,"n = 17"),
raiz(130,147,240,105),
nuestro(50,180,"Dónde "),
nuestro(50,220,"S = desviación estándar muestral "),
nuestro(50,240,"n = frecuencia de cada clase "),
nuestro(50,260,"Xi = marca de clase de las clases"),
nuestro(50,280,"n = la media de la muestra "),
largo(50,281,55,283),
nuestro(50,300,"n = el número total de elementos"),
nuestro(50,320,"S = la muestra "),
nuestro(50,340," = la suma de todos los valores"),
sumato(50,340),
nuestro(50,360,"(Xi - a)2"),
largo(118,365,123,365),
apuntar(" % 2F, varianza^n"), nuestro(390,108,a),
mi largo = tamaño(a),
nuestro(340,120,"S = "),
raiz(170,147,370,"largo=35,105),
largo(180,127,380,"largo=20,127),
apuntar(" %e^n-1), nuestro(390,"(largo-tamaño(a))/2,132,a),
apuntar(" Desviación estándar muestral % 2F,deviacion"),
nuestro(340,180,a),
realismo("02"),
CIERRAORAFICOSO,
}

//Desviación estándar muestral para datos sencillos
void DRSVIA_ESTM4(int n,float media)
{
float devianst, varianza,
char s[50];

// Cálculo
varianza=MOD_VARIANZA(n,media);
devianst = sqrt(varianza);

//Impresión de resultados
INICRAFICOSO,
MARCOOQ,
titulos("DESVIACION ESTANDAR MUESTRAL",14,20,2),
astrolot(13),
nuestro(50,60,"FORMULA ");
nuestro(340,60,"DESARROLLO ");
nuestro(143,108," * a - a ^ 2");
sumato(143,108),
largo(200,110,205,110),
nuestro(100,120,"S = -----"),
nuestro(170,132,"n = 17"),
raiz(130,147,230,105),
nuestro(50,180,"Dónde "),
nuestro(50,220,"S = la observación "),
nuestro(50,240,"n = la media de la muestra "),
largo(50,281,55,283),
nuestro(50,300,"n = el número total de elementos"),
nuestro(50,320,"S = la muestra "),
nuestro(50,340," = la suma de todos los valores"),
sumato(50,300),
nuestro(50,360,"(ai - a)2"),
largo(118,365,123,365),
apuntar(" % 2F, varianza^n"), nuestro(390,108,a),
mi largo = tamaño(a),
nuestro(340,120,"S = "),
raiz(170,147,370,"largo=35,105),
largo(180,127,380,"largo=20,127),
apuntar(" %e^n-1), nuestro(390,"(largo-tamaño(a))/2,132,a),
apuntar(" Desviación estándar muestral % 2F,deviacion"),
nuestro(340,180,a),
realismo("02"),
CIERRAORAFICOSO,
}

// Coeficiente de variación para datos agrupados
void COEFICIENTM4(int n, float media, float astrol)
{
float c_v,devianst, devianst,
char s[50];

//Cálculo
if (media == 1)
varianza=MOD_VARIANZA(n,media);
else {
float sumator=0;
for (int i=0; i<n; i++)
sumator+= (ast[i]*XG-media)*(ast[i]*XG-media)/ast[i];
varianza = sumator/n-1;
devianst = sqrt(varianza);
c_v = (devianst / media) * 100;
}

//Impresión de resultados
INICRAFICOSO,
MARCOOQ,
titulos("COEFICIENTE DE VARIACION MUESTRAL",14,30,2),
astrolot(15),
nuestro(50,60,"FORMULA ");
nuestro(340,60,"DESARROLLO ");
nuestro(160,100,"S"),
}

```

Anexo D. Código fuente del sistema

```

outstextoy(100,120,"c v = ----- x (100)",
outstextoy(100,140,"c"),
line(100,143,165,143),
outstextoy(50,180,"Dónde"),
outstextoy(50,220,"c v = coeficiente de variación"),
outstextoy(50,240,"s = desviación estándar"),
outstextoy(50,260,"x = media de la desviación"),
line(65,263,70,263),
aprentf(" %2f",desviana),
outstextoy(195,100,s),
int largo = lastrow(dif),
outstextoy(140,120,"c v ="),
line(185,128,185+largo+20,128),
outstextoy(185+largo+30,120," x 100"),
aprentf(" %2f",desviana),
outstextoy(140,180,"El coeficiente de variación"),
aprentf(" %s 3f %s",c,v),
outstextoy(140,200,s),
resumen("02"),
CIERRAGRAFICOS(),
)

// Coeficiente de variación poblacional
void COEFICIENTEVB(int n, float media)
{
float e, v, varianza, desviacion;
char s[50];

//Calculo
varianza=MOD_VARIANZA(n,media),
desviacion = sqrt(varianza),
e, v = (desviacion / media) * 100,

//Impresión de resultados
INICIAGRAFICOS(),
MARCO(),
limpedos("2"),
TITULOS("COEFICIENTE DE VARIACION POBLACIONAL",14,30,2),
outstextoy(50,60,"FORMULA"),
outstextoy(140,60,"1/S.E.SARRDILLO"),
outstextoy(165,110,"7"),
line(165,112,170,112),
outstextoy(100,120,"c v = --- x 100"),
outstextoy(100,135,"c v"),
outstextoy(50,180,"Dónde"),
outstextoy(50,220,"c v = coeficiente de variación"),
outstextoy(50,240,"s = desviación estándar"),
line(64,247,69,247),
outstextoy(50,260,"x = media de la desviación"),
aprentf(" %2f",desviana), outstextoy(190,100,s),
int largo = lastrow(dif),
outstextoy(140,120,"c v ="),
line(180,128,180+largo+20,128),
outstextoy(180+largo+30,120," x 100"),
aprentf(" %2f",desviana),
outstextoy(140,180,"El coeficiente de variación"),
aprentf(" %s 3f %s",c,v),
outstextoy(140,200,s),
resumen("02"),
CIERRAGRAFICOS(),
)

// Crea un conjunto de datos
void ORDENAR(int n)
{
int prt, k;
float temp=0;

for (k=0, h=n, k++)
for (prt=0, prt<=h-2; prt++){
if (num[prt]>num[prt+1]) {
temp=num[prt];
num[prt]=num[prt+1];
num[prt+1]=temp; }
}
}

// Calcula la media
void MEDIA(float *media, int n)
{
int i;
float suma=0;

for (i=1; i <= n; i++)
suma +=num[i];
*media = suma/n;
}

// BUSCA[M] // n esta en la temperatura posecion
int i;
// tipo de datos en la presunta
for (i=-MAX-1; i <= 0; i++)
if (num[i]!=0)
return (num[i]+2);
return 0;
}

// Regresa 1 para datos leídos, 2 si no tabla
int TIPODATOS()
{
for (int i=-MAX-1; i <= 0; num[i]!=-0; i++)
return num[i];
}

// Busca el número máximo de decimales empleados
// en la lectura de los datos
int DECIMALES(int n)
{
char s[20];
int max=0;

for (int i=0; i <= n; i++)
int dec = 3;
aprentf(" %2f",num[i]);
for (int j=num[dec]; s[j]!='.'; j++)
if (s[j]!='0') dec--;
if (dec <= max) max = dec;
}
return max;
}

// Determina la diferencia entre el límite real superior
// de una clase y el límite real inferior de la siguiente
// según el número de decimales empleados
float DIP(int dec)
{
float dif;
switch (dec) {
case 0: dif = 1; break;
case 1: dif = .1; break;
case 2: dif = .01; break;
case 3: dif = .001;
}
return dif;
}

//Calcula la varianza
float MOD_VARIANZA(int n, float media)
{
float varianza;
int i;
varianza =0;
for (i=0; i <= n; i++)
varianza +=(num[i]-media)*(num[i]-media);
if (temp==1)
varianza = varianza/n;
else varianza =varianza/(n-1);
return (varianza);
}
}

```


Anexo D. Código fuente del sistema

```

}
} while (key != ESC);
return 0;
}

void CASO2(float moda,float dev)
{
char s[50];
int key;
float valor1=valor2=0,s1,s2,v1;

para (impres) {
getoxy(30,2);cpromf("DISTRIBUCION NORMAL");
getoxy(11,5);cpromf("La m (moda) es = %d",moda);
getoxy(11,8);cpromf("Ingresa entre que valores deseas ?");
getoxy(1,10);cpromf("Valor 1 = ");
if (leer(s;valor1);1.1,116,9,13200,FLOAT) == ESC) return;
getoxy(1,12);cpromf("Valor 2 = ");
if (leer(s;valor2);1.1,126,9,13200,FLOAT) == ESC) return;
unscarcba(s;valor1;valor2);
s1=(valor1+moda)/dev;
s2=(valor2+moda)/dev;

if (s1<-3.0 || s1 >= 3.0 || s2<-3.0 || s2 >= 3.0) {
FOR_NORMAL(s);
cpromf(s,"%d %d %d",valor1,moda);
int la = textwidth(s);
outscxy(430,100,s);
intscxy(430,110,s1);
intscxy(430,120,s2);intscxy(430,130,s);
cpromf(s,"%d",dev);
outscxy(430,(la-textwidth(s))/2,122,s);
outscxy(340,160,"El numero de desviaciones estándar");
cpromf(s,"para %d 2f es %d 2f",valor1,s1);
outscxy(340,170,s);
cpromf(s,"%d %d %d",valor2,moda);
la = textwidth(s);
outscxy(430,230,s);
outscxy(390,240,"s2 = ");
intscxy(430,248,s2);intscxy(430,258,s);
cpromf(s,"%d",dev);
outscxy(430,(la-textwidth(s))/2,250,s);
outscxy(340,260,"El numero de desviaciones estándar");
cpromf(s,"de %d 2f es %d 2f",valor2,s2);
outscxy(340,268,s);
MENSJE(s);
} else {
INGRAFICOS);
Gráfica G(modas,dev,3);
float v1=Z_buscamos(s1), /*no puede desplazar a la vez la tabla
(float v2=Z_buscamos(s2);
if (v1 >= 0.0 && v2 >= 0.0)
v1=v2=0.1;
else if (v1 < 0.0 && v2 < 0)
v1=v1+0.2;
else
v1=v1+0.2;
int key=0;
do {
scribidos(modas,dev,valor1,valor2);
cpromf(s,v1=" %d",v1); outscxy(460,220,s);
G_dibuja(s);
G_buscamos(valor1,valor2);
G_pondetacromas(valor1,valor2);
G_puntavercos(valor1,valor2);
intscxy(078,2,0);
do {
if ((key=vermes("078?"))==F3)
Z_desplazabla(s);
if (key == F2)
pruntavercos(0,0,getmax(s)-1);getmax(s)-20);
} while (key == ESC);
} while (key != ENTER && key != ESC);
}
}

CERRARGRAFICOS);
prin dibujo(s);
}

void probabilidad(float z, float vt, float valor, float moda, int caso)
{
char s[50];
cpromf(s,v1=" %d",vt); outscxy(460,220,s);
switch (caso) {
case 1 if (z < 0) vt=0.5-vt;
else vt=0.5+vt;
cpromf(s,v1=" %d",vt); outscxy(460,220,s);
break;
case 3 if (z < 0) vt=0.5-vt;
else vt=0.5+vt;
cpromf(s,v1=" %d",vt); outscxy(460,220,s);
break;
case 4 if (z < 0) vt=0.5-vt;
else vt=0.5+vt;
cpromf(s,v1=" %d",vt); outscxy(460,220,s);
}
} outscxy(460,250,s);
}

void CASO3(float moda, float dev, int caso)
{
float z,valor=0,v1;
int key;
char s[50];

para (impres) {
getoxy(30,2);cpromf("DISTRIBUCION NORMAL");
getoxy(11,5);cpromf("La m (moda) es = %d",moda);
getoxy(11,8);cpromf("Ingresa entre que valores deseas ?");
getoxy(1,10);cpromf("Valor mayor = ");
break;
case 3 cpromf("Valor mayor que "); break;
case 4 cpromf("Valor mayor que "); break;
}
if (key = leer(s;valor);1.8,6,0,3200,FLOAT) == ESC) return;
v1=(valor+moda)/dev;
if (z < 0 || z >= 3.0) { //3
FOR_NORMAL(s);
cpromf(s,"%d %d %d",valor,moda);
int la = textwidth(s);
outscxy(430,100,s);
outscxy(390,110,"s = ");
intscxy(430,110,s1);intscxy(430,120,s2);intscxy(430,(la-textwidth(s))/2,122,s);
cpromf(s,"%d",dev);
outscxy(390,162,s);
outscxy(340,200,"El numero de desviaciones estándar");
cpromf(s,"de %d 2f es %d 2f",valor,s);
outscxy(340,210,s);
MENSJE(s);
CERRARGRAFICOS);
} else {
INGRAFICOS);
Gráfica G(modas,dev,3);
v1=Z_buscamos(s);
int key=0;
do {
scribidos(modas,dev,valor,z);
probabilidad(z,v1,valor,moda,caso);
G_dibuja(s);
switch (caso) {
case 1 G_buscamos(modas+dev*3), valor); break;
case 3 G_buscamos(valor,modas+dev*3); break;
case 4 G_buscamos(valor,moda); break;
}
G_puntavercos(valor,moda);
G_puntavercos(valor,moda);
}
}
}
}

```


PROGRAMAS DE ESTADISTICA II

```

outmatxy(50.292,"p = probabilidad de éxito ");
outmatxy(50.312,"q = probabilidad de fracaso ");
setmatxy(SMALL, FONT, NORIZ, DIR, 4);
sprmatf("P= %d, Q= %d", p, q);
int lon1 = int(strlen(s));
sprmatf(" %d", (lon1-1)*2);
lon2 = int(strlen(s));
SIFACTORIAL(340-1+lon1+lon2+3,165);
SIFACTORIAL(340-1+lon1+strlen(s)+3,165);
lon1 = (340-1+lon1, 157, 340-1+lon1+lon2, 157);
sprmatf(" %d", p);
outmatxy(340+1+lon1+(lon2+strlen(s))*2, 141, s);
SIFACTORIAL(340-1+lon1+(lon2+strlen(s))*2+strlen(s)+3,141);
lon1 = lon1+lon2+5;
sprmatf(" = %d", 2F, p);
lon1 = int(strlen(s));
sprmatf(" %d", s);
lon1 = int(strlen(s));
sprmatf(" = %d", 2F, q);
lon1 = int(strlen(s));
sprmatf(" %d", (lon1+lon2)*2);
outmatxy(340,260,4);
outmatxy(SMALL, FONT, NORIZ, DIR, 5);
sprmatf("La probabilidad de %d, es %d", p);
sprmatf("si se repite n = %d", n, probabilidad);
outmatxy(340, 260, 4);
realms("O");
CIERRAGRAFICOS);
}

// Desarrollo de la distribución binomial como aproximación a la distribución Poisson
void BINOMAPROX int r, int n, float p, double resultado)
{
int a;
float pp;
double p;
char s[60];
a = 1;
pp = pow(p,n);
p = (pp+(1-p))**exp(-n)/resultado;
INICIAGRAFICOS;
MAZCOGCO;
lonms("O");
TITULOS("DISTRIBUCION DE POISSON COMO APROXIMACION",14,5,3);
TITULOS("A LA DISTRIBUCION BINOMIAL",14,30,3);
int lon1;
outmatxy(50,80,"FORMULA ");
outmatxy(50,80,"DESARROLLO");
outmatxy(100,120," a = n * p ");
outmatxy(100,130," (pp) = p^n ");
outmatxy(100,140," resultado = (n! / (n-a)! * a^a) * pp ");
outmatxy(100,150," a = n ");
SIFACTORIAL(195,152);
outmatxy(50,180,"Ejemplo");
outmatxy(50,230,"n = número de sucesos ");
outmatxy(50,240," a = número de éxitos ");
outmatxy(50,260," p = sucesos exitosos dividido ");
outmatxy(50,280," a la potencia ");
outmatxy(50,300," n = potencia de a ");
SIFACTORIAL(68,302);
outmatxy(50,320,"P(a) = probabilidad de a sucesos ");
sprmatf(" (%.3f)", p);
int col = 80 + int(strlen(s));
outmatxy(340,190,4);
sprmatf(" %d", a);
outmatxy(col,120,4);
outmatxy(col,130," a^n ");
outmatxy(col,140," * p^n ");
outmatxy(col,150," * n! / (n-a)! ");
sprmatf(" %d", 2F, p);
outmatxy(340+(col-340)*2,130,4);
SIFACTORIAL(180+(col-360)*2+3,150);
sprmatf("La probabilidad de %d, es %d", a);
outmatxy(340, 200, 4);
outmatxy(125,870,"Este fórmula se utiliza sólo cuando a >= 20 y p <= 05");
realms("O");
CIERRAGRAFICOS);
}

defines NMBRN 13
defines NCF 43
defines MAZ 100
#include <stdio.h>
#include <conio.h>
#include <alloc.h>
#include <string.h>
#include <ctype.h>
#include <graphics.h>
#include <conio.h>
#include <math.h>
#include <dos.h>
#include <stdlib.h>
#include <unistd.h>

TablaZ;
TablaT;

void main()
{
int op, opn = 0, n = 0;
float x[30], y[30];

for(int i=0; i<30; i++)
{
for(int j=0; j<30; j++)
{
}
}

}

```

Anexo D: Código fuente del sistema

```

matmul(1) unselectmen(12,12,"UNA MEDIA", "Hipotesis de una media", 8,-1),
matmul(1) unselectmen(13,13,"TRES MEDIDAS", "Hipotesis de independencia", 9,-1),
matmul(1) unselectmen(14,3,4,"UNA PROPORCION", "Hipotesis de una proporción", -2,-1),
matmul(1) unselectmen(15,3,5,"DOS PROPORCIONES", "Hipotesis de dos proporciones", -2,-1),

matmul(4) unselectm(2,16),
matmul(4) unselectmen(51,10,67,13,2),
matmul(4) unselectmen(51,11,3,4,"**",**,-1,-1),
matmul(4) unselectmen(16,3,2,"FI TUA I RANADA", "Cruce una prueba de independencia", -2,-1),
matmul(4) unselectmen(17,3,3,"INFERENCIA", "Distribución de la variancia de la media", 7,-1),

matmul(5) unselectm(3,18),
matmul(5) unselectmen(62,10,76,14,2),
matmul(5) unselectmen(15,11,3,4,"**",**,-1,-1),
matmul(5) unselectmen(18,3,2,"INSTRUMENTOS", "Instrumentos los puntos observados", -2,0),
matmul(5) unselectmen(19,3,3,"REGRESION", "Máxima variancia cuadrada", -1,0,0),
matmul(5) unselectmen(20,3,4,"CORRELACION", "Coeficiente de determinación", -2,0),

matmul(6) unselectm(2,21),
matmul(6) unselectmen(23,15,48,18,2),
matmul(6) unselectmen(15,11,3,4,"**",**,-1,-1),
matmul(6) unselectmen(21,3,2,"INTERVALO DE CONFIANZA", -2,2),
matmul(6) unselectmen(22,3,3,"TAMAÑO ADECUADO DE n", -2,2),

matmul(8) unselectm(2,25),
matmul(8) unselectmen(34,17,63,20,2),
matmul(8) unselectmen(15,11,3,4,"**",**,-1,-1),
matmul(8) unselectmen(23,3,2,"DISTRIBUION NORMALMENTE", -2,-2,3),
matmul(8) unselectmen(26,3,3,"NO DISTRIBUION NORMALMENTE", -2,-2,3),

matmul(9) unselectm(2,27),
matmul(9) unselectmen(39,19,56,20,2),
matmul(9) unselectmen(15,11,3,4,"**",**,-1,-1),
matmul(9) unselectmen(27,3,2,"DEPENDIENTES", -2,-2,3),
matmul(9) unselectmen(28,3,3,"INDEPENDIENTES", -2,-2,3),

matmul(11) unselectm(4,29),
matmul(11) unselectmen(50,16,76,21,3),
matmul(11) unselectmen(51,1,3,4,"**",**,-1,-1),
matmul(11) unselectmen(29,3,2,"VER TAILLA", -2,-2,3),
matmul(11) unselectmen(30,3,3,"VER GRAPICA", -2,-2,3),
matmul(11) unselectmen(31,3,4,"CALCULO O Y b", -2,-2,3),
matmul(11) unselectmen(32,3,5,"ERRORE ESTANDAR", -2,-2,3),

matmul(7) unselectm(3,33),
matmul(7) unselectmen(45,15,74,19,2),
matmul(7) unselectmen(51,1,3,4,"**",**,-1,-1),
matmul(7) unselectmen(33,3,2,"VARIANZA DE LA POBLACION", -2,-2,4),
matmul(7) unselectmen(34,3,3,"ESTIMAO PUNA VARIANZA", -2,-2,4),
matmul(7) unselectmen(35,3,4,"ESTIMAO DE LA VARIANZA", -2,-2,4),

de(
  lselectm(130),
  prin actvar(0),
  for (n=2, 10, 100) { gselect(6,7), gprntf(" "),
  prin avr, colvar,
  gselect(20,3),
  gselect("UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO"),
  gselect(27,5), gprntf(" B T A D I S T I C A I"),
  opt = menu(m) unselect,
  opt = last_opt/prn opt,
  len_opt(6,7),
  prin lselect(0),
  switch (opt) {
    case 600 COMBINANA), break,
    case 700 PERMUTAJA), break,
    case 800 PERMUTACIONES), break,
    case 900 PROBABIOPO), break,
    case 1021 ESTIMAMEDIAS), break,
    case 1022 CALCULO DE LOS MEDIOS), break,
    case 1121 PROBABILIDADES), break,
    case 1122 CALCULO DE LOS PERCENTAJES), break,
    case 1000 JETUAI RANADA), break,
    case 1733 INFERENCIA I), break,
    case 1734 INFERENCIA II), break,
    case 1735 UNA VARIANZA), break,
    case 1225 HIPERDISTRIBUION), break,
    case 1226 HIPERDISTRIBUION), break,
    case 1400 HIPERDISTRIBUION), break,
    case 1500 HIPERDISTRIBUION), break,
    case 1527 MEDIDAS DE MOMENTOS FISICA), break,
    case 1528 MEDIDAS DE MOMENTOS), break,
    case 1801 n = LEER, PUNTO DE V), break,
    case 1929 if (n=0) TABLA DE DISTRIBUCIONES), break,
    case 1940 if (n=0) GRAFICA DE DISTRIBUCIONES), break,
    case 1931 if (n=0) CALCULO DE LA MEDIA), break,
    case 1932 if (n=0) ERROR ESTANDAR DE LA MEDIA), break,
    case 2000 if (n=0) CORRELACION DE PUNTO), break,
    case 3 unselect("AYUDA A LOS 21") prin 0, break,
  } switch
  prin lselect(1),
  l select (opt = -1),
  CERRARGRAFICOS(),
  window(1,1,80,25),
  textbackgr(break(0)),
  clear(),
}

FUNCIONES DE DISTRIBUCION

// Calcula la probabilidad de las muestras
void PROGRAMAS()
{
  float media=0,dev=0,Z,erro,valor=0,valor1=0,tabla,
  un, n = 0, N=0, num=0, opt=0, key,
  clear(20), res,
  lselect("0"),
  del,
  prin lselect(1),
  gselect(16,2),
  gprntf("CALCULO DE PROBABILIDAD DE LA MEDIA MUESTRAL"),
  vmen avrvar("Introduzca los siguientes datos"),
  gselect(25,0), gprntf("Media poblacional (mu)"),
  if ((key = last(media,53,6,8,0,5000,0,0,0)) == ESC) return,
  gselect(25,0), gprntf("Desviación estándar (sigma)"),
  if ((key = last(dev,53,7,9,0,5000,0,0,0)) == ESC) return,
  gselect(25,0), gprntf("Tamaño de la muestra (n)"),
  if ((key = last(n,53,8,6,1,0,5000,0,0)) == ESC) return,
  vmen avrvar(0),
  opt = POBLACION(),
  res = " ",
  if (opt == 1) { Finian
  gselect(25,9), gprntf("Tamaño de la población (N)"),
  if ((key = last(N,53,9,6,0,1,0,5000,0,0)) == ESC) return,
  l else {
    res = NORMAL(),
    key = realdev("02"),
  }
  if ((key = last(ESC,dev, res = "2")
  while ((num=OPCIONRES()) != -2) {
    prin lselect(0),
    gselect(10,3), gprntf("La media es = %2f, media),
    if (num == 1) {
      gselect(10,3),
      menu(num) {
        case 0 gprntf("Valor Menor Que ="), break,
        case 1 gprntf("Valor Mayor Que ="), break,
        case 3 gprntf("Valor Observado ="),
      }
      if ((key = last(valor,29,8,8,0,5000,0,0)) == ESC || (valor) return,
      if (opt == 1) { (float) (float(N)-n)/sqrt(float(N)-n)),
      error=(dev/sqrt(n))*sqrt(float(N)-n)),
    }
  }
}

```

Anexo D: Código fuente del sistema

```

else //Infinita
  error="divisor(n)";
  z=(valor-media)/error;
  if(z < -3 || z > 3)
    FILTRO(z);
  else {
    Z=ZT buscarzeta(z);
    tabla=Z;
    COMPARA(x,media,error,valor,z);
    switch (nom) {
      case 0: //z=0) Z= 5-Z;
        else Z = 5-Z;
        aprntf("P% % 1) = % 4f",valor,Z); break;
      case 1: //z=0) Z= 3-Z;
        else Z = 5-Z;
        aprntf("P% % 1) = % 4f",valor,Z); break;
      case 3: //z=0) Z= 3+Z;
        else Z = 5-Z;
        aprntf("P% % 3) = % 4f",media, valor.Z);
    }
    outstaty(460,240,1);
    IMPRIMEZ(media,devr,valor,error,Z,tabla);
    TITULOS("PROBABILIDAD DE LAS MEDIAS
    MUESTRALES",14,20,3);
    if (opc == 1) TITULOS("FINITAS",2,50,3);
    else TITULOS ("INFINITAS",2,50,3);
    TLCLAS();
  }
} else {
  vrom=valor("Introduzca los valores ");
  gntoxy(10, 8); aprntf("valor 1 =");
  gntoxy(10,10); aprntf("valor 2 =");
  if (key = last(devr, 20,8,7,1,32000,FLOAT)) == EBC $ valor) return;
  if (key = last(devr, 1,20,10,1,32000,FLOAT)) == EBC $ (valor) ) return;
  vrom=valor(z);
  if (opc == 1)
    error="(devr/sqrt(n))*sqrt((float)(N/n))*(float)(N-1));
  else
    error=devr/sqrt(n);
  z=(float)(valor-media)/error;
  x1=(float)(valor 1+media)/error;
  Z=ZT buscarzeta(x1); //devr
  if(z < -3 || z > 3 || x1 < -3 || x1 > 3)
    FILTRO(z);
  else {
    if(z=0)
      Z = 5-Z;
    else Z = 5-Z;
    BICOMPARA(x1,media,error,media,z);
    TITULOS("PROBABILIDAD DE LAS MEDIAS
    MUESTRALES",14,20,3);
    if (opc == 1)
      TITULOS("FINITAS",2,50,3);
    else TITULOS("INFINITAS",2,50,3);
    DATOSR(media,devr,valor,error,1,x1,Z,error);
    aprntf("P% % 4) = % 4f",valor,Z); outstaty(410,250,1);
    TLCLAS();
  }
} // while
} while(key != EBC);

// Pregunta sobre las conclusiones de la población, para poder utilizar
// la distribución muestral
else NORMAL(mn) {
  char res;

  window(24,13,58,18);
  clrscr();
  prin advort(0);
  gntoxy(17,15); aprntf("Está distribuida normalmente la población S/N?");
  res = LERR(65,15,"Introduzca sólo un S ó N");
  prin limpiar();
  limpiar("02");
  if (res == "N" && n < 30) {
    gntoxy(10, 7);
    aprntf("Como no se conoce el tamaño de la Población y n (tamaño de la),
    gntoxy(10, 8);
    aprntf("muestral es pequeña, no se puede hacer absolutamente nada ");
    return "N";
  }
  else {
    if (res == "N") {
      gntoxy(27, 2); aprntf("TEOREMA DEL LIMITE CENTRAL");
      gntoxy(10, 6);
      aprntf("El Teorema del Limite Central permite hacer inferencias de la",
      gntoxy(10, 7);
      aprntf("poblaciones sin más conocimiento sobre la forma de la dist.");
      gntoxy(10, 8);
      aprntf("basado de frecuencias de la población, que el que se obtiene de");
      gntoxy(10, 9); aprntf("la muestra ");
      gntoxy(10,10);
      aprntf("la distribución del muestreo de la media se acerca a la Nor.");
      gntoxy(10,11);
      aprntf("realidad a medida que crece el tamaño de la muestra ");
      realmea("02");
      prin limpiar();
    }
    gntoxy(29, 2); aprntf("DISTRIBUCION NORMAL");
    gntoxy(15, 5); aprntf("Para realizar afirmaciones de Probabilidad sobre la");
    gntoxy(15, 6); aprntf("medida) se utiliza la Distribución Normal Estender");
    gntoxy(20, 8); aprntf("Formula ");
    gntoxy(20,10); aprntf(" ");
    gntoxy(20,11); aprntf(" ");
    gntoxy(20,12); aprntf(" ");
    gntoxy(20,13); aprntf(" ");
    gntoxy(20,14); aprntf(" ");
    gntoxy(20,16); aprntf("Donde ");
    gntoxy(20,17); aprntf(" ");
    gntoxy(20,18); aprntf(" ");
    gntoxy(20,19); aprntf(" ");
    gntoxy(20,20); aprntf(" ");
    gntoxy(20,21); aprntf(" ");
    return "C";
  }
}

// Muestra el mensaje de error cuando las desviaciones superen el rango de - 3
en FILTRO(float x) {
  gntoxy(10,13); aprntf("El número de desviaciones es = %f",x);
  MARCOS(15,16,64,18,12,0);
  limpiar(15);
  gntoxy(17,17); aprntf("Dimensione desviaciones para poder graficarlas");
  limpiar("02");
  if (realmea("02") == EBC) return 1;
  return 0;
}

// Dibuja la grafica de distribución cuando en de una sola
void COMPARA(float z, float media, float devr, int nom) {
  Grafica grafico(media,devr,3);
  INICIALPFORMO();
  Grafica grafico(media,devr,3);
  grafica grafico(z);
  switch (nom) {
    case 0: grafica contornar(media+devr*3,media+(devr*3)); break;
    case 1: grafica contornar(media+devr*2,media+(devr*3)); break;
    case 3: grafica contornar(media,media+devr*2);
  }
  grafica postvisualizar(media+devr*_media);
  grafica postvisualizar(media+devr*_media,0);
  return;
}

```


Anexo D: Código fuente del sistema

```

para (loopser);
gotoxy(10,5); aprunt("La probabilidad de azar es % 2f",p);
gotoxy(10,8);
if (num1 != 4) {
  intic(N,num1);
  case 0 aprunt("Valor mayor que ="), break;
  case 1 aprunt("Valor mayor o igual que ="), break;
  case 2 aprunt("Valor menor que ="), break;
  case 3 aprunt("Valor menor o igual que ="),
  }
  vmen avaco("Introduzca número decimal ");
  if (num1 == 0 | num1 == 2) col = 28;
  else col = 37;
  if ((key = leer(&valor, col, 8, 0, 0, 99999, FLOAT)) == ESC | valor == 0)
  return;
  if (opc == 1) //Finia
  factor = (float)pow(10, N);
  error = sqrt(p*q)*(float(N)) * sqrt((float)(N-n)/(float)(N-1));
  if (factor < 0.05) z = ERRORRINFINITOSUMA(p,q,valor,n,signo);
  else z = ERRORRINFINITOSUMA(p,q,valor,N,n,signo);
  } else //Infinita
  error = sqrt(p*q)*(float(N));
  z = ERRORRINFINITOSUMA(p,q,valor,n,signo);
}
if (p <= 3 | z >= 3)
  FILTER(x);
else {
  Z = TZ.buscar(x,z);
  table Z;
  COMPARAP(x,p,valor,valor,num);
  avaco(N,num);
  case 0: if (x <= 0) z = 5+z;
  else z = 5-z;
  aprunt("x %f <= %d 2f) = %f 4f, valor Z); break;
  case 1: if (x <= 0) z = 5+z;
  else z = 5-z;
  aprunt("x %f <= %d 2f) = %f 4f, valor Z); break;
  case 2: if (x <= 0) z = 5+z;
  else z = 5-z;
  aprunt("x %f <= %d 2f) = %f 4f, valor Z); break;
  case 3: if (x <= 0) z = 5+z;
  else z = 5-z;
  aprunt("x %f <= %d 2f) = %f 4f, valor Z);
}
outastxy(420,220,a);
aprunt("x %f = %f 4f, tabla);
outastxy(420,220,a);
INSTRUP(p, valor, z, signo, error, n);
TITULOS("PROBABILIDAD DE LAS PROPORCIONES
MUESTRALES", 14, 20, 3);
if (opc == 1) TITULOS("FINITAS", 2, 50, 3);
else TITULOS("INFINITAS", 2, 50, 3);
TCLAS(a);
}
} else //Entre dos valores
limasaco("INT");
vmen avaco("Introduzca número decimal");
gotoxy(10, 8); aprunt("Valor 1 = ");
if ((key = leer(&valor, 21, 8, 5, 0, 0, 99999, FLOAT)) == ESC | valor == 0)
return;
gotoxy(10, 10); aprunt("Valor 2 = ");
if ((key = leer(&valor, 21, 10, 5, 0, 0, 99999, FLOAT)) == ESC | valor == 0)
return;
if (opc == 1) //Finia
error = sqrt(p*q)*(float(N));
error = sqrt(p*q)*(float(N)) * sqrt((float)(N-n)/(float)(N-1));
if (factor < 0.05)
z = ERRORRINFINITOSUMA(p,q,valor,a,1);
z = ERRORRINFINITOSUMA(p,q,valor,a,2);
else {
z = ERRORRINFINITOSUMA(p,q,valor,N,a,1);
z1 = ERRORRINFINITOSUMA(p,q,valor,N,a,2);
} else //Infinita
error = sqrt(p*q)*(float(N));
z = ERRORRINFINITOSUMA(p,q,valor,a,1);

```

```

z1 = ERRORRINFINITOSUMA(p,q,valor,1,a,2);
}
if (p <= 3 | z >= 3 | z1 >= 3 | z1 >= 3)
  FILTER(x);
else {
  Z = VTBI.COOPARA(x,z,1);
  table Z;
  HICOMPARA(x,p,valor,valor,valor,1);
  aprunt("x %f = %f 4f, tabla); outastxy(420,220,a);
  aprunt("x %f = %f 2f) = %f 4f, valor valor Z);
  outastxy(420,260,a);
  INPRIN(PP, valor, z, 1, error, n);
  TITULOS("PROBABILIDAD DE LAS PROPORCIONES
MUESTRALES", 14, 20, 3);
  if (opc == 1) TITULOS("FINITAS", 2, 50, 3);
  else TITULOS("INFINITAS", 2, 50, 3);
  TCLAS(a);
}
}
}
} while (key != ESC);
// Preguntar entre que valores esta la probabilidad para proporciones
int VALORPO {
  int numnum = 0;
  prin dibuque();
  tustocole(12);
  gotoxy(16,5); aprunt("La Probabilidad de la Proporción (Manual ser ");
  numcolor(15);
  limasaco("INT");
  numtab(0) colvmen(15,1,1,3,4);
  numtab(0) colvmen(5,0);
  numtab(0) numvmen(7,9,50,15,2);
  numtab(0) numvmen(0,2,2, "PROPORCION = ";
  numtab(0) numvmen(0,1,3,3, "PROPORCION > ";
  numtab(0) numvmen(0,2,3,4, "PROPORCION < ";
  numtab(0) numvmen(0,2,3,4, "PROPORCION < ";
  numtab(0) numvmen(0,3,5,5, "PROPORCION < ";
  numtab(0) numvmen(0,4,3,4, "ENTRE DOS VALORES");
  numtab(0) numvmen(0,4,3,4, "PROBABILIDAD entre dos valores < ");
}
prin netvco(0);
int error(1,-1);
num = leer_ahc(num,0);
tm = error(6,7);
prin_ahc(error(0);
return(num);
}
// Cálculo el error infinito para proporciones
float error_z;
error = sqrt(p*q)*(float(n));
if (opc == 1)
float error_z = error*(1/(2*(float(n)))));
else error = error*(1/(2*(float(n)))));
else error = error;
return(z);
}
// Cálculo error finito para proporciones
float error_z;
error = sqrt(p*q)*(float(n));
error = sqrt(p*q)*(float(n)) * sqrt((float)(N-n)/(float)(N-1));
error = error*(1/(2*(float(n)))));
error = error*(1/(2*(float(n)))));
error = error;
return(z);
}

```

Anexo D: Código fuente del sistema

• Imprime datos y resultados en la grafica de distribución de la proporción
 void IMPRINTH(float p, float valor, float z, int signo, float error, int n)
 {
 char s(30);
 float q = 1-p;

```

setextstyle(SMALLI, FONT_HORIZ, DIR_4), setcolor(2);
outtextxy(80,60,"DATOS");
outtextxy(400,60,"FORMULA");
setcolor(15);
sprintf("p = %2f",valor);
line(80, 80, 85, 80);
sprintf("q = %2f",p);
sprintf("q = %2f",q);
sprintf("n = %2f",n);
outtextxy(400,90,"p = ");
outtextxy(430,90,"q = ");
outtextxy(450,97,"n = ");
outtextxy(420,105,"z = ");
line(85, 115, 449, 115);
setextstyle(SMALLI, FONT_HORIZ, DIR_3);
outtextxy(400,115,"p");
setextstyle(SANS_SERIF, FONT_HORIZ, DIR_3);
outtextxy(400, 81, (""));
outtextxy(450, 130, (""));
setextstyle(SMALLI, FONT_HORIZ, DIR_4);
sprintf("2*%d",n);
sprintf(" * %f",p);
if (signo == 1) sprintf(" * %f",valor);
else sprintf(" * %f",valor);
outtextxy(450,135,"");
line(450,160,600,160);
outtextxy(500,162,"");
sprintf("Z = %2f desviaciones",z);
    
```

// Dibuja la grafica para a distribución en proporciones
 void CMAPARAF(float x, float media, float desv, int num)

```

{
    Grafica graficamedia,desv,3);
    IMPRINTH(FUNC);
    Grafica graficamedia,desv,3);
    Grafica dibuja);
    switch (num) {
        case 0: grafica sombread(media+desv*_z,media+(desv*3)); break;
        case 1: grafica sombread(media+desv*_z,media+(desv*1)); break;
        case 2: grafica sombread(media+desv*_z,media+(desv*2)); break;
        case 3: grafica sombread(media+desv*_z,media+(desv*4)); break;
    }
    grafica pondeviaciones(media+desv*_z,media);
    grafica pondeviaciones(media+desv*_z,media,1);
}
return;
    
```

// Muestra el teorema de las proporciones
 void THEOREMA(void)

```

{
    int key;
    prin limpiar();
    gotoxy(20, 2);sprintf("DISTRIBUCION NORMAL");
    gotoxy(15, 4);sprintf("Se puede utilizar la Distribucion Normal");
    gotoxy(15, 6);sprintf("Formulas");
    gotoxy(20, 8);sprintf("U = ");
    gotoxy(20, 9);sprintf("U = ");
    gotoxy(20, 10);sprintf("Z = ");
    gotoxy(20, 11);sprintf("Z = ");
    gotoxy(20, 12);sprintf("Z = ");
    gotoxy(20, 13);sprintf("p = ");
    gotoxy(15, 14);sprintf("Dado");
    gotoxy(20, 14);sprintf("p = proporción");
    gotoxy(20, 15);sprintf("n = ");
    gotoxy(20, 17);sprintf("n = factor de corrección de continuidad");
    
```

```

    gotoxy(20, 18);sprintf("2n");
    gotoxy(20, 19);sprintf("1 - proporción en población");
    gotoxy(20, 21);sprintf("E = error estándar de la proporción en");
    gotoxy(20, 22);sprintf("p = el muestreo");
    key=waitkey("D");
}
return key;
    
```

```

// Pregunta por el tipo de población (finita o infinita)
int PRH_ALEJANI()
{
    int num=0;opc;
    vmen avamos();
    setcolor(12);
    gotoxy(24, 13);sprintf("La Poblacion se considere como");
    menu(1) contenzema(1,1,1,1,4);
    menu(0) intenzema(2,1,5,4,18,0);
    menu(1) intenzema(2,15,4,18,0);
    menu(0) intenzema(1,3,2, "FINITA", "", -2);
    menu(0) intenzema(2,3,3, "INFINITA", "", -2);
    int op=0;
    op= leer sub(0,0);
    int cont=7;
    window(24,13,48,18);
    clear();
    prin activar();
    return op;
}
    
```

// Permite enviar a impresion la grafica en pantalla

```

void TELAASH()
{
    int key;
    loc:seek("r",2);
    do {
        if (key=waitkey("278"))=F3)
            TZ desplaytablas();
        if (key = F2)
            printpage(0,0,getmaxc()-1, getmaxr()-20);
    } while (key !=ESC || key=ENTER);
}
CLEARGRAPHICS();
    
```

//FUNCIONES DE PRUEBA DE HIPOTESIS

// Prueba de hipotesis para una media
 int IMPRINTHAI(int dist, num)

```

{
    float alfa, sa, Zc;
    float media_0, media_0+desv_0;
    int tipo=n-1;
    char s(30);
    int key = ENTER;
}
    
```

```

do {
    prin limpiar();
    gotoxy(23, 3);sprintf("PRUEBA DE HIPOTESIS PARA LA MEDIA");
    vmen avos("Introduzca los siguientes datos");
    gotoxy(25, 9);sprintf("Tamaño de la muestra(n)");
    gotoxy(25, 11);sprintf("Media poblacional (μ)");
    gotoxy(41, 12);sprintf(" ");
    gotoxy(25, 13);sprintf("Media muestral (x)");
    gotoxy(25, 15);sprintf("Desviación (S o σ)");
    if (key = leer(8;media, 50, 9, 5, 1, 32000,INT)) = EISC) return 1;
    if (key = leer(8;media, 50, 11, 8, 0, 32000)) = EISC) return 1;
    if (key = leer(8;media, 50, 13, 8, 0, 32000)) = EISC) return 1;
    if (key = leer(8;desv, 50, 15, 8, 0, 32000)) = EISC) return 1;
    vmen avamos();
    while (CELEBRAR(Af(alfa+n*3)) != 2)
        while (tipo = TIPOPRUEBA() != -3)
            if (dist != Zc = (media+media_0)/desv) (desv*(n)),
                else Zc=(media-media_0)/desv;
            if (alfa > 0 || Zc <= -3) alfa=0;
            if (tipo = 3) vt = 0.5 - alfa/2;
            else vt = 0.5 - alfa;
            if (vt <= 0 || dist != "nov") {T buscar(vt); vt = T buscar(alfa, n-1);
                INICIAFUNC();
            }
    }
}
    
```


Anexo D: Código fuente del sistema

```

do {
  TITULOS["PRUEBA DE HIPOTESIS PARA LA MEDIDA",14,20,3],
  setactlay(SMALL, FONT,HORIZ,DIR,4),
  setcolor(15),
  sprinff(a,"% %4d",n),
  sprinff(a,"% %4f",media),
  outtextxy(60,85,a),
  line(60,97,65,97),
  sprinff(a,"% %4f",media),
  outtextxy(60,99,a),
  sprinff(a,"% %4f",medprob),
  outtextxy(60,115,a),
  sprinff(a,"% %4f",Zc),
  outtextxy(60,125,a),
  line(525,102,530,102),
  outtextxy(525,150,a),
  line(545,128,565,120),
  line(555,178,560,170),
  sprinff(a,"% %4f",Zc),
  outtextxy(490,160,"Zc ="),
  outtextxy(490,200,""),
  outtextxy(490,160,"Zc ="),
  outtextxy(490,200,""),
  outtextxy(490,230,a),
  }
  HIPOTESIS("P",a,ntpo),
  CONCLUSION(tipo,vt,Zc),
  GRAFICAH(tipo,media,deav,vt),
  ALFA(tipo,alfa),
  key=waitkey(0.278),
  if (key == F2 | key == F3)
  {
    if (date)
      ESPERA(n,key),
      else ESPERA(n,key),
      while (key != ESC && key != ENTER),
      CERRARGRAFICOS(),
      prin dibujar(),
      }
  }while (key != ESC),
  textcolor(15),
  textbackground(0),
  clrscr(),
  return 0,
}
// Prueba de hipótesis para una proporción
int HIPROPC(void)
{
  float alfa,vt,Zc,
  float medprob,pro_pno,
  int tipo,key,n=0,
  chw a[30],
}
do {
  prin limpiarc(),
  gotoxy(27,3), sprinff("HIPOTESIS PARA LA PROPORCION"),
  vtrom=screen/Introduzca los siguientes datos",
  gotoxy(28,8), sprinff("Tamaño de la muestra(n)"),
  gotoxy(20,10), sprinff("Proporción de las poblaciones(P)"),
  gotoxy(46,11), sprinff("Zc"),
  gotoxy(20,12), sprinff("Nómbra de las proporciones(P)"),
  if (key == line (alm,54,8,5,1,32000,INT)) == ESC) return 1,
  if (key == line (alpro,50,34,10,8,0,1)) == ESC) return 1,
  vtrom=screen/0 si no coincide la media",
  if (key == line (almedprom,54,12,8,0,1)) == ESC) return 1,
  vtrom=screen/0",
  while ((LIBERALFA(alfa,n) == -2)
  while (tipo == TIPOPRUEBAAC) == -2)
  if (alfa > 0.1 && n <= 30) alfa = 0.1,
  if (tipo == 3) vt = 0.5 - alfaZ,
  else vt = 0.5 - alfa,
  if (n <= 30) T=Int(sqrt(chw/vt)), vt = T * bucaer(alfa,n,1),
  else vt = Z * bucaer(vt),
  do {
    INIGRAFICOS(),
    TITULOS["PRUEBA DE HIPOTESIS PARA LA PROPORCION",14,20,3],
    setactlay(SMALL, FONT,HORIZ,DIR,4),
    setcolor(15),
    sprinff(a,"% %4d",n),
    sprinff(a,"% %4f",pro_pno),
    sprinff(a,"% %2f",alfa),
    if (medprob)
      {
        Zc = (medprob - pro_pno)/sqrt((pro_pno*(1-pro_pno))),
        sprinff(a,"% %4f",medprob),
        sprinff(a,"% %4f",Zc),
        outtextxy(525,100,a),
        line(520,115,570,115),
        line(530,133,560,133),
        sprinff(a,"% %2f",Zc),
        sprinff(a,"% %2f",medprob,pro_pno),
        line(520,190,525+textwidth(a)+3,170),
        line(525,165,525+textwidth(a)+3,165),
        line(525,182,525+textwidth(a)+3,182),
        sprinff(a,"% %2f",Zc),
        sprinff(a,"% %4f",Zc),
        if (n <= 30)
          outtextxy(490,110,"Zc ="),
          outtextxy(490,150,""),
          outtextxy(490,110,"Zc ="),
          outtextxy(490,150,""),
          outtextxy(490,210,a),
          }
        else
          outtextxy(490,110,"Zc ="),
          sprinff(a,"% %4f",Zc),
          sprinff(a,"% %4f",Zc),
          sprinff(a,"% %4f",Zc),
          CONCLUSION(tipo,vt,Zc),
        } else {
          medprob = vt * (sqrt((pro_pno*(1-pro_pno))) + pro_pno),
          outtextxy(60,97,"P ="),
          line(60,97,65,97),
          line(533,115,563,113),
          outtextxy(530,100,"k ="),
          if (n <= 30)
            outtextxy(450,105,"vt = % %f",
            else outtextxy(450,105,"Zc = % %f",
            raiz(515,121,365,100),
            outtextxy(450,110,"k ="),
            line(523,152,590,152),
            sprinff(a,"% %2f",Zc),
            sprinff(a,"% %4f",Zc),
            outtextxy(450,145,a),
            outtextxy(450,155,a),
            sprinff(a,"% %4f",medprob),
            line(450,180,455,180),
            outtextxy(450,200,a),
            CONCLUSION(tipo,vt,medprob),
            }
          HIPOTESIS("P",a,ntpo),
          setcolor(15),
          line(223,85,228,85),
          line(223,100,228,100),
          GRAFICAH(tipo,medprob,10,vt),
          ALFA(tipo,alfa),
          key=waitkey(0.278),
          if (key == F2 | key == F3)
            ESPERA(n,key),
            while (key != ESC && key != ENTER),
            CERRARGRAFICOS(),
            prin dibujar(),
            }
          } while (key != ESC),
          return 0,
        }
        // Prueba de hipótesis para dos proporciones
        int HIP2PROPC(void)
        {
          int P1,P2,tip,key,n1 = 0, n2 = 0,
          float p1,p2, q1,q2,alfa,Zc,vt,

```

Anexo D: Código fuente del sistema

```

char s[50],
P1=P2=0,
do {
    prin limpa();
    getch();
    getch(23,3), printf("HIPOTESIS PARA DOS FRECUENCIAS"),
    getch(20,6), printf("Datos de la primera muestra"),
    getch(20,15), printf("Datos de la segunda muestra"),
    getch(25,8), printf("Tamaño de la población"),
    getch(25,18), printf("Distribucion de la muestra"),
    getch(25,17), printf("Tamaño de la población"),
    getch(25,19), printf("Distribucion de la muestra"),
    printf("Tamaño de las pruebas: control"),
    if (key == leer (dn1, 53, 8, 1, 32000, INT)) -- ESC) return 1,
    if (key == leer (dn1p, 53, 10, 4, 1, 32000, INT)) -- ESC) return 1,
    if (key == leer (dn2, 53, 17, 6, 1, 32000, INT)) -- ESC) return 1,
    if (key == leer (dn2p, 53, 19, 4, 1, n2, 1, INT)) -- ESC) return 1,
    p1 = P1/(float)n1, p2 = P2/(float)n2,
    q1 = 1-p1, q2 = 1-p2,
    getch(25,12), printf("p1 = %3f q1 = %3f p1,q1"),
    getch(25,21), printf("p2 = %3f q2 = %3f p2,q2"),
    realineat(" ");
    printf("\n\n");
    while ((LEERALFA=leer(alfa)) <= 2)
    while (tipo = TIFRACASION) {
        if (alfa < 0.1 && n1 = 30 && n2 = 30) alfa = 0.1,
        if (tipo = 3) alfa = 0.5, alfa2 =
        else alfa = 0.5, alfa,
        if (n1 != 30 && n2 != 30) if (archivos), w = T, bcaract(alfa, n1-1),
        else w = Z, bcaract(alfa),
        Z = (p1-p2)/sqrt((p1*q1)/(float)n1) + (p2*q2)/(float)n2),
        do {
            INGRAFICASY:
            TITULO="PRUEBA DE HIPOTESIS PARA DOS
            PROPORCIONES";
            settextstyle(BKGL, FONT_IBMPC, DIR_4);
            outtextxy(50,85), outtextxy(60,85),
            printf("n1 = %d",n1), outtextxy(60,105),
            printf("n2 = %d",n2), outtextxy(60,125),
            printf("p1 = %d",p1), outtextxy(60,145),
            printf("p2 = %d",p2), outtextxy(60,165),
            printf("q1 = %d",q1), outtextxy(60,185),
            printf("q2 = %d",q2), outtextxy(60,205),
            printf("alfa = %d",alfa), outtextxy(60,225),
            // Formas
            line(480,115,560,115), line(480,132,515,132),
            outtextxy(500,100,"P1", "P2"), outtextxy(490,120,"p1 q1 = p2 q2"),
            // Caracteres
            printf("%3f", %3f, %3f, p1, p2), outtextxy(505,150,a),
            outtextxy(440,160,"Z"),
            printf("%3f", %3f, %3f, p1, p2, q1, p2, q2),
            line(480,165,480+textwidth(b)+1,165),
            line(480,192,480+textwidth(b)+1,192),
            printf("%3f", %3f, %3f, p1, q1), outtextxy(480,170,a),
            line(480,182,480+textwidth(b),182),
            outtextxy(480+textwidth(b)+1,170,"+"),
            printf("%3f", %3f, %3f, p2, q2), outtextxy(480,182,a),
            line(565,182,565+textwidth(b),182),
            outtextxy(565,170,a), printf("n1",n1),
            outtextxy(510,182,a), printf("n2",n2),
            outtextxy(440,210,a), printf("Z = %3f,Za"),
            outtextxy(440,215,a), printf("t = %3f,t1"),
            CONCLUSION="No se",
            HIPOTESIS="P1", "P2", "p1",
            CRITICO="Alfap=0.05, 0.10, 0.1",
            ALFA="p=0.05",
            line(480,"077",2,0),
            key = realineat("077");
            if (key == F2) { key = F3 }
        }
    }
} while (key != ESC),
while (key != ESC) { key = ENTER),
while (key != ESC),
return 0,
}
// Prueba de hipotesis para dos medias grandes
// MEDIAS IGUALES
float alfa, w, Zc,
float med1, med2, n1, dev1, n2, dev2, u,
float sum, sumas,
int tipo, key, n1, n2, n,
float art[10],
char s[50],
do {
    prin limpa();
    getch(20,2), printf("HIPOTESIS PARA DOS MEDIAS"),
    getch(28,20), printf("Son datos sueltos [S/N]?"),
    char res = LEER(S); printf("Introduzca solo S o N"),
    prin limpa();
    if (res == 'S')
    getch(20,2), printf("HIPOTESIS PARA DOS MEDIAS
    INDEPENDIENTES"),
    getch(11,4), printf("Datos de la primera muestra"),
    getch(11,18), printf("Tamaño muestral (n)"),
    if (key == leer (dn1, 33, 6, 5, 1, 30, INT)) -- ESC) break,
    for (i = 0; i < 35; i++) art[i] = 0,
    if ((LEERDATOS=S), u, n1, art) == ESC) return 1,
    sum=0,
    for (i=0; i<n1; i++) summas += pow(art[i],med1),
    med1 = sum/(float)n1,
    sumas=0,
    for (i=0; i<n2; i++) summas += pow(art[i],med2),
    dev1 = sqrt(sumas/(float)n2),
    getch(11,16), printf("Medias %3f,med1"),
    getch(11,18), printf("Desviacion %3f,dev2"),
    realineat(" ");
    prin limpa();
    getch(20,2),
    printf("HIPOTESIS PARA DOS MEDIAS INDEPENDIENTES"),
    getch(11,4), printf("Datos de la segunda muestra"),
    getch(11,8), printf("Tamaño muestral (n)"),
    if (key == leer (dn2, 33, 6, 5, 1, 30, INT)) -- ESC) break,
    for (i = 0; i < 35; i++) art[i] = 0,
    if ((LEERDATOS=S), u, n2, art) == ESC) return 1,
    sum=0,
    for (i=0; i<n2; i++) summas += pow(art[i],med1),
    med2 = sum/(float)n2,
    sumas=0,
    for (i=0; i<n2; i++) summas += pow(art[i],med2),
    dev2 = sqrt(sumas/(float)n2),
    getch(11,16), printf("Medias %3f,med2"),
    getch(11,18), printf("Desviacion %3f,dev2"),
    realineat(" ");
    } else {
    getch(20,2), printf(" HIPOTESIS PARA DOS MEDIAS"),
    getch(20,5), printf("Datos de la primera muestra"),
    getch(20,14), printf("Datos de la segunda muestra"),
    getch(30,7), printf("Tamaño muestral (n)"),
    getch(45,8), printf(" "),
    getch(30,9), printf("Medias muestrales (a)"),
    getch(30,11), printf("Distribucion (b o j)"),
    getch(30,16), printf("Tamaño muestral (n)"),
    getch(30,16), printf("Distribucion (b o j)"),
    getch(30,18), printf("Medias muestrales (a)"),
    getch(30,20), printf("Distribucion (b o j)"),
    if (key == leer (dn1, 53, 7, 5, 1, 32000, INT)) == ESC) return 1,
    if (key == leer (dn1med, 53, 9, 10, 0, 65000)) == ESC) return 1,
    if (key == leer (dn1dev, 53, 11, 10, 0, 65000)) == ESC) return 1,
    if (key == leer (dn2, 53, 16, 5, 1, 65000, INT)) == ESC) return 1,
}
}

```

Anexo D. Código fuente del sistema

```

if ((key = leer (&med2, 53,10,10, 0, 65000)) == EESC) return 1,
if ((key = leer (&dev2, 53,20,10, 0, 65000)) == EESC) return 1,
}
while ((LIBERAL/Fa(alfa,n1) != -2))
while ((tupo = TIPCOPRUEBAZC) != -231)
  if ((alfa <= 0.1 && n1 <= 30 && n2 <= 30) && alfa <= 0.1,
  if ((tupo == 3) && vt = 0.5 && dev2,
  else vt = 0.5 && alfa,
  if ((n1<=30 && n2<=30) { T leerarchivo(), vt = T buscar(alfa, n1+n2-2),
  Zc = sqrt(((float)1)/((float)n1))*((float)1)/((float)n2)),
  Zc = sqrt(((float)1)/((float)Ndev1))*((n2-1)*((dev2*dev2))
  ((float)(n1+n2-2)) * Zc,
  Zc = (med1-med2)/Zc,
  }
  else {vt = Z buscar(vt),
  Zc = (med1-med2)/sqrt(((dev1*dev1) + (dev2*dev2))/n2)),
  }
do {
  INICIARFICOSO,
  TITULO(S)PRUEBA DE HIPOTESIS PARA DOS MEDIAS", 14,20,1),
  mensaje(vf3),MALLI, POINT,HORIZ,DIR_A,
  accion(15),
  sprnfl("n1= %d",n1), outstxy(60,85,a),
  sprnfl("n2= %d",n2), outstxy(60,105,a),
  sprnfl("n1= %3f",med1), outstxy(60,105,a),
  line(60,107.65,107), line(60, 117.65, 117),
  sprnfl("s2= %3f",med2), outstxy(60,115,a),
  sprnfl("n1= %3f",med1), outstxy(60,125,a),
  sprnfl("s2= %3f",dev2), outstxy(60,135,a),
  sprnfl("alfa= %2f",alfa), outstxy(60,145,a),
  //Forma
  if ((n1<=30 && n2 <= 30) {
  line(450,115.570,115), raiz(450,140.570,120),
  line(480,132.590,132), raiz(480,160.620,120),
  line(460,132.570,132), line(608,132.621,132),
  //Forma
  line(400,102,405,102), line(400, 102,405,102),
  line(520,102,525,102), outstxy(420,110,"s="),
  outstxy(460,120,"n1" + s1 + "(n2-1)*s2"),
  outstxy(480,132,"n1" + s2 + ")",
  outstxy(589,120,"1" + ")",
  sprnfl("s=" %3f", Zc), outstxy(420,150,a),
  sprnfl("v=" %2f",vt), outstxy(420,190,a),
  } else {
  line(500,115.575,115), raiz(500,142.575,120),
  line(530,132.530,132), line(530,132.575,132),
  outstxy(460,110,"Zc="), line(550,102,555,102),
  outstxy(510,120,"(S1)" + ")", outstxy(510,132,"n1" + ")",
  }
  // Descripcio
  sprnfl("s=" %3f, %3f",med1, med2), outstxy(460,160,"Zc="),
  outstxy(460,160,"Zc="),
  sprnfl("v=" %2f) + "(% 2f)",dev2), outstxy(510,170,a),
  line(500,165,500+int(dev1*(n1+10),165), raiz(500,192.500+int(dev1*(n1+10),170),
  sprnfl("v=" %2f)",dev1), outstxy(510,170,a),
  ut lon = 510 + int(dev1*(n1+12)), outstxy(lon-5,170,"*),
  line(510,182,lon,182), outstxy(lon+10,170,a),
  sprnfl("v=" %2f)",dev2), outstxy(520,182,a),
  line(lon-15,182,lon+13+int(dev1*(n1+12)), outstxy(600,200,a),
  outstxy(600,240,a),
  sprnfl("s=" %4f",n1,n2), outstxy(520,182,a),
  sprnfl("Zc = %3f",Zc), outstxy(600,240,a),
  sprnfl("v=" %2f",vt),
  }
  CONCLUSION(tupo, vt, Zc),
  HIPOTESIS("M1", "M2", tupo),
  GRAFICAR(Hip(tupo, med1, dev1, vt),
  ALFA(tupo,alfa),
  key: realmod("O278"),
  if ((key == F2) && key == F3)
  ESPERA(n, key),
  while ((key = EESC && key != ENTER),
  CERRARAFICOSO,
  prin dibaja(),
  }
  }
  }
  }
  // (tupo=-2),
  } while (key != EESC),
  return 0,
  }
  // Prueba de hipotesis para dos medias dependientes
  en MEDIASDEPENDIENTES()
  {
  int muestra1[31],muestra2[31],
  double Zc, suma1,(suma1/2),dif,dif2,
  float alfa,v,
  int i,libro,tupo,ren,
  char s[30],
  do {
  prin limpia(),
  linea(60,"O"),
  goney(22,2), sprnfl("HIPOTESIS PARA DOS MEDIAS DEPENDIENTES"),
  goney(27,15), sprnfl("Titulo de la muestra 1"),
  if ((key = leer (&n1, 30, 12, 3, 8, 30)) != EESC) return 1,
  goney(28,20), sprnfl("Son datos muestrales [S/N]?"),
  char res = LINEA(53,20,"Introduce solo una S o N ");
  prin limpia(),
  goney(22,2), sprnfl("HIPOTESIS PARA DOS MEDIAS DEPENDIENTES"),
  if ((res == 'S') {
  suma1=suma1/2+0,
  for (i=0, i<n1, i++) {
  if ((i%30==0)) {
  prin limpia();
  goney(22,2),
  sprnfl("HIPOTESIS PARA DOS MEDIAS DEPENDIENTES"),
  goney(10,4),
  sprnfl("s=" %d Muestra 1 Muestra 2 d d",n1),
  vmen avn="Introduce el dato y pulse ENTER.",
  ren = 0,
  }
  goney(13,ren), sprnfl("%d",i+1),
  if ((key = leer (&suma1[i],25,ren, 5, 0,32000)) == EESC) return,
  if ((key = leer (&suma2[i],41,ren, 5, 0,32000)) == EESC) return,
  dif=0,
  dif2=0,
  dif= suma1[i]-1+suma2[i],
  goney(32,ren),
  suma1+=dif,
  dif=dif+dif,
  goney(64,ren), sprnfl("%d 2f", dif),
  suma2+=dif,
  ren++;
  }
  vmen avn=ZC(),
  goney(50,ren), sprnfl("-----"),
  goney(52,ren+1), sprnfl("%d 2f",suma1/2),
  goney(52,ren), sprnfl("-----"),
  goney(64,ren+1), sprnfl("%d 2f",suma2/2),
  realmod("O"),
  } else {
  goney(15, 8), sprnfl("Titulo de la muestra (n) %d",n1),
  goney(15,10), sprnfl("Titulo de las diferencias (d) %d",n2),
  goney(15,12), sprnfl("Suma de las diferencias (suma_d) %d",*),
  if ((key = leer (&med1,53,10, 10, 0, 600), 999999999)) == EESC) return,
  if ((key = leer (&med2,53,12, 10, 0, 600), 999999999)) == EESC) return,
  while ((LIBERAL/Fa(alfa,n1) != -2))
  while ((tupo = TIPCOPRUEBAZC) != -2) {
  if ((alfa <= 0.1) && alfa <= 0.1,
  if ((tupo == 3) && vt = 0.5 && dev2,
  else vt = 0.5 && alfa,
  vt = T buscar(alfa, n1-1),
  d = suma1/n1,
  Sd = (suma1/2)-(suma1*suma1)/(float)n1)/((float)(n1-1),
  Zd = sqrt(Sd),
  Zc = d/(Sd*sqrt(n)),
  }
  do {
  INICIARFICOSO,

```

Anexo D: Código fuente del sistema

TTILOS "PRUEBA DE HIPOTESIS PARA DOS MEDIAS
DEPENDIENTES",14,20,3)

testestatoy(SMALL_FONT_HORIZ_DIR,4)

setcolor(15)

aprendi("e" + "%f",n1)

aprendi("d" + "%f",d)

line(60,90,50,90)

aprendi("2d" + "%f",2d)

aprendi("g" + "%f",n1)

outstatoy(470,105,"d")

line(470,105,475,105)

outstatoy(515,111,"e")

outstatoy(470,150,"2d")

line(508,150,545,150)

outstatoy(525,150,"g")

max(503,sectores(d) * d - ",145),

testestatoy(SMALL_FONT_HORIZ_DIR,2)

outstatoy(528,165,"2")

testestatoy(SMALL_FONT_HORIZ_DIR,4)

line(500,150,545,195)

line(517,190,522,190)

outstatoy(470,190,"e")

aprendi("e" + "%f",d)

outstatoy(470,230,"e")

line(508,235,545,235)

max(503,sectores(a)+10,248,10+3,240)

aprendi("e" + "%f",Ze)

*(%n) (L) (R) (S) (Z) (e)

H(n) (L) (S) (M) (Z) (e) (p)

float key = 0; repeat (float n1)

GRAFIA "AHF" (ops, d, 140, 500, 3)

ALFA (t) (p) (a) (f)

key = readline("077")

if (key == F2 a key == F3)

EFERAR(n1, key)

while (key != ESC a key != ENTER)

CHERRAJRAF(C)(S)

pern dibujar()

while (key != '\n')

return 0;

if (veritas seleccionar el tipo de prueba para hipotesis

en TIPOPRUEBA(void)

{

int op; ren = 11;

pern limpiar();

setcolor(12);

getoyoy(21,3);

*(%n) (L) (R) (S) (Z) (e) (p) (a) (f)

testestatoy(5)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

outstatoy(n1,8,4)

outstatoy(60,9,3)

outstatoy(60,10,4)

line(508,111,530,111)

outstatoy(515,100,"d")

max(503,100)

outstatoy(508,145,"d" + "f")

max(503,145)

line(573,155,578,145)

outstatoy(570,145,"2")

max(525,200,545,200)

outstatoy(505,180,"d")

outstatoy(505,200,"2d" + "n")

outstatoy(510,220,"a")

aprendi("e" + "%f",n1)

int ren = 503; sectores(d)

aprendi("e" + "%f",2d)

outstatoy(470,270,"a")

outstatoy(470,310,"a")

pern limpiar();

setcolor(12);

getoyoy(31,5);

*(%n) (L) (R) (S) (Z) (e) (p) (a) (f)

testestatoy(15)

renmax(4,3)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

*(%n) (L) (R) (S) (Z) (e) (p) (a) (f)

testestatoy(12)

getoyoy(31,5);

*(%n) (L) (R) (S) (Z) (e) (p) (a) (f)

testestatoy(15)

renmax(4,3)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

*(%n) (L) (R) (S) (Z) (e) (p) (a) (f)

testestatoy(12)

getoyoy(31,5);

*(%n) (L) (R) (S) (Z) (e) (p) (a) (f)

testestatoy(15)

renmax(4,3)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

renmax(11)

Anexo D: Código fuente del sistema

```

c[6] = outxxy(480,62,"Bilateral"), strepy(cad3,"n")};
setcolor(7);
sproutf("H4 %s %s",cad1,cad2), outxxy(200,85,a),
sproutf("H1 %s %s",cad1,cad3,cad2), outxxy(200,100,a),
}
//GRAFICA(GRAPH),
}
// Muestra el grafico
void GRAFICAHIP(mi tipo, float media, float dev, float vt)
{
GRAFICA(media,dev,3);
}
}
}
C dibujo;
{vt = 3; do { vt -- 3};
if (tipo == 1 || tipo == 3) { // Unilateral por la izquierda
int n = outxxy(media,3*dev,media+vt*dev);
G puntoxxy(media,media,media+vt*dev);
}
if (tipo == 2 || tipo == 3) { // Unilateral por la derecha
int n = outxxy(media,3*dev,media,3*dev+vt*dev);
G puntoxxy(media,media,media+vt*dev);
}
}
}
// Pone en pantalla el valor de alfa
void ALFA(mi tipo, float alfa)
{
char buf[10];
setcolor(15);
outxxy(50,SMALL_FONT,HORIZ_DIR,4),
sproutf(buf," %2f",alfa),
outxxy(50,90,buf),
if (tipo == 3) alfa = alfa/2,
sproutf(buf," %3f",alfa),
if (tipo == 1 || tipo == 3)
outxxy(125,35,buf),
if (tipo == 2 || tipo == 3)
outxxy(470,350,buf),
setcolor(GREEN);
}
// Escribe el valor de tablas encontradas, en el grafico
void VALORTABLAS(float vt, int n, float vt, signed int sev)
{
char buf[8];
setcolor(15);
line(100,415,350,415),
setcolor(SMALL_FONT,HORIZ_DIR,4),
outxxy(15,vt,buf),
setcolor(GREEN);
}
}
// FUNCIONES DE REGRESION
// Genera la tabla de mínimos cuadrados
void TABLAMINIMOS2(int n, float x[], float y[])
{
int i, n1, n2;
char s[40];
float suma, suma2, suma3, suma4,
float media, medy, sumy2,
suma=0, suma2=0, suma3=0, suma4=0, sumy2=0,
}
INGRAFICOS(),
MARCO(),
lineado("O"),
TITULOS("MÉTODO DE MÍNIMOS CUADRADOS",14,20,3),
setcolor(SMALL_FONT,HORIZ_DIR,4),
setcolor(15);
outxxy(30,60,"Mediante las siguientes ecuaciones",
outxxy(30,75,"se obtienen la pendiente y la inter.",
outxxy(30,90,"intersección en Y de la línea de regresión",
outxxy(30,105,"de la",
outxxy(30,120,"PENDIENTE DE LA LÍNEA",
line(195,157,311,157),
outxxy(140,157,"XY = nX^2",
outxxy(160,170,"Y = -----",
outxxy(140,188,"X^2 = nX^2",
line(193,188,199,188),
outxxy(30,210,"INTERSECCION EN EL EJE Y",
outxxy(122,235,"-",
outxxy(160,245,"a = Y - bX",
outxxy(220,250,"Medida de",
outxxy(240,60,"Cálculo de b",
outxxy(240,210,"Cálculo de a",
outxxy(SMALL_FONT,HORIZ_DIR,4),
outxxy(40,290,"a = intersección en el eje Y",
outxxy(40,305,"b = pendiente de la línea de regresión",
outxxy(40,320,"del mejor ajuste",
outxxy(40,335,"X = valores de la variable independiente",
outxxy(40,350,"Y = valores de la variable dependiente",
outxxy(40,365,"n = medida de los valores de la variable",
line(40,365,46,365),

```

```

autotaxy(40,3,80," independencia ").
autotaxy(40,395," y = media de los valores de la variable ").
line(40,395,46,395).
autotaxy(40,410," dependencia ").
autotaxy(40,425," y = numero de puntos de datos ").
sprndf(a,"% 2f - (%a)X + 20 (%b) 20",autaxy,n,media,medy).
autotaxy(379,110,5).
line(30,112,367,102+autax(a)+10,122).
sprndf(a,"% 2f - (%a)X + 20",suma2,n,media,*meda).
autotaxy(385,125,a).
sprndf(a,"% 3f - (%a) 3").
autotaxy(340,165,a).
sprndf(a,"% 2f - (%a) 3X + 30",medy,b,meda).
autotaxy(340,260,a).
sprndf(a,"% 4f - (%a) 4").
autotaxy(340,310,a).
autotaxy(702).
CLEARGRAPHICS.
prin dibayar().
}

// Clavea la grafica de regresion
void GRAFICAMINIMOS(int n, float x[], float y[])
{
int i,clavaw,clavew,key;
float mnav, minv, mmax, mash;
float escala, espaciado, ox, yx,a,b;
char s[20];

prin limpar();
lineado(0");
gotoxy(25,2); sprndf("METODO DE MINIMOS CUADRADOS");
gotoxy(20,10); sprndf("A los de la variable independiente ");
if (key = leer (aux , 57,10,5,0,99999)) == ESC) return;

float sumax=0,suma2=0,suma=0,sumy=0, meda=0, medy=0;

for (d = 0, d++, d++) {
sumax=sumax+x[d];          sumy=sumy+y[d];
suma=suma+x[d];          suma2=suma2+x[d]*y[d];
}
meda=suma/(float)n;      medy=sumy/(float)n;
a=(sumy-d*meda*medy)/(suma2-(n*meda*meda));
b=medy-a*b*meda;        yy = a + b*xa;

// hacer matrices y matrices
for (mash=0; mash<n; mash++) {
if (mash==x[d] mash==y[d]);
if (mash==x[d]) mash==x[d];
}
for (mnav=0; mnav<n; mnav++) {
if (mnav==y[d]) mnav==y[d];
if (mnav==y[d]) mnav==y[d];
}
if (mash==x) mash==x;
if (mnav==y) mnav==y;
if (mnav==a) mnav==a;
if (mnav==y) mash yy == 0) mnav==y;
if (mnav==0) mnav=1;
if (mnav==0) mash=1;

CLASSR(a,clavaw, &mash, &mnav);
CLASSR(a,clavew, &mnav, &mnav);

escala=(float)X25/mnav;
espaciado=(float)X305/mash;

if (EJER(1) == 1) return;
INGRAFICOS();
TITULO("DIAGRAMA DE DISPERSION", 14, 20, 3);
EJER(2);
setcolor(15);
line(140,120,140,370);
// dibujo las flechas
line(140,5,120,140,5,120);
line(140,120,10,140,5,120);
line(460,370,5,460,370,5);
line(460,370,5,470,370);

INGRAFICOS();
EJER(3);
lineado(0");
// dibujo los textos
line(140,5,120,140,5,120);
line(460,370,5,460,370,5);
line(460,370,5,470,370);
}

setcolor(15);
POINT_HORIZZ, DIR_5);
autotaxy(140,120,25," y ");
autotaxy(460,370,10," X ");
setcolor(15);
POINT_HORIZZ, DIR_4);
autotaxy(125,370,15," y ");

// Datos verticales
for (i = 1; i <= autax; i++)
line(140,5,370-escala*i*mmv,140,5,370-escala*i*mmv);
sprndf(a,"% 0f,mmv*");
autotaxy(130-2*autax(i),370-7*escala*i*mmv,a);
}

// Datos horizontales
for (i = 1; i <= autaw; i++)
line(140+espaciado*i*mmh,370,5,140+espaciado*i*mmh,370,5);
sprndf(a,"% 0f,mmh*");
autotaxy(140+espaciado*i*mmh,370,15,5);
}

for (i = 0; i <= n - 1; i++) // verificar los datos
DIBUJARPUNTO(140+espaciado*i,170-escala*y[i],4);
setcolor(4);

//linea de regresion
if (a == 0)
line(140,370-escala*n,140+espaciado*n*mmh,370-escala*(a+b*mmh));
else
line(140+((0-a*b)/escaciado,370,
140+espaciado*n*mmh,370-escala*(a+b*mmh));
if (ax)
setcolor(10);
line(140+espaciado*ax,370-escala*y);
line(140+espaciado*ax,370-escala*y);
line(133,370-escala*(a+b*mmh),147,370-escala*(a+b*mmh));
sprndf(a,"% 1f a+b*x", autotaxy(105+4*ax,140,365-escala*(a+b*mmh),a);
}
setcolor(10);
autotaxy(500,230,"");
setcolor(15);
autotaxy(500,120,"Ecuacion de regresion ");
autotaxy(500,135,"");
autotaxy(500,155,"");
autotaxy(500,160);
autotaxy(510,220,"Numero observaciones");
autotaxy(510,230,"Numero estimado");
autotaxy(510,240,"Linea del mejor ajuste");
line(500,245,500,245);
if (realms("Q27") == F2)
setcolor(0);
gotoxy(140,3);
CLEARGRAPHICS();
prin dibayar().

// Calcula el error estándar
void ERRORSTANDAR(int n, float x[], float y[])
{
int d;
char s[50];
float a,b;
float suma2=0,suma=0,sumy=0,meda=0,medy=0;

for (d = 0, d++, d++) {
suma=suma+x[d];          sumy=sumy+y[d];
suma2=suma2+x[d]*y[d];
sumy2=sumy2+y[d]*y[d];
}
meda=suma/(float)n;
b=(sumy-d*meda*medy)/(suma2-(n*meda*meda));
a=medy-d*b*meda;

INGRAFICOS();
EJER(3);
TITULO("ERROR ESTANDAR DE LA ESTIMACION",14,20,3);
setcolor(15);
POINT_HORIZZ, DIR_5);
autotaxy(140,120,25," y ");
autotaxy(460,370,10," X ");
setcolor(15);
POINT_HORIZZ, DIR_4);
autotaxy(125,370,15," y ");
}

```

Anexo D: Código fuente del sistema

```

outstatxy(30,60,"El error estándar de la estimación"),
outstatxy(30,80,"Mide la variabilidad o dispersión"),
outstatxy(30,100,"Mide los valores observados alrede"),
outstatxy(30,120,"der de la línea de regresión"),
outstatxy(30,150,"normala"),
outstatxy(120,190," $Y^2 = a \cdot b \cdot X \cdot Y$ "),
outstatxy(60,200,"Se = ....."),
outstatxy(150,210," $r^2 = ?$ "),
rstat(105,225, 245, 190),
outstatxy(30,250,"Donde"),
outstatxy(120,260,"Cromosoma"),
estatxy(5)(SMALL_POINT,HORIZ_DIR,4),
outstatxy(40,280,"X = Valores de la variable independiente"),
outstatxy(60,300,"Y = Valores de la variable dependiente"),
outstatxy(40,320,"a = Interseccion en Y"),
outstatxy(60,340,"b = Pendiente de la ecuación"),
outstatxy(60,360,"n = Numero de puntos de datos"),
sprstat("a",% 2f, (% 2f)(% 2f)(% 2f)(% 2f),sumxy2,a,sumxy,b,sumxy),
outstatxy(395, 92,a),
outstatxy(345,100,"Se = "),
low(390, 107, 390+int(round(h(a)*10,10)),
rstat(380, 120, 390+int(round(h(a)*10,10)),
sprstat("ad",2,"n"),
outstatxy(470,110,a),
sprstat("a",% 2f, % 2f, % 2f, sumxy2,a,sumxy,b,sumxy),
rstat(380, 190, 390+int(round(h(a)*10,160)),
outstatxy(395,162,a),
rstat(390, 177,390+int(round(h(a)*10,177)),
sprstat("ad",2,"n"),
sumxy2=sumxy2+a*sumxy,b*sumxy,
sprstat("a",% 2f, sumxy2),
outstatxy(395,232,a),
rstat(380, 260, 390+int(round(h(a)*10,230)),
outstatxy(405,250,a),
sprstat("ad",2,"n"),
sumxy2=sumxy2+(sumxy2-x)*sumxy2),
sprstat("a",% 2f, sumxy2),
outstatxy(340,310,"Se = "),
outstatxy(390,310,a),
rstat(380, 320, 380+int(round(h(a)*10,310)),
sprstat("a",% 2f, % 2f error estándar "segt(sumxy2)",
restat("02"),
CIBERACORAFICOSO,
para dibuajar),
)
// Dibuja los puntos de la linea de regresion
void DIBUJARPUNTO(x en x, use y, use color)
{
  setcolor(color),
  rectangle(x-1,y-1,x+2,y2),
  setcolor(15),
  floodfill(x,y,color),
}
// Lee los puntos para la grafica de regresion
void LEER_PUNTOB(float x[], float y[])
{
  int i,key,col,row,n=0,
  en sumxy, sumxyin,
  float dato;
  para leerkey()
  {
    intreaded("P"),
    getkey(30,2); sprstat("LECTURA DE LOS PUNTOS"),
    getkey(25,10); sprstat("Cuantos puntos son [1-25]?:"),
    int n = 1;
    if ((key = leer (dim,55,10, 2, 3, 25)) == ESC) return n;
  }
  para leerkey()
  {
    getkey(30,2); sprstat("LECTURA DE LOS PUNTOS"),
    CLEARCX(16,65,20,10,0),
    LINEACX(17,765,7),
    LINEACX(16,63,20),
    LINEA(48,648,20),
    getkey(32,5); sprstat("....."),
    getkey(32,5); sprstat("....."),
  }
}
getkey(16, 7); sprstat("....."),
row = 9, col = 25,
sumxyin = 1;
x[] = leerkey(1,8, 9, 6, 0, 9999),
while (sumxyin < n*2)
  if (sumxyin > sumxyin)
    { key = ponercol(col,row,6),
    if (key == ENTER)
      dato = modificar(col, row, 6, 0, 9999),
    }
  else if (sumxyin == sumxyin)
    dato = leerkey(col, row, 6, 0, 9999),
  else dato = modificar(col, row, 6, 0, 9999),
  switch (key)
  {
    case ENTER: if ((sumxyin % 2) != 0)
      n[1] = dato;
    else
      i = (sumxyin/2),
      y[i] = dato;
      if (sumxyin == sumxyin) // saca anterior y actual enter
        sumxyin++,
        rowsumxy++,
        break;
    case ESC: return ESC;
    case RIGHT: // flecha derecha
      if ((sumxyin % 2) != 0)
        sumxyin++,
        else if (sumxyin+19 == sumxyin)
          sumxyin++,
          break;
    case LEFT: // flecha izquierda
      if (sumxyin % 2)
        sumxyin--,
        break;
    case DOWN: // flecha abajo
      if (sumxyin - 2 <= rowsumxy)
        sumxyin -= 2;
        break;
    case UP: // flecha arriba
      if (sumxyin >= 2 && row > 8)
        sumxyin -= 2;
        break;
    case END: sumxyin = sumxyin,
        break;
    case HOME: sumxyin = 0,
        break;
  }
  if (sumxyin
  col = 2 + (int)(sumxyin/20) * 16 + ((sumxyin % 2) * 7),
  row = 9 + (int)(sumxyin % 20),
  } // while
  return n;
}
// Calcula la correlacion
void CORRELACION(float n, float x[], float y[])
{
  float r(0);
  float ab;
  float sumxy=0,sumxy2=0,sumxy2=0,sumxy=0,moda=0,modb=0,
  float r2,r;
  for (int d = 0, d <= n, d++)
  {
    float sumxy+=x[d],
    sumxy2+=x[d]*x[d],
    sumxy+=y[d],
    sumxy2+=y[d]*y[d],
    moda+=moda+(x[d]-moda)*(x[d]-moda),
    modb+=modb+(y[d]-modb)*(y[d]-modb),
    r2 += ((sumxy + b*sumxy - n*moda*moda)/(sumxy2 - n*moda*moda)),
    r = sqrt(r2);
  }
}

```

```

INGRAFICOS).
MARCO(X).
lineas(Y,2).
TITULOS ("CORRELACION SIMPLE",14,20,3).
seccion(15).
outstaty(30,60 "Los estadisticos han inventado don.").
outstaty(30,80 "medidas para describir la correla.").
outstaty(30,160 "son entre dos variables ").
outstaty(30,130 "EL COEFICIENTE DE DETERMINACION").
outstaty(133,164 "Y + b.XY - nY^2"), outstaty(80,170,"n^2").
outstaty(160,182 "n^2 - nY^2"), line(30,177,260,177).
line(245,160,250,160), line(213,181,218,181).
outstaty(30,220 "EL COEFICIENTE DE CORRELACION").
outstaty(80,250,"r = r^2"), outstaty(200,263,145,248).
outstaty(30,280 "Dadas "). outstaty(340,60 "Desarrolla ").
seccaoy(32,SMALL_FONT,HORIZ_DIR,4).
outstaty(40,300 "r = coeficiente muestral de determinacion").
outstaty(40,313 "a = interseccion en el eje Y").
outstaty(40,326 "b = pendiente de la linea de estimacion").
outstaty(40,339 " del mejor ajuste").
outstaty(40,352 "n = numero de puntos dados").
outstaty(40,365 "X = valores de la variable independiente").
outstaty(40,378 "Y = valores de la variable dependiente").
outstaty(40,392 "Y = media de los valores observados de la").
outstaty(40,405 " variable dependiente").
line(39,392,45,392).
apunt(a,"% 17% 10) + (% 17% 10) - (% 07% 10)",
a,summary,h,summary,n,ready*ready).
outstaty(370,160).
line(363,117,365+lineaD(x),5,117).
apunt(a,"% 20) (% 06% 20) summary 2, n, ready*ready).
outstaty(400,170).
outstaty(340,110,"Y = ").
apunt(a,"r = %4 coeficiente de determinacion",r2).
outstaty(340,160,1).
apunt(a,"n = %d",n).
outstaty(340,210).
r=175,230,340+lineaD(x)=10,208).
apunt(a,"r = %4 coeficiente de correlacion",r).
outstaty(340,260,1).
residual("2").
CIERRAINGRAFICOS).
prin dibuay).
}
// Determina el numero de clases horizontales y verticales en una grafica
void TABLASCAN("clase, float tam, float rmm")
{
float cla.
"clase="tam*(rmm).
if("clase"(rmm) == "tam") {
"clase="1.
tam="clase*(rmm).
cla=(float)clase.
while(cia=10) {
cla=cla*2.
rmm="tam*2.
clase=(cla).
"clase="(rmm).
tam="clase*(rmm).
}
}
// Cálculo del intervalo de confianza en proporciones
us PROPORCIONES.
{
float err,a,a1, factor.
float p,q=0.
float alfa.
char u[2].
any key,n=0,N=0.
do {
prin limpar().
gotxy(23,2).
apunt("CALCULO DEL INTERVALO EN PROPORCIONES").
gotxy(25,8). apunt("Probabilidad de éxito (p) ").
gotxy(25,12). apunt("Tamamto de la poblacion (N) ").
vmen avios("Introduzca solo decimales ").
if(key == leer(4p,54,8,8,0,1)) == "ESC" break.
if(key == leer(4N,54,10,5,0,99999)INT) == "ESC" break.
vmen avios("Introduzca 0 si es infinito").
do {key == leer(4u,54,12,5,0,99999)INT,1
while(key != ENTER : (N != 0 && N != 1)).
vmen avios(N).
if(N) {
factor = (float)(float)N.
else factor = 0.
q = 1 - p.
while (1/(1+ERR*PA(alfa,n)) - 2) |
if (n - 10) q = 2. bucaos(n)%(alfa,2)).
else if (float)(float)N | p = 1. bucaos(alfa,n,1).
if (factor = 0) u[1] = "Infinito".
err = sqrt(p*(1-p)/N).
} else {
p = (p)*q/(float)N.
a1 = (float)(N*n)/(float)(N-1).
err = sqrt(float(a1)*sqrt(a1)*2).
}
}
INGRAFICOS().
do {
if(N)
TITULO("CALCULO DEL INTERVALO
PARA LA PROPORCION INFINITA",14,20,3).
else TITULO("CALCULO DEL INTERVALO
PARA LA PROPORCION FINITA",14,20,3).
setseccaoy(SMALL_FONT,HORIZ_DIR,4).
outstaty(60,60 "DATOS ").
outstaty(450,60 "FORMULA ").
autoy(15).
apunt(a,"p = % 3f",p), outstaty(80,80,a).
apunt(a,"q = % 3f",q), outstaty(80,90,a).
apunt(a,"n = %d",n), outstaty(80,100,a).
apunt(a,"N = %d",N), outstaty(80,110,a).
apunt(a,"Z = % 2f",Z = "% 0f %4",1.96+alfa *100),outstaty(80,120,a).
if (factor = 0) u[1] = "N".
FORMULA_ERROR INFINITO(p, err,a,n,N).
else FORMULA_ERROR FINITO(p, err,a,n,N).
CLA_ESTIMACION(alfa,p,err,N).
lineaD("178",2,0).
key = readme("02",2).
if (key == F2 | key == F3)
ESPERAR(n, key).
while (key == ESC && key != ENTER).
CIERRAINGRAFICOS().
prin dibuay).
}
while (key != ESC).
return 0.
}
// Cálculo el tamaño adecuado de n en proporciones
us CALC_ADIC_PROPOR.
{
float a,m=0,n=0,p=0,q=0.
float alfa.
int N=0, key.
Tablas Z.
do {
prin limpar().
gotxy(15,2).
apunt("CALCULO DEL TAMARJO ADECUADO DE N EN
PROPORCIONES").
gotxy(20,8). apunt("El valor del error (e) ").
gotxy(20,12). apunt("El tamamto de la poblacion (N) ").
if(key == leer(4e,51,8,10,0,99999)) == "ESC" break.
if(key == leer(4N,51,10,10,0,1)) == "ESC" break.
}
}

```



```

vmen avaso("Introduzca 0 si es infinito"),
do {key = leer(&N, 31,12, 0, 0, 99999, INT), }
while(key != ENTER & (N != 0 && N%10 != 0)),
vmen avaso(),
q = 1-p,
while ( (LEERALFA(alfa,N)) != -2) {
  prin limpar(),
  x = z.bussaco(0-5*(alfa/2)),
  if (N) { r=norma
    n=(x**2*p/q)*(erf+er),
    gotxy(23,2).epuraf("FORMULA DEL CALCULO ADECUADO (n*)",
    gotxy(23,3).epuraf(" PROPORCIONES INFINITAS"),
    gotxy(30,0).epuraf(" q a **2"),
    gotxy(23,4).epuraf(" n = -----"),
    gotxy(32,8).epuraf(" z"),
    gotxy(30,13).epuraf(" %3 f * % 3f * % 3f * p,q,e**2),
    gotxy(23,13).epuraf(" n = -----"),
    gotxy(35,14).epuraf(" %4 1f erf+er),
    gotxy(23,18).epuraf(" n = % 4f",n),
  }
} else {
  n=(1+er*p*q)/(float(N)*(x**2*p/q)+((float(N)-1)*er+er)),
  gotxy(23,3).epuraf("FORMULA DEL CALCULO ADECUADO (n*)",
  gotxy(23,3).epuraf(" PROPORCIONES FINITAS"),
  gotxy(30,0).epuraf(" p q a **2"),
  gotxy(23,7).epuraf(" n = -----"),
  gotxy(30,8).epuraf(" p q a **2*(N-1) e**2"),
  gotxy(30,13).epuraf(" % 3f*3f*3f*3f*p,q,e**2,N),
  gotxy(23,13).epuraf(" n = -----"),
  gotxy(30,14),
  epuraf(" %4 3f*3f*3f*(1+er) %4 3f*p,q,e**2,N,erf+er),
  gotxy(23,18).epuraf(" n = % 4f",n),
}
}
key = realim("12"),
} while (key != ESC),
return 0,
// Cálculo del intervalo de confianza en medias
un ESTDMMABIDIAS,
int key= ENTER,U,N,
float devst, media, varia,art[35],
float sum, sumas,
do {
  prin limpar(),
  gotxy(24,2).epuraf("CALCULO DEL INTERVALO PARA MEDIAS"),
  gotxy(20,8).epuraf("El tamaño de la muestra (n)"),
  if ((key = leer(&n,40,8,6,1,99999,INT)) == ESC) break,
  gotxy(20,10).epuraf("Tamaño de la población (N)"),
  vmen avaso("Introduzca 0 si es infinito"),
  do {
    key = leer (&N,40,10,6,0,99999,INT), }
  while(key != ENTER (N != 0 && N%10 != 0)),
  vmen avaso("Introduzca 0 si es descomocida"),
  gotxy(20,13).epuraf("La desviación en de ( )"),
  if ((key = leer(&devst,40,12,10,0,9999999)) == ESC) return,
  vmen avaso(),
  cha res = "n",
  if (devst) {
    if (n <= 30) {
      gotxy(27,20).epuraf("son datos sueltos [N/N]"),
      res = LEER(S3, 20, "Introduzca solo una 3 ó N"),
    }
    if (res == "3") {
      prin limpar(),
      gotxy(28,2).epuraf("CALCULO DEL INTERVALO PARA MEDIAS"),
      for (i = 0; i <= 35; i++) art[i] = 0,
      if (LEERDATOS(a,0,0,0,art)) == ESC) return 1,
      sum=0,
      for (i = 0; i < n; i++) sum += num+art[i],
      media = sum/(float)n,
      sumas=0,
      for (i = 0; i < n; i++) sumas += pow(abs(art[i]-media),2),
      varia = sumas/(float)n,

```

```

aprend(a,"% 5f %n",
GRAB(400,150,515,140),
lineas(1078,2,1),
key="calculo(0278)",
if(key=="F2",key=="F3")
ESPH(RA(30, key),
jwsh( key + "ESC" && key + "ENTER"),
"CHERRA(IRA(1078))",
pen dibuja(1),
}
}
// Cálculo del intervalo de confianza para la media cuando la desviación es
+ desnocida
int DESV_DESNOCIDADA(int n, float var, float media)
{
float alf, err, z, factor;
char s[20];
int key;

prin limpar();
while( (ERR=ALFA(alf,n)) != -2) {
if (n<30)
z=Z_buscad(0.5-(alf/2));
else {
T_buscad(alf,n,1);
}
if (N)
factor=(float)(float)n;
if (N && factor < 0.05)
err=(sqrt(var/a))*(sqrt(n)) * sqrt((float)(N-n)/(float)(N-1))*z),
else
err=(sqrt(var/a))*sqrt(n) * z),
INTORAF(ONS);
do {
n=(N+TITULOS+"CALCULO DEL INTERVALO PARA LA MEDIA
FINITA",14,20,3);
sete TITULOS+"CALCULO DEL INTERVALO PARA LA MEDIA
FINITA",14,20,3);
write(ave (SMALL_FONT)HORIZ_DIR,4),
outtextxy(40,60,"D(1,038)",
outtextxy(40,60,"ERR(11,4)",
outtextxy(250,60,"DESVIACION DESNOCIDA",
autolim(15),
outtextxy(80,80),
outtextxy(80,90),
outtextxy(80,100),
aprend(a,"N: %i",100,alf*100),
if (N)
aprend(a,"N = %d",N),
outtextxy(80,120),
outtextxy(497,74,""),
line(490,510,500),
outtextxy(500,95,"n"),
outtextxy(490,120),
line(483,115,520,135),
outtextxy(497,140),
}
}
else
outtextxy(520,83," z"),
aprend("z: %3f",z),
}
else {
if (n<30) outtextxy(570,83," z"),
else outtextxy(570,83," t"),
outtextxy(545,80,"N="),
outtextxy(545,94,"N="),
line(540,565,565,520),
aprend(a,"%d %d",N,n),
raiz(535,150,540-textwidth(s)+3,120),
line(540,135,545+textwidth(s),135),
int la textwidth(s),
aprend(a,"% 3f",z),
outtextxy(551+la,130),
}
}

```

```

aprend(a,"%nd-1%N",
outtextxy(520,130,""),
}
}
aprend(a,"% 5f %n",
GRAB(ERITAMA(1078),media, err,"M"),
key="resulca(0278)",
if(key=="F2",key=="F3")
ESPH(RA(30, key),
jwsh( key + "ESC" && key + "ENTER"),
"HERMADRA(1078)",
pen dibuja(),
}
return 0;
// Cálculo del tamaño adecuado de n en medias
int CALA_ADEU_MIBER(1)
{
float z, err, t(n), t(a), dev, o,alf,
int key, N, 0;
do {
prin limpar();
getint(20,2);
aprend("CALCULO DEL TAMAÑO ADECUADO DE n EN MEDIAS"),
getint(21,3);
aprend("El valor del error es: ");
getint(21,10);
aprend("El valor de la desviación ");
getint(21,12);
aprend("El tamaño de la población ");
if (key=="ESC" && err > 0.320001) "ESC" return 1;
if (key=="ESC" && dev > 0.320001) "ESC" return 1;
venc avisa("Introduzca un n en su límite");
do {
key=N;
while(key=="ENTER" || N<=0 || dev<=0 || N<1);
venc avisa(1);
while( (ERR=ALFA(alf,n)) != -2) {
z=Z_buscad(0.5-(alf/2));
if (N) {
n=(dev*dev*z*z/err)+err;
prin limpar();
getint(23,5);
aprend("FORMULA DEL CALCULO ADECUADO (n)"),
getint(32,6);
aprend("MEDIDAS INFINITAS"),
getint(30,9);
aprend(" + z"),
getint(25,10);
aprend("n = ....."),
getint(30,11);
aprend(" z"),
getint(30,15);
aprend(" + %3f * %3f * dev*dev * z"),
getint(25,12);
aprend("n = ....."),
getint(30,17);
aprend(" + %3f * err * err"),
getint(25,20);
aprend("n = %d",n),
}
else {
t=(dev*dev*z*z/err)*t+(N-1)*err*err),
n=(dev*dev*z*z/err*(float)N)+((N-1)*err*err),
prin limpar();
getint(23,3);
aprend("FORMULA DEL CALCULO ADECUADO (n)"),
getint(32,4);
aprend("MEDIDAS FINITAS"),
getint(30,7);
aprend(" + t"),
getint(25,8);
aprend("n = ....."),
getint(30,9);
aprend(" t = (N-1) * z"),
getint(30,13);
aprend(" + %3f * %3f * dev*dev * z"),
getint(25,14);
aprend("n = ....."),
getint(30,15);
aprend(" + %3f * err * err"),
getint(25,19);
aprend("n = %d",n),
}
resulca("2");
}
while (key!="ESC");
return 0;
}
// Dibuja la grafica de estimacion
void GRA_BSTIMACI( float alf, float z, float media, float err, char s[])
{
char s[30];
Otra(OF(media,1,3));
}

```

Anexo D: Código fuente del sistema

```

C) dibujo(s),
C) nombre(media, 3, media, z),
C) poncarvacion(media, z, media + z),
C) promedio(media, media),
aprox1a, "% 2f" alfa),
aprox1a, "% 3f" alfa(float)2),
aprox1a, "% 3f" alfa(float)2),
aprox1a, "% 4f" - " % 4f", media-err, col, media+err),
outstext(255, 450, a),
}

// Escribe la fórmula del error infino
void FORMULA_ERROR_INFINITA(float p, float q, float err, int n, float z)
{
char q[20];

raiz(480, 110, 510, 80),
outstext(490, 80, "p q"),
line(485, 95, 510, 95),
if ((n-3) outstext(515, 90, " z"),
else outstext(515, 90, " i"),
raiz(480, 160, 560, 130),
outstext(490, 130, a),
line(485, 145, 565, 145),
outstext(515, 130, a),
outstext(580, 140, a),
outstext(450, 180, a),
}

// Escribe la fórmula del error finito
void FORMULA_ERROR_FINITA(float p, float q, float err, int n, int N, float z)
{
char q[20];

raiz(450, 100, 480, 70),
outstext(420, 85, "p q"),
outstext(460, 90, "i"),
outstext(520, 90, "N"),
line(515, 90, 540, 90),
if ((n-3) outstext(550, 82, " z"),
else outstext(550, 82, " i"),
raiz(450, 155, 535, 128),
outstext(460, 128, a),
line(460, 140, 535, 140),
outstext(490, 142, a),
aprox1a( "%4d%4d", N, n),
raiz(540, 152, 555, textwidth(a)+5, 128),
line(535, 140, 555, textwidth(a)+2, 140),
int la = textwidth(a),
aprox1a( "%4d", n, N),
aprox1a( "% 3f", z),
aprox1a( "% e - % 5f", err),
}

// Lee los datos de entintacion
int LERIDA(TC)(int n, int numfact, int numfin, float numf[])
{
int col, row;
int key = ENTNR;

col = 20 + ((numfact % 7) - 1) * 9,
row = 9 + ceil(numfact / 7),
while (numfin < n) {
if (numfact == numfin) {
key = poncol(col, row, B),
if (key == ENTNR)
if (key == leer (&num[numfact][col, row, 8.0, 999999, FLOAT, 1, 1]),
else if (numfact == numfin)
if (key == leer (&num[numfact][col, row, 8.0, 999999, FLOAT, 1, 1]),
switch (key) {
case ENTNR: if (numfact == numfin)
numfact++,
break;
case ESC: return ESC;
}
}
}

```

```

case RIGHT: if (numfact == numfin) // flecha derecha
numfact++;
break;
if (numfact > 0) // flecha izquierda
numfact--;
break;
case DOWN: if (numfact = 7 <= numfin) // flecha abajo
numfact = 7,
break;
if (numfact == 7) // flecha arriba
numfact = 7,
break;
case END: numfact = numfin,
break;
case HOME: numfact = 0,
break;
// switch
col = 20 + ((numfact % 7) - 1) * 9,
row = 9 + ceil(numfact / 7),
// while
return ENTNR;
}

```

FUNCIONES COMUNES

```

//FUNCIONES DE LA GRAFICA NORMAL.
#define pi 3.1415927
#define e 2.71828181

class Grafica {
private:
float media, desv, ncv,
double factor,
int ndeav,
float sta, fin,
public:
Grafica(float m, float d, int nd=3),
void dibujo(),
void poncarvacion(float val1, float val2),
void ponpromedio(float val1, float val2,
void nombrar(float val1, float val2),
float calcula(float valor),
},
void intercalado(float &val1, float &val2),
// Constructor
Grafica: Grafica(float m, float d, int nd)
{
float fe,
media = m,
desv = d,
ndcv = nd,
// Parametros para graficar
gu = media - desv * ndcv,
fu = media + desv * ndcv,
Dx = 1 / sqrt(2 * pi * desv * desv),
factor = (1 / (2 * pi * desv * desv)),
int = (1 / (2 * pi * desv * desv)) * 400, // 400 pixels de la grafica
float Grafica calcula(float valor)
{
float z,
z = (valor - media) / desv,
return (0),
}
}

```

```

// Escribe las dimensiones en la grafica
void Grafica_posdevariacion(float val1, float val2)
{
    char a[15];
    int col, int1, fin1;
    float x;

    setcolor(15);
    settextstyle(SMALL_FONT,HORIZ_DIR,4);
    outtextxy(530,400,"Z");

    x = calcula(val1);
    sprintf(a,"%2f",x);
    col = 120+((val1-uv)*uv);
    line(col+1,403,col+1,407);
    outtextxy(col+(textwidth(a)/2),407,a);
    int1 = col+(textwidth(a)/2);
    fin1 = col+(textwidth(a)/2);

    x = calcula(val2);
    sprintf(a,"%2f",x);
    col = 120+((val2-uv)*uv);
    line(col+1,403,col+1,407);
    if ((int1 + col+(textwidth(a)/2) + 3) < (fin1+3 + col+(textwidth(a)/2)))
        outtextxy(col+(textwidth(a)/2),407,a);
}

// Escribe los valores s en la grafica
void Grafica_posvaloresa(float val1, float val2, int epc)
{
    char a[15];
    int mod1, mod2, mod3;
    int int1, int2, int3;
    int fin1, fin2, fin3;

    setcolor(15);
    settextstyle(SMALL_FONT,HORIZ_DIR,4);

    line(120,420,520,420);
    if (epc)
        outtextxy(530,415,"P");
    else
        outtextxy(530,415,"X");
    sprintf(a,"%2f",mod1);
    mod1 = 120+((mod1-uv)*uv);
    outtextxy(mod1+(textwidth(a)/2),422,a);
    fin1 = (mod1 + textwidth(a)/2);

    // Valor 1
    sprintf(a,"%2f",val1);
    line(mod2,417,mod2,422);
    fin2 = (mod2 + textwidth(a)/2);

    // Valor 2
    sprintf(a,"%2f",val2);
    line(mod3,417,mod3,422);
    fin3 = (mod3 + textwidth(a)/2);

    if (val1 != mod1) {
        if (int2 > fin1+3 & fin2+3 < int1)
            outtextxy(int1,422,a);
        else
            outtextxy(int2,432,a);
    }

    // Valor 3
    if (val2 != mod3) {
        sprintf(a,"%2f",val2);
        line(mod3,417,mod3,422);
        fin3 = (mod3 + textwidth(a)/2);

        if ((int3 - fin1 - 3 & fin3+3 < int1) && (int3 > fin2+3 & fin3+3 < int3))
            outtextxy(int1,422,a);
        else
            if (int2 < fin3+3 & fin2+3 < int3)
                outtextxy(int3,442,a);
            else
                outtextxy(int1,432,a);
    }
}

// Dibuja la grafica
void Grafica_dibujar()
{
    double ax, fx, bx;
    float x;
    int col;

    fx = bx - 1 (sqrt(2)*pi*dev*dev);
    x = uv;
    col = 120;

    // Grafica
    setcolor(15);
    line(col,405,col+400,405);
    while (x < fin1)
        ax = pow(a+media,2)*2*dev*dev;
        ax = pow(e,-ax);
        fx = bx*ax;
        col += 1;
        plotax(col,400-factor*fx,10);
        x+=uv;
    }

// Selecciona una region de la grafica
void Grafica_ambresar(float val1, float val2)
{
    double ax, fx, bx;
    float x;
    int col;

    if (val1 < val2) intercambia(val1, val2);
    bx = 1(sqrt(2)*pi*dev*dev);
    x = val1;
    col = 120+((val1-uv)*uv);

    // Selecciona
    setcolor(2);
    while (x < val2) {
        ax = pow(a+media,2)*2*dev*dev;
        ax = pow(e,-ax);
        fx = bx*ax;
        col += 1;
        line(col,400-factor*fx, col, 400);
        x+=uv;
    }

// Intercambia los valores que recibe
void intercambia(float &val1, float &val2)
{
    float temp = val1;
    val1 = val2;
    val2 = temp;
}

// FUNCIONES DE GRAFICOS
// Inicializa los graficos en memoria
void INICIALIZAR()
{
    int controlador,modos,error;

    controlador=DEFECTO;
    modos=graficos(2000);
    error=graficos(&controlador,&modos,"");
    error = graficos(0);
    if (error != 0) {
        printf("Error en graficos ");
        getch(30,10); printf("Pres. graficos=error");
        return("02");
    }
    exit(1);
}

```

Anexo D: Código fuente del sistema

```

}
setfillstyle(1,2);

// Cambia el estado de la pantalla, de texto a gráfico
void INIOGRAFICOS()
{
  setgraphmode(getgraphmode());
}

// Regresa a pantalla tipo texto
void CHIRRAGRAFICOS()
{
  rastrocremode();
}

// Quita de la memoria los controladores gráficos
void CHIRRAGRAFICOSII()
{
  rastrocremode();
  closegraph();
}

// Escribe el título de las gráficas
void TITULOS(char cad[], int color, int ran, int tam)
{
  int largo;

  setcolor(color);
  settextstyle(SANS_SERIF_FONT,HORIZ_DIR,tam);
  largo=tam*ran/cad;
  outtextxy(5*0,largo/2,ran,cad);
  outtextxy(5*SMALL_FONT,HORIZ_DIR,5,
  setcolor(CRREEN);
}

// Dibuja el símbolo de números
void numm(int n, int y, int tipo)
{
  outtextxy(SMALL_FONT, HORIZ_DIR,4);
  line (x, y, x+15, y);
  line (x, y, x+5, y+10);
  line (x+5, y+10, x, y+20);
  line (x, y+20, x+15,y+20);
  if (tipo)
    outtextxy(x+5,y+15,"N");
  else outtextxy(x+5,y+15,"n");
  outtextxy(x,y+20,"1");
  settextxy(SMALL_FONT, HORIZ_DIR,5);
}

// Dibuja el símbolo pequeño de números
void numm(int n, int y)
{
  line (x, y, x+8,y);
  line (x, y, x+4,y+7);
  line (x+4,y+8, x, y+14);
  line (x, y+14,x+8,y+14);
}

// Dibuja el símbolo pequeño de números
void numm(int n, int y)
{
  line (x, y, x+5,y);
  line (x, y, x+2,y+8);
  line (x+2,y+8, x, y+14);
  line (x, y+8, x+5,y+8);
}

// Dibuja la raíz cuadrada
void raiz(int x1, int y1, int x2, int y2)
{
  line (x1+5,y2,x2,y2);
  line (x1,y1,x1+5,y2);
  line (x1,y1,x1-3,y1-y2,y2);
}

```

```

// Dibuja el signo de la derivación estándar poblacional
void signon(int col, int ran)
{
  outtextxy(col,ran,"σ");
  line(col+5,ran+7,col+10,ran+7);
}

// FUNCIONES DE LECTURA
// Lee una "S" o "N"
char LEEB(int x,int y, char *m){
  char resp="";
  do {
    int cont(6,7);
    getche(x,y);
    res=tolower(getche());
    if (res != 'S' && res != 'N'){
      getchy(x,y);
      sprintf("ser,%u",
      system("clear"));
      printf("\n");
    } while (res != 'S' && res != 'N');
    return(res);
  }

// Esta función lee un número float o int, regresa la última tecla presionada.
int lee(float *var, int n, int y, int lon, float min,
float max, int formato, int escape, int mod)
{
  int punto=0, fin=0, cont=0, valide=0;
  float valor=0;
  char s[12];

  setcolor(0);
  if (formato == FLOAT && mod){
    sprintf(s,"%f",*(float *)var);
    if (len(s)>12);
    if (len(s)) strcpy(s," "); fin=0;
  } else {
    // Diagrama como
    int i= strlen(s)-1, punto=0;
    while ((s[i]!='\0' && s[i]!='.') && punto == 0){
      if (s[i] == '\n')
        sprintf(s,"");
      if (s[i] != '\n'){
        sprintf(s,"%f",
        punto = 1;
      }
    };
    fin = strlen(s)-1;
    if (cont(s,".") punto = 1;
  } else if (mod) {
    for (i=0; i < lon; i++){
      sprintf(s,"%i,%i,%i,%i",s[i]);
      if (s[i] != '\n') fin++;
    }
    if (len(s)>0);
    if (fin != 0) fin = strlen(s)-1;
    if (cont(s,".") punto = 1;
  }

  if (formato == INT) {
    sprintf(s,"%d",*(int *)var);
    if (len(s)>12);
    if (len(s)) strcpy(s," "); fin=0;
    else fin = strlen(s)-1;
  }
}

```

Anexo D: Código fuente del sistema

```

describe(s,lon,x,y),
gotoxy(1,1),
do {
  key = getch(),
  switch (key) {
  case RIGHT: if (fin == 0 || escape) salida = 1;
              else if (cont == fin || (cont) != "0") cont++;
              break;
  case LEFT:  if (fin == 0 || escape) salida = 1;
              else if (cont == 1) cont--;
              break;
  case UP:   if (fin == 0 || escape) salida = 1; break;
  case DOWN: if (fin == 0 || escape) salida = 1; break;
  case ENTER: if (fin == 0) {
               s[fin-1] = "\r",
               if (formato == 2)
                 valor = mod(f),
                 else valor = atoi(s),
                 if (valor == min || valor == max) {
                   quitacolor(s,lon,x,y),
                   prin activar(0),
                   if (formato == FLOAT)
                     "%(cont %f) = valor",
                     return key;
                   else *(int *)var = valor,
                     return key;
                 }
               } else {char m[25];
                 punto = 0,
                 aprand(m,"Valor fijo del rango * 0f * %s int",numran),
                 vran = strtoc(m),
                 fin=0, cont=0,
                 s[0] = "\r",
                 describe(s,lon,x,y),
               }
              }
  } break;
  case BACKSPACE: if (cont==0)
                  s[fin-1] = "\r",
                  cont--;
                  if (s[cont] == " ") punto = 0,
                  supr(s,cont),
                  describe(s,lon,x,y),
                  break;
  case DEL: if (cont == fin) {
             s[fin-1] = "\r",
             if (s[cont] == " ") punto = 0,
             supr(s,cont),
             } else {s[fin] = "\r", fin++, cont++;
             describe(s,lon,x,y),
             break;
  case HOME: if (fin == 0 || escape) salida = 1;
             else cont = 0; break;
  case END:  if (fin == 0 || escape) salida = 1;
             else cont = fin; break;
  }
  if (key == " " || punto == 0 || cont == lon-1 || formato == 2) {
    punto = 1,
    s[cont] = key;
    aprand("\a",key);
    if (cont == fin || fin == lon-1) {fin++, cont++;
    }
    else if (cont == lon-1) cont++;
  }
  if (modget(key) || fin == lon-1) {
    if (s[cont] == " ") punto = 0,
    s[cont] = key;
    aprand("\a",key);
    if (cont == fin || fin == lon-1) {fin++, cont++;
    } else if (cont == lon-1) cont++;
  }
  }
  gotoxy(cont+1,1),
  while (key != ESC || salida),
  quitacolor(s,lon,x,y),
  prin activar(0),
  if (formato == FLOAT)
    "%(cont %f) = valor",
    else *(int *)var = valor,
    return key;
}

```

```

void describe(char s[12], int lon, int x, int y)
{
  window(s,v,x,y,lon-1,y);
  clear();
  gotoxy(1,1),
  write(s,v,x,y,lon,y),
  aprand("\a",s);
}

void suprara(char s[], int pos)
{
  for (int i = pos, s[i] = "\r", i++)
    s[i] = s[i+1];
}

void quitacolor(char s[10], int lon, int x, int y)
{
  coloroff();
  describe(s,lon,x,y);
}

int poncolor(int x, int y, int lon)
{
  char s[15];
  int key;

  int _serv(-1,1),
  _serv(s[0]);
  for (int i = 0, j = lon, i++)
    getch("%c",y,x,y,s[i],s[i]),
    s[lon] = "\r",
    describe(s,lon,x,y);

  // apagar el teclado
  do {
    key = getch();
  } while (key != ENTER || key != ESC || key != UP || key != DOWN ||
  key != LEFT || key != RIGHT || key != HOME || key != END);
  int _serv(6,7);
  return key;
}

// Inverte el color
void invcolor()
{
  textbackground(3);
  textcolor(1);
}

// Rige el color normal
void coloroff()
{
  textbackground(1);
  textcolor(15);
}

// FUNCIONES DE IMPRESION
#define K_ESC '\a'
#define LFT 0
#define LPT 1
#define pr_puls(x) hsonprnt(0,x,LLPT1)
#define MODIFICAR 1
#define LECTURA 0
#define FLOAT 2
#define INT 1

static void brrimage(int Nbytes)
{
  register int n1, n2,
  n2 = Nbytes - 2,
  n1 = Nbytes - (n2 - 2);
}

```

```

    prin_puls(K, ESC),
    prin_puls(" ");
    prin_puls(0);
    prin_puls(1);
    prin_puls(2);
}

static unsigned char getScifid(int x, int y)
{
    unsigned char firstPuls;

    firstPuls = (getpuls(x, y+3)==0)? 0x80:
    firstPuls = (getpuls(x, y+2)==0)? 0x40:
    firstPuls = (getpuls(x, y+1)==0)? 0x20:
    firstPuls = (getpuls(x, y)==0)? 0x10:
    firstPuls = (getpuls(x, y+3)==0)? 0x08:
    firstPuls = (getpuls(x, y+2)==0)? 0x04:
    firstPuls = (getpuls(x, y+1)==0)? 0x02:
    firstPuls = (getpuls(x, y) ==0)? 0x01;

    return firstPuls;
}

int printImage(int left, int top, int right, int bottom)
{
    int x, y, width, height;

    width = right-left;
    height = bottom-top;

    // Inicializar línea de espacio de 7/72"
    prin_puls(K, ESC),
    prin_puls(" ");

    for (y=0, y<height, y+=8)
    {
        bitImage(width);

        for (x=0, x<width, x++)
            prin_puls(getScifid(x, y));

        prin_puls("\n");
    }

    return 0;
}

```