



**UNIVERSIDAD NACIONAL**  
**“SANTIAGO ANTUNEZ DE MAYOLO”**

---

**ESCUELA DE POSTGRADO**

**ANALISIS Y DISEÑO DE UN SISTEMA DE CONTROL  
PARA LA CONSOLIDACIÓN DE SUELOS**

**Tesis para optar el grado de maestro  
en Ciencias e Ingeniería  
Mención en Computación e Informática**

**MARCOS ZAMBRANO FERNÁNDEZ**

**Asesor: Dr. JESUS ESPINOLA GONZALES**

**Huaraz - Perú  
2008**

**N° Registro: T0109**



ACTA DE SUSTENTACION DE TESIS DEL Br. MARCOS ZAMBRANO FERNANDEZ. (Resolución de consejo de la EP-UNASAM N° 153-2007-UNASAM-EPG)

En el local del Auditorio de la EP de la UNASAM, a los 18 días del mes de Enero de 2008, se reunió el Jurado Evaluador de la Tesis "Análisis y Diseño de un sistema para el Control de la Consolidación de Suelos", a las 19:15 horas para su sustentación por parte del Br. Marcos Zambrano Fernandez, en presencia del Jurado Presidido por el Mag. Marcos Silva Lindo, actuando como Secretario el Mag. Carlos Antonio Reyes Paraja y como vocal el Dr. Jesús Edilberto Espinola Gonzales; a continuación el Presidente del Jurado le concedió al Bachiller 30 minutos para que realice la defensa de la respectiva tesis, al final del cual se procedió a efectuar las respectivas preguntas

Juego el Jurado procedió a realizar la calificación correspondiente de conformidad con la TABLA de CALIFICACION del Reglamento de la EPG. El RESULTADO en promedio fue de APROBADO CON MENCION con el calificativo de 17,0 (diecisiete).

Siendo las 20:00 horas del mismo día, se concluyó con el proceso de Evaluación; en FE de CONFORMIDAD de la presente Acta, firman los miembros del Jurado.

Mg. Marcos Silva Lindo  
Presidente

Mg. Carlos Antonio Reyes Paraja  
Secretario

Dr. Jesús Edilberto Espinola Gonzales  
Vocal.

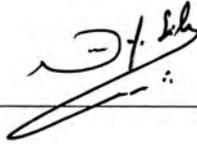
Unidad e Ingeniería  
Computación e Informática



## MIEMBROS DEL JURADO

*Magister Marco Antonio Silva Lindo*

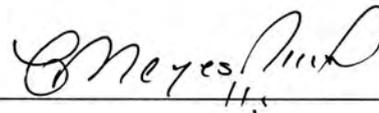
Presidente



---

*Magister Carlos Antonio Reyes Pareja*

Secretario



---

*Doctor Jesus Edilberto Espinola Gonzáles*

Vocal



---

ASESOR

*Doctor Jesús Espinola González*

## **AGRADECIMIENTO**

- A la Universidad Nacional Santiago Antúnez de Mayolo por apoyarme en mi perfeccionamiento permanente.
- A mi asesor Dr. Jesús Espinola, por su apoyo y paciencia en el desarrollo de la presente tesis.
- Agradezco profundamente y con mucho cariño a mis colegas que me apoyaron moralmente para continuar en culminar el trabajo de esta tesis.

A mi padre y guía:  
Adrián Zambrano Fuentes.

## INDICE

	<b>Página</b>
Resumen	
Abstract	
I. INTRODUCCIÓN .....	1
Objetivos .....	2
II. MARCO TEÓRICO .....	3
2.1. Antecedentes .....	3
2.2. Bases teóricas .....	3
Clasificación de Suelos .....	3
Consolidación .....	7
Tasa de Consolidación .....	10
Capacidad de Carga .....	14
Esfuerzo Vertical .....	20
Asentamientos .....	24
III. MATERIALES Y METODOS .....	31
3.1. Tipo y Diseño de la Investigación .....	31
3.2. Instrumentos de Recoleccion de la Información .....	31
3.3. Plan de Procesamiento .....	31

IV. RESULTADOS .....	36
V. DISCUSIÓN .....	73
VI. CONCLUSIONES .....	83
VII. RECOMENDACIONES .....	85
VIII. REFERENCIAS BIBLIOGRAFICAS .....	86
ANEXO .....	88

## RESUMEN

El desarrollo de la computación e informática ha permitido en la actualidad que muchas de las áreas del conocimiento humano se automaticen, entre ellos, la Mecánica de Suelos. La aplicación de los conceptos teóricos del cálculo diferencial e integral ha permitido en este trabajo el diseño de un sistema de control para la consolidación de suelos considerando el comportamiento de los suelos arcillosos y con la ayuda de los lenguajes de programación, herramientas informáticas se ha logrado la implementación computacional de un sistema de control que coadyuvará a la toma de una decisión adecuada en cuanto a la consolidación de suelos arcillosos.

Se ha implementado un sistema de control, para ello se ha transformado el proceso de consolidación de suelos usando la teoría general de Terzaghi, en un esquema modular para prever las capacidades de determinados suelos y sus consolidaciones con respecto al tiempo.

Se desarrollan aporte a la teoría del esfuerzo vertical, desarrollada por Bousinessq, incorporando transformaciones lineales, para cargas puntuales dirigidas a superficies que soportan cargas debido a una cimentación.

Finalmente, se presenta los resultados referentes a la consolidación de suelos mediante la ejecución del sistema de control a un problema real.

**Palabras clave:** Teoría General de Terzaghi, Consolidación de Suelos, Lenguajes de Programación.

## ABSTRACT

The development in the computer and informatic technology has permitted, at the present time many of human knowledge make automatic, between them, Mechanics of Grounds. The application of the theoretical concepts from the differential calculus, has permitted at the present work, the design of a control system for the consolidation of the grounds, considering the behaviour the clay grounds and with help of programming language, computer tools was gotten the development of a control system that it will collaborate to make a decision on the consolidation in clay grounds.

It was created a control system, and for this work, the process of consolidation was transformed, using the General Terzaghi's theory, to a modular scheme to foresee the charge capacities of specific grounds and their consolidations considering time.

Certain contributions to vertical enforce theory was developed, theory started by Bousinessq, adding linear transforming, in punctual charges, through out surfaces, whose, support charges by a consolidation.

Finally, the results are showed considering the consolidation of grounds trough out the execution of the control system, based on a real problem.

**Key Words:** General Terzaghi's Theory, Consolidation of the Grounds, Programming Languages.

## I. INTRODUCCIÓN

Los análisis y cálculos a los que se someten los suelos y la ausencia de una libre automatización para los mismos es lo que me motivó a realizar la presente tesis titulada: Diseño de un Sistema de Control para la Consolidación de Suelos.

En este trabajo se realiza el desarrollo de un sistema encargado de realizar el análisis y cálculos que involucran el fenómeno de consolidación en suelos que están conformados por depósitos de arcilla.

En el marco teórico se tocan temas relacionados a las herramientas de la mecánica de suelos necesarias para el desarrollo del sistema de control, en primer lugar se hacen mención a las herramientas matemáticas para el esfuerzo, la relación de vacíos y luego los relacionados con las cimentaciones que involucran el esfuerzo vertical para la consolidación de suelos.

Dentro de los métodos que se utilizan son el diseño modular para un sistema de control estableciendo la forma de realizar los análisis y cálculos, luego se dan los pasos para estructurar el algoritmo.

Posteriormente se hacen mención al diseño software, planteando, el ambiente para la ejecución de aplicaciones destinadas a resolver problemas descritos por algoritmos y se termina con la programación en el lenguaje de programación Java.

Los resultados se muestran en la forma de un sistema de control para la consolidación de suelos denominado JGROUND, luego se presentan los módulos que interactúan en JGROUND, finalmente se muestran las relaciones o conexiones que existen entre los módulos con el núcleo del sistema de control.

## **Objetivos del estudio**

### **Objetivo general:**

El objetivo general es el diseño y la implementación de un sistema de control para la consolidación de suelos.

### **Objetivos específicos**

1. Analizar la consolidación de los suelos y elaborar una metodología para predecir y vigilar dicha consolidación.
2. Diseñar e implementar un software encargado de predecir la consolidación de suelos con respecto al tiempo.
3. Analizar el comportamiento de las capacidades admisibles de carga y automatizarlas para suelos arcillosos.
4. Utilizar el lenguaje de programación Java para desarrollar una automatización del sistema de control.

## II. MARCO TEÓRICO

### 2.1. Antecedentes

En nuestro país no hay información alguna disponible, sobre la existencia de estudios y software, que permita planificar el proceso de consolidación de suelos arcillosos.

En la Universidad Nacional “Santiago Antúnez de Mayolo” se ha edificado a lo largo de varios años edificaciones en Shancayán, para varias facultades, Ciencias, Agrícola, etc. Puede apreciarse que en las mismas existen rajaduras, lo cual se debe a que el asentamiento en los suelos ha implicado un desequilibrio en la misma estructura trayendo consigo un desnivel que se manifiesta en la mencionada rajadura.

La ausencia de un estudio riguroso en la consolidación de suelos y la falta de automatización en el proceso de consolidación conforman un antecedente para el desarrollo de un sistema de control, objetivo de la presente tesis.

### 2.2. Bases Teóricas

Previamente antes de entrar a la solución del problema daremos algunos fundamentos teóricos que servirán como soporte para solucionar con éxito el problema planteado.

#### **Clasificación de Suelos**

Cuando un suelo arcilloso se mezcla con una cantidad abundante de agua, dicho suelo, puede fluir como un semilíquido, Si el suelo es secado gradualmente, se comportará como un material *plástico, semisólido o sólido* dependiendo de su contenido de agua, este, en porcentaje necesario con que

el suelo cambia de un estado líquido a un estado plástico se define como *límite líquido (LL)*, de manera similar los contenidos de agua, en porcentaje necesario con el que el suelo cambia de un estado plástico a un semisólido se define como *límite plástico(PL)* y de un semisólido a un sólido se define como el *límite de contracción(SL)*, todos estos límites reciben el nombre de *límites de Atterberg*(BRAJA, DAS. 2001). Se define como *índice de plasticidad (PI)* a la diferencia entre el límite líquido y el límite plástico de un determinado suelo:

$$PI = LL - PL$$

Los sistemas de clasificación de suelos como se menciona en (BRAJA, DAS. 2001) consideran a estos en dos grandes grupos en base a sus propiedades ingenieriles tales como su distribución granulométrica.

Estos dos sistemas principales que están actualmente en uso son:

1. El sistema AASHTO (American Association of State Highway and Transportation Officials).
2. El Unified Soil Classification System (También ASTM).

### **Sistema Aashto**

Fue propuesto originalmente por el Highway Research Board's Comité, sobre clasificación de materiales para caminos de tipo regular, este sistema se usa principalmente para clasificación de las capas de carreteras.

De acuerdo a este sistema, los suelos se clasifican según ocho grupos principales, A1-A8 en base a su distribución granulométrica. Los suelos comprendidos en A-1, A-2 y A-3 son materiales de grano grueso. Los

comprendidos en A4, A5, A6 y A-7 son de grano fino. La turba, el lodo y otros suelos altamente orgánicos están clasificados en el grupo A-8.

### **Sistema Unificado (ASTM)**

Fue propuesto originalmente por Casagrande en 1942 y después revisado y adoptado por el Bureau of Reclamation de Estados Unidos y por el cuerpo de Ingenieros. Actualmente este sistema es usado en todo trabajo de geotecnia.

El sistema unificado usa símbolos para algunas acepciones de suelos, estos son:

G: Grava      S: Arena      M: Limo      C: Arcilla

O: Limos Orgánicos    Pt: Turba y Suelos

H: Alta Plasticidad    L: Baja Plasticidad

W: Bien Graduados    P: Mal Graduados

Las características y los nombres típicos mediante divisiones principales se realizan con un tamiz número 200, estas se dan a continuación de acuerdo a (BERRY, PETER L. 1993):

**Suelos e Granos Gruesos:** debido a que el 50% o más es retenido por el tamiz número 200:

1. **Gravas:** 50% o más de la fracción gruesa se retiene en un tamiz número 4.

<b>Símbolo</b>	<b>Nombre</b>
<b>GW</b>	Gravas bien gradadas y mezclas de arena y grava con pocos finos o sin finos. GRAVAS LIMPIAS.
<b>GP</b>	Gravas y mezclas de gravas y arenas mal gradadas con pocos finos o sin finos. GRAVAS LIMPIAS.
<b>GM</b>	Gravas limosas, mezcla de gravas, arena y limo. GRAVAS CON FINOS.
<b>GC</b>	Gravas arcillosas, mezcla de grava, arena y arcilla. GRAVAS CON FINOS.

2. **Arenas:** más del 50% de la fracción gruesa pasa por el tamiz.

<b>Símbolo</b>	<b>Nombre</b>
<b>SW</b>	Arena y arenas gravosas bien gradadas con pocos finos o sin finos. ARENAS LIMPIAS.
<b>SP</b>	Arenas y arenas gravosas mal gradadas con pocos finos o sin finos. ARENAS LIMPIAS.
<b>SM</b>	Arenas limosas, mezcla de arena y limo. ARENAS CON FINOS.
<b>SC</b>	Arenas arcillosas, mezcla de arena y arcilla. ARENA CON FINOS.

**Suelos de Granos Finos:** debido a que el 50% o más pasa por el tamiz número 200:

1. **Limos y Arcillas:** con LL de 50% o inferior

<b>Símbolo</b>	<b>Nombre</b>
<b>ML</b>	Limos orgánicos, arenas muy finas, polvo de roca. Arenas finas limosas o arcillosas.
<b>CL</b>	Arcillas inorgánicas de plasticidad baja a media, arcillas gravosas, arcillas arenosas, arcillas limosas, suelos sin mucha arcilla.
<b>OL</b>	Limos orgánicos y arcillas orgánicas de baja plasticidad.

2. **Limos y Arcillas con LL superior: a 50%**

Símbolo	Nombre
MH	Limos orgánicos, arena fina, limos micáceos o de diatomeas. Limos elásticos.
CH	Arcillas inorgánicas de alta plasticidad, arcillas grasas.
OH	Arcillas orgánicas de plasticidad alta o media.

**Consolidación**

Cuando se realiza una construcción, el esfuerzo sobre una capa de arcilla se incrementa y así la presión de poro de agua se incrementará. La arcilla posee una permeabilidad hidráulica muy pequeña, esto implicará que sea necesario algún tiempo para que el exceso de presión de poro de agua se disipe y el incremento del esfuerzo se transfiera gradualmente a la estructura del suelo.

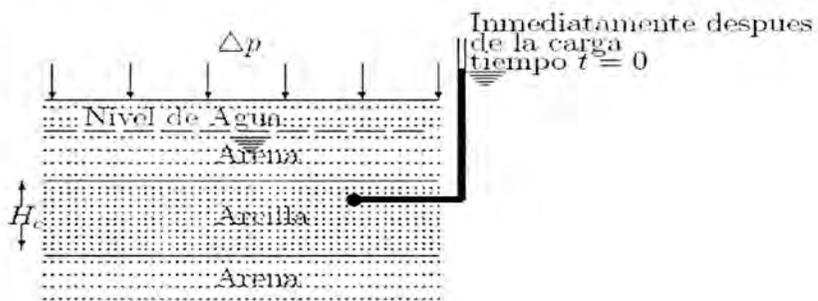


Figura 2.1: Transferencia de la Presión

De acuerdo a la figura 1, si  $\Delta p$  es una sobrecarga en la superficie del terreno sobre un área muy grande, el incremento del esfuerzo total  $\Delta\sigma$  en cualquier profundidad del estrato de arcilla será igual a  $\Delta p$ , es decir:

$$\Delta\sigma = \Delta p$$

Inmediatamente después de la aplicación del esfuerzo (en el tiempo  $t = 0$ ), el *exceso de presión de poro del agua* en cualquier profundidad  $\Delta u$  será igual a  $\Delta p$ :

$$\Delta u = \Delta p \text{ (En el tiempo } t = 0)$$

Además el *incremento de esfuerzo efectivo*  $\Delta \sigma'$  en el tiempo  $t = 0$  es:

$$\Delta \sigma' = \Delta \sigma - \Delta u = 0$$

Teóricamente, en el tiempo  $t = \infty$  todo exceso de presión de poro en el estrato de arcilla se ha disipado como una consecuencia del drenado hacia los estratos de arena.

$$\Delta u = 0 \text{ (En el tiempo } t = \infty)$$

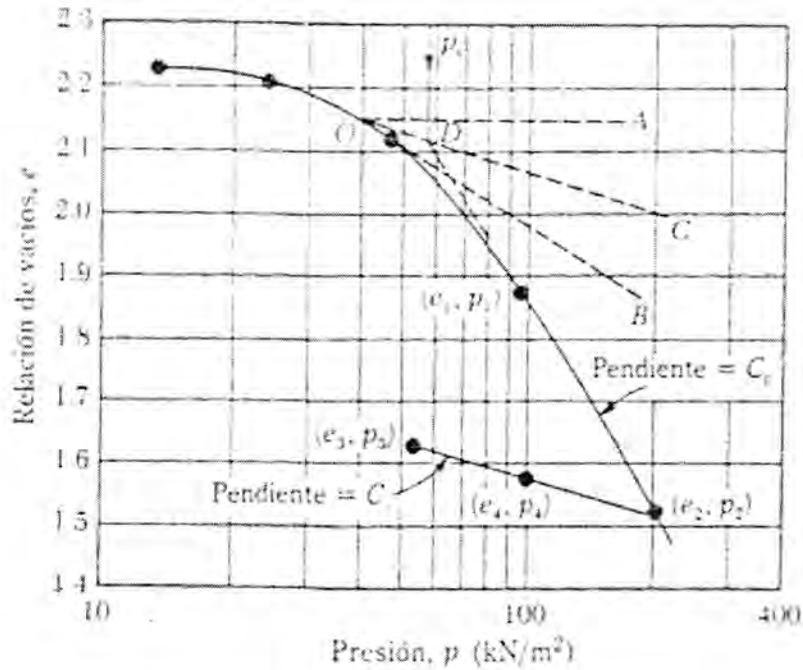
Luego el incremento de esfuerzo efectivo en la capa de arcilla es:

$$\Delta \sigma' = \Delta \sigma - \Delta u = \Delta p - 0 = \Delta p$$

Este incremento gradual se vera reflejado en asentamientos durante cierto tiempo y se conoce como *Consolidación* (BRAJA, DAS. 2001).

A fin de determinar los asentamientos por consolidación sobre un depósito de arcilla, se extrae una muestra saturada e inalterada a fin de someterla a una prueba designada con el nombre de D-2435 del ASTM en laboratorio.

En base a pruebas de laboratorio, donde se aplican cargas sobre la muestra de manera que el esfuerzo vertical total sea igual a  $p$ , se traza una gráfica que muestre la variación de la *relación de vacíos  $e$*  (*cociente del volumen de vacío entre el volumen de sólido*) al final de la consolidación contra el esfuerzo vertical correspondiente a  $p$ . La gráfica mencionada es semilogarítmica:  $e$  esta en escala aritmética y  $p$  esta en escala logarítmica.



**Figura 2.2:** Grafica de presión – relación de vacíos

Una vez que la presión de consolidación se ha logrado, la muestra se descarga gradualmente lo que resulta en expansión. De la curva  $e-\log p$  mostrada en la figura 2.2, se determinan tres parámetros necesarios para calcular el asentamiento por consolidación como se indica en (BRAJA, DAS. 2001):

1. **La carga de pre consolidación  $p_c$**  es la máxima sobrecarga efectiva a la que en el pasado estuvo sometido el suelo. Esta se calcula usando un procedimiento gráfico propuesto por Casagrande (1936), de acuerdo a la figura.

Sea  $p_0$  la presión inicial en las pruebas de laboratorio. Los depósitos naturales de suelo pueden estar *normalmente consolidados* o *sobre consolidados*.

Si  $p_0 = p_c$  el suelo está normalmente consolidado, si  $p_0 < p_c$  se considera sobre consolidado.

2. El **Índice de Compresibilidad**  $C_c$  es la pendiente de la porción recta de la curva (última parte de la curva de carga):

$$C_c = \frac{e_1 - e_2}{\log p_2 - \log p_1} = \frac{e_1 - e_2}{\log \frac{p_2}{p_1}}$$

3. El **Índice de Expansibilidad**  $C_s$  es la pendiente de la porción de descarga de la curva  $e - \log p$ , se define como:

$$C_s = \frac{e_3 - e_4}{\log p_4 - \log p_3} = \frac{e_3 - e_4}{\log \frac{p_4}{p_3}}$$

### Tasa de Consolidación

La consolidación es resultado de la disipación gradual del exceso de presión de agua en un estrato de arcilla, por tanto, se requiere conocer la rapidez de la disipación del exceso de presión de poro del agua.

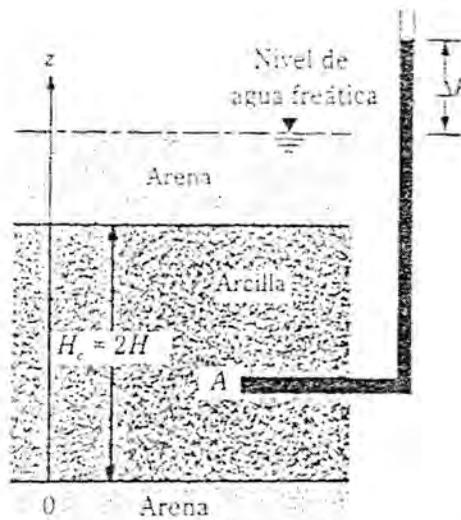


Figura 2.3: Estrato de Arcilla en Consolidación

En la figura se muestra un estrato de arcilla de espesor  $H_c$  que tiene estratos de arena altamente permeables arriba y abajo. Aquí, el exceso de presión de poro en cualquier punto  $A$  en un tiempo  $t$  una vez aplicada la carga es:

$$\Delta u = (\Delta h)\gamma_w$$

Donde:  $\gamma_w$  es el peso específico del agua.

En un drenaje vertical (sólo en la dirección de  $z$ ) para el estrato de arcilla, Terzaghi obtuvo la siguiente ecuación diferencial:

$$\frac{\partial(\Delta u)}{\partial t} = C_v \frac{\partial^2(\Delta u)}{\partial z^2}$$

Donde  $C_v$  es llamado *coeficiente de consolidación vertical* y está dado por:

$$C_v = \frac{k}{m_v \gamma_w} = \frac{k}{\frac{\Delta e}{\Delta p(1+e_{av})} \gamma_w}$$

Donde:

$k$  = permeabilidad hidráulica de la arcilla.

$\Delta e$  = cambio total de la relación de vacíos causada por un incremento  $\Delta p$  del esfuerzo.

$e_{av}$  = relación de vacíos promedio durante la consolidación.

$m_v$  = *coeficiente volumétrico de compresibilidad*

La ecuación diferencial de Terzaghi, puede resolverse a fin de obtenerse  $\Delta u$  en función del tiempo  $t$  usando las siguientes condiciones de frontera (TERZAGHI, K y PECK, R. 1955):

1. En  $z = 0$  y en  $z = H_c$  se encuentran estratos de arena los cuales son altamente permeables y de ahí que el exceso de presión de poro en esos puntos será inmediatamente disipada, con ello:

$$\Delta u = 0 \text{ En } z = 0$$

$$\Delta u = 0 \text{ En } z = H_c = 2H$$

Donde:  $H$  = longitud de la trayectoria máxima de drenaje.

2. En el momento inicial  $t = 0$ ,  $\Delta u = \Delta u_0$  = exceso inicial de presión de poro del agua después de la aplicación de la carga.

Con las condiciones anteriormente mencionadas se tiene que la solución está dada por:

$$\Delta u = \sum_{m=0}^{\infty} \left[ \frac{2(\Delta u_0)}{M} \sin\left(\frac{Mz}{H_c}\right) \right] e^{-M^2 T_v}$$

Donde:

$$M = [(2m + 1)\pi] / 2$$

$$m = \text{un entero} = 1, 2, 3, \dots$$

$$T_v = \text{factor de tiempo adimensional} = (C_v t) / H^2$$

El valor de  $C_v$  se determina por lo general en el laboratorio.

El *grado promedio de consolidación* del estrato de arcilla se define como:

$$U = \frac{S_i}{S_{\max}}$$

Donde:

$U$  = grado de consolidación promedio.

$S_t$  = asentamiento del estrato de arcilla en el tiempo  $t$  después de la aplicación de la carga.

$S_{\max}$  = asentamiento máximo por consolidación que la arcilla experimentará bajo una determinada carga.

Para una distribución de la presión de poro del agua inicial  $\Delta u_0$  constante respecto a la profundidad, el grado promedio de consolidación puede también expresarse como:

$$U = \frac{S_t}{S_{\max}} = \frac{\int_0^{2H} (\Delta u_0) dz - \int_0^{2H} (\Delta u) dz}{\int_0^{2H} (\Delta u_0) dz}$$

O

$$U = 1 - \frac{\int_0^{2H} (\Delta u) dz}{2H(\Delta u_0)}$$

Combinando la última expresión con la solución de la ecuación de Terzaghi, se obtiene:

$$U = \frac{S_t}{S_{\max}} = 1 - \sum_{m=0}^{\infty} \left( \frac{2}{M^2} \right) e^{-M^2 T_v}$$

Con esta última expresión se puede calcular la variación de  $U$  con  $T_v$ , la validez, se tiene para un estrato impermeable que se encuentra en el fondo del estrato de arcilla. Para el caso que la longitud máxima de drenaje es  $H = H_c$ , La variación de  $T_v$  con  $U$  puede ser aproximada por:

$$T_v = \frac{\pi}{4} \left( \frac{U\%}{100} \right)^2 \text{ Para } U = 0 - 60\% \text{ y}$$

$$T_v = 1.781 - 0.933(100 - U\%) \text{ Para } U > 60\%$$

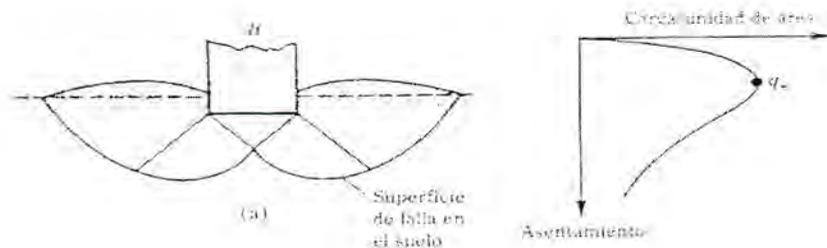
Sivaram y Swamee (1977) desarrollaron también una relación empírica entre  $T_v$  y  $U$ , la cual es válida para  $U$  variando entre 0 y 100%. Esta relación es:

$$T_v = \frac{\left( \frac{\pi}{4} \right) \left( \frac{U\%}{100} \right)^2}{\left[ 1 - \left( \frac{U\%}{100} \right)^{5.6} \right]^{0.357}}$$

De acuerdo a (BRAJA, DAS. 2001).

### Capacidad de Carga

Consideremos una cimentación con un ancho igual a  $B$ , la cual descansa sobre la superficie de suelo firme como se muestra en la figura:



**Figura 2.4:** Falla en el suelo por Capacidad de Carga

Al aplicarse gradualmente la carga a la cimentación, el asentamiento se incrementará. En la gráfica se muestra la variación de la carga por unidad de área  $q$  sobre la cimentación.

En cierto momento, cuando la carga por unidad de área sea igual a  $q_u$  ocurrirá una falla repentina en el suelo que soporta la cimentación y la zona de falla en el suelo se extenderá hasta la superficie del terreno. A esta carga por área unitaria  $q_u$  se le llama *capacidad de carga última*. Cuando este tipo de falla sucede, se denomina *falla general por corte*.

Terzaghi (1943) fue el primero en presentar una teoría completa para evaluar la capacidad de carga última en cimentaciones superficiales. En una cimentación de ancho  $B$  y longitud  $L$ , denominamos *cimentación corrida*, cuando la razón,  $B/L$  tiende a cero. Asimismo una *cimentación superficial* si la profundidad  $D_f$  de la cimentación es menor o igual que el ancho de la misma. Investigaciones posteriores afirman que cimentaciones con  $D_f$  igual a 3 o 4 veces el ancho de la cimentación pueden ser definidas como cimentaciones superficiales.

Terzaghi usando un análisis de equilibrio (TERZAGHI, K y PECK, R. 1955) expreso la capacidad de carga última mediante:

$$q_u = cN_c + qN_q + \frac{1}{2}\gamma BN_\gamma \text{ (Cimentación corrida)}$$

Donde:

$c$  = cohesión del suelo.

$\gamma$  = peso específico del suelo.

$$q = \gamma D_f$$

$N_c, N_q, N_\gamma$  = factores de capacidad de carga adimensionales que están en función del ángulo  $\phi$  de fricción del suelo.

Los factores de capacidad de carga  $N_c$  y  $N_q$  se definen mediante:

$$N_c = (N_q - 1) \cot \phi \quad \text{Y} \quad N_q = \frac{e^{2\left(\frac{3\pi}{4} - \frac{\phi}{2}\right) \tan \phi}}{2 \cos^2\left(\frac{\pi}{4} + \frac{\phi}{2}\right)}$$

Para los valores de  $N_\gamma$ , se tiene la siguiente tabla (KUMBHOJKAR, A. 1993):

$\phi$	$N_\gamma$								
0	0,00	10	0,56	20	3,64	30	19,13	40	115,31
1	0,01	11	0,69	21	4,31	31	22,65	41	140,51
2	0,04	12	0,85	22	5,09	32	26,87	42	171,99
3	0,06	13	1,04	23	6,00	33	31,94	43	211,56
4	0,10	14	1,26	24	7,08	34	38,04	44	261,60
5	0,14	15	1,52	25	8,34	35	45,41	45	325,34
6	0,20	16	1,82	26	9,84	36	54,36	46	407,11
7	0,27	17	2,18	27	11,60	37	65,27	47	512,84
8	0,35	18	2,59	28	13,70	38	78,61	48	650,67
9	0,44	19	3,07	29	16,18	39	95,03	49	831,99

Para calcular la capacidad de carga última de *cimentaciones cuadradas o circulares* la ecuación de capacidad de carga última de Terzaghi se convierte en:

$$q_u = 1.3cN_c + qN_q + 0.4\gamma BN_\gamma \quad (\text{Cimentación cuadrada})$$

$$q_u = 1.3cN_c + qN_q + 0.3\gamma BN_\gamma \quad (\text{Cimentación circular})$$

En la ecuación penúltima  $B$  es igual a la dimensión de cada lado de la cimentación y en la última ecuación  $B$  es igual al diámetro de la cimentación.

Cuando existe una falla local por corte en suelos, Terzaghi modificó las ecuaciones anteriores se transforman en:

$$q_u = \frac{2}{3}cN'_c + qN'_q + \frac{1}{2}\gamma BN'_\gamma \quad (\text{Cimentación corrida})$$

$$q_u = 0.867cN'_c + qN'_q + 0.4\gamma BN'_\gamma \quad (\text{Cimentación cuadrada})$$

$$q_u = 0.867cN'_c + qN'_q + 0.3\gamma BN'_\gamma \quad (\text{Cimentación circular})$$

A  $N'_c, N'_q, N'_\gamma$  se les llama *factores de capacidad de carga modificada*, los cuales se calculan usando las ecuaciones para  $N_c, N_q$  y la tabla para  $N_\gamma$  reemplazando  $\phi$  por  $\phi' = \arctan\left(\frac{2}{3} \tan \phi\right)$ .

Se llama *capacidad de carga admisible* de cimentaciones superficiales a la razón dada por:

$$q_a = \frac{q_u}{FS} = \frac{\text{capacidad carga ultima}}{\text{factor de seguridad}}$$

Por lo general los ingenieros adoptan un factor de seguridad igual a 3.

Las ecuaciones de capacidad de carga última son únicamente para cimentaciones continuas, cuadradas y circulares, estas no se aplican al caso de cimentaciones rectangulares ( $0 < \frac{B}{L} < 1$ ). Además, las ecuaciones no toman en cuenta factores de forma y profundidad, los cuales son datos empíricos basados en datos experimentales. A fin de tomar en consideración

estos factores Meyerhof (1963) planteo la *Ecuación General de Capacidad de Carga*:

$$q_u = cN_c F_{cs} F_{cd} + qN_q N_{qs} F_{qd} + \frac{1}{2} \gamma B N_\gamma F_{\gamma s} F_{\gamma d}$$

Donde:

$c$  = cohesión.

$q$  = esfuerzo efectivo al nivel del fondo de la cimentación.

$\gamma$  = peso específico del suelo.

$B$  = ancho de la cimentación (= diámetro para c. circulares)

$F_{cs}, F_{qs}, F_{\gamma s}$  = factores de forma.

$F_{cd}, F_{qd}, F_{\gamma d}$  = factores de profundidad.

$N_c, N_q, N_\gamma$  = factores de capacidad de carga.

Para los factores de capacidad de carga, las relaciones para  $N_c$  y  $N_q$  pueden obtenerse con:

$$N_q = \tan^2 \left( 45 + \frac{\phi}{2} \right) e^{\pi \tan \phi}$$

Y

$$N_c = (N_q - 1) \cot \phi$$

De las ecuaciones anteriores:  $N_c$  fue obtenida por Prandtl (1921) y  $N_q$  fue presentada por Reissner (1924). Caquot y Kerisel (1953) y Vesic (1973) dieron la ecuación para  $N_\gamma$  (BRAJA, DAS. 2001) como:

$$N_\gamma = 2(N_q + 1) \tan \phi$$

En cuanto a los valores de  $N_\gamma$  existe controversia entre los distintos autores de texto y libros, esto es debido a que existen otras relaciones para  $N_\gamma$  estas son:

Meyerhof en 1963:

$$N_\gamma = (N_q - 1) \tan(1.4\phi)$$

Hansen en 1970:

$$N_\gamma = 1.5(N_q - 1) \tan \phi$$

Y Lungren y Mortesen en 1953 (BRAJA, DAS. 2001):

$\phi(\text{grados})$	$N_\gamma$
0	0
5	0,17
10	0,46
15	1,4
25	6,92
30	15,32
35	35,19
40	86,46
45	215,0

Las relaciones para los factores de forma y profundidad recomendados son:

**Forma:**

$$F_{cs} = 1 + \frac{BN_q}{LN_c}, F_{qs} = 1 + \frac{B}{L \tan \phi}, F_{\gamma s} = 1 - 0.4 \frac{B}{L}$$

**Profundidad:**

1. Para  $D_f / B \leq 1$ :

$$F_{cd} = 1 + 0.4 \frac{D_f}{B}, F_{qd} = 1 + 2 \tan \phi (1 - \sin \phi)^2 \frac{D_f}{B}, F_{\gamma d} = 1$$

2. Para  $D_f / B > 1$ :

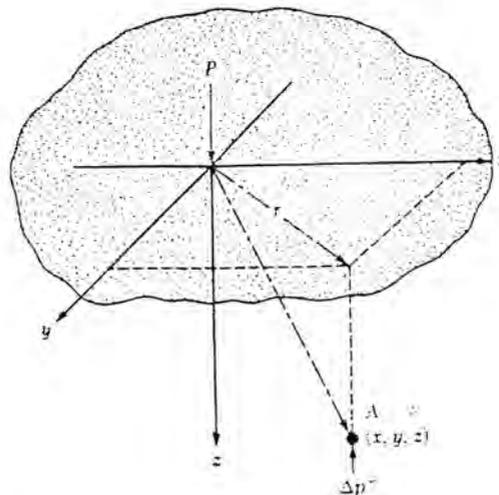
$$F_{cd} = 1 + 0.4 \tan^{-1} \left( \frac{D_f}{B} \right),$$

$$F_{qd} = 1 + 2 \tan \phi (1 - \sin \phi)^2 \tan^{-1} \left( \frac{D_f}{B} \right),$$

$$F_{\gamma d} = 1$$

### Esfuerzo Vertical

En 1885, Boussinesq desarrolló relaciones matemáticas para la determinación de los esfuerzos normal y de corte en un punto cualquiera dentro de medios homogéneos, elásticos e isotrópicos debido a una carga puntual concentrada y localizada en la superficie como se muestra en la siguiente figura:



**Figura 2.5:** Esfuerzo vertical en un punto A

De acuerdo al análisis de Boussinesq (BRAJA, DAS. 2001) el *incremento de esfuerzo vertical*  $\Delta p$  en el punto A causado por una carga puntual  $P$  es:

$$\Delta p = \frac{3p}{2\pi z^2 \left[ 1 + \left( \frac{r}{z} \right)^2 \right]^{5/2}}$$

Donde:

$$r = \sqrt{x^2 + y^2}$$

$x, y, z$  = coordenadas del punto  $A$ .

El procedimiento de integración de la ecuación de Boussinesq permite evaluar el esfuerzo vertical en cualquier punto  $A$  debajo de una esquina de una superficie flexible rectangular cargada como se muestra en la figura 2.6.

Para ello consideramos un elemento de área  $dA = dxdy$  sobre la superficie cargada, con una carga por unidad de área  $q_0$  la carga total de dicho elemento es:

$$dP = q_0 dxdy$$

Esta carga  $dP$  es una carga puntual.

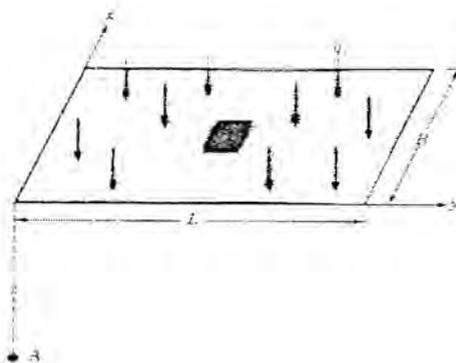


Figura 2.6: Esfuerzo debajo de una esquina de rectángulo

El incremento del esfuerzo vertical en A causado por  $dP$  se calcula usando la fórmula del incremento del esfuerzo vertical, sustituyendo  $dP = q_0 dx dy$  por  $P$  y  $x^2 + y^2$  por  $r^2$  para tener:

$$\text{Inc del esfuerzo en A es causado por } dP = \frac{3q_0(dx dy)}{2\pi(x^2 + y^2 + z^2)^{5/2}}$$

Se integra la ecuación anterior para obtener el incremento total del esfuerzo causado por el área total cargada en el punto A:

$$\Delta p = \int_{y=0}^L \int_{x=0}^B \frac{3q_0(dx dy)z^3}{2\pi(x^2 + y^2 + z^2)^{5/2}} = q_0 I$$

Donde  $\Delta p$  = incremento del esfuerzo en A y la cantidad I es llamada *factor de influencia*.

En 1935 Newmark elabora cuadros con valores experimentales sobre el factor de influencia en base a ensayos realizados, en la actualidad con estos valores se construyen un conjunto de curvas en un diagrama, denominado ábaco (BERRY, PETER L. 1993). En la presente tesis se automatiza todo este procedimiento, mediante un tratamiento de construcción de expresiones para la evaluación del factor de influencia.

Esto resulta ventajoso por que permite evaluar sistemáticamente el proceso de consolidación, debido a que con el factor de influencia se puede calcular el incremento del esfuerzo vertical y con ello el asentamiento por consolidación.

Otra ventaja es que la lectura del ábaco puede resultar en algunos casos imprecisa debido a un manejo manual de curvas que se encuentran muy

cercanas unas de otras, lo cual hace en algunas ocasiones tediosa la determinación del factor de influencia.

En 1942 Newmark propuso un procedimiento gráfico para poder determinar el incremento del esfuerzo vertical, este procedimiento consistía en construir un círculo, en el cual se consideran escalas métricas y circunferencias concéntricas.

El procedimiento gráfico propuesto por Newmark, si bien es útil, no llegó a automatizarse en dicha época, por razones que la tecnología no lo permitía.

En el siguiente cuadro se muestra, valores calculados del esfuerzo vertical debajo de una esquina de un área rectangular flexible cargada, dados por Newmark:

$\frac{r}{L}$	$\frac{L}{r}$									
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
0.1	0.004	0.009	0.013	0.016	0.019	0.022	0.024	0.025	0.026	0.027
0.2	0.009	0.017	0.025	0.032	0.038	0.043	0.047	0.050	0.052	0.054
0.3	0.013	0.025	0.037	0.047	0.055	0.062	0.068	0.073	0.076	0.079
0.4	0.016	0.032	0.047	0.060	0.071	0.080	0.087	0.093	0.097	0.101
0.5	0.019	0.038	0.055	0.071	0.084	0.094	0.103	0.110	0.115	0.120
0.6	0.022	0.043	0.062	0.080	0.094	0.106	0.116	0.124	0.131	0.136
0.7	0.024	0.047	0.068	0.087	0.103	0.116	0.127	0.136	0.143	0.149
0.8	0.025	0.050	0.073	0.093	0.110	0.124	0.136	0.146	0.153	0.159
0.9	0.026	0.052	0.076	0.097	0.115	0.131	0.143	0.153	0.161	0.168
1.0	0.027	0.054	0.079	0.101	0.120	0.136	0.149	0.159	0.168	0.175
1.2	0.029	0.057	0.083	0.106	0.126	0.143	0.157	0.168	0.177	0.185
1.4	0.030	0.058	0.085	0.109	0.130	0.147	0.161	0.173	0.183	0.191
1.6	0.030	0.059	0.087	0.111	0.132	0.150	0.165	0.177	0.187	0.195
1.8	0.030	0.060	0.088	0.112	0.133	0.152	0.167	0.179	0.189	0.198
2.0	0.031	0.061	0.088	0.113	0.134	0.153	0.168	0.181	0.181	0.191
2.5	0.031	0.062	0.089	0.114	0.136	0.154	0.170	0.183	0.193	0.202
3.0	0.032	0.063	0.089	0.114	0.136	0.155	0.171	0.184	0.194	0.203
4.0	0.032	0.063	0.090	0.115	0.137	0.155	0.171	0.184	0.195	0.204
5.0	0.032	0.063	0.090	0.115	0.137	0.156	0.171	0.184	0.195	0.204
10.0	0.036	0.063	0.090	0.115	0.137	0.156	0.171	0.185	0.195	0.205

## Asentamientos

El asentamiento de una cimentación se divide en dos categorías principales (BRAJA, DAS. 2001):

**Asentamiento Elástico:** o inmediato, el cual tiene lugar durante o inmediatamente después de la construcción de la estructura.

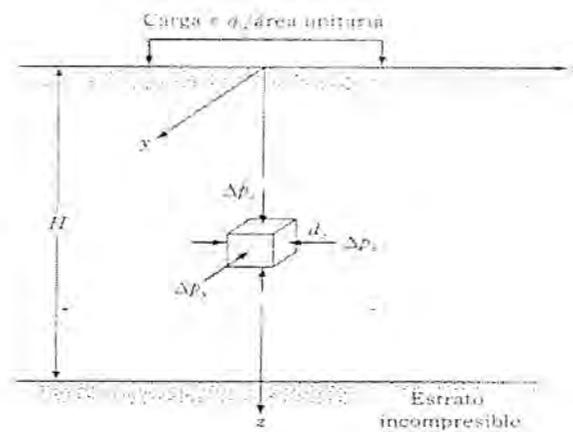


Figura 2.7: Asent. Elástico de una Cimentación Superficial.

**Asentamiento por Consolidación:** ocurre a lo largo del tiempo. El agua de los poros es expulsada de los espacios vacíos de los suelos arcillosos saturados sumergidos en agua.

El *asentamiento total* de una cimentación es la suma de los asentamientos elásticos y por consolidación:

$$S_T = S_e + S_c$$

El asentamiento elástico de una cimentación superficial se calcula usando la teoría de elasticidad. De acuerdo a la figura, aplicando la ley de Hooke:

$$S_e = \int_0^H \varepsilon_z dz = \frac{1}{E_s} \int_0^H (\Delta p_z - \mu_s \Delta p_x - \mu_s \Delta p_y) dz$$

Donde:

$S_e$  = asentamiento elástico.

$E_s$  = módulo de elasticidad del suelo.

$H$  = espesor del estrato de suelo.

$\mu_s$  = relación de Poisson del suelo.

$\Delta p_x, \Delta p_y, \Delta p_z$  = incremento del esfuerzo debido a la carga neta aplicada a la cimentación, en las direcciones  $x, y$  y  $z$  respectivamente.

Teóricamente, si la profundidad de la cimentación  $D_f = 0, H \rightarrow \infty$ , por Harr (1966), el asentamiento se expresa mediante:

$$S_e = \frac{Bq_0}{E_s} (1 - \mu_s^2) \frac{\alpha}{2} \quad (\text{Esquina de la cimentación})$$

$$S_e = \frac{Bq_0}{E_s} (1 - \mu_s^2) \alpha \quad (\text{Centro de la cimentación})$$

Donde:

$$\alpha = \frac{1}{\pi} \left[ \ln \left( \frac{\sqrt{1+m_1^2} + m_1}{\sqrt{1+m_1^2} - m_1} \right) + m \ln \left( \frac{\sqrt{1+m_1^2} + 1}{\sqrt{1+m_1^2} - 1} \right) \right]$$

$$m_1 = \frac{L}{B}$$

$B$  = ancho de la cimentación.

$L$  = longitud de la cimentación

El asentamiento por consolidación se da a lo largo del tiempo y sucede en suelos arcillosos saturados cuando son sometidos a una carga creciente causada por la construcción de una cimentación.

El asentamiento unidimensional por consolidación de una capa de arcilla con espesor  $H_c$  puede calcularse como:

$$S_c = \frac{\Delta e}{1 + e_0} H_c$$

Donde:

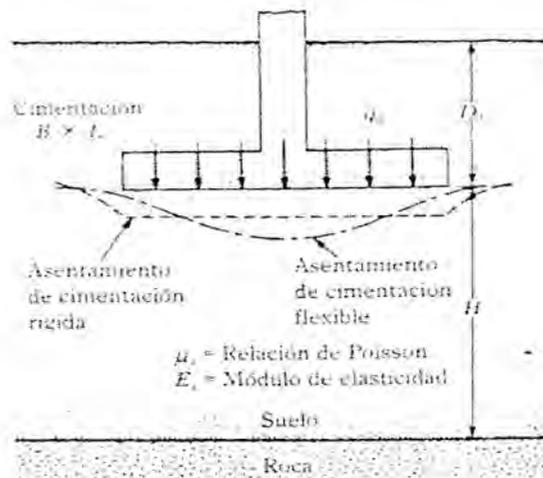
$S_c$  = asentamiento por consolidación.

$\Delta e$  = cambio total de la relación de vacíos causada por la aplicación de una carga adicional.

$e_0$  = relación de vacíos de la arcilla antes de la aplicación de la carga.

Obsérvese que:

$$\frac{\Delta e}{1 + e_0} = \varepsilon_v = \text{deformación unitaria vertical.}$$



**Figura 2.8:** Asentamiento en una cimentación flexible y rígida

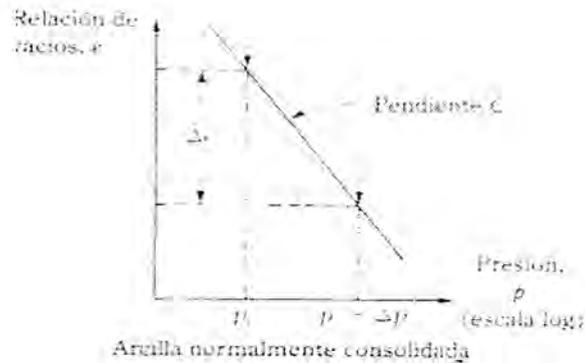
Con respecto a la arcilla pueden presentarse dos casos:

**Arcilla Normalmente Consolidada:** en ese caso la curva  $e-\log p$  tendrá la forma mostrada en la figura 2.9.

Si  $p_0$  = presión de sobrecarga efectiva promedio inicial sobre el estrato de arcilla y  $\Delta p$  = incremento promedio de presión sobre el estrato de arcilla, causado por la carga agregada, entonces  $C_c$  cumple con:

$$C_c = \frac{\Delta e}{\log p_0 + \Delta p - \log p_0}$$

Con lo cual tenemos:



**Figura 2.9:** Arcilla Normalmente Consolidada

$$\Delta e = C_c \log \frac{p_0 + \Delta p}{p_0}$$

Con lo cual tenemos que:

$$S_c = \frac{C_c H_e}{1 + e_0} \log \frac{p_0 + \Delta p}{p_0}$$

**Arcilla Sobre consolidada:** en este caso la curva de campo  $e-\log p$  será como en la figura 2.10.

El valor de  $\Delta p$  puede implicar dos casos:

**Primera:**  $p_0 + \Delta p < p_c$ :

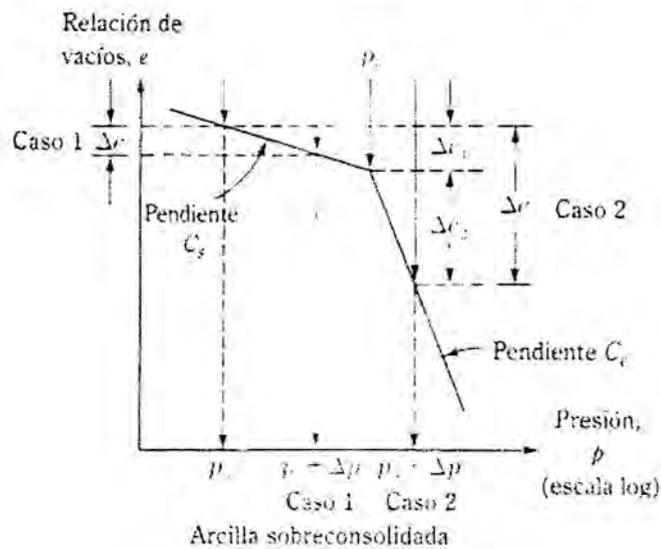
$$\Delta e = C_s \log \frac{p_0 + \Delta p}{p_0}$$

Con lo cual el asentamiento por consolidación es:

$$S_c = \frac{C_s H_c}{1 + e_0} \log \frac{p_0 + \Delta p}{p_0}$$

**Segunda:**  $p_0 < p_c < p_0 + \Delta p$

$$\Delta e = \Delta e_1 + \Delta e_2 = C_s \log \frac{p_c}{p_0} + C_c \log \frac{p_0 + \Delta p}{p_0}$$



**Figura 2.10:** Arcilla Sobre Consolidada

Con lo cual el asentamiento por consolidación es:

$$S_c = \frac{C_s H_c}{1 + e_0} \log \frac{p_c}{p_0} + \frac{C_c H_c}{1 + e_0} \log \frac{p_0 + \Delta p}{p_c}$$

Podemos observar que:

$\Delta e$  = cambio de la relación de vacíos

$$= f(p_0, p_c, \Delta p)$$

Resumiendo, con lo expuesto tenemos:

**Arcillas Normalmente Consolidadas:**

$$S_c = \frac{C_c H_c}{1 + e_0} \log \frac{p_0 + \Delta p}{p_0}$$

**Arcillas PreConsolidadas:**  $p_0 + \Delta p_{prom} < p_c$ :

$$S_c = \frac{C_s H_c}{1 + e_0} \log \frac{p_0 + \Delta p_{prom}}{p_0}$$

**Arcillas Preconsolidadas:**  $p_0 < p_c < p_0 + \Delta p_{prom}$ :

$$S_c = \frac{C_s H_c}{1 + e_0} \log \frac{p_c}{p_0} + \frac{C_c H_c}{1 + e_0} \log \frac{p_0 + \Delta p_{prom}}{p_c}$$

Donde:

$p_0$  = presión efectiva promedio sobre el estrato de arcilla antes de la construcción de la cimentación.

$\Delta p_{prom}$  = incremento promedio de la presión sobre el estrato de arcilla por la construcción de la cimentación.

$p_c$  = presión de pre consolidación.

$e_0$  = relación de vacíos inicial del estrato de arcilla.

$C_c$  = índice de compresión.

$C_s$  = índice de expansibilidad

$H_c$  = espesor de la capa de arcilla.

Obsérvese que el incremento de la presión  $\Delta p$ , sobre el estrato de arcilla no es constante con la profundidad. El valor de  $\Delta p$  decrecerá con el incremento de la profundidad medida desde el fondo de la cimentación. No obstante, el incremento promedio  $\Delta p_{prom}$  puede aproximarse mediante:

$$\Delta p_{prom} = \frac{1}{6}(\Delta p_t + 4\Delta p_m + \Delta p_b)$$

Donde  $\Delta p_t$ ,  $\Delta p_m$  y  $\Delta p_b$  son los incrementos de presión *arriba*, en *medio* y en el *fondo* del estrato de arcilla ocasionados por la construcción de la cimentación.

En el presente trabajo se desarrollará un sistema de control para arcillas normalmente consolidadas.

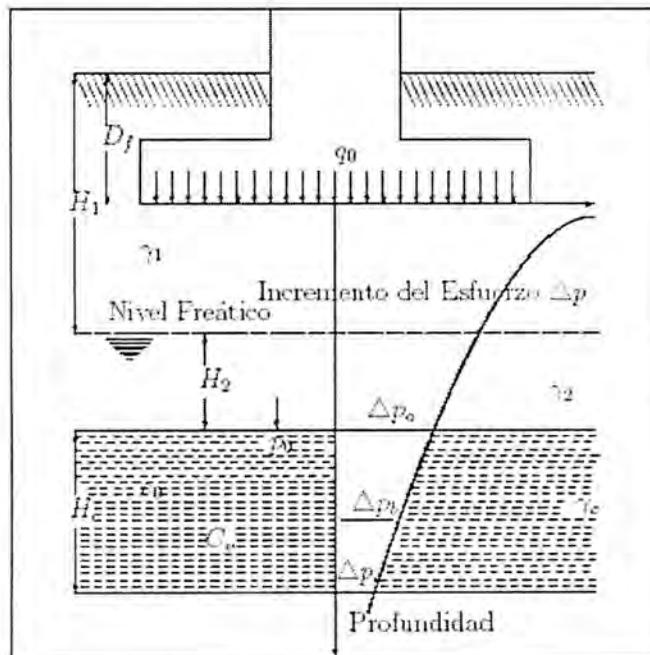


Figura 2.11: El Asentamiento por Consolidación

### III. MATERIALES Y METODOS

#### 3.1. Tipo y Diseño de la Investigación

La presente tesis es un trabajo de investigación Teórico – Aplicativa.

#### 3.2. Instrumentos de la Recolección de la Información

- El modelo matemático que describe el fenómeno de la consolidación denominado *Ecuación de diferencial de Terzaghi*.
- El Lenguaje de programación *Java*.
- El Sistema Operativo Windows.
- Un Scanner HP de alta definición
- Una Laptop HP-Pavilion ZV-5000
- CDs y memorias USB.
- El Entorno de Manipulación Grafica GIMP
- Memorias de Almacenamiento SD.
- Una impresora láser HP.
- Un Editor de Texto Notepad.

#### 3.3. Plan de Procesamiento

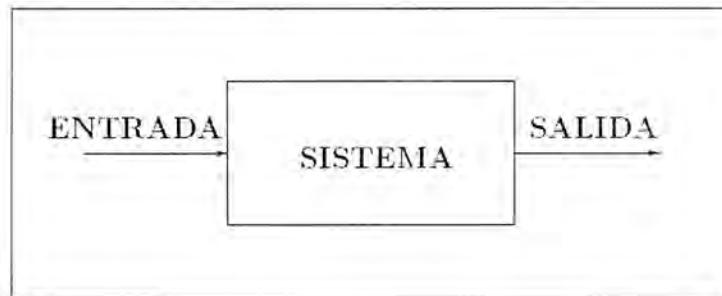
##### Sistema de Control

El desarrollo de un sistema atraviesa por distintas etapas, una de las cuales es el análisis de los objetivos y las tareas para los objetivos, dichas tareas se reparten en subtareas menores, cada una de las cuales puede ser resuelta mediante un módulo, de esta forma el objetivo principal es alcanzada por tareas que se encuentran en diversos módulos. A fin de que esto se lleve a

cabo de forma adecuada se planifica el desarrollo de los distintos módulos, a esta planificación, se denomina *Diseño Modular*.

Cada uno de los módulos que forman parte de un sistema deben interactuar entre si a fin de poder establecer una comunicación entre ellos, para recibir y emitir resultados.

En un sistema de control, como en todo sistema, existe un flujo de datos de entrada hacia el sistema, asimismo, debe existir un flujo de datos de salida, hacia el exterior del sistema.



**Figura 3.1:** Entrada y Salida en un Sistema

En el interior del sistema existen diversos módulos entre los cuales uno de ellos toma especial importancia, dicho módulo se denomina núcleo del sistema, este está encargado de dirigir la realización de las actividades de todo el sistema.

A fin de ello recibe los datos de entrada y realiza la división, de las tareas a ejecutarse y las encarga a cada uno de los módulos que componen el sistema, para que estos al finalizar retornen al núcleo los resultados esperados por este y luego dichos resultados se exterioricen mediante el flujo de datos de salida.

Lo fundamental en un sistema de control, es vigilar un determinado proceso que se lleva a cabo y otra característica de importancia es que tiene la capacidad de predecir con un grado aceptable de certeza, el comportamiento del proceso a lo largo del tiempo.

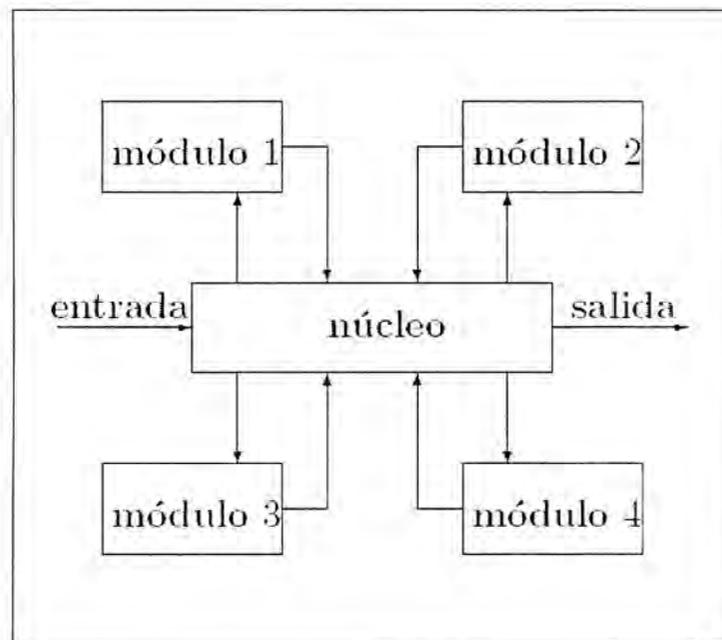


Figura 3.2: El Núcleo gobierna el sistema.

El sistema de control que se desarrolla en el presente trabajo, está dirigido al proceso del asentamiento de suelos arcillosos y es sobre este proceso que se orienta a vigilar y predecir su comportamiento.

Como se ha mencionado, el proceso de asentamiento de suelos, tiene dos etapas: el asentamiento elástico y el asentamiento por consolidación, ambos son de interés al sistema de control que desarrolla el presente trabajo.

El asentamiento elástico, se da inmediatamente de construida la edificación, su tiempo de duración es relativamente corto, aquí se tienen en cuenta los cálculos relacionados con la estructura del suelo, además de su capacidad

admisible y para ello existen tres autores que han desarrollado una teoría que es utilizada por ingenieros, dichos autores son Terzaghi, Vesic y Meyerhof.

El asentamiento por consolidación es un proceso que se da a lo largo del tiempo, pudiendo tardar días, incluso meses, aquí se adopta la teoría de Terzaghi sobre la consolidación unidimensional, la cual por lo general se adopta en las edificaciones y las fórmulas Siravam y Swamee (1977) sobre la relación entre el porcentaje de consolidación y el factor de tiempo vertical (BRAJA, DAS. 2001).

Uno o más módulos pueden destinarse a los procedimientos y calculos para el asentamiento elástico y el asentamiento por consolidación, pero además es necesario en el sistema de control la existencia de módulos que se encarguen del flujo de datos hacia el exterior y hacia el interior del sistema, asimismo módulos que puedan interactuar con el usuario que utiliza el sistema de control.

Al analizar el asentamiento en un determinado lugar, frecuentemente, la estructura granulometrica del suelo, así como sus propiedades del mismo suelo de una zona, difieran con otra zona del mismo lugar, en este caso, el asentamiento será diferente para uno y otro, teniéndose que trabajar por separado las zonas.

El sistema de control, por lo tanto, deberá ser capaz de realizar el estudio para cualquier número de zonas que puedan existir en un lugar, extrayendo

de cada zona datos relacionados con la estructura de suelo y propiedades de la misma a fin de que su resultado sea lo más confiable posible.

## IV. RESULTADOS

### Algoritmo

Un algoritmo, es un conjunto finito de pasos para realizar un determinado procedimiento. El sistema de control del presente trabajo realiza actividades orientadas al asentamiento de suelos, para lo cual, se establece un orden de las tareas, que el núcleo del sistema ha de ejecutar. Estas tareas conforman el conjunto de pasos que se traducen en el siguiente algoritmo:

1. **Especificación del Proyecto:** El Nombre del Proyecto, el nombre del Autor y el número de Zonas. Una determinada área geográfica, puede ser estudiada, en su consolidación, dividiendo dicha área en un número determinado de zonas, cada una de las cuales posee propiedades suficientemente homogéneas, con esto, se realiza un estudio más detallado, para ello, el sistema de control que desarrolla la presente tesis, esta capacitado para realizar un estudio minucioso de un área dividiéndolo en  $n$  zonas, este entero  $n > 0$  es capturado por un módulo, teniéndolo este como una propiedad, al igual que el nombre del proyecto y el nombre del autor. A fin de llevar esto a cabo, el sistema de control, crea un objeto denominado **NewProject** y se activa un método llamado **BuildEntries**, el cual crea la interfaz para interactuar con el usuario a través de tres paneles: **Ingreso de Datos, Salida de Datos y Ayuda.**
2. **Ingreso de Datos:** Recopilación de los datos de cada una de las zonas que forman parte del mismo lugar. Una vez especificado el número de zonas, el sistema de control, genera tablas que consideran todos los datos que

intervienen en el proceso de consolidación de suelos arcillosos para cada zona de estudio, en ello se considera, los parámetros de resistencia del suelo, las propiedades elásticas, la estructura del suelo en la cual intervienen las densidades de cada capa, las dimensiones de las zapatas y los datos de laboratorio para cada una de las zonas, todos estos datos ingresados son almacenados por el sistema de control y almacenados en memoria a través de arrays multidimensionales. Este paso tiene en consideración un módulo que puede ser activado de dos formas: Con argumentos y sin argumentos. El módulo **panelIN**, al ser activado sin argumentos, genera una entrada de datos desde cero, pero puede darse el caso que sean invocados con argumentos, lo cual implica que se carga los datos desde un archivo, situación que ocurre cuando el flujo de entrada tiene como vía de acceso la apertura de un proyecto existente.

3. **Clasificación:** Organización de los datos para planificar la distribución de las tareas. Una vez que los datos han sido ingresados por parte del usuario, se ejecuta un procedimiento de clasificación de datos por zona, de esta forma quedan distribuidas las tareas que se ejecutarán, cada zona se convierte en un ente diferente e independiente de las demás zonas, con ello cada zona está preparada para ser el argumento de un constructor de una clase que se encargará de incorporar los datos como variables miembro de una instancia que representa una zona. Generada la interfaz de usuario, y luego de ingresar los datos, el sistema, crea un archivo que tendrá una participación temporal en el presente proyecto, la generación del

mencionado archivo se da de una forma sistemática a través del flujo de datos, hacia el mismo, para poseer los datos que se encuentran en las tablas.

4. **Distribución:** El núcleo envía datos a los distintos módulos para que estos resuelvan una determinada tarea y retornen resultados. Una vez realizada, la clasificación, el sistema de control, considerando n zonas de estudio, realiza la creación de n objetos, cada uno de los cuales tiene por finalidad, el cálculo del proceso de consolidación, teniendo para ello los datos dados en las tablas. La alternativa, para el ingreso de datos aquí proviene de un archivo con extensión **jpg**, el cual tiene datos clasificados de acuerdo a los criterios de tablas: Parámetros de Resistencia, Propiedades elásticas, Estructura del Suelo, Dimensiones de Zapatas y Datos de Laboratorio. El Archivo mencionado tiene un formato ASCII, y puede ser editado con cualquier procesador de texto, de esta forma, es posible crear manualmente un archivo de proyecto para el sistema de control y cargarlo con el mismo.
  
5. **Ejecución:** Un módulo recibe datos relacionados con la estructura del suelo de una determinada zona y realiza los cálculos, teniendo en cuenta dos tipos de asentamientos: elástico y por consolidación. Dicho módulo retorna un conjunto de datos que el módulo que lo invocó debe ser capaz de interpretar. Una vez realizada la clasificación y la creación de n objetos cada uno de los cuales representa una zona determinada, el sistema de control pasa cada uno de los objetos por un proceso denominado **NewPanel**, el cual está encargado de generar un panel conteniendo todos los resultados de una zona como resultado de la ejecución de los cálculos del proceso de consolidación, incluido el gráfico de dicho proceso. Todos estos resultados, incluidos la

estructura para la generación de la grafica se ponen en memoria en forma de arrays multidimensionales de números reales. Un módulo denominado **jwork**, es el encargado de recibir datos de cada una de las zonas internamente y luego realizar los procedimientos, teniendo como secuencia, el cálculo de las capacidades últimas de carga, las capacidades admisibles, los asentamientos elásticos y finalmente los asentamientos por consolidación.

6. **Recopilación de Resultados:** Se juntan los resultados para que el sistema de control los muestre al usuario. En este punto, se generan objetos de paneles que representen a las zonas y se realiza una ubicación de los resultados en cada uno de los paneles, este proceso implica una generacion de tablas mostrando los resultados de los asentamientos elásticos y por consolidación además, mostrando porcentajes y sus equivalentes en consolidación, cabe mencionar que se construye la gráfica del proceso y se muestra dicho panel.
7. **Salida de Datos:** El sistema de control realiza un flujo de datos hacia el exterior. Como todo sistema de control, debe existir un flujo de datos al exterior, con ese fin, el sistema de control genera un archivo de extensión *doc* en el cual escribe el reporte del análisis del proceso de asentamiento elástico y por consolidación de cada una de las zonas, donde también incluye todas las características que resultaron de los cálculos.

Este archivo de extensión *doc* puede ser visualizado con el procesador de documentos *Microsoft Word* o si se desea con el *OpenOffice – Word Procesor*.

Para realizar este paso, se activa un módulo llamado **WriteReport**, el cual está encargado de recolectar los resultados de cada uno de los paneles, de esta forma todos estos resultados se derivan a través de un flujo de datos hacia el exterior, hacia el archivo anteriormente mencionado.

### **Diseño Software**

En la actualidad, una gran cantidad de actividades que involucran procesos relacionados con las ciencias e ingeniería se controlan mediante la implementación de un software relacionado con una actividad específica.

En las últimas décadas se ha observado un incremento exponencial en el desarrollo de la computación, mediante el avance de la tecnología del hardware, asimismo acompañado del desarrollo del software, esto implicó, la automatización de ciertos procesos que se realizan en la actividad diaria, para la automatización se pone de relieve el desarrollo de un programa de computadora que controle los parámetros de la mencionada automatización.

El diseño software, considerado, como el desarrollo de una aplicación para computadora, destinado a resolver un problema específico. Para dicho diseño, existen ciertas cuestiones que se deben tener en consideración, entre las mas relevantes tenemos:

**Plataforma Destino:** es el sistema operativo, donde el software, tendrá lugar a ejecutarse, por lo general, dichas plataformas son MS-Windows, aunque en la actualidad, se observa un incremento considerable, para la plataforma destino Linux (SÁNCHEZ, SEBASTIÁN Y GARCÍA, ÓSCAR. 2005).

**Lenguaje de Programación:** es la comunicación entre los procesos que se dan en el software y su ejecución. En la actualidad existen una gran cantidad de lenguajes de programación, entre aquellos que gozan de gran reputación podemos citar:

**C:** Considerado un lenguaje potente y flexible para llevar a cabo el desarrollo de aplicaciones, desarrollado por Denis Ritchie y Ken Thompson para la creación de aplicaciones capaces de controlar una gran cantidad de procesos.

**Java:** Creado en los laboratorios de *Sun-Microsystem*, es un lenguaje orientado al desarrollo de aplicaciones para los distintos sistemas operativos, Java originalmente fue creado para correr aplicaciones que puedan ejecutarse en Internet, debido al auge del Internet, el lenguaje alcanzó una considerable fama, en la actualidad, cada 6 meses, la empresa *Sun-Microsystems*, actualiza el lenguaje, brindándole una mayor capacidad y flexibilidad (SANCHO. ADELA Y BOBADILLA, JESUS. 2003).

**C#:** Creado por la empresa MS-Microsoft, para el desarrollo de aplicaciones destinadas a la ejecución en red, su nacimiento viene después de C y Java poseendo aspectos similares a estos.

**Entorno de Ejecución:** denominado también ambiente de ejecución, es un conjunto de herramientas que se proveen para que una aplicación pueda acceder a ellos en tiempo de ejecución. El entorno de ejecución va fuertemente ligado al lenguaje de programación, debido a que para cada lenguaje existe un determinado entorno de ejecución. Entre los principales entornos podemos mencionar:

**Nativo:** Al código resultante, al usar un determinado lenguaje se denomina nativo, si dicho código genera una aplicación capaz de interactuar directamente

con el sistema operativo en el cual se ejecuta, brindando una rapidez considerable a la aplicación. En la actualidad, es muy poco frecuente, el desarrollo de aplicaciones en código nativo, debido a que los mismos comprometen de manera considerable al sistema operativo sobre el cual se ejecutan.

**JRE:** es el entorno de ejecución creado por la empresa *Sun-Microsystems* (FROUFE, AGUSTIN 2000), para que los programas creados usando el lenguaje de programación Java, puedan ejecutarse, el entorno es distribuido gratuitamente por la empresa *Sun-Microsystem*, cada seis meses. JRE viene de *Java Runtime Environment* (Entorno de Ejecución de Java). Este es el soporte para las aplicaciones ya creadas usando el lenguaje Java, una de las características es que el entorno maneja de manera eficiente la administración de memoria para la aplicación, además de poder ejecutar multiproceso.

**Net-FrameWork:** está destinado para dar soporte a un conjunto de lenguajes que conforman el **Visual Studio Net**, creación de la empresa MS-Microsoft, entre estos lenguajes podemos mencionar: C#, Visual C++.NET, Visual Basic.NET entre otros. Visual Studio.NET está orientado a al creación de aplicaciones que se ejecutarán en el sistema operativo MS-Windows, del cual, el mismo MS-Microsoft es propietario y Net-Framework es el apoyo para dichas aplicaciones, la empresa distribuye Net-Framework de manera gratuita para los sistemas operativos creados por MS-Microsoft (ZUKOWSKI, JHON. 2003).

Una vez especificados: La plataforma destino, el lenguaje y el entorno, se debe considerar de forma muy cuidadosa la relación entre el lenguaje y su entorno, de forma muy rigurosa, las versiones de ambos, para lograr una buena compatibilidad entre ambos.

Habiendo ya sido considerado, los aspectos anteriores, empieza el proceso de elaboración y planificación de los procesos que involucrará el desarrollo de la aplicación, ello implica la implementación de tres actividades:

1. Elaboración del algoritmo, destinado a resolver el problema.
2. Programación o implementación computacional del algoritmo.
3. Ejecución y Depuración de la aplicación.

### **Programación**

Programación es la planificación de un conjunto de tareas, en cuanto al orden de su ejecución, principio y fin de dicho conjunto.

La programación computacional, considera planificar el conjunto de tareas de una aplicación, siguiendo un algoritmo, el lenguaje de programación en esta parte se convierte en un aspecto muy importante, debido a que gracias al lenguaje, se llevará a cabo la implementación del software de la aplicación.

En el presente trabajo, se llevará a cabo la implementación computacional de un sistema de control, encargado de automatizar el proceso de asentamiento de suelos, para dicha implementación se usará el lenguaje de programación Java.

El lenguaje de programación Java, aparece en el año 1995 (ZUKOWSKI, JHON. 2003), como lenguaje compatible con Internet y no asociado a una única plataforma destino, desde entonces Java ha evolucionado hasta convertirse en un medio de creación de programas para casi cualquier cosa; Java es creación de James Gosling, desarrollado en los laboratorios de *Sun-Microsystems*.

De acuerdo al libro blanco de Java, disponible en <http://java.sun.com>, los objetivos de diseño de Java eran ser un lenguaje sencillo, orientado a objetos, distribuido, interpretado, robusto, seguro, de arquitectura neutral, portátil, de gran rendimiento, multitarea y dinámico.

**Simple:** en la primera conferencia C++, Hill Joy, cofundador y ahora científico jefe y jefe oficial ejecutivo de *Sun-Microsystems*, dijo que C++ es demasiado complicado, lo que yo quiero es C++--++. En Java eso es lo que él y todo el mundo obtiene. Java incorpora nuevas tareas como un recolector de elementos no utilizados automático (el ++) y elimina aspectos de C++ confusos y muy poco utilizados, como la sobrecarga de operadores.

Otro aspecto de la simplicidad de Java, es que nada en Java es realmente nuevo, si se observa el conjunto de funciones de Java junto a la historia de la informática, se descubre que todo procede de algún otro lugar.

**Orientado a Objetos:** Java está orientado a objetos. Programación orientada a objetos (OOP), es un modo de desarrollar software, describiendo un problema mediante el uso de elementos desde el espacio del problema y no mediante un conjunto de pasos secuenciales que se ejecutarán. Un buen diseño conlleva a componentes reutilizables, extensibles y sostenibles. Estos componentes son lo bastante flexibles como para controlar los cambios que se pueden producir en el entorno a lo largo del tiempo. Dado que el trabajo principal de estos objetos es precisamente intercambiar mensajes entre sí.

Esencialmente, los API Core de Java son una colección de estos componentes previamente creados, llamados bibliotecas de clase. En lugar de crearlo de nuevo

con Java, se comienza con las bibliotecas. Estas bibliotecas estándar han evolucionado a lo largo del tiempo, en más componentes previamente creados con cada nueva edición de Java (ZUKOWSKI, JHON. 2003).

**Distribuido:** desde su comienzo en 1982, la visión de *Sun-Microsystems*, ha sido “**La Red es el Ordenador**”, con Java se quería un lenguaje de programación que reconociera la red y que admitiera el acceso a objetos distribuido tan fácilmente como a los objetos locales, este objetivo aunque presente desde el principio, ha cambiado en cuanto a las posibilidades de Java.

**Interpretado:** Los programas de Java son interpretados. En lugar de que sean compilados a ejecutables nativos, el código de Java es traducido en códigos de bytes no asociados a una plataforma.

Estos códigos de bytes se ejecutan en cualquier plataforma que tenga Java Runtime Environment (JRE), de este modo pueden ejecutarse sin volver a compilarlos o revincularlos. Este proceso puede sugerir que Java es lento, pero esto no es cierto, dado que los códigos neutrales de bytes en realidad poseen una información adicional que puede usarse para optimizar la velocidad y rendimiento en tiempo de ejecución.

**Robusto:** La robustez es la medida de la fiabilidad de un programa. En Java existen varias funciones integradas que mejoran la fiabilidad de programas escritos en Java:

- Java es un lenguaje basado en tipos: El compilador y el cargador de clase aseguran la corrección de todas las llamadas de métodos, lo cual evita las diferencias implícitas entre tipos y las incompatibilidades entre versiones.

- Java no tiene punteros: No se puede hacer referencia a un puntero en la memoria, corrompiendo accidentalmente la memoria o pasarse el fin de una pila. En Java hay referencias en lugar de punteros y no se puede mover su referencia para manipular directamente el espacio en memoria.
- Java realiza automáticamente la recolección de elementos sin uso: Los programadores no pueden olvidarse accidentalmente liberar memoria y no tienen que preocuparse por decidir las partes de la memoria que deben liberarse.
- Java fomenta el uso de interfaces en lugar de clases, las interfaces definen un conjunto de comportamientos y las clases implementan estos comportamientos.

**Seguro:** Java tiene muchas funciones de seguridad integradas: El verificador de código de bit, el cargador de clases y el gestor de seguridad. Todo esto garantiza la seguridad del código que se está ejecutando y evita que el código no realice operaciones no seguras, como leer el disco duro. El último aspecto de la seguridad refiere a la autenticación, autorización y encriptación para proteger la privacidad y asegurar la integridad de datos.

**De Arquitectura Neutral:** La “Neutralidad de la Arquitectura”, refiere a los códigos de bytes neutrales en Java. En lugar de compilarlos en binarios específicos de cada plataforma, los programas de Java se realizan para que se ejecuten en cualquier equipo sin compilarlos o revincularlos. Si una compañía desarrolla nuevo hardware, no tiene que desechar sus inversiones en software,

sino que sólo tiene que colocar un JRE en la nueva plataforma (ZUKOWSKI, JHON. 2003).

**Portátil:** El clásico lema de Java es “**Escribe una vez, Ejecuta en cualquier sitio (EE)**”, el objetivo de Java es ser capaz de tomar el mismo programa y ejecutarlo en cualquier arquitectura. Aunque esta portabilidad no se consigue solamente con los códigos de bytes de plataforma neutral.

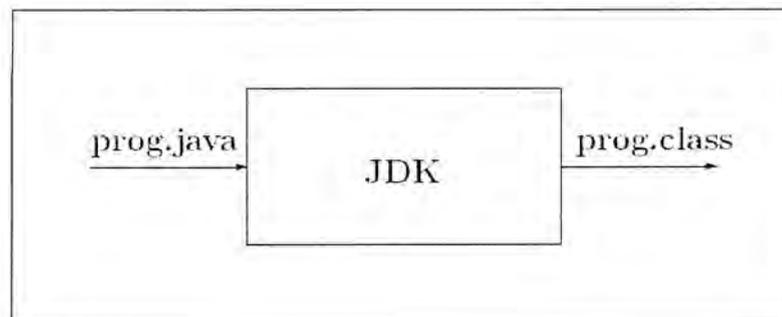
**Alto Rendimiento:** Los códigos de bytes de plataforma neutral realmente pueden convertirse en tiempo de ejecución en código de máquina específico de la CPU, ejecutándose casi tan rápidamente como código compilado C y C++. Hay dos herramientas de traducción en Java para hacer esto de forma automática: La herramienta de primera generación recibe el nombre de compilador Justo a Tiempo (JIT). La herramienta de segunda generación de *Sun-Microsystems*, se llama **HotSpot**. Esencialmente el HotSpot y el compilador JIT hacen lo mismo: traducir en el tiempo de la ejecución al conjunto de instrucciones nativas todo el código de bytes. Sin embargo, el HotSpot supervisa el código mientras se ejecuta y optimiza la velocidad en los bloques que son más frecuentemente ejecutados, en lugar de todo.

**Multitarea:** Puede definirse un subproceso como un contexto en ejecución. Cada uno de estos contextos que son tareas, se realizan en subprocesos por separados. Cuando un programa ejecuta varias tareas o subprocesos simultáneamente, se dice que el programa es multitarea. Los programas multitarea comparten memoria y tienen comunicación entre subprocesos. El lenguaje de programación Java y las bibliotecas estándar de Java, incluyen muchos recursos para ayudarle en este proceso de comunicación, brindando seguridad a los procesos.

**Lenguaje Dinámico:** Java es un lenguaje dinámico, esto no quiere decir que el lenguaje esté continuamente evolucionando, la frase tiene que ver más con las bibliotecas. En concreto, significa que aunque las bibliotecas están continuamente cambiando, los programas no deben volver a vincularse.

Los primeros códigos de bytes de plataforma neutral continuarán con su funcionamiento después que aparezcan nuevas bibliotecas, bien por que *Sun-Microsystems*, presente una nueva versión de Java o por que alguna otra compañía cree alguna biblioteca. Mientras las partes de la biblioteca empleadas no desaparezcan, los programas seguirán funcionando, incluso aunque se agreguen nuevos elementos a las bibliotecas. Esta naturaleza dinámica también se cumple en la preferencia de Java por las interfaces sobre las clases.

Para el proceso de creación de una aplicación usando el lenguaje Java se tiene dos fases, la primera de ellas ligadas por una herramienta llamada JDK (Java Development Kit), esta herramienta recibe un código escrito por el programador, el cual debe residir en un archivo de texto de extensión **.java**, con ello el JDK, transforma dicho código a código de bytes, que residirá en un archivo de extensión **.class**, este archivo class está listo para ejecutarse por el JRE y viene a ser nuestra aplicación.

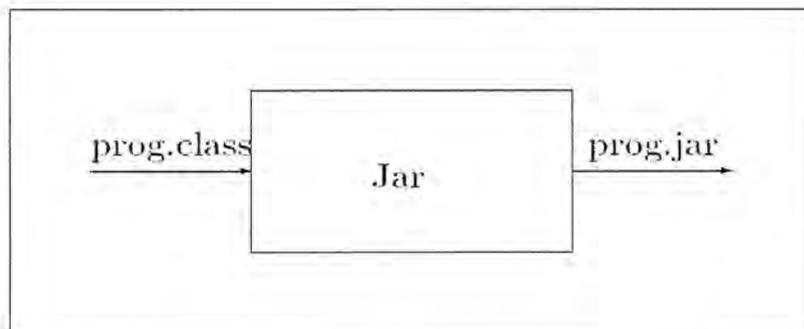


**Figura 4.1:** Creación de un Programa Java: **prog**

Para que una aplicación llamada *prog* se debe considerar la creación de una clase en el interior del archivo **prog.java**, de esta forma el compilador JDK generará el correspondiente **prog.class**.

En el presente trabajo, se desarrolla un sistema de control, para el cual, se diseñan un conjunto de módulos, cada uno de los cuales debe residir en su propio archivo de extensión **.java**, de esta manera se tiene que la entrada al compilador JDK son varios archivos de extensión **.java**.

La siguiente fase, consiste en usar la herramienta de *Sun-Microsystems* llamada **Jar**, esta herramienta, toma de entrada archivos de extensión **class** y los transforma en archivos ejecutables Java o archivos de extensión **.jar**.



**Figura 4.2:** Transformación de Class a Jar

Los archivos con extensión **.jar**, pueden haciendo solamente doble clic sobre ellos. Estos reúnen todo lo concerniente a la aplicación, desde los archivos con extensión **.class** hasta las figuras, etc.

En la actualidad, el JDK disponible es el JDK 1.5.12 con el cual se va a diseñar el sistema de control para el asentamiento de suelos, la ejecución de una aplicación creada con dicha herramienta, se realiza con el JRE 1.5.12. De esta forma se tiene

la compatibilidad, de la aplicación con la máquina virtual de Java anteriormente mencionada.

### **El Sistema de Control JGround**

El sistema de control JGround, es una aplicación software, cuyo objetivo es controlar el fenómeno de la consolidación, desde el punto de vista de su evaluación y prevención para los asentamientos elásticos y por consolidación, el cual ocurre al transcurrir el tiempo.

**JGround** es un sistema implementado, usando el lenguaje de programación Java, en la versión 1.5.12, herramienta distribuida por *Sun-Microsystems*. Para su implementación se hizo uso de herramientas como el editor de texto de Linux, VI y el entorno de ejecución para pruebas como JRE. JGround está desarrollado usando herramientas de software libre.

Como todo sistema de control JGround, recibe datos de entrada y realiza un procedimiento involucrado con su objetivo para posteriormente exteriorizar los resultados mediante datos de salida, esto va en concordancia con el algoritmo anteriormente mencionado.

1. **Especificación del Proyecto:** JGround, recibe por parte del usuario datos relacionados con el proyecto, como nombre del proyecto, el número de zonas de estudio de una determinada área sobre el cual se va a analizar el estudio de la consolidación y el autor del proyecto o el usuario que hace uso de JGround. Los datos en la especificación del proyecto, los puede ingresar el usuario, mediante el teclado o mediante la apertura de un archivo que reside en el ordenador en el cual se encuentra JGround.

2. **Ingreso de Datos:** JGround elabora 5 tablas, en cada uno de los cuales, se tiene especificaciones para cada una de las zonas de estudio, esas tablas están caracterizadas por datos específicos, los cuales son:

**Parámetros de Resistencia:** En esta tabla, para cada zona, se especifica: El ángulo de fricción del suelo  $\phi$ , la cohesión que existe en el suelo en gramos sobre centímetro cuadrado  $c(kg/cm^2)$  y los pesos específicos del suelo saturado y no saturado  $\gamma_1, \gamma_2$ .

**Propiedades Elásticas:** Están caracterizadas por la elasticidad que posee el suelo  $E_s$  y su coeficiente de Poisson  $\mu_s$ , estos valores toman relevancia en el cálculo del asentamiento elástico.

**Estructura del Suelo:** Existen siete valores que nos indican como está compuesto el suelo, en cuanto a sus alturas y pesos específicos y además, se menciona el valor  $q(kg/cm^2)$ , llamado *esfuerzo neto transmisible*, fuerza que el suelo va a soportar como resultado de la aplicación de una carga por la construcción de una edificación.

**Dimensiones de la Zapata:** Aquí se especifican las dimensiones de la zapata que va a soportar la estructura, las dimensiones se determinan por su ancho  $B$ , largo  $L$  y su profundidad con respecto al nivel del suelo  $D_f$ .

**Datos de Laboratorio:** Los datos en esta tabla son resultados que se obtienen en un laboratorio, dedicado a suelos, mediante pruebas ASTM que se explicó anteriormente. Los datos en esta tabla son: Coeficiente de consolidación  $C_c$ , cual nos indica la pendiente de la recta que describe la

carga aplicada al suelo con respecto a la relación de vacíos que experimenta el suelo. Luego el coeficiente de expansibilidad  $C_s$ , al retirar la carga aplicada al suelo, el suelo empieza a expandirse, esta expansión se determina mediante dicho coeficiente. A continuación se tiene la relación de vacíos inicial  $e_0$  y luego el índice de preconsolidación  $P_c$ , el cual nos indica como estuvo el suelo sometido en el pasado. Finalmente se tiene el coeficiente de consolidación vertical  $C_v$ , este coeficiente nos permite determinar los tiempos para determinar los tiempos para diversos grados de consolidación.

Cada uno de los datos que se encuentran presentes en estas tablas puede ser recibido también mediante la apertura de un archivo.

- 3. Clasificación:** Una vez ingresado los datos, JGround, empieza la clasificación de todos ellos, de forma interna, creando para cada uno de ellos un conjunto de datos específico a fin de que contenga todo lo relacionado con una zona determinada. Para tal efecto, se extraen de cada una de las 5 tablas, datos que corresponden a cada zona de estudio. El sistema internamente genera una tabla y con ello realiza la creación de un conjunto de datos en el cual incorpora la tabla correspondiente a dicha zona.
- 4. Distribución:** Diversos módulos que componen JGround, reciben cada uno de los conjuntos de datos que se realizó en la clasificación. Estos módulos los procesan para que tengan un aspecto más formal, para el cálculo de las magnitudes relacionadas con el fenómeno de la consolidación.

5. **Ejecución:** JGround, recibe los datos previamente, preprocesados en la clasificación y distribución y los reparte en los módulos de ejecución, los cuales, tienen como finalidad calcular las distintas magnitudes correspondientes al asentamiento elástico y el asentamiento por consolidación.
6. **Recopilación de Resultados:** Cada uno de los resultados en la parte de ejecución, se encuentran internamente en el sistema de control. En la parte de recopilación, todo esos datos, se agrupan por zonas y se crean interfaces de salida para cada una de las zonas, a fin de que se muestren sus resultados, para ello intervienen una serie de módulos de JGround, aquí interviene un módulo específico para la gráfica de la consolidación con respecto al tiempo.
7. **Salida de Datos:** Una vez calculadas todas las magnitudes relacionadas con la consolidación, crea un mecanismo de salida de datos, a través del flujo de datos hacia el exterior, con ello se muestra el reporte de la consolidación del área de estudio. Para dicho reporte, se crea un archivo **report.doc**, el cual puede ser visualizado con el programa MS-WORD o con el OPEN-OFFICE.

**JGround** al ser desarrollado en el lenguaje de programación Java, se ejecuta en computadores personales que tienen instalada la herramienta de ejecución Java, esta herramienta, es llamada JRE, la versión es la 1.5.12 , es posible ejecutar el sistema de control con el JRE 1.5.12 o superior, como por ejemplo la 1.6.3, etc. Esto se debe que se cumple la compatibilidad hacia atrás. Las herramientas internas de JRE 1.5.12 se encuentran presentes en las versiones posteriores.

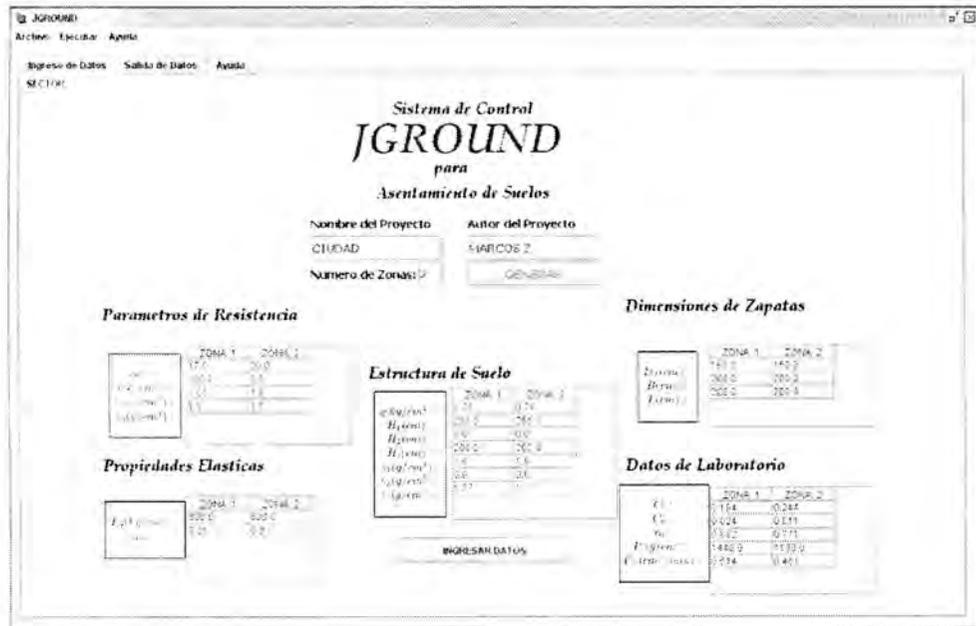


Figura 4.3: JGround

### Módulos de Ejecución para JGround

El sistema de control JGround se compone de diversos módulos, estos módulos interactúan con un módulo que constituye el kernel del sistema de control, es **jground.class**, el cual dirige toda la actividad del sistema de control, desde el inicio al recibir los datos provenientes de un proyecto hasta el término, cuando se realiza un reporte de la consolidación. Los módulos son:

**Jgroud.class:** Es el kernel del sistema de control JGround, contiene el punto de inicio de la aplicación, JGround crea las variables y ejecuta procedimientos encargados de crear el entorno donde se va a desarrollar el estudio de la consolidación, en el entorno creado, presenta opciones mediante menús y crea paneles que contendrán resultados y la ayuda del sistema de control.

**NewProject.class:** Este módulo es invocado por JGround, para que el usuario ingrese los datos del proyecto como: Nombre del proyecto, número de zonas del

proyecto y autor del mismo. NewProject una vez que recibe estas especificaciones, el módulo los almacena para una posterior manipulación de los datos.

**panelIN.class:** Este módulo es invocado por JGround para que el usuario ingrese los datos, al iniciar su actividad, recibe los datos que NewProject.class obtuvo, genera 5 tablas e internamente reserva el nombre del proyecto y el autor del mismo. Las tablas que genera NewProject son: Parámetros de Resistencia, propiedades elásticas, estructura del suelo, dimensiones de la zapatas y datos de laboratorio, cada una de esas tablas presenta columnas para cada una de las zonas que se especificó.

**panelOUT.class:** Este módulo se encarga de exteriorizar los resultados obtenidos por el sistema de control, para ello, elabora el número de paneles que va acorde con el número de zonas en el proyecto. A su vez internamente, genera una cadena de caracteres, que contiene los resultados de todas las zonas, esto será un valor de retorno para el kernel del sistema de control, dado que esto va a constituir una parte muy importante para la elaboración del reporte.

**PanelLine.class:** Este módulo, genera la interfaz inicial para ser ubicado como un panel en el procedimiento realizado por **panelOUT.class**.

**InfoPanel.class:** Este módulo genera un panel para ser presentado al usuario, en dicho panel, se especifica a grandes rasgos lo que realiza el sistema de control, este panel se dispone, para ser visto en cualquier momento.

**NewPanel.class:** Para cada zona en el proyecto, se tiene un conjunto de resultados, cada conjunto es enviado por separado a NewPanel.class para que este

módulo se encargue de generar la interfaz, mediante tablas y etiquetas, para que con ello, puedan presentarse los resultados de esa zona.

**Jwork.class:** Este módulo se encarga de realizar las operaciones que involucran el cálculo de los asentamientos elásticos y por consolidación, para ello, inicializa tablas que maneja internamente, los cuales serán necesarios para el cálculo de las cargas admisibles en una determinada zona, que a su vez serán usadas en la evaluación del asentamiento elástico y por consolidación.

**QPanel.class:** Módulo encargado de realizar el trazado de la gráfica que mostrará el asentamiento con respecto al tiempo, dicho asentamiento, estará trazado para el 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80% y 90%, además de la gráfica, se indicará el asentamiento por consolidación total de la zona.

**FileChooser.class:** Este módulo, presenta una interfaz de marco para que el usuario pueda ingresar los datos desde un archivo de disco, esto sucede cuando se apertura un archivo para cargar datos desde el mismo, esta opción está en **Abrir proyecto existente**.

**aboutAuthor.class:** Este módulo es invocado, cuando desde la interfaz de presentación, se ingresa al menú **ayuda** y se activa la opción **Acerca del Autor**, entonces, el módulo **aboutAuthor.class** genera una interfaz en el cual presenta datos relacionados con el autor del sistema de control.

**AboutProgram.class:** Este módulo se activa cuando desde el menú **Ayuda**, se presiona sobre la opción **Acerca de JGround**, entonces se elabora una interna, en el cual se muestra información sobre el sistema de control, el lenguaje con el cual se desarrolló y la plataforma necesaria para su ejecución.

**aCreate.class:** Sirve de vínculo para que el módulo panelIN, genere las tablas relacionadas con los parámetros de resistencia, propiedades elásticas, estructura del suelo, dimensiones de zapatas y datos de laboratorio.

**aINDATA.class:** Sirve de vínculo para que el módulo panelIN recepcione los datos que se ingresaron en las tablas y con ello se disponga todos estos datos de las zonas en una memoria, para que más adelante se tomen datos desde allí.

**aNew.class:** Este módulo es el vínculo para activar un procedimiento de JGround, el cual tiene por finalidad, presentar una interfaz para la creación de un nuevo proyecto.

**aOpen.class:** Este módulo se encarga de activar una interfaz para que el sistema de control, se conecte con el sistema de archivos de la computadora a fin de localizar un \*.jg que contenga especificaciones de un proyecto, para que así, se cargue y el sistema de control pueda leer los datos que residen en el, esto viene a ser el flujo de entrada para JGround.

**aEnd.class:** Es el vínculo para que termine sus actividades el sistema de control. Cuando se activa la opción **Salir de JGround** del menú archivo, se activa el módulo aEnd.class, el cual redirección su actividad a la terminación del funcionamiento de JGround.

**aEdit.class:** Este módulo, activa el funcionamiento de JGround, para el inicio de los cálculos que involucra el proceso de consolidación. El kernel, espera que este módulo sea invocado para que se editen los datos ingresados en la tabla y que ahora resida en una fuente interna, para que el kernel acceda a ellos y realice los cálculos necesarios.

## Enlazamiento en JGround

En esta sección se describe, las conexiones internas del kernel con los módulos en el sistema de control JGround. El punto de entrada a la aplicación se encuentra en el módulo **jground.class**, con este módulo se inicia y termina su actividad el sistema de control.

El kernel invoca a los demás módulos a realizar ciertas actividades, enviando a ellos datos de entrada y recibiendo de ellos directa o indirectamente datos con los cuales, trabajará para obtener los resultados concernientes a la consolidación.

Los datos ingresan al sistema de control, mediante dos vías: Mediante el teclado, cuando el usuario activa la opción **Nuevo Proyecto** del menú archivo, como lo muestra la figura 4.4:

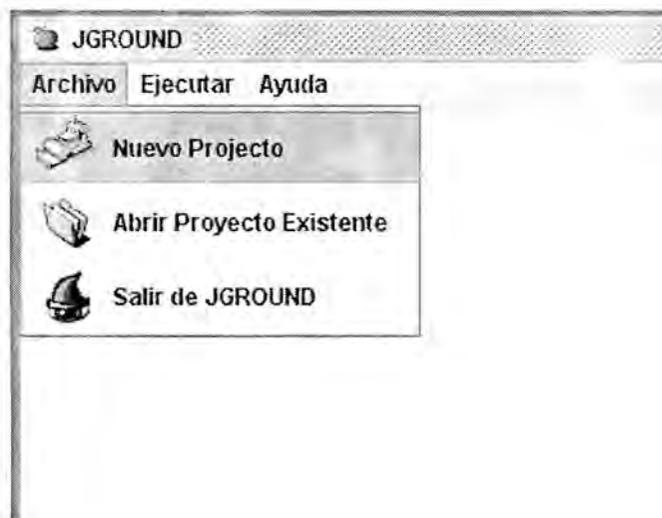


Figura 4.4: Nuevo Proyecto

La otra vía para el ingreso de datos, es cuando se carga un archivo, del sistema de archivos y se pasa como argumento al sistema de control, para ello se dispone de la opción, **Abrir Proyecto Existente**, en el menú archivo.

De esta forma se tiene un flujo de datos de entrada en doble vía.

Una vez que los datos se encuentran en el sistema, el kernel invoca al módulo panelIN a fin de que este módulo reciba los datos, esto sucede al leer los datos que se encuentran en las tablas.

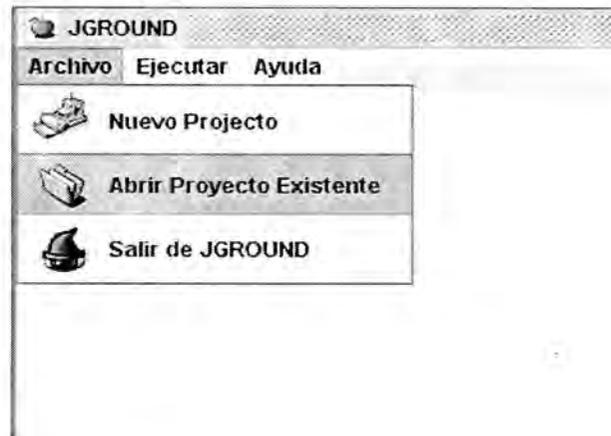


Figura 4.5: Abrir Proyecto Existente

El módulo panelIN.class, se encarga de generar un conjunto de datos que posteriormente la aplicación utilizara para realizar los cálculos del asentamiento.

La siguiente actividad, ocurre cuando el usuario, activa la opción **Ejecutar** del menú ejecutar en la interfaz del sistema de control.

Cuando se activa la opción indicada, el kernel carga un conjunto de datos generados por panelIN.class, una vez cargados los datos, se invoca al módulo panelOUT.class enviándoles los datos previamente cargados.

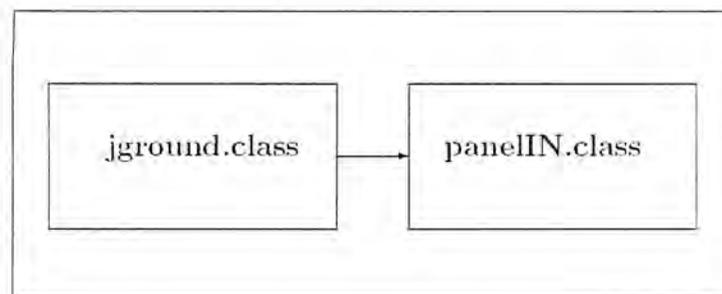


Figura 4.6: JGround.class invoca a PanelIN.class

Este invoca al modulo NewPanel.class para que este módulo, inicie una clasificación de los datos de las tablas de acuerdo a la zona, una vez que realiza esto, el módulo NewPanel trabaja con dos módulos, uno para realizar los cálculos y otro para el trazado de la gráfica de los días versus la consolidación.

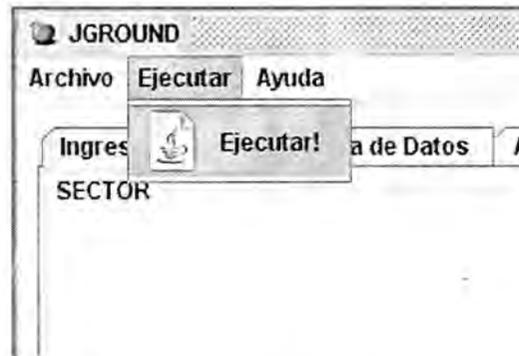


Figura 4.7: Ejecutar el procedimiento de Consolidación

NewPanel activa dos módulos: jwork.class y QPanel. El módulo jwork.class realiza las operaciones que tienen que ver con las capacidades últimas de carga, los asentamientos elásticos y los asentamientos por consolidación.

Los valores que retorna el módulo jwork.class, los selecciona NewPanel.class después de lo cual, envía datos a QPanel.class para que dicho módulo realice el trazado de la gráfica de los días respecto al asentamiento por consolidación en centímetros.

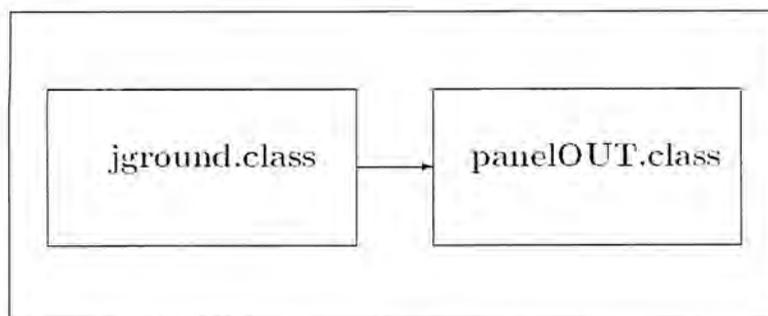
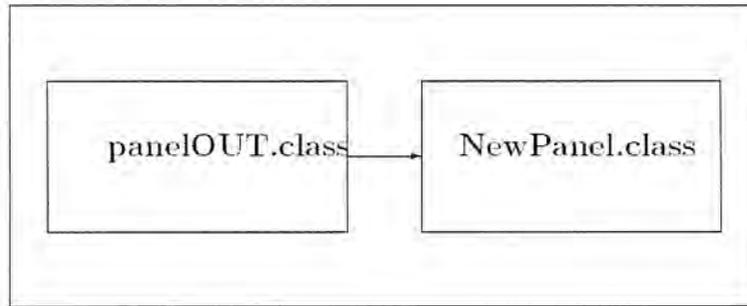


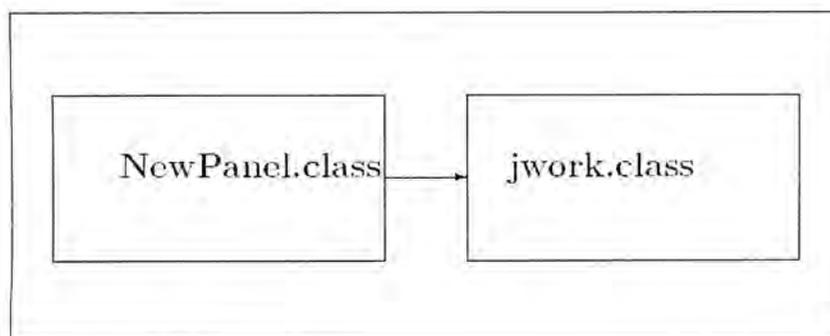
Figura 4.8: jground.class invoca a panelOUT.class



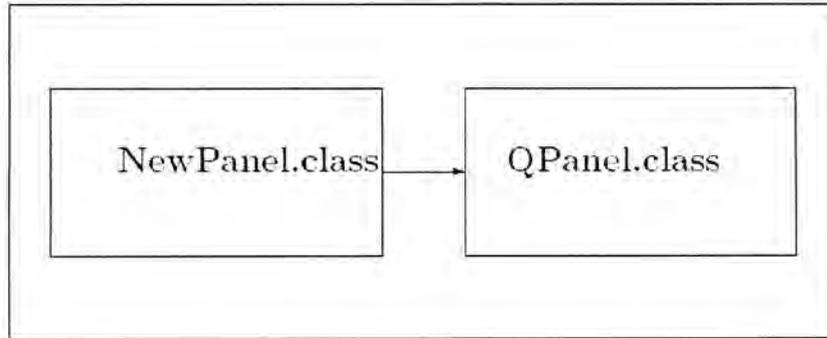
**Figura 4.9:** panelOUT.class activa el módulo NewPanel.class

El módulo NewPanel.class, obtiene de esta forma los resultados numéricos de jwork.class y el trazado de la gráfica de QPanel.class, este conjunto de resultados que es para cada zona, se muestran en las interfaces producidas por panelOUT.class a la vez internamente el reporte se elabora para ser emitido mediante un flujo de datos hacia el exterior en un archivo llamado **report.doc**.

Los distintos paneles, se integran dentro de la interfaz que es aquella que el módulo incorpora al núcleo del sistema de control JGround, el cual administra los distintos paneles para los distintos procesos que tienen lugar durante la ejecución de un proyecto en el sistema de control.



**Figura 4.10:** NewPanel.class activa el módulo jwork.class



**Figura 4.11:** NewPanel.class activa el módulo QPanel.class

### Ejecución en JGround

En esta sección describiremos paso a paso, como el sistema de control realiza un estudio del proceso de asentamiento de suelos, en base a estudios realizados a la estructura del suelo en la ciudad universitaria de la UNASAM que se sitúa en Shancayan.

En este caso el area total de la ciudad se ha dividido en cuatro zonas para cada una de las cuales, se tiene el análisis respectivo en cuanto a sus datos de laboratorio:

DATOS	ZONA I	ZONA II	ZONA III	ZONA IV
$C_r$	0,194	0,244	0,114	0,148
$C_r$	0,024	0,011	0,010	0,031
$e_0$	0,0682	0,771	0,611	0,585
$P_c(g/cm^2)$	1440	1130	1570	1310
$C_v(cm^2/min)$	0,514	0,461	0,246	0,610

**Tabla 4.12:** Datos de Laboratorio

Los resultados del cuadro anterior fueron obtenidos del CENTRO JAPONÉS DE INVESTIGACIONES SISMICAS Y MI TIGACIONES DE DESASTRES, de la **Universidad Nacional de Ingeniería** (Facultad de Ingeniería Civil).

Las propiedades elásticas son:

DATOS	ZONA I	ZONA II	ZONA III	ZONA IV
$E_s(kg/cm^2)$	800	600	800	400
$\mu_s$	0,25	0,30	0,25	0,30

**Tabla 4.13:** Propiedades Elásticas

Los parámetros de resistencia son:

DATOS	ZONA I	ZONA II	ZONA III	ZONA IV
$\phi(\circ)$	17	20	22	24
$c(kg/cm^2)$	0,20	0,00	0,00	0,00
$\gamma_1(g/cm^3)$	2,06	1,80	2,06	1,84
$\gamma_2(g/cm^3)$	1,8	1,70	1,75	1,75

**Tabla 4.14:** Parámetros de Resistencia

Las dimensiones de la zapatas y la estructura del suelo son:

DATOS	ZONA I	ZONA II	ZONA III	ZONA IV
$D_f(cm)$	160	150	130	140
$B(cm)$	200	200	200	200
$L(cm)$	200	200	200	200

**Tabla 4.15:** Dimensiones de las Zapatas

DATOS	ZONA I	ZONA II	ZONA III	ZONA IV
$q_0(kg/cm^2)$	0,78	0,78	0,78	0,78
$H_1(cm)$	255	255	255	255
$H_2(cm)$	0,00	0,00	0,00	0,00
$H_c(cm)$	200	200	200	200
$\gamma_1(g/cm^3)$	1,90	1,90	1,90	1,90
$\gamma_2(g/cm^3)$	0,00	0,00	0,00	0,00
$\gamma_c(g/cm^3)$	1,77	1,77	1,77	1,77

**Tabla 4.16:** Estructura del Suelo

Los datos de las tablas anteriores, se encuentran en un archivo de proyecto llamado **dat.jg**. Este será el flujo de entrada para el sistema de control.

A continuación se ejecuta el sistema de control JGround en el cual vamos a introducir los datos de las tablas anteriores, activando la opción de **Abrir Proyecto Existente**, del menú archivo, para ello se selecciona el archivo llamado **dat.jg** y se tienen los datos cargados en el sistema.

## Sistema de Control **JGROUND** para Asentamiento de Suelos

Nombre del Proyecto	Autor del Proyecto
SHANCAYAN	MARCOS
Numero de Zonas: 4	GENERAR

Figura 4.17: Especificación del Proyecto

### **Parametros de Resistencia**

	ZONA: 1	ZONA: 2	ZONA: 3
$\phi(\circ)$ :	17.0	20.0	22.0
$c(g/cm^2)$ :	0.2	0.0	0.0
$\gamma_1(g/cm^3)$ :	2.06	1.8	2.06
$\gamma_2(g/cm^3)$ :	1.8	1.7	1.75
	◀		▶

Figura 4.18: Parámetros de Resistencia

### **Propiedades Elásticas**

	ZONA: 1	ZONA: 2	
$E_s(kg/cm^2)$ :	800.0	600.0	8
$\mu_s$ :	0.25	0.3	0
	◀		▶

Figura 4.19: Propiedades Elásticas

### Estructura de Suelo

	ZONA: 1	ZONA: 2	ZONA: 3
$q(kg/cm^2)$ :	0,78	0,78	0,78
$H_1(cm)$ :	255,0	255,0	255,0
$H_2(cm)$ :	0,0	0,0	0,0
$H_3(cm)$ :	200,0	200,0	200,0
$\gamma_1(g/cm^3)$ :	1,9	1,9	1,9
$\gamma_2(g/cm^3)$ :	0,0	0,0	0,0
$\gamma_3(g/cm^3)$ :	1,77	1,77	1,77

INGRESAR DATOS

Figura 4.20: Estructura del Suelo

### Dimensiones de Zapatas

	ZONA: 1	ZONA: 2	ZONA: 3
$D_f(cm)$ :	160,0	150,0	130,0
$B(cm)$ :	200,0	200,0	200,0
$L(cm)$ :	200,0	200,0	200,0

Figura 4.21: Dimensiones de las Zapatas

El siguiente paso, es ingresar los datos al sistema, mediante la presión sobre el botón, **INGRESAR DATOS**, con ello el sistema tendrá la referencia de los datos con los cuales va a realizar el análisis del asentamiento de suelos.

Una vez que los datos han ingresado al sistema, se ejecuta el procedimiento para el cálculo de los asentamientos, para ello se activa la opción **Ejecutar!** Del menú ejecutar, del sistema de control.

Realizados los procedimientos del sistema de control sobre los datos, el sistema de control posee los resultados y los muestra en el panel de **Datos de Salida**.

Debido a que existen cuatro zonas en el presente proyecto, entonces en el panel **Datos de Salida**, existen cuatro subpaneles cada uno de los cuales tiene los resultados sobre el asentamiento elástico y por consolidación de una zona determinada.

Cada uno de estos cuatro subpaneles tiene la siguiente información:

**Capacidad de Carga** ( $kg/cm^2$ ): Se muestra la información sobre la capacidad de carga última y la capacidad de carga admisible de acuerdo a Terzaghi, Vesic y Meyerhof.

**Asentamiento Elástico** ( $cm$ ): Se muestra el resultado del asentamiento elástico o inmediato mediante Terzaghi, Vesic y Meyerhof.

**Asentamiento por Consolidación** ( $cm$ ): Se muestra el resultado del asentamiento por consolidación para el 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%, para cada uno de los cuales también se muestra el tiempo en días para que se lleve a cabo el porcentaje de asentamiento correspondiente.

**Gráfica Días – Consolidación:** Se muestra la gráfica del asentamiento por consolidación desde el 10% hasta el 90% mediante dicha gráfica puede evaluarse como va teniendo lugar el asentamiento por consolidación y se muestra el asentamiento total en milímetros.

A continuación se muestra los resultados obtenidos por el sistema de control.

## ZONA I

### Capacidad de Carga(kg/cm<sup>2</sup>) - Cimentaciones

	Terzaghi		Vesic		Meyerhof	
	Corrido	Cuadrado	Corrido	Rectangular	Corrido	Rectangular
$q_u$ :	2.02	1.93	2.1	2.23	1.72	2.03
$q_a$ :	0.67	0.67	0.7	0.7	0.57	0.57

Figura 4.22: Capacidad de Carga de la zona I

### Asentamiento Elastico(cm)

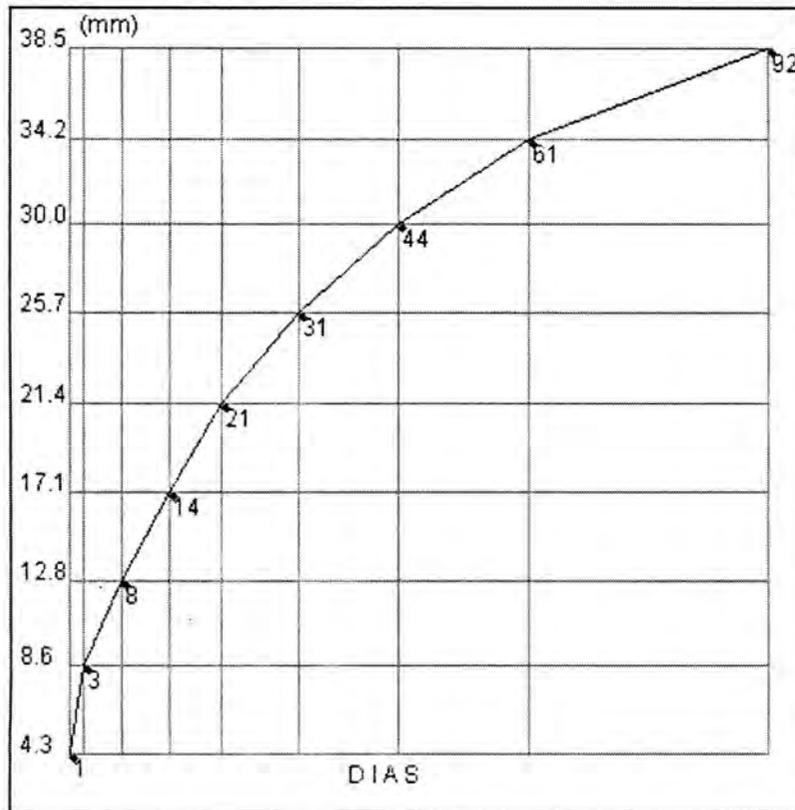
	Terzaghi		Vesic		Meyerhof	
	Corrido	Cuadrado	Corrido	Rectangular	Corrido	Rectangular
$S_e$ :	0.18	0.18	0.18	0.18	0.15	0.15

Figura 4.23: Asentamiento Elástico de la zona I

### Asentamiento por Consolidacion(cm)

	30%	40%	50%	60%	70%	80%	90%
$U\%$ :	64	13.61	21.38	31.21	43.82	61.29	91.76
$t_{dias}$ :	0.7	0.13	0.2	0.29	0.41	0.57	0.85
$T_v$ :	28	1.71	2.14	2.57	3.0	3.42	3.85
$S_c$ :							

Figura 4.24: Asentamiento por Consolidación de la zona I



**Dias versus Consolidación(en mm)**

Asentamiento Total:  mm

Figura 4.25: Gráfica Dias – Consolidación de la zona I

**ZONA II**

**Capacidad de Carga(kg/cm<sup>2</sup>) - Cimentaciones**

	Terzaghi		Vesic		Meyerhof	
	Corrido	Cuadrado	Corrido	Rectangular	Corrido	Rectangular
$q_u$ :	2.55	2.42	2.6	2.81	2.15	2.59
$q_a$ :	0.85	0.85	0.87	0.87	0.72	0.72

Figura 4.26: Capacidad de Carga de la zona II

### Asentamiento Elastico(cm)

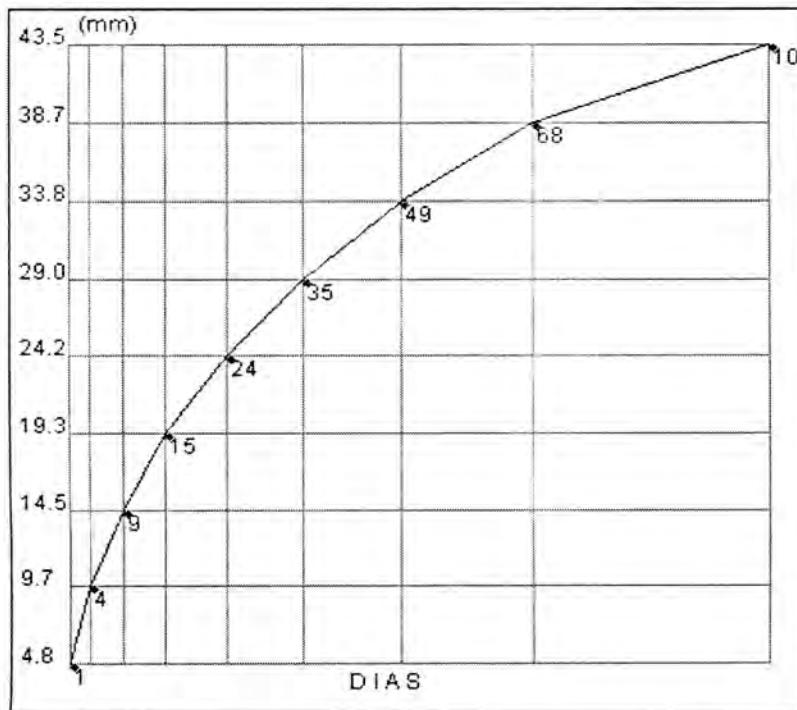
$S_e$ :	Terzaghi		Vesic		Meyerhof	
	Corrido	Cuadrado	Corrido	Rectangular	Corrido	Rectangular
	0,29	0,29	0,3	0,3	0,25	0,25

Figura 4.27: Asentamiento Elástico de la zona II

### Asentamiento por Consolidacion(cm)

$U\%$ :	30%	40%	50%	60%	70%	80%	90%	
	$t_{dias}$ :	15,18	23,84	34,8	48,86	68,34	102,31	
	$T_v$ :	0,07	0,13	0,2	0,29	0,41	0,57	0,85
	$S_c$ :	4,5	1,93	2,42	2,9	3,38	3,87	4,35

Figura 4.28: Asentamiento por Consolidación de la zona II



### Días versus Consolidacion(en mm)

Asentamiento Total:  mm

Figura 4.29: Gráfica Días – Consolidación de la zona II

### ZONA III

#### Capacidad de Carga(kg/cm<sup>2</sup>) - Cimentaciones

	Terzaghi		Vesic		Meyerhof	
	Corrido	Cuadrado	Corrido	Rectangular	Corrido	Rectangular
	$q_u$ :	3,14	2,93	3,25	3,38	2,62
$q_a$ :	1,05	1,05	1,08	1,08	0,87	0,87

Figura 4.30: Capacidad de Carga de la zona III

#### Asentamiento Elastico(cm)

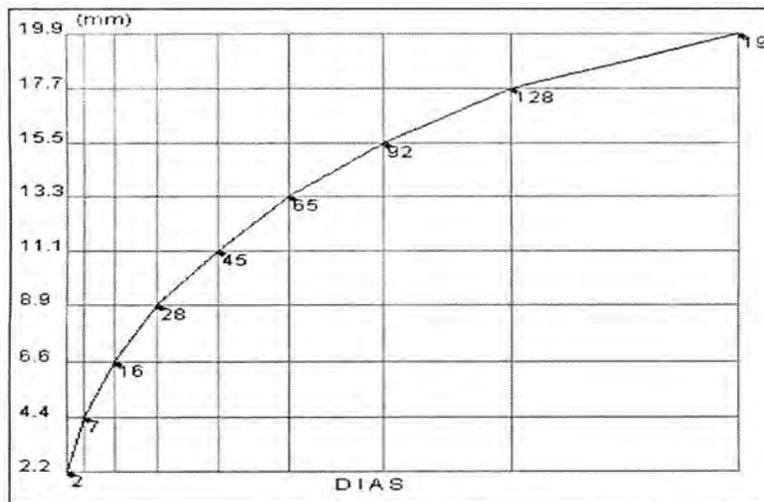
	Terzaghi		Vesic		Meyerhof	
	Corrido	Cuadrado	Corrido	Rectangular	Corrido	Rectangular
	$S_e$ :	0,28	0,28	0,28	0,28	0,23

Figura 4.31: Asentamiento Elástico de la zona III

#### Asentamiento por Consolidación(cm)

	10%	20%	30%	40%	50%	60%	70%	
	$U\%$ :	1,77	7,1	15,97	28,44	44,67	65,21	91,56
	$t_{días}$ :	0,01	0,03	0,07	0,13	0,2	0,29	0,41
$T_v$ :	0,22	0,44	0,66	0,89	1,11	1,33	1,55	
$S_c$ :								

Figura 4.32: Asentamiento por Consolidación de la zona III



#### Días versus Consolidación(en mm)

Asentamiento Total:  mm

Figura 4.33: Gráfica Días – Consolidación de la zona III

## ZONA IV

### Capacidad de Carga(kg/cm<sup>2</sup>) - Cimentaciones

$q_u$ :	<i>Terzaghi</i>		<i>Vesic</i>		<i>Meyerhof</i>	
	Corrido	Cuadrado	Corrido	Rectangular	Corrido	Rectangular
$q_a$ :	4,1	3,84	4,09	4,44	3,4	4,21
	1,37	1,37	1,36	1,36	1,13	1,13

Figura 4.34: Capacidad de Carga de la zona IV

### Asentamiento Elastico(cm)

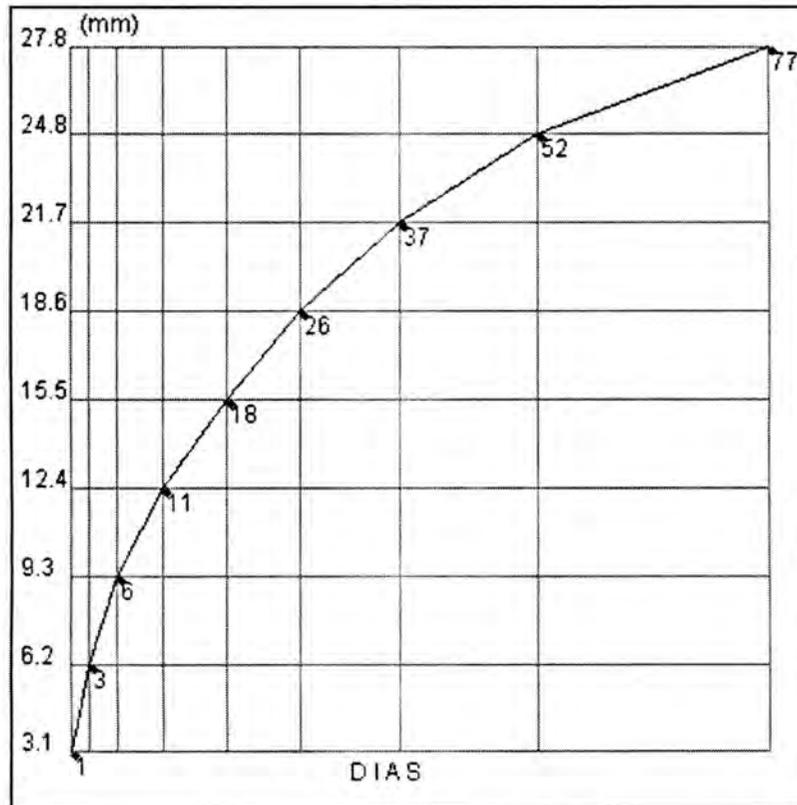
$S_e$ :	<i>Terzaghi</i>		<i>Vesic</i>		<i>Meyerhof</i>	
	Corrido	Cuadrado	Corrido	Rectangular	Corrido	Rectangular
	0,7	0,7	0,69	0,69	0,58	0,58

Figura 4.35: Asentamiento Elástico de la zona IV

### Asentamiento por Consolidacion(cm)

$U\%$ :	10%	20%	30%	40%	50%	60%	70%
$t_{dias}$ :	0,72	2,86	6,44	11,47	18,02	26,3	36,92
$T_v$ :	0,01	0,03	0,07	0,13	0,2	0,29	0,41
$S_c$ :	0,31	0,62	0,93	1,24	1,55	1,86	2,17

Figura 4.36: Asentamiento por Consolidación de la zona IV



***Dias versus Consolidacion(en mm)***

Asentamiento Total:  mm

Figura 4.37: Gráfica Dias – Consolidación de la zona IV

## V. DISCUSIÓN

### Aportes al Esfuerzo Vertical

En el presente trabajo se desarrolla un aporte a la capacidad de carga y por ende al asentamiento admisible, este aporte consiste en un análisis riguroso a la formulación de Bousinessq obteniendo ciertos resultados y extendiendo los resultados del mismo (BRAJA, DAS. 2001).

La cantidad:

$$dP = \frac{3q_0(dx dy)z^3}{2\pi(x^2 + y^2 + z^2)^{5/2}}$$

Dicho esfuerzo Bousinessq da como puntual. El incremento del esfuerzo en el punto A, debajo de una superficie rectangular puede analizarse para una carga  $q_0$  constante con lo que:

$$\Delta P = q_0 \Delta M$$

Considerando:

$$\Delta M = \frac{3z^2}{2\pi(x^2 + y^2 + z^2)^{5/2}} \Delta x \Delta y$$

Para una región rectangular de dimensiones  $B$  de ancho y  $L$  de largo, generamos una partición sobre dicha región:

$$\begin{aligned} 0 &= x_0 < x_1 < x_2 < \dots < x_n = B \\ 0 &= y_0 < y_1 < y_2 < \dots < y_m = L \end{aligned}$$

En cada rectángulo  $\Delta x_i \Delta y_j$  que se genera mediante dicha partición tenemos un incremento del esfuerzo  $\Delta P_{ij}$  dado por:

$$\Delta P_{ij} = q_0 \Delta M_{ij}$$

El límite de la suma del incremento del esfuerzo para cada una de esos pequeños rectángulos sucede cuando  $m \rightarrow \infty$  y  $n \rightarrow \infty$  con lo que tenemos:

$$I = \lim_{m, n \rightarrow \infty} \sum_{i=1}^{i=n} \sum_{j=1}^{j=m} \Delta M_{ij}$$

De esta manera tenemos la cantidad escalar, denominada *factor de influencia* para el incremento del esfuerzo vertical. Dicha cantidad escalar podemos expresarla mediante la integral:

$$I = \int_0^L \int_0^B \frac{3z^3}{2\pi(x^2 + y^2 + z^2)^{5/2}} dx dy$$

Para evaluar dicha integral la expresamos como:

$$I = \frac{3z^3}{2\pi} \int_0^L \int_0^B \frac{dx dy}{(x^2 + y^2 + z^2)^{5/2}}$$

Y procedemos a evaluar:

$$\begin{aligned} \int_0^L \int_0^B \frac{dx dy}{(x^2 + y^2 + z^2)^{5/2}} &= \int_0^L \left( \int_0^B \frac{dx}{(x^2 + y^2 + z^2)^{5/2}} \right) dy \\ \int_0^L \int_0^B \frac{dx dy}{(x^2 + y^2 + z^2)^{5/2}} &= \frac{B}{3} \int_0^L \frac{3y^2 + 3z^2 + 2B^2}{(y^2 + z^2)^2 (B^2 + y^2 + z^2)^{3/2}} dy \end{aligned}$$

Con lo cual:

$$\int_0^L \int_0^B \frac{dx dy}{(x^2 + y^2 + z^2)^{5/2}} = \frac{1}{3z^3} \left( \frac{BLz(B^2 + L^2 + 2z^2)}{(B^2 + z^2)(L^2 + z^2)\sqrt{B^2 + L^2 + z^2}} + \tan^{-1} \left( \frac{BL}{z\sqrt{B^2 + L^2 + z^2}} \right) \right)$$

Reemplazando en la expresión con I se tiene que:

$$I = \frac{1}{2\pi} \left( \frac{BLz(B^2 + L^2 + 2z^2)}{(B^2 + z^2)(L^2 + z^2)\sqrt{B^2 + L^2 + z^2}} + \tan^{-1} \left( \frac{BL}{z\sqrt{B^2 + L^2 + z^2}} \right) \right)$$

Se hace:

$$m = \frac{B}{z} \quad \text{Y} \quad n = \frac{L}{z}$$

Con lo cual se tiene:

$$I = \frac{1}{2\pi} \left( \frac{mn(m^2 + n^2 + 2)}{(m^2 + 1)(n^2 + 1)\sqrt{m^2 + n^2 + 1}} + \tan^{-1} \frac{mn}{\sqrt{m^2 + n^2 + 1}} \right)$$

Fórmula que difiere con la presentada por Boussinesq y publicada por Braja M.

Das en el 2001, la cual es:

$$I = \frac{1}{4\pi} \left( \frac{2mn\sqrt{m^2 + n^2 + 1}(m^2 + n^2 + 2)}{(m^2 + 1)(n^2 + 1)(m^2 + n^2 + 1)} + \tan^{-1} \frac{2mn\sqrt{m^2 + n^2 + 1}}{m^2 + n^2 + 1 - m^2n^2} \right)$$

La fórmula anterior, para ciertos valores de I para B/z y L/z bastantes pequeños, no concuerda con los valores experimentales obtenidos por Newmark en 1935, analizando, se tiene que esto sucede debido a que el argumento en la función  $\tan^{-1}$  cambia bruscamente de comportamiento, razón por la cual, en el presente trabajo se presente la siguiente formulación:

1. Si  $m^2 + n^2 + 1 \geq m^2n^2$ :

$$I = \frac{1}{4\pi} \left( \frac{2mn\sqrt{m^2 + n^2 + 1}(m^2 + n^2 + 2)}{(m^2 + 1)(n^2 + 1)(m^2 + n^2 + 1)} + \tan^{-1} \frac{2mn\sqrt{m^2 + n^2 + 1}}{m^2 + n^2 + 1 - m^2n^2} \right)$$

2. Si  $m^2 + n^2 + 1 < m^2n^2$ :

$$I = \frac{1}{4\pi} \left( \frac{2mn\sqrt{m^2 + n^2 + 1}(m^2 + n^2 + 2)}{(m^2 + 1)(n^2 + 1)(m^2 + n^2 + 1)} + \pi + \tan^{-1} \frac{2mn\sqrt{m^2 + n^2 + 1}}{m^2 + n^2 + 1 - m^2n^2} \right)$$

Con ello se tiene que los resultados del cálculo de  $I$  concuerdan con los valores obtenidos por Newmark.

Un aporte de la presente de la presente tesis es la reformulación de la evaluación del factor de influencia mediante la fórmula:

$$I = \frac{1}{2\pi} \left( \frac{mn(m^2 + n^2 + 2)}{(m^2 + 1)(n^2 + 1)\sqrt{m^2 + n^2 + 1}} + \tan^{-1} \frac{mn}{\sqrt{m^2 + n^2 + 1}} \right)$$

Con frecuencia el esfuerzo debajo del centro de una superficie rectangular es de importancia y se da la siguiente formulación:

$$\Delta p = q_0 I_c$$

Donde:

$$I_c = \frac{2}{\pi} \left[ \frac{m_1 n_1}{\sqrt{1 + m_1^2 + n_1^2}} \frac{1 + m_1^2 + 2n_1^2}{(1 + n_1^2)(m_1^2 + n_1^2)} \right] + \sin^{-1} \frac{m_1}{\sqrt{m_1^2 + n_1^2} \sqrt{1 + n_1^2}}$$

Para:

$$m_1 = \frac{L}{B}$$

$$n_1 = \frac{z}{(B/2)}$$

Consideremos ahora la situación en la que el esfuerzo es debido a una area cargada circularmente, sea  $B$  el radio de una circunferencia cargada y  $q_0$  la carga uniforme distribuida en dicha región para determinar el esfuerzo generamos una partición en los intervalos:

$$[0, B] \text{ Y } [0, 2\pi]$$

Mediante:

$$0 = r_0 < r_1 < r_2 < \dots < r_n = B \quad \text{Y} \quad 0 = \theta_0 < \theta_1 < \theta_2 < \dots < \theta_m = 2\pi$$

Consideremos ahora un área elemental del círculo, dicha área será un segmento circular y producirá una carga puntual, la cual podemos expresar como:

$$q_0 r d\theta dr$$

Reemplazando esta expresión para el diferencial de presión, tenemos:

$$dp = \frac{3(q_0 r d\theta dr)}{2\pi z^2 \left[ 1 + \left( \frac{r}{z} \right)^2 \right]^{5/2}}$$

El incremento total del esfuerzo que produce la región circular se obtiene integrando:

$$p = \int dp = \int_0^{2\pi} \int_0^B \frac{3(q_0 r d\theta dr)}{2\pi z^2 \left[ 1 + \left( \frac{r}{z} \right)^2 \right]^{5/2}}$$

Debido a que la carga  $q_0$  es uniforme tenemos entonces:

$$p = q_0 I$$

Donde:

$$I = \frac{3}{2\pi} \int_0^{2\pi} \int_0^B \frac{r d\theta dr}{z^2 \left[ 1 + \left( \frac{r}{z} \right)^2 \right]^{5/2}}$$

Calculando la integral:

$$\int_0^{2\pi} \int_0^B \frac{rd\theta dr}{z^2 \left[ 1 + \left( \frac{r}{z} \right)^2 \right]^{5/2}} = \int_0^{2\pi} \left( \int_0^B \frac{z^3 r dr}{(r^2 + z^2)^{5/2}} \right) d\theta$$

$$\int_0^{2\pi} \int_0^B \frac{rd\theta dr}{z^2 \left[ 1 + \left( \frac{r}{z} \right)^2 \right]^{5/2}} = \int_0^{2\pi} z^3 \left( -\frac{1}{3} \frac{z^3 - (B^2 + z^2)^{3/2}}{z^3 (B^2 + z^2)^{3/2}} \right) d\theta$$

$$\int_0^{2\pi} \left( \int_0^B \frac{z^3 r dr}{(r^2 + z^2)^{5/2}} \right) d\theta = \frac{1}{3} \int_0^{2\pi} \left( 1 - \frac{z^3}{(B^2 + z^2)^{3/2}} \right) d\theta$$

$$\int_0^{2\pi} \left( \int_0^B \frac{z^3 r dr}{(r^2 + z^2)^{5/2}} \right) d\theta = \frac{2\pi}{3} \left( 1 - \frac{1}{\left[ 1 + \left( \frac{B}{z} \right)^2 \right]^{3/2}} \right)$$

Con lo cual:

$$I = 1 - \frac{1}{\left[ 1 + \left( \frac{B}{z} \right)^2 \right]^{3/2}}$$

El siguiente aporte al esfuerzo vertical es una formulación del procedimiento para calcular el esfuerzo debido a una carga concentrada bajo el centro de una superficie flexible.

Consideremos a  $D$  como la región donde se aplica la carga  $q$ , la descripción en coordenadas polares para obtener el incremento del esfuerzo vertical es:

$$\Delta p = \iint_D \frac{3qrdrd\theta}{2\pi z^2 \left[ 1 + \left( \frac{r}{z} \right)^2 \right]^{5/2}}$$

La cantidad  $rdrd\theta$  representa el diferencial de área en dicha región.

Cuando se desea calcular el esfuerzo vertical de una carga concentrada sobre una región  $S$ , parametrizamos cada punto  $(x, y) \in S$ :

$$\begin{aligned}x &= f(u, v) \\y &= g(u, v)\end{aligned}$$

Para  $f, g \in C^1(W)$ , donde  $W$  es la preimagen de  $S$ , de esta forma, siendo  $j(u, v)$  el jacobiano de la mencionada transformación, la cual lleva  $W$  a  $S$  tenemos que el incremento al esfuerzo vertical está dado por:

$$\Delta p = \frac{3qz^3}{2\pi} \iint_S \frac{|j(u, v)| du dv}{[z^2 + f^2(u, v) + g^2(u, v)]^{5/2}}$$

Usando la ecuación anterior podemos calcular el esfuerzo vertical para una región cualquiera.

Veamos a continuación la aplicación de la mencionada fórmula a una región circular de radio  $B$ , para ello, formulamos la siguiente parametrización:

$$\begin{aligned}x &= Br \cos \theta \\y &= Br \sin \theta\end{aligned}$$

Donde  $r \in [0, 1]$  y  $\theta \in [0, 2\pi]$ . El jacobiano para dicha transformación es:

$$j(r, \theta) = B^2 r dr d\theta$$

Luego en la ecuación del incremento del esfuerzo vertical tenemos:

$$\Delta p = \frac{3qz^3}{2\pi} \int_0^{2\pi} \int_0^1 \frac{B^2 r dr d\theta}{[B^2 r^2 + z^2]^{5/2}}$$

Además:

$$\begin{aligned}
\int_0^{2\pi} \int_0^1 \frac{B^2 r dr d\theta}{[B^2 r^2 + z^2]^{5/2}} &= \int_0^{2\pi} B^2 \left( \int_0^1 \frac{r dr}{[B^2 r^2 + z^2]^{5/2}} \right) d\theta \\
&= \int_0^{2\pi} B^2 \left( \frac{1}{3B^2 z^3} - \frac{1}{3B^2 (B^2 + z^2)^{3/2}} \right) d\theta \\
&= \frac{2\pi}{3} \left( \frac{1}{z^3} - \frac{1}{(B^2 + z^2)^{3/2}} \right)
\end{aligned}$$

Reemplazando concientemente para calcular  $\Delta p$  se tiene que:

$$\Delta p = q \left( 1 - \frac{1}{\left( 1 + \left( \frac{r}{z} \right)^2 \right)^{3/2}} \right)$$

Lo cual concuerda con la expresión anteriormente obtenida.

Utilizando el sistema de control **JGround**, es posible realizar un estudio del asentamiento de suelos en sus dos principales partes, el asentamiento elástico y el asentamiento por consolidación.

El sistema de control **JGround**, permite la evaluación de la capacidad de carga última y la capacidad admisible, para ello, considera un factor de seguridad estandar utilizado por los ingenieros, el cual es el escalar 3. Las capacidades de carga se evalúan usando los criterios de Terzaghi, Vesic y Meyerhof, en la actualidad, en la actualidad dichos criterios son los más utilizados para el cálculo de las capacidades de carga.

Para el asentamiento elástico, el sistema de control tiene la capacidad de evaluar bajo tres criterios: Según Terzaghi, Vesic y Meyerhof.

El asentamiento por consolidación que ocurre al transcurrir el tiempo es evaluado usando el asentamiento total que gracias al procedimiento de Bousinessq para

obtener el incremento de esfuerzo vertical a una determinada profundidad en el suelo que se encuentra sometido a una carga específica.

El estudio del asentamiento de suelos en una determinada área tiene un estudio más global y más minucioso cuando dicha área se divide en zonas de estudio, el sistema de control JGround, está diseñado para desarrollar el estudio de una determinada área, considerando una división en  $n$  sectores de dicha zona, para cada una de las cuales realiza la proyección de los asentamientos elásticos y por consolidación.

El presente trabajo desarrolla la implementación de un sistema de control usando el lenguaje de programación Java y otras herramientas informáticas, de esta forma su diseño está ligado al desarrollo de software usando herramientas de software libre, estas herramientas se brindan mediante Internet a la comunidad mundial para el desarrollo aplicaciones computacionales.

La programación computacional orientada a objetos permite el diseño de aplicaciones en las cuales es posible la generalización de procedimientos como se muestra en la flexibilidad en analizar la consolidación para cualquier número de zonas en una determinada área.

El sistema de control JGround, está sustentado en la tecnología orientada a objetos que el lenguaje de programación Java proporciona y está comprobada la estabilidad de la ejecución del sistema de control.

El software libre, resulta una parte importante en el desarrollo del sistema de control, usando programación estructurada, se desarrollan los módulos que

interactúan con el núcleo, cual se encarga de la ejecución de JGround, desde el ingreso de datos hasta la emisión de los resultados.

El desarrollo de software, usando herramientas gratuitas, resulta frecuentemente en una tarea ardua debido a que dicho software de creación al ser gratuito no tiene un soporte de apoyo, eso implica que el desarrollo por parte del programador debe ser cuidadoso en cada fase de su construcción a fin de ajustarse estrictamente a las reglas impuestas por el lenguaje programación como la compatibilidad por ejemplo.

El sistema de control JGround, puede ser ejecutado en cualquier plataforma, es decir, puede correr en MS-Windows o en Linux, siempre y cuando dicha plataforma tenga instalado el **Java Runtime Environment (JRE)**, cual es una herramienta gratuita distribuida por Sun – Microsystems. Así podemos afirmar que JGround es un sistema de Control multiplataforma. Esto brinda flexibilidad en cuanto a su ejecución.

El sistema de Control JGround proporciona una visualización gráfica de la consolidación con respecto al tiempo, esto da al usuario una apreciación del proceso de consolidación en cuanto a su velocidad y proyección. Pero aparte de la visualización gráfica, el usuario puede tener a disposición los asentamientos por consolidación determinados en porcentajes, en factores de tiempo vertical y en días. Esto produce una mejor evaluación de la consolidación del suelo a través del tiempo.

## VI. CONCLUSIONES

El presente trabajo se orienta al diseño e implementación de un sistema de control, los resultados permiten ver que dicho sistema de control JGround, está enfocado a la determinación de la consolidación de suelos, con ello se tienen las siguientes conclusiones:

1. El sistema de control JGround, evalúa la capacidad de carga y la capacidad de carga admisible de una determinada zona, para las capacidades de carga, es esto son usados los criterios de Terzaghi, Vesic y Meyerhof.
2. El asentamiento elástico es calculado por el sistema de control JGround de acuerdo a los criterios de Terzaghi, Vesic y Meyerhof.
3. El asentamiento por consolidación se evalúa para el 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80% y 90%, determinándose con ello, los días que deben transcurrir, el factor de tiempo vertical y los centímetros para cada uno de los porcentajes mencionados.
4. El sistema de control JGround, elabora gráficas de días versus consolidación, para tener una visualización de la consolidación conforme transcurre el tiempo.

5. El lenguaje de programación Java, permite el diseño de un software, usando la programación orientado a objetos, a su vez que mediante Java, se construye módulos que interactúen entre sí y en coordinación con un módulo principal denominado **núcleo**, el cual dirige el proceso de cálculo para la determinación del asentamiento de suelos.
6. La herramienta de desarrollo JDK de la empresa *Sun-Microsystems* permitió el diseño y la implementación del sistema de control JGround.
7. Para la ejecución del sistema de control JGround, es necesaria la instalación del *Java Runtime Environment*, cual constituye la base para que JGround, realice los procedimientos relacionados con la consolidación.
8. El software libre es una herramienta gratuita, que permite el desarrollo de aplicaciones que permiten resolver problemas de la vida real, como es el caso de la consolidación de suelos y su proyección con respecto al tiempo.

## VII. RECOMENDACIONES

1. Realizar la simulación del proceso de consolidación en dos y tres dimensiones, valiéndose para ello de una tecnología generalizada sobre la mecánica de suelos.
2. Construir un sistema de control usando software libre, Dicho sistema, debe tener como objetivo principal el análisis de la dinámica de suelos.
3. Realizar una simulación numérica del proceso de consolidación en dos y tres dimensiones, considerando factores dinámicos y para dicha simulación hacer uso del lenguaje de programación Java.

## VIII. REFERENCIAS BIBLIOGRÁFICAS

- ALLEN, MARK (2000): "*Estructura de Datos en Java*". Editorial Addison Wesley. Madrid - España.
- BERRY, PETER L. (1993). "*Mecánica de Suelos*". Editorial McGraw-Hill, Colombia.
- BRAJA, DAS. (2001). "*Cimentaciones*", Editorial Internacional Thompson Editores. México.
- BURDEN, RICHARD y FAIRES, DOUGLAS. (2002). "*Análisis Numérico*". Editorial Grupo Ibero América. México.
- CHURCHILL, R. (1941). "*Fourier Series and Boundary Value Problems*". Editorial McGRAW-HILL, Colombia.
- FROUFE, AGUSTIN (2000). "*JAVA 2 – Manual de usuario y tutorial*", Editorial AlfaOmega – RaMa, México.
- KUMBHOJKAR, A. (1993). "*Numerical Evaluation of Terzaghi's  $N_7$* ". Editorial Journal of Geotechnical Engineering, ASCE. USA.
- SÁNCHEZ, SEBASTIÁN Y GARCÍA, ÓSCAR. (2005). "*Unix y Linux – Guía Práctica*". Editorial AlfaOmega-RaMa, Madrid.
- SANCHO. ADELA Y BOBADILLA, JESUS. (2003). "*Java – a través de ejemplos*". Editorial AlfaOmega. México.

- TERZAGHI, K y PECK, R. (1955). *“Mecánica de Suelos en la Ingeniería Práctica”*. Editorial El Ateneo. Colombia.
- ZUKOWSKI, JHON. (2003). *“Programación en JAVA 2 – J2SE 1.5”*. Anaya Multimedia, Madrid.

## **A N E X O:**

**Código Fuente del Programa**

**JGROUND**

## Sistema de Control JGround

Esta aplicación evalúa la consolidación de los suelos considerando, el asentamiento elástico y por consolidación:

**Jground:** es el núcleo del sistema de control JGround:

```
/****** JGROUND *****/  
  
import java.awt.*;  
  
import java.awt.event.*;  
  
import java.io.*;  
  
import java.util.*;  
  
import javax.swing.*;  
  
public class jground extends JFrame {  
  
    /* My Serial Version */  
  
    private static final long serialVersionUID = 1L;  
  
    JPanel panel;  
  
    ImageIcon iconNew, iconOpen, iconSave, iconSaveas, iconQuit;  
  
    ImageIcon iconTerzaghi, iconVesic, iconMeyerhof, iconHansen, iconLM;  
  
    ImageIcon iconExe, iconFactor, iconWizard, iconAuthor, iconJGround;  
  
    NewProject np; JTabbedPane tp; static Object[][] ar; FileReader fin;  
  
    BufferedReader lector = null; String titulo, autor; int numero;  
  
    String linea; StringTokenizer str;  
  
    double[][] first, second, third, fourth, fifth;  
  
    boolean flagin;  
  
    public jground() { // Constructor to Configure the Environment of the Application  
  
        np = new NewProject(); np.setVisible(false);
```

```

panel = (JPanel)this.getContentPane(); panel.setLayout(null);

panel.setBackground(Color.white); panel.add(np);

this.setTitle("JGROUND"); this.setLocation(0, 0); this.setSize(1180, 750);

this.setDefaultCloseOperation(EXIT_ON_CLOSE);

this.setResizable(false);

// Frame's icon

ImageIcon icon = new ImageIcon("icons/apple-red.png");

Image image = icon.getImage(); this.setIconImage(image);

iniicon(); inimenu(); flagin = false; }

public void iniicon(){

iconNew = new ImageIcon("icons/jnew.png");

iconOpen = new ImageIcon("icons/jopen.png");

iconSave = new ImageIcon("icons/jsave.png");

iconSaveas = new ImageIcon("icons/jsaveas.png");

iconQuit = new ImageIcon("icons/jquit.png");

iconTerzaghi = new ImageIcon("icons/jTerzaghi.png");

iconVesic = new ImageIcon("icons/jVesic.png");

iconMeyerhof = new ImageIcon("icons/jMeyerhof.png");

iconHansen = new ImageIcon("icons/jHansen.png");

iconLM = new ImageIcon("icons/jLM.png");

iconExe = new ImageIcon("icons/sun-java.png");

iconFactor = new ImageIcon("icons/jFactor.png");

iconWizard = new ImageIcon("icons/jinform.png");

iconAuthor = new ImageIcon("icons/jauthor.png");

```

```

iconJGround = new ImageIcon("icons/jhome.png"); }

public void inimenu(){

// Frame's menu

JMenuBar mb = new JMenuBar();

JMenu wFile = new JMenu("Archivo");

// Item New Project

JMenuItem iNew = new JMenuItem("Nuevo Proyecto", iconNew);

iNew.addActionListener(new aNew(this));

wFile.add(iNew);

// Item Open Project

JMenuItem iOpen = new JMenuItem("Abrir Proyecto Existente", iconOpen);

iOpen.addActionListener(new aOpen(this));

wFile.add(iOpen);

// Item Quit of the App

JMenuItem iQuit = new JMenuItem("Salir de JGROUND", iconQuit);

iQuit.addActionListener(new aEnd(this));

wFile.add(iQuit);

Menu wExecute = new JMenu("Ejecutar");

JMenuItem iExe = new JMenuItem("Ejecutar!", iconExe);

iExe.addActionListener(new aEdit(this));

wExecute.add(iExe);

JMenu xHelp = new JMenu("Ayuda");

JMenuItem xAuthor = new JMenuItem("Acerca de el Autor", iconAuthor);

xAuthor.addActionListener(new aAuthor(this));

```

```

xHelp.add(xAuthor);

xHelp.addSeparator();

// Item for the App

JMenuItem xJGround = new JMenuItem("Acerca de JGround", iconJGround);

xJGround.addActionListener(new aProgram(this));

xHelp.add(xJGround);

mb.add(wFile);

mb.add(wExecute);

//mb.add(wTools);

mb.add(xHelp);

this.setJMenuBar(mb);

}

public void mNew(ActionEvent e){

np.setVisible(true);

panel.repaint();

}

public void BuildNewPanel(ActionEvent e){

BuildEntries();

tp.setComponentAt(0, new panelIN());

panel.repaint();

}

public void BuildOpenPanel(ActionEvent e){

BuildEntries();

JFileChooser chooser = new JFileChooser(".");

```

```

int status = chooser.showOpenDialog(new FileChooser());

if(status == JFileChooser.APPROVE_OPTION){

File f = chooser.getSelectedFile();

LoadTables(f.toString());

tp.setComponentAt(0, new panelIN(this.titulo, this.autor, this.numero, this.first,
this.second, this.third, this.fourth, this.fifth));

panel.repaint(); } }

public void BuildEntries(){

int N = 3;

tp = new JTabbedPane();

tp.setBounds(10, 10, 1140, 670);

ar = new Object[N][2];

ar[0][0] = "Ingreso de Datos";

ar[0][1] = PanelLine.class;

ar[1][0] = "Salida de Datos";

ar[1][1] = PanelLine.class;

ar[2][0] = "Ayuda";

ar[2][1] = InfoPanel.class;

for(int i = 0; i < N; i++){

tp.addTab((String)ar[i][0], GeneratePanel((Class)ar[i][1]));

}

panel.add(tp); panel.repaint();

}

static JPanel GeneratePanel(Class clase) {

```

```

JPanel p = null;

try{

p = (JPanel)clase.newInstance();

} catch(Exception e){

System.out.println(e);

}

return(p);

}

public void LoadTables(String filein){

try{

fin = new FileReader(filein); lector = new BufferedReader(fin);

linea = lector.readLine(); linea = lector.readLine();

titulo = linea; linea = lector.readLine(); linea = lector.readLine();

autor = linea; linea = lector.readLine(); linea = lector.readLine();

numero = Integer.parseInt(linea);

first = new double[4][numero];

second = new double[2][numero];

third = new double[3][numero];

fourth = new double[7][numero];

fifth = new double[5][numero];

// Parametros de Resistencia

linea = lector.readLine();

for(int i = 0; i < 4; i++){

linea = lector.readLine();

```

```

str = new StringTokenizer(linea);
for(int j = 0; j < numero; j++)
first[i][j] = Double.parseDouble(str.nextToken()); }

// Propiedades Elasticas
linea = lector.readLine();
for(int i = 0; i < 2; i++){
linea = lector.readLine();
str = new StringTokenizer(linea);
for(int j = 0; j < numero; j++)
second[i][j] = Double.parseDouble(str.nextToken());
}

// Dimensiones de Zapatas
linea = lector.readLine();
for(int i = 0; i < 3; i++){
linea = lector.readLine();
str = new StringTokenizer(linea);
for(int j = 0; j < numero; j++)
third[i][j] = Double.parseDouble(str.nextToken());
}

// Estructura del Suelo
linea = lector.readLine();
for(int i = 0; i < 7; i++){
linea = lector.readLine();
str = new StringTokenizer(linea);

```

```

for(int j = 0; j < numero; j++)
fourth[i][j] = Double.parseDouble(str.nextToken());
}
// Datos de Laboratorio
linea = lector.readLine();
for(int i = 0; i < 5; i++){
linea = lector.readLine(); str = new StringTokenizer(linea);
for(int j = 0; j < numero; j++)
fifth[i][j] = Double.parseDouble(str.nextToken());
}
lector.close(); fin.close();
} catch(Exception ex){
System.out.println("Error al ingresar datos a la aplicacion");
} }
public void mOpen(ActionEvent e){
panel.repaint();
}
public void mSave(ActionEvent e){
panel.repaint();
}
public void mSaveAs(ActionEvent e){
panel.repaint();
}
public void mEnd(ActionEvent e){

```

```

System.exit(0);

}

public void mExe(ActionEvent e){

LoadTables("dat.jg");

panelOUT panelout = new panelOUT(titulo, autor, numero, first, second, third,
fourth, fifth);

WriteReport(panelout.retReport()); tp.setComponentAt(1, panelout);

this.repaint(); }

public void mAutor(ActionEvent e){

aboutAuthor win = new aboutAuthor();

win.setVisible(true); }

public void mProgram(ActionEvent e){

aboutProgram win = new aboutProgram();

win.setVisible(true); }

public void WriteReport(String str){

File freport = new File("report.doc");

FileOutputStream canalReport;

try{

String informe;

canalReport = new FileOutputStream(freport);

//////////////////// REPORTING DATA //////////////////////

String str_Title = "\t\t\t\t\tINFORME FINAL\n\t\t\t\t\tSOBRE CONSOLIDACION
DE SUELOS\n\n";

String str_Name = "\tNOMBRE DEL PROYECTO:\t" + this.titulo + "\n\n";

```

```

String str_Autor = "\tAUTOR DEL PROYECTO:\t" + this.autor + "\n\n";
String str_Numero = "\tNUMERO DE ZONAS:\t\t" + this.numero + "\n\n";
informe = str_Title + str_Name + str_Autor + str_Numero + str;
canalReport.write(informe.getBytes());
canalReport.close();
} catch (Exception ex)
{
System.out.println("Error en la escritura de datos al archivo!");
} }
public static void main(String[] args){
JFrame.setDefaultLookAndFeelDecorated(true);
jground win = new jground();
win.setVisible(true); } }

```

**NewProject.java:** Módulo para el inicio de un nuevo proyecto.

```

// Specification for a New Project
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.border.*;
public class NewProject extends JPanel {
private static final long serialVersionUID = 2L;
String nom; int sec; String aut;
JLabel tit, mens, name, sector, autor;
JTextField tname, tsector, tautor;

```

```

JButton nuevo;

NewProject(){

tit = new JLabel("JGROUND"); tit.setForeground(Color.red);

tit.setFont(new java.awt.Font("Palatino Linotype", Font.BOLD | Font.ITALIC,
50)); tit.setBounds(170, 50, 300, 60);

mens = new JLabel("Sistema de Control para Asentamiento de Suelos");

mens.setForeground(Color.blue);

mens.setFont(new java.awt.Font("Palatino Linotype", Font.BOLD | Font.ITALIC,
20));

mens.setBounds(70, 110, 450, 40); name = new JLabel("Nombre del Proyecto:");

name.setForeground(Color.black);

name.setFont(new java.awt.Font("Tahoma", Font.BOLD, 14));

name.setBounds(100, 180, 180, 30); sector = new JLabel("Numero de Sectores:");

sector.setForeground(Color.black);

sector.setFont(new java.awt.Font("Tahoma", Font.BOLD, 14));

sector.setBounds(100, 210, 180, 30);

autor = new JLabel("Autor del Proyecto:");

autor.setForeground(Color.black);

autor.setFont(new java.awt.Font("Tahoma", Font.BOLD, 14));

autor.setBounds(100, 240, 180, 30); tname = new JTextField(10);

tname.setForeground(Color.magenta);

tname.setFont(new java.awt.Font("Tahoma", Font.BOLD, 14));

tname.setBounds(280, 180, 200, 30);

tsector = new JTextField(10); tsector.setForeground(Color.magenta);

```

```

tsector.setFont(new java.awt.Font("Tahoma", Font.BOLD, 14));
tsector.setBounds(280, 210, 200, 30);
tautor = new JTextField(10); tautor.setForeground(Color.magenta);
tautor.setFont(new java.awt.Font("Tahoma", Font.BOLD, 14));
tautor.setBounds(280, 240, 200, 30);
nuevo = new JButton("CREAR NUEVO PROYECTO");
nuevo.setForeground(Color.orange);
nuevo.setFont(new java.awt.Font("Tahoma", Font.BOLD, 14));
nuevo.addActionListener(new aFinishNew(this));
nuevo.setBounds(140, 300, 300, 30);
Border br = BorderFactory.createLineBorder(Color.magenta);
this.setLayout(null);
this.setBounds(300, 100, 600, 400); this.add(tit); this.add(mens);
this.add(name); this.add(sector); this.add(autor);
this.add(tname); this.add(tsector); this.add(tautor); this.add(nuevo);
this.setBorder(br); }

public void FinishNew(ActionEvent e){
nom = tname.getText().trim();
sec = (int)Double.parseDouble(tsector.getText().trim());
aut = tautor.getText().trim(); tname.setText("");
tsector.setText(""); tautor.setText("");
this.setVisible(false); } }

```

**panelIN.java:** Módulo para el ingreso de datos:

```
// Panel to work data on the present Project
```

```

import java.io.*;

import java.awt.*;

import java.awt.event.*;

import javax.swing.*;

import javax.swing.border.*;

public class panelIN extends JPanel {

    /** My Serial Version */

    private static final long serialVersionUID = 7L;

    int npe; int ZonesN; Border br;

    ImageIcon vparResist, vpropElast, vdimZap, vestSue, vdatLab;

    JLabel et1, et2, et3, et4, et5;

    JTable tbPE;

    JTable tb1, tb2, tb3, tb4, tb5;

    JScrollPane js1, js2, js3, js4, js5;

    JLabel lb1, lb2, lb3, lb4, lb5;

    JButton inData; String nom; int sec; String aut;

    JLabel tit, mens, mens2, mens3, name, sector, autor;

    JTextField tname, tsector, tautor;

    JButton nuevo; String titulo, xautor; int numero;

    double[][] first, second, third, fourth, fifth; boolean flagin;

    panelIN(){

    flagin = false; InitPanel();

    panelone(); }

```

```

panelN(String titulo, String autor, int numero, double[][] first, double[][] second,
double[][] third, double[][] fourth, double[][] fifth){

    flagin = true;

    this.titulo = titulo; this.xautor = autor; this.numero = numero;

    this.first = first; this.second = second; this.third = third; this.fourth = fourth;
    this.fifth = fifth;

    this.ZonesN = this.numero; InitPanel(); panelone();

    this.tname.setText("" + this.titulo); this.tautor.setText("" + this.xautor);
    this.tsector.setText("" + this.numero); this.nom = this.titulo;

    this.aut = this.xautor; paneltwo(); this.repaint(); }

    public void InitPanel(){

    br = BorderFactory.createTitledBorder("SECTOR");

    this.setLayout(null); this.setBackground(Color.white); this.setBorder(br);

    this.setBounds(15, 25, 1140, 630); }

    public void panelone(){

    tit = new JLabel("JGROUND");

    tit.setForeground(Color.red);

    tit.setFont(new java.awt.Font("Palatino Linotype", Font.BOLD | Font.ITALIC,
    50)); tit.setBounds(400, 50, 300, 60);

    mens = new JLabel("Sistema de Control");

    mens.setForeground(Color.blue);

    mens.setFont(new java.awt.Font("Palatino Linotype", Font.BOLD | Font.ITALIC,
    20)); mens.setBounds(450, 20, 450, 40); mens2 = new JLabel("para");

    mens2.setForeground(Color.blue);

```

```

mens2.setFont(new java.awt.Font("Palatino Linotype", Font.BOLD
Font.ITALIC, 20));

mens2.setBounds(500, 90, 450, 40);

mens3 = new JLabel("Asentamiento de Suelos");

mens3.setForeground(Color.blue);

mens3.setFont(new java.awt.Font("Palatino Linotype", Font.BOLD
Font.ITALIC, 20));

mens3.setBounds(430, 120, 450, 40);

name = new JLabel("Nombre del Proyecto");

name.setForeground(Color.black);

name.setFont(new java.awt.Font("Tahoma", Font.BOLD, 14));

name.setBounds(350, 160, 160, 30);

autor = new JLabel("Autor del Proyecto");

autor.setForeground(Color.black);

autor.setFont(new java.awt.Font("Tahoma", Font.BOLD, 14));

autor.setBounds(540, 160, 160, 30);

sector = new JLabel("Numero de Zonas:");

sector.setForeground(Color.black);

sector.setFont(new java.awt.Font("Tahoma", Font.BOLD, 14));

sector.setBounds(350, 220, 160, 30);

tname = new JTextField(10);

tname.setForeground(Color.magenta);

tname.setFont(new java.awt.Font("Tahoma", Font.BOLD, 14));

tname.setBounds(350, 190, 160, 30);

```

```

tautor = new JTextField(10);

tautor.setForeground(Color.magenta);

tautor.setFont(new java.awt.Font("Tahoma", Font.BOLD, 14));

tautor.setBounds(540, 190, 160, 30);

tsector = new JTextField(4);

tsector.setForeground(Color.magenta);

tsector.setFont(new java.awt.Font("Tahoma", Font.BOLD, 14));

tsector.setBounds(480, 220, 30, 30);

nuevo = new JButton("GENERAR");

nuevo.setForeground(Color.green);

nuevo.setFont(new java.awt.Font("Tahoma", Font.BOLD, 14));

nuevo.addActionListener(new aCreate(this));

nuevo.setBounds(540, 220, 160, 30);

this.add(tit); this.add(mens); this.add(mens2); this.add(mens3);

this.add(name); this.add(sector); this.add(autor); this.add(tname);

this.add(tsector); this.add(tautor); this.add(nuevo); }

public void paneltwo() {

iniicon(); CreateLabel(); buildTables(); }

public void CreateNew(ActionEvent e){

this.nom = tname.getText().trim();

this.ZonesN = (int)Double.parseDouble(tsector.getText().trim());

this.aut = tautor.getText().trim();

paneltwo(); }

public void CreateLabel(){

```

```

et1 = new JLabel("Parametros de Resistencia");
et1.setForeground(Color.black);
et1.setFont(new java.awt.Font("Palatino Linotype", Font.BOLD | Font.ITALIC,
20));
et1.setBounds(100, 270, 260, 30);
this.add(et1); et2 = new JLabel("Propiedades Elasticas");
et2.setForeground(Color.black);
et2.setFont(new java.awt.Font("Palatino Linotype", Font.BOLD | Font.ITALIC,
20)); et2.setBounds(100, 450, 220, 30);
this.add(et2); et3 = new JLabel("Dimensiones de Zapatas");
et3.setForeground(Color.black);
et3.setFont(new java.awt.Font("Palatino Linotype", Font.BOLD | Font.ITALIC,
20)); et3.setBounds(730, 260, 220, 30);
this.add(et3); et4 = new JLabel("Estructura de Suelo");
et4.setForeground(Color.black);
et4.setFont(new java.awt.Font("Palatino Linotype", Font.BOLD | Font.ITALIC,
20));
et4.setBounds(420, 340, 220, 30); this.add(et4);
et5 = new JLabel("Datos de Laboratorio");
et5.setForeground(Color.black);
et5.setFont(new java.awt.Font("Palatino Linotype", Font.BOLD | Font.ITALIC,
20));
et5.setBounds(730, 450, 300, 30); this.add(et5);
inData = new JButton("INGRESAR DATOS");

```

```

inData.setForeground(Color.red);

inData.addActionListener(new aINDATA(this));

inData.setBounds(460, 550, 200, 30); this.add(inData); this.repaint();
}

public void iniicon(){

vparResist = new ImageIcon("icons/parResist.png");
vpropElast = new ImageIcon("icons/propElast.png");
vdimZap = new ImageIcon("icons/dimZap.png");
vestSue = new ImageIcon("icons/estSue.png");
vdatLab = new ImageIcon("icons/datLab.png");

lb1 = new JLabel(vparResist); lb1.setBounds(100, 300, 100, 160);
lb2 = new JLabel(vpropElast); lb2.setBounds(100, 460, 100, 160);
lb3 = new JLabel(vdimZap); lb3.setBounds(730, 290, 100, 160);
lb4 = new JLabel(vestSue); lb4.setBounds(420, 370, 100, 160);
lb5 = new JLabel(vdatLab); lb5.setBounds(715, 480, 120, 130); }

public void buildTables(){

int N = this.ZonesN; String[] tit = new String[N]; String[][] dat;

// First table parResist

dat = new String[4][N];

if(flagin){

for(int i = 0; i < 4; i++)

for(int j = 0; j < N; j++)

dat[i][j] = "" + first[i][j];

}

```

```

for(int i = 0; i < N; i++){
tit[i] = "ZONA: " + (i + 1);
}
tb1 = new JTable(dat, tit); tb1.setColumnSelectionAllowed(true);
tb1.setAutoResizeMode(JTable.AUTO_RESIZE_OFF);
js1 = new JScrollPane(tb1); js1.setBounds(200, 320, 200, 120);
this.add(js1); this.add(lb1);

// Second table propElast
dat = new String[2][N];
if(flagin){
for(int i = 0; i < 2; i++)
for(int j = 0; j < N; j++)
dat[i][j] = "" + second[i][j];
}
for(int i = 0; i < N; i++){
tit[i] = "ZONA: " + (i + 1);
}
tb2 = new JTable(dat, tit);
tb2.setColumnSelectionAllowed(true);
tb2.setAutoResizeMode(JTable.AUTO_RESIZE_OFF);
js2 = new JScrollPane(tb2); js2.setBounds(200, 500, 160, 110);
this.add(js2); this.add(lb2);

// Third table dimZap
dat = new String[3][N];

```

```

if(flagin){
for(int i = 0; i < 3; i++)
for(int j = 0; j < N; j++)
dat[i][j] = "" + third[i][j];
}
for(int i = 0; i < N; i++){
tit[i] = "ZONA: " + (i + 1);
}
tb3 = new JTable(dat, tit);
tb3.setColumnSelectionAllowed(true);
tb3.setAutoResizeMode(JTable.AUTO_RESIZE_OFF);
js3 = new JScrollPane(tb3);
js3.setBounds(830, 320, 200, 100); this.add(js3); this.add(lb3);
// Fourth table estSue
dat = new String[7][N];
if(flagin){
for(int i = 0; i < 7; i++)
for(int j = 0; j < N; j++)
dat[i][j] = "" + fourth[i][j];
}
for(int i = 0; i < N; i++){
tit[i] = "ZONA: " + (i + 1);
}
tb4 = new JTable(dat, tit); tb4.setColumnSelectionAllowed(true);

```

```

tb4.setAutoResizeMode(JTable.AUTO_RESIZE_OFF);

js4 = new JScrollPane(tb4);

js4.setBounds(520, 370, 200, 160); this.add(js4); this.add(lb4);

// Fifth table datLab

dat = new String[5][N];

if(flagin){

for(int i = 0; i < 5; i++)

for(int j = 0; j < N; j++)

dat[i][j] = "" + fifth[i][j]; }

for(int i = 0; i < N; i++){

tit[i] = "ZONA: " + (i + 1); }

tb5 = new JTable(dat, tit);

tb5.setColumnSelectionAllowed(true);

tb5.setAutoResizeMode(JTable.AUTO_RESIZE_OFF);

js5 = new JScrollPane(tb5); js5.setBounds(830, 490, 200, 130);

this.add(js5); this.add(lb5); }

public void readDATA(ActionEvent e){

int N = this.ZonesN;

double[][] datos01 = new double[4][N];

double[][] datos02 = new double[2][N];

double[][] datos03 = new double[3][N];

double[][] datos04 = new double[7][N];

double[][] datos05 = new double[5][N];

File fin = new File("dat.jg");

```

```

FileOutputStream canalOUT;

try{

for(int i = 0; i < 4; i++)

for(int j = 0; j < N; j++)

datos01[i][j] = Double.parseDouble(tb1.getValueAt(i, j).toString());

for(int i = 0; i < 2; i++)

for(int j = 0; j < N; j++)

datos02[i][j] = Double.parseDouble(tb2.getValueAt(i, j).toString());

for(int i = 0; i < 3; i++)

for(int j = 0; j < N; j++)

datos03[i][j] = Double.parseDouble(tb3.getValueAt(i, j).toString());

for(int i = 0; i < 7; i++)

for(int j = 0; j < N; j++)

datos04[i][j] = Double.parseDouble(tb4.getValueAt(i, j).toString());

for(int i = 0; i < 5; i++)

for(int j = 0; j < N; j++)

datos05[i][j] = Double.parseDouble(tb5.getValueAt(i, j).toString());

canalOUT = new FileOutputStream(fin);

canalOUT.write("%Nombre del Proyecto:\n".getBytes());

nom = this.tname.getText();

canalOUT.write(nom.getBytes()); canalOUT.write("\n".getBytes());

canalOUT.write("%Autor del Proyecto:\n".getBytes());

aut = this.tautor.getText();

canalOUT.write(aut.getBytes());

```

```

canalOUT.write("\n".getBytes());

canalOUT.write("%Numero de Zonas:\n".getBytes());

canalOUT.write((N + "").getBytes());

canalOUT.write("\n".getBytes());

// Cuadro 01

canalOUT.write("%Parametros de Resistencia:\n".getBytes());

for(int i = 0; i < 4; i++){

for(int j = 0; j < N; j++)

canalOUT.write((datos01[i][j] + "\t").getBytes());

canalOUT.write("\n".getBytes());

}

// Cuadro 02

canalOUT.write("%Propiedades Elasticas:\n".getBytes());

for(int i = 0; i < 2; i++){

for(int j = 0; j < N; j++)

canalOUT.write((datos02[i][j] + "\t").getBytes());

canalOUT.write("\n".getBytes());

}

// Cuadro 03

canalOUT.write("%Dimensiones de Zapatas:\n".getBytes());

for(int i = 0; i < 3; i++){

for(int j = 0; j < N; j++)

canalOUT.write((datos03[i][j] + "\t").getBytes());

canalOUT.write("\n".getBytes());

```

```

}
// Cuadro 04
canalOUT.write("%Estructura del Suelo:\n".getBytes());
for(int i = 0; i < 7; i++){
for(int j = 0; j < N; j++)
canalOUT.write((datos04[i][j] + "\t").getBytes());
canalOUT.write("\n".getBytes());
}

```

```

// Cuadro 05
canalOUT.write("%Datos de Laboratorio:\n".getBytes());
for(int i = 0; i < 5; i++){
for(int j = 0; j < N; j++)
canalOUT.write((datos05[i][j] + "\t").getBytes());
canalOUT.write("\n".getBytes()); }
canalOUT.close();
} catch(Exception ex){
System.out.println("Error en la escritura de datos al archivo!");} } }

```

**panelOUT:** módulo para la salida de datos:

```

// Panel to work data on the present Project
import java.awt.*;
import javax.swing.*;
public class panelOUT extends JPanel {
/** My Serial Version */
private static final long serialVersionUID = 7L;

```

```

JTabbedPane tpout;

Object[][] ar; int nzonas; String Titulo, Autor; double[][] zonedata;

NewPanel xpanel; String xreport; int count;

panelOUT(String Titulo, String Autor, int nzonas, double[][] t1f, double[][] t2f,
double[][] t3f, double[][] t4f, double[][] t5f){

this.nzonas = nzonas; this.Titulo = Titulo; this.Autor = Autor;

this.zonedata = new double[this.nzonas][21];

for(int i = 0; i < this.nzonas; i++){

this.zonedata[i][0] = t1f[0][i]; this.zonedata[i][1] = t1f[1][i]; this.zonedata[i][2] =
t1f[2][i]; this.zonedata[i][3] = t1f[3][i];

this.zonedata[i][4] = t2f[0][i]; this.zonedata[i][5] = t2f[1][i];

this.zonedata[i][6] = t3f[0][i]; this.zonedata[i][7] = t3f[1][i]; this.zonedata[i][8] =
t3f[2][i];

this.zonedata[i][9] = t4f[0][i]; this.zonedata[i][10] = t4f[1][i]; this.zonedata[i][11]
= t4f[2][i]; this.zonedata[i][12] = t4f[3][i];

this.zonedata[i][13] = t4f[4][i]; this.zonedata[i][14] = t4f[5][i];

this.zonedata[i][15] = t4f[6][i];

this.zonedata[i][16] = t5f[0][i]; this.zonedata[i][17] = t5f[1][i];

this.zonedata[i][18] = t5f[2][i]; this.zonedata[i][19] = t5f[3][i];

this.zonedata[i][20] = t5f[4][i]; }

xreport = "";

count = 1; BuildResult();

this.setLayout(null); this.setBackground(Color.white);

this.setBounds(15, 25, 1140, 630);}

```

```

public NewPanel GetPanel(int k){
NewPanel pan; double[] par = new double[4];
double[] prop = new double[2]; double[] dim = new double[3];
double[] est = new double[7]; double[] dat = new double[5];
for(int i = 0; i < 4; i++) par[i] = this.zonedata[k][i];
for(int i = 0; i < 2; i++) prop[i] = this.zonedata[k][i + 4];
for(int i = 0; i < 3; i++) dim[i] = this.zonedata[k][i + 6];
for(int i = 0; i < 7; i++) est[i] = this.zonedata[k][i + 9];
for(int i = 0; i < 5; i++) dat[i] = this.zonedata[k][i + 16];
pan = new NewPanel(this.Titulo, this.Autor, par, prop, dim, est, dat);
xreport = xreport + "\n\t\t\t\tZONA DE ESTUDIO: " + (count) +
"\n\t\t\t\t*****\n" + pan.getInform();
count = count + 1;
return pan;}

public String retReport(){
return this.xreport;}

public void BuildResult(){
int N = this.nzonas; tpout = new JTabbedPane();
tpout.setBounds(10, 10, 1100, 650); ar = new Object[N][2];
for(int i = 0; i < N; i++){ ar[i][0] = "ZONA: " + (i + 1);
ar[i][1] = PanelLine.class; }
for(int i = 0; i < N; i++){
tpout.addTab((String)ar[i][0], GeneratePanel((Class)ar[i][1]));
tpout.setComponentAt(i, GetPanel(i));

```

```

tpout.repaint(); }

this.add(tpout); this.repaint(); }

static JPanel GeneratePanel(Class clase) {

JPanel p = null;

try{ p = (JPanel)clase.newInstance(); }catch(Exception e){

System.out.println(e); }

return(p); } }

```

**PanelLine.java:** Módulo temporal para la creacion de objetos dirigido a las pestañas:

```

// Una Linea para el Panel

import java.awt.*;

import javax.swing.JPanel;

public class PanelLine extends JPanel {

/** * My Serial Version */

private static final long serialVersionUID = 4L;

PanelLine(){

this.setLayout(null); this.setBackground(Color.white);

this.setBounds(15, 25, 1140, 630); } }

```

**InfoPanel.java:** Módulo para la presentacion de la ayuda del sistema de control:

```

// Panel para Informacion sobre el Sistema de Control

import java.awt.Color;

import java.awt.Font;

import javax.swing.*;

```

```

import javax.swing.border.Border;

public class InfoPanel extends JPanel {

/** My Serial Version */

private static final long serialVersionUID = 9L;

Border br; JLabel tit, mens, mens2, mens3;

JLabel l01, l02, l03, l04, l05, l06, l07, l08, l09, l10, l11, l12, l13, l14, l15, l16, l17,
l18, l19, l20;

ImageIcon im1, im2, im3, im4, im5, im6, im7, im8, im9, im10, img;

JLabel le1, le2, le3, le4, le5, le6, le7, le8, le9, le10, leg;

InfoPanel(){

br = BorderFactory.createTitledBorder("MECANISMO DEL SISTEMA DE
CONTROL JGROUND");

tit = new JLabel("JGROUND"); tit.setForeground(Color.red);

tit.setFont(new java.awt.Font("Palatino Linotype", Font.BOLD | Font.ITALIC,
50)); tit.setBounds(90, 30, 300, 60);

mens = new JLabel("Sistema de Control"); mens.setForeground(Color.blue);

mens.setFont(new java.awt.Font("Palatino Linotype", Font.BOLD | Font.ITALIC,
20)); mens.setBounds(370, 20, 450, 40);

mens2 = new JLabel("para"); mens2.setForeground(Color.blue);

mens2.setFont(new java.awt.Font("Palatino Linotype", Font.BOLD |
Font.ITALIC, 20)); mens2.setBounds(430, 40, 450, 40);

mens3 = new JLabel("Asentamiento de Suelos");

mens3.setForeground(Color.blue);

```

```

mens3.setFont(new java.awt.Font("Palatino Linotype", Font.BOLD |
Font.ITALIC, 20)); mens3.setBounds(360, 60, 450, 40);

this.setLayout(null); this.setBackground(Color.white); this.setBorder(br);

this.setBounds(15, 25, 1140, 630); this.add(tit); this.add(mens);

this.add(mens2); this.add(mens3); setL(); setLE(); }

public void setL(){

l01 = new JLabel("JGROUND Sistema de Control:");

l01.setForeground(Color.black);

l01.setFont(new java.awt.Font("Palatino Linotype", Font.BOLD | Font.ITALIC,
18)); l01.setBounds(50, 90, 450, 20);

l02 = new JLabel("Considera:"); l02.setForeground(Color.black);

l02.setFont(new java.awt.Font("Palatino Linotype", Font.BOLD | Font.ITALIC,
18));

l02.setBounds(50, 110, 450, 20); l03 = new JLabel(" Asentamiento Elastico.");

l03.setForeground(Color.black);

l03.setFont(new java.awt.Font("Palatino Linotype", Font.BOLD | Font.ITALIC,
18)); l03.setBounds(90, 130, 450, 20);

l04 = new JLabel(" Asentamiento por Consolidacion.");

l04.setForeground(Color.black);

l04.setFont(new java.awt.Font("Palatino Linotype", Font.BOLD | Font.ITALIC,
18)); l04.setBounds(90, 150, 450, 20);

l05 = new JLabel("El Asentamiento Elastico se calcula mediante:");

l05.setForeground(Color.black);

```

```

105.setFont(new java.awt.Font("Palatino Linotype", Font.BOLD | Font.ITALIC,
18)); 105.setBounds(50, 170, 450, 20);

106 = new JLabel("Para q0 capacidad de carga admisible segun");
106.setForeground(Color.black);

106.setFont(new java.awt.Font("Palatino Linotype", Font.BOLD | Font.ITALIC,
18)); 106.setBounds(50, 250, 450, 20);

107 = new JLabel("Terzaghi, Vesic y Meyerhof.");
107.setForeground(Color.black);

107.setFont(new java.awt.Font("Palatino Linotype", Font.BOLD | Font.ITALIC,
18));

107.setBounds(50, 270, 450, 20);

108 = new JLabel("El Asentamiento por Consolidacion se calcula por:");
108.setForeground(Color.black);

108.setFont(new java.awt.Font("Palatino Linotype", Font.BOLD | Font.ITALIC,
18));

108.setBounds(50, 290, 450, 20);

109 = new JLabel("Para:"); 109.setForeground(Color.black);

109.setFont(new java.awt.Font("Palatino Linotype", Font.BOLD | Font.ITALIC,
18));

109.setBounds(50, 370, 450, 20);

110 = new JLabel("Coeficiente de Compresion"); 110.setForeground(Color.black);

110.setFont(new java.awt.Font("Palatino Linotype", Font.BOLD | Font.ITALIC,
18));

110.setBounds(130, 370, 450, 20);

```

```
111 = new JLabel("Estrato Compresible"); 111.setForeground(Color.black);
111.setFont(new java.awt.Font("Palatino Linotype", Font.BOLD | Font.ITALIC,
18));
111.setBounds(130, 400, 450, 20);
112 = new JLabel("Relacion de Vacios Inicial"); 112.setForeground(Color.black);
112.setFont(new java.awt.Font("Palatino Linotype", Font.BOLD | Font.ITALIC,
18));
112.setBounds(130, 430, 450, 20);
113 = new JLabel("Presion Inicial"); 113.setForeground(Color.black);
113.setFont(new java.awt.Font("Palatino Linotype", Font.BOLD | Font.ITALIC,
18));
113.setBounds(130, 460, 450, 20);
115 = new JLabel("El Factor de Tiempo Adimensional es:");
115.setForeground(Color.black);
115.setFont(new java.awt.Font("Palatino Linotype", Font.BOLD | Font.ITALIC,
18));
115.setBounds(50, 525, 450, 20);
117 = new JLabel("El Grado Promedio de Consolidacion es:");
117.setForeground(Color.black);
117.setFont(new java.awt.Font("Palatino Linotype", Font.BOLD | Font.ITALIC,
18));
117.setBounds(590, 430, 450, 20);
118 = new JLabel("El Factor de Tiempo respecto al Tiempo:");
118.setForeground(Color.black);
```

```

118.setFont(new java.awt.Font("Palatino Linotype", Font.BOLD | Font.ITALIC,
18));
118.setBounds(590, 520, 450, 20);
119 = new JLabel("JGround calcula el asentamiento respecto al tiempo en dias");
119.setForeground(Color.black);
119.setFont(new java.awt.Font("Palatino Linotype", Font.BOLD | Font.ITALIC,
18));
119.setBounds(590, 590, 500, 20);
120 = new JLabel("para el 10%, 20%, ..., 90% grado de consolidacion");
120.setForeground(Color.black);
120.setFont(new java.awt.Font("Palatino Linotype", Font.BOLD | Font.ITALIC,
18));
120.setBounds(590, 610, 500, 20); this.add(l01); this.add(l02);
this.add(l03); this.add(l04); this.add(l05); this.add(l06);
this.add(l07); this.add(l08); this.add(l09); this.add(l10);
this.add(l11); this.add(l12); this.add(l13); this.add(l15);
this.add(l17); this.add(l18); this.add(l19); this.add(l20); }
public void setLE(){
this.im1 = new ImageIcon("icons/le1.png");
this.im2 = new ImageIcon("icons/le2.png");
this.im3 = new ImageIcon("icons/le3.png");
this.im4 = new ImageIcon("icons/le4.png");
this.im5 = new ImageIcon("icons/le5.png");
this.im5 = new ImageIcon("icons/le5.png");

```

```

this.im6 = new ImageIcon("icons/le6.png");
this.im7 = new ImageIcon("icons/le7.png");
this.im8 = new ImageIcon("icons/le8.png");
this.im9 = new ImageIcon("icons/le9.png");
this.im10 = new ImageIcon("icons/le10.png");
this.img = new ImageIcon("icons/leg.png"); this.le1 = new JLabel(im1);
this.le1.setBounds(90, 200, 200, 50); this.le2 = new JLabel(im2);
this.le2.setBounds(90, 310, 240, 50); this.le3 = new JLabel(im3);
this.le3.setBounds(100, 370, 30, 30); this.le4 = new JLabel(im4);
this.le4.setBounds(100, 400, 30, 30); this.le5 = new JLabel(im5);
this.le5.setBounds(100, 430, 30, 30); this.le6 = new JLabel(im6);
this.le6.setBounds(100, 460, 30, 30); this.le7 = new JLabel(im7);
this.le7.setBounds(100, 480, 240, 50); this.le8 = new JLabel(im8);
this.le8.setBounds(100, 550, 240, 90); this.le9 = new JLabel(im9);
this.le9.setBounds(590, 450, 320, 70); this.le10 = new JLabel(im10);
this.le10.setBounds(590, 540, 200, 50); this.leg = new JLabel(img);
this.leg.setBounds(620, 20, 400, 400); this.add(le1); this.add(le2); this.add(le3);
this.add(le4); this.add(le5); this.add(le6); this.add(le7);
this.add(le8); this.add(le9); this.add(le10); this.add(leg); } }

```

**Qpanel.java:** Módulo para la elaboración de las gráficas del sistema de control:

```

// Asentamiento por Tiempo
import java.awt.*;
import javax.swing.*;
import javax.swing.border.*;

```

```

public class QPanel extends JPanel {
    /** My Serial Version */

    private static final long serialVersionUID = 7L;

    int incr; int iniU; Border bored;

    double[] dias; double[] ass; int[] diasint; double[] assint;

    int daymin, daymax; double assmin, assmax;

    QPanel(double[] dias, double[] ass){
        this.dias = dias; this.ass = ass;

        this.diasint = new int[9]; this.assint = new double[9]; dayass();

        this.setLayout(null); this.setBounds(680, 100, 400, 450);

        bored = BorderFactory.createLineBorder(Color.black, 2);

        this.setBorder(bored); }

    public void dayass(){
        for(int i = 0; i < 9; i++){

            diasint[i] = (int)Math.round(dias[i]);

            assint[i] = ((int)Math.round(100 * ass[i])) / 10.0; }

        this.daymin = diasint[0]; this.daymax = diasint[0];

        this.assmin = assint[0]; this.assmax = assint[0];

        for(int i = 1; i < 9; i++){

            if(this.daymin > diasint[i]) this.daymin = diasint[i];

            if(this.daymax < diasint[i]) this.daymax = diasint[i];

            if(this.assmin > assint[i]) this.assmin = assint[i];

            if(this.assmax < assint[i]) this.assmax = assint[i];

        }}

```

```

public void paint(Graphics g){
g.setColor(Color.black); g.drawRect(0, 0, 399, 399);
g.setColor(Color.green); g.drawRect(30, 20, 350, 350);
double width, height; int[] xp = new int[9]; int[] yp = new int[9];
width = 350.0 / (this.daymax - this.daymin); height = 350.0 / (this.assmax -
this.assmin);
for(int i = 0; i < 9; i++){
xp[i] = 30 + (int)((this.diasint[i] - this.daymin) * width);
yp[i] = 370 - (int)((this.assint[i] - this.assmin) * height);
}
g.setColor(Color.red); g.fillOval(xp[0], yp[0], 5, 5); g.setColor(Color.black);
g.drawString("" + this.diasint[0], xp[0] + 2, yp[0] + 12);
g.drawString("" + this.assint[0], 5, yp[0]);
for(int i = 1; i < 9; i++){
g.setColor(Color.green); g.drawLine(xp[i], 20, xp[i], 370);
g.drawLine(30, yp[i], 380, yp[i]); g.setColor(Color.red);
g.fillOval(xp[i], yp[i], 5, 5); g.setColor(Color.black);
g.drawString("" + this.diasint[i], xp[i] + 2, yp[i] + 12);
g.drawString("" + this.assint[i], 5, yp[i]);
g.setColor(Color.blue);
g.drawLine(xp[i - 1], yp[i - 1], xp[i], yp[i]);
}
g.setColor(Color.black); g.drawString("D I A S", 170, 385);
g.drawString("(mm)", 35, 13);

```

```

g.setFont(new java.awt.Font("Palatino Linotype", Font.BOLD | Font.ITALIC,
20));

g.setColor(Color.red);

g.drawString("Dias versus Consolidacion(en mm)", 40, 430);

}

public void show()

{

System.out.println("Valores:");

for(int i = 0; i < 9; i++)

System.out.println("Dia: " + diasint[i] + "\tSc: " + assint[i]);

}}

```

**Jwork.java:** Módulo para realizar los cálculos de las capacidades de carga, asentamientos elásticos y por consolidación:

```

/* A Class to Develop all of the Computations of the Control System */

public class jwork {

int an; double phi, c, g1, g2; // From the First table

double Es, mus; // From the Second table

double q, H1, H2, Hc, g1C, g2C, gcC; // From the Third table

double Df, B, L; // From the Fourth table

double Cc, Cr, e0, pc, Cv;

jwork(double[] t1, double[] t2, double[] t3, double[] t4, double[] t5) {

this.an = (int)t1[0];

this.phi = t1[0] * Math.PI / 180; this.c = t1[1] / 1000.0; this.g1 = t1[2]; this.g2 =

t1[3];

```

```

this.Es = t2[0]; this.mus = t2[1];

this.q = t3[0]; this.H1 = t3[1]; this.H2 = t3[2]; this.Hc = t3[3];

this.g1C = t3[4]; this.g2C = t3[5]; this.gcC = t3[6];

this.Df = t4[0]; this.B = t4[1]; this.L = t4[2];

this.Cc = t5[0]; this.Cr = t5[1]; this.e0 = t5[2]; this.pc = t5[3] / 1000.0; this.Cv =
t5[4];
}

public double[] fTe(){

double[] fT = new double[3];

double anr = this.phi; double[] TT = new double[51]; double Nq, Nc, Ng;

TT[0] = 0.00; TT[1] = 0.01; TT[2] = 0.04; TT[3] = 0.06; TT[4] = 0.10; TT[5] =
0.14; TT[6] = 0.20; TT[7] =0.27;

TT[8] = 0.35; TT[9] = 0.44; TT[10]= 0.56; TT[11]= 0.69; TT[12]= 0.85; TT[13]=
1.04; TT[14]= 1.26; TT[15]=1.52;

TT[16]= 1.82; TT[17]= 2.18; TT[18]= 2.59; TT[19]= 3.07; TT[20]= 3.64;

TT[21]= 4.31; TT[22]= 5.09; TT[23]=6.00;

TT[24]= 7.08; TT[25]= 8.34; TT[26]= 9.84; TT[27]= 11.60;TT[28]=
13.70;TT[29]= 16.18;TT[30]= 19.13;TT[31]=22.65;

TT[32]=26.87; TT[33]=31.94; TT[34]=38.04; TT[35]=45.41; TT[36]=
54.36;TT[37]= 65.27;TT[38]= 78.61;TT[39]=95.03;

TT[40]=115.31;TT[41]=140.51;TT[42]=171.99;TT[43]=211.56;TT[44]=261.60;

TT[45]=325.34;TT[46]=407.11;TT[47]=512.84;

TT[48]=650.67;TT[49]=831;TT[50]=1072.80;

Nq = Math.pow(Math.E, 2*(3*Math.PI/4.0 - anr / 2.0)*Math.tan(anr));

```

```

double Nqd = (2*Math.pow(Math.cos(Math.PI / 4 + anr / 2.0), 2));
Nq = Nq / Nqd; Nc = (Nq - 1) / Math.tan(anr + 0.000001); Ng = TT[this.anr];
fT[0] = Nc; fT[1] = Nq; fT[2] = Ng;
return fT;}

public double[] fVe(){
double[] fV = new double[3];
double Nq, Nc, Ng; double anr = this.phi;
Nq = Math.pow(Math.E, Math.PI * Math.tan(anr)) *
Math.pow(Math.tan(Math.PI/4+anr/2), 2);
Nc = (Nq - 1) / Math.tan(anr + 0.000001);
Ng = 2 * (Nq + 1) * Math.tan(anr); fV[0] = Nc; fV[1] = Nq; fV[2] = Ng;
return fV;
}

public double[] fMe(){
double[] fM = new double[3];
double Nq, Nc, Ng; double anr = this.phi;
Nq=Math.pow(Math.E,Math.PI*Math.tan(anr))*Math.pow(Math.tan(Math.PI/4+a
nr/2), 2);
Nc = (Nq - 1) / Math.tan(anr + 0.000001); Ng = (Nq - 1) * Math.tan(1.4 * anr);
fM[0] = Nc; fM[1] = Nq; fM[2] = Ng;
return fM;
}

public double[][] qTe(){
double[][] qT = new double[2][2]; double[] N = fTe();

```

```

double qu1 = this.c * N[0] + this.g2 * this.Df * N[1] / 1000 + 0.5 * this.g1 *
this.B * N[2] / 1000;

double qa1 = qu1 / 3.0;

double qu2 = 1.3 * this.c * N[0] + this.g2 * this.Df * N[1] / 1000 + 0.4 * this.g1 *
this.B * N[2] / 1000; double qa2 = qu1 / 3.0;

qT[0][0] = Math.round(100 * qu1) / 100.0; qT[1][0] = Math.round(100 * qa1) /
100.0;

qT[0][1] = Math.round(100 * qu2) / 100.0; qT[1][1] = Math.round(100 * qa2) /
100.0;

return qT;
}

public double[][] qVe(){
double[][] qV = new double[2][2]; double[] N = fVe();

double Fcs = 1 + (B / (1.0 * L)) * (N[1] / N[0]);

double Fqs = 1 + (B / (1.0 * L)) * Math.tan(this.phi); double Fgs = 1 - 0.4 * (B /
(1.0 * L));

double qu1 = this.c * N[0] + this.g2 * this.Df * N[1] / 1000 + 0.5 * this.g1 *
this.B * N[2] / 1000;

double qa1 = qu1 / 3.0;

double      qu2      =      this.c      *      N[0]*Fcs      +
this.g2*this.Df*N[1]*Fqs/1000+0.5*this.g1*this.B*N[2] * Fgs / 1000;

double qa2 = qu1 / 3.0;

qV[0][0] = Math.round(100 * qu1) / 100.0; qV[1][0] = Math.round(100 * qa1) /
100.0;

```

```

qV[0][1] = Math.round(100 * qu2) / 100.0; qV[1][1] = Math.round(100 * qa2) /
100.0;

return qV;

}

public double[][] qMe(){

double[][] qM = new double[2][2]; double[] N = fMe();

double Fcs = 1 + (B / (1.0 * L)) * 0.2 * Math.pow(Math.tan(Math.PI/4+this.phi /
2), 2);

double Fqs = 1 + (B / (1.0 * L)) * 0.1 * Math.pow(Math.tan(Math.PI/4+this.phi /
2), 2);

double Fgs = Fqs;

double qu1 = this.c * N[0] + this.g2 * this.Df * N[1] / 1000 + 0.5 * this.g1 *
this.B * N[2] / 1000;

double qa1 = qu1 / 3.0;

double      qu2      =      this.c      *      N[0]*Fcs      +
this.g2*this.Df*N[1]*Fqs/1000+0.5*this.g1*this.B*N[2] * Fgs / 1000;

double qa2 = qu1 / 3.0;

qM[0][0] = Math.round(100 * qu1) / 100.0; qM[1][0] = Math.round(100 * qa1) /
100.0;

qM[0][1] = Math.round(100 * qu2) / 100.0; qM[1][1] = Math.round(100 * qa2) /
100.0;

return qM;

}

public double alpha(){

```

```

double alp;

double m = this.L / (this.B * 1.0);

double ins = Math.sqrt(1 + Math.pow(m, 2));

double p1 = (ins + m) / (ins - m); double p2 = (ins + 1.0) / (ins - 1.0);

alp = (Math.log(p1) + m * Math.log(p2)) / Math.PI;

return alp;
}

public double[] SeTe(){

double[] SeT = new double[2]; double[][] qT = qTe();

SeT[0] = (this.B * qT[1][0]) / (this.Es) * (1 - Math.pow(this.mus, 2)) * alpha();

SeT[0] = Math.round(100 * SeT[0]) / 100.0;

SeT[1] = (this.B * qT[1][1]) / (this.Es) * (1 - Math.pow(this.mus, 2)) * alpha();

SeT[1] = Math.round(100 * SeT[1]) / 100.0;

return SeT;

}

public double[] SeVe(){

double[] SeV = new double[2]; double[][] qV = qVe();

SeV[0] = (this.B * qV[1][0]) / (this.Es) * (1 - Math.pow(this.mus, 2)) * alpha();

SeV[0] = Math.round(100 * SeV[0]) / 100.0;

SeV[1] = (this.B * qV[1][1]) / (this.Es) * (1 - Math.pow(this.mus, 2)) * alpha();

SeV[1] = Math.round(100 * SeV[1]) / 100.0;

return SeV;

}

public double[] SeMe(){

```

```

double[] SeM = new double[2]; double[][] qM = qMe();
SeM[0] = (this.B * qM[1][0]) / (this.Es) * (1 - Math.pow(this.mus, 2)) * alpha();
SeM[0] = Math.round(100 * SeM[0]) / 100.0;
SeM[1] = (this.B * qM[1][1]) / (this.Es) * (1 - Math.pow(this.mus, 2)) * alpha();
SeM[1] = Math.round(100 * SeM[1]) / 100.0;
return SeM;
}

public double Ic(double m, double n){
double I, a, b;
a = Math.pow(m, 2) + Math.pow(n, 2); b = 1.0 + Math.pow(n, 2);
I = 2 * ((m * n / Math.sqrt(1.0 + a)) * ((a + b) / (a*b)) + Math.asin(m /
Math.sqrt(a*b))) / Math.PI;
return I; }

public double Sc(){
double FSc, p0, da, dm, db, dprom;
p0 = this.H1 * this.g1C + this.H2 * (this.g2C - 1) + (this.Hc / 2.0) * (this.gcC - 1);
da = this.q * Ic(this.L / this.B, ((this.H1 - this.Df) + this.H2) / (this.B / 2));
dm = this.q * Ic(this.L / this.B, ((this.H1 - this.Df) + this.H2 + this.Hc / 2) /
(this.B / 2));
db = this.q * Ic(this.L / this.B, ((this.H1 - this.Df) + this.H2 + this.Hc) / (this.B / 2));
dprom = (da + 4 * dm + db) / 6;
FSc=((this.Cc*this.Hc)/(1+this.e0))*Math.log10(((p0/1000.0)+dprom)/(p0/1000.0
));
return FSc;
}

```

```

}

public double[][] Scp(){
    double tmin, tv, sc, sci;

    double[][] scp = new double[3][9]; sc = Sc();

    for(int i = 10; i < 100; i = i + 10){

        tv = Tv(i);

        tmin = (tv * Math.pow(this.Hc, 2)) / this.Cv; sci = (i / 100.0) * sc;

        scp[0][i / 10 - 1] = Math.round(100 * tmin / (60 * 12)) / 100.0;

        scp[1][i / 10 - 1] = Math.round(100 * tv) / 100.0;

        scp[2][i / 10 - 1] = Math.round(100 * sci) / 100.0;}

    return scp;

}

public double Tv(int up){

    double tv1;

    tv1 = ((Math.PI / 4) * Math.pow(up / 100.0, 2)) / Math.pow(1 - Math.pow(up /
    100.0, 5.6), 0.357);

    return tv1;

}}

```