

**Tecnológico de Costa Rica**  
**Escuela de Ingeniería Electrónica**



**Sistema de redes de sensores para la predicción y prevención de  
deslizamientos de tierra**

**Informe de Proyecto de Graduación para optar por el título de Ingeniero en  
Electrónica con el grado académico de Licenciatura**

**Emanuel Zúñiga Infante**  
**200621772**

**Cartago, Junio de 2012**

Cartago, Junio 2012  
INSTITUTO TECNOLÓGICO DE COSTA RICA  
ESCUELA DE INGENIERÍA ELECTRÓNICA  
PROYECTO DE GRADUACIÓN  
Tribunal Evaluador

Proyecto de Graduación defendido ante el presente Tribunal Evaluador como requisito para optar por el título de Ingeniero en Electrónica con el grado académico de Licenciatura, del Instituto Tecnológico de Costa Rica.

Miembros de Tribunal



Ing. Nestor Hernández H.

Profesor Lector



Ing. Roberto Pereira

Profesor Lector



Ing. Marvin Hernández Cisneros

Profesor Asesor

Los miembros de este Tribunal dan fe de que el presente trabajo de graduación ha sido aprobado y cumple con las normas establecidas por la Escuela de Ingeniería Electrónica.

Cartago, de Junio 2012

## **DECLARATORIA DE AUTENTICIDAD**

Declaro que el presente Anteproyecto de Graduación ha sido realizado enteramente por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos propios.

En los casos en que he utilizado bibliografía, he procedido a indicar las Fuentes mediante las respectivas citas bibliográficas.

En consecuencia, asumo la responsabilidad total por el trabajo de anteproyecto realizado y por el contenido expuesto del mismo.

Cartago, Junio de 2012



---

Emanuel Zúñiga Infante

Cédula: 1-1353-0032

## **Resumen**

La comunidad de Jucó ubicada en el Valle de Orosi, Cartago, se encuentra expuesta al riesgo de deslizamientos de tierra o aludes, por lo tanto se pretende diseñar un sistema de monitorización de variables climáticas que permita por medio de la utilización de modelados matemáticos para deslizamientos, la prevención y predicción de este tipo de fenómenos. Esto con el fin de alertar a la población en caso de la necesidad de evacuar la zona activa.

Por medio de la utilización de una red de estaciones meteorológica inalámbricas se realiza la medición y monitorización en tiempo real de las variables en estudio. Dichas estaciones están interconectadas a un nodo central en el cual se tiene una unidad de control. En esta unidad de control se utiliza un modelado matemático que por medio de los datos obtenidos, genera diferentes alertas respecto a posibles deslizamientos de tierra con la finalidad de que la población tome las medidas necesarias para prevenir los efectos de este tipo de fenómenos naturales

**Palabras claves:** Alud, deslizamiento, red, estaciones meteorológicas inalámbricas, modelado matemático, medición, monitorización.

## **Abstract**

Jucó community in Orosi Valley, Cartago, is exposed to a landslide risk, for that reason is intended to design a monitoring system of a weather variables that allows via mathematic landslide modeling, the prevention and prediction of this kind of phenomena, this in order to alert the Jucó population in case of the need to evacuate the active landslide zone.

**Keywords:** Landslide, wireless weather station network, mathematic model, measurement, monitoring.

## **Dedicatoria**

*En primera instancia dedico este trabajo a Dios y a la virgen de los ángeles, quien me proveen todo lo que tengo y lo que soy.*

*A mis padres Ana y Guido, que han estado siempre cerca de mí, me han apoyado y han permitido que en este momento me encuentre aquí, a punto de concluir mi carrera, son un ejemplo de lucha y de sacrificio que me enseña a valorar lo que tengo.*

*A mis hermanas Andrea y Marián, que siempre me han manifestado su apoyo y cariño incondicional.*

*A mi novia Keren, quien ha sido un gran apoyo y motivación y me ha hecho mejorar en muchos aspectos de mi vida, y ha estado siempre dispuesta a brindarme su respaldo y cariño.*

## **Agradecimiento**

*A Dios por haberme dado todas las herramientas para poder finalizar este ciclo tan importante en mi vida.*

*A todos los profesores de la carrera por haberme brindado la oportunidad de adquirir el conocimiento que tengo. Y por todo el tiempo y colaboración que me han dado para poder estar aquí finalizando mi proyecto de graduación.*

*A todos mis compañeros del TEC y de la vida, que siempre estuvieron como soporte en el quehacer como estudiantes y como amigos.*

*A Johnny Meléndez que ya no nos acompaña en cuerpo presente pero que siempre estará en nuestros corazones como un gran amigo y compañero, que siempre estuvo ahí para apoyarme y regalarme su amistad.*

## ÍNDICE

1. Capítulo 1: Introducción.....	1
1.1 Problema existente e importancia de su solución. ....	1
1.2 Solución seleccionada.....	2
2. Capítulo 2: Meta y objetivos .....	3
2.1 Meta .....	3
2.2 Objetivo general.....	3
2.3 Objetivos específicos.....	3
3. Capítulo 3: Marco teórico.....	4
3.1 Descripción del sistema .....	4
3.2 Red de Sensores .....	4
3.3 Monitorización, notificaciones y aplicación del modelado matemático.....	4
3.4 Sensores Oregon Scientific .....	5
3.5 Transmisión de datos .....	12
3.6 Modelado matemático: Pendiente infinita.....	13
4. Capítulo 4: Procedimiento metodológico.....	19
4.1 Reconocimiento y definición del problema .....	19
4.2 Obtención y análisis de la información .....	21
4.3 Evaluación de las alternativas y síntesis de una solución .....	23
4.4 Implementación de la solución .....	24
5. Capítulo 5: Descripción detallada de la solución.....	28
5.1 Análisis de soluciones y solución final .....	28
5.2 Monitoreo de las variables físicas obtenidas.....	31
5.3 Implementación del modelo matemático y obtención del factor de riesgo.....	34
5.4 Análisis del factor de riesgo y generación de alertas por diversos medios.....	36
5.5 Notificaciones automáticas. ....	47
5.6 Simulación de condiciones de catástrofe y pruebas de la implementación. ....	51
6. Capítulo 6: Análisis de resultados.....	55
6.1 Resultados.....	55
6.2 Análisis.....	64
7. Capítulo 7: Conclusiones y recomendaciones.....	67
7.1 Conclusiones.....	67

7.2	Recomendaciones .....	68
8.	Capítulo 8: Bibliografía y anexos .....	69
8.1	Bibliografía.....	69
9.	Apéndices y anexos .....	70
9.1	Apéndices.....	70
A.1	Glosario, abreviaturas y simbología.....	70
A.2	Hoja de información del proyecto.....	71
A.3	Descripción de la institución .....	72
A.4	Descripción del departamento en la que se realizó el proyecto.....	72
10.	Formulas.....	73



## ÍNDICE DE FIGURAS

Figura 1.1 Diagrama general de la solución.....	2
Figura 3.1. Diagrama de primer nivel de la solución .....	4
Figura 3.2. Diagrama de segundo nivel de la solución.....	4
Figura 3.3. Diagrama detallado de la solución.....	5
Figura 3.4. Estación meteorológica utilizada.....	7
Figura 3.5. Base de la estación meteorológica utilizada.....	7
Figura 3.6. Visualización del despliegue de datos en la pantalla principal de la consola.....	8
Figura 3.7. Visualización del despliegue de datos en la pantalla principal de la consola.....	9
Figura 3.8. Visualización del despliegue de velocidad y dirección del viento.....	10
Figura 3.9. Visualización del despliegue del barómetro.....	11
Figura 3.10. Pantalla principal de weather OS de las estaciones Oregon.....	12
Figura 3.11 Deslizamiento con superficie de falla plana.....	13
Figura 3.12 Componentes del modelado de pendiente infinita.....	13
Figura 3.13 Componentes de la pendiente sin presión de poros.....	14
Figura 3.14 Parámetros de la pendiente sin presión de poros.....	15
Figura 3.15 Parámetros de la pendiente con presión de poros.....	17
Figura 0.1 Diagrama de primer nivel de la solución planteada.....	28
Figura 0.2 Diagrama de segundo nivel de la solución planteada a nivel de software.....	29
Figura 0.3 Ubicación del archivo en el que se almacenan las variables.....	34
Figura 0.4 Interfaz gráfica de usuario resaltando las partes de notificación vía Twitter.....	38
Figura 0.5 Sitio web de Twitter para desarrolladores.....	40
Figura 0.6 Ligar cuenta de twitter a facebook.....	43
Figura 0.7 Manipulación de variables para la simulación de condiciones.....	47
Figura 0.8 Ubicación del archivo de texto del cual se extraen las propiedades de notificaciones automáticas.....	48
Figura 0.9 Manipulación de variables para la simulación de condiciones.....	52
Figura 6.1 Notificaciones vía twitter.....	55
Figura 6.2 Notificaciones vía Facebook.....	56
Figura 6.3 Envío de una notificación manual.....	57
Figura 6.4 Visualización de una notificación manual.....	57
Figura 6.5 Establecer notificaciones automáticas cada minuto.....	58
Figura 6.6 Visualización de las notificaciones automáticas cada minuto en facebook.....	59
Figura 6.7 Visualización de las notificaciones por email con alertas automáticas cada.....	60
Figura 6.8 Emulador de puertos serie.....	61
Figura 6.9 Comprobar la funcionalidad del envío por puerto serie.....	62
Figura 6.10 Software de manipulación de las variables físicas.....	63
Figura 6.11 Archivo de salida de las variables físicas.....	63

## ÍNDICE DE TABLAS

Tabla 3.1	Variables de medición de las estaciones meteorológicas utilizadas.....	6
Tabla 3.2	Descripción de las variables de despliegue principal. ....	8
Tabla 3.3	Descripción de las variables de temperatura y relacionadas.....	9
Tabla 3.4	Descripción de las variables de velocidad del viento y relacionadas.....	10
Tabla 3.5	Descripción de las variables de presión atmosférica y relacionada.....	11
Tabla 0.1	Partes diferenciables de la solución del proyecto y a qué nivel se implementan. ....	31
Tabla 0.2	Diversos métodos de notificación y sus variantes.....	37
Tabla 0.3	Partes de la interfaz encargadas de comunicarse con Twitter. ....	39

## **1. Capítulo 1: Introducción**

En este apartado se exponen las generalidades del proyecto realizado, así como su importancia, sus características más relevantes y los aspectos que lo hacen necesario para la comunidad, así mismo se da un esbozo de lo que posteriormente se analizará como los procedimientos establecidos para lograr una óptima solución e implementación.

### **1.1 Problema existente e importancia de su solución.**

La comunidad de Jucó se encuentra ubicada al pie de las montañas del Valle de Orosi, por esta razón, la comunidad está expuesta al riesgo de posibles deslizamientos de tierra debido a que esta zona se caracteriza por tener un alto índice de precipitaciones durante la estación húmeda.

De esta situación nace la necesidad de desarrollar una herramienta que permita determinar el potencial peligro de deslizamientos en la zona con el fin de alertar a la comunidad en caso de la necesidad de evacuación.

Beneficios:

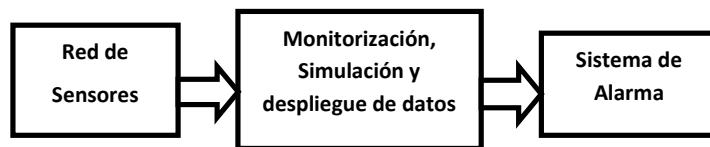
- Conservación de vidas humanas
- Prevención de posibles deslizamientos

## 1.2 Solución seleccionada

La solución planteada se divide en 3 partes funcionales:

- Sistema de red de sensores
- Monitorización, simulación y despliegue de datos (nodo central)
- Sistema de alarma para alertar de la posibilidad de un alud o deslizamiento

En la figura siguiente se muestra el diagrama de bloques de primer nivel para la solución



**Figura 1.1** Diagrama general de la solución.

La solución planteada consiste en una red de sensores meteorológicos de la marca OregonScientific, los cuales se muestran en la Figura 1.2 (A). Estos sensores se conectan de forma inalámbrica a un nodo central por medio de una estación base o consola de la misma marca, la que se muestra en la figura 1.2 (B). A su vez esta consola se conecta por medio del puerto USB a una computadora en donde se realizan tareas como recolección de datos, simulaciones de condiciones críticas, aplicación de modelos matemáticos, graficas de las variables medidas, creación de bases de datos de condiciones previas del sistema, despliegue de datos, alarmas tanto visuales como sonoras y factores de seguridad de pendiente, ente otros.

## **2. Capítulo 2: Meta y objetivos**

### **2.1 Meta**

Diseñar e Instalar un sistema de monitorización, predicción y prevención de deslizamientos de tierra en la región de Jucó de Orosi para la vigilancia de los taludes activos.

### **2.2 Objetivo general**

Desarrollar un sistema de predicción y prevención de deslizamientos eficiente, que permita a la población de Jucó de Orosi poder ser alertada sobre la ocurrencia de este tipo de fenómenos

### **2.3 Objetivos específicos**

**2.3.1** Implementar un monitoreo continuo de las variables físicas obtenidas por la parte del proyecto realizada por Rodrigo Alfaro.

**2.3.2** Desarrollar una interfaz gráfica y un sistema de software capaz de utilizar el modelado matemático establecido y calcular el actor de seguridad del terreno.

**2.3.3** Elaborara un sistema de notificaciones eficaz por diferentes medios para alertar a la población en caso de detectar posibilidad de riesgo.

**2.3.4** Desarrollar rutinas a nivel de software que permitan simular condiciones ambientales para realizar pruebas de implementación del sistema.

### 3. Capítulo 3: Marco teórico

#### 3.1 Descripción del sistema

En la siguiente figura se muestra el diagrama de primer nivel del sistema:

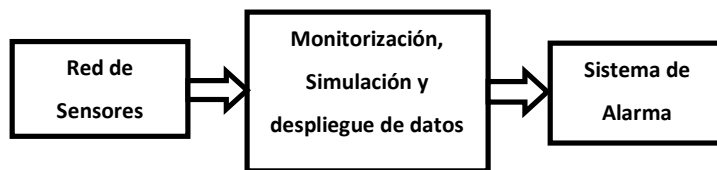


Figura 3.1. Diagrama de primer nivel de la solución

#### 3.2 Red de Sensores

La red de sensores consta de dos etapas, la estación meteorológica y la consola que permite la conexión a la PC. El diagrama de bloque de esta etapa se muestra en la siguiente figura.

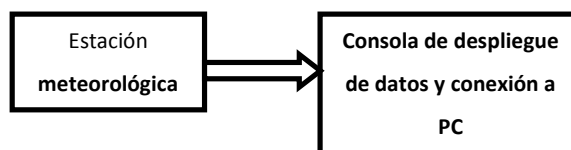
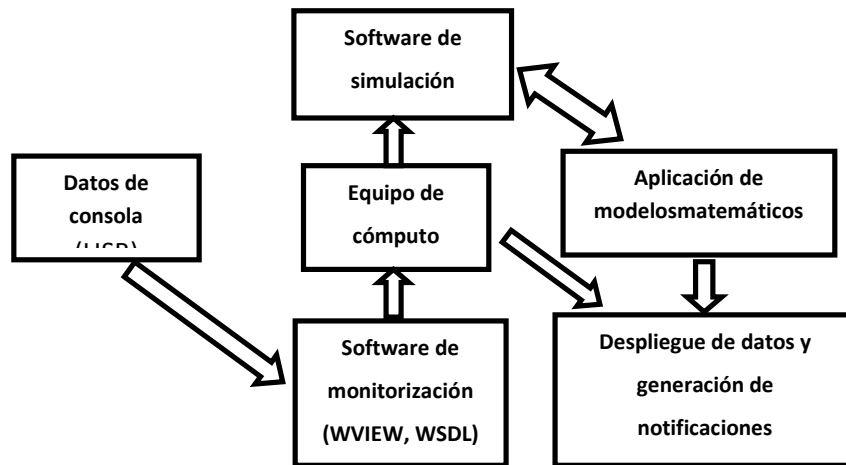


Figura 3.2. Diagrama de segundo nivel de la solución.

#### 3.3 Monitorización, notificaciones y aplicación del modelado matemático.

Esta etapa del sistema consiste en una computadora con herramientas de software que permiten la obtención de los datos así como software de

manipulación matemática para realizar simulaciones y diferentes operaciones sobre los datos obtenidos. El diagrama de segundo nivel de esta etapa se muestra en la figura a continuación.



**Figura 3.3.** Diagrama detallado de la solución.

La etapa de monitorización y despliegue muestra los datos obtenidos de la red de sensores y monitorea los cambios en los mismos. En esta etapa es donde los datos se introducen en el modelo matemático para deslizamientos el cual devuelve un valor proporcional al riesgo de un alud o deslizamiento, de esta manera al tener un valor que represente un alto riesgo, se enviara una señal a la etapa de alarmas.

### **3.4 Sensores Oregon Scientific**

#### **3.4.1 Variables de medición**

La red de sensores utiliza estaciones meteorológicas de la marca OregonScientific, modelo WMR100, para la recolección y envío de datos de las diferentes físicas que se requieren medir. En la tabla 3.1 se mencionan las diferentes variables que se pueden medir con este equipo.

**Tabla 3.1** Variables de medición de las estaciones meteorológicas utilizadas.

<b>Variables</b>	<b>Unidades</b>
Temperatura interna (Base)	Grados Celsius (°C), grados Fahrenheit (°F)
Temperatura externa (Estación)	Grados Celsius/Fahrenheit
Presión atmosférica	Milímetros de mercurio(mmHg), pulgadas de mercurio (inHg), milibares /hectopascales (mb/hPa)
Humedad	Nivel alto, nivel medio, nivel bajo
Intensidad de rayos UV	Bajo, medio, alto, muy alto, extremadamente alto
Cantidad de precipitación	Pulgadas por hora (In/ hr), milímetros por hora (mm/hr)
Velocidad del viento	m/s, km/h, mph o nudos
Dirección del viento	N, NE, NO, E, O, S, SE, SO, también grados de desviación con respecto al norte
Fase lunar	

### **3.4.2 Descripción de la consola o base**

Para la visualización de las diferentes condiciones meteorológicas, las estaciones que se muestran en la figura 3.4 a continuación cuentan con una consola o base la cual recibe la información de forma inalámbrica y la despliega en una pantalla, la consola se muestra en la figura 3.5.



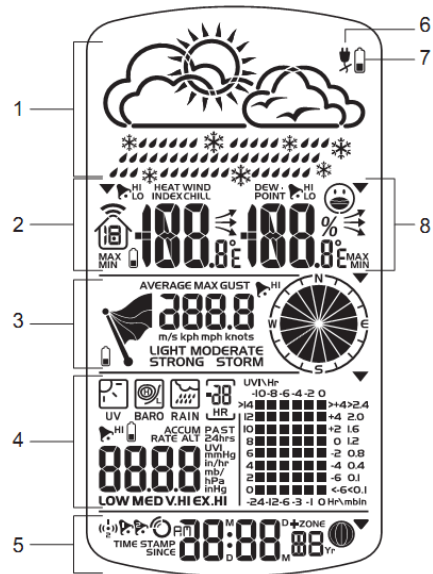


**Figura 3.4.** Estación meteorológica utilizada.



**Figura 3.5.** Base de la estación meteorológica utilizada.

En esta consola se pueden realizar las lecturas de todas las variables mencionadas en la tabla 3.1 además se puede configurar para mostrar valores promedio, máximos y mínimos, así como también estadísticas de cada una de las variables por períodos dados de tiempo. En la figura 3.6 se muestra el despliegue de la consola con la descripción de cada espacio numerado.



**Figura 3.6.** Visualización del despliegue de datos en la pantalla principal de la consola.

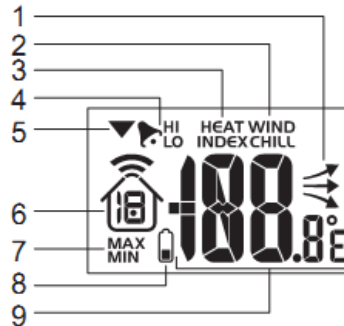
En la tabla 3.2 se muestra la descripción de las partes señaladas en la misma.

**Tabla 3.2** Descripción de las variables de despliegue principal.

N.	Descripción
1	Pronóstico del tiempo
2	Temperatura, índice de calor, area de viento frío
3	Velocidad y dirección del viento
4	Barómetro / nivel de precipitación / intensidad UV
5	Reloj / alarma / calendario / fase lunar
6	Ícono de conexión AC (cuando está conectado)
7	Ícono de batería baja
8	Humedad / punto de rocío

Cada uno de los campos de la tabla se describe a continuación.

### 3.4.3 Temperatura, índice de calor, área de viento frío



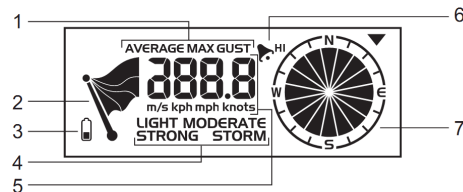
**Figura 3.7.** Visualización del despliegue de datos en la pantalla principal de la consola.

En la tabla 3.3 se muestra la descripción de los componentes señalados en la figura 3.7

**Tabla 3.3** Descripción de las variables de temperatura y relacionadas.

N.	Descripción
1	Tendencia de temperatura
2	Nivel de viento frío, muestra temperatura
3	Índice de calor, muestra temperatura
4	Alto/bajo temperatura, índice de calor, índice de viento, alarmas colocadas
5	Ícono de área seleccionada
6	Canal de temperatura y humedad en interiores/exteriores
7	Máximo/Mínimo de temperatura
8	Batería de sensor externo baja
9	Temperatura C°/F°

### 3.4.4 Velocidad y dirección del viento



**Figura 3.8.** Visualización del despliegue de velocidad y dirección del viento.

En la tabla 3.4 se muestra la descripción de las partes señaladas en la figura 3.8.

**Tabla 3.4** Descripción de las variables de velocidad del viento y relacionadas.

N.	Descripción
1	Niveles de velocidad el viento
2	Indicador del nivel de la velocidad del viento
3	Batería de sensor externo baja
4	Descripción del nivel de velocidad del viento
5	Lectura de la velocidad del viento
6	Alarma de ráfagas de viento altas fijada
7	Dirección del viento

### 3.4.5 Barómetro / nivel de precipitación / intensidad UV

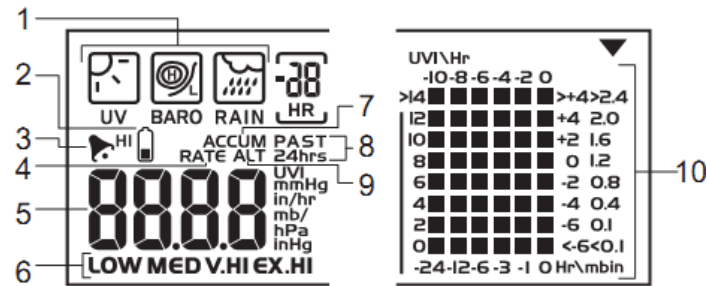


Figura 3.9. Visualización del despliegue del barómetro.

En la siguiente tabla se muestra una descripción más detallada de las variables relacionadas con la medición de presión atmosférica.

Tabla 3.5 Descripción de las variables de presión atmosférica y relacionada.

N.	Descripción
1	Lectura de Barómetro/Intensidad UV/Precipitación
2	Batería baja del sensor externo de UV/sensor de lluvia
3	Alarma de UV/Barómetro/Precipitación fijada
4	Rango de lluvia
5	Lectura de UVI / Presión barométrica / Precipitación
6	Indicador de nivel UV
7	Precipitación acumulada
8	Precipitación de las últimas 24 horas
9	Altitud
10	Histograma de barras de UVI / Presión barométrica / Precipitación

### 3.5 Transmisión de datos

Las estaciones OregonScientific transmiten datos de forma inalámbrica a la consola o base el cual función de receptor. A su vez, esta consola almacena los datos registrados en las últimas 24 horas. El manual del fabricante se especifica que la frecuencia de transmisión es de datos de 433Mhz, con una distancia de conexión de 100m en espacios abiertos.

Para la interconexión de las estaciones, cada consola cuenta con 10 canales, por lo cual, una sola consola puede tener conectadas 10 estaciones como las mostradas en la figura 3.9.

#### 3.5.1 Conexión con PC

Para la conexión con la PC, cada base cuenta con un puerto USB para este propósito. Se deben instalar los controladores para la base así como el software Weather OS de OregonScientific para poder ver los datos en pantalla. En la figura 3.10 se muestra la pantalla principal del Weather OS



**Figura 3.10.** Pantalla principal de weather OS de las estaciones Oregon.

### 3.6 Modelado matemático: Pendiente infinita

El modelo de pendiente infinita, es un modelo aplicable a deslizamientos que se considera que poseen una superficie de falla plana, es decir, se considera que el deslizamiento ocurre sobre una superficie paralela al mismo. En la figura 3.11 se muestra un deslizamiento con superficie de falla plana.

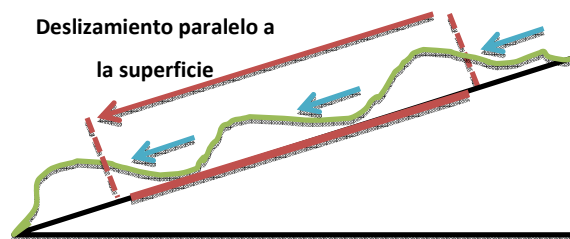


Figura 3.11 Deslizamiento con superficie de falla plana.

#### 3.6.1 Factor de seguridad sin pendiente infinita

Las diferentes componentes de la pendiente para modelo se muestran en la siguiente figura.

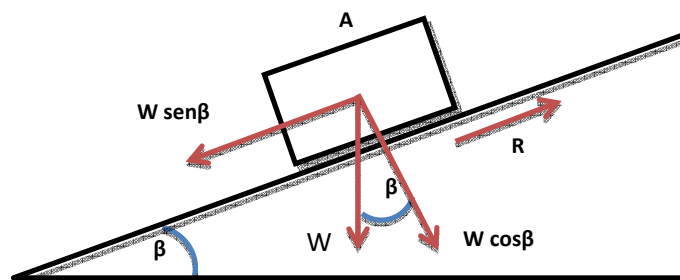


Figura 3.12 Componentes del modelado de pendiente infinita.

Componente de corte del peso:  $W \sin \beta$

Componente normal del peso:  $W \cos \beta$

Fuerza de corte:

$$\sigma = \frac{W \sin \beta}{A} \quad (0.1)$$

Criterio de Mohr-Coulomb:

$$s = c + \sigma \tan \phi \quad (0.2)$$

Fuerza de la caída del bloque:

$$R = cA + W \cos \beta \tan \phi \quad (0.3)$$

### 3.6.2 Factor de seguridad

La ecuación (0.4) muestra el factor de seguridad:

$$FS = \frac{cA + W \cos \beta \tan \phi}{W \sin \beta} \quad (0.4)$$

- $c$ : Cohesión en kPa
- $\phi$ : ángulo interno de fricción, en grados

### 3.6.3 Factor de seguridad con pendiente infinita, sin presión de poros

Las diferentes componentes de la pendiente, para modelo sin presión de poros, se muestran en la figura a continuación.

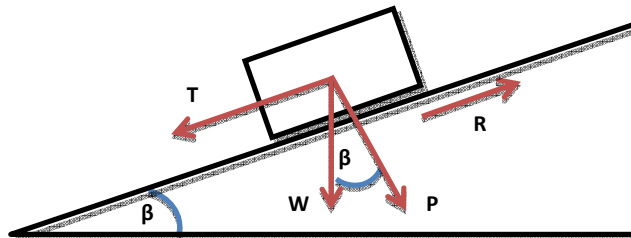


Figura 3.13 Componentes de la pendiente sin presión de poros.



Siendo  $W = \gamma Z$  Peso del bloque, con  $\gamma$  : unidad de peso del suelo en  $N/m^3$

Componente de corte del peso:

$$T = W \sin \beta \quad (0.5)$$

Sustituyendo  $W$ :

$$T = \gamma Z \sin \beta \quad (0.6)$$

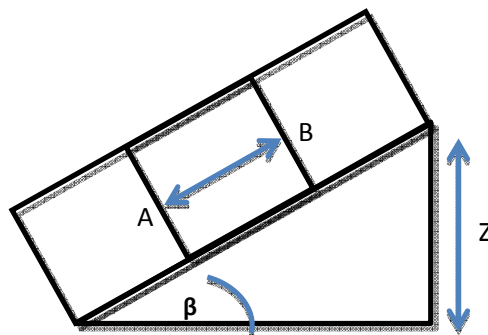
Componente normal del peso:

$$P = W \cos \beta \quad (0.7)$$

Sustituyendo  $W$ :

$$P = \gamma Z \cos \beta \quad (0.8)$$

Las condiciones de la parte alta y baja de la pendiente son ignoradas



**Figura 3.14** Parámetros de la pendiente sin presión de poros.

Tenemos que:

$$AB = \frac{1}{\cos \beta} \quad (0.9)$$

Siendo el stress igual a fuerza entre área, se tiene la fórmula para el stress normal:

$$\sigma_n = \frac{P}{AB} \quad (0.10)$$

Sustituyendo A, B y P de (0.9) y (0.8):

$$\sigma_n = \frac{\gamma Z \cos \beta}{\frac{1}{\cos \beta}} = \gamma Z \cos^2 \beta \quad (0.11)$$

El stress de corte se representa por:

$$\tau = \frac{T}{AB} \quad (0.12)$$

Sustituyendo A, B y T de (0.9) y (0.6), respectivamente, se obtiene:

$$\tau = \frac{\gamma Z \sin \beta}{\frac{1}{\cos \beta}} = \gamma Z \cos \beta \sin \beta \quad (0.13)$$

Entonces, el factor de seguridad viene dado por:

$$FS = \frac{c' + \sigma_n \tan \phi}{\tau} \quad (0.14)$$

$$FS = \frac{c' + \gamma Z \cos^2 \beta \tan \phi}{\gamma Z \cos \beta \sin \beta}$$

### 3.6.4 Factor de seguridad con pendiente infinita y presión de poros

El factor de seguridad con pendiente infinita y presión de poros es mejor que los factores de seguridad anteriores (secciones: 3.6.3 y 3.6.4), ya que relaciona la presión de poros y la presencia del agua en el nivel freático de causantes de la estabilidad de la montaña. En la figura N se puede observar un diagrama de los parámetros de este modelo.

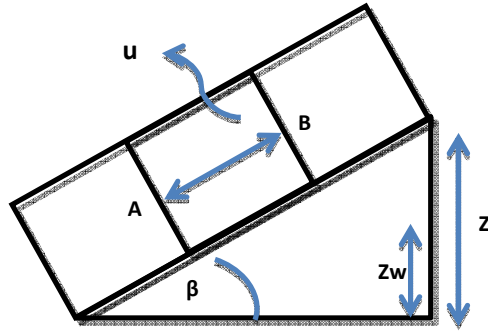


Figura 3.15 Parámetros de la pendiente con presión de poros.

### 3.6.5 Componente normal del peso del agua

El peso del agua viene dado por:  $W_w = \gamma_w Z_w$ , si  $m = \frac{Z_w}{Z}$ , es la relación entre la altura de la pendiente y la altura del nivel freático, se tiene que:

$$W_w = \gamma_w Z m \quad (0.15)$$

$$P_w = W_w \cos \beta \quad (0.16)$$

Componente normal del peso del agua:  $P_w = \gamma_w Z m \cos \beta$

### 3.6.6 Presión de poros en AB

La presión de poros es la presión que ejerce el agua entre las partículas, reduciendo la resistencia al cizallamiento.

$$u = \frac{P_w}{AB} \quad (0.17)$$

$$u = \frac{\gamma_w Z m \cos \beta}{\frac{1}{\cos \beta}} \quad (0.18)$$

Entonces la presión de poros está dada por:

$$u = \gamma_w Z m \cos^2 \beta \quad (0.19)$$

Y el factor de seguridad con presión de poros está dado por:

$$FS = \frac{c' + (\gamma Z \cos \beta - u) \tan \phi}{\gamma Z \cos \beta \sin \beta} \quad (0.20)$$

Sustituyendo (0.19) en (0.20) se obtiene:

$$FS = \frac{c' + (\gamma - m \gamma_w) Z \cos^2 \beta \tan \phi}{\gamma Z \cos \beta \sin \beta} \quad (0.21)$$

La ecuación (0.21) tiene los siguientes parámetros:

- $c'$ : Cohesión efectiva [ $\text{N/m}^2$ ]
- $\gamma$ : Peso unitario del suelo [ $\text{N/m}^3$ ]
- $m$ : Relación de alturas
- $\gamma_w$ : Peso unitario del agua [ $\text{N/m}^3$ ]
- $Z$ : Profundidad de la falla [m]
- $\beta$ : Inclinación de la pendiente ( $^\circ$ )
- $\phi$ : Ángulo efectivo de la resistencia a la fricción ( $^\circ$ )

Analizando el factor de seguridad obtenido, como  $m = \frac{Z_w}{Z}$ , al aumento paulatino de las precipitaciones, el nivel freático,  $Z_w$ , aumenta, y por ende  $m$  crece, disminuyendo el numerador conforme llueve, hasta el punto que el factor de seguridad podría volverse negativo. Con respecto al factor de seguridad, se tienen las siguientes relaciones:

**FS > 1**, Pendiente estable

**FS = 1**, Pendiente a punto de fallar

**FS < 1**, Pendiente inestable

## **4. Capítulo 4: Procedimiento metodológico**

El desarrollo e implementación de este proyecto requiere de un método de diseño en ingeniería, donde se plantee de manera concisa el problema que se enfrenta para proceder a generar una o varias alternativas que traigan consigo la solución del mismo. Para finalmente, luego de un proceso selectivo de análisis elegir la o las soluciones más eficientes que resuelven el problema establecido.

En los siguientes apartados se presentan las etapas que fueron afrontadas en el desarrollo de este proyecto, tomando en cuenta desde la etapa inicial que origina su planteamiento hasta su solución y ejecución de la misma.

### **4.1 Reconocimiento y definición del problema**

Debido a los lamentables acontecimientos ocurridos en el pasado en donde deslizamientos de tierra han afectado a la comunidad de Jucó de Orosi, los ingenieros Marvin Hernández C. y Nestor Hernández H. han planteado la posibilidad de desarrollar un sistema que sea capaz de predecir posibles eventos catastróficos y alertar a la población cuando se esté cerca de la ocurrencia de un alud de tierra en la comunidad.

En el pasado reciente se ha desarrollado un primer acercamiento para brindar una solución a este problema, pero ha quedado inconclusa, por lo que el primer paso es evaluar la solución anterior y rescatar las características de la solución propuesta anteriormente que se deseen conservar y de este modo lograr rescatar el trabajo para no empezar la investigación desde cero.

Se debe recalcar que la Universidad de Costa Rica ha efectuado diversas investigaciones acerca de los eventos de deslizamientos producidos en los años anteriores, efectuando análisis sobre el terreno y obteniendo constantes muy importantes y necesarias para los diversos modelados, esta información se debe

recopilar y utilizar de manera adecuada, esto con el fin de ahorrar tiempo, ya que el mismo se limita a seis meses, que es el tiempo establecido por el departamento de Ingeniería Electrónica del TEC.

Además de esto se debe adecuar la solución a los recursos que se brindan ya que el presupuesto es limitado, esto es factor preponderante para proporcionar una solución que se adecue y presente las necesidades y requerimientos tanto de precio como de efectividad a la hora de prevenir los deslizamientos.

Otro factor a tomar en cuenta es el consumo energético ya que al tratarse de una zona de difícil acceso se debe tener presente que el mantenimiento de las estaciones de obtención de datos debe ser mínimo, ya que no se puede estar dando atención a las estaciones con mucha regularidad.

Para el sistema de predicción se debe tener un porcentaje de efectividad mayor al 98 por ciento, esto debido a las características de sistema y su propósito, ya que una alarma equivocada puede causar desde evacuaciones innecesarias hasta el gasto de dinero excesivo, ya que las evacuaciones cuestan dinero y esfuerzo humano importante. Por otra parte si no se da la alarma con un margen de tiempo suficiente antes de la ocurrencia de un evento esto puede causar desde daños materiales muy importantes hasta la pérdida de vidas humanas, lo cual sería catastrófico.

Es por ello que las simulaciones para probar los diseños son de vital importancia, ya que es muy difícil conseguir situaciones y condiciones de desastre

En cuanto a las notificaciones y sistema de alarma que es la parte principal de este proyecto, se deben tomar en cuenta muchos factores. Uno de ellos es que se debe ser capaz de alertar a la población por diferentes medios, ya que en este caso cuanto más alertas se emitan y por diferentes medios mejor, ya que la idea es que toda la población y a su vez las autoridades pertinentes se enteren de la

posibilidad de deslizamiento inmediatamente, para que se tomen las medidas del caso.

## **4.2 Obtención y análisis de la información**

Como fuente primaria para conocer el desarrollo de sistema anteriormente desarrollado como primera etapa de experimentación , se contó con el informe de proyecto de graduación para optar por el título de Ingeniero en Electrónica con el grado académico de Licenciatura, del Ingeniero Randall Arrieta Duarte denominado: “Proyecto de investigación del sistema de monitorización y alarma contra los deslizamientos etapa I”, del cual se extrae información de primera mano y de mucha importancia en la comprensión del problema que se afronta debido a la que este proyecto es continuación de el mencionado proyecto de investigación.

En este informe podemos encontrar información como lo es el modelado utilizado en el proyecto anterior, el cual sirve como base para esta nueva etapa, además de eso la información del sistema de software desarrollado, esto con el fin de evaluar la posible utilización de ese sistema de un buen inicio para el nuevo sistema que se pretende desarrollar.

Con respecto al nuevo desarrollo que se quiere plantear es necesaria la investigación acerca del lenguaje de programación adecuado para poder implementar un sistema que pueda cumplir con todos los requerimientos, ya que al ser los objetivos de este proyecto principalmente el poder aplicar el modelado matemático determinado y con los resultados de esta utilización poder desarrollar un sistema de alarma eficaz y capaz de hacer saber a la población y a las autoridades simultáneamente del riesgo de deslizamiento.

Es por eso que para la etapa de software, el requerimiento de lenguaje de programación Visual Basic .NET sirvió de norte para orientar la investigación en ésta área. A partir de ello se inicia la investigación respectiva que da como

resultado el sitio web oficial de los desarrolladores de esta aplicación, así como en los diversos foros que existen, con el fin de determinar los mejores medios para implementar la solución de una manera eficiente, ahorrando tiempo y recursos, y al mismo tiempo cumplir con todos los requerimientos.

Una de las características en las que se quiere orientar la solución es al uso de las redes sociales como medio de comunicación inmediata, esto debido a que es un medio por el cual cualquier persona que posea acceso a internet puede ser parte de estas redes sociales por lo que si se utilizan estos medios para dar la alerta de posibles deslizamientos. Es por ello que la búsqueda de posibles medios de solución fue orientada al desarrollo de aplicaciones que pudieran tener acceso a las redes sociales más significativas como los son Facebook y Twitter, para ello se empezó por investigar en la web oficial para desarrolladores de Twitter en donde se encontró información muy valiosa.

Así mismo, las notificaciones por correo electrónico se plantean como un medio muy importante para almacenar los resultados y poder consultarlos en ocasiones futuras en las que se necesite llevar control de las condiciones presentes en diversos periodos de tiempo. Para ello se ha decidió el utilizar la plataforma de desarrollo de Google, con su servicio de correo electrónico de Gmail, por lo que la investigación también se ha orientado a buscar el conectar aplicaciones con los servicios de Google, en este caso Gmail.

Las principales páginas web y manuales utilizados como información para el desarrollo del proyecto se pueden observar en el apartado de referencias bibliográficas de este documento.



### **4.3 Evaluación de las alternativas y síntesis de una solución**

Una vez dados los parámetros y condiciones necesarias para el desarrollo del proyecto, así como hecha una investigación previa de las características del equipo disponible y de las aplicaciones desarrolladas tanto en versiones anteriores del proyecto, como las disponibles con código abierto en la web, y tomando en cuenta las recomendaciones de los profesores asesores, y la orientación de estos en cuanto a la investigación se determinó una posible solución para el problema planteado.

Para verificar el funcionamiento real de las herramientas de Visual Basic.Net en el control y manipulación de archivos y de comunicación con aplicaciones de correo electrónico y redes sociales, fue necesaria la creación e implementación de programas de prueba de cada aplicación por separado para conocer el desempeño y funcionamiento real de dichos instrumentos a nivel de software.

Una vez comprobado el punto anterior, se puede precisar que las capacidades a nivel de hardware y software de las estaciones de monitoreo brindadas, sumadas al alto volumen de procesamiento de datos de una PC en aplicaciones de lenguaje de alto nivel, es que se define la aptitud de esta herramienta para desarrollar un software de monitoreo y alerta que origina este proyecto.

Otro aspecto importante referente a las herramientas de software, es que su costo económico es nulo debido a que las fuentes de bibliotecas de Visual basic.Net se clasifican como código abierto disponible para ser utilizado libremente en cualquier aplicación, además de que el compilador utilizado Microsoft Visual basic.Net 2010 Express, está en línea y es brindado de manera gratuita con fines educativos por la compañía Microsoft.

#### **4.4 Implementación de la solución**

Teniendo claros los objetivos de implementación así como las capacidades de las estaciones de monitoreo y los aspectos de hardware y software, se procede a la implementación de la solución propuesta.

Primeramente se tiene que enfatizar los diversos componentes de esta solución que son 4 a saber:

1. Monitoreo continuo de las variables físicas obtenidas.
2. Implementación del modelo matemático y obtención del factor de riesgo.
3. Análisis del factor de riesgo y generación de alertas por diversos medios.
4. Simulación de condiciones de catástrofe y pruebas de la implementación de los puntos anteriores.

Teniendo estos puntos claros se puede dividir la solución en etapas o módulos para el desarrollo efectivo de la misma.

Para el primer punto que consiste en el monitoreo continuo de las variables físicas obtenidas, hay que tener en cuenta que estas se deben obtener de un archivo .txt en formato CSV o archivo por comas, el cual es generado por la parte del proyecto desarrollada por el compañero Rodrigo Alfaro, por lo que para obtener esos datos y analizarlos es necesario de desarrollar una aplicación de software capaz de obtener esa información y almacenarla en variables para su posterior utilización en el análisis.

Es por ello que se recurre a la plataforma de desarrollo Visual Basic.net la cual brinda todos los requerimientos necesarios para la implantación de la solución esperada y que además es de carácter gratuito ya que Microsoft la proporciona abierta a estudiantes para el desarrollo con fines académicos.

Utilizando esta plataforma se comienza por desarrollar un módulo capaz de leer desde un archivo de texto con formato de valores separado por comas (CSV) y el almacenamiento de los datos obtenidos en variables globales del programa para ser utilizadas posteriormente en el análisis matemático.

Cabe destacar que cada una de las rutinas descritas es desarrollada por medio de métodos separados, esto con el fin de ser utilizadas en múltiples ocasiones durante la implementación del programa, por lo que cada método tiene sus variables de entrada y salida específicamente determinadas y únicas.

Además de ellos, como ya se sabe el archivo que se desea leer es actualizado cada minuto (tiempo en que se refrescan las mediciones por parte del hardware de monitoreo de variables físicas) es por ello que se desarrolla también un módulo capaz actualizar estas variables en el programa también cada minuto, esto con la ayuda de un módulo de ejecución por tiempo o timer en el cual se puede establecer un valor predeterminado en el cual se ejecute cierta labor, en este caso se ejecuta cada minuto el método descrito anteriormente encargado de almacenar los datos del archivo salida de la implementación paralela del proyecto (A cargo del compañero Rodrigo Alfaro).

Por otra parte el segundo punto nos conduce a la implementación del modelado matemático, el mismo es elegido en la investigación previa realizada por Rodrigo en la primera parte de esta etapa del proyecto, y ya con los datos almacenados en variables resulta simple desarrollar un módulo que reciba como entrada los datos leídos y mediante procedimientos matemáticos calcule y dé a la salida el valor numérico del factor de riesgo, esta forma se tiene almacenado en una variable el valor del factor de riesgo y de esta forma se pueda usar el mismo para determinar las posibilidades reales de un deslizamiento.

Ya teniendo los dos primeros puntos claros se procede a desarrollar una de las partes fundamentales del sistema como lo son los métodos de notificación. Los mismos se van a dividir en 3 categorías:

1. Notificación por medio de redes sociales:
  - Facebook.
  - Twitter.
2. Notificación por correo electrónico.
3. Notificación luminosa y sonora en la comunidad.

Con estos métodos definidos comienza el desarrollo de cada uno. Para la notificación por medio de redes sociales se debe primero consultar la página para desarrolladores de Twitter, en donde se debe registrar la aplicación que va a acceder a una cuenta determinada, en este caso se ha creado una cuenta específica para el monitoreo que se puede encontrar como @jucoMonitoreo, y que cualquier persona con una cuenta en esta red social puede seguir para darse cuenta de las notificaciones enviadas tanto manual, como automáticamente por medio del software de monitoreo.

Ya con la aplicación registrada se utiliza una librería ya desarrollada específicamente con el fin de postear y actualizar el status de una cuenta de twitter. Con esta librería se consigue desarrollar un método capaz de “postear” en Twitter un estado determinado, esto luego se utilizará para poder publicar las notificaciones acerca del terreno y de las variables físicas medidas.

Utilizando la herramienta que posee Twitter en la que nos permite conectar nuestra cuenta con una cuenta de Facebook se logra que los mismos Tweets que publiquemos sean visualizados en un perfil creado para este fin, el cual se puede buscar como Juco Monitoreo, y suscribirse a sus actualizaciones para poder visualizar las diversas notificaciones enviadas por este medio.

Así mismo utilizando una librería previamente creada para conectar una aplicación con una cuenta de Gmail se logran enviar notificaciones también por correo electrónico a una lista establecida de contactos, dichos contactos pueden ser modificados a placer, para así incluir o excluir a las personas necesarias en las notificaciones incluso se podría incluir algún correo de la Comisión nacional de emergencias, cruz roja, hospitales etc. Para así lograr una comunicación rápida con las autoridades pertinentes y que estén alerta de la situación.

Por último en el punto de las notificaciones de alerta, pero no menos importante están las notificaciones físicas, que consisten en sistemas de alarma de tipo sonoro y luminoso. En este punto el sistema de software también influye, ya que al generarse una alerta el sistema el mismo enviará una señal por medio del puerto serie de la PC, que a su vez será transmitido por medio de un transmisor inalámbrico hacia donde se encuentren instalados los sistemas de alerta. Esta señal es recibida e interpretada por un circuito lógico que se encarga de activar las sirenas de emergencia y los dispositivos luminosos en caso de ser necesario, esto con el fin de alertar a la población.

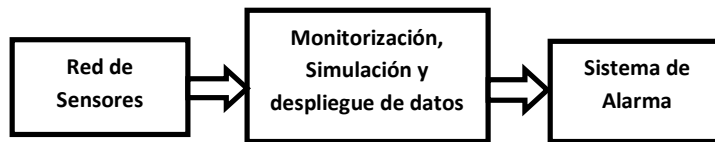
Ya desarrollado todo el sistema de monitoreo y capaz de determinar la posibilidad de deslizamiento así como implementando las diversas notificaciones antes descritas, es necesario el desarrollar un método de prueba, en el cual se simulen las variables ambientales y generar así condiciones de riesgo para así comprobar la funcionalidad del sistema.

Esto se logra a través de un segundo sistema de software, que lo que básicamente hace es dar al usuario la posibilidad de modificar los valores de las variables y guardar sobre el mismo archivo que lee el software de monitoreo nuevos valores, para que así se simulen condiciones de desastre y poder observar cómo responde el modelo ante estas variaciones.

## 5. Capítulo 5: Descripción detallada de la solución

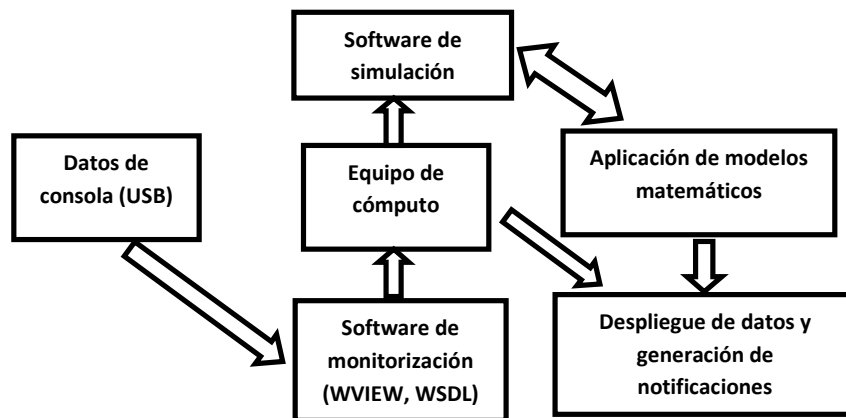
### 5.1 Análisis de soluciones y solución final

La solución general para las dos partes del proyecto planteadas consiste de una serie de requerimientos a nivel de hardware y software importantes en todo diseño de ingeniería. La figura 5.1 presenta un diagrama de primer nivel de la solución implementada, en donde se muestran tres bloques claramente definidos, como lo son la red de sensores, el sistema de monitoreo, simulación y despliegue de datos y el sistema de alarma en sí, que corresponde a todos los tipos de notificación expuestos anteriormente.



**Figura 0.1** Diagrama de primer nivel de la solución planteada.

Ya presentada este primer esbozo de la solución podemos ir más allá y analizar un esquema de segundo nivel, en el cual se desglosan las partes principales que luego se convertirán en los módulos de la solución, en la siguiente figura observamos dicho esquema de segundo nivel y posteriormente se analizarán cada una de esas partes.



**Figura 0.2** Diagrama de segundo nivel de la solución planteada a nivel de software.

De estos componentes que podemos observar en la figura anterior, se debe hacer mención que la mayoría de ellos son trabajados a nivel de software. Esto debido a las condiciones de la solución, en donde se deben tomar decisiones lógicas con base a variables físicas que son recolectadas por medio de la red de sensores elaborada en la etapa numero uno de este proyecto.

De los bloques mostrados, los trabajados y abarcados en esta etapa del proyecto son los que se explican a continuación en los siguientes puntos.

- **Equipo de cómputo:** Esto es básicamente una PC, la cual se encarga de ejecutar todas las rutinas necesarias y diseñadas para interconectar el hardware y la medición de las variables con el análisis matemático por medio del modelado establecido para este fin.
- **Aplicación de modelos matemáticos:** En esta etapa es en la que se calcula el factor de riesgo por medio del modelado matemático y la aplicación de los diversos valores de las variables físicas recolectadas por medio de la red de sensores, esta etapa es crucial ya que el éxito o fracaso de la predicción acertada de los eventos de deslizamiento

depende de ella directamente, ya que si el modelado no es aplicado de una manera adecuada se pueden generar falsas alarmas que pueden conducir a evacuaciones innecesarias o por el contrario a la no notificación de una situación de riesgo en donde se pueden dar desde daños materiales hasta pérdidas humanas lamentables.

- **Software de simulación:** Es también de vital importancia para las etapas previas a la instalación definitiva del sistema en la comunidad, ya que es estrictamente necesario simular condiciones de riesgo para hacer todas las pruebas de funcionamiento, esto debido a que es muy difícil instalar el equipo y conseguir condiciones de riesgo mediante el. Es por ello que se debe diseñar un software especial con el fin de modificar las condiciones a placer y con ellos simular esas condiciones de riesgo deseadas para probar el funcionamiento del sistema.
- **Despliegue de datos y generación de alertas:** En esta parte del sistema es en que se alerta a la población de las condiciones de la pendiente, ya sea la condición de riesgo o por el contrario que todo está en orden y que no hay peligro de deslizamiento, para ellos se ha decidido utilizar los medios de comunicación más efectivos en el momento, como lo son las redes sociales, y el correo electrónico, así como una alarma local con sonido y efectos luminosos, esto con el fin de alertar, tanto a la población de la zona como a las autoridades pertinentes, para que se empiece a gestionar todas las labores necesarias en caso de una emergencia inminente.

Es por ello que se han determinado las partes más importantes del proyecto, enumerándolas en la siguiente tabla que nos permite tener una mejor visualización de las mismas y nos explica si son implementadas a nivel de software o de hardware.



**Tabla 0.1** Partes diferenciables de la solución del proyecto y a qué nivel se implementan.

<b>Parte del proyecto</b>	<b>Nivel al que se implementan</b>
Monitoreo continuo de las variables físicas obtenidas	Software
Implementación del modelo matemático y obtención del factor de riesgo	Software
Análisis del factor de riesgo y generación de alertas por diversos medios.	Software/Hardware
Simulación de condiciones de catástrofe y pruebas de la implementación de los puntos anteriores.	Software

Queda claro entonces que la mayor parte de este proyecto es implementada a nivel de software de la PC, por lo que es necesario explicar cada una de las rutinas utilizadas de una manera detallada para la mejor comprensión de la solución.

## **5.2 Monitoreo de las variables físicas obtenidas**

En este apartado se describe con más detalle la implementación de el monitoreo continuo de las variables físicas que son obtenidas por medio de la red de sensores, a cargo de la parte del proyecto elaborada por el compañero Rodrigo Alfaro.

Para esto, como ya se ha mencionado anteriormente, hay que tener en cuenta que las variables físicas se deben obtener de un archivo .txt en formato CSV o valores separados por comas, por lo que para obtener esos datos y analizarlos es necesario de desarrollar una aplicación de software capaz de obtener esa información y almacenarla en variables para su posterior utilización en el análisis.

Se ha elegido para esto utilizar la plataforma de desarrollo Visual Basic.Net la cual se ajusta excelentemente a las necesidades del diseño planteadas para poder obtener la solución esperada y que además es de carácter gratuito ya que Microsoft la proporciona abierta a estudiantes para el desarrollo con fines académicos.

En el siguiente fragmento de código representa la rutina elaborada para la lectura de archivos en Visual Basic.Net, la misma lee un archivo, que se establece como un archivo separado por comas y utiliza la última línea de este archivo para encontrar los valores más actualizados de las variables físicas necesarias para la implementación.

Cabe mencionar que al final de la rutina los valores son guardados en variables globales del sistema por lo que pueden ser utilizadas en cualquier momento, estas variables son actualizadas cada minuto en el archivo por medio de la red de sensores, y así mismo son actualizadas cada minuto en el sistema.

```
'#####  
'Comienza lectura del archivo de valores de variables atmosfericas  
  
Using MyReader As New  
Microsoft.VisualBasic.FileIO.TextFieldParser("E:\variables.txt")  
  
MyReader.TextFieldType = FileIO.FieldType.Delimited  
MyReader.SetDelimiters(",")  
  
Dim a As String  
Dim currentRow As String()  
Dim i As Integer  
i = 0  
  
While Not MyReader.EndOfData  
Try  
currentRow = MyReader.ReadFields()  
a = MyReader.LineNumber  
  
Dim currentField As String  
  
For Each currentField In currentRow  
MsgBox(currentField)
```

```

    If a = -1 Then
        atmosfericas(i) = currentField

        '#####convertir a numeros

        If IsNumeric(atmosfericas(i)) = True Then
            VariablesNumeros(i) = CDb1(atmosfericas(i))

            ' Call UnProcedimiento(OtraVariable)
        Else
            VariablesNumeros(i) = 0
        End If

        '#####termina convertir a numeros
        i = i + 1
    End If

Next

    Catch ex As
Microsoft.VisualBasic.FileIO.MalformedLineException
        MsgBox("Line " & ex.Message & _
            "is not valid and will be skipped.")

    End Try

End While

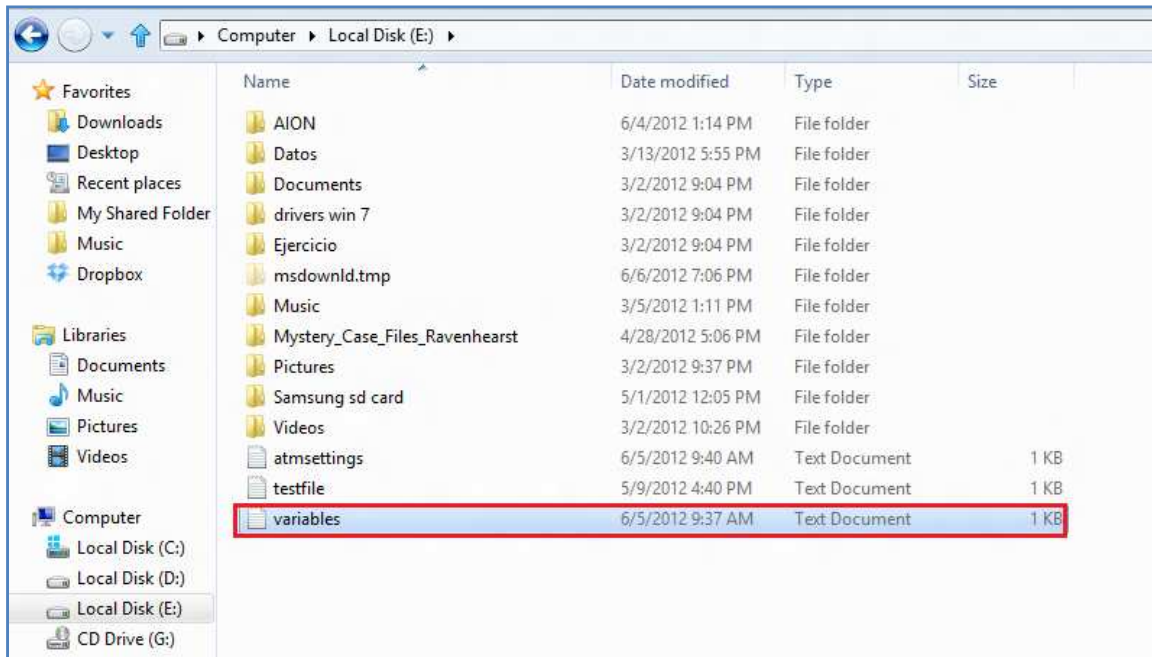
End Using

'temina lectura de variables atmosfericas, a este punto se tiene las
'variables listas para operar entre si
'#####

```

Además como ya se mencionó, el archivo que se desea leer es actualizado cada minuto (tiempo en que se refrescan las mediciones por parte del hardware de monitoreo de variables físicas) es por ello que se desarrolla también un módulo capaz actualizar estas variables en el programa también cada minuto, esto con la ayuda de un módulo de ejecución por tiempo o timer en el cual se puede establecer un valor predeterminado en el cual se ejecute cierta labor, en este caso se ejecuta cada minuto el método descrito anteriormente encargado de almacenar los datos del archivo salida de la implementación paralela del proyecto (A cargo del compañero Rodrigo Alfaro).

En la siguiente figura se puede observar la ubicación del archivo en el sistema de la PC, como se puede observar el mismo se encuentra en la siguiente dirección “E:\variables.txt”. Además de esto



**Figura 0.3** Ubicación del archivo en el que se almacenan las variables.

### **5.3 Implementación del modelo matemático y obtención del factor de riesgo.**

En este apartado se describe la implementación del modelo matemático, el cual es una simple ecuación obtenida por medio del análisis realizado por el compañero Rodrigo Alfaro en el proyecto complementario a este, la ecuación se mostro en el apartado de marco teórico de este documento, y la misma relaciona diversos factores de forma o constantes del terreno y las variables ambientales que son monitoreadas por la red de sensores.

Con la implementación de la ecuación se obtiene un factor de riesgo, que en condiciones normales es menor que uno, por lo que si ese factor se va acercando a la unidad eso quiere decir que el terreno está cediendo y que la posibilidad de deslizamiento crece.

Con dicho factor de riesgo es muy sencillo saber el momento preciso en el que se deben activar las diversas alertas (que se exponen en el siguiente apartado de este documento).

Ya que las variables son monitoreadas cada minuto, el cálculo de este factor de riesgo se hace igualmente cada minuto, esto con el fin de mantener un constante monitoreo sobre el terreno. Lo mismo se logra mediante la implementación de un timer o temporizador capaz de ejecutar labores específicas cada cierto periodo de tiempo establecido, a continuación se muestra un segmento de código que permite habilitar el timer y ejecutar la rutina cada minuto.

```
'rutina para ejecutar el timer cada minuto

Private Sub Timer2_Tick(ByVal sender As System.Object, ByVal e As_
System.EventArgs) Handles Timer2.Tick

    Dim Variables(4) As String
    Dim atmosfericas(4) As String
    Dim VariablesNumeros(4) As Double

'calculo de indice de riesgo
'modelado matemático

    indiceriesgo = VariablesNumeros(0)

'si el indice es mayor que uno

    If indiceriesgo >= 1 Then

'se envian mensajes de alerta
mensajedealerta = "ALERTA!!!! REVISAR CONDICIONES DE TERRENO Y ALERTAR A
LA POBLACION"

        Call Send("%")
        Call Send("ALERTA")
    End If
End Sub
```

```

Call Send("%")
Call Send("/n")
Call AlertaTwitter("AUTOMATICA", mensajedealerta)

Call Enviarmail("AUTOMATICA", "Este mensaje es enviado automaticamente
desde programa de monitoreo" & vbNewLine & vbNewLine & "ACTUALIZACION : "
& mensajedealerta & vbNewLine & vbNewLine & "Saludos cordiales," &
vbNewLine & "Monitoreo automatico Juco")

Else

'si no, el mensaje de alerta se setea como, no hay alertas reportadas

    mensajedealerta = "No hay alertas reportadas a esta hora"

End If

End Sub

```

Analizando el código anterior nos podemos dar cuenta de que la variable “mensajedealerta” es la encargada de contener el mensaje que se enviará por los diversos medios de notificación, y esta varia con respecto al valor del índice de riesgo calculado. Como ya se mencionó si este índice de riesgo es mayor a la unidad, las probabilidades de desastre son muy altas y se debe empezar a alertar a la población, por ello en la rutina anterior se puede observar que se hace un llamado a las subrutinas de envío de mensajes de alarma por diversos medios, estos medios así como su funcionalidad se describirán a continuación.

#### **5.4 Análisis del factor de riesgo y generación de alertas por diversos medios.**

Al tener ya establecidos los primeros puntos establecidos y a este punto teniendo un sistema capaz de obtener y monitorear las variables de salida de la primera parte del proyecto, por medio de la lectura de un archivo separado por comas, se puede empezar a desarrollar una de las partes fundamentales del mismo como lo son los diversos sistemas de notificación y de alerta, podemos separar los mismos en tres grandes grupos con sus respectivos subgrupos, los mismos son listados a continuación en la siguiente tabla.

**Tabla 5.2** Diversos métodos de notificación y sus variantes.

<b>Método de notificación</b>	<b>Variantes</b>
Notificación por medio de redes sociales	Facebook/Twitter
Notificación por medio de correo electrónico	Gmail/Hotmail o cualquier dirección de correo electrónico
Notificación a nivel de área local.	Sonora/Luminosa

De la tabla anterior podemos observar que se cuentan con métodos de alerta muy diversos, todos de igual importancia y con una función específica, ya que en una situación como la expuesta las notificaciones son de vital importancia, tanto como el análisis mismo. Se han implementado diversas formas de notificación esto para que en dado caso que un método fallara los siguientes pueden servir de respaldo.

A continuación se explican detalladamente cada uno de estos medios y su implementación por descripción de código y su respectiva implementación gráfica mediante la interfaz diseñada.

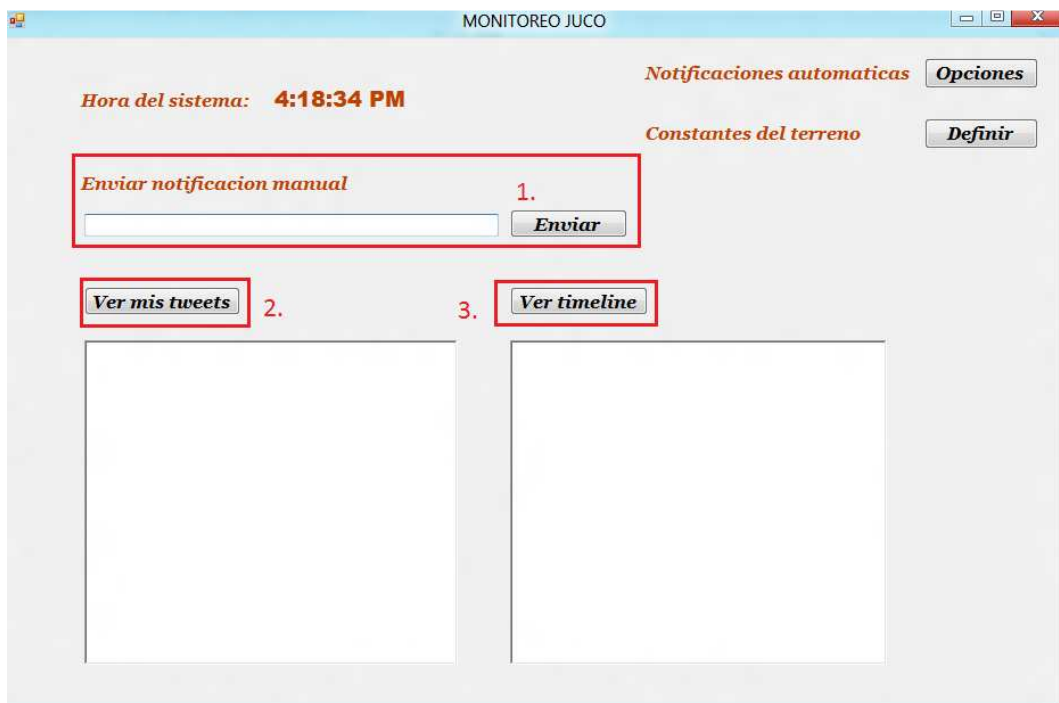
#### **5.4.1 Notificación por medio de redes sociales:**

Como es sabido, en nuestra época las redes sociales son el medio de comunicación más rápido y efectivo que existe, ya que, ahora la mayoría de la población tiene acceso a internet y por ende a este tipo de medio de comunicación y entretenimiento, en nuestros días las redes sociales que cuentan con mayor auge y cantidad de usuarios son Facebook y Twitter, por lo que se ha elegido a estas como los medios principales de notificación, a continuación se describe cada una de las implementaciones específicas según la red social utilizada.

## A. Twitter:

Con respecto a Twitter, esta es una red social cuya idea es que cada usuario puede describir lo que está pensando por medio de una frase de menos de 140 caracteres, esta modalidad será aprovechada para los intereses de este proyecto, ya que al enviar un Tweet desde el software diseñado para el proyecto cualquier persona que posea una cuenta de esta red social y que siga a la cuenta creada específicamente para las notificaciones podrá leer el estado que se acaba de publicar. La cuenta creada con el fin de publicar las notificaciones se puede encontrar bajo el siguiente usuario @jucoMonitoreo y se puede seguir libremente para conocer las notificaciones enviadas.

En la siguiente figura se puede observar la interfaz gráfica del programa creado para monitoreo y notificación de las alertas, en ella se señalan los componentes específicos relacionados con las notificaciones en Twitter.



**Figura 0.4** Interfaz gráfica de usuario resaltando las partes de notificación vía Twitter.



En la figura anterior podemos observar enumeradas partes importantes de la interfaz de usuario encargadas de comunicarse con la aplicación de Twitter. Con la siguiente tabla podemos explicar un poco mejor cada una de esas partes.

**Tabla 0.3** Partes de la interfaz encargadas de comunicarse con Twitter.

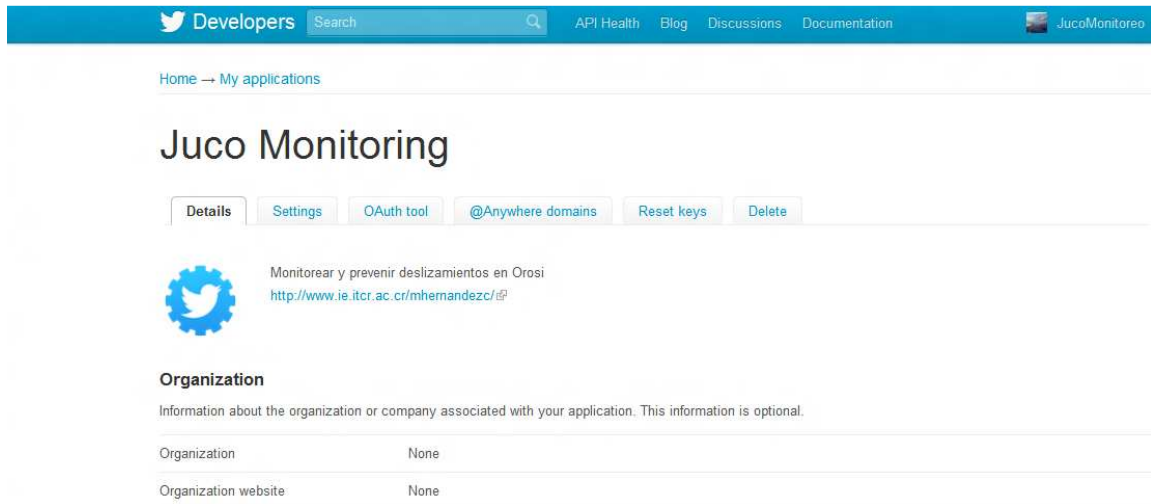
<b>Número</b>	<b>Descripción</b>
1.	Enviar notificación manual introduciendo una cadena de caracteres
2.	Visualizar los últimos 25 Tweets enviados.
3.	Ver los últimos 25 mensajes del timeline de nuestra cuenta de Twitter

Dado esto se puede comentar la función de cada parte individualmente. Con el número uno podemos observar la parte dedicada para notificaciones manuales, en esta área podemos introducir una cadena de caracteres y la misma será posteada en Twitter, utilizando para eso un formato específico en donde se incluye la hora, la fecha y la aclaración de que la actualización se hizo manualmente.

Con el número dos podemos observar el área en donde se pueden ver los últimos 25 mensajes o Tweets enviados por nuestra cuenta, esto con el fin de monitorear si los mensajes enviados se están publicando de manera correcta sin tener que acceder a la cuenta específica. Finalmente con el número tres observamos el área de visualización del timeline donde se observan los Tweets de otras cuentas a las que le hemos dado seguir desde nuestra cuenta de twitter.

Para el desarrollo de esta aplicación a nivel de código se han utilizado una librería previamente desarrollada para comunicar aplicaciones de visual basic.net con Twitter la misma se denomina `TwitterVB2.lib`. Para lograr esto se debe además registrar la aplicación a desarrollar en un sitio de twitter para desarrolladores (que se puede ver en la sección de referencias), en el cual se brinda una serie de códigos de autenticación que permiten acceder a la cuenta deseada de una manera segura, para evitar cualquier inconveniente relacionado

con la seguridad del sitio, a continuación se muestra una imagen del sitio de twitter para desarrolladores utilizado para acceder y registrar la aplicación creada.



**Figura 0.5** Sitio web de Twitter para desarrolladores.

El segmento de código que permite realizar dicha autenticación y enviar notificaciones vía twitter se puede observar a continuación.

```
Private Sub AlertaTwitter(ByVal tipo As String, ByVal mensaje As String)

    'postear en twitter

    Dim twitter As New TwitterAPI
    twitter.AuthenticateWith("McVjfYgqXCvR1rbjMjb98Q",
    "BjAt5yNTxZHpwPJg7NSThuo6zSa9R0E5F4mED2twZY", "576595435-
    0p0ZytGncnq8EwnHVqKufTcQDguXMJ8qrOLwux6S",
    "m46FsH93ak2PmXGFRueVZG8vMK6jcv5N1B6FpobGQ")

    Dim status As TwitterStatus = twitter.Update("ACTUALIZACION " &
    tipo & " " & DateString & " " & TimeOfDay & ":" & mensaje)

End Sub
```

Así mismo, para visualizar los últimos tweets de la cuenta deseada se utiliza el siguiente fragmento de código.

```
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click

    'rutina para visualizar el user timeline de twitter en un richbox

    Dim twitter As New TwitterAPI

    'credenciales

    twitter.AuthenticateWith("McVjfYgqXCvR1rbjMjb98Q",
    "BjAt5yNTxZHpwPJg7NSThuo6zSa9R0E5F4mED2twZY", "576595435-
    0p0ZytGncnq8EwnHVqKufTcQDguXMJ8qrOLwux6S",
    "m46Fsh93ak2PmXGFRueVZG8vMK6jcv5N1B6FpobGQ")

    txttweets.Clear()

    For Each tweet As TwitterStatus In twitter.UserTimeline

        txttweets.AppendText(tweet.User.ScreenName & vbNewLine &
        tweet.Text & vbNewLine & vbNewLine)

    Next

End Sub
```

Y por último para lograr visualizar el timeline complete basta con hacer una variación en los parámetros de la definición como se muestra a continuación.

```
Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click

    'rutina para visualizar el timeline de twitter en un richbox
    Dim twitter As New TwitterAPI

    'credenciales
    twitter.AuthenticateWith("McVjfYgqXCvR1rbjMjb98Q",
    "BjAt5yNTxZHpwPJg7NSThuo6zSa9R0E5F4mED2twZY", "576595435-
    0p0ZytGncnq8EwnHVqKufTcQDguXMJ8qrOLwux6S",
    "m46Fsh93ak2PmXGFRueVZG8vMK6jcv5N1B6FpobGQ")

    RichTextBox1.Clear()

    For Each tweet As TwitterStatus In twitter.HomeTimeline

        RichTextBox1.AppendText(tweet.User.ScreenName & vbNewLine &
        tweet.Text & vbNewLine & vbNewLine)
    Next
End Sub
```

Como en ocasiones anteriores estos segmentos de código representan módulos, los cuales pueden ser utilizados y llamados desde cualquier punto del sistema, por lo que facilita la implementación de los mismos.

## **B. Facebook:**

Como es sabido, facebook es la red social más utilizada en el mundo, y en nuestro país no es la excepción, por lo que utilizar este sitio web social como medio de comunicación da un valor agregado a la aplicación, ya que facebook tiene múltiples sistemas de notificación incluso a nivel de teléfono celular, por lo que las notificaciones serian inmediatas en incluso llegando a medios tan cotidianos como los teléfonos celulares.

Para poder enviar mensajes y publicarlos en el muro de una cuenta específica se ha utilizado una herramienta muy simple y poderosa que posee la cuenta de twitter, la cual nos permite ligar una cuenta de facebook a la de twitter, por lo que podemos de esta forma visualizar los mensajes que enviamos a twitter publicados en nuestra cuenta de facebook.

En la siguiente imagen podemos apreciar la forma en la que ligamos las diferentes cuantas de datos.



**Figura 0.6** Ligar cuenta de twitter a facebook.

#### **5.4.2 Notificación por medio de correo electrónico:**

Continuando con el apartado de comunicaciones y notificaciones encontramos un método que siempre será efectivo, sobre todo para almacenar información y analizarla después.

Para notificar vía email se utiliza una librería previamente creada e incluida en Visual Basic.net para conectar una aplicación con una cuenta de Gmail, por este medio se logran enviar notificaciones también por correo electrónico a una lista establecida de contactos, dichos contactos pueden ser modificados a placer, para así incluir o excluir a las personas necesarias en las notificaciones incluso se podría incluir algún correo de la Comisión nacional de emergencias, cruz roja, hospitales etc. Para así lograr una comunicación rápida con las autoridades pertinentes y que estén alerta de la situación.

En este caso se han incluido pocas direcciones pero en caso de ser necesario se pueden incluir todas las direcciones de correo que se desee, al llamarse una rutina de correo se envía el mismo a todas las direcciones de correo deseadas desde la dirección [juco.monitoring@gmail.com](mailto:juco.monitoring@gmail.com) que se ha creado con este fin.

A continuación se muestra el código encargado de esta tarea.

```
Public Sub Enviarmail(ByVal tipo As String, ByVal mensaje As String)
    'Crear un objeto de tipo mail

    Dim MyMailMessage As New MailMessage()

    'Email del cual se envia el mensaje
    MyMailMessage.From = New MailAddress("juco.monitoring@gmail.com")

    'direcciones de destinatarios
    MyMailMessage.To.Add("emanuelziga@gmail.com")
    MyMailMessage.To.Add("juco.monitoring@gmail.com")

    'Subject
    MyMailMessage.Subject = "ACTUALIZACION DE MONITOREO " & tipo & ":
" & DateString & " " & TimeOfDay

    'Cuerpo del mensaje
    MyMailMessage.Body = mensaje

    'Crear objeto SMTPClient con el SMTP server de GMail
    Dim SMTPServer As New SmtplibClient("smtp.gmail.com")

    'puerto de Gmail
    SMTPServer.Port = 587
    'logueo con las credenciales
    SMTPServer.Credentials = New
System.Net.NetworkCredential("juco.monitoring", "emma101421")

    'muy importante declarar que la conexion sea tipo seguro
    SMTPServer.EnableSsl = True

    'enviar mail
    Try
        SMTPServer.Send(MyMailMessage)
    Catch ex As SmtplibException
        MessageBox.Show(ex.Message)
    End Try
End Sub
```

Para poder conectar la aplicación con el correo electrónico es necesario utilizar las credenciales de la cuenta por lo que la seguridad está en todo momento asegurada.

#### **5.4.3 Notificación a nivel de área local:**

Por último en el punto de las notificaciones de alerta, pero no menos importante están las notificaciones físicas, que consisten en sistemas de alarma de tipo sonoro y luminoso. En este punto el sistema de software también influye, ya que al generarse una alerta el sistema el mismo enviará una señal por medio del puerto serie de la PC, que a su vez será transmitido por medio de un transmisor inalámbrico hacia donde se encuentren instalados los sistemas de alerta. Esta señal es recibida e interpretada por un circuito lógico que se encarga de activar las sirenas de emergencia y los dispositivos luminosos en caso de ser necesario, esto con el fin de alertar a la población.

Para conectar la aplicación con el puerto serie de la PC es necesario utilizar una librería llamada `System.IO.lib` la cual permite de una forma simple definir el puerto por el cual se desea enviar una señal, y de esta forma enviarla de manera sencilla, cabe destacar que esta comunicación es “half dúplex” por lo que solo se envía en un solo sentido, en donde se ha codificado la señal para una transmisión segura.

Ya enviada esta señal será recibida en el punto o los diversos puntos en donde se desee establecer la alarma física, y la misma será activada al recibir los identificadores “AA” seguidos de la palabra “ALERTA” esto debido a que no se quieren crear falsas alarmas si se recibiera un código erróneo.

A continuación se muestra la porción de código necesaria para esta labor.

```

Public Sub Send(ByVal data As String)

    Using com1 As IO.Ports.SerialPort =
My.Computer.Ports.OpenSerialPort("COM1")

        com1.WriteLine(data)

    End Using

End Sub

```

Un ejemplo de cómo utilizar este módulo se muestra a continuación:

```

Dim mensajedealerta As String

Dim indiceriesgo As Double

'calculo de indice de riesgo
'aqui debe ir el modelado matematico

indiceriesgo = VariablesNumeros(0)

If indiceriesgo >= 1 Then

    mensajedealerta = "ALERTA!!!! REVISAR CONDICIONES DE TERRENO
Y ALERTAR A LA POBLACION"
    Call Send("%")
    Call Send("ALERTA")
    Call Send("%")
    Call Send("/n")
    Call AlertaTwitter("AUTOMATICA", mensajedealerta)
    Call Enviarmail("AUTOMATICA", "Este mensaje es enviado
automaticamente desde programa de monitoreo" & vbNewLine & vbNewLine &
"ACTUALIZACION : " & mensajedealerta & vbNewLine & vbNewLine & "Saludos
cordiales," & vbNewLine & "Monitoreo automatico Juco")

Else

    mensajedealerta = "No hay alertas reportadas a esta hora"

End If

```



## 5.5 Notificaciones automáticas.

Para complementar el monitoreo de las variables se ha creado una opción el software de notificaciones automáticas personalizables las cuales permiten monitorear y calcular el índice de riesgo en intervalos de tiempo establecidos por el usuario esto con el fin de tener puntos de control en cuanto a las condiciones del terreno. En la siguiente imagen se puede observar la interfaz desarrollada para esta parte.

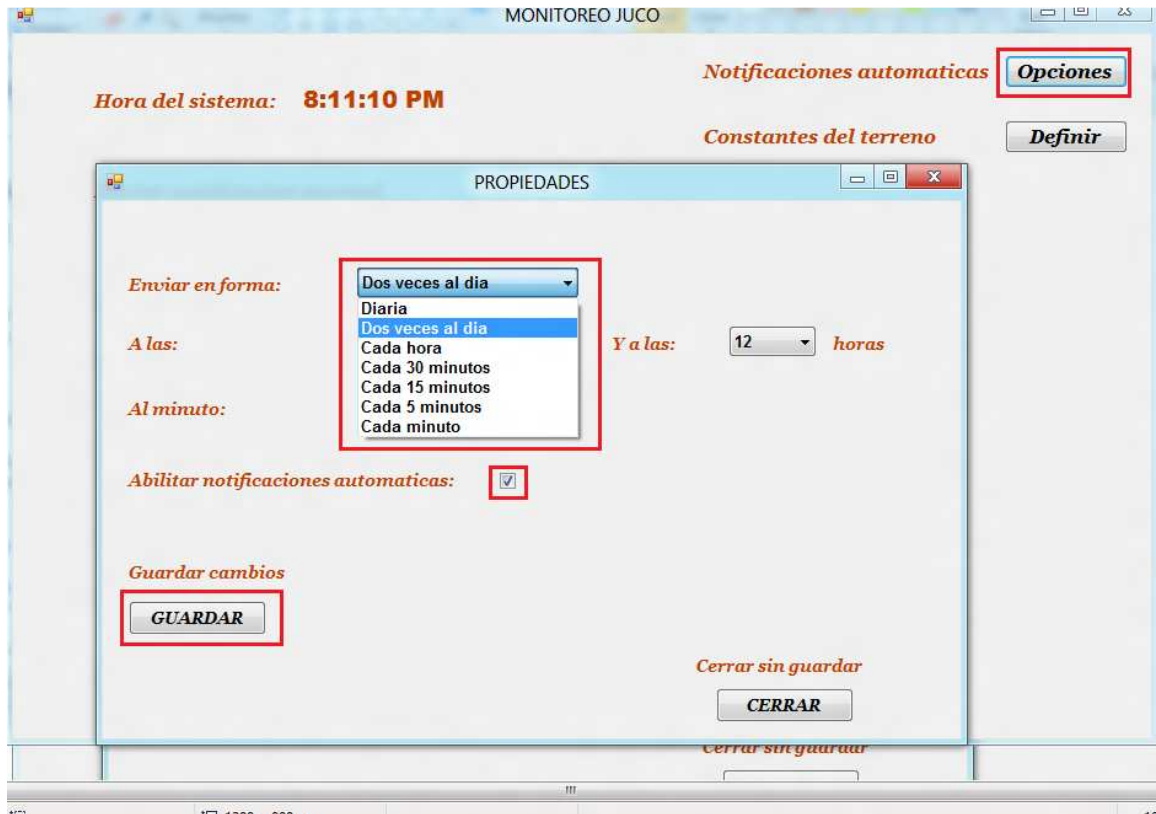


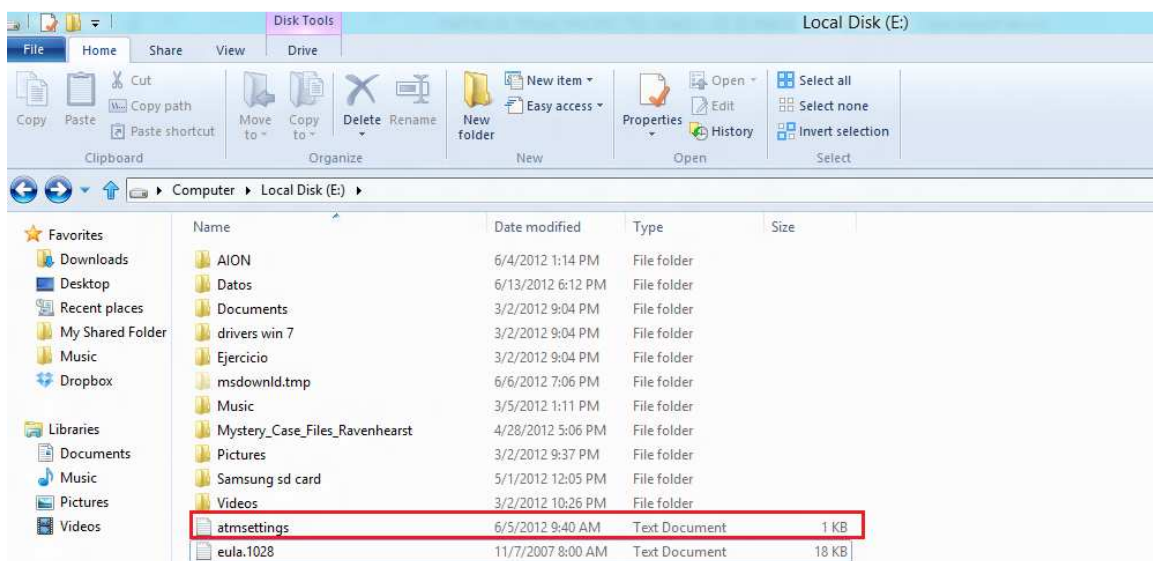
Figura 0.7 Manipulación de variables para la simulación de condiciones.

En la figura anterior podemos observar diversos aspectos del software mencionado, uno de ellos es que las características o propiedades de estas notificaciones son almacenadas en un archivo y pre cargadas al iniciar el

programa, esto para que las alertas se ejecuten de acuerdo a lo establecido la ultima vez que se ejecutó el programa, el mismo posee diversas opciones de calendarización como lo son ejecución diaria en donde se permite elegir la hora y el minuto al cual se debe enviar la notificación, dos veces al día, en donde igualmente son personalizables las horas y los minutos, cada hora, cada media hora, cada 15 minutos, cada 5 minutos, y cada minuto.

Así mismo se pueden desactivar o activar las notificaciones automáticas y también guardar o salir sin guardar los cambios.

Como ya se menciono las características de estas notificaciones son guardadas y leídas desde un archivo llamado "E:\atmsettings.txt" que es igualmente un archivo de texto en formato CSV (archivo separado por comas) la ubicación de este archivo se muestra a continuación.



**Figura 0.8** Ubicación del archivo de texto del cual se extraen los propiedades de notificaciones automáticas.

Para la ejecución en intervalos establecidos de las notificaciones se utilizan timers establecidos para tal fin con los tiempos deseados. Parte del código utilizado para la manipulación y envío de las alarmas automáticas se utilizan los siguientes fragmentos de código.

```
'#####  
    'Comienza lectura del archivo de setting de notificaciones  
  
    Using MyReader As New  
Microsoft.VisualBasic.FileIO.TextFieldParser("E:\atmsettings.txt")  
  
        MyReader.TextFieldType = FileIO.FieldType.Delimited  
        MyReader.SetDelimiters(",")  
  
        Dim currentRow As String()  
  
        Dim i As Integer  
        i = 0  
  
        'leer las propiedades y guardar en variable  
        While Not MyReader.EndOfData  
            Try  
                currentRow = MyReader.ReadFields()  
  
                Dim currentField As String  
  
                For Each currentField In currentRow  
  
                    Variables(i) = currentField  
  
                    i = i + 1  
  
                Next  
  
            Catch ex As  
Microsoft.VisualBasic.FileIO.MalformedLineException  
                MsgBox("Line " & ex.Message & _  
                    "is not valid and will be skipped.")  
  
            End Try  
  
        End While
```

Luego de esto se compara con las horas establecidas y si la opción de notificaciones automáticas está configurada se envían las notificaciones, parte del código se muestra a continuación.

```
'comienzan ifs para notificaciones
'#####

If Variables(0) = "Diaria" And Variables(4) = "Enable" Then

    If TimeOfDay.Hour = Variables(1) And TimeOfDay.Minute =
Variables(3) Then

        Call AlertaTwitter("AUTOMATICA", mensajedealerta)
        Call Enviarmail("AUTOMATICA", "Este mensaje es
enviado automaticamente desde programa de monitoreo" & vbNewLine &
vbNewLine & "ACTUALIZACION : " & mensajedealerta & vbNewLine & vbNewLine
& "Saludos cordiales," & vbNewLine & "Monitoreo automatico Juco")

        End If

    End If

'#####

If Variables(0) = "Dos veces al dia" And Variables(4) =
"Enable" Then

    If TimeOfDay.Hour = Variables(1) And TimeOfDay.Minute =
Variables(3) Then

        Call AlertaTwitter("AUTOMATICA", mensajedealerta)
        Call Enviarmail("AUTOMATICA", "Este mensaje es
enviado automaticamente desde programa de monitoreo" & vbNewLine &
vbNewLine & "ACTUALIZACION : " & mensajedealerta & vbNewLine & vbNewLine
& "Saludos cordiales," & vbNewLine & "Monitoreo automatico Juco")

        End If

    If TimeOfDay.Hour = Variables(2) And TimeOfDay.Minute =
Variables(3) Then

        Call AlertaTwitter("AUTOMATICA", mensajedealerta)
        Call Enviarmail("AUTOMATICA", "Este mensaje es
enviado automaticamente desde programa de monitoreo" & vbNewLine &
vbNewLine & "ACTUALIZACION : " & mensajedealerta & vbNewLine & vbNewLine
& "Saludos cordiales," & vbNewLine & "Monitoreo automatico Juco")

        End If

    End If

'#####
```

## **5.6 Simulación de condiciones de catástrofe y pruebas de la implementación.**

Esta parte del proyecto constituye una de las más importantes, ya que en condiciones normales probar la funcionalidad de un sistema diseñado para predecir desastres naturales es una tarea muy difícil, por lo que simular las condiciones es la mejor opción, si se quiere estar seguro de que el sistema responde como se desea.

Además de ello al ser esta aplicación tan crítica y en donde una mala notificación puede desde causar daños económicos muy considerables hasta causar la pérdida de vidas humanas es estrictamente necesario el simular las condiciones estableciendo valores para las variables establecidas, valores que dicho sea de paso deben reflejar condiciones posibles para así poder simular y probar el sistema al máximo.

Para tal fin se ha creado un programa de software específico, que lo que en esencia hace es simular la escritura en el mencionado archivo E:\variables.txt como si lo hiciera la salida de la primera parte de implementación de este proyecto (parte desarrollada por Rodrigo Alfaro).

A continuación se puede observar la figura correspondiente a la interfaz utilizada para manipular las variables físicas a placer.



**Figura 0.9** Manipulación de variables para la simulación de condiciones.

El software visualizado anteriormente al iniciarse se encarga de pre cargar los valores que posee el archivo en su última modificación, como valores predeterminados, para después poder modificar esos valores mediante simples botones y al presionarse el botón guardar se guarda en el archivo mencionado, entonces si los dos software coexisten en utilización el software de monitoreo reconocerá los nuevos valores de las variables y calculará el índice de riesgo con estos desplegando así las alarmas de ser necesario.

Para realizar esta labor de simulación se utiliza el siguiente fragmento de código para leer y pre cargar los valores de las variables.

```

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

    Dim atmosfericas(4) As String
    '#####
    'Comienza lectura del archivo de valores de variables
    atmosfericas

    Using MyReader As New
Microsoft.VisualBasic.FileIO.TextFieldParser("E:\variables.txt")

        MyReader.TextFieldType = FileIO.FieldType.Delimited
        MyReader.SetDelimiters(",")

        Dim a As String
        Dim currentRow As String()
        Dim i As Integer
        i = 0

        While Not MyReader.EndOfData
            Try
                currentRow = MyReader.ReadFields()
                a = MyReader.LineNumber

                Dim currentField As String

                For Each currentField In currentRow
                    'MsgBox(currentField)

                    If a = -1 Then

                        atmosfericas(i) = currentField
                        i = i + 1

                    End If

                Next

            Catch ex As
Microsoft.VisualBasic.FileIO.MalformedLineException
                MsgBox("Line " & ex.Message & _
                    "is not valid and will be skipped.")

            End Try

        End While

        Label6.Text = atmosfericas(0)
        Label7.Text = atmosfericas(1)
        Label8.Text = atmosfericas(2)
        Label9.Text = atmosfericas(3)
        Label10.Text = atmosfericas(4)

    End Using

End Sub

```

Ahora para guardar los valores en el archivo para luego ser leídas por el software de monitoreo se utiliza el siguiente código, cabe destacar que las variables se escriben en un formato CSV para que puedan ser reconocidas por el programa de monitoreo.

```
Public Sub escribir(ByVal dato1 As String, ByVal dato2 As String,
ByVal dato3 As String, ByVal dato4 As String, ByVal dato5 As String)

    'rutina para escribir texto separado por comas en la ultima linea
de archivo .txt

    Dim miArchivolog As String = ("E:\variables.txt")

    Dim sw As System.IO.StreamWriter

    If System.IO.File.Exists(miArchivolog) = False Then

        sw = System.IO.File.CreateText(miArchivolog)

    Else

        sw = System.IO.File.AppendText(miArchivolog)

    End If

    sw.WriteLine(dato1 & "," & dato2 & "," & dato3 & "," & dato4 &
", " & dato5)
    sw.Flush()
    sw.Close()

End Sub
```



## 6. Capítulo 6: Análisis de resultados

### 6.1 Resultados

En parte del capítulo se mostrarán los principales resultados de la implementación realizada, tanto en aspectos de diseño como de funcionamiento para esto se utilizarán en gran medida las notificaciones mediante los diferentes medios ya que estos son la salida en si del sistema.

Las notificaciones son el resultado final de todo el análisis matemático ya que representan al producto de saber si se ha o no alcanzado un nivel de peligro considerable.

#### 6.1.1 Notificaciones vía redes sociales.

La implementación de las notificaciones por medio de redes sociales se pueden observar en las siguientes imágenes, las cuales muestran una serie de actualizaciones realizadas por medio de twitter:



Figura 6.1 Notificaciones vía twitter.

Así mismo se pueden observar las notificaciones vía Facebook.



Figura 6.2 Notificaciones vía Facebook.

También se pueden observar las notificaciones enviadas manualmente, en el siguiente ejemplo se ve como se envía una cadena corta de caracteres y la misma aparece en facebook y en twitter.

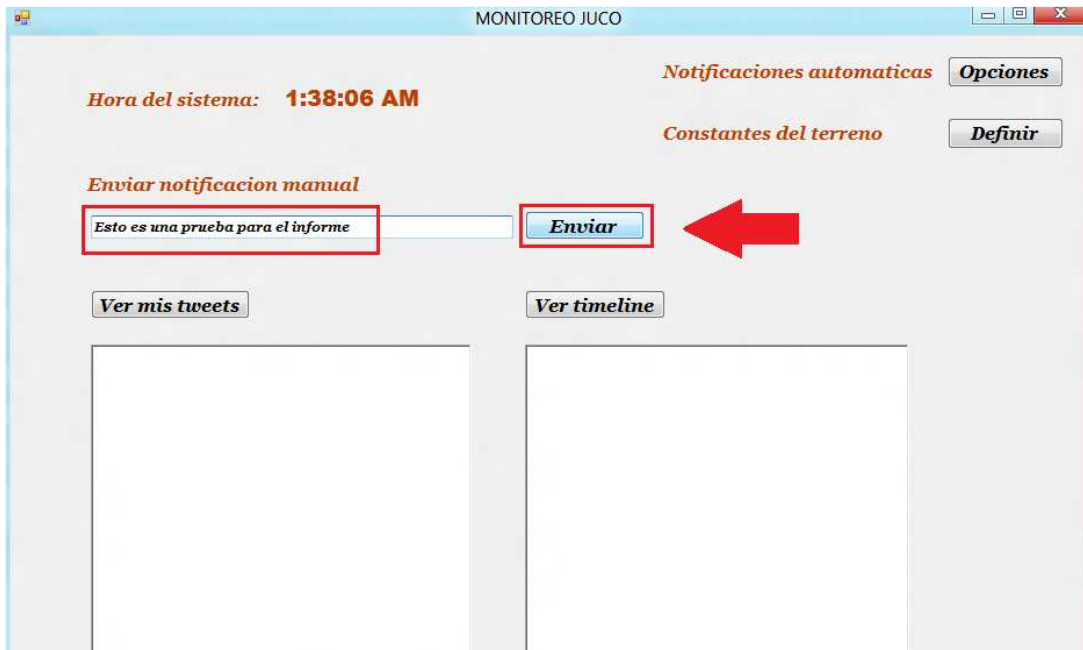
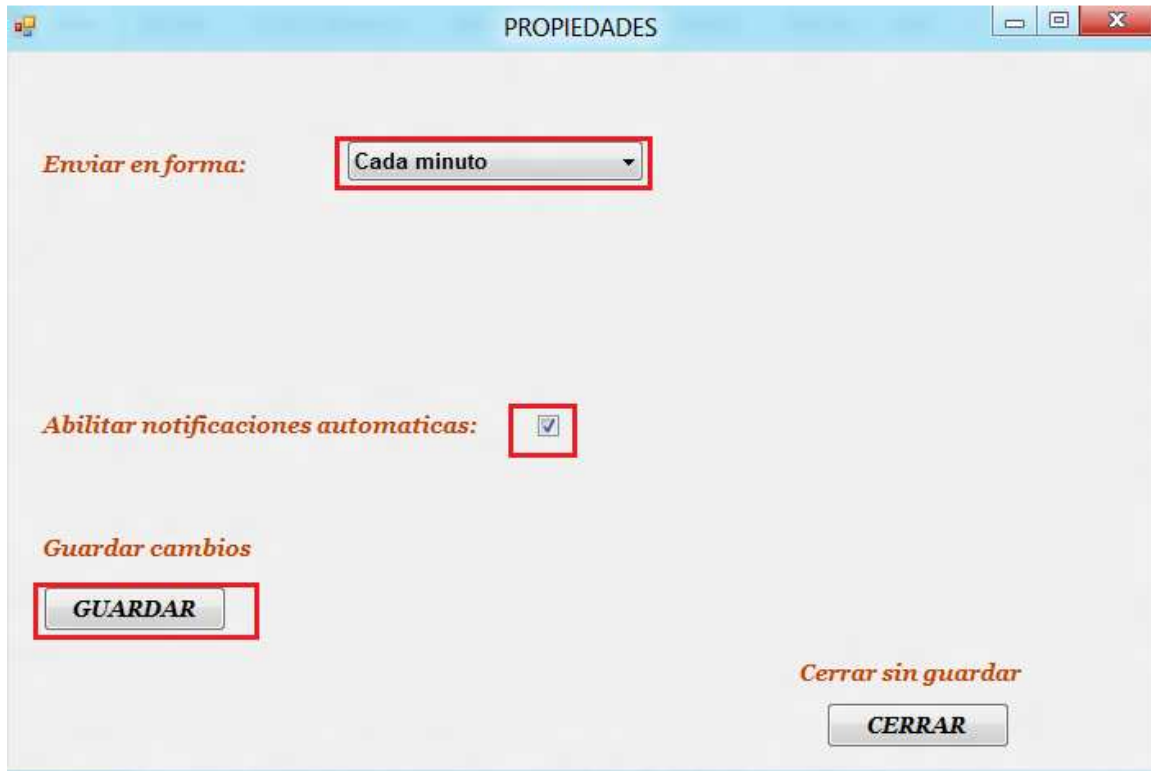


Figura 6.3 Envío de una notificación manual.



Figura 6.4 Visualización de una notificación manual.

Para la parte de notificaciones automáticas los resultados se pueden observar al establecer como periodo de monitoreo y luego se observará después de unos minutos una serie de notificaciones enviadas.



**Figura 6.5** Establecer notificaciones automáticas cada minuto.

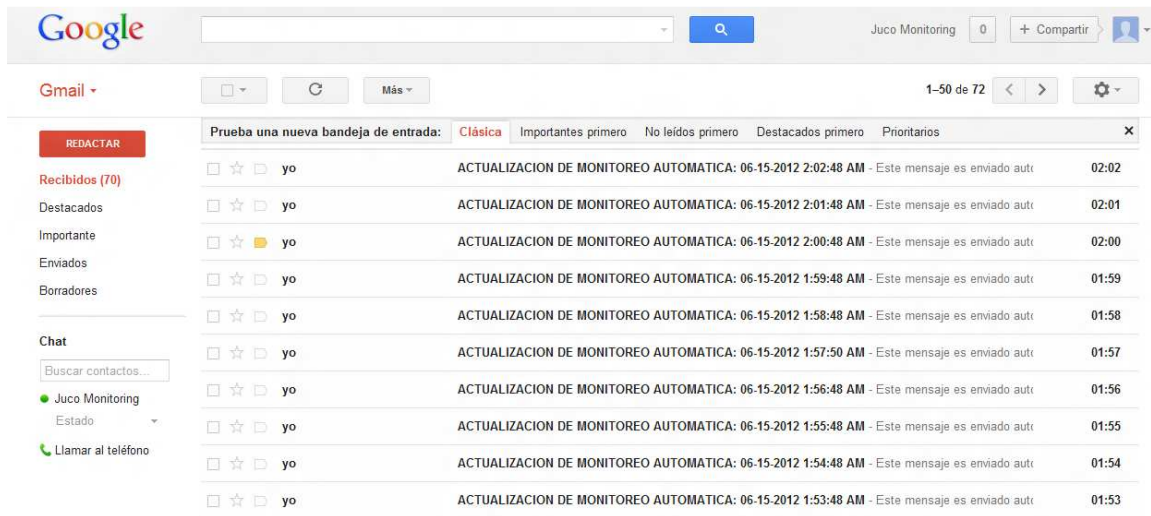
En la siguiente figurase pueden observar las notificaciones hechas cada minuto, esto visualizándolas mediante Facebook.



**Figura 6.6** Visualización de las notificaciones automáticas cada minuto en facebook.

### 6.1.2 Notificaciones vía email.

Para la sección de notificaciones por correo electrónico se podrán observar los resultados al analizar la bandeja de entrada de la cuenta de correo [juco.monitoring@gmail.com](mailto:juco.monitoring@gmail.com) que fue creada para las notificaciones automáticas y esta copiado en las direccion de correo a las que se notifica automáticamente.

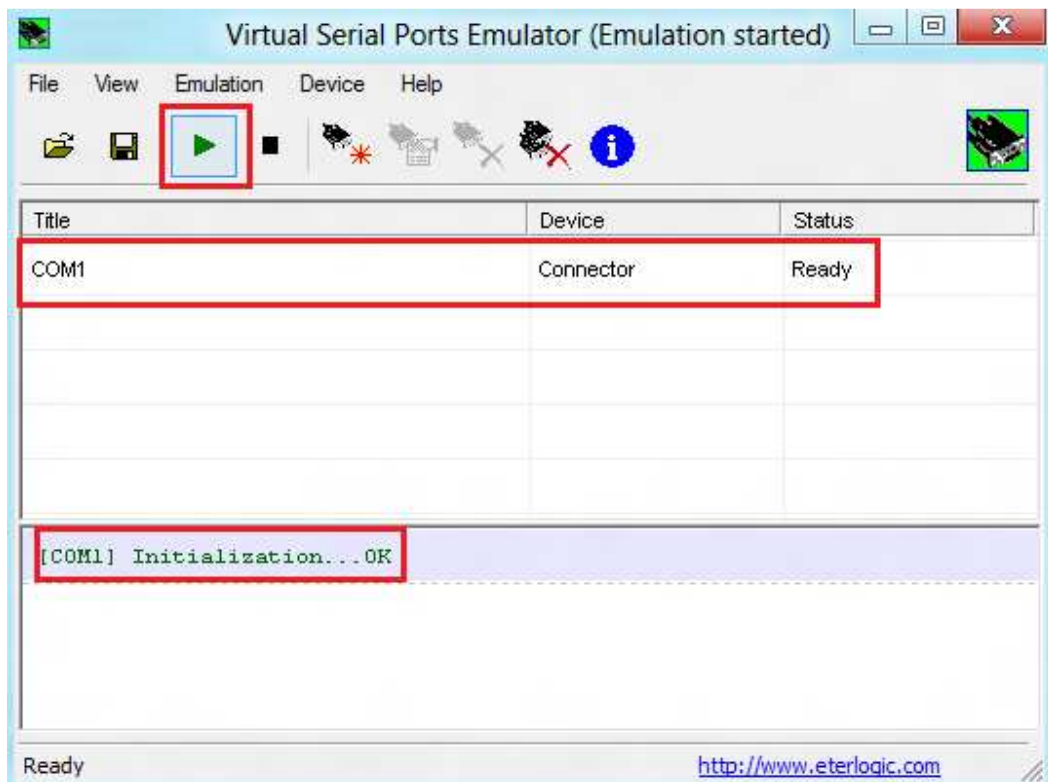


**Figura 6.7** Visualización de las notificaciones por email con alertas automáticas cada.

### 6.1.3 Notificaciones locales mediante envío de datos por puerto serie.

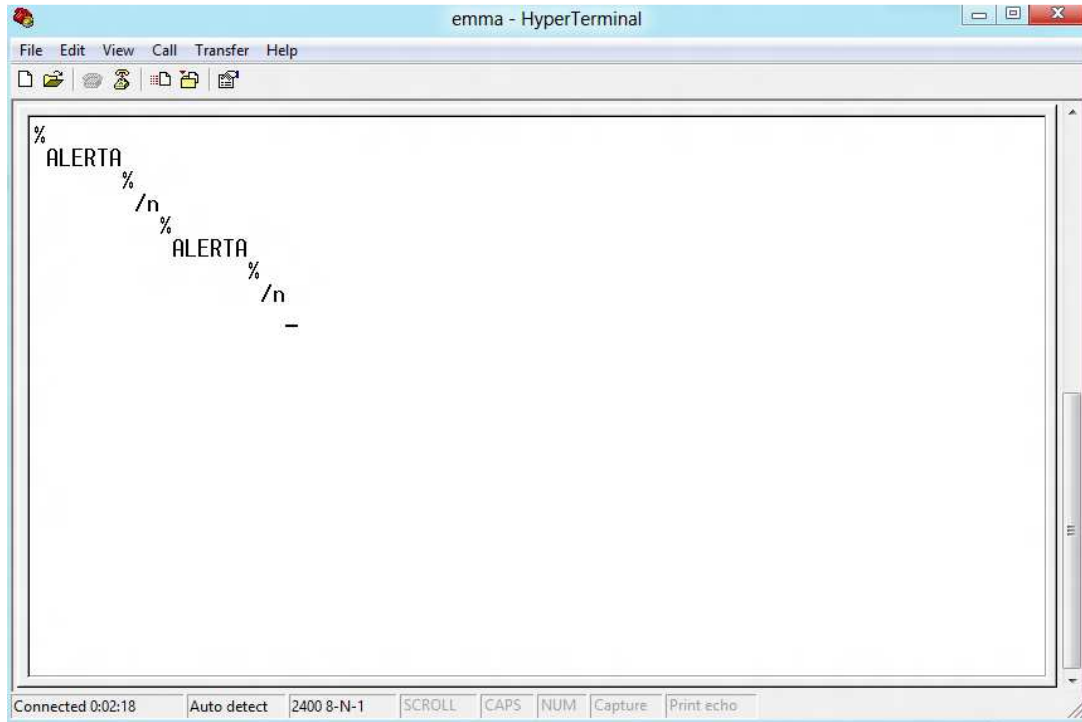
Para comprobar la funcionalidad de esta sección del sistema se utilizaron dos programa distintos, uno llamado “Virtual serial port emulador” que permite simular la existencia de un puerto serie en la PC, llamo COM1 y que es usado por el sistema para identificarlo y enviar mediante el las notificaciones de alerta en caso de que el índice de riesgo este cerca de la unidad.

A continuación podemos ver el funcionamiento de este emulador de puertos.



**Figura 6.8** Emulador de puertos serie.

Además de este software se utiliza el programa llamado hyperterminal por que se puede monitorear el trafico por un determinado puerto serie el mismo se escogerá como el COM1 que se ha creado virtualmente y por el que se envía el mensaje de alerta del sistema.



**Figura 6.9** Comprobar la funcionalidad del envío por puerto serie.

#### **6.1.4 Simulación de condiciones ambientales.**

Con respecto a la simulación de condiciones ambientales, como ya se ha mencionado se desarrollo un segundo sistema de software en donde las variables fueran fácilmente manipulables dicho software se muestra a continuación, así como la salida del archivo de las variables con su formato.





Figura 6.10 Software de manipulación de las variables físicas.

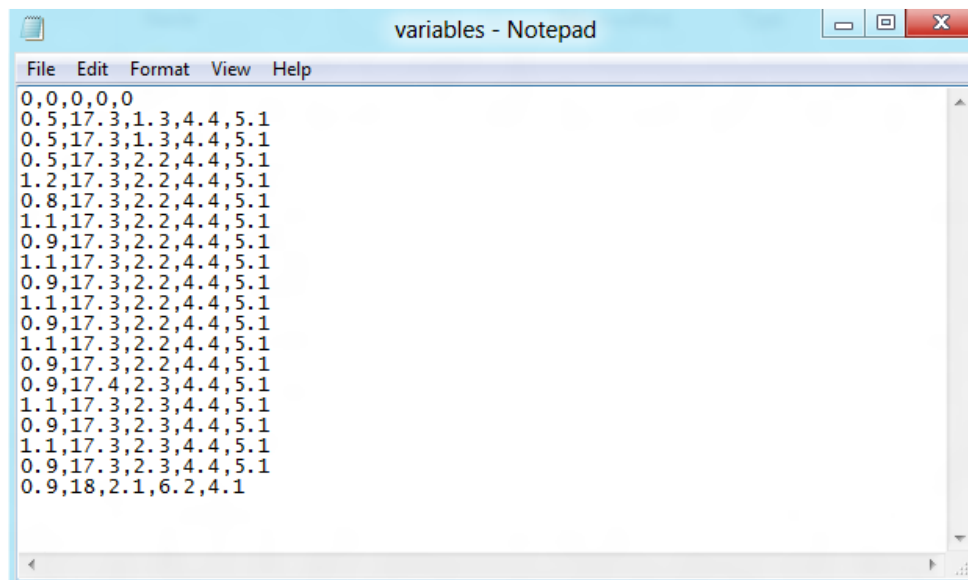


Figura 6.11 Archivo de salida de las variables físicas.

## 6.2 Análisis

Observando la solución planteada en el capítulo 5 así como en la sección 6.1 se evalúan los valores obtenidos a nivel de software en donde se pueden observar las diversas partes diferenciables del proyecto.

Primeramente se pueden observar los resultados en la parte de monitoreo continuo de las variables en donde se pueden establecer periodos específicos de monitoreo y de notificación de los diversos medios, con esta parte del software se ha logrado implementar un sistema que pueda dar tranquilidad a la población, ya que si ellos están constantemente recibiendo mensajes que confirmen que las condiciones ambientales están normales y no tienen riesgo de deslizamiento, eso les dará seguridad en sus vidas diarias.

En la sección 6.1.1 se observan los detalles de las notificaciones de redes sociales, estas brindan una comunicación certera y rápida a todas las personas que posean acceso a internet, es bien sabido que las redes sociales son el medio de comunicación más utilizado en estos momentos, y Facebook y twitter son las mayores redes sociales existentes, por lo que con este medio se está cubriendo la mayor parte de la población.

Además de ello mediante estas mismas redes sociales, se está al alcance y permiten la notificación por otros medios como lo son los teléfonos celulares, ya que, los teléfonos inteligentes que se manejan hoy en día cuentan con sistemas de notificación para las redes sociales y alertan cuando hay un nuevo mensaje o actualización.

Así mismo en la sección 6.1.2 se pueden observar las notificaciones por medio del correo electrónico, estas notificaciones completan la gama de alertas por medio de internet, el correo electrónico es un medio eficaz de comunicación si se mantienen monitoreadas las bandejas de entrada, así mismo si se envía una

notificación por cada evento se pueden mantener una base de datos de las alertas enviada que se pueden consultar en el futuro y para posteriores análisis estadísticos que se consideren pertinentes, al enviar un correos electrónicos se pueden tomar acciones por parte de las autoridades justo en los momentos posteriores al envío de alertas.

Además de esto se sabe que los correos electrónicos representan un medio de comunicación más formal y se pueden catalogar como comunicaciones oficiales, se pueden incluir mensajes más personalizados, como por ejemplo mensajes exclusivos para la comisión nacional de emergencias con los pasos a seguir para los casos de deslizamientos y otros manejos apropiados en estos caso. Por este medio se es capaz de incluir a las personas que se consideren necesarias con información más confidencial de ser el caso.

En la sección 6.1.3 se observan los resultados de una de las partes fundamentales del proyecto como lo son las notificaciones físicas o por medio de dispositivos que logren alertar a la población residente en el área para que comiencen el proceso de evacuación y puedan salir de sus casas en momentos en los que un desastre es inminente, así se agilizará el proceso de evacuación para de esta manera lograr salvar las vidas de los habitantes, lo que al fin y al cabo es el propósito fundamental de este proyecto.

Una parte final del proyecto que se puede analizar es la implementación de las simulaciones de las variables físicas, lo que proporciona una manera fácil de probar el funcionamiento del sistema, como se observa en la imagen 6.11 el archivo de salida del programa es la entrada del programa de monitoreo lo que proporciona una simulación de manera ágil y rápida, que puede ser manipulada por cualquier usuario.

En la parte de la implementación del modelado matemático que fue elegido por el compañero Rodrigo Alfaro en la primera parte de este proyecto se ha logrado

por medio de software el crear un sistema apropiado que permita la predicción de los posibles deslizamientos de tierra que puedan ocurrir. Un aspecto muy interesante a tomar en cuenta es que el modelo depende de muchas variables físicas del terreno por lo que si se desea implementar de una manera adecuada se deben hacer los estudios necesarios.

El sistema de software es capaz de recibir esas variables del terreno y almacenarlas en un archivo, así mismo cargarlas a la hora de abrir el programa, para que si se desea, el mismo software sea implementado en otros ambientes u otros terrenos simplemente se haga un análisis del terreno y los datos sean introducidos al programa de una manera natural.

Si se habla de recomendaciones para este proyecto, es recomendable realizar un análisis estadístico de la salida del modelado, ya que, al ser una aplicación crítica a la hora de la implementación final se debe estar seguro de que esta funciona al 100 por ciento esto para evitar complicaciones así como alarmas innecesarias o tardías.

También se pueden utilizar paneles solares como método alternativo de alimentación de las estaciones meteorológicas, además de utilizar una antena que permita ampliar el rango de comunicación entre la estación meteorológica y la consola de datos.

Una alternativa es validar el modelo de factor de seguridad por medio de un deslizamiento a escala y realizar una comparación con otros tipos de factores de seguridad.

## 7. Capítulo 7: Conclusiones y recomendaciones

### 7.1 Conclusiones

- El modelo de pendiente infinita permite obtener un factor de seguridad que relaciona las características de la zona de estudio con estabilidad de la pendiente en un área determinada
- El lenguaje de programación Visual Basic.net permite desarrollar rutinas simples que implementen el modelado matemático de pendiente infinita para calcular el factor de seguridad en una zona de riesgo de deslizamientos.
- Las redes sociales son un medio de comunicación que permite la rápida y eficiente comunicación a la hora de enviar alertas como en estos caso en los que la rápida acción es fundamental para salvar vidas.
- El correo electrónico permite notificar y almacenar los datos de las comunicaciones enviadas para el monitoreo de variables, y para su posterior análisis estadístico.
- Es necesario implementar un sistema que permita enviar notificaciones físicas como lo son alarmas sonoras y luminosas para alertar a la población en caso de detectarse posibilidades de eventos catastróficos.
- Para las prueba correctas del modelado y el programa de monitoreo se deben emplear programas de software para simular las condiciones ambientales que disparen los sistemas de alarma y así comprobar su funcionalidad.

## 7.2 Recomendaciones

- Utilizar paneles solares como método alternativo de alimentación de las estaciones meteorológicas.
- Validar el modelo de factor de seguridad por medio de un deslizamiento a escala y realizar una comparación con otros tipos de factores de seguridad.
- Comprobar con otros modelados la funcionalidad del sistema.
- Implementar las notificaciones por medio de alarmas y sonidos a gran escala, haciendo pruebas de alcance y percepción por parte de la población.
- Contactar a las autoridades pertinentes para conseguir los contactos necesarios para incluirlos en las notificaciones de correo electrónico.

## 8. Capítulo 8: Bibliografía y anexos

### 8.1 Bibliografía

[1] Arrieta, R. (2011). Proyecto de investigación del sistema de monitorización y alarma contra los deslizamientos etapa 1. Cartago.

[2] Microsoft® Visual Studio. Visual Studio 2010 Express Products. Visual C++ 2010 Express. Consultado 14 abril 2012. Disponible en: <http://www.microsoft.com/visualstudio/en-us/products/2010-editions/express>.

[3] Chinchilla, R. M., & Huertas, G. P. (Mayo de 2004). Documento Digitalizado de la Comisión Nacional de Prevención de Riesgos y Atención de Emergencias de Costa Rica (C.N.E.). Recuperado el 30 de marzo de 2012, de <http://www.cne.go.cr/CEDO-Riesgo/2021.html>

[4] Kloess, G. (2012). Oregon Scientific WMR100/WMR200/RMS300A USB protocol client. Recuperado el 19 de Marzo de 2012, de barnybug/wmr100 · GitHub: <https://github.com/barnybug/wmr100>

[5] Westen, C. v. (2005). Análisis Determinístico para el mapeo de Amenaza por Deslizamientos. Estudio de caso: Manizales, Colombia. Enschede, Holanda.

[6] Software Libre Community. (2012). wview - Linux/FreeBSD/Unix/Mac OSX Weather Station Software. Recuperado el 13 de Marzo de 2012, de <http://www.wviewweather.com/>

## 9. Apéndices y anexos

### 9.1 Apéndices

#### A.1 Glosario, abreviaturas y simbología

**Cohesión efectiva:** atracción intermolecular por el cual los elementos de un cuerpo se mantienen unidas.

**Peso unitario del suelo:** Es el peso, por unidad de volumen, de la parte sólida (partículas) de un suelo.

**Peso unitario del agua:** Es el peso de la unidad de volumen del agua; normalmente igual a 1 gr/cm<sup>3</sup>.



## A.2 Hoja de información del proyecto

### Información del estudiante:

**Nombre:** Emanuel Zúñiga Infante

**Cédula:** 1-1353-0032                      **Carné ITCR:** 200621772

**Dirección de su residencia en época lectiva:** San Pablo de Heredia.

**Dirección de su residencia en época no lectiva:** San Isidro de Pérez Zeledón, San José.

**Teléfono en época lectiva:** 8519-1722

**Teléfono época no lectiva:** 2737-0297

**Email:** [emanuelziga@gmail.com](mailto:emanuelziga@gmail.com)

### Información del proyecto:

**Nombre del Proyecto:** Sistema de redes de sensores para la predicción y prevención de deslizamientos de tierra

**Área del Proyecto:** Programación y Sistemas de Control

### Información de la empresa:

**Nombre:** Instituto Tecnológico de Costa Rica

**Zona:** Cartago

**Dirección:** Basílica de Cartago, 600 metros sur.

**Teléfono:** 2550 - 9184

**Actividad Principal:** Investigación y desarrollo en las áreas de programación y automatización.

### Información del encargado/asesor en la empresa:

**Nombre:** Marvin Hernández Cisneros

**Puesto que ocupa:** Profesor Ing. Electrónica

**Departamento:** Escuela de Electrónica

**Profesión:** Ingeniero Electrónica

**Grado académico:** Licenciatura, Máster

**Teléfono:** 2550 - 9184

**Email:** [mhernandezc@itcr.ac.cr](mailto:mhernandezc@itcr.ac.cr)

### **A.3 Descripción de la institución**

El Instituto Tecnológico de Costa Rica (ITCR), fue creado el 10 de junio de 1971, mediante la Ley No.4777, es una de las cuatro universidades públicas del país, es una institución nacional autónoma de la educación superior universitaria, principalmente se dedica a formar profesionales en diversas áreas, también se dedica a la investigación y la extensión de la tecnología.

### **A.4 Descripción del departamento en la que se realizó el proyecto**

El presente proyecto se desarrolló en el Laboratorio de Sistemas Electrónicos para la Sostenibilidad (SESLab), es un laboratorio que se dedica a la investigación teniendo como objetivo el desarrollo de sistemas electrónicos para potenciar la sostenibilidad de procesos, entornos o sistemas físicos.

En el SESLab trabaja estudiantes de pregrado y postgrado, profesores e ingenieros de diferentes reas las cuales son: Electrónica, Diseño Industrial, Construcción y Ciencias Biológicas.

Actualmente, se desarrollan proyectos de investigación financiados por entes gubernamentales, organizaciones internacionales sin fines de lucro y la industria privada.

SESLab pertenece a la Escuela de Ingeniería en Electrónica y se encuentra localizado en las instalaciones de dicha Escuela, en el Campus Central, Cartago, Costa Rica.

## 10. Formulas

$$\sigma = \frac{W \sin \beta}{A} \quad (0.22)$$

$$s = c + \sigma \tan \phi \quad (0.23)$$

$$R = cA + W \cos \beta \tan \phi \quad (0.24)$$

$$FS = \frac{cA + W \cos \beta \tan \phi}{W \sin \beta} \quad (0.25)$$

$$T = W \sin \beta \quad (0.26)$$

$$T = \gamma Z \sin \beta \quad (0.27)$$

$$P = W \cos \beta \quad (0.28)$$

$$P = \gamma Z \cos \beta \quad (0.29)$$

$$AB = \frac{1}{\cos \beta} \quad (0.30)$$

$$\sigma_n = \frac{P}{AB} \quad (0.31)$$

$$\sigma_n = \frac{\gamma Z \cos \beta}{\frac{1}{\cos \beta}} = \gamma Z \cos^2 \beta \quad (0.32)$$

$$\tau = \frac{T}{AB} \quad (0.33)$$

$$\tau = \frac{\gamma Z \sin \beta}{\frac{1}{\cos \beta}} = \gamma Z \cos \beta \sin \beta \quad (0.34)$$

$$FS = \frac{c' + \sigma_n \tan \phi}{\tau} \quad (0.35)$$

$$FS = \frac{c' + \Upsilon Z \cos^2 \beta \tan \phi}{\Upsilon Z \cos \beta \sin \beta}$$

$$W_w = \Upsilon_w Z m \quad (0.36)$$

$$P_w = W_w \cos \beta \quad (0.37)$$

$$u = \frac{P_w}{AB} \quad (0.38)$$

$$u = \frac{\Upsilon_w Z m \cos \beta}{\frac{1}{\cos \beta}} \quad (0.39)$$

$$u = \Upsilon_w Z m \cos^2 \beta \quad (0.40)$$

$$FS = \frac{c' + (\Upsilon Z \cos \beta - u) \tan \phi}{\Upsilon Z \cos \beta \sin \beta} \quad (0.41)$$

$$FS = \frac{c' + (\Upsilon - m \Upsilon_w) Z \cos^2 \beta \tan \phi}{\Upsilon Z \cos \beta \sin \beta} \quad (0.42)$$