

SOFTWARE DE APLICACIÓN PARA EL ANÁLISIS Y DISEÑO DE
ESTRUCTURAS DE CONTENCIÓN

JORGE ARMANDO FLÓREZ ONOFRE
JAIRO FABIÁN ESPAÑA ENRÍQUEZ

UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA CIVIL
SAN JUAN DE PASTO
2015

SOFTWARE DE APLICACIÓN PARA EL ANÁLISIS Y DISEÑO DE
ESTRUCTURAS DE CONTENCIÓN

JORGE ARMANDO FLÓREZ ONOFRE
JAIRO FABIÁN ESPAÑA ENRÍQUEZ

TRABAJO DE GRADO EN LA
MODALIDAD DE INVESTIGACIÓN

Director
JAVIER OSWALDO MORENO MESÍAS I. C. ESP.

UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA CIVIL
SAN JUAN DE PASTO
2015

NOTA DE RESPONSABILIDAD

“las ideas y conclusiones aportadas en la tesis de grado, son responsabilidad exclusiva del autor”.

Artículo 1 del acuerdo No. 324 de octubre 11 de 1966, emanado del honorable Consejo Directivo de la Universidad de Nariño.

“La Universidad de Nariño no se hace responsable de las opiniones o resultados obtenidos en el presente trabajo y para su publicación priman las normas sobre el derecho de autor”.

Artículo 13, Acuerdo N. 005 de 2010 emanado del Honorable Consejo Académico.

NOTA DE ACEPTACIÓN

JAVIER OSWALDO MORENO MESÍAS
Director tesis.

VICENTE PARRA SANTACRUZ.
Jurado.

JESÚS INSUASTY PORTILLA.
Jurado.

RESUMEN

Las estructuras de contención son un elemento esencial en el desarrollo de obras civiles, debido a la geometría de la pared tiene la capacidad de resistir los esfuerzos de empuje del suelo de relleno, brindando protección a las construcciones contiguas permitiendo el desarrollo de proyectos de ingeniería. El procedimiento para calcular estas obras de contención requiere una serie de chequeos y comprobaciones los cuales demandan que se ingresen nuevos datos y se revise la validez de los mismos. Con la intención de agilizar y facilitar estos procesos de cálculo hay muchas alternativas para la automatización del diseño, mediante el uso de lenguajes de programación para desarrollar aplicaciones informáticas.

En el presente proyecto de investigación se desarrolla un software denominado ***Trasdós***, utilizando el lenguaje de programación Visual Basic, para el análisis y cálculo de muros de contención, esta herramienta informática aprovecha las ventajas de los computadores modernos para agilizar el proceso de cálculo y facilitar el análisis y manejo de los datos obtenidos de acuerdo a los parámetros de diseño.

ABSTRACT

The retaining structures are an essential element in the development of civil works, due to the wall geometry has the capacity to resist thrust efforts soil of filler, providing protection to adjacent structures allowing the development of engineering projects. The procedure for calculating these works containment requires a series of checks and tests which demand new data are entered and validity reviewed. In an attempt to expedite and facilitate these processes of calculation are many alternatives for design automation, using programming languages to develop computer applications.

In this research project is developed a software called ***Trasdós***, using Visual Basic programming language, for analysis and calculation of retaining walls, this software tool takes advantage of modern computers to speed up the calculation process and facilitate, the analysis and management of data according to the design parameters.

TABLA DE CONTENIDO

Pág.

INTRODUCCIÓN	11
1 MARCO REFERENCIAL.....	17
1.1 MARCO TEÓRICO – CONCEPTUAL	17
1.1.1 Muros de contención.....	17
1.1.1.1 Tipos de muros de contención	18
1.1.1.2 Presión activa y pasiva	20
1.1.1.3 Teoría de coulomb	21
1.1.1.3.1 Empuje activo en suelos granulares	23
1.1.1.3.2 Empuje pasivo en suelos granulares	24
1.1.1.4 Método de rankine	26
1.1.1.4.1 Empuje activo en suelos granulares	28
1.1.1.4.2 Empuje pasivo en suelos granulares	28
1.1.1.4.3 Empuje pasivo en suelos cohesivos	28
1.1.1.5 Presiones sísmicas desarrolladas por un suelo contra un muro	29
1.1.1.6 Método de mononobe – okabe	31
1.1.1.6.1 Empuje activo sísmico	33
1.1.1.7 Dimensionamiento de muros de contención	33
1.1.1.8 Análisis de estabilidad de un muro de contención	34
1.1.1.9 Análisis estructural de un muro de contención.....	38
1.1.1.9.1 Método de la resistencia última.....	38
1.1.2 Algoritmo	42
1.1.2.1 Diagramas de flujo	42
1.1.3 Lenguaje unificado de modelado	43
1.1.3.1 Diagramas de clases uml.....	43
1.1.3.2 Diagramas de casos de uso de uml	46
1.1.3.3 Diagramas de secuencia uml:.....	49
1.1.4 El proceso unificado de desarrollo	51
1.1.5 El proceso unificado está dirigido por casos de uso	52
1.1.6 El proceso unificado está centrado en la arquitectura.....	52
1.1.7 El proceso unificado es iterativo e incremental	53
1.1.8 La vida del proceso unificado.....	53
1.1.9 El producto.....	53
1.1.10 Fases dentro de un ciclo	54
1.1.11 La captura de casos de uso	55

1.1.12	El modelo de casos de uso representa los requisitos funcionales	55
1.1.13	Análisis, diseño e implementación para realizar los casos de uso.....	56
1.1.14	Creación del modelo de análisis a partir de los casos de uso.....	56
1.1.15	¿Qué es una arquitectura?	56
1.1.15.1	¿Cómo se obtiene?.....	56
1.1.15.2	¿Cómo se describe?.....	56
1.1.16	Análisis.....	57
1.1.16.1	Artefacto: modelo de análisis	57
1.1.17	Diseño.....	57
1.1.17.1	Artefacto: modelo de diseño	58
1.1.18	Implementación.....	58
1.1.18.1	Artefacto: modelo de implementación.....	59
1.1.19	Prueba	59
1.1.19.1	Artefacto: modelo de pruebas	60
1.1.19.2	Artefacto: caso de prueba	60
1.2	MARCO JURÍDICO.....	61
2	METODOLOGÍA	62
3	CONCLUSIONES	65
4	RECOMENDACIONES	67
	REFERENCIAS BIBLIOGRAFICAS.....	69

LISTA DE FIGURAS

Pág.

Figura 1	Esquema general de un muro de contención	17
Figura 2	Muro de gravedad.....	18
Figura 3	Muro de semigravedad	19
Figura 4	Muro ménsula o cantiléver.....	19
Figura 5	Muro con contrafuertes	20
Figura 6	Empuje activo	21
Figura 7	Estado de reposo.....	22
Figura 8	Estado pasivo	26
Figura 9	Estado pasivo y estado activo.....	27
Figura 10	Comprobación de la estabilidad externa de muros de contención de tierras.....	31
Figura 11	Momento de volcamiento y momento resistente.....	34
Figura 12	Reacciones muro – suelo de fundación	36
Figura 13	Esquema de diagrama de clases.....	44
Figura 14	Esquema de diagrama de clases.....	45
Figura 15	Esquema de diagrama de clases.....	46
Figura 16	Esquema de diagrama de casos de uso.....	47
Figura 17	Esquema de diagrama de casos de uso.....	48
Figura 18	Esquema de diagrama de secuencia.....	49

GLOSARIO

Alzado o cuerpo: parte del muro que se levanta a partir de los cimientos de este, y que tiene una altura y un grosor determinados en función de la carga a soportar.

Concreto reforzado: Concreto estructural reforzado con no menos de la cantidad mínima de acero de pre esforzado o refuerzo no pre esforzado.

Concreto simple: Concreto estructural sin refuerzo o con menos refuerzo que el mínimo especificado para concreto reforzado.

Diente o dentellón: parte del cimiento que se introduce en el suelo para ofrecer una mayor sujeción.

Esfuerzo: Fuerza por unidad de área.

Factor de reducción de resistencia: También se denomina coeficiente de reducción de resistencia. Coeficiente que multiplica la resistencia nominal para convertirla en resistencia de diseño.

Intradós o proyección delantera: superficie externa del alzado.

Puntera o zarpa: parte de la base del muro (cimiento) que queda debajo del intradós y no introducida bajo el terreno contenido.

Talón: parte del cimiento opuesto a la puntera, quedo por debajo del trasdós y bajo el terreno contenido.

Trasdós o proyección trasera: superficie interna del alzado, está en contacto con el terreno contenido.

INTRODUCCIÓN

En la actualidad, la mayoría de profesiones requieren resultados inmediatos para la solución de problemas específicos, por esta razón, la correlación entre las distintas áreas de conocimiento es fundamental para cumplir con estas exigencias. Para el caso específico de la ingeniería civil, las tareas manuales de cálculo son muy extensas y dependen de muchas variables, en particular, determinar las dimensiones y el acero de refuerzo de una estructura de contención es un proceso, que inicia con asumir datos razonables, luego hay que realizar una serie de tanteos para encontrar los valores adecuados, según el chequeo de los factores de seguridad, son labores que demandan mucho tiempo.

Por lo expresado anteriormente, con la realización de este proyecto se obtendrá un software de aplicación denominado **Trasdós**, para el análisis y diseño de estructuras de contención; de gravedad, de semigravedad, en voladizo, con contrafuerte interno y con contrafuerte externo, fundamentado en la revisión de la estabilidad del muro y su posterior análisis estructural, por medio de la teoría de la última resistencia.

Por lo tanto, se estudiará las teorías clásicas sobre presión de tierras, además de las expresiones y conceptos matemáticos que modelaran el escenario, entre las fuerzas que afectan el equilibrio del muro de contención y las reacciones del muro para contrarrestarlas.

En lo que tiene que ver con la metodología para desarrollar el software se implementará el concepto de casos de uso, debido a que es el método más utilizado en la actualidad para la captura de los requerimientos del software. De acuerdo a lo que promulga el proceso unificado de desarrollo de software, los casos de uso son el punto de partida para utilizar esta metodología, con los requisitos capturados se obtiene un modelo de análisis y diseño que se documenta mediante los diagramas de clases y secuencia. Entonces, se continúa con el modelo de implementación que no es más que el código de programación y un archivo ejecutable con sus documentos de operación. Pero antes de que el usuario pueda disponer del software, es necesario hacer unas pruebas para encontrar errores en los pasos de los casos de uso y en los cálculos que se espera obtener. Con respecto a la arquitectura se desarrolla en paralelo al camino que se tomó desde la captura de requisitos hasta las pruebas del software.

TITULO

SOFTWARE DE APLICACIÓN PARA EL ANÁLISIS Y DISEÑO DE ESTRUCTURAS DE CONTENCIÓN

MODALIDAD

PRODUCCIÓN DE SOFTWARE

ÁREAS

GEOTECNIA

ESTRUCTURAS Y CONSTRUCCIÓN

ALCANCE Y DELIMITACIÓN

- La realización de este proyecto contempla el desarrollo de un software para el análisis y diseño de estructuras de contención de gravedad, semigravedad, voladizo, y con contrafuertes.
- Se busca optimizar los procesos existentes de análisis y diseño de las diferentes obras de contención.
- Para los cálculos estructurales se utilizará el método LRFD.
- Para la elaboración del software se utilizará el método RUP y el lenguaje UML.
- Todos los procedimientos de diseño y cálculo que se realicen en este proyecto estarán fundamentados en la Norma Sismo resistente colombiana (NSR-10).
- Para compilar el código fuente del software, se utilizará el lenguaje de programación Visual Basic.

DEFINICIÓN DEL PROBLEMA

ANTECEDENTES

La necesidad de agilizar los procesos de cálculo en la mayoría de actividades laborales que compete la ingeniería civil, implicó la relación con otras profesiones, como la ingeniería de sistemas o de software.

Con la aparición del computador unas décadas atrás y su imponente utilidad, se ha transformado en una de las herramientas más importantes para el ser humano, porque con ellos se puede hacer desde una tarea tan sencilla como hacer un escrito o algo tan complejo como controlar el viaje de un cohete espacial, es decir, gracias a la efectividad del computador para hacer tareas complejas y repetitivas.

Para el caso específico de la ingeniería civil, la forma en que contribuyen; son los programas de computador o software, debido a que son y serán siempre una herramienta muy versátil. A nivel mundial, especialmente en Europa se han creado programas para los fines planteados. Los más sobresalientes son:

- Fine – Civil Engineering Software: es una compañía checa que desarrolla y distribuye programas geotécnicos y estructurales. Su programa es **GEO5. Software Geotécnico**.
- **GeoStru MDC Software** (Italia): es una empresa que desarrolla software técnico profesional para ingeniería geotécnica, geología, geo-mecánica, hidrología y pruebas en los terrenos.
- Sigma Ingeniería: desarrollo **GeoWalls** un programa para facilitar la resolución de todo tipo de muros de contención.
- En un nivel más artesanal y gracias a funciones de la hoja de cálculo Excel hay muchas programaciones realizadas por ingenieros que son muy buenas pero su practicidad está limitado al contexto de la persona que lo desarrollo.

DESCRIPCIÓN

Gracias a las funciones que ofrecen las estructuras de contención en la retención de cualquier tipo de material, es preciso mejorar el proceso de análisis y diseño de este tipo de estructuras. Teniendo en cuenta que, para determinar las dimensiones

de un muro de contención, se inicia con asumir valores según el criterio y la experiencia del ingeniero, sin embargo, estos valores no son los definitivos.

Para lograr obtener los valores correctos se debe realizar una sucesión de tanteos hasta ajustar los valores asumidos de la estructura, este proceso se debe suspender cuando las dimensiones obtenidas cumplan con los factores de seguridad estipulados.

Con la difícil labor de realizar estos diseños en forma manual, han surgido muchas soluciones como tablas y métodos gráficos; si bien son de mucha utilidad, no brindan soluciones precisas. Por otro lado, en el mundo virtual abundan las programaciones para el diseño de muros de contención, en especial en hojas de cálculo Excel, su aplicación no es muy útil, porque están limitados al contexto de las personas que los realizaron, además, no hay un manual para entender su manejo y funcionamiento. En un nivel más alto existen programas más tecnificados, pero, son muy costosos por el valor de sus licencias, nos son muy conocidos en el medio y no tienen la flexibilidad de adecuarse a las condiciones geográficas y normas colombianas.

Por lo tanto, es preciso *optimizar el proceso de análisis y diseño de muros de contención*, mediante el desarrollo de un software de aplicación en Visual Basic.

FORMULACIÓN

Este proyecto consiste en el estudio de los conceptos teóricos sobre la presión lateral de tierra debido a su influencia en las fuerzas horizontales que tiene que contrarrestar una estructura de contención para su estabilidad; además, hay que tener en cuenta; el peso propio de la estructura, la masa de tierra que sirve de relleno, el nivel freático del agua, las características del suelo de cimentación, la zona de amenaza sísmica y la determinación de la armadura de refuerzo, para conseguir un diseño adecuado de muros de contención.

Posteriormente se identificará los casos de uso que se emplearán como cimiento para el desarrollo del software, porque son la guía para las diferentes actividades del proceso. Los casos de uso representan la función del producto, mientras que la arquitectura es la forma, entonces se debe tener en cuenta las restricciones y riesgos del proyecto para su definición, además debe ser flexible ante los cambios que habrá en el desarrollo del software.

El proceso continúa con la división de la totalidad del proyecto en mini proyectos con el fin de hacer una sucesión de iteraciones, que consisten en el tratamiento de

cada mini proyecto por un flujo de trabajo conformado por: requisitos, análisis, diseño, implementación, prueba e integración. Sin embargo, previo a esto hay que realizar una planificación para la selección de las iteraciones con más riesgos y una evaluación al final de la iteración para verificar el estado de los requisitos.

Esta secuencia de iteraciones conforman las fases del ciclo de vida del RUP que son: Concepción, Elaboración, Construcción y Transición. La efectividad de la metodología RUP se notará en el crecimiento de cada versión de producto obtenida con cada iteración.

JUSTIFICACIÓN

Las vías de comunicación son fundamentales para el desarrollo de un país, pero el relieve colombiano es un gran obstáculo para el diseño y construcción de vías seguras y cómodas, por esta razón las obras de contención gozan de gran demanda por las funciones que cumple en su vida útil, en esencia, como su nombre lo dice; es contener diferentes tipos de materiales, que es ideal para el terreno ondulado y escarpado que predomina en Colombia.

Por lo expresado anteriormente, se pretende desarrollar un software para el análisis y diseño de obras de contención, puesto que ofrece al profesional de la ingeniería civil, múltiples oportunidades de evaluar y comparar sus pretensiones de diseño.

Además se mostrará las ventajas de utilizar las herramientas de programación en la solución de problemas determinados en la vida laboral del ingeniero civil, porque reduce con mucha eficiencia los tiempos de cálculo de este tipo de estructuras, que es un factor muy influyente en las utilidades de una obra civil y aún más para una obra de contención. Por otro lado se obtendrán datos precisos y confiables.

En lo que se refiere a la proporción de acero y concreto en la estructura, el software logrará obtener diseños óptimos, es decir, se evitará conseguir diseños sobre reforzados y subreforzados, beneficio que se observará en los costos y la vida útil de la estructura de contención.

OBJETIVOS

OBJETIVO GENERAL

Desarrollar un software de aplicación con el fin de optimizar el análisis y diseño de estructuras de contención de gravedad, de semigravedad, en voladizo, y con contrafuertes. Mediante el uso del Lenguaje Unificado de Modelado UML y el Proceso Unificado de Desarrollo de Software RUP.

OBJETIVOS ESPECÍFICOS

- ✓ Estudiar las teorías clásicas de Coulomb y Rankine sobre la presión de tierras.
- ✓ Utilizar el método LRFD para el diseño en concreto reforzado de los muros de contención.
- ✓ Utilizar las reglas y recomendaciones del Lenguaje Unificado de Modelado para hacer los diagramas (planos) del software.
- ✓ Identificar los casos de uso y estructurarlos dentro de un modelo de casos de uso.
- ✓ Crear la arquitectura del sistema guiados en los casos de uso, teniendo en cuenta las restricciones del proyecto.
- ✓ Realizar una planificación (inicio proceso iteración), análisis, evaluación (fin proceso iteración) y otras actividades específicas para cada iteración.
- ✓ Aplicar los cinco flujos de trabajo (requisitos, análisis, diseño, implementación y prueba) que indica el RUP para cada iteración.
- ✓ Obtener una herramienta informática capaz de determinar las dimensiones y el acero de refuerzo de una estructura de contención.

1 MARCO REFERENCIAL

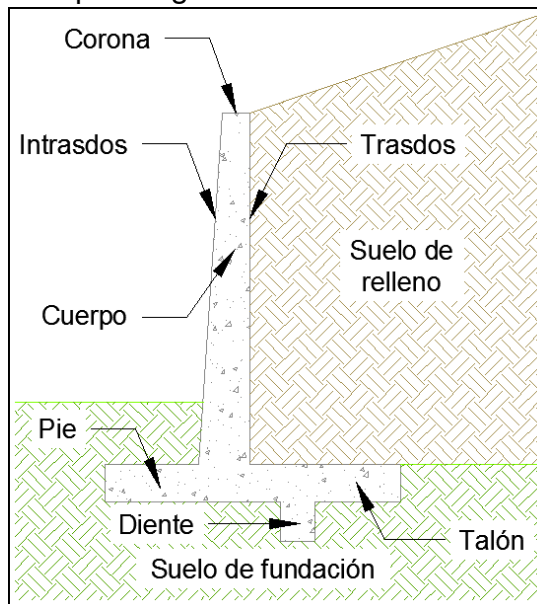
1.1 MARCO TEÓRICO – CONCEPTUAL

1.1.1 Muros de contención. Según Acuña, et al: Se denomina muro de contención a un tipo estructura de contención rígida, destinada a contener algún material, generalmente tierras.

Los muros de contención tienen como finalidad resistir las presiones laterales o el empuje producido por el material retenido detrás de ellos, su estabilidad la deben fundamentalmente al peso propio y al peso del material que está sobre su fundación. Los muros de contención se comportan básicamente como voladizos empotrados en su base.

Designamos con el nombre de empuje, las acciones producidas por las masas que se consideran desprovistas de cohesión, como arenas, gravas, cemento, trigo, etc. En general los empujes son producidos por terrenos naturales, rellenos artificiales o materiales almacenados¹.

Figura 1 Esquema general de un muro de contención



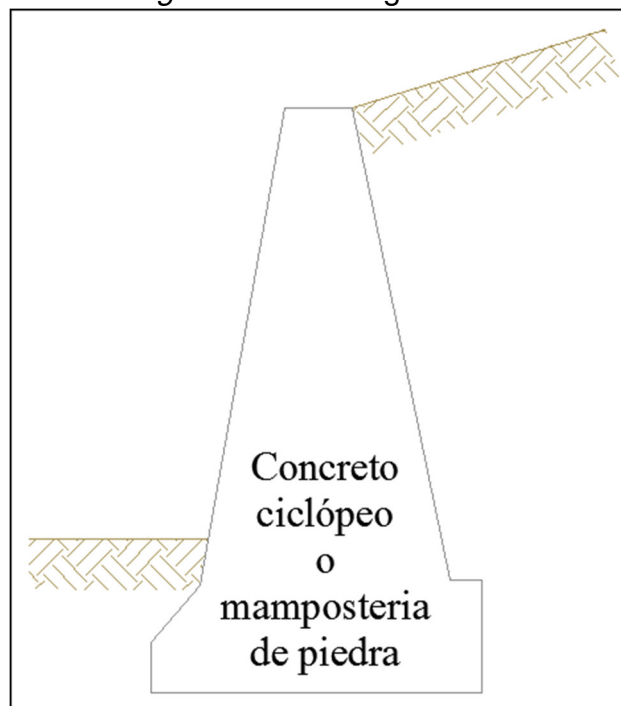
¹ SERVICIO NACIONAL DE APRENDIZAJE. Informe exposición muros de contención. [en línea]. <<http://es.slideshare.net/johannaximena1/muros-de-contencion>> [citado el 18 Abril de 2013]

Tomando el caso más común de un muro de contención, emplearemos las designaciones que se indican en la Figura 1.

1.1.1.1 Tipos de muros de contención

Los *muros de contención de gravedad* (Figura 2) se construyen con concreto ciclópeo o mampostería de piedra, ellos dependen de su propio peso y del suelo que descansa sobre la mampostería para su estabilidad. Este tipo de construcción no es económica para muros altos.

Figura 2 Muro de gravedad



En muchos casos, una pequeña cantidad de acero se usa para la construcción de los muros de gravedad, minimizando así el tamaño de las secciones de los muros, y se conocen generalmente como *muros de contención de semigravedad* (Figura 3).

Los *muros de contención en voladizo* (Figura 4) se construyen de concreto reforzado y consisten en un tallo delgado y una losa de base; son económicos hasta una altura de aproximadamente 8 m.

Figura 3 Muro de semigravedad

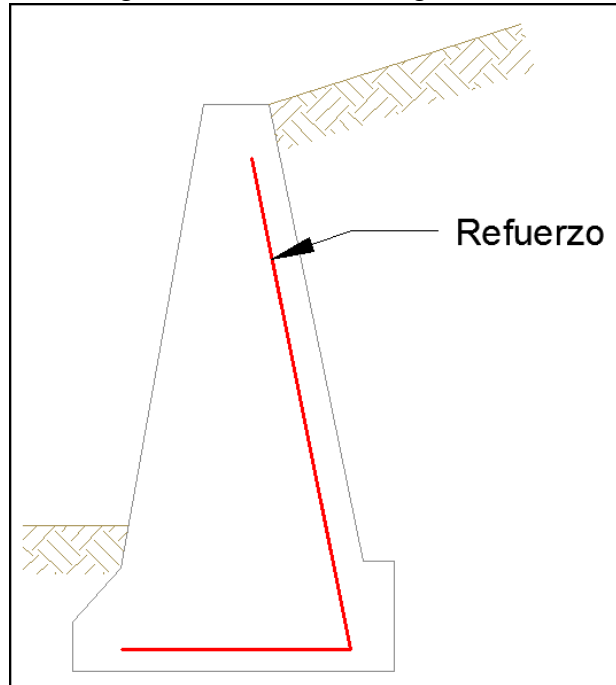
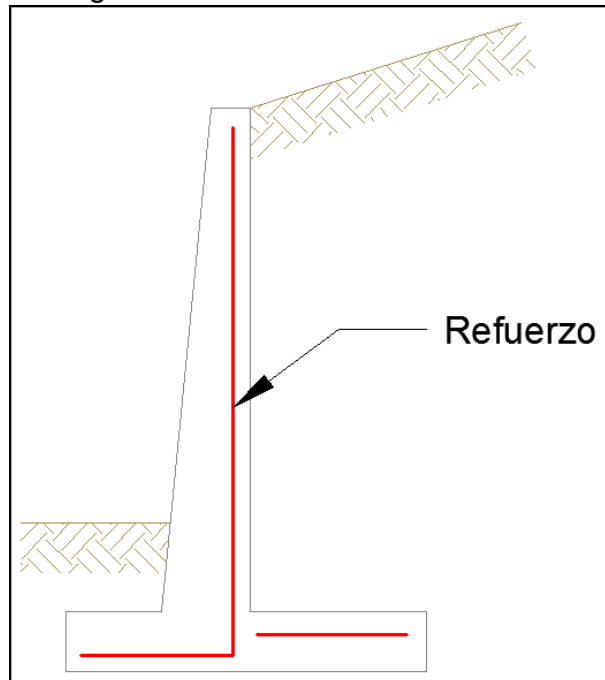


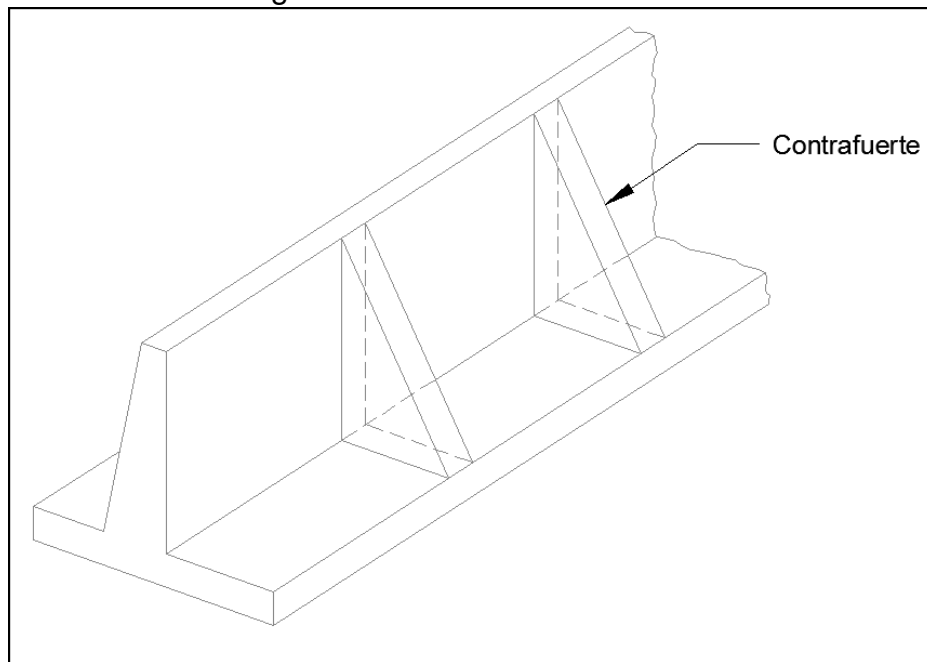
Figura 4 Muro ménsula o cantiléver



Los *muros de contención con contrafuertes* (Figura 5) son similares a los muros en voladizo. Sin embargo, a intervalos regulares tienen losas verticales delgadas de concreto conocidas como contrafuertes que se unen entre sí el muro con la losa de la base.

El propósito de los contrafuertes es reducir las fuerzas cortantes y los momentos flexionantes.

Figura 5 Muro con contrafuertes



1.1.1.2 Presión activa y pasiva

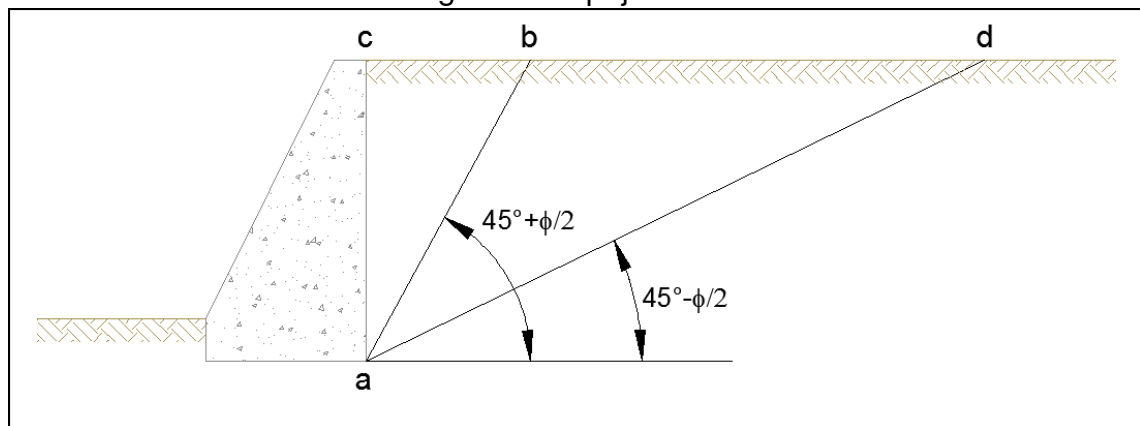
Debido a la presión lateral del suelo, los muros se mueven ligeramente, se deforman, pues entre otras causas, están contruidos de materiales elásticos, y además, están asentados sobre suelos compresibles, a menos que sea roca (y aun así puede haber deformaciones). Por esta razón, los muros de contención se construyen frecuentemente con una leve inclinación hacia el relleno en la cara expuesta.

Si el muro de la Figura 6 se aleja del relleno, se intenta formar un plano de deslizamiento, curvo en la realidad e idealizado como recto ab , con ángulo respecto de la horizontal ($45^\circ + \phi/2$), en la masa del suelo, y la cuña abc que se desliza a lo largo del plano, ejerce una presión contra el muro. El ángulo ϕ se conoce como ángulo de fricción interna, es decir su tangente es igual al coeficiente de fricción inter granular, determinable en forma experimental con ensayos

apropiados. Dado que es el suelo el que hace la presión contra el muro, este caso se denomina *presión de tierra activa*.

Si por el contrario, fuera el muro el que empuja contra el relleno, se formaría un plano idealizado con recta ad , con ángulo respecto de la horizontal ($45^\circ - \phi/2$), en la masa del suelo, y la cuña adc es empujada hacia arriba por el muro, a lo largo de ese plano. En este caso la presión que esta cuña mayor ejerce contra el muro (reacción) se conoce como *presión de tierra pasiva*. Este caso también se presenta en la cara izquierda del muro, cuando el muro cede hacia la izquierda y empuja al suelo de relleno allí ubicado.

Figura 6 Empuje activo



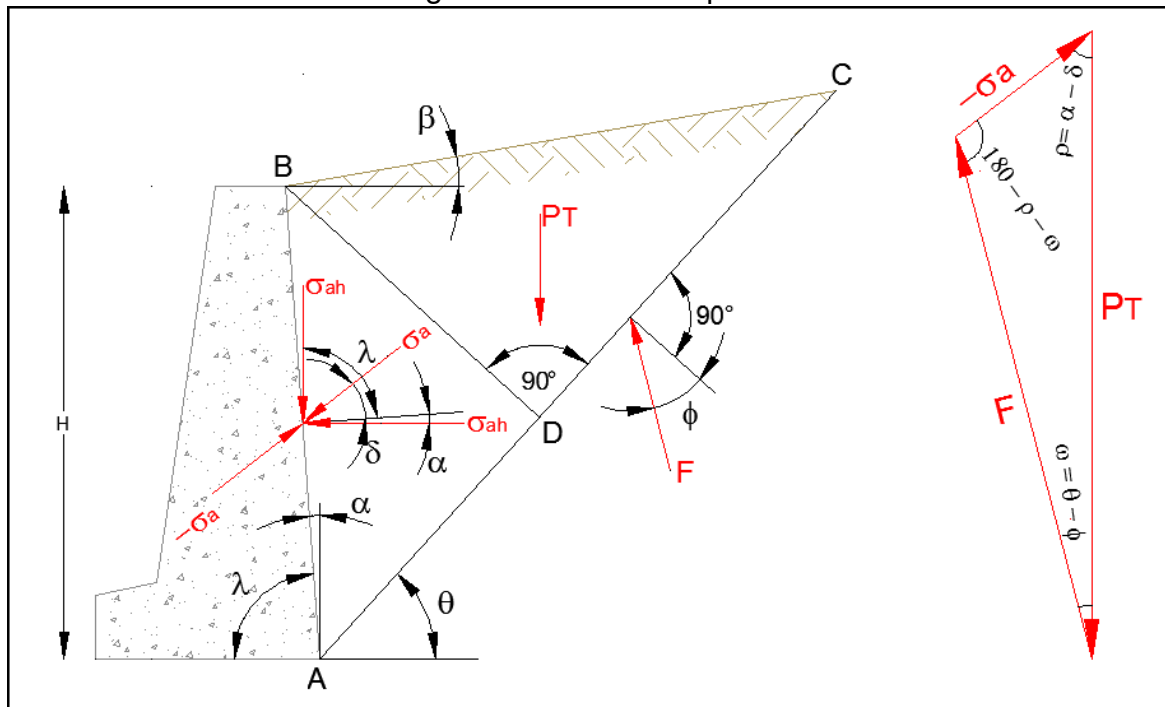
1.1.1.3 Teoría de coulomb

Uno de los primeros métodos para calcular las presiones del terreno sobre un muro rígido, fue propuesto en 1776, por el ingeniero militar francés Charles-Agustín de Coulomb observó que en los muros reventados por la artillería, el trasdós se derrumbaba siguiendo siempre una forma inclinada más bien plana, (Figura 7); en base a esto propuso un modelo de estimación de los empujes del terreno planteando el equilibrio de la masa del mismo derramado en el trasdós (cuña de rotura). Dicho modelo supone:

1. El terreno detrás del muro es homogéneo, isótropo, de naturaleza granular, sin cohesión y capaz de resistir tensiones de compresión y cortante solamente.
2. La resistencia al cortante ha sido movilizada tanto en la espalda del muro como en la superficie de falla.
3. La superficie de rotura se supone que es plana, pasando por el talón del muro y formando un ángulo θ con la horizontal.

4. El ángulo de rozamiento es constante.
5. Se considera el rozamiento δ entre el trasdós del muro y el relleno.

Figura 7 Estado de reposo



Dónde:

P_T = Peso de la masa de tierra contenida en la cuña.

σ_a = Empuje activo en el trasdós del muro.

$-\sigma_a$ = Empuje del trasdós sobre el terreno (reacción del muro sobre la cuña).

F = Reacción del suelo sobre la cuña.

ϕ = Ángulo de rozamiento interno del muro.

δ = Ángulo de rozamiento muro - terreno.

θ = Angulo que forma la cuña o plano de rotura con la horizontal.

λ = Angulo del trasdós del muro con la horizontal.

α = Angulo del trasdós del muro con la vertical.

En esta situación el muro está sometido básicamente al empuje en el trasdós y a su propio peso.

Bajo estas condiciones el muro tiende a girar alrededor del punto más bajo de su intradós, con lo que el terreno detrás del muro experimenta una expansión que le lleva al estado activo.

La principal deficiencia en la teoría de Coulomb es considerar el suelo ideal y la superficie de rotura plana. Otra desventaja de esta teoría es que aunque se establezca el equilibrio, existen otras soluciones. Aparte de estas objeciones, la teoría de Coulomb da resultados que han sido sancionados en la práctica, por lo que constituye un buen método de cálculo de muros de contención.

La presión activa indica la magnitud de la fuerza que debe desarrollar la masa de relleno para producir el movimiento del muro, consistente en una rotación alrededor de la puntera y/o traslación del mismo.

1.1.1.3.1 Empuje activo en suelos granulares

El empuje activo según Coulomb en suelos granulares se determina a partir del polígono de fuerzas de la Figura 7 y es igual a:

$$\sigma_a = \frac{1}{2} \gamma H^2 K_a$$

Dónde:

σ_a = Empuje activo del suelo [Tn/m]

γ = Peso específico del material contenido [Tn/m³]

H = Altura del muro [m]

K_a = Coeficiente de empuje activo [-]

Para calcular el *coeficiente de empuje activo* se utiliza la siguiente formula:

$$K_a = \frac{\cos^2(\phi - \alpha)}{\cos^2(\alpha) \cdot \cos(\delta + \alpha) \cdot \left[1 + \sqrt{\frac{\sin(\phi + \delta) \cdot \sin(\phi - \beta)}{\cos(\delta + \alpha) \cdot \cos(\beta - \alpha)}} \right]^2}$$

Si suponemos el caso de un muro bien cimentado, el muro tenderá a subir, volcando, al girar en el colapso respecto al punto A (Figura 7). La orientación que define la dirección del empuje activo σ_a dependerá del movimiento que tenga el muro en el proceso de colapso. Dicha orientación, definida según δ , se opone al movimiento del muro mejorando en este caso el equilibrio (reduciendo el momento de vuelco).

Lo contrario ocurre en muros cimentados sobre terrenos más blandos que puedan asentar. En este caso el muro desciende respecto a la cuña de rotura, δ va en sentido contrario al caso anterior (se considera en este caso negativo) y el momento de vuelco aumenta. El valor de δ nunca va a ser mayor que el ángulo de rozamiento interno del terreno; típicamente adoptará valores entre $\phi/3$ y $2\phi/3$. En casos extremos tenemos que en terrenos muy húmedos y superficies de muro muy lisas δ tenderá a valores casi nulos ($\delta = 0$), mientras que en condiciones bien drenadas y superficies de muro muy rugosas δ resultará igual a ϕ , que será su valor máximo posible.

1.1.1.3.2 Empuje pasivo en suelos granulares

El empuje pasivo según Coulomb en suelos granulares se determina a partir del polígono de fuerzas de la Figura 8 y es igual a:

$$\sigma_p = \frac{1}{2} \gamma H^2 K_p$$

Dónde:

σ_a = Empuje pasivo del suelo [Tn/m]

γ = Peso específico del material contenido [Tn/m³]

H = Altura del muro [m]

K_p = Coeficiente de empuje pasivo [-]

Para calcular el *coeficiente de empuje pasivo* se utiliza la siguiente formula:

$$K_p = \frac{\cos^2(\phi + \alpha)}{\cos^2(\alpha) \cdot \cos(\delta - \alpha) \cdot \left[1 - \sqrt{\frac{\sin(\phi + \delta) \cdot \sin(\phi + \beta)}{\cos(\delta - \alpha) \cdot \cos(\beta - \alpha)}} \right]^2}$$

Dónde:

P_T = Peso de la masa de tierra contenida en la cuña.

σ_p = Empuje pasivo en el trasdós del muro.

$-\sigma_p$ = Empuje del trasdós sobre el terreno (reacción del muro sobre la cuña).

F = Reacción del suelo sobre la cuña.

ϕ = Angulo de rozamiento interno del muro.

β = Angulo superficie del terreno con la horizontal.

δ = Angulo de rozamiento muro - terreno.

θ = Angulo que forma la cuña o plano de rotura con la horizontal.

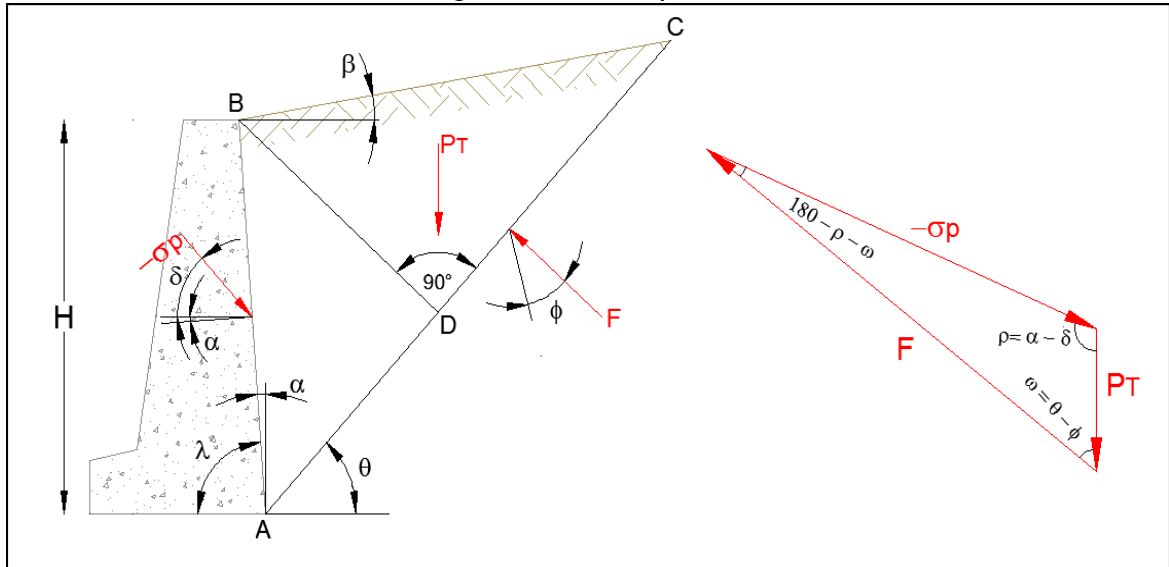
λ = Angulo del trasdós del muro con la horizontal.

α = Angulo del trasdós del muro con la vertical.

El diagrama de intensidad de la presión real de las tierras es muy complejo, por lo que es corriente adoptar una distribución lineal de dicha presión debida a los empujes activo o pasivo. Se supone que la intensidad aumenta con la altura en función del peso del material, de modo que a la presión horizontal de la tierra contra el muro suele llamársela frecuentemente presión de fluido equivalente siendo esta una distribución triangular.

No obstante de ser una aproximación, en el caso de un relleno inclinado tal como se indica en la Figura 8 se considera que la fuerza (F) es paralela a la superficie libre del relleno, encontrándose localizada a un tercio de la altura total del muro tomada a partir de su base, así pueden determinarse fácilmente las componentes horizontal y vertical de dicha fuerza.

Figura 8 Estado pasivo



1.1.1.4 Método de rankine

En 1857, Rankine estableció el equilibrio de un suelo en estado plástico y usó prácticamente las mismas hipótesis que Coulomb. Es decir, supuso que el movimiento producido del muro es suficiente para que el suelo alcance un estado de equilibrio plástico. Por tanto ese supone que todo el terreno tras el muro está en estado de rotura, plastificado completamente.

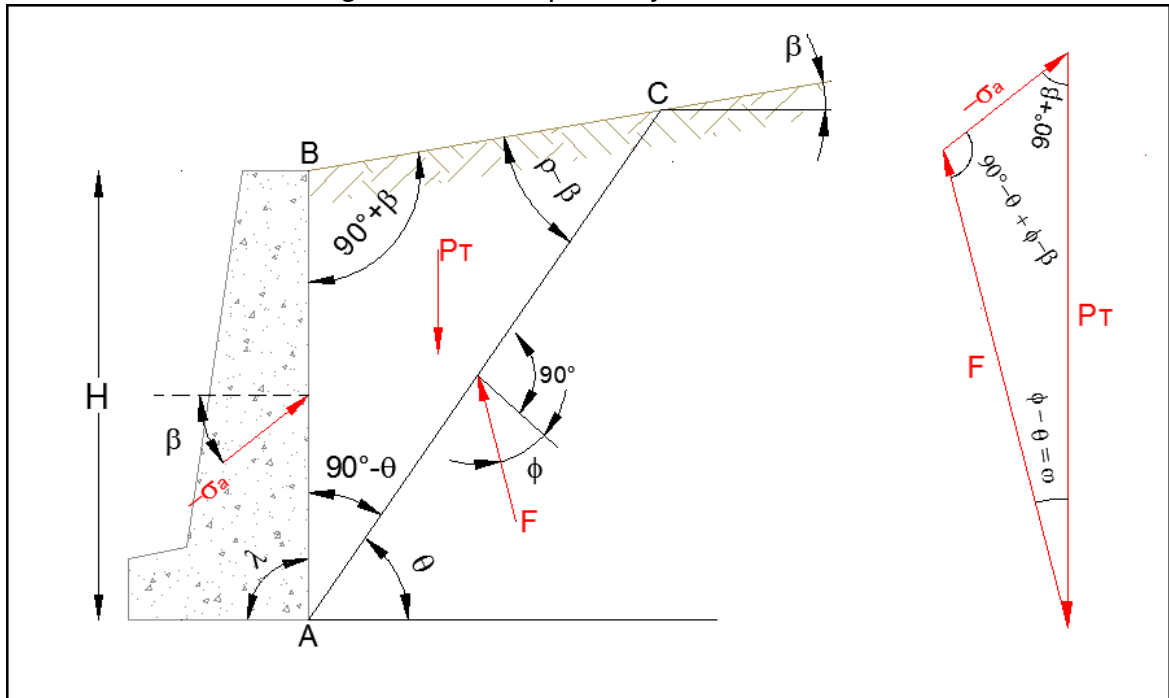
Sin embargo, Rankine consideró que no hay rozamiento entre el terreno y el muro y además que el trasdós de este es vertical.

En este último supuesto si el muro es vertical, el empuje es paralelo a la superficie y forma con la perpendicular al muro el ángulo β del talud del terreno en superficie.

Por tanto, las hipótesis que normalmente se consideran en el método de Rankine son:

1. El trasdós del muro es vertical.
2. No existe rozamiento tierras – muro ($\delta = 0$).
3. La dirección del empuje siempre es paralela a la superficie del terreno.
4. Los planos de rotura siempre forman con el eje principal mayor el ángulo $\left(\frac{\pi}{4} + \frac{\phi}{2}\right)$.

Figura 9 Estado pasivo y estado activo



Dónde:

P_T = Peso de la masa de tierra contenida en la cuña.

$-\sigma_a$ = Empuje del trasdós sobre el terreno (reacción del muro sobre la cuña).

F = Reacción del suelo sobre la cuña.

ϕ = Angulo de rozamiento interno del suelo de relleno.

β = Angulo superficie del terreno con la horizontal.

θ = Angulo que forma la cuña o plano de rotura con la horizontal.

λ = Angulo del trasdós del muro con la horizontal.

En una masa de terreno en *estado de Rankine* todos sus puntos están en situación de rotura (plastificados), es decir, que en cada punto el círculo de Mohr correspondiente a su estado tensiional es tangente a la línea de resistencia, aún siendo más teórico que el método de Coulomb, es también más difícil de aplicar

con geometrías mínimamente complejas es por ello que el método de Rankine se suele usar cuando es fácil de utilizar y, en particular, con superficie del terreno horizontal ($\beta = 0$), muro con trasdós vertical ($\alpha = 0$), y $\delta = 0$.

1.1.1.4.1 Empuje activo en suelos granulares

El empuje activo según Rankine en suelos granulares se determina a partir del polígono de fuerzas de la Figura 9 y es igual a:

$$\sigma_a = \frac{1}{2} \gamma H^2 K_a$$

Dónde:

σ_a = Empuje activo del suelo [Tn/m]

γ = Peso específico del material contenido [Tn/m³]

H = Altura del muro [m]

K_a = Coeficiente de empuje activo [-]

Para calcular el *coeficiente de empuje activo* K_a se utiliza la siguiente formula:

$$K_a = \cos \beta \cdot \frac{\cos \beta - \sqrt{\cos^2 \beta - \cos^2 \phi}}{\cos \beta + \sqrt{\cos^2 \beta - \cos^2 \phi}}$$

Dónde:

ϕ = Angulo de rozamiento interno del suelo de relleno.

β = Angulo superficie del terreno con la horizontal.

1.1.1.4.2 Empuje pasivo en suelos granulares

Para el cálculo del *coeficiente de empuje pasivo* K_p se utiliza la siguiente formula:

$$K_p = \cos \beta \cdot \frac{\cos \beta + \sqrt{\cos^2 \beta - \cos^2 \phi}}{\cos \beta - \sqrt{\cos^2 \beta - \cos^2 \phi}}$$

1.1.1.4.3 Empuje pasivo en suelos cohesivos

Tanto Rankine como Coulomb presentan teorías para calcular las presiones pasivas. La teoría de Rankine generalmente subvalora la presión pasiva, porque

en la mayoría de los casos la dirección de la fuerza pasiva es asumida en forma incorrecta, en cambio la teoría de Coulomb sobre estima la presión pasiva, debido al error de asumir un incremento muy rápido con el aumento de valores de δ .

La presión pasiva está dada por la relación:

$$\sigma_p = \frac{1}{2} \cdot \gamma \cdot h^2 \cdot K_p + 2 \cdot h \cdot C \cdot \sqrt{K_p}$$

Donde:

σ_p = Empuje pasivo del suelo [Tn/m]

γ = Peso específico del material contenido [Tn/m³]

h = Altura del muro [m]

C = Cohesión del material contenido [Tn/m²]

K_p = Coeficiente de empuje pasivo [-]

Para calcular el *coeficiente de empuje pasivo* K_p , se utiliza la teoría de Coulomb o Rankine.

1.1.1.5 Presiones sísmicas desarrolladas por un suelo contra un muro

Los movimientos y las presiones de un suelo contra un muro dependen de varios factores: la naturaleza del suelo bajo el muro, la respuesta del terreno en el trasdós, las características inerciales del muro y el tipo de movimientos aplicados al terreno.

En el caso de acciones sísmicas sobre un muro se tiene que:

1. El muro sometido a acciones sísmicas puede sufrir movimientos de traslación y rotación. La magnitud de estos depende de las características estructurales del muro y de la naturaleza y tipo del terreno.
2. La magnitud y distribución de las presiones sísmicas sobre un muro dependen del tipo de movimiento del muro (traslación, rotación alrededor de la base, flexión de la cabeza del muro).

3. El valor mínimo del empuje se produce cuando el movimiento del muro es alejándose del terreno. El máximo del empuje se produce cuando el movimiento del muro es hacia el terreno del trasdós.
4. La distribución de presiones sobre el muro, cambia con el movimiento de este. El punto de aplicación del empuje se mueve hacia arriba y hacia abajo con el movimiento del muro. La posición mas alta del empuje se produce cuando el muro se ha movido hacia el terreno y las más baja cuando el muro se ha alejado del terreno.
5. Las presiones sísmicas dependen de la respuesta de la pared y del relleno y pueden aumentar significativamente cerca de la frecuencia natural del sistema terreno – muro.
6. Los incrementos de la presión residual pueden permanecer después de un fuerte movimiento sísmico.

Cuando se produce un sismo, tanto el muro como el relleno en su trasdós están sometidos a un movimiento de tipo oscilatorio, pudiendo producirse los siguientes tipos de situaciones:

1. La velocidad del movimiento del muro y de la cuña actúante es la misma: En este caso no se produce interacción adicional entre el muro y la cuña de rotura. Por tanto las presiones del terreno sobre el mismo coinciden con las estáticas que permanecen inalteradas.
2. La pared se mueve alejándose del relleno a mayor velocidad que la cuña del terreno: En este caso la interacción entre el muro y el terreno se reduce y el empuje sobre el muro disminuye en comparación con el empuje activo en condiciones estáticas.
3. La velocidad de la cuña deslizante es mayor que la del muro: En este caso las presiones sobre el muro aumentan y el empuje es mayor que en condiciones estáticas.
4. Que el movimiento del muro se produzca hacia el terreno: En este caso como el terreno no puede moverse en esa dirección por ser un semiespacio elástico, las presiones sobre el muro aumentan también con respecto a la condición estática.

La condición para que se de el mínimo empuje activo o el máximo empuje pasivo es que el muro sufra un movimiento suficiente alejándose o acercándose al terreno para generar dichos valores del empuje.

Estos métodos tienen como objetivo determinar el empuje dinámico y su punto de actuación para una vez determinados, junto con el empuje estático, calcular el empuje total y el momento de vuelco, verificando que los coeficientes de seguridad al vuelco y al deslizamiento estén dentro de los valores admisibles propuestos por la norma NSR-10.

1.1.1.6 Método de mononobe – okabe

Mononobe y Okabe desarrollaron un método pseudoestático para determinar las presiones de un terreno sobre un muro en el caso de la actuación de un terremoto. El procedimiento es una extensión del método de Coulomb para determinar el empuje de un terreno sobre un muro en condiciones sísmicas.

El empuje de suelos pseudo-estático se obtiene entonces a partir del equilibrio de la cuña. Las fuerzas actuando sobre una cuña activa, en el caso de un suelo seco sin cohesión, se muestra en la Figura 10.

Además de las fuerzas que existen bajo condiciones estáticas, sobre la cuña también actúan fuerzas pseudo-estáticas horizontales y verticales cuyas magnitudes están relacionadas con la masa de la cuña mediante aceleraciones pseudo-estáticas:

$$a_h = k_h \cdot g \quad y \quad a_v = k_v \cdot g$$

Donde g es la aceleración de la gravedad.

Si la pared del muro sufre un movimiento suficiente, las presiones del terreno sobre el muro se reducen hasta que alcanza un punto crítico en el que la cuña del terreno se separa del medio y se mueve con el muro. En esta situación se considera que se ha producido el fallo del sistema.

Las condiciones consideradas en el Método de Mononobe – Okabe son:

1. El relleno esta constituido por un material granular con un angulo de rozamiento interno ϕ , sin cohesión ($C = 0$) y su resistencia al corte τ esta definida por la ley de Morh-Coulomb: $\tau = \sigma \tan \phi$.
2. El relleno esta en condiciones secas, de forma que no puede producirse licuefacción.
3. El movimiento del muro hacia el exterior es suficiente para que se separe una cuña de terreno del resto del relleno.
4. El deslizamiento de la cuña de terreno se produce como un solido rígido, con la misma aceleración en todos sus puntos.
5. La superficie de rotura es plana.
6. A lo largo de la superficie de rotura se moviliza completamente la resistencia a cortante.
7. Se considera que la anchura del muro es suficientemente grande para poder despreciar los efectos de borde.

1.1.1.6.1 Empuje activo sísmico

Si la pared esta inclinada con respecto a la vertical el angulo α , el relleno tiene una pendiente definida por el angulo β y el sistema esta sometido a un sismo con componentes de la aceleración $a_h = k_h \cdot g$ y $a_v = k_v \cdot g$, el máximo empuje vale:

$$\sigma_s = \frac{1}{2} \gamma H^2 k_{ad}$$

Para calcular el *coeficiente de empuje activo dinámico* K_{ad} se utiliza la siguiente formula²:

$$K_{ad} = \frac{(1 - k_v) \cos^2(\phi - \alpha - \Psi)}{\cos(\Psi) \cdot \cos^2(\alpha) \cdot \cos(\delta + \alpha + \Psi) \cdot \left[1 + \sqrt{\frac{\sin(\phi + \delta) \cdot \sin(\phi - \beta - \Psi)}{\cos(\delta + \alpha + \Psi) \cdot \cos(\beta - \alpha)}} \right]^2}$$

Donde: $\Psi = \tan^{-1} \left(\frac{k_h}{1 - k_v} \right)$ y $\phi - \beta \geq \Psi$

1.1.1.7 Dimensionamiento de muros de contención

Al diseñar muros de contención, se deben suponer algunas de las dimensiones, llamado esto *predimensionamiento*, para revisar las secciones de prueba por estabilidad. Si la revisión de estabilidad no da buenos resultados, las secciones se cambian y vuelven a revisarse.

La parte superior o corona de cualquier muro de contención no debe ser menor que aproximadamente 0.3 m de ancho para garantizar que los agregados de la mezcla del concreto se acomoden de acuerdo a su tamaño. La profundidad E al fondo de la losa de base debe ser un mínimo de 0.6 m.

Para muros de contención con contrafuertes, la proporción general de la corona y la losa de base es la misma que para muros en voladizo. Sin embargo, las losas de los contrafuertes deben ser aproximadamente 0.3 de espesor y estar espaciadas a distancias centro a centro de 0.3 a 0.7 de la altura del muro de contención.

² CABAÑES JURADO, carlós [On line]. Problemas de interacción suelo estructura en cimentaciones y muros de contención. Influencia de los fenómenos de despegue y deslizamiento. Tesis Doctoral en Ingeniería Civil. Madrid: Universidad Politécnica de Madrid, Escuela técnica superior de ingenieros de caminos, canales y puertos, 2012. 425 p. [citado el 15 de Agosto de 2014] Disponible en < http://oa.upm.es/13731/1/Carlos_Jurado_Cabanes_1.pdf >

1.1.1.8 Análisis de estabilidad de un muro de contención

Para revisar la estabilidad de un muro de contención, se toman los siguientes pasos:

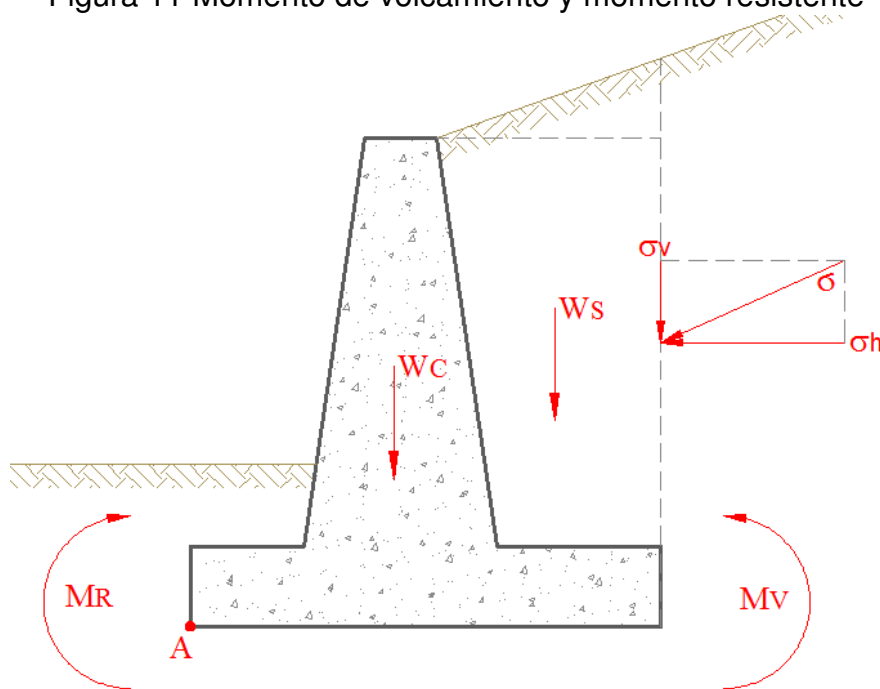
1. Revisión por volteo respecto al pie del muro.
2. Revisión de falla por deslizamiento a lo largo de su base.
3. Revisión de la falla por capacidad de carga de la base.

1.1.1.8.1.1 Revisión por volcamiento

La Figura 11 muestra las fuerzas que actúan sobre un muro de contención, este tiende a voltearse por la arista de la base del pie alrededor del punto A.

El volteo es causado por las componentes horizontales de las fuerzas actuantes. El peso propio del muro W_c , el peso del suelo de relleno sobre el talón W_s y las componentes verticales de las fuerzas actuantes tienden a equilibrar el efecto del momento de volcamiento a provocando un momento resistente o estabilizador.

Figura 11 Momento de volcamiento y momento resistente



El factor de seguridad al volcamiento respecto a la punta del pie, es decir, al punto A en la Figura 11, se expresa como:

$$FSV = \frac{\Sigma M_R}{\Sigma M_V}$$

Donde:

ΣM_R = suma de los momentos de las fuerzas que tienden a volcar la estructura respecto al punto A.

ΣM_V = suma de los momentos de las fuerzas que tienden a resistir el volteo respecto al punto A.

1.1.1.8.1.2 Revisión de falla por deslizamiento a lo largo de su base

El factor de seguridad contra deslizamiento se expresa por la ecuación:

$$FSD = \frac{\mu \cdot W_T \cdot \tan \phi_f + B \cdot C_f + \sigma_p}{\Sigma \sigma_h}$$

Donde:

μ = coeficiente de rozamiento suelo de fundación – base muro.

W_T = peso total muro de contención.

ϕ_f = ángulo de rozamiento interno suelo de fundación.

B = base muro de contención.

C_f = cohesión suelo de fundación.

σ_p = fuerza pasiva.

$\Sigma \sigma_h$ = suma de las fuerzas actúantes horizontales.

En algunos casos ciertos muros de contención no dan un factor de seguridad deseado. Para incrementar su resistencia al deslizamiento se usa un diente en la base. La fuerza pasiva en la punta sin un diente es:

$$\sigma_p = \frac{1}{2} \gamma_f h^2 K_p + 2 C_f h \sqrt{K_p}$$

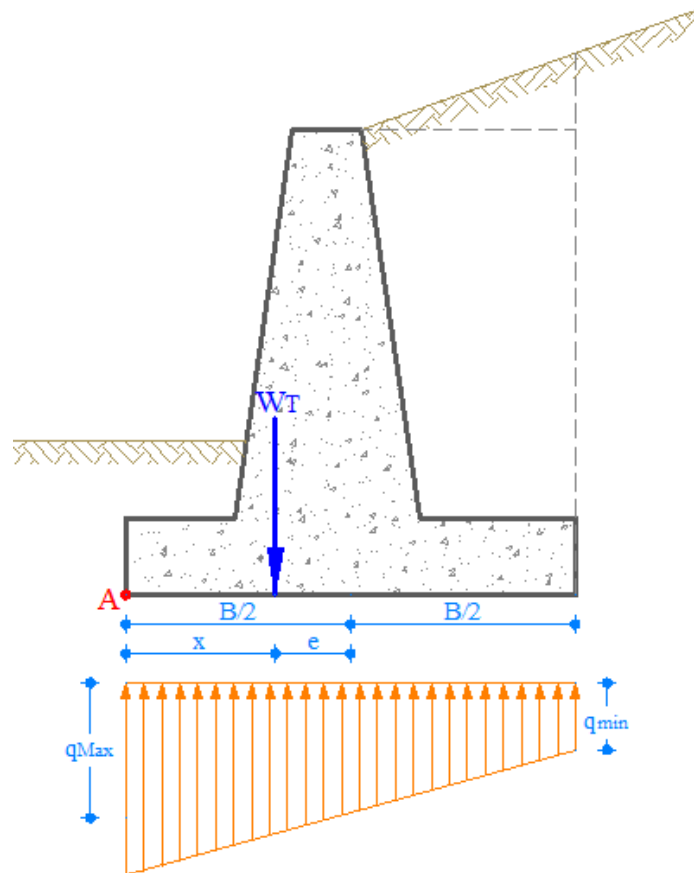
Sin embargo, si se incluye un diente, la fuerza pasiva por unidad de longitud del muro es:

$$\sigma_p = \frac{1}{2} \gamma_f (h + h')^2 K_p + 2C_f (h + h') \sqrt{K_p}$$

Donde h' es la altura del diente y h es la altura del suelo de fundación.

Un diente ayudara obviamente a incrementar la resistencia pasiva en la punta y por tanto el factor de seguridad al deslizamiento. Usualmente, el diente de base se construye debajo del cuerpo del muro y parte del acero principal se lleva dentro del diente.

Figura 12 Reacciones muro – suelo de fundación



1.1.1.8.1.3 Revisión de falla por capacidad de carga

La presión vertical transmitida al suelo por la losa de base del muro de contención debe revisarse contra la capacidad de carga admisible del suelo. La naturaleza de la variación de la presión transmitida por la losa de base al suelo se muestra en la Figura 12. Las presiones máxima y mínima ocurren en los extremos de las secciones pie y talón respectivamente.

La resultante de la sumatoria del peso propio del muro W_c , el peso del suelo sobre el muro W_s y las fuerzas actúantes verticales producen la fuerza normal sobre el suelo de fundación que nunca puede ser mayor a la presión admisible de este, cuyo valor es diferente para cada tipo de suelo y se determina realizando varios ensayos en laboratorio.

Se requiere que la fuerza normal pase por el tercio medio (tercera parte de la base de cimentación), debido a que, si la reacción del terreno pasa por esta zona se logra una distribución de esfuerzos tolerable.

Para determinar si la fuerza normal pasa por el tercio medio se utiliza la ecuación:

$$x = \frac{M_R - M_V}{W_T}$$

Si x cumple con la condición $B/3 \leq x \leq 2B/3$, la fuerza normal pasa por el tercio medio, en caso contrario estaría fuera del tercio medio.

Para calcular los esfuerzos producidos en el suelo de fundación por el muro se debe calcular la excentricidad, que es la distancia entre la mitad de la base de la losa de cimentación y el lugar donde actúa la fuerza normal, se calcula con la siguiente ecuación:

$$e = \frac{B}{2} - x$$

Para que la fuerza normal pase por el tercio medio el valor de e debe ser menor que $B/6$.

Para determinar los esfuerzos ejercidos por el muro de contención en el suelo de fundación se usa la siguiente ecuación:

$$q_{Max} = \frac{W_T}{BL} + \frac{6M}{B^2L}$$

$$q_{min} = \frac{W_T}{BL} - \frac{6M}{B^2L}$$

Donde $M = e \cdot W_T$

Si el análisis de un diseño muestra que $e > B/6$, q_{min} tendrá valores negativos. Entonces, habrá tracciones en la sección extrema del talón. Este esfuerzo no es deseable porque la resistencia a tensión del suelo de fundación es muy pequeña.

Por lo tanto, hay que variar el peso total del muro para cumplir con la condición de que la fuerza normal pase por el tercio medio de la base de cimentación del muro de contención³.

1.1.1.9 Análisis estructural de un muro de contención

Los elementos estructurales de la estructura de contención; pantalla, pie, talón, diente y contrafuerte se diseñaran para tener en todas las secciones una resistencia de diseño mayor o igual a la resistencia requerida, la cual se calculará para cargas y fuerzas mayoradas y según las combinaciones que se estipulen en la norma sismoresistente.

1.1.1.9.1 Método de la resistencia última

Por el método elástico o de los esfuerzos de trabajo, el diseñador obtiene los esfuerzos y deformaciones que se presentan en una estructura sometida a las cargas para las cuales se diseña, suponiendo parámetros elásticos de los materiales.

Por el método de la resistencia ultima, también llamado solamente método de la resistencia, el diseñador podrá estudiar el comportamiento de la estructura en el instante de falla; por tanto, si este instante se hace lo suficientemente mayor que el de su trabajo para las cargas que soporta normalmente, se podrá tener un diseño con factores de seguridad apropiados.

Este trabajo de la estructura en su última resistencia no es posible conocerlo a partir del método elástico, en vista de que el comportamiento de los materiales inelásticos en el instante de falla es diferente al supuesto dentro del periodo elástico de su funcionamiento, es decir, para su trabajo con cargas, esfuerzos y deformaciones admisibles.

Como una introducción al estudio del comportamiento de la estructura en el instante de falla, se anota que los elementos de concreto reforzado sujeto y

³ M DAS, braja. Fundamentos de Ingeniería Geotécnica Sacramento: International Thomson Editores, 2001. 545 p.

diseñado a flexión por cualquier método apropiado, deben fallar cuando el acero a tracción alcanza su límite elástico. En otras palabras, si gradualmente se aumenta la carga hasta que $f_s = f_y$, el elemento resistirá carga adicional en la medida en que se aumenta el brazo del par interior resistente y hasta que la falla definitiva se presente por aplastamiento del concreto a compresión, aunque inicialmente la falla se debió a la fluencia del acero.

1.1.1.9.1.1 Estados límites

Se denomina estado límite de una estructura o elemento estructural cuando llega al límite de su uso programado. Para las estructuras de concreto podemos considerar los siguientes estados límites:

1. Estado límite de falla: corresponde al colapso parcial o total de una estructura con características como la pérdida de equilibrio total o parcial, la rotura del elemento o de los elementos principales o básicos que conduzcan al colapso, la falla progresiva por diferentes motivos, la formación de mecanismos plásticos y la fatiga del material.
2. Estado límite de servicio: corresponde a la interrupción del uso de la estructura, sin que conlleve al colapso, por factores como deflexiones excesivas, fisuras o grietas importantes o vibraciones excesivas.
3. Estados límites especiales: corresponden a daños o fallas debidos a condiciones especiales como movimientos sísmicos anormales, intervención del fuego, explosiones, colisiones de diverso tipo, corrosión, deterioro por factores no contemplados en el diseño.

Para el diseño por estados límites se procede teniendo en cuenta los posibles modos de falla, determinando los factores de seguridad apropiados para cada estado límite y diseñado de acuerdo al estado límite escogido. En consecuencia, se puede identificar el diseño por el denominado Método Elástico o de los Esfuerzos Admisibles, así como el Método de la Resistencia dentro de algunos de los estados límites mencionados.

1.1.1.9.1.2 Hipótesis de diseño

Equilibrio de las fuerzas y compatibilidad de las deformaciones El cálculo de la resistencia de un elemento o de una sección transversal mediante el Método de Diseño por Resistencia exige que se satisfagan dos condiciones básicas: equilibrio estático y compatibilidad de las deformaciones.

La primera condición exige que las fuerzas de compresión y tracción que actúan en la sección transversal para la resistencia "última" estén en equilibrio, mientras que la segunda condición exige que también se satisfaga la compatibilidad entre

las deformaciones del hormigón y de la armadura bajo condiciones "últimas" dentro de las hipótesis de diseño permitidas por la normas NSR - 10.

El término "último" es muy utilizado en relación con el Método de Diseño por Resistencia. Pero es necesario tener en cuenta que la resistencia "nominal" calculada de acuerdo con los requisitos del código puede no necesariamente ser el valor último real. Dentro de las hipótesis de diseño permitidas, se desprecian algunas de las propiedades de los materiales y se establecen otros límites conservadores para utilizar en el diseño práctico.

Esto contribuye a la posibilidad de que la "resistencia última" sea menor que la obtenida mediante ensayos. La resistencia nominal calculada se debería considerar exclusivamente como una resistencia definida por la norma sismo resistente. En consecuencia, al definir la resistencia calculada de un elemento no se utiliza el término "último," sino que se usa el término "nominal."

1.1.1.9.1.3 Condición de deformación balanceada

En una sección transversal existe un condición de deformación balanceada cuando la máxima deformación específica en la fibra comprimida extrema llega a $\epsilon_u = 0,003$ en el mismo instante en que se produce la primera deformación de fluencia $\epsilon_s = \epsilon_y = f_y/E_s$ en la armadura de tracción.

1.1.1.9.1.4 Factores de seguridad

Para acometer el diseño a partir del momento ultimo resistente de una sección, es evidente que se pretende que dicho momento no se presente colocándose un factor de seguridad apropiado.

El reglamento NSR – 10 especifica un factor de seguridad obtenido de un parte, aumentando las cargas o los momentos que ellas producen por medio de los llamados "factores o coeficientes de carga (U)" y por otra parte disminuyendo la resistencia de las secciones por medio de los denominados "factores o coeficientes de reducción de resistencia (ϕ)".

1.1.1.9.1.5 Factor de coeficiente de carga

Es un factor de seguridad con respecto a las cargas actuantes. El titulo B del Reglamento NSR – 10 lo define como "un coeficiente que tiene en cuenta las desviaciones inevitables de las cargas reales con respecto a las cargas nominales y las incertidumbres que se tienen en el análisis estructural al transformar las cargas en efectos internos de los elementos, y por la probabilidad de que más de una carga extrema ocurra simultáneamente".

Una carga mayorada es, por consiguiente, la carga resultante de multiplicar la carga nominal por un factor o coeficiente de carga.

En consecuencia el requisito básico para el diseño por resistencia se expresa así:

$$Resistencia\ de\ diseño \geq Resistencia\ requerida$$

$$\phi \times Resistencia\ nominal \geq U$$

Factor o Coeficiente de Reducción de Resistencia ϕ

Es un coeficiente que reduce la capacidad de la sección para tener en cuenta la probabilidad de existencia de elementos con una resistencia baja debido a variaciones en la resistencia de los materiales y las dimensiones, las aproximaciones en las ecuaciones de diseño que reflejan el grado de ductilidad y confiabilidad requerida para el elemento sometido a la carga que le corresponde y para tener en cuenta la importancia del elemento en la estructura.

Por lo tanto, la resistencia de diseño que tiene un elemento, sus conexiones con otros miembros y cualquier parte o sección de él, en términos de momento flector, carga axial, cortante y torsión, debe ser igual a su resistencia nominal calculada de acuerdo con los requisitos y suposiciones del título C del Reglamento NSR – 10 multiplicada por un coeficiente de reducción de resistencia ϕ :

$$Resistencia\ de\ diseño = \phi Resistencia\ nominal \geq Resistencia\ requerida$$

$$\phi M_n = M_u = U \cdot M$$

M = Momento de flexión o momento actuante

U = Coeficiente de carga

ϕM_n = Momento actuante último o momento de diseño

M_n = Momento de falla a la resistencia última o momento Nominal

ϕ = Coeficiente de reducción⁴.

⁴ SEGURA FRANCO, Jorge. Estructura de concreto I. Bogotá: Universidad Nacional de Colombia. 2011. 594 p.









1.1.2 Algoritmo. Es un conjunto de acciones que determinan la secuencia de los pasos a seguir para resolver un problema específico. Sus pasos deben estar definidos con precisión de forma que no existan ambigüedades que den origen a elegir una opción equivocada. Los algoritmos son finitos; es decir, su ejecución termina en un número determinado de pasos.

Los algoritmos pueden representarse a través de un conjunto de palabras por medio de las cuales se puede representar la lógica de un programa. Este conjunto de palabras constituyen lo que se conoce como pseudocódigo. Además, los algoritmos se pueden representar gráficamente a través de un diagrama de flujo.

1.1.2.1 Diagramas de flujo

Es una representación gráfica de un algoritmo o de una parte del mismo. La ventaja de utilizar un diagrama de flujo es que se le puede construir independientemente del lenguaje de programación, pues al momento de llevarlo a código se puede hacer en cualquier lenguaje.

Tabla 1. Símbolos diagramas de flujo

Símbolo	Nombre	Significado
	Terminal	Representa el inicio y fin de un programa
	Proceso	Son acciones que el programa tiene que realizar
	Decisión	Indica operaciones lógicas o de comparación
	Entrada	Nos permite ingresar datos
	Salida	Es usado para indicar salida de resultados
	Conector	Enlaza dos partes cualesquiera de un programa
	Línea de flujo	Indica dirección de flujo del diagrama. Las flechas de flujo no deben cruzarse. Los diagramas se leen de arriba hacia abajo y de izquierda a derecha
	Conector	Referencia el flujo del diagrama en otra pagina

Dichos diagramas se construyen utilizando ciertos símbolos de uso especial como son rectángulos, óvalos, pequeños círculos, etc.; estos símbolos están conectados

entre sí por flechas conocidas como líneas de flujo. En la Tabla 1 se presentan estos símbolos y su significado.

1.1.3 Lenguaje Unificado de Modelado. Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. EL Lenguaje Unificado de Modelado (LUM o UML, por sus siglas en inglés, Unified Modeling Language); está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico basado en diagramas para la especificación, visualización, construcción y documentación de cualquier sistema complejo (para nuestro caso sistemas de software). UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo.

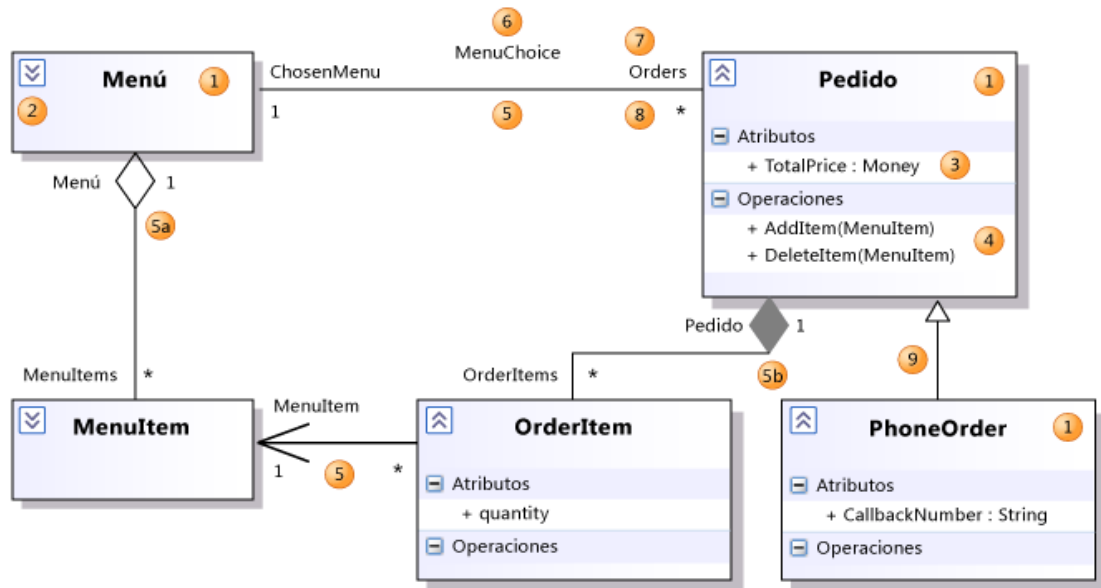
Se puede aplicar en el desarrollo de software gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado Racional o RUP), pero no especifica en sí mismo qué metodología o proceso usar.

UML no puede compararse con la programación estructurada, pues UML significa Lenguaje Unificado de Modelado, no es programación, solo se diagrama la realidad de una utilización en un requerimiento. Mientras que, programación estructurada, es una forma de programar como lo es la orientación a objetos, sin embargo, la programación orientada a objetos viene siendo un complemento perfecto de UML, pero no por eso se toma UML sólo para lenguajes orientados a objetos.

1.1.3.1 Diagramas de clases uml

En los diagramas de clases de UML, se describen el objeto y las estructuras de información que se utilizan en la aplicación, tanto de forma interna como en la comunicación con los usuarios. Esta información se describe sin hacer referencia a ninguna implementación concreta. Las clases y relaciones se pueden implementar de muchas maneras, por ejemplo, en tablas de bases de datos, en nodos XML o en composiciones de objetos de software.

Figura 13 Esquema de diagrama de clases



1 **Clase:** Definición de objetos que comparten ciertas características estructurales y de comportamiento.

1 **Clasificador:** Nombre general de una clase, interfaz o enumeración. Los componentes, casos de uso y actores también son clasificadores.

2 **Control de expandir y contraer:** Si no puede ver los detalles de un clasificador, haga clic en el botón de expansión situado en la parte superior izquierda del clasificador. Es posible que también necesite hacer clic en el icono [+] de cada segmento.

3 **Atributo:** Valor con tipo que se adjunta a cada instancia de un clasificador

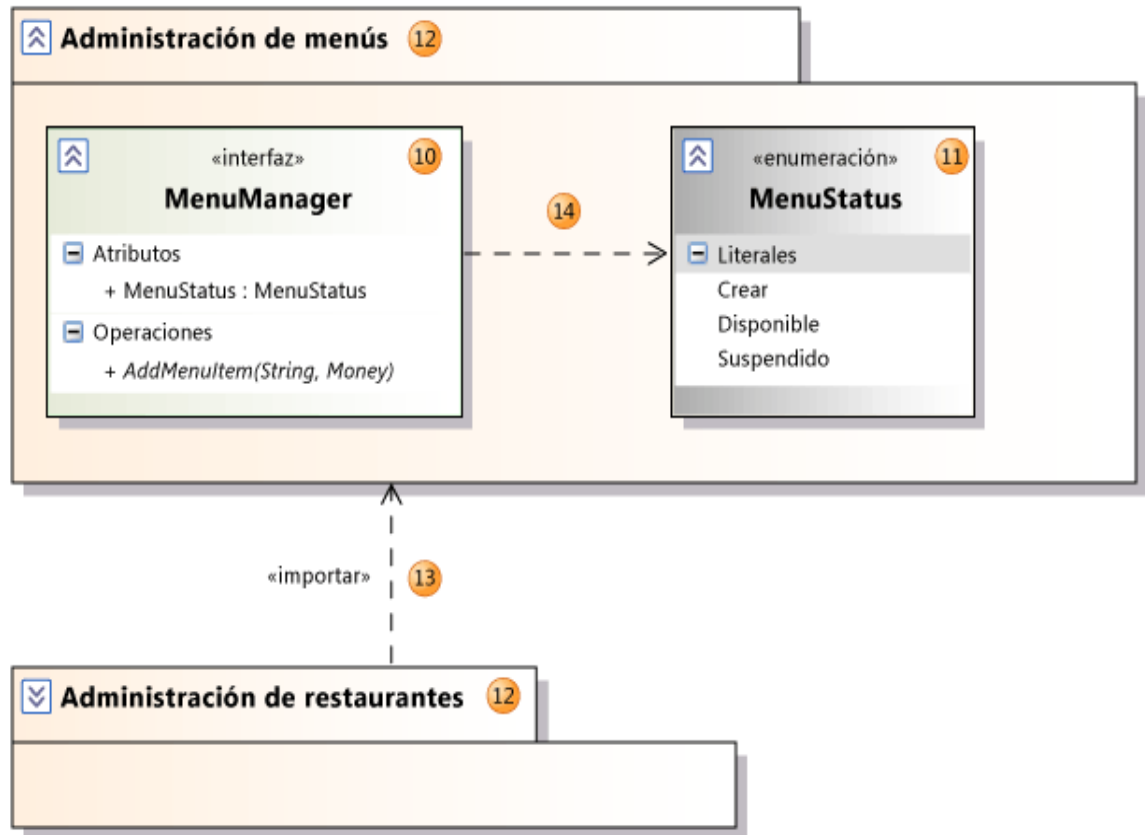
4 **Operación:** Método o función que pueden realizar las instancias de un clasificador.

5 **Asociación:** Relación entre los miembros de dos clasificadores.

5a **Agregación:** Asociación que representa una relación de propiedad compartida. La propiedad Aggregation del rol del propietario se establece en Shared.

5b **Composición:** Asociación que representa una relación parte/todo.

Figura 14 Esquema de diagrama de clases



6 Nombre de asociación: Nombre de una asociación. El nombre puede quedarse vacío.

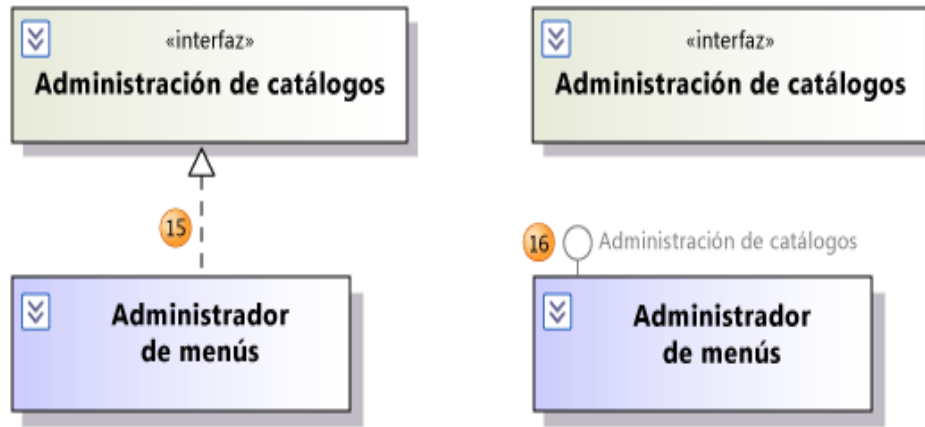
7 Nombre de rol: Nombre de un rol, es decir, un extremo de una asociación. Puede utilizarse para hacer referencia al objeto asociado.

8 Multiplicity: Indica cuántos de los objetos de este extremo se pueden vincular a cada objeto del otro. * significa que no hay límite superior en el número de vínculos que se pueden establecer.

9 Generalización: El clasificador específico hereda parte de su definición del clasificador general. El clasificador general se encuentra en el extremo del conector de la flecha. El clasificador específico hereda los atributos, las asociaciones y las operaciones.

10 Interfaz: Definición de parte del comportamiento de un objeto que puede observarse desde el exterior.

Figura 15 Esquema de diagrama de clases



11 **Enumeración:** Clasificador que se compone de un conjunto de valores literales.

12 **Paquete:** Grupo de clasificadores, asociaciones, acciones, líneas de vida, componentes y paquetes. En un diagrama de clases lógicas se muestra que los paquetes y clasificadores de miembros están incluidos dentro del paquete.

13 **Import:** Relación entre paquetes que indica que un paquete incluye todas las definiciones de otro.

14 **Dependencia:** La definición o implementación del clasificador dependiente podría cambiar si el clasificador situado en el extremo con la punta de flecha se modifica.

15 **Realización:** La clase implementa las operaciones y atributos definidos por la interfaz.

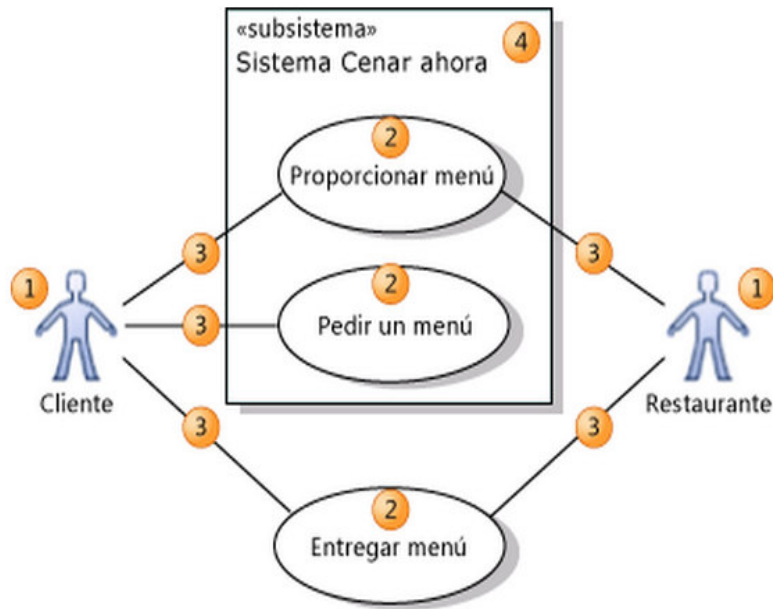
16 **Realización:** Presentación alternativa de la misma relación. La etiqueta del símbolo circular identifica la interfaz.

1.1.3.2 Diagramas de casos de uso de uml

Un diagrama de casos de uso actúa como foco en la descripción de los requisitos del usuario. En él se describen las relaciones entre los requisitos, los usuarios y los componentes primarios. No se describen en detalle los requisitos, ya que esto

puede hacerse en otros diagramas o documentos que pueden vincularse a cada caso de uso.

Figura 16 Esquema de diagrama de casos de uso



1 **Actor:** Representa un usuario, organización o sistema externo que interactúa con la aplicación o sistema. Un actor es una clase de tipo.

2 **Caso de uso:** Representa las acciones que realizan uno o varios actores para conseguir un objetivo determinado. Un caso de uso es una clase de tipo.

3 **Asociación:** Indica que un actor forma parte de un caso de uso.

4 **componente:** Sistema o aplicación (o alguno de sus componentes) en el que se está trabajando. Puede abarcar desde una red grande hasta una única clase de una aplicación.

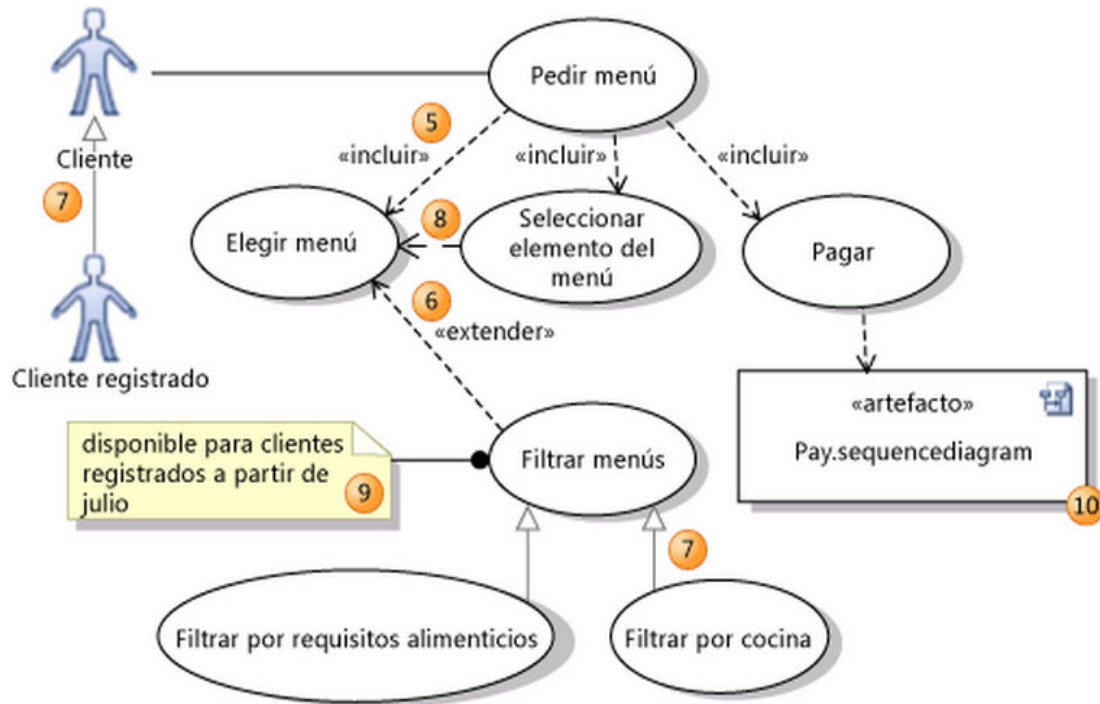
Los casos de uso que un sistema o componente admiten aparecen dentro de su rectángulo. Para aclarar el ámbito del sistema, puede resultar útil mostrar algunos casos de uso fuera del rectángulo.

Estructurar los casos de uso

5 **Incluir:** Un caso de uso de inclusión llama o invoca al caso de uso incluido. La inclusión se utiliza para mostrar cómo se divide un caso de uso en pasos más

pequeños. El caso de uso incluido se encuentra en el extremo con la punta de flecha.

Figura 17 Esquema de diagrama de casos de uso



6 **Extender:** Un caso de uso de extensión agrega objetivos y pasos al caso de uso extendido. Las extensiones solamente funcionan en ciertas condiciones. El caso de uso extendido se encuentra en el extremo con la punta de flecha.

7 **Herencia:** Relaciona un elemento especializado y un elemento general. El elemento general se encuentra en el extremo con la punta de flecha.

Un caso de uso especializado hereda los objetivos y actores de su generalización y puede agregar objetivos más específicos y los pasos para llevarlos a cabo.

Un actor especializado hereda los casos de uso, atributos y asociaciones de su generalización y puede agregar más elementos.

8 **Dependencia:** Indica que el diseño del origen depende del diseño del destino.

9 **Comment:** Se utiliza para agregar notas generales al diagrama.

10 **Artefacto:** Un artefacto proporciona un vínculo a otro diagrama o documento.

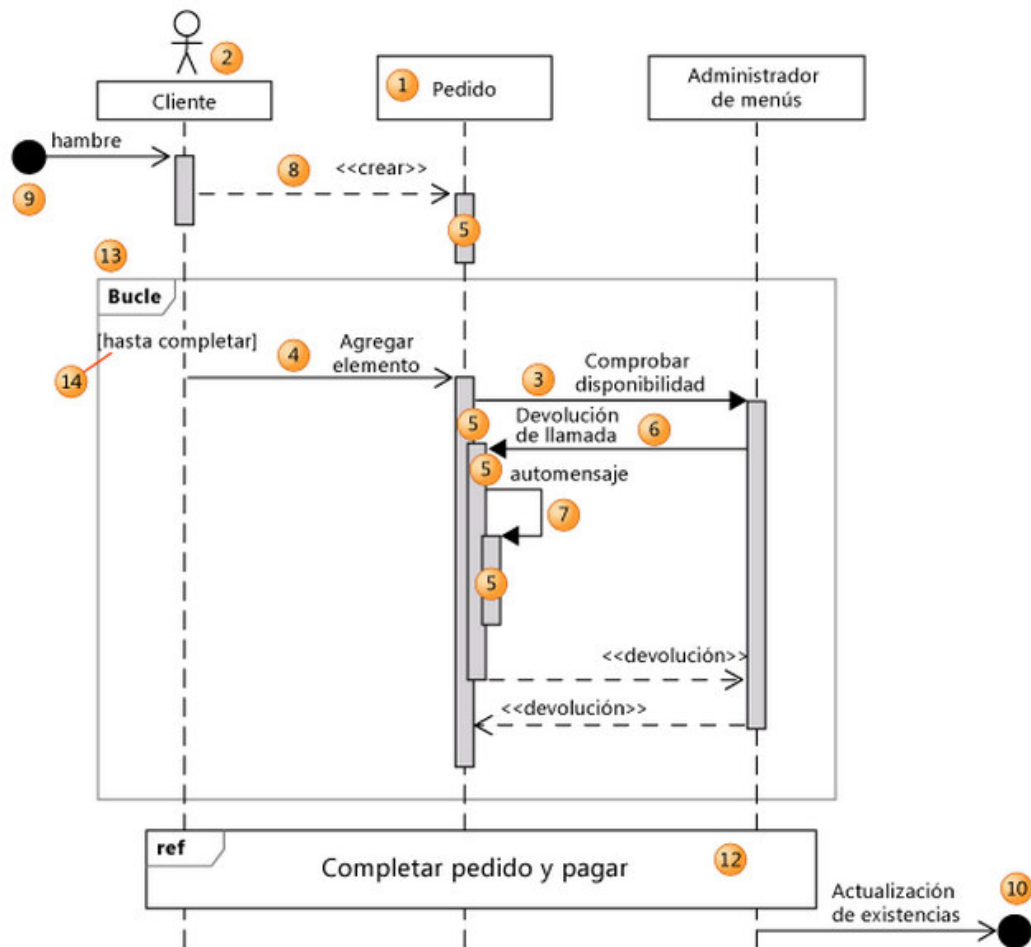
11 (no se muestra)

1.1.3.3 Diagramas de secuencia uml

Un diagrama de secuencia muestra una interacción, que representa la secuencia de mensajes entre las instancias de clases, componentes, subsistemas o actores. El tiempo fluye hacia abajo en el diagrama y muestra el flujo de control de un participante a otro.

El diagrama muestra instancias y eventos de ejemplo, en lugar de clases y métodos; más de una instancia del mismo tipo puede aparecer en el diagrama y más de una aparición del mismo mensaje también puede aparecer.

Figura 18 Esquema de diagrama de secuencia



1 Línea de vida Una línea vertical que representa la secuencia de eventos que se producen en un participante durante una interacción, mientras el tiempo avanza. Este participante puede ser una instancia de una clase, componente o actor.

2 Actor Un participante que es externo al sistema que está desarrollando. Puede hacer que aparezca un símbolo de actor en la parte superior de una línea de vida estableciendo su propiedad Actor.

3 Mensaje sincrónico El remitente espera una respuesta a un mensaje sincrónico antes de continuar. El diagrama muestra la llamada y el retorno. Los mensajes sincrónicos se utilizan para representar llamadas de función ordinarias dentro de un programa, así como otros tipos de mensaje que se comportan de la misma manera.

4 Mensaje asincrónico Un mensaje que no requiere una respuesta antes de que el remitente continúe. Un mensaje asincrónico muestra sólo una llamada del remitente. Se utiliza para representar la comunicación entre subprocesos diferentes o la creación de un nuevo subproceso.

5 Incidencia de ejecución Un rectángulo sombreado vertical que aparece en la línea de la vida de un participante y representa el período durante el que el participante está ejecutando una operación.

La ejecución comienza donde el participante recibe un mensaje. Si el mensaje inicial es un mensaje sincrónico, la ejecución finalizará con una flecha de «devolución» al remitente.

6 Mensaje de devolución de llamada Un mensaje que vuelve a un participante que está esperando la devolución de una llamada anterior. La aparición de ejecución resultante aparece encima de la existente.

7 Automensaje Un mensaje de un participante a sí mismo. La aparición de ejecución resultante aparece encima de la ejecución de envío.

8 Crear mensajes Un mensaje que crea un participante. Si un participante recibe un mensaje de creación, este debe ser el primer mensaje que recibe.

9 Mensaje encontrado Un mensaje asincrónico de un participante desconocido o no especificado.

10 Mensaje perdido Un mensaje asincrónico a un participante desconocido o no especificado.

11 Comentario Un comentario se puede adjuntar a cualquier punto de una línea de vida.

12 Uso de interacción Agrega una secuencia de mensajes que se definen en otro diagrama.

13 Fragmento combinado Una colección de fragmentos. Cada fragmento puede agregar uno o más mensajes. Existen distintos tipos de fragmentos combinados.

14 Protección de fragmentos Se puede utilizar para enunciar una condición relativa a si el fragmento se producirá⁵.

1.1.4 El proceso unificado de desarrollo. Es un proceso de desarrollo de software y el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software.

Sin embargo, el Proceso Unificado es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto.

El proceso Unificado está basado en componentes, lo cual quiere decir que el sistema software en construcción está formado por componentes software interconectados a través de interfaces bien definidas.

El proceso unificado utiliza el Lenguaje Unificado de Modelado UML para preparar todos los esquemas de un sistema de software. Sin embargo, los verdaderos aspectos definitorios del Proceso Unificado se resumen en tres frases clave; dirigido por casos de uso, centrado en la arquitectura, e iterativo e incremental.

Estos conceptos de; desarrollo dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental, son de igual importancia. La arquitectura proporciona la estructura sobre la cual girar las iteraciones, mientras que los casos de uso definen los objetivos y dirigen el trabajo de cada iteración. La eliminación de una de las tres ideas reduciría drásticamente el valor del Proceso Unificado. Es como un taburete de tres patas. Sin una de ellas, el taburete se cae.

⁵ MICROSOFT. Manual de ayuda Visual Studio 2013 [On line]. Citado Agosto 25 de 2014. Disponible en < <https://msdn.microsoft.com/es-es/library/dd831853.aspx>>

1.1.5 El proceso unificado está dirigido por casos de uso. Un sistema software ve la luz para dar servicio a sus usuarios. Por tanto, para construir un sistema con éxito debemos conocer lo que sus futuros usuarios necesitan y desean. El término usuario representa alguien o algo (como otro sistema fuera del sistema en consideración) que interactúa con el sistema que estamos desarrollando.

Esta interacción se la denomina caso de uso. Un caso de uso es un fragmento de funcionalidad del sistema que proporciona al usuario un resultado importante. Los casos de uso representan los requisitos funcionales. Todos los casos de uso juntos constituyen el modelo de casos de uso, el cual describe la funcionalidad total del sistema. Basándose en el modelo de casos de uso, los desarrolladores crean una serie de modelos de diseño e implementación que llevan a cabo los casos de uso.

Los desarrolladores revisan cada uno de los sucesivos modelos para que sean conformes al modelo de casos de uso. Los ingenieros de prueba prueban la implementación para garantizar que los componentes del modelo de implementación implementan correctamente los casos de uso.

De este modo, los casos de uso no solo inician el proceso de desarrollo sino que le proporcionan un hilo conductor. Dirigido por casos de uso quiere decir que el proceso de desarrollo sigue una secuencia a través de una serie de flujos de trabajo que parten de los casos de uso.

1.1.6 El proceso unificado está centrado en la arquitectura. El papel de la arquitectura software es parecido al papel que juega la arquitectura en la construcción de edificios. El edificio se contempla desde varios puntos de vista: estructura, servicios, conducción de la calefacción, fontanería, electricidad, etc. Esto permite a un constructor ver una imagen completa antes de que comience la construcción. Análogamente, la arquitectura en un sistema software se describe mediante diferentes vistas del sistema en construcción.

La arquitectura es una vista del diseño completo con las características más importantes resaltadas, dejando los detalles de lado. Pero, ¿Cómo se relacionan los casos de uso y la arquitectura? Cada producto tiene una función y una forma. En esta situación, la función corresponde a los casos de uso y la forma a la arquitectura. Debe haber interacción entre los casos de uso y la arquitectura, deben evolucionar en paralelo.

1.1.7 El proceso unificado es iterativo e incremental. El desarrollo de un producto software comercial supone un gran esfuerzo que puede durar entre varios meses hasta posiblemente un año o más. Es práctico dividir el trabajo en partes más pequeñas o mini proyectos. Cada mini proyecto es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en el flujo de trabajo, y los incrementos, al crecimiento del producto.

Los desarrolladores basan la selección de lo que se implementara en una iteración en dos factores. En primer lugar, la iteración trata un grupo de casos de uso que juntos amplían la utilidad del producto desarrollado hasta ahora. Al ser mini proyectos, comienzan con los casos de uso y continúan a través del trabajo del desarrollo subsiguiente; análisis, diseño, implementación y prueba que termina convirtiéndose en código ejecutable. Por supuesto, un incremento no necesariamente es aditivo.

1.1.8 La vida del proceso unificado. El Proceso Unificado se repite a lo largo de una serie de ciclos que constituyen la vida de un sistema. Cada ciclo concluye con una versión del producto para los clientes.

Cada ciclo consta de cuatro fases: inicio, elaboración, construcción y transición. Cada fase se subdivide a su vez en iteraciones.

1.1.9 El producto. Cada ciclo produce una nueva versión del sistema, y cada versión es un producto preparado para su entrega. Consta de un cuerpo de código fuente incluido en componentes que puede compilarse y ejecutarse, además de manuales y otros productos asociados.

El producto terminado incluye los requisitos, casos de uso, especificaciones no funcionales y casos de prueba. Incluye un modelo de la arquitectura y el modelo visual (artefactos modelados con el Lenguaje Unificado de Modelado).

Aunque los componentes ejecutables sean los artefactos más importantes desde la perspectiva del usuario, no son suficientes por si solos. Esto se debe a que el entorno cambia. Se mejoran los sistemas operativos, los sistemas de base de datos y las maquinas que los soportan. De hecho, el que los requisitos cambien es una de las constantes del desarrollo de software.

1.1.10 Fases dentro de un ciclo. Cada ciclo se desarrolla a lo largo del tiempo. Este tiempo, a su vez, se divide en cuatro fases; Inicio, Elaboración, Construcción y Transición. A través de una secuencia de modelos, los implicados visualizan lo que está sucediendo en estas fases. Cada fase termina con un hito. Cada hito se determina por la disponibilidad de un conjunto de artefactos; es decir, ciertos modelos o documentos han sido desarrollados hasta alcanzar un estado predefinido.

En cada fase se llevan a cabo flujos de trabajo o disciplinas: requisitos, análisis, diseño, implementación y prueba. Cada fase se divide normalmente en iteraciones o mini proyectos. Una iteración típica pasa por los cinco flujos de trabajo.

Durante la fase de inicio, se desarrolla una descripción del producto final a partir de una buena idea y se presenta el análisis de negocio para el producto. En esta fase se responde a las siguientes preguntas; ¿Cuáles son las principales funciones del sistema para sus usuarios más importantes?, ¿Cómo podría ser la arquitectura del sistema?, ¿Cuál es el plan de proyecto y cuanto costara desarrollar el producto?

La respuesta a la primera pregunta se encuentra en un modelo de casos de uso simplificado que contenga los casos de uso más críticos. La arquitectura es provisional, y consiste típicamente en un simple esbozo que muestra los subsistemas más importantes. En esta fase, se identifican y priorizan los riesgos más importantes, se planifica en detalle la fase de elaboración, y se estima el proyecto de manera aproximada.

Durante la fase de elaboración, se especifican en detalle la mayoría de los casos de uso del producto y se diseña la arquitectura del sistema. La relación entre la arquitectura del sistema y el propio sistema es primordial. Una manera simple de expresarlo es decir que la arquitectura es análoga al esqueleto cubierto por la piel pero con muy poco musculo (el software) entre los huesos y la piel, solo lo necesario para permitir que el esqueleto haga movimientos básicos. El sistema es el cuerpo entero con esqueleto, piel y músculos.

Por tanto la arquitectura se expresa en forma de vistas de todos los modelos del sistema, los cuales juntos representaran al sistema entero. Esto implica que hay vistas arquitectónicas del modelo de casos de uso, del modelo de análisis, del modelo de diseño, del modelo de implementación y modelo de despliegue. La vista del modelo de implementación incluye componentes para probar que la arquitectura es ejecutable. Durante esta fase del desarrollo, se realizan los casos de uso más críticos que se identificaron en la fase de inicio. El resultado de esta fase es un línea base de la arquitectura.

Durante la fase de construcción se crea el producto, se añaden los músculos (software terminado) al esqueleto (la arquitectura). En esta fase, la línea base de la arquitectura crece hasta convertirse en el sistema completo. La descripción evoluciona hasta convertirse en un producto preparado para ser entregado a la comunidad de usuarios. El grueso de los recursos requeridos se emplea durante esta fase del desarrollo. Al final de esta fase, el producto contiene todos los casos de uso seleccionados para el desarrollo de esta versión. Sin embargo, puede que no esté completamente libre de defectos. Muchos de estos defectos se descubrirán y solucionarán durante la fase de transición.

La fase de transición cubre un periodo durante el cual el producto se convierte en versión beta. En la versión beta un número reducido de usuarios con experiencia prueba el producto e informa defectos y deficiencias. Los desarrolladores corrigen los problemas e incorporan algunas mejoras sugeridas en una versión general dirigida a la totalidad de la comunidad de usuarios.

La fase de transición conlleva actividades como la fabricación, formación del cliente, al proporcionar una línea de ayuda y asistencia, y la corrección de los defectos que se encuentren tras la entrega. El equipo de mantenimiento suele dividir esos defectos en dos categorías: los que tienen suficiente impacto en la operación para justificar una versión incrementada y los que pueden corregirse en la siguiente versión del producto.

1.1.11 La captura de casos de uso. Durante el flujo de trabajo de los requisitos se identifica las necesidades de usuarios y clientes como requisitos. Los requisitos funcionales se expresan como casos de uso en un modelo de casos de uso, y los demás requisitos o bien se adjuntan a los casos de uso a los que afectan, o bien se guardan en una lista aparte o se describen de alguna otra forma.

1.1.12 El modelo de casos de uso representa los requisitos funcionales. El modelo de casos de uso ayuda al cliente, a los usuarios y a los desarrolladores a llegar a un acuerdo sobre cómo utilizar el sistema. La mayoría de los sistemas tienen muchos tipos de usuarios.

Cada tipo de usuario se representa mediante un actor. Los actores utilizan el sistema al interactuar con los casos de uso. Todos los actores y casos de uso del sistema forman un modelo de casos de uso. Un diagrama de casos de uso describe parte del modelo de casos de uso y muestra un conjunto de casos de uso y actores como una asociación entre cada par actor/caso de uso que interactúan.

1.1.13 Análisis, diseño e implementación para realizar los casos de uso.

Durante el análisis y el diseño, transformamos el modelo de casos de uso mediante un modelo de análisis en un modelo de diseño, es decir, es una estructura de clasificadores y realizaciones de casos de uso. El objetivo es realizar los casos de uso de una forma económica de manera que ofrezca un rendimiento adecuado y pueda evolucionar en el futuro.

1.1.14 Creación del modelo de análisis a partir de los casos de uso. El modelo de análisis crece incrementalmente a medida que se analizan más y más casos de uso. En cada iteración, elegimos un conjunto de casos de uso y los reflejamos en el modelo de análisis. Construimos el sistema como una estructura de clasificadores (clases de análisis) y relaciones entre ellas. También describimos las colaboraciones que llevan a cabo los casos de uso, es decir, las realizaciones de los casos de uso.

1.1.15 ¿Qué es una arquitectura? Es lo que especifica el arquitecto en la descripción de la arquitectura. La descripción de la arquitectura permite al arquitecto controlar el desarrollo del sistema desde la perspectiva técnica. La arquitectura software se centra tanto en los elementos estructurales significativos del sistema, como subsistemas, clases, componentes y nodos, como en las colaboraciones que tienen lugar entre estos elementos a través de las interfaces.

Los casos de uso dirigen la arquitectura para hacer que el sistema proporcione la funcionalidad y uso deseados, alcanzando a la vez objetivos de rendimiento razonables. Una arquitectura debe ser completa, pero también debe ser suficientemente flexible como para incorporar nuevas funciones, y debe soportar la reutilización del software existente.

1.1.15.1 ¿Cómo se obtiene?

La arquitectura se desarrolla de forma iterativa durante la fase de elaboración pasando por los requisitos, el análisis, el diseño, la implementación y las pruebas. Utilizamos los casos de uso significativos para la arquitectura y un conjunto de otras entradas para implementar la línea base de la arquitectura o esqueleto del sistema. Este conjunto de entradas incluye requisitos del software del sistema, middleware, que sistemas heredados deben utilizarse, requisitos no funcionales, y demás.

1.1.15.2 ¿Cómo se describe?

La descripción de la arquitectura es una vista de los modelos del sistema, de los modelos de casos de uso, análisis, diseño, implementación y despliegue. La descripción de la arquitectura describe las partes del sistema que es más importante que comprendan los desarrolladores y otros interesados.

1.1.16 Análisis. Durante el análisis, analizamos los requisitos que se describen en la captura de requisitos, refinándolos y estructurándolos. El objetivo de hacerlo es conseguir una comprensión más precisa de los requisitos y una descripción de los mismos que fácil de mantener y que ayude a estructurar el sistema entero.

En el análisis hay que razonar sobre los aspectos internos del sistema, y por tanto resolver aspectos relativos a la interferencia de casos de uso y demás. También podemos utilizar un lenguaje más formal para apuntar detalles relativos a los requisitos del sistema.

Además, en el análisis podemos estructurar los requisitos de manera que nos facilite su comprensión, su preparación, su modificación, y, en general, su mantenimiento. Esta estructura (basada en clases de análisis y paquetes) es independiente de la estructura que se dio a los requisitos. Sin embargo, existe una trazabilidad directa entre esas distintas estructuras, de forma que podemos hacer la traza de diferentes descripciones del mismo requisito y mantener su consistencia mutua con facilidad. De hecho, esta trazabilidad directa se define entre casos de uso del modelo de casos de uso y realizaciones de casos de uso en el modelo de análisis.

Por último, la estructura de los requisitos que proporciona el análisis también sirve como entrada fundamental para dar forma al sistema en su totalidad; esto es debido a que queremos construir el sistema como un todo sostenible, y no solo describir sus requisitos.

1.1.16.1 Artefacto: modelo de análisis

El modelo de análisis se representa mediante un sistema de análisis que denota el paquete de más alto nivel del modelo. La utilización de otros paquetes de análisis es por tanto una forma de organizar el modelo de análisis en partes más manejables que representan abstracciones de subsistemas y posiblemente capas completas del diseño del sistema. Las clases de análisis representan abstracciones de clase y posiblemente de subsistemas del diseño del sistema. Dentro del modelo de análisis, los casos de uso se describen mediante clases de análisis y sus objetos. Esto se representa mediante colaboraciones dentro del modelo de análisis que llamamos realizaciones de casos de uso-análisis.

1.1.17 Diseño. En el diseño modelamos el sistema y encontramos su forma (incluida la arquitectura) para que soporte todos los requisitos que se le suponen. Una entrada esencial en el diseño es el resultado del análisis, esto es, el modelo de análisis. Este proporciona una comprensión detallada de los requisitos. Y lo que es más importante, impone una estructura del sistema que se debe conservar lo más fielmente posible cuando se da forma al sistema. En concreto, los propósitos del diseño son:

Adquirir una comprensión en profundidad de los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos, tecnologías de distribución y concurrencia, tecnologías de interfaz de usuario, tecnologías de gestión de transacciones, etc.

Crear una entrada apropiada y un punto de partida para actividades de implementación subsiguientes capturando los requisitos o subsistemas individuales, interfaces y clases.

Ser capaces de descomponer los trabajos de implementación en partes más manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo, teniendo en cuenta la posible concurrencia. Esto resulta útil en los casos en los que la descomposición no puede ser hecha basándose en los resultados de la captura de requisitos.

1.1.17.1 Artefacto: modelo de diseño

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso centrándose en como los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar. Además, el modelo de diseño sirve de abstracción de la implementación de las actividades de implementación.

En el modelo de diseño, los casos de uso son realizados por las clases de diseño y sus objetos. Esto se representa por colaboraciones en el modelo de diseño y denota realización de caso de uso-diseño. Adviértase que realización de caso de uso-diseño es diferente de la realización de caos de uso-análisis. Lo anterior describe como se realiza un caso de uso en términos de interacción entre objetos del diseño, mientras que el último describe como se realiza un caso de uso en términos de interacción entre objetos de análisis.

1.1.18 Implementación. En la implementación empezamos con el resultado del diseño e implementamos el sistema en términos de componentes, es decir, ficheros de código fuente, ficheros de código binario, ejecutables y similares.

Los propósitos principales de la implementación son:

1. Planificar las integraciones de sistema necesarias en cada iteración. Seguimos para ello un enfoque incremental, lo que da lugar a un sistema que se implementa en una sucesión de pasos pequeños y manejables.

2. Distribuir el sistema asignando componentes ejecutables a nodos en el diagrama de despliegue. Esto se basa fundamentalmente en las clases activas encontradas durante el diseño.
3. Implementar las clases y subsistemas encontrados durante el diseño. En particular, las clases se implementan como componentes de fichero que contienen código fuente.
4. Probar los componentes individualmente, y a continuación integrarlos compilándolos y enlazándolos en uno o más ejecutables, antes de ser enviados para ser integrados y llevar a cabo las comprobaciones del sistema.

1.1.18.1 Artefacto: modelo de implementación

El modelo de implementación describe como los elementos del modelo de diseño, como las clases, se implementan en términos de componentes, como ficheros de código fuente, ejecutables, etc.

El modelo de implementación describe también como se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados, y como dependen los componentes unos de otros.

1.1.19 Prueba. En el flujo de trabajo de la prueba verificamos el resultado de la implementación probando cada construcción, incluyendo tanto construcciones internas como intermedias, así como las versiones finales del sistema a ser entregadas a terceros. Los objetivos de la prueba son:

Planificar las pruebas necesarias en cada iteración, incluyendo las pruebas de integración y las pruebas de sistema. Las pruebas de integración son necesarias para cada construcción dentro de la iteración, mientras que las pruebas de sistema son necesarias solo al final de la iteración.

Diseñar e implementar las pruebas creando los casos de prueba que especifican que probar, creando los procedimientos de prueba que especifican como realizar las pruebas y creando, si es posible, componentes de prueba ejecutables para automatizar las pruebas.

Realizar las diferentes pruebas y manejar los resultados de cada prueba sistemáticamente. Las construcciones en las que se detectan defectos son probadas de nuevo y posiblemente devueltas a otro flujo de trabajo, como diseño o implementación, de forma que los defectos importantes puedan ser arreglados.

1.1.19.1 Artefacto: modelo de pruebas

El modelo de pruebas describe principalmente como se prueban los componentes ejecutables en el modelo de implementación con pruebas de integración y de sistema. El modelo de pruebas puede describir también como han de ser probados aspectos específicos del sistema; por ejemplo, si la interfaz de usuario es utilizable y consistente o si el manual de usuario del sistema cumple con su cometido.

1.1.19.2 Artefacto: caso de prueba

Un caso de prueba especifica una forma de probar el sistema, incluyendo la entrada o resultado con la que se ha de probar y las condiciones bajo las que ha de probarse. En la práctica, lo que se prueba puede venir dado por un requisito o colección de requisitos del sistema cuya implementación justifica una prueba que es posible realizar y que no es demasiado cara de realizar.

Los siguientes son casos de prueba comunes:

Un caso de prueba que especifica como probar un caso de uso o un escenario específico de un caso de uso. Un caso de prueba de este tipo incluye la verificación del resultado de la interacción entre los actores y el sistema, que se satisfacen las precondiciones y poscondiciones especificados por el caso de uso. Un caso de prueba basado en un caso uso especifica típicamente una prueba del sistema como caja negra, es decir, una prueba del comportamiento perceptible exteriormente en el sistema.

Un caso de prueba que especifica como probar una realización de caso de uso-diseño o un escenario específico de la realización. Un caso de prueba de este tipo puede incluir la verificación de la interacción entre los componentes que implementan dicho caso de uso. De modo que los casos de prueba basados en una realización de caso de uso típicamente especifican una prueba del sistema como caja blanca, es decir, una prueba de la interacción interna entre los componentes del sistema⁶.

⁶ JACOBSON, Ivar, BOOCH, Grady, RUMBAUGH, James. El proceso unificado de desarrollo de software. Madrid: Pearson Educación. S. A, 2000. 464 p.

1.2 MARCO JURÍDICO

La realización de este proyecto se apoyara en lo siguiente:

- Para la adquisición del programa Visual Basic, los propietarios de *Microsoft Corporation* que lo fabrica y distribuye ofrecen a los estudiantes la iniciativa **DREAMSPARK** para la descarga gratuita del programa para estudiantes universitarios comprobados. *Microsoft Corporation* comprueba el estado estudiantil mediante la dirección de correo electrónico personal (dominio) de la Universidad de Nariño. En la siguiente página electrónica se puede realizar esta verificación:
<https://www.dreamspark.com/Student/Default.aspx?tab=2#>
- Norma Sismo resistente de Colombia NSR - 10
 - ✓ Título A: Requisitos Generales de Diseño y Construcción Sismo Resistente
 - ✓ Título B: Cargas
 - ✓ Título C: Concreto Estructural
 - ✓ Título H: Estudio Geotécnico

2 METODOLOGÍA

OBTENCIÓN DE LA INFORMACIÓN

Para desarrollar el proyecto es necesario buscar información para conocer los procedimientos analíticos para el análisis y diseño de los diferentes tipos muros de contención y su respectiva normatividad (NSR-10).

Igualmente será necesario acudir a libros especializados en Ingeniería de Software, para consultar sobre el Lenguaje Unificado de Modelado UML y el Proceso Unificado de Desarrollo de Software RU. Además se debe conocer el lenguaje de programación Visual Basic.

ANÁLISIS Y CLASIFICACIÓN DE LA INFORMACIÓN RECOLECTADA

Analizar la información que se va a incluir en el proyecto, y organizarla según cada área del conocimiento de la siguiente manera.

- ✓ Requerimientos técnicos y normatividad colombiana NSR-10
- ✓ Análisis y diseño de muros de contención y análisis estructural.
- ✓ Ingeniería de software.
- ✓ Lenguaje de Programación.

DEFINIR LOS REQUERIMIENTOS DEL SOFTWARE

En esta etapa hay que hacer una recolección, análisis y verificación de las necesidades del usuario, que serían los requerimientos del software. Hay que distinguir los funcionales y los no funcionales, los primeros definen las funciones que el sistema será capaz de realizar, y los segundos son los que definen las limitantes del sistema. La técnica para la toma de requerimientos es el Lenguaje Unificado de Modelado UML.

IDENTIFICACIÓN Y ELABORACIÓN DE LOS CASOS DE USO

Es la etapa más importante del proyecto, ya que los casos de uso expresan los pasos interactivos que un usuario deberá seguir para obtener un resultado deseado.

Para evitar la aglomeración de casos de uso, es necesario modularizar los casos de uso utilizando múltiples flujos o eventos, pero para evitar confusiones hay que referenciar el inicio y fin de cada evento.

CREACIÓN DE ALGORITMOS

En esta etapa se elaboraran los algoritmos del software, partiendo de toda la información útil sobre el proceso de diseño de muros de contención.

De modo, que cada proceso se descompondrá en subprocesos, para simplificar y modularizar la creación de los algoritmos, en lo que se refiere a los pasos analíticos de los diferentes procedimientos de diseño.

Eventualmente según el desarrollo del proyecto se optara por utilizar los diagramas de flujo o los pseudocódigos para representar los algoritmos.

DEFINICIÓN DE LA ARQUITECTURA

En esta etapa se determinara la arquitectura del software con base en los atributos de calidad que se espera obtener con la realización del proyecto, por eso, hay que definir las responsabilidades para cada componente del software.

Es importante probar la arquitectura obtenida con los requerimientos actuales o futuros. Además hay que, pronosticar las restricciones que influyen en la determinación de la arquitectura.

PLANIFICACIÓN Y REALIZACIÓN DE ITERACIONES

En esta etapa, es fundamental integrar las anteriores etapas con el fin de realizar una secuencia de iteraciones. Cada iteración representa un mini proyecto, que está compuesta por un flujo de trabajo, que son; requisitos, análisis, diseño, implementación y prueba. El objetivo de hacer esto es conseguir que el desarrollo del software sea iterativo e incremental.

Además hay que hacer una planificación para la realización de la secuencia de iteraciones, y al final de cada una de estas un análisis para revisar la satisfacción de los requerimientos determinados y futuros.

Para el caso de verificar si hay un incremento con la realización de la iteración, se examina el estado del sistema llamado línea base, entonces, un incremento se notara en las diferencias entre dos líneas bases sucesivas.

CODIFICACIÓN DE LOS ALGORITMOS

Al verificar que el proceso de iteración cumple con los objetivos planificados, entonces se traducirá los diagramas de flujo o pseudocódigos al lenguaje de programación Visual Basic, para que el usuario examine los resultados obtenidos.

DISEÑO DE LA INTERFAZ DE USUARIO

En esta etapa se busca utilizar los elementos hardware y software que le permitan interactuar al usuario con la información recibida y el ordenador. Para lo cual se utilizara las metáforas del mundo real, que asocien un campo de acción que domine el usuario con la posible interfaz (Ej.: Cuarto de habitación con escritorio de computador). De modo, que hay que conocer los deseos de los usuarios y los programadores. Para que esta etapa sea exitosa, hay que tener en cuenta tres principios a saber; dar control al usuario (en lo que se refiere a manejo y navegación), aliviar la carga de memoria a corto plazo y la consistencia, es decir, que el usuario pueda utilizar sus conocimientos adquiridos en el manejo de otros programas. Según la evolución del diseño se tomara la decisión de evaluar la interfaz obtenida.

3 CONCLUSIONES

Mediante la realización de este proyecto, se consiguió elaborar una herramienta informática denominada **Trasdós**, capaz de procesar datos de entrada, de acuerdo, a los diagramas de flujo. Obteniendo resultados rápidos y precisos, con sus respectivas tablas y esquemas.

Debido a las múltiples fuerzas a las que se ve sometido un muro de contención, se optó por individualizar su estudio para lograr hacer combinaciones de carga de diseño; el empuje del suelo de relleno, el empuje debido a la sobrecarga, el empuje debido al sismo y el empuje por presencia de nivel freático. Por lo tanto, se estudió las teorías y expresiones matemáticas para su determinación.

Debido a las semejanzas que hay entre los diferentes procesos para diseñar un muro de contención, en lo que se refiere a los diagramas de flujo se implementó estas semejanzas para crear módulos dentro de los algoritmos que en el avance del proyecto se modificarían de acuerdo a la tipología del muro de contención.

Se observa que los casos de uso son fundamentales en el desarrollo de software, sin importar su nivel de complejidad, ya que permiten comprender los pasos que hay que seguir para cumplir con los requisitos funcionales del sistema.

Es necesario que; el usuario posea los conocimientos suficientes para el diseño de muros de contención para conseguir resultados óptimos y consecuentes con la utilización del software, ya que esta herramienta es simplemente un apoyo en el desarrollo de proceso de diseño y no sustituye en ningún momento el criterio que debe tener un profesional en ingeniería civil.

Para un estudiante de ingeniería civil que desee ingresar en el estudio del diseño de muros de contención, el uso de **Trasdós**, le facilitara comprender mejor conceptos fundamentales como; el empuje, el momento de volcamiento, el momento resistente, los factores de seguridad, ya que su interfaz se organizó estratégicamente para examinar los datos procesados, acorde a como se realiza el proceso de diseño.

Con nuestro poco conocimiento en el campo de las metodologías para desarrollar software, el proceso unificado de desarrollo RUP ofrece los pasos que hay que seguir para obtener un archivo ejecutable y su respectiva documentación, como son el modelo de casos de uso, el modelo de análisis, el modelo de diseño, el

modelo de implementación y el modelo de pruebas. Que son útiles en el ciclo de vida del software.

Un muro estable, quiere decir que los factores de seguridad por deslizamiento y volcamiento calculados son mayores a los factores de seguridad por deslizamiento y volcamiento estipulados en la norma. Esto implica que para lograr cumplir con estas condiciones el usuario puede exagerar en los datos de entrada para conseguir la estabilidad del muro, llevando a sobredimensionar el muro. Entonces, es obligación del usuario buscar dimensiones óptimas, aprovechando las ventajas que ofrece **Trasdós**, ya que el usuario puede hacer muchas pruebas variando los datos de entrada necesarios para llegar un diseño refinado.

4 RECOMENDACIONES

Respetar el origen de los datos, en lo que se refiere al tipo de muro y la combinación de carga, es decir, si se trabajó en un muro en voladizo con la condición de carga suelo + agua, entonces, para abrir este archivo se tiene que escoger muro de voladizo en la ventana para seleccionar el tipo de muro y en la ventana de selección de caso se tiene que escoger suelo + agua, se advierte esto porque de no hacerlo así, se procesan los datos pero no serían coherentes al contexto del muro que se está analizando.

Tener especial cuidado en el momento de ingresar las variables iniciales al sistema, verificando que se encuentren en las unidades que requiere el software **Trasdós** y las condiciones específicas para cada tipo de muro, ya que de esta operación depende el éxito del proceso de cálculo y la obtención de los resultados coherentes.

Seleccionar el tipo de muro más adecuado según las condiciones del terreno o la importancia de la obra civil, el usuario debe contar con el conocimiento técnico suficiente y el criterio adecuado, con el fin de garantizar la estabilidad de la estructura sin incurrir en altos costos o sobredimensionar la estructura.

MARCAS REGISTRADAS

IBM ® Rational Unified Process ® (RUP ®) son marcas registradas de **IBM Rational Software**.

GEO5 SOFTWARE GEOTÉCNICO ® es una marca registrada de **fine – Civil Engineering Software**.

Geostru MDC ® es una marca registrada de **GeoStru Software**.

GeoWalls ® es una marca registrada de **SIGMA INGENIERÍA – SOFTWARE Y SERVICIOS**.

Unified Modeling Language™ (UML ®) son marcas registradas de **OBJECT MANAGEMENT GROUP**

Visual Studio ® Visual Basic ® son marcas registradas de **Microsoft Corporation**.

REFERENCIAS BIBLIOGRAFICAS

ASOCIACIÓN COLOMBIANA DE INGENIERÍA SÍSMICA. Reglamento colombiano de Construcción Sismo Resistente NSR – 10. Bogotá: 2010.

DAS, Braja. Principios de Ingeniería de Cimentaciones: Presión lateral de tierra. 5 ed. Sacramento: Thomson Editores, 2006. 742 p.

ECU RED CONOCIMIENTO CON TODOS Y PARA TODOS. Visual Basic [en línea]. <http://www.ecured.cu/index.php/Visual_Basic> [citado en 28 de Abril de 2013]

ESCUELA POLITÉCNICA SUPERIOR DE AVILA. Breve resumen de la ingeniería civil [en línea]. <<http://ocw.usal.es/eduCommons/enseanzas-tecnicas/ingenieria-civil/contenido/TEMA%203-%20MUROS.pdf>> [citado en 20 Abril 2013]

GONZÁLEZ ROJAS, john. Visual Basic 6.0 Guía. [en línea]. Disponible en internet < <http://johngonzalezrojas.files.wordpress.com/2012/01/visual-basic-6-0-guia.pdf> > [citado el 15 de Agosto de 2013]

INSTITUTO COLOMBIANO DE NORMAS TÉCNICAS. Normas Colombianas para presentación de trabajo. Sexta actualización. Santa fe de Bogotá D.C. ICONTEC, 2008, 41 p.

CABAÑES JURADO, carlós [On line]. Problemas de interacción suelo estructura en cimentaciones y muros de contención. Influencia de los fenómenos de despegue y deslizamiento. Tesis Doctoral en Ingeniería Civil. Madrid: Universidad Politécnica de Madrid, Escuela técnica superior de ingenieros de caminos, canales y puertos, 2012. 425 p. [citado el 15 de Agosto de 2014] Disponible en < http://oa.upm.es/13731/1/Carlos_Jurado_Cabanes_1.pdf>

JARA MORI, Gonzalo. Estudio de la aplicabilidad de materiales compuestos al diseño de estructuras de contención de tierras y su interacción con el terreno, para su empleo en obras de infraestructura viaria: Tipologías de muros de contención. Tesis Doctoral en Ingeniería Civil. Madrid: Universidad Politécnica de Madrid. Departamento de Ingeniería y Morfología del terreno; 2008. 393 p. Disponible en Internet <http://oa.upm.es/1169/1/GONZALO_ANDRES_JARA_MORI.pdf> [citado en 14 Abril 2013]

L. BERRY, Peter y REID, David. Mecánica de Suelos. McGraw-Hill. University of Salford, 1993, 398 p.

QUISPE, Vilma; HUAMANTUCO, Dante y VARGAS, José. METODOLOGÍA RUP (RATIONAL UNIFIED PROCESS): Rational Unified Process (RUP). Puno, 2011, 48 p. Monografía (Ingeniería de Sistemas). Universidad Nacional del Altiplano. Facultad de Ingeniería Mecánica, Eléctrica, Electrónica y Sistemas. Disponible en Internet < <http://www.monografias.com/trabajos-pdf4/metodologia-rup-una-puno/metodologia-rup-una-puno.pdf>> [citado el 25 de Septiembre de 2013]

PROCEDIMIENTOS UNO S.L. Empuje de tierras sobre muros y zapatas corridas. Aplicación a ESwin. [en línea]. <<http://www.arqui.com/arquitienda/articulo.asp?pad=1&id=1442>> [citado el 14 Septiembre de 2013]

RETAMOZO MARTÍNEZ, Ludwig. Diseño Estructural de un edificio de Viviendas de Dos Sótanos y Cinco pisos, ubicado en San Isidro – Lima. Tesis de grado en Ingeniería Civil. Lima: Pontificia Universidad Católica del Perú. Facultad de Ingeniería, 2011, 95 p. Disponible en Internet <<http://es.scribd.com/doc/75514500/45/CONCEPTO-DE-MURO-DE-SOTANO>> [citado en 24 Abril 2013]

RIVERA CAICEDO, Luciano. Muros de Contención de Suelo Reforzado con Cintas Metálicas y Geo sintéticos. Popayán, Universidad del Cauca, 2004, 196 p.

RUMBAUGH, James, JACOBSON, Ivar y BOOCH, Grady. El Lenguaje Unificado de Modelado. Manual de Referencia. Madrid: Pearson Educación S. A, 2000. 552 p.

RUMBAUGH, James, JACOBSON, Ivar y BOOCH, Grady. El Proceso Unificado de Desarrollo de Software. Madrid: Pearson Educación S. A, 2000. 464 p.

SEGURA FRANCO. Jorge. Estructuras de Concreto I. Colombia, Universidad Nacional de Colombia, 2002, 505 p.

SERVICIO NACIONAL DE APRENDIZAJE. Informe exposición muros de contención. [en línea]. <<http://es.slideshare.net/johannaximena1/muros-de-contencion>> [citado en 18 Abril 2013]