



BUAP

BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

Facultad de Ingeniería

Secretaría de Investigación y Estudios de Posgrado

**APLICACIÓN ANDROID PARA VIGAS DE
CONCRETO REFORZADO EN BASE AL CÓDIGO
ACI 318-19 Y NTC 2021**

TESIS

Que para obtener el grado de

MAESTRO EN INGENIERÍA EN ESTRUCTURAS

Presenta:

ING. HÉCTOR SAMUEL PÉREZ CASTRO

Director de Tesis:

M.I. SILVIA CONTRERAS BONILLA

Codirector de Tesis:

DR. JOSÉ CAMARGO PÉREZ

Puebla, Pue.

Septiembre 2022



BUAP

“HUP, 50 años de enseñanza y salud”

Oficio No. SIEP/0069/2022

C. Héctor Samuel Pérez Castro

Matrícula 220470055

Pasante de la Maestría en Ingeniería
con opción terminal en Estructuras

Facultad de Ingeniería, BUAP.

Presente

Por medio del presente, el suscrito M.I. Angel Cecilio Guerrero Zamora, Director de la Facultad de Ingeniería, de acuerdo a su solicitud de aprobación de Tema de Tesis, le autoriza desarrollar el tema titulado: **Aplicación Android para vigas de concreto reforzado en base al código ACI 318-19 y NTC 2021**. Para obtener el grado de Maestro en Ingeniería con opción terminal en Estructuras. Asignándose como Directora de tesis a la M.I. Silvia Contreras Bonilla y Codirector al Dr. José Camargo Pérez.

Sin otro particular de momento, reciba un cordial saludo.

Atentamente

“Pensar bien, para vivir mejor”

H. Puebla de Zaragoza, enero 11 de 2022

M.I. Angel Cecilio Guerrero Zamora

Director



C.c.p. M.I. Silvia Contreras Bonilla y Dr. José Camargo Pérez, Director y Codirector del Tema de Tesis

C.c.p. Archivo

AEPS/JPHT/sco*

Facultad
de Ingeniería

Bld. Valsequillo y Av. San Claudio
s/n, edif. ING 4, Col. San Manuel,
Ciudad Universitaria,
Puebla, Pue. C.P. 72570
222 229 55 00 Ext. 7610

M. en I. Ángel Cecilio Guerrero Zamora
Director de la Facultad de Ingeniería
Benemérita Universidad Autónoma de Puebla
P r e s e n t e:

Por este medio reciba un cordial saludo, al mismo tiempo le informo que el **C. Héctor Samuel Pérez Castro** matricula **220470055** alumno de la Maestría en Ingeniería con opción terminal en Estructuras ha elaborado el trabajo titulado:

“Aplicación Android para vigas de concreto reforzado en base al código ACI 318-19 y NTC 2021”

Como tema de tesis, el cual ha sido revisado y no existe inconveniente alguno en autorizar su impresión. Lo anterior, para efectos académicos a que haya lugar.

Sin otro particular y agradeciendo su atención, quedo como su segura servidora.

ATENTAMENTE

H. Puebla de Z., a 5 de septiembre de 2022



M. en I. Silvia Contreras Bonilla
Directora de Tesis

DEDICATORIA

A mi Madre, quien me ayudó a tener confianza en mí mismo y me apoyó en todo momento para terminar una carrera universitaria y ahora una maestría.

Te agradezco por haberme acompañado durante todos estos años en México, y a pesar de las diversas dificultades por las que pasamos, pudimos salir adelante.

Te agradezco por haber creído en mí cuando nadie más lo hizo, por cuidarme, iluminarme, y darme los mejores consejos de vida.

Este logro es de nosotros dos, te quiero mucho mamá. MUCHAS GRACIAS.

AGRADECIMIENTO

A mi universidad, la Benemérita Universidad Autónoma de Puebla, que me ha dado una formación de calidad en la carrera técnica en Música como Pianista concertista, una carrera universitaria en Ingeniería Civil y ahora una maestría en Estructuras.

A mi asesora de tesis, M.I. Silvia Contreras Bonilla, por haberme apoyado en todo momento en la creación de este ejemplar, sin duda alguna una de las mejores docentes que he tenido el gusto de conocer.

A mi coasesor de tesis, Dr. José Camargo Pérez, por sus enseñanzas y recomendaciones.

Al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo económico para solventar mi maestría.

ÍNDICE

Resumen	VIII
Introducción	X
I. ANTECEDENTES	11
I.1. Introducción al proyecto	12
I.2. Sistema operativo Android	16
I.3. Estructuración del proyecto UI/UX	19
II. NORMATIVAS PARA VIGAS DE CONCRETO REFORZADO	25
II.1. Vigas de concreto reforzado	26
II.2. Código ACI 318-19	28
II.2.1. Generalidades y propiedades mecánicas según el código ACI 318-19	30
II.2.2. Diseño de vigas a flexión por ACI 318-19	37
II.2.3. Diseño de vigas a cortante por ACI 318-19	46
II.3. Norma NTC 2021	53
II.3.1. Generalidades y propiedades mecánicas según las normas NTC 2021	54
II.3.2. Diseño de vigas a flexión por NTC 2021	58
II.3.3. Diseño de vigas a cortante por NTC 2021	61

III. DESARROLLO EN ANDROID	64
III.1. Generalidades de la aplicación móvil	65
III.2. Arquitectura MVVM	68
III.2.1. Comparativa con MVC y MVP	72
III.2.2. Model de la aplicación	75
III.2.3. View de la aplicación	78
III.2.4. ViewModel de la aplicación	82
III.3. Lógica	85
III.3.1. Apartado del cálculo matricial	86
III.3.2. Apartado del diagrama de Momento y Cortante	96
III.3.3. Apartado del diseño de la sección de la viga	99
III.4. Resultados	102
CONCLUSIÓN	110
RECOMENDACIONES	112
BIBLIOGRAFÍA	113

RESUMEN

El presente proyecto de investigación muestra desde el desarrollo hasta la aplicación móvil para el sistema operativo Android que funcione como apoyo didáctico para el diseño y revisión de vigas de concreto reforzado. Esto con el fin de asegurar una comprensión en todos los casos de estudio que el alumno podría enfrentarse en la vida profesional, dándole la seguridad y la habilidad para diseñar dichos elementos una vez concluidos sus estudios. Para tal objetivo, se emplean las normativas ACI 318-19 y las NTC 2021, siendo las primeras no solo válidas en México, sino en Estados Unidos y varios países de Latinoamérica.

El único recurso que se utiliza para lograr esta herramienta es el programa de Android Studio, el cual, es de código abierto, lo que significa que su licencia es gratuita, además de los respectivas librerías.

ABSTRACT

This research project exhibits both development and the full mobile application for the Android operating system which functions as a didactic support for design and revision of reinforced concrete beams. The main aim is to ensure an understanding in all the study cases that the student could face in his or her professional life, giving him or her the security and the ability to design these elements once the student finishes his or her college education. For this purpose, the ACI 318-19 and NTC 2021 regulations are used in this project. The ACI 318-19 are not only valid in Mexico, but in the United States and several Latin American countries.

The only tool used to complete this mobile application is Android Studio software, which is open source, meaning that its license is free, and the respective libraries.

INTRODUCCIÓN

Conforme pasa el tiempo, la tecnología va en ascenso y dando pasos de gigante, surgiendo nuevos aportes para las diferentes áreas de interés social y profesional, y el sector educativo no es la excepción. El uso de herramientas computacionales y digitales mejora el aprendizaje de los estudiantes y facilita la enseñanza para los docentes, simplemente se ha vuelto parte esencial para la formación educativa del alumno.

El aprender a diseñar elementos de concreto reforzado no es tan sencillo como pareciera, se toman en cuenta muchas variables y cada país tiene sus respectivas normas que el ingeniero debe seguir para proponer un diseño funcional. El transmitir este conocimiento y asegurarse de que el estudiante entienda como calcular los elementos estructurales está en función de la duración de la asignatura y los ejercicios que se hagan en clase. Desafortunadamente, el tiempo no es suficiente para abarcar todos los casos posibles que el estudiante podría enfrentarse en el futuro.

El presente proyecto busca crear y proporcionar una aplicación móvil para el sistema operativo Android que contenga los casos de estudio que se requieren para diseñar vigas de concreto reforzado. Dicha herramienta está enfocada para servir como apoyo didáctico, cumpliendo las exigencias que requieren tanto los docentes como los estudiantes. Sin embargo, también puede ser de utilidad para cualquier ingeniero estructurista profesional.

El enfocarlo en un dispositivo móvil, permite obtener resultados como los que proporcionan programas comerciales computacionales pero con la ventaja de viajar cómodo y ligero llevando consigo un celular o tableta en vez de una computadora portátil, el obtener resultados más rápidos, tener una interfaz más intuitiva y amigable, y el ahorro económico al ser gratuita. Además, la aplicación

se basa en la Normativa Mexicana NTC 2021 y la Americana ACI 318-19, asegurando de esta manera su funcionalidad en México y en otros países. De igual forma, está disponible en el idioma Español e Inglés.

El desarrollo del trabajo del proyecto que se presenta se divide en tres capítulos, en el primero encontrará una descripción amplia referente al sistema operativo Android que se utilizó para la elaboración de la aplicación móvil; además, información relevante de la necesidad que surge al subsanar la importancia de utilizar aplicaciones tecnológicas en el área de la enseñanza. En el capítulo dos se muestra el análisis, comparativa y desgloce paso a paso del diseño y revisión a flexión y cortante en vigas de concreto reforzado, usando las normas actualizadas del ACI 318-19 y NTC 2021. Finalmente, en el capítulo tres se muestra el desarrollo y el diseño de las interfaces que conforman a la aplicación; así mismo, se muestra la precisión de los resultados que se obtienen en los ejercicios de apoyo para la validación del producto final.

CAPÍTULO I
ANTECEDENTES

I.1. GENERALIDADES DEL PROYECTO

La ingeniería civil es una de las carreras más difíciles que hay, tomando en cuenta que tiene varias ramas de especialización. Una de ellas se encarga de estudiar todo lo relacionado con las estructuras, siendo compleja ya que se deben considerar diversas variables como son los materiales a emplear, el tipo de estructuración, la realización de un análisis sísmico y/o de viento, comportamiento y características del suelo para diseñar cimentaciones, entre otras.

El completar estos cursos en la universidad se vuelve un verdadero reto tanto para el estudiante como para el docente, debido a que debe cumplir un temario dentro de un lapso de tiempo establecido.

Una de las asignaturas más importantes en el área de las estructuras, es “Diseño de Elementos de Concreto Reforzado”. En la Benemérita Universidad Autónoma de Puebla (BUAP), esta materia contempla la revisión y diseño de vigas. A pesar de abarcar todo un semestre para su estudio, el tiempo que se dedica a la teoría y a diversos casos convencionales de revisión y diseño estructural resulta ser limitado, y en ocasiones insuficientes. Este problema repercute en el nivel de conocimiento del estudiante, inclusive puede en un futuro generar un egresado incompetente e inseguro para revisar y diseñar vigas de concreto reforzado. Además, esta materia tiene vínculo con otras asignaturas referente al diseño estructural, por lo que es importante tener dominio del tema.

Otro punto, es el uso de normativas en las cuales el ingeniero civil se tiene que apoyar y cumplir para diseñar estructuras bajo un marco legal que le permita librarse de cualquier responsabilidad en caso de que algún evento extraordinario ponga en riesgo la estabilidad estructural.

Estas normas permiten diseñar elementos dentro de un rango económico y seguro. En México, de acuerdo a la región, se pueden usar las Normas Técnicas Complementarias y/o el Código ACI (*American Concrete Institute, en Inglés*) para el diseño de elementos de concreto reforzado. Debe quedar claro que dichas normas sirven de guía para construir edificaciones civiles estables; sin embargo, como se había mencionado con anterioridad, si la estructura falla debido a un evento natural como un terremoto, un tsunami, entre otros; que sobre pase lo calculado en el análisis, ya que no depende del calculista si la edificación presenta daños estructurales.

Un ingeniero estructurista toma en cuenta todos estos factores y trata de calcular y diseñar de la mejor manera con los recursos que están disponibles, siempre con el objetivo principal de salvaguardar la vida de los usuarios que habitan las edificaciones. Queda claro que el hombre jamás podrá ganarle a la naturaleza, por mucho que se trate de diseñar por el lado de la seguridad, inclusive si el costo del proyecto aumenta, eventualmente algún acontecimiento desastroso le hará saber a la humanidad quien manda.

Todo esto puede provocar que el estudiante se sienta agobiado e inclusive opte por buscar interés en otra rama de la ingeniería civil al sentirse presionado de todo lo que debe contemplar y la responsabilidad que conlleva realizar el cálculo estructural. Debe quedar claro para el alumno que su objetivo principal es el diseñar edificaciones seguras y dignas para cualquier persona, y al lograrlo implícitamente se están salvado vidas también.

A causa de tales imprevistos, el emplear apoyo tecnológico en las clases se vuelve imprescindible para aumentar la curva de aprendizaje. En donde el estudiante trabaje con un programa computacional, software, y/o simulador, que permita ser una herramienta didáctica lúdica, logrando así un aprendizaje fluido, constructivo y dinámico. Por ende, el uso de estos dispositivos electrónicos

permite cubrir un amplio rango de temas y ejercicios en corto tiempo que al docente se le facilita mostrar diversos casos sobre el tema.

El uso de computadoras portátiles permite elevar las clases a otro nivel, facilitando el entendimiento de temas complejos de ingeniería estructural que requieren de un análisis detallado y extenso; no obstante, es una herramienta que no está al alcance de todos. En adición, se debe tomar en cuenta la posibilidad de poseer un programa computacional potente, con una interfaz amigable e intuitiva, capaz de realizar dichos procesos matemáticos de manera inmediata, y que además sea de licencia gratuita o económica para que la mayor parte de los estudiantes tengan acceso a ella. Esto por ende, limita a un cierto porcentaje de la población estudiantil a tener la oportunidad de adquirir el conocimiento que merece.

Inclusive, otro inconveniente de usar computadoras portátiles es la incomodidad de estar cargando dicho equipo a todas partes en el transcurso del día cuando no se necesita, esto debido al horario que tenga el estudiante que lo obligue a permanecer en las instalaciones de la universidad. De igual manera, el riesgo que implica el llevar consigo dicho dispositivo en transporte público pone en peligro su integridad física al llamar al atención y sufrir un asalto.

En definitiva, no deja de ser una excelente herramienta siempre y cuando se cuente con la posibilidad de obtener el mayor provecho si las circunstancias del estudiante se lo permiten. Existe la variante de utilizar un celular inteligente o tableta que pueda reemplazar la utilización de una computadora portátil; sin embargo, la interrogante sería si puede desempeñarse de la misma manera. Sin duda alguna es factible, a pesar de que una computadora posee componentes con mayor capacidad debido a su tamaño como lo es una memoria de almacenamiento, memoria RAM y procesadores más avanzados, la realidad es que un programa que realice cálculos complejos depende más del lenguaje en el que fue creado y los algoritmos que se emplearon.

Hoy en día existen muchos lenguajes de programación potentes que se emplean para el desarrollo de software. Los dos lenguajes más poderosos y favoritos de los desarrolladores son C++ y Java por sus extensas librerías y funciones que permiten crear todo tipo de programa para las diferentes áreas profesionales como; la medicina, ingeniería, entre otras. Sin embargo, recientemente Google le ha dado más impulso al lenguaje Kotlin el cual hereda o nace de Java, haciendo que este lenguaje aproveche al máximo las ventajas que poseía Java e inclusive incorpore funcionalidades nuevas y avanzadas.

Siendo más específico, el lenguaje Kotlin es utilizado principalmente para el desarrollo de aplicaciones móviles dirigidos hacia el sistema operativo Android, el cual tiene muchas ventajas si se compara con su competencia inmediata que es iOS y más si se hace uso de la programación declarativa con JetPack Compose.

En conclusión, la presente investigación promueve la creación de una aplicación móvil para el sistema operativo Android que tenga integrado cada caso de estudio para la revisión y diseño de vigas de concreto reforzado, cumpliendo las expectativas del docente, estudiante, e inclusive profesionistas que requieran de la aplicación para corroborar cálculos de proyectos de la vida real.

Esto resolvería los inconvenientes de usar una computadora portátil. Además, existe la posibilidad de difundir dicha aplicación a nivel estatal, nacional e internacional, esto debido a que el producto final va a estar disponible en el idioma Español e Inglés, brindando una solución para la comunidad docente y estudiantil de la carrera de ingeniería civil. La probabilidad de que esto suceda es buena, ya que las normas que va a tener integrada la aplicación son las Normas Técnicas Complementarias del 2021 que se emplean en casi todos los estados de la República Mexicana y el ACI 318-19 de origen Estadounidense que se usa no solo en México, sino en muchos países de Latinoamérica.

I.2. SISTEMA OPERATIVO ANDROID

Android es un sistema operativo dirigido hacia dispositivos móviles con pantalla táctil como lo son los teléfonos inteligentes y tabletas, aunque también esta presente en dispositivos como relojes inteligentes, televisores o inclusive en los sistemas multimedia de algunos modelos de automóviles.

Este sistema operativo fue creado por Android Inc. en el 2003 con el objetivo principal de funcionar exclusivamente en cámaras digitales que pudieran conectarse a las computadoras sin necesidad de cables. Posteriormente, fue adquirido por Google en el 2005 al ver el potencial que esto podría conllevar, y dos años después liberan la primera versión beta del sistema operativo junto con la creación de la “Alianza del Dispositivo Móvil Abierto” (*Open Handset Alliance, OHA, por sus siglas en Inglés*), un consorcio de 84 compañías que se dedican al desarrollo de hardware, software y telecomunicaciones dedicadas al desarrollo de estándares abiertos para dispositivos móviles. En adición, a finales de ese mismo año Google lanza públicamente el “Paquete de Desarrollo de Software” (*Software Development Kit, SDK, por sus siglas en Inglés*) para que desarrolladores pudieran crear aplicaciones bajo licencia de Apache, lo cuál permite que sea libre y de código abierto. Un año después, a finales del 2008, se da a conocer la primera versión Android, 1.0 Apple Pie, y se habilita mundialmente la PlayStore, una plataforma donde no solo se pueden descargar aplicaciones Android, sino también películas, música y libros.

Desde entonces la empresa Google se ha dedicado a mejorar e incorporar servicios adicionales como GPS, comando de voz, base de datos, entre otros conforme se liberan versiones para este sistema operativo con el fin de alentar a programadores a desarrollar aplicaciones de calidad. Se debe mencionar que al empezar un proyecto, se especifica la versión mínima de Android que puede ejecutar dicha aplicación, ya que cada una tiene características que

pueden llevarse a cabo en versiones nuevas pero no en antiguas. Cada versión puede estar subdividida en actualizaciones que se conocen como “Interfaz de Programación de Aplicaciones” (*Application Programming Interface, API, por sus siglas en Inglés*). Tabla I.1.

Nombre	Número de Versión	Fecha de Lanzamiento	Nivel de API
-	1.0	Sep - 2008	1
-	1.1	Feb - 2009	2
Cupcake	1.5	Abr - 2009	3
Donut	1.6	Sep - 2009	4
Eclair	2.0 - 2.1	Oct - 2009	5 - 7
Froyo	2.2 - 2.2.3	May - 2010	8
Gingerbread	2.3 - 2.3.7	Dic - 2010	9 - 10
Honeycomb	3.0 - 3.2.6	Feb - 2011	11 - 13
Ice Cream Sandwich	4.0 - 4.0.5	Oct - 2011	14 - 15
Jelly Bean	4.1 - 4.3.1	Jul - 2012	16 - 18
KitKat	4.4 - 4.4.4	Oct - 2013	19 - 20
Lollipop	5.0 - 5.1.1	Nov - 2014	21 - 22
Marshmallow	6.0 - 6.0.1	Oct - 2015	23
Nougat	7.0 - 7.1.2	Jun - 2016	24 - 25
Oreo	8.0 - 8.1	Ago - 2017	26 - 27
Pie	9.0	Ago - 2018	28
Android 10	10.0	Sep - 2019	29
Android 11	11.0	Sep - 2020	30
Android 12	12.0	Sep - 2021	31

Tabla.I.1. Historial de Android. Fuente: Android Developers (2021)

Es importante tener en cuenta el API en el cual se va a desarrollar la aplicación ya que de eso depende la cantidad de usuarios que puedan disponer de la misma. Entre más bajo sea el nivel, mayor el porcentaje de dispositivos que pueden correr la aplicación; sin embargo, el codificar en una de las primeras API sacrifica ciertas características y servicios que solo están disponibles en versiones más recientes. Además, el producto final puede carecer de diseño y funcionalidades que pueden afectar la experiencia de usuario.

El presente proyecto se codifica a partir del API 21, que corresponde a la versión Lollipop 5.0. La razón por la cual se toma esta decisión, es porque a partir de esta actualización surge un nuevo enfoque de codificación y diseño que hasta la actualidad es usada y recomendada por la empresa Google, y con la ventaja de que la aplicación sea ejecutable en aproximadamente el 95% de los dispositivos de diferentes marcas y resoluciones que existen en el mercado.

Es a partir de esta actualización cuando se puede hacer uso de la librería “Material Design”, creada por la misma empresa Google, que permite personalizar el diseño de todos los elementos que conforman la interfaz como lo son botones, cuadros de texto, imágenes, menús de navegación, entre otros, modificando propiedades como la forma, fuente, colores, animaciones, transiciones y crear inclusive un propio tema para la marca si se cuenta con una. Figura.I.1.

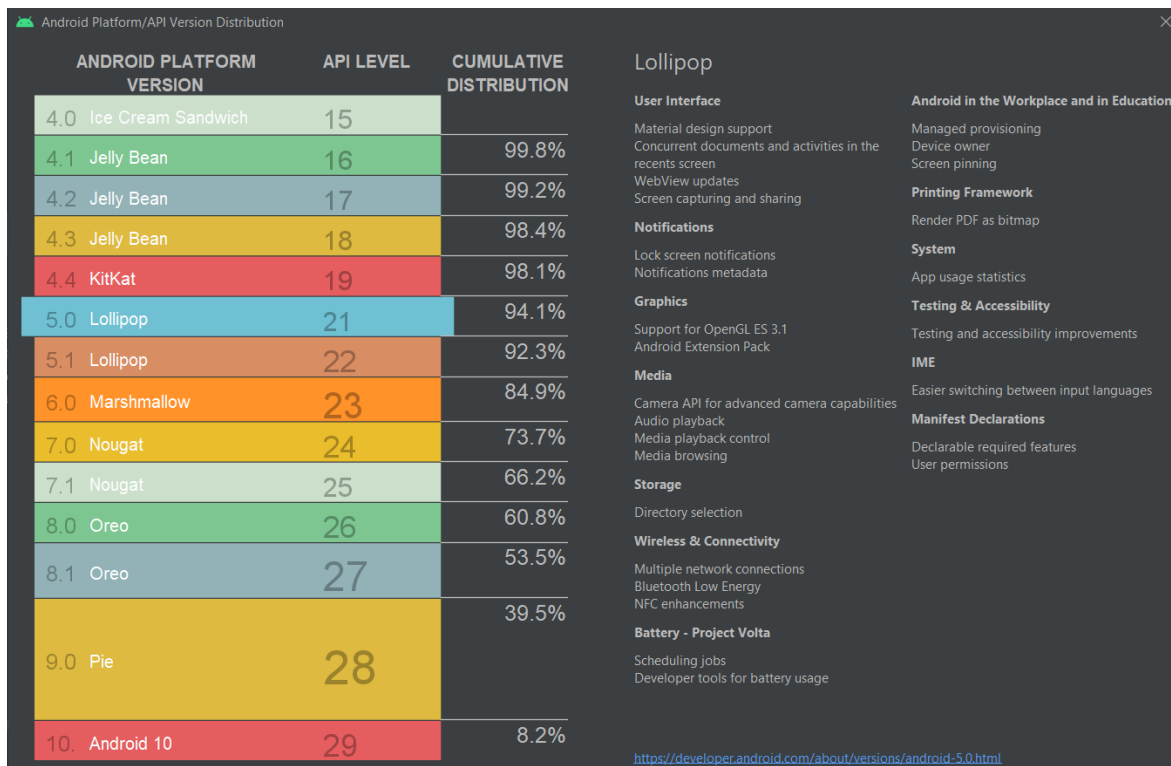


Figura.I.1. Características de la API 21, Lollipop. Fuente: AndroidStudio (2021)

En resumen, la aplicación ya finalizada cuenta con las ventajas que brinda la API 21, lo que representa en un producto que cumple con las recomendaciones que pide Google al distribuir una aplicación móvil de calidad, con un diseño de interfaz y navegación bien pensadas para garantizar una excelente experiencia de usuario.

Esto implica seriedad en la aplicación para ser usada en universidades de diferentes ciudades, estados e inclusive países, dándole confianza tanto al docente como al estudiante.

I.3. ESTRUCTURACIÓN DEL PROYECTO UI/UX

La “Interfaz de Usuario” (*UI, por sus siglas en Inglés*) y la “Experiencia de Usuario” (*UX, por sus siglas en Inglés*) trabajan en armonía con el objetivo de brindar una sensación de bienestar al utilizar un producto digital. Son diversas variables las que intervienen para crear un funcional UI/UX en las que participan juegos de colores, formas de los objetos, animaciones, transiciones, funcionalidad, textos multidioma, audio y video. Se busca que el usuario se sienta cómodo , satisfecho y que responda a sus necesidades con la aplicación móvil que se presenta; por lo cual, el mismo usuario se atreverá a compartir con sus compañeros y/o profesionistas de la misma área y hasta de otras universidades.

Actualmente existe un número limitado de aplicaciones Android que realizan un diseño o revisión de vigas de concreto reforzado; no obstante, todas ellas tienen detalles negativos como lo es el diseño de interfaz, la usabilidad, el idioma, carencia de contenido útil, poca precisión, costo para obtener una versión completa, por mencionar otros. Todo esto explica el por qué hay docentes que consideran que el uso de aplicaciones móviles en el aula no está justificada, y en cierta forma representa un riesgo utilizarlas en la formación del futuro ingeniero civil. Por lo tanto, tomando en cuenta estas debilidades se busca crear una

herramienta que satisfaga y cumpla todas las necesidades que se exigen en la asignatura de vigas de concreto reforzado, e inclusive sirva como herramienta para profesionistas que la requieran para revisar o diseñar vigas de concreto reforzado en proyectos de la vida real.

A continuación, se muestran aquellas aplicaciones móviles mejor calificadas con sus respectivas descripciones, justificando de esta manera la importancia y potencial de crear una aplicación móvil que haga un mejor trabajo que las que ya existen.

- *Diseño de vigas NTC - RSEE 2017, Ingeniería Civil México*

Aplicación que está completa en versión de paga, y aún así solo tiene disponible dos casos de estudio: vigas simplemente apoyadas y doblemente empotradas.

Además, carece de diagramas de momentos y cortantes, solo tiene disponible la norma NTC 2017 y 2004, únicamente está en el idioma Español y en cuestión a la experiencia de usuario (UX), ya sabe que datos ingresar por lo que no está dirigido a estudiantes.

En adición, obliga al usuario a ingresar los valores en unidades ya establecidas en el sistema Gravitacional e Inglés, haciendo mezclas de unidades que pueden inducir al usuario a cometer un error si no tiene los datos en las unidades que exige la aplicación.

- *Diseño en concreto armado, Yonathan*

Esta aplicación fue desarrollada en Perú y solo permite analizar vigas simplemente y doblemente reforzadas. Desafortunadamente, no especifica la norma que utiliza y los únicos resultados que despliega es la cantidad de acero de diseño que se requiere, área de acero mínima, cuantía balanceada y cuantía máxima de acero.

También carece de un diseño funcional, además de que es poco intuitiva y la interfaz carece de diseño lo que repercute en la experiencia de usuario. Aunado a que solo está disponible en el idioma Español. En cuestión de las unidades, solo está disponible en el sistema Gravitacional con unidades fijas y no permite ninguna conversión.

- *Perfect Structure - Reinforced Concrete Design, BlackSea Labs*

Esta aplicación solo está completa si se compran los módulos por separado, y no cuenta con casos de estudio por lo que el usuario está obligado a insertar todos los resultados de un análisis ya hecho por su propia cuenta.

Carece de diagramas de momentos y cortantes, y solo tiene disponible la norma ACI 318, aunque no se especifica si es la más reciente.

Por el tipo de diseño y programación, no da detalladamente el procedimiento que se realizó para obtener los resultados, por lo que está dirigido para ingenieros con experiencia y no para estudiantes. Algunas ventajas que tiene, es que permite la exportación en formato PDF y CSV.

- *Beam Master, CivilE. Soft*

Permite diseñar vigas de concreto reforzado de perfiles rectangulares, tipo L y T a partir de un predimensionamiento, obteniendo como resultado final una propuesta de acero para el armado de la vigas. El inconveniente de esta aplicación es que se usa un código desactualizado (ACI 318-14).

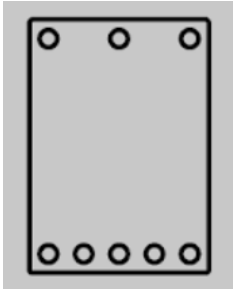
Se presenta en el idioma Inglés, poca disponibilidad de unidades, y para ciertos cálculos adicionales se requiere hacer un pago extra. Tiene un diseño de interfaz agradable pero en cuestión de la experiencia de usuario es poco amigable e intuitiva.

- *VIGA HIC, HERRAMIENTAS ING CIVIL*

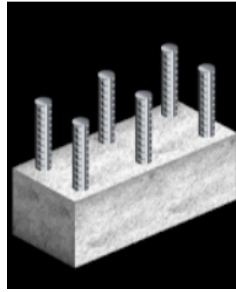
Se basa en un código ACI desactualizado. Siendo una aplicación muy sencilla permite exportar en formato PDF una memoria de cálculo de una sección de tipo rectangular. Realiza solamente revisión, careciendo de otros datos esenciales que se requieren para un curso universitario.

Se presenta en el idioma español, y las únicas unidades en las que puede trabajar el usuario es en Sistema Internacional (SI). La experiencia de usuario y la interfaz dejan mucho que desear, lo que, la convierte en una aplicación simple.

Las demás aplicaciones tienen problemas de ser imprecisos en los resultados, usar un código nacional, lo cual, lo restringe solo para esa región, y/o ser de paga. Figura I.2.



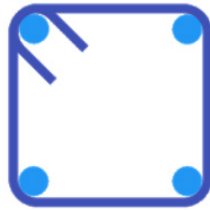
Diseño de vigas NTC
RSEE 2017



Diseño en concreto
armado



Perfect Structure
Reinforced Concrete
Design



Beam Master



Viga Hic

Figura.I.2. Logotipos de las aplicaciones Android para diseño y revisión de vigas de concreto reforzado mejor calificadas. Fuente: PlayStore (2021)

Tomando como referencia las aplicaciones previamente mencionadas, se puede obtener provecho de las deficiencias y carencias que tienden a desarrollar una herramienta que destaque entre todas las que hay en el mercado. De esta manera, es posible crear un producto de calidad que tenga todos los elementos que requieren los universitarios, docentes y profesionistas empleando las normativas NTC 2021 y el ACI 318-19 tanto en el idioma Inglés como en Español.

Además, se pretende poner mucha énfasis en el diseño de la interfaz y la experiencia del usuario, lo que determinará que la aplicación sea todo un éxito y bien aceptada no solamente en México, sino en otros países.

En conclusión, se ha dado toda la información necesaria en el presente capítulo justificando la creación de una aplicación Android para diseñar y revisar vigas de concreto reforzado para uso pedagógico y profesional.

Hoy en día, no hay suficientes aplicaciones en esta área que sean lo suficientemente efectivas como para ser empleadas en un curso universitario. El capítulo subsecuente se enfoca más en la esencia del proyecto que es todo lo relacionado a vigas de concreto reforzado, y una explicación detallada de las normas que se usan como referencia para el cálculo de estos elementos estructurales.

CAPÍTULO II
NORMATIVAS PARA VIGAS DE CONCRETO REFORZADO

II.1. VIGAS DE CONCRETO REFORZADO

En la ingeniería civil, la estructura es la parte fundamental de toda construcción ya que provee resistencia y rigidez. Es un conjunto de elementos enlazados mediante conexiones idealizadas como uniones simples, articulaciones, o uniones rígidas con el fin de recibir una serie de fuerzas externas, resistirlas internamente y transmitir las a sus apoyos sin que se altere la configuración de la misma.

El presente proyecto de investigación se enfoca en uno de estos elementos estructurales, siendo más específico, en vigas. Se puede definir a la viga como un cuerpo delgado que soporta cargas, usualmente en dirección a la gravedad y perpendicular a su eje longitudinal. Estas pueden estar hechas de madera, acero o concreto reforzado.

El concreto está compuesto de agregados finos y gruesos como la arena y la grava respectivamente, unidos mediante un aglutinante conformado por la mezcla de cemento y agua. En adición, se pueden emplear aditivos para modificar ciertas propiedades de este producto como la ductilidad, durabilidad y tiempo de fraguado.

Cuando el concreto se combina con el acero, recibe el nombre de “concreto reforzado”, siendo el refuerzo proporcionado por el acero, el cual, aporta resistencia a la tensión, una característica poco representativa en el concreto a pesar de tener una alta resistencia a la compresión.

Sin duda alguna, este tipo de elementos estructurales son muy usados en diferentes tipos de construcciones debido a su amplia versatilidad y costo-beneficio. A continuación, se enlistan algunas ventajas de construir vigas empleando concreto reforzado:

- La resistencia aumenta con el paso de los años.
- Son muy rígidas.
- Requieren de poco mantenimiento.
- Tienen un excelente comportamiento ante la intemperie. De hecho, con el correcto recubrimiento, las varillas de refuerzo no sufren ningún daño ante la presencia de agua o fuego.
- El costo de los componentes y mano de obra para construirlos son muy económicos a comparación del acero.

Cabe destacar que también hay desventajas al emplear este material como lo es el factor tiempo. El problema del concreto es que su resistencia máxima la obtiene a los 28 días, lo que significa en términos de costo que, la cimbra se cobra aparte y puede llegar a elevar el precio, y permanecer por lo menos dos semanas para alcanzar aproximadamente un 90% de su resistencia total.

Además, si no hay una supervisión constante, se corre el riesgo de construir estos elementos de manera errónea lo que puede comprometer la seguridad y funcionalidad de la estructura.

Aun así, no deja de ser el material más usado para la construcción. El cálculo de una viga de concreto reforzado para diseñarlo o revisarlo depende de la normativa que se use, que no son mas que guías y recomendaciones mínimas para la elaboración de elementos estructurales seguros y eficientes. En México, actualmente se pueden usar ya sean las NTC 2021 o el ACI 318-19, depende de la reglamentación municipal o estatal donde se planea edificar.

II.2. CÓDIGO ACI 318-19

Este código es proveniente de Estados Unidos y su nombre formal es “Código de Construcción de los Requerimientos para el Concreto Estructural” (*Building Code Requirements for Structural Concrete, en Inglés*) del Instituto Americano del Concreto (*American Concrete Institute, ACI, por sus siglas en Inglés*).

Desde los inicios del siglo XX hasta principios de la década de 1960, casi todo el diseño estructural de concreto reforzado en Estados Unidos se hizo empleando el método de diseño de esfuerzos de trabajo (llamado también diseño por esfuerzos permisibles o diseño lineal). En este método, también conocido WSD (Working Stress Design), se calculan primero las cargas muertas y vivas, llamadas cargas de trabajo o cargas de servicio, las cuales tienen que ser resistidas por el elemento estructural. Luego se determina el tamaño de los miembros de la estructura, de manera que los esfuerzos calculados por medio del método de la sección transformada no excedan de ciertos valores límites o permisibles.

No fue hasta 1963, que el método de diseño por resistencia última fue ganando rápidamente mucha popularidad, especialmente por los siguientes puntos:

- Usa un enfoque más racional que el método de diseño de esfuerzos de trabajo (WSD).
- Usa una consideración más realista del concepto de seguridad.
- Conduce a diseños más económicos.

El enfoque de este método (actualmente también conocido como *Diseño por Resistencia*), las cargas actuantes muertas y vivas se amplifican por ciertos factores de carga (equivalentes a factores de seguridad) y los valores resultantes se llaman *cargas factorizadas*. Los miembros se calculan y diseñan de manera que teóricamente fallen justo bajo estas cargas factorizadas.

Algunas de las ventajas que tiene el método de *Diseño por Resistencia*, el cual, se basa el Código ACI 318-19, en comparación con el método de *Esfuerzos de Trabajo* son las siguientes:

- La obtención de las expresiones toma en cuenta la forma no lineal del diagrama esfuerzo-deformación unitaria, esto implica que cuando se usan las ecuaciones resultantes se obtienen mejores estimaciones de la capacidad de carga.
- Se emplea un factor de seguridad más realista, permitiendo estimar las magnitudes de las cargas muertas que una estructura tendrá que soportar con más exactitud que al estimar las cargas vivas y ambientales.
- Una estructura diseñada con este método tendrá un factor de seguridad más uniforme contra el colapso ya que aprovecha ventajosamente los aceros de alta resistencia. Por lo tanto, el resultado es una mayor economía al emplear el diseño por resistencia.
- Permite diseños más flexibles debido a que el porcentaje de acero puede variar bastante. En consecuencia, se pueden usar secciones grandes con porcentajes pequeños de acero, o vice versa.

Para el desarrollo del presente proyecto, se hará mención de ciertos capítulos del código que tienen relación con la funcionalidad de la aplicación móvil, siendo más específico, aquellos capítulos que traten con el diseño a flexión y cortante.

II.2.1 GENERALIDADES Y PROPIEDADES MECÁNICAS SEGÚN EL CÓDIGO ACI 318-19

Previo al cálculo, se deben definir bien algunas propiedades mecánicas de las vigas de concreto reforzado como lo es su módulo de elasticidad (E), resistencia a la compresión ($f'c$), fluencia del acero de refuerzo (f_y); de igual manera, datos generales que se requieren como el predimensionamiento, combinación de cargas, etcétera. A continuación, se presenta cada una de estas propiedades y la fórmula o consideración correspondiente según el código ACI 318-19 para obtenerla.

- *Resistencia a la compresión ($f'c$)*

Esta propiedad mecánica muestra la capacidad del material para soportar sobre un área determinada y una carga específica. Se expresa en términos de esfuerzo, ya sea en kg/cm^2 usando el sistema MKS, MPa con el sistema internacional (SI) o en libras por pulgada cuadrada mediante el sistema inglés estadounidense (*Pound per square inch, psi, por sus siglas en Inglés*).

En la sección 19.2.1.1 del Código ACI 318-19 hace mención del valor mínimo de $f'c$ que puede usarse para diseñar elementos de concreto reforzado, para el caso de vigas es 2500 psi o 17 MPa. (Tabla II.1).

Sistema Inglés Americano	Sistema MKS	Sistema Internacional (SI)
1000 lb/in ²	70.31 kg/cm ²	6.89 MPa
2000 lb/in ²	140.61 kg/cm ²	13.79 Mpa
3000 lb/in ²	210.93 kg/cm ²	20.68 Mpa
4000 lb/in ²	281.22 kg/cm ²	27.58 Mpa
5000 lb/in ²	351.53 kg/cm ²	34.47 Mpa
6000 lb/in ²	421.84 kg/cm ²	41.37 Mpa

Tabla.II.1. Resistencia a la compresión ($f'c$) en el Sistema Americano, MKS e Internacional

Este valor también es necesario para construir el diagrama de esfuerzo-deformación del concreto, el cual, tiene características importantes a tratar. En primera instancia, tiene un comportamiento lineal conforme la carga incrementa, se mantiene de esa forma entre un tercio y un medio de la resistencia última. Posterior a ese intervalo, tiende a un comportamiento no lineal.

Referente a la deformación unitaria cuando el material alcanza la ruptura, usualmente esta puede variar de 0.003 a 0.004. Para fines de cálculo, se supone que el concreto falla al llegar al valor de 0.003, se especifica en la sección 22.2.2.1 del código ACI 318-19. (Figura II.1)

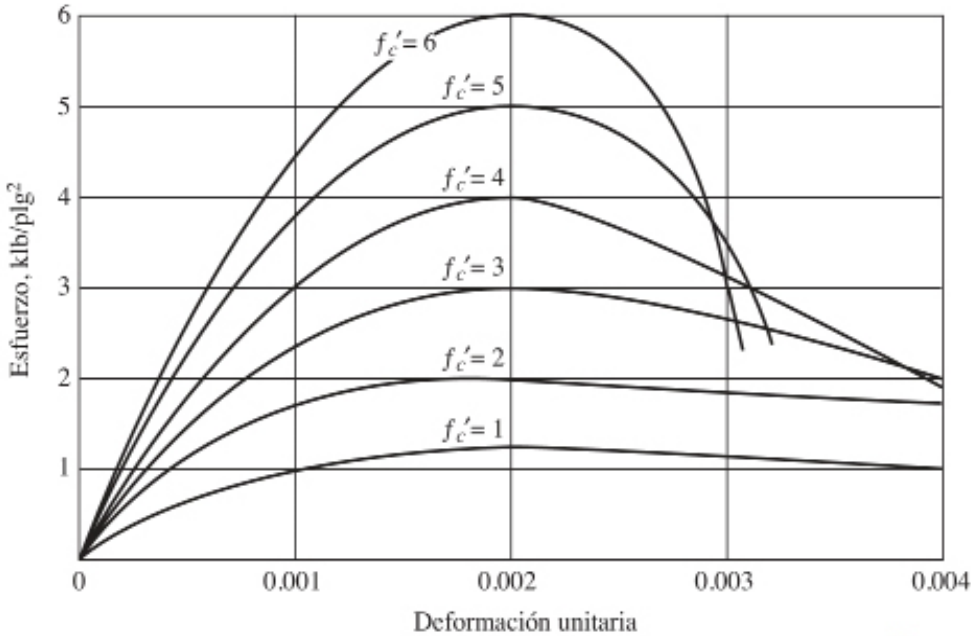


Figura.II.1. Curva esfuerzo-deformación unitaria típica del concreto, con carga de corto plazo. Fuente: Diseño de Concreto Reforzado, McCormarc (2017)

Haciendo referencia al código ACI 318-19 los requerimientos para las mezclas de concreto reforzado se basan en la filosofía de que deben proveer resistencia y durabilidad adecuadas.

El reglamento define un valor mínimo para $f'c$ el cual, ya se definió anteriormente; sin embargo, no hay un valor máximo a no ser que se especifique en algún requisito. En adición, bajo ciertas circunstancias, los requisitos de durabilidad pueden exigir un valor de $f'c$ mayor que el requerido para fines estructurales.

Esta propiedad del concreto es una de las requeridas en el desarrollo de la aplicación para el diseño de la viga, ya que en función de esta y dependiendo del sistema de unidades que se elija se obtiene el valor del módulo de elasticidad que propone el código ACI 318-19.

- *Módulo de Elasticidad o Young (E)*

Esta es una propiedad característica de los materiales que indica la rigidez mediante la relación entre los incrementos de tensión que se aplican y los incrementos de deformación longitudinal unitaria que se generan.

El módulo de elasticidad del concreto puede variar dependiendo de la resistencia a la compresión ($f'c$), la edad, tipo de agregado que se emplea, proporción de agua-cemento, entre otros. Se puede identificar en la gráfica de esfuerzo-deformación como la pendiente de una línea trazada desde el origen hasta un 45% aproximadamente de su resistencia última a la compresión.

La sección 19.2.2.1 del código ACI 318-19, establece que se puede usar la siguiente expresión matemática para calcular el módulo de elasticidad del concreto cuyo peso esté dentro del rango de 90 a 160 lb/pie³.

SI (MPa)	MKS (kg _f /cm ²)	US units (psi)
$E_c = w_c^{1.5} 0.043 \sqrt{f'c}$	$E_c = w_c^{1.5} 0.14 \sqrt{f'c}$	$E_c = w_c^{1.5} 33 \sqrt{f'c}$

Donde:

E_c = *Módulo de elasticidad*

w_c = *Peso del concreto*

$f'c$ = *Resistencia a la compresión especificada a los 28 días.*

Para concreto de peso normal, aproximadamente 145 lb/pie³ (≈2300 kg/m³), el código ACI 218-19 simplifica la fórmula del módulo de elasticidad por una más compacta.

SI (MPa)	MKS (kg _f /cm ²)	US units (psi)
$E_c = 4,700 \sqrt{f'c}$	$E_c = 15,100 \sqrt{f'c}$	$E_c = 57,000 \sqrt{f'c}$

El valor del módulo de elasticidad del concreto es sensible a un número de variables que incluyen el tipo de agregado, los constituyentes del concreto, la adherencia entre la pasta y el agregado, la dosificación de la mezcla, y la edad del concreto.

Las fórmulas para obtener el valor del módulo de elasticidad, según el código ACI 318-19, han mostrado ser por muchos años adecuadas para la mayoría de las aplicaciones. No obstante, para ciertas aplicaciones, estas ecuaciones pueden no ser tan precisas. Esto se ha observado con concreto liviano, de alta resistencia ($f'c > 56$ MPa) y en mezclas con bajo volumen de agregado grueso como ocurre en concretos autos nivelantes.

Para el presente proyecto, esta propiedad va a estar en función de la resistencia a compresión del concreto ($f'c$) y del sistema de unidades que se elija, dándole la libertad al usuario de obtener el valor que más le convenga.

- *Fluencia del acero (f_y)*

Está ligada al acero de refuerzo y no al concreto; sin embargo, al hablar de concreto reforzado, se hace referencia a la unión de estos materiales que en conjunto forman uno solo con las mejores ventajas que cada uno puede contribuir para el comportamiento estructural. Esto incluye la propiedad de aumentar la flexión gracias a la aportación del acero que el concreto por sí solo no lograría.

La fluencia del acero (f_y) se refiere a una deformación irrecuperable debido a una fuerza aplicada, la cual, solo se recuperará la parte de su deformación correspondiente a la deformación elástica, quedando una deformación irreversible. Este punto donde se presentan estas deformaciones permanentes se conoce como límite de fluencia, donde se marca el fin de la zona elástica del material y el inicio de la zona plástica

Como menciona el autor McCormac en su obra "Diseño de Concreto reforzado", las varillas de refuerzo se fabrican con base en acero de lingotes, de ejes o de rieles. Aunque, sólo ocasionalmente se laminan usando viejos rieles ferroviarios o ejes de locomotoras. Estos últimos aceros, después de trabajarlos en frío durante muchos años, no son tan dúctiles como los aceros de lingote.

Hay varios tipos de varillas de refuerzo con designaciones de la ASTM, estos aceros están disponibles en varios grados diferentes, como el Grado 50, Grado 60, 80, etc., donde Grado 60 significa que el acero tiene un punto de fluencia específico de 60,000 lb/plg² (420MPa), el Grado 80 significa 80,000 lb/plg² (550MPa), etcétera.

Debido a su costo, las varillas de Grado 60 (420Mpa) son las más comúnmente usadas en el diseño de elementos de concreto reforzado. Cabe mencionar, que antes del año 2019, solo se permitía usar el Grado 60 en sistemas especiales de resistencia sísmica; sin embargo, en la actualidad es permitido usar en ciertos casos acero de Grado 80 y 100 para resistir momentos, fuerzas axiales y cortantes, aunque el costo de estos aceros aumenta además de su difícil adquisición.

En la práctica es muy común considerar el uso de acero Grado 60 (420MPa) para el diseño estructural de concreto reforzado, y de igual manera que satisfaga los requisitos que se mencionan el código ACI 318-19 en el apartado 20.2.2.5.

A continuación, se muestra en la tabla II.2 la tabla 20.2.2.4 del código ACI 318-19 con los diferentes tipos de acero de acuerdo a su grado y uso de aplicación, se debe observar que de acuerdo al máximo valor de fluencia (f_y) es el acero que se debe emplear.

Uso	Aplicación		Valor máximo de f_y o f_{yt} permitido para cálculos de diseño, MPa	Normas ASTM aplicables			
				Barras corrugadas	Alambres corrugados	Refuerzo de alambre electrosoldado	Parrillas de barras corrugadas soldadas
Flexión, fuerza axial, retracción y temperatura	Sistemas sísmicos especiales	Pórticos a momento especiales	550	A706M ^[2]	No permitido	No permitido	No permitido
		Muros estructurales especiales ^[1]	690				
	Otras	690 ^{[3][4]}	A615M, A706M, A955M, A996M, A1035M	A1064M, A1022M	A1064M, A1022M	A184M ^[5]	
Apoyo lateral de barras longitudinales o confinamiento del concreto	Sistemas sísmicos especiales		690	A615M, A706M, A955M, A996M, A1035M	A1064M, A1022M	A1064M ^[6] , A1022M ^[6]	No permitido
	Espirales		690	A615M, A706M, A955M, A996M, A1035M	A1064M, A1022M	No permitido	No permitido
	Otras		550	A615M, A706M, A955M, A996M	A1064M, A1022M	A1064M, A1022M	No permitido
Cortante	Sistemas sísmicos especiales ^[7]	Pórticos a momento especiales ^[8]	550	A615M, A706M, A955M, A996M	A1064M, A1022M	A1064M ^[6] , A1022M ^[6]	No permitido
		Muros estructurales especiales ^[9]	690				
	Espirales		420	A615M, A706M, A955M, A996M	A1064M, A1022M	No permitido	No permitido
	Fricción cortante		420	A615M, A706M, A955M, A996M	A1064M, A1022M	A1064M, A1022M	No permitido
	Estribos, estribos cerrados de confinamiento		420	A615M, A706M, A955M, A996M, A1035M	A1064M, A1022M	A1064M y alambre liso electrosoldado A1022M	No permitido
			550	No permitido	No permitido	A1064M y alambre corrugado electrosoldado A1022M	No permitido
Torsión	Longitudinal y transversal		420	A615M, A706M, A955M, A996M	A1064M, A1022M	A1064M, A1022M	No permitido
Refuerzo de anclajes	Sistemas sísmicos especiales		550	A706M ^[2]	No permitido	No permitido	No permitido
	Otros		550	A615M, A706M, A955M, A996M	A1064M, A1022M	A1064M, A1022M	A184M ^[5]
Regiones diseñadas usando método de puntal-tensor	Tensores longitudinales		550	A615M, A706M, A955M, A996M	A1064M, A1022M	A1064M, A1022M	No permitido
	Otros		420				

^[1] Todos los componentes de muros estructurales especiales, incluyendo vigas de acople y machones de muro.

^[2] Se permite ASTM A615M Grado 420 si se cumple con los requisitos de 20.2.2.5(b).

^[3] En losas y vigas que no sean parte de un sistema especial de resistencia sísmica, las barras que pasan a través o se extienden de un sistema de muros estructurales especiales deben cumplir con 20.2.2.5.

^[4] Refuerzo longitudinal con $f_y > 550$ MPa no se permite en pórticos intermedios a momento o en pórticos ordinarios a momento que resistan demandas sísmicas E .

^[5] Las parrillas de barras corrugadas soldadas se permite que sean ensambladas usando barras corrugadas que cumplen con A615M ó A706M Grado 420 ó Grado 550.

^[6] No se permite usar ASTM A1064M y A1022M en sistemas sísmicos especiales cuando se requiere que la soldadura resista esfuerzos provenientes de confinamiento, soporte lateral de barras longitudinales, cortante u otras acciones.

^[7] Esta aplicación también incluye refuerzo para cortante con un valor máximo de f_y o f_{yt} de 550 MPa para los cálculos de diseño en diafragmas o cimentaciones para combinaciones de carga que incluyan fuerzas sísmicas cuando sean parte de una edificación con un sistema especial de resistencia sísmica.

^[8] El refuerzo de cortante en esta aplicación incluye estribos, estribos cerrados de confinamiento y espirales en pórticos especiales a momento.

^[9] El refuerzo de cortante en esta aplicación incluye todo el refuerzo transversal en muros estructurales especiales, vigas de acople y machones de muro. Las barras diagonales en vigas de acople deben cumplir con ASTM A706M o la nota de pie de tabla [2].

Tabla.II.2. Refuerzo corrugado no preesforzado, Tabla 20.2.2.4.

Fuente: Código ACI 318-19

II.2.2 DISEÑO DE VIGAS A FLEXIÓN POR ACI 318-19

Uno de los objetivos del presente proyecto consiste en calcular la cuantía de acero de refuerzo longitudinal para que la viga pueda resistir momentos.

El procedimiento consiste como primer paso, en obtener la cuantía de acero mínima (ρ_{\min}) y máxima (ρ_{\max}), esto con el fin de estar dentro del rango permitido del área de acero que garantiza un comportamiento adecuado para que la viga flexione. Posteriormente, el siguiente paso consiste en calcular el *Coefficiente de Resistencia a la Flexión* (R_n).

Estos tres parámetros son esenciales ya que el cálculo de cuantía (también llamado porcentaje) de acero ($\rho_{\text{teórico}}$) está en función del Coeficiente de Resistencia a la Flexión (R_n), la Resistencia a la Compresión (f'_c) y la Fluencia del Acero (f_y), y una vez obtenido el valor, este se compara con la cuantía de acero mínimo y máximo con el fin de estar dentro del rango permitido.

Como tercer paso, una vez que se cuenta con la cuantía de acero ($\rho_{\text{teórico}}$) y es validado, se multiplica por el área transversal de la sección de la viga, dando como resultado el área de acero requerida para que la viga esté diseñada por flexión.

Esta área de acero es teórica, ya que en la práctica se debe proponer un área que sea igual o mayor al calculado, por lo que el siguiente paso consiste buscar una combinación de varillas en la cual la suma de sus áreas transversales de un valor ligeramente arriba del calculado, asegurando, por obvias razones, que no sobrepase el área de acero máximo.

Con la cuantía de acero real (ρ_{real}) calculada, se procede a hacer una comprobación de su resistencia, por lo que se recalcula el Coeficiente de Resistencia a la Flexión

(R_n) pero esta vez en función de la cuantía de acero real (ρ_{real}), la Resistencia a la Compresión (f'_c) y la Fluencia del Acero (f_y).

Como último paso, se calcula el valor del Momento Nominal (M_n) ya factorizada y se compara con el Momento último (M_u), corroborando que el Momento Nominal (M_n) sea mayor al Momento último (M_u). De esta forma termina el diseño por flexión. Cada uno de estos valores se detallarán a continuación, usando las especificaciones de como obtenerlas según el Código ACI 318-19 y en función del sistema de unidades que se use.

Para el desarrollo móvil se contempla cada uno de estos valores dando la libertad al usuario de personalizar su diseño, en el capítulo tres del presente proyecto se muestra la lógica, el diseño y la arquitectura empleada para el diseño a flexión y como se involucran todos estos parámetros cuidando la experiencia de usuario y flujo de la misma.

- *Cuantía de Acero mínimo (ρ_{min})*

El Código ACI 318-19 estipula que el porcentaje de acero mínimo tiene el objetivo de que la resistencia a flexión exceda la resistencia de fisuración con un margen apropiado. La viga debe ser capaz de sostener la carga impuesta después del comienzo de la fisuración por flexión y deflexiones visibles, de tal manera que adviertan una posible sobrecarga.

En ocasiones, debido a diseños arquitectónicos o funcionales, se seleccionan dimensiones para las vigas que son mucho mayores que las requeridas sólo por la flexión; por lo tanto, estos miembros requieren en teoría muy poco acero de refuerzo.

En una viga ligeramente reforzada, cabe la posibilidad de que esta falle si el momento resistente último de la sección es menor que su momento de

agrietamiento; consecuentemente, la sección fallará tan pronto como se forme una grieta. Este tipo de falla puede ser repentina y sin advertencia. Para impedir tal posibilidad, el código ACI 318-19, en la sección 9.6.1.2 especifica cierta cantidad mínima de refuerzo que debe emplearse en cada sección de los miembros a flexión donde se requiere esfuerzo de tensión de acuerdo con el análisis, ya sea por momento positivo o negativo. En adición, el valor de la fluencia de acero (f_y) debe limitarse a un máximo de 550 MPa.

A continuación, se muestran las respectivas expresiones matemáticas para calcular la cuantía de acero mínimo (ρ_{\min}) dependiendo del sistema de unidades que se este empleando, dicho valor debe ser el mayor entre a y b.

	<u>SI (MPa)</u>	<u>MKS (kg_f/cm^2)</u>	<u>US units (psi)</u>
a)	$\frac{0.25 \sqrt{f'c}}{f_y}$	$\frac{0.80 \sqrt{f'c}}{f_y}$	$\frac{3 \sqrt{f'c}}{f_y}$
b)	$\frac{1.4}{f_y}$	$\frac{14}{f_y}$	$\frac{200}{f_y}$

- *Cuantía de Acero balanceado (ρ_{bal})*

Se dice que una viga tiene una cuantía de acero balanceada (ρ_{bal}) cuando el acero de refuerzo en la zona de tensión alcanza su punto de fluencia al mismo tiempo que las fibras extremas del concreto en la zona de compresión alcanzan una deformación unitaria igual a 0.003. El riesgo de diseñar una viga con una cuantía de acero balanceada, es decir, que este controlado por su lado de compresión, es que la viga puede presentar una falla de tipo frágil de manera repentina sin previo aviso.

A medida que la carga aumenta en una viga de este tipo, las deflexiones no son visiblemente notorias, aun cuando el concreto queda sometido a esfuerzos de compresión muy elevados y la falla probablemente ocurra sin mostrar ningún tipo de aviso para los usuarios de la estructura. Estos miembros están controlados en compresión y se conocen como miembros frágiles. Es evidente que estos miembros deben ser evitados a toda costa.

La razón por la cual es necesario entender este parámetro es porque a partir de él se puede conocer la cuantía de acero máximo (ρ_{max}), el cual, propicia que el miembro sometido a flexión este controlado por tensión. Además, para el desarrollo de aplicación móvil se da este valor ya que tiene fines didácticos y es importante que el alumno entienda de donde provienen estos parámetros y sus importancias. Figura II.2.

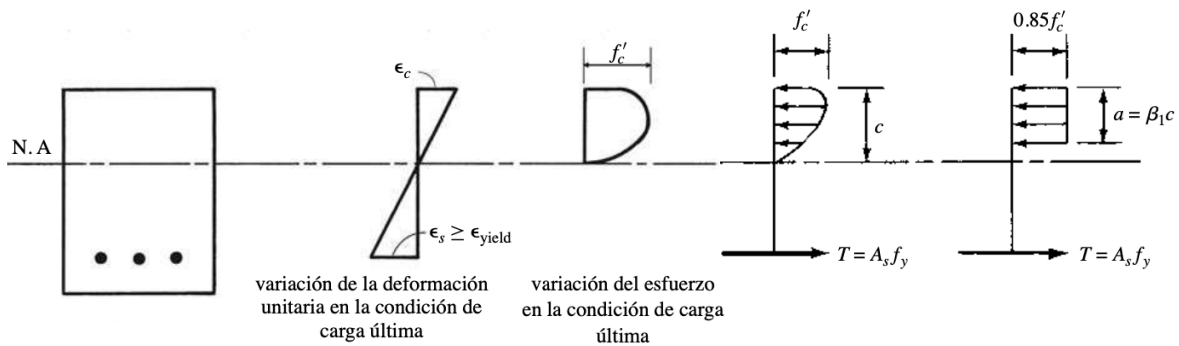


Figura.II.2. Distribución de esfuerzos no lineal en condiciones de carga última.
Fuente: Diseño de Concreto Reforzado, McCormarc (2017)

Los esfuerzos de compresión varían aproximadamente en forma lineal hasta que el esfuerzo máximo es igual a aproximadamente $0.50 f'_c$. No obstante, éste no es el caso cuando los esfuerzos se incrementan. Cuando se alcanza la carga última, las variaciones de las deformaciones unitarias y de los esfuerzos son aproximadamente como se muestra en el diagrama anterior.

Si se supone que el concreto se aplasta bajo una deformación unitaria de aproximadamente 0.003 y el acero cede bajo la fluencia (f_y), es posible obtener fórmulas de flexión para vigas sin necesidad de conocer la distribución exacta de los esfuerzos. Sin embargo, es necesario conocer el valor de la compresión total y su centroide.

Whitney reemplazó el bloque curvo de esfuerzos como se muestra en la figura II.2 por un bloque rectangular equivalente de intensidad igual a $0.85 f'_c$ y una altura $a = \beta_1 c$. El área de este bloque rectangular debe ser igual a la del bloque curvo de esfuerzos y los centroides de los dos bloques deben coincidir.

Como McCormac menciona en su obra *Diseño de Concreto Reforzado* (2017), existen suficientes resultados de pruebas en vigas de concreto como para poder obtener la altura del bloque de esfuerzos rectangular equivalente. Por medio de los valores β_1 dados en el código ACI 318-19 en la Tabla 22.2.2.4.3, se puede obtener este resultado. Tabla II.3.

f'_c , psi	β_1	
$2500 \leq f'_c \leq 4000$	0.85	(a)
$4000 < f'_c < 8000$	$0.85 - \frac{0.05(f'_c - 4000)}{1000}$	(b)
$f'_c \geq 8000$	0.65	(c)

Tabla.II.3. Valores de β_1 , Tabla 22.2.2.4.3
Fuente: Código ACI 318-19

Una vez que se conocen los parámetros anteriores, se puede proceder a obtener la fórmula de la cuantía de acero balanceado (ρ_{bal}), para lo cual, se requiere de un coeficiente que se obtiene a partir de una relación de triángulos con los esfuerzos en las fibras extremas de la sección de la viga.

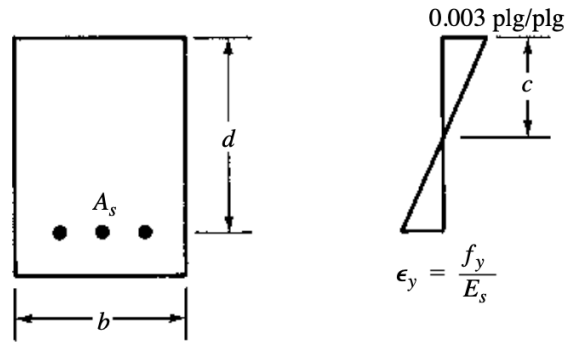


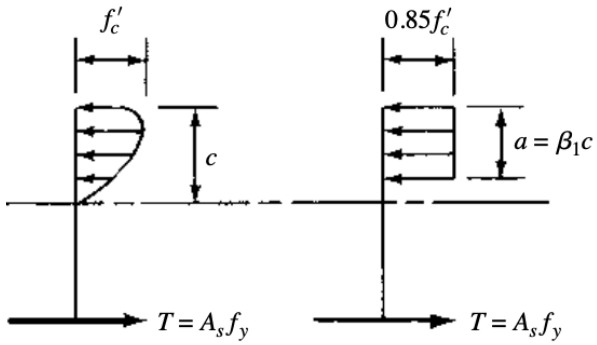
Figura.II.3. Coeficiente ϵ_y
 Fuente: Diseño de Concreto Reforzado, McCormac (2017)

Como se muestra en la figura anterior, se puede conocer el valor de c_b por medio de una relación de triángulos y un respectivo despeje. Teóricamente el concreto alcanzará la deformación unitaria a compresión de 0.003 y el acero cederá simultáneamente.

$$\frac{0.003}{\epsilon_y + 0.003} = \frac{c_b}{d}$$

$$c_b = \frac{0.003}{\epsilon_y + 0.003} d$$

Continuando con la deducción de la fórmula para la cuantía del acero balanceado (ρ_{bal}), se procede a encontrar otra ecuación de C_b pero en términos de cuantía de acero (ρ), para lo cual, se requiere del diagrama de esfuerzos equivalentes que se mostró en la figura II.2, iniciando con la hipótesis de que la fuerza equivalente en la fibra extrema en compresión está equilibrada con la fuerza equivalente en la fibra extrema en tensión.



$$0.85 \cdot f'_c \cdot ab = A_s f_y$$

$$a = \frac{A_s f_y}{0.85 \cdot f'_c \cdot b} = \frac{\rho f_y d}{0.85 \cdot f'_c}$$

donde $\rho = A_s / bd$

$$c = \frac{a}{\beta_1} = \frac{\rho f_y d}{0.85 \cdot \beta_1 \cdot f'_c}$$

∴

$$c_b = \frac{0.003}{\epsilon_y + 0.003} d = \frac{\rho f_y d}{0.85 \cdot \beta_1 \cdot f'_c}$$

$$\rho_{bal} = \left(\frac{0.85 \cdot \beta_1 \cdot f'_c}{f_y} \right) \left(\frac{0.003}{\epsilon_y + 0.003} \right)$$

- *Cuantía de Acero máximo (ρ_{max})*

Se dice que un miembro está controlado por tensión cuando la deformación unitaria en la fibra extrema en tensión es mayor a $(\epsilon_{ty} + 0.003)$, lo que significa que se puede esperar una advertencia clara previa a una falla por flexión y fisuración excesiva, este límite provee suficiente ductilidad en la mayoría de los casos.

Antes del reglamento del 2019, el límite de comportamiento controlado por tensión estaba definido como 0.005, tomando en cuenta un acero de refuerzo Grado 60 (420MPa). El actual Código ACI 318-19 considera que el acero de refuerzo puede ser de mayor resistencia, por lo que el valor recomendado debe ser superior o igual a $(\epsilon_{ty} + 0.003)$.

Tomando en cuenta la expresión matemática para el acero balanceado (ρ_{bal}), se puede sustituir el término ϵ_y por $(\epsilon_{ty} + 0.003)$, de esta manera se estaría diseñando

una viga controlada por tensión y garantizando así un comportamiento dúctil, de tal manera que el miembro al estar sometido a fuerzas que la flexionen avisen con anticipación ante una posible falla.

A continuación, se muestra la fórmula para obtener el porcentaje máximo de acero (ρ_{max}). Cabe mencionar que el valor de 0.003 corresponde a la deformación unitaria del concreto ϵ_{cu} .

$$\rho_{max} = \left(\frac{0.85 \cdot \beta_1 \cdot f'_c}{f_y} \right) \left(\frac{0.003}{(\epsilon_y + 0.003) + 0.003} \right)$$

$$\rho_{max} = \left(\frac{0.85 \cdot \beta_1 \cdot f'_c}{f_y} \right) \left(\frac{\epsilon_{cu}}{(\epsilon_y + 0.003) + \epsilon_{cu}} \right)$$

- *Coefficiente de Resistencia a la Flexión (Rn)*

Este coeficiente se puede expresar de dos formas, la primera corresponde a una ecuación en términos de las dimensiones de la sección, el factor de reducción y el Momento último, esto debido a que son los únicos parámetros que se conocen hasta ese momento.

$$Rn = \frac{Mu}{\phi b d^2}$$

Donde:

Mu = Momento último

ϕ = Factor de reducción de resistencia igual a 0.9 según ACI 318-19, Tabla 21.2.2

b = Base de la sección.

d = Peralte efectivo, distancia del eje extremo a compresión al centroide de las varillas de acero de refuerzo a tensión.

La segunda expresión se emplea una vez encontrada la cuantía de acero real (ρ_{real}), y va a ser esencial para el cálculo del momento nominal (M_n).

$$R_n = \rho \cdot f_y \left(1 - \frac{\rho f_y}{1.7 f'_c} \right)$$

- *Cuantía de Acero Teórico ($\rho_{\text{teórico}}$)*

Este coeficiente nos va a permitir calcular el porcentaje de acero teórico que se necesita para que las fibras extremas de la viga se estén deformando simultáneamente tanto en compresión como en tensión.

Esta ecuación está en términos de la resistencia a la compresión del concreto (f'_c), la fluencia del acero (f_y) y el coeficiente de resistencia a la flexión (R_n). Estos tres parámetros están expresados en unidades de esfuerzo; sin embargo, el resultado de la ecuación es un valor adimensional, el cual, representa el porcentaje de acero de refuerzo equivalente al porcentaje de concreto de toda la sección de la viga de tal manera que resista el Momento último.

$$\rho = \frac{0.85 f'_c}{f_y} \left(1 - \sqrt{1 - \frac{2 R_n}{0.85 f'_c}} \right)$$

- *Momento Nominal (M_n)*

Es el momento de diseño, el cual, la viga puede resistir empleando la cuantía de acero real propuesta. Cabe mencionar, que este momento se debe multiplicar por un factor de reducción igual a 0.9, el cual viene especificado en el código ACI 318-19 tabla 21.2.2, y posteriormente compararlo con el Momento último garantizando que sea mayor a este.

$$M_n = R_n b d^2 \qquad \phi M_n \geq M_u$$

II.2.3 DISEÑO DE VIGAS A CORTENTE POR ACI 318-19

Otro de los objetivos del presente proyecto consiste en calcular la cuantía de acero de refuerzo transversal para que la viga pueda resistir fuerzas cortantes. Se debe recordar que la principal finalidad del diseño de vigas de concreto reforzado es proponer miembros dúctiles que den aviso a fallas inesperadas. Por lo tanto, para lograr esta meta, el Código ACI 318-19 provee valores del cortante de diseño que tienen factores de seguridad mayores contra fallas por cortante que los previstos para fallas por flexión.

El problema de las fallas por cortante en comparación con las de flexión, ocurren repentinamente sin mostrar evidencia previa que la viga esta siendo sometida bajo una carga excesiva. Por consiguiente, las vigas se diseñan para que fallen por flexión bajo este tipo de cargas que son considerablemente menores a aquellas que causarían las fallas por cortante, logrando de esta manera que este tipo de vigas sufran una falla de tipo dúctil. Pueden agrietarse y sufrir grandes deflexiones si se sobrecargan, pero no colapsan como lo harían si la falla por cortante sucediera, a lo que se conoce como una falla de tipo frágil.

El procedimiento consiste primero en definir los límites en los cuales la sección es capaz de resistir las fuerzas cortantes debido a la aportación tanto del concreto ($V_{c_{max}}$) como del acero ($V_{s_{max}}$). Cabe mencionar que para proceder con el análisis es necesario que estos límites no se rebasen, de lo contrario se requeriría que se proponga otra sección de preferencia más aperaltada.

Posteriormente, se calcula el cortante máximo (V_c) que aporta la sección de la viga para lo cual, el código propone tres expresiones matemáticas dependiendo del sistema de unidades que se esté trabajando. En adición, como filtro de seguridad, este valor se ve afectado por un factor que reduce la capacidad de carga en un 25%, siendo este factor según la norma de $0.75 (\phi)$.

Con los primeros parámetros previamente obtenidos, se calcula el área de acero mínimo ($A_{v_{min}}$) que debe llevar la sección y la separación máxima entre los estribos (S_{max}). Estos valores son teóricos, por lo que, el proyectista debe proponer un área de acero que sea ligeramente igual o mayor al calculado, y en el caso de la separación entre estribos un valor ligeramente menor o igual al teórico y que sea práctico para el constructor.

De esta manera, es posible calcular el cortante que aporta la sección en términos del área de acero mínimo y la separación máxima propuesta. Este valor de cortante ya reducido por un factor (ϕV_s) y en conjunto con el valor de cortante que aporta el concreto de igual forma ya reducido (ϕV_c) será válido para ciertas secciones de la viga donde la fuerza de cortante actuante sea de menor magnitud.

Por último, se debe encontrar el área de acero y la separación entre estribos para resistir la fuerza cortante máxima presentada en la viga, por lo que el código ACI 318-19 dicta que hay un punto crítico a una distancia igual al peralte efectivo de la sección separada del apoyo, por ende, no se toma el valor máximo que se ve en el diagrama de cortante, sino uno que corresponda a esa distancia establecida en el código. Esta área de acero y separación entre estribos será válido para resistir valores iguales al cortante máximo y menores.

Dependerá del criterio del calculista proponer otros valores para diferentes secciones de la viga, esto está en función de las magnitudes de las fuerzas cortantes y distancias que mejor convenga para proponer otra separación; no obstante, se debe tomar en cuenta que en la práctica, el constructor va a optar por diseños más funcionales en las cuales el ahorro de tiempo sea más significativo que el económico, y el hecho de proponer diferentes tramos con características diferentes de estribos puede no ser viable para la construcción.

Todos estos parámetros y sus expresiones matemáticas para obtenerlos serán descritos a continuación ya que forman parte también de la funcionalidad de la aplicación móvil, objeto de investigación del presente proyecto.

- *Cortante máximo aportado por el concreto (Vc_{max})*

Este valor no debe ser sobrepasado según el código ACI 318-19 sección 22.5.5.1.1 ya que marca el límite en el cual, el concreto aporta el cortante necesario para que la viga siga teniendo un comportamiento dúctil.

<u>SI (MPa)</u>	<u>MKS (kg_f/cm^2)</u>	<u>US units (psi)</u>
$0.42 \lambda \sqrt{f'c} b_w d$	$1.33 \lambda \sqrt{f'c} b_w d$	$5 \lambda \sqrt{f'c} b_w d$

- *Cortante máximo aportado por el acero (Vs_{max})*

Este valor no debe ser sobrepasado según el código ACI 318-19 sección 22.5.1. En esta sección el código ya contempla la suma del aporte del concreto y del acero con un factor de reducción. Sin embargo, para el presente proyecto se separa la expresión matemática correspondiente al acero.

<u>SI (MPa)</u>	<u>MKS (kg_f/cm^2)</u>	<u>US units (psi)</u>
$0.66 \sqrt{f'c} b_w d$	$2.2 \sqrt{f'c} b_w d$	$8 \sqrt{f'c} b_w d$

- *Cortante que aporta el concreto en la sección (Vc)*

Este valor representa el cortante que aporta el concreto empleando un área de acero mínimo y otra área de acero que resista el cortante máximo. El valor obtenido para este parámetro debe ser menor que el cortante máximo aportado

por el concreto ($V_{c_{max}}$), el cual cumplirá siempre y cuando no haya presencia de fuerza axial en compresión, de lo contrario dependiendo de la magnitud de esta puede no cumplirse este filtro y la sección tendrá que ser rediseñada con otras dimensiones más amplias.

La ecuación para calcular este parámetro se encuentra en la sección 22.5.5.1 y puede expresarse de tres formas diferentes dependiendo del sistema de unidades que se elija. Cabe mencionar que el código ACI 318-19 propone tres formas de obtener este valor en función de las variables que se tengan. No obstante, para el presente proyecto y en el desarrollo de la aplicación móvil, solo se toma en cuenta la que se indica en el inciso a). Figura II.4.

$$\begin{array}{ccc} \underline{\text{SI (MPa)}} & \underline{\text{MKS (kg}_f\text{/cm}^2\text{)}} & \underline{\text{US units (psi)}} \\ \left(0.17\lambda\sqrt{f'_c} + \frac{N_u}{6A_g}\right)b_w d & \left(0.53\lambda\sqrt{f'_c} + \frac{N_u}{6A_g}\right)b_w d & \left(2\lambda\sqrt{f'_c} + \frac{N_u}{6A_g}\right)b_w d \\ \\ & \frac{N_u}{6A_g} \leq 0.05 f'_c & \end{array}$$

Criterio	V_c		
$A_v \geq A_{v,min}$	Cualquiera de los dos	$\left(0.17\lambda\sqrt{f'_c} + \frac{N_u}{6A_g}\right)b_w d$	(a)
		$\left(0.66\lambda(\rho_w)^{1/3}\sqrt{f'_c} + \frac{N_u}{6A_g}\right)b_w d$	(b)
$A_v < A_{v,min}$	$\left(0.66\lambda_s\lambda(\rho_w)^{1/3}\sqrt{f'_c} + \frac{N_u}{6A_g}\right)b_w d$		(c)

Notas:

1. La carga axial, N_u , es positiva para compresión y negativa para tracción.
2. V_c no debe tomarse menor que cero.

Figura.II.4. Cortante que aporta el concreto (V_c), Tabla 22.5.5.1
Fuente: Código ACI 318-19

- *Área de acero mínimo ($A_{v_{min}}$)*

Como se menciona el código ACI 318-19 en la sección 9.6.3.4 se ha demostrado a partir de ensayos incrementar el área mínima de refuerzo a cortante en la medida que la resistencia del concreto aumenta para evitar fallas repentinas de cortante cuando se producen fisuras inclinadas.

A continuación, se muestran un par de fórmulas matemáticas para calcular el área de acero mínimo ($A_{v_{min}}$). El inciso a) hace referencia a un incremento gradual del área mínima del refuerzo transversal en la medida que la resistencia del concreto aumenta, mientras que el inciso b) conlleva a un área mínima de refuerzo transversal independiente de la resistencia del concreto y controlada para resistencias a la compresión del concreto (f'_c) menores a 30MPa.

Puesto que ambas expresiones son válidas, se debe tomar el mayor de estos. El valor obtenido representa el área de acero mínimo por unidad de longitud. Nuevamente, dependiendo de las unidades que se elijan esta expresión está disponible de tres formas.

	<u>SI (MPa)</u>	<u>MKS (kg_f/cm²)</u>	<u>US units (psi)</u>
a)	$\frac{A_{v_{min}}}{s} \geq 0.062 \sqrt{f'_c} \frac{b_w}{f_{yt}}$	$\frac{A_{v_{min}}}{s} \geq 0.2 \sqrt{f'_c} \frac{b_w}{f_{yt}}$	$\frac{A_{v_{min}}}{s} \geq 0.75 \sqrt{f'_c} \frac{b_w}{f_{yt}}$
b)	$\frac{A_{v_{min}}}{s} \geq 0.35 \frac{b_w}{f_{yt}}$	$\frac{A_{v_{min}}}{s} \geq 3.5 \frac{b_w}{f_{yt}}$	$\frac{A_{v_{min}}}{s} \geq 50 \frac{b_w}{f_{yt}}$

- *Separación máxima entre estribos (S)*

El código ACI 318-19 establece en la sección 9.7.6.2.2 que un espaciamiento reducido entre los estribos a lo largo de la viga asegura una transferencia más uniforme de la compresión diagonal del alma de la viga, mejorando su capacidad a cortante. Figura II.5.

V_s requerido	s máximo, mm				
		Viga no preesforzada		Viga preesforzada	
		A lo largo de la longitud	A través del ancho	A lo largo de la longitud	A través del ancho
$\leq 0.33\sqrt{f'_c}b_wd$	El menor de:	$d/2$	d	$3h/4$	$3h/2$
		600 mm			
$> 0.33\sqrt{f'_c}b_wd$	El menor de:	$d/4$	$d/2$	$3h/8$	$3h/4$
		300 mm			

Figura.II.5. Espaciamiento máximo entre estribos (S), Tabla 9.7.6.2.2
Fuente: Código ACI 318-19

- Cortante que aporta el acero en la sección (V_s)

De acuerdo al código ACI 318-19 en el apartado 22.5.8.5.1 se propone la expresión matemática para obtener el cortante que aporta la sección en función de la cuantía de acero y la separación entre estribos.

Este valor debe compararse con el cortante máximo que aporta el acero ($V_{s_{max}}$) para hallar una relación, la cual, dependiendo del resultado, se puede verificar si se cumple con la separación máxima entre estribos como se menciona en la sección 9.7.6.2.2.

El cortante aportado por el acero (V_s) se debe multiplicar por un factor de reducción de 0.75 ya que adicionado con el cortante aportado por el concreto (V_c), igualmente factorizado, el resultado será el cortante nominal (ϕV_n) que debe ser mayor al cortante último (V_u).

$$V_s = \frac{A_v f_y d}{s}$$

- *Cortante Nominal (ϕV_n)*

Este valor es esencial para determinar si la sección pasa por cortante, lo que se busca es que sea mayor al cortante último (V_u) el cual ya está amplificado por factores de seguridad que marca el código ACI 318-19 sección 5.3.

Teóricamente hablando, las vigas se diseñan para que fallen por flexión, por lo que antes de que se llegue a una carga que provoque que la sección colapse por cortante, anteriormente ya falló por flexión.

Se muestra a continuación la ecuación para determinar el cortante por diseño; además, de otras ecuaciones que derivan de esta misma, y que son de utilidad para proponer diferentes separaciones entre estribos u otra área de acero para diferentes cortantes que se presentan a lo largo de la viga. Cabe mencionar que las siguientes expresiones matemáticas serán empleadas para el desarrollo de la aplicación móvil.

$$\phi V_n \geq V_u$$

$$\phi V_n = \phi V_c + \phi V_s$$

$$V_s \geq \frac{V_u - \phi V_c}{\phi}$$

$$\frac{A_v}{s} = \frac{(V_u - \phi V_c)}{\phi f_y d}$$

II.3. NORMA NTC 2021

Para el presente proyecto se va a tomar en cuenta la última versión actualizada de las Normas Técnicas Complementarias del Reglamento de Construcciones para el Distrito Federal (NTC) publicada en Julio del año 2021 ya que forma parte también del desarrollo de la aplicación móvil.

Estas normas son disposiciones técnicas que establecen los requisitos técnicos mínimos para el diseño y construcción de todo tipo de edificaciones con las especificaciones y excepciones que en ellas se indican, esto con el fin de asegurar los siguientes puntos:

- Buen funcionamiento
- Accesibilidad
- Habitabilidad
- Higiene
- Servicios
- Acondicionamiento ambiental
- Evaluación y prevención de emergencias
- Integración al contexto e imagen urbana
- Buen funcionamiento de las instalaciones
- Seguridad estructural de las construcciones de la Ciudad de México.

Como se puede observar, estas normas solo aplican para la Ciudad de México debido a que se basan en materiales propias de la región para las pruebas de laboratorio que, teóricamente, justifican los parámetros y expresiones matemáticas presentes en dichas normas.

Sin embargo, cabe destacar que las Normas NTC 2021 pueden ser empleadas en otras ciudades de la república mexicana en caso de que alguna de ellas no cuente con sus propias normas de construcción, y siempre y cuando también esté establecido por el código reglamentario de dicha ciudad.

Un punto negativo u observación referente a estas normas es que se basan en el código ACI 318-14, el cual, ya no es válido en Estados Unidos debido a que dicha versión tiene parámetros que ya no son recomendados.

II.3.1 GENERALIDADES Y PROPIEDADES MECÁNICAS SEGÚN LAS NORMAS NTC 2021

Como se especificó en el subcapítulo anterior, de igual manera se presentan algunas propiedades mecánicas según las normas NTC 2021, así como sus expresiones matemáticas para calcularlas. A diferencia del código ACI 318-19, en estas normas solo se emplean dos sistemas de unidades, el Sistema Internacional (SI) y el Gravitacional.

<u>Sistema Internacional</u>		<u>Gravitacional</u>	
Fuerza	N (Newton)	Fuerza	kgf (kilogramo fuerza)
Longitud	mm (milímetro)	Longitud	cm (centímetro)
Momento	N - mm	Momento	kgf - cm
Esfuerzo	MPa (megapascal)	Esfuerzo	kgf / cm ²

Referente al concreto, las normas contemplan el uso de dos clases:

- Concreto clase 1, con un peso volumétrico en estado fresco superior a 22 kN/m (2.2 t/m³).
- Concreto clase 2, con un peso volumétrico en estado fresco comprendido entre 19 y 22 kN/m³ (1.9 y 2.2 t/m³).

Como se menciona en las normas NTC 2021, el concreto clase 1 es más denso, más rígido, con menor flujo plástico y menor contracción que el concreto clase 2. Las clases de concreto se distinguen, principalmente, por el valor del módulo de elasticidad que se puede alcanzar con ellos y no por el nivel de resistencia a compresión, siendo más rígidos (mayor módulo de elasticidad) los de clase 1.

Las normas establecen que las estructuras del grupo A o B1 se construyan con concreto clase 1; sin embargo, dejan en libertad a los responsables de obra para que puedan emplear un concreto de clase 2, si a su criterio y bajo su responsabilidad consideran que conviene usarlo para alguna obra de las mencionadas.

Puede darse el caso en que sea muy difícil conseguir el agregado denso para el concreto de clase 1 y los responsables encuentren que el agregado disponible puede dar lugar a una estructura satisfactoria. Se requiere evaluar las propiedades del concreto clase 2 que deseen usar y con ellas justificar en la memoria de cálculo que la estructura se comportará satisfactoriamente; en particular, que se alcanzará la resistencia, durabilidad y rigidez necesaria. En el diseño con concreto clase 2 se requieren incluir los efectos de la mayor deformabilidad y mayor contracción del concreto.

- *Resistencia a la compresión ($f'c$)*

Las normas NTC 2021 en la sección 2.1.2 menciona que los concretos clase 1 tendrán una resistencia $f'c$ igual o mayor a 25 MPa (250 kg/cm²). La resistencia especificada de los concretos clase 2 no será menor a 20 MPa (200 kg/cm²).

En ambas clases deberá comprobarse que el nivel de resistencia del concreto estructural de toda construcción cumpla con la resistencia especificada. En adición, todo concreto estructural debe mezclarse a través de medios mecánicos y dosificarse por peso.

La resistencia a compresión $f'c$ representa una resistencia tal que la probabilidad de que no sea alcanzada es del 10% para los concretos clase 1 y del 20% para los de clase 2, esto se debe a las operaciones de transporte, colocación, compactación, curado, entre otros factores.

Es común que la resistencia del concreto en la estructura sea menor que la obtenida en los cilindros de control, ambas determinadas a la misma edad. Para alcanzar la resistencia de diseño en las estructuras, se toma en cuenta los factores de resistencia, FR, empleados en el diseño estructural.

- *Módulo de Elasticidad (E)*

Es usual que para el diseño estructural no se cuente con información de pruebas de laboratorio, en este caso el módulo de elasticidad (E); no obstante, se pueden emplear las expresiones matemáticas que provee las normas en la sección 2.1.4 para concretos de clase 1 y 2. Estos valores corresponden a una probabilidad de no ser alcanzados de 10% y 20%, respectivamente.

Pueden usarse otros valores para el módulo de elasticidad (E) que estén suficientemente respaldados por resultados de laboratorio. Para el caso de una revisión estructural ya existente, puede aplicarse el módulo de elasticidad determinado en corazones de concreto extraídos de la estructura que formen una muestra representativa de ella.

	<u>MPa</u>	<u>kg_f/cm²</u>
<i>a) Para concretos Clase 1 con agregado grueso calizo.</i>	$4400 \sqrt{f'c}$	$14000 \sqrt{f'c}$
<i>b) Para concretos Clase 1 con agregado grueso basáltico</i>	$3500 \sqrt{f'c}$	$11000 \sqrt{f'c}$
<i>c) Para concretos Clase 2</i>	$2500 \sqrt{f'c}$	$8000 \sqrt{f'c}$

- *Fluencia del Acero (fy)*

Para esta propiedad, la norma NTC 2021 indica en el apartado 2.2 una fluencia del acero tanto para el refuerzo longitudinal como el transversal de 420 MPa (4200 kg/cm²) para elementos estructurales de ductilidad baja, media y alta. Sin embargo, es permitido usar inclusive uno de 550 MPa (5500 kg/cm²).

II.3.2 DISEÑO DE VIGAS A FLEXIÓN POR NTC 2021

Otro de los objetivos del presente proyecto consiste en calcular la cuantía de acero de refuerzo longitudinal de la viga por medio de la norma NTC 2021 para elementos estructurales de concreto reforzado.

A diferencia del código ACI 318-19, el procedimiento para obtener el área de acero por flexión usando las normas NTC 2021 es mucho más sencillo. En la sección 5 de esta norma se encuentran las expresiones matemáticas para calcular la cuantía de acero mínimo, máximo, teórico y el momento resistente para comprobación. Algunos de los parámetros ya tienen valores constantes y otros dependen de condiciones más simples comparado con el código ACI 318-19.

Como primer paso, se deben calcular los límites de cuantía de acero, estos son el mínimo y el máximo que se pueden emplear en la sección de la viga.

El área de acero mínimo se calcula con la ecuación 5.1.18; sin embargo, el valor obtenido no puede ser menor a dos barras corridas de 12.7 mm de diámetro (Número 4), de lo contrario se implementarán dichas varillas como refuerzo a flexión.

$$\begin{array}{ccc} \text{MPa} & & \text{kg}_f/\text{cm}^2 \\ \hline & & \hline A_{s_{min}} = \frac{0.22 \sqrt{f'_c}}{f_y} b d & & A_{s_{min}} = \frac{0.7 \sqrt{f'_c}}{f_y} b d \end{array}$$

La norma indica que para obtener la cuantía de acero máxima para miembros que no deban resistir fuerzas sísmicas será del 90% de la falla balanceada de la sección considerada.

La falla balanceada ocurre cuando el acero llega a su esfuerzo de fluencia y simultáneamente el concreto alcanza su deformación máxima de 0.003 en compresión. Este criterio es general y se aplica a secciones de cualquier forma sin acero de compresión o con él.

MPa	kg _f /cm ²
$\beta_1 = 0.85, \text{ si } f'c \leq 28 \text{ MPa}$	$\beta_1 = 0.85, \text{ si } f'c \leq 280 \text{ kgf/cm}^2$
$\beta_1 = 1.05 - \frac{f'c}{140} \geq 0.65, \text{ si } f'c > 28 \text{ MPa}$	$\beta_1 = 1.05 - \frac{f'c}{1400} \geq 0.65, \text{ si } f'c > 280 \text{ kgf/cm}^2$
$f''c = 0.85 f'c$	$f''c = 0.85 f'c$
$\rho_{bal} = \frac{f''c}{f_y} \left(\frac{600 \beta_1}{600 + f_y} \right)$	$\rho_{bal} = \frac{f''c}{f_y} \left(\frac{6000 \beta_1}{6000 + f_y} \right)$
$\rho_{max} = 0.9 \rho_{bal}$	$\rho_{max} = 0.9 \rho_{bal}$

Para la cuantía de acero teórica se despeja la variable ρ en la ecuación del momento resistente (M_R) 5.1.2 de las normas NTC 2021. Para el factor de resistencia (F_R), la norma establece en la sección 3.7 el valor de 0.9 para flexión.

$$q = \left(1 - \sqrt{1 - \frac{2 M_R}{F_R b d^2 f''c}} \right)$$

$$\rho = q \frac{f_y}{f''c}$$

$$A_s = \rho b d$$

Una vez que se cuenta con el porcentaje de acero teórico, se debe proponer un área de acero que sea igual o ligeramente mayor. Por consiguiente, se puede usar la ecuación 5.1.5 para calcular la cuantía de acero real, y posteriormente, emplearlo en la ecuación de momento resistente (M_R) para determinar la resistencia a flexión de la viga, este valor debe ser igual o mayor al momento último (M_u).

$$\rho = \frac{A_s}{b d}$$

$$q = \frac{\rho f_y}{f''c}$$

$$M_R = F_R b d^2 f''c q (1 - 0.5 q)$$

II.3.3 DISEÑO DE VIGAS A CORTANTE POR NTC 2021

La norma NTC 2021 menciona en la sección 5.3.3 que la resistencia a fuerza cortante de un elemento sujeto a flexión o a flexocompresión será la suma de la fuerza cortante que toma el concreto (V_{CR}), y de la fuerza cortante que toma el acero de refuerzo (V_{SR}).

$$Vu \leq V_{CR} + V_{SR}$$

Como primer paso se calcula la fuerza cortante que puede proporcionar el concreto (V_{CR}), para esto, dependiendo de la cuantía de acero a flexión que posee la viga puede haber dos opciones.

MPa

kg_f/cm²

si $\rho < 0.015$

$$V_{CR} = F_R (0.2 + 20 \rho) 0.30 \sqrt{f'c} b d$$

$$V_{CR} = F_R (0.2 + 20 \rho) \sqrt{f'c} b d$$

si $\rho \geq 0.015$

$$V_{CR} = F_R 0.16 \sqrt{f'c} b d$$

$$V_{CR} = F_R 0.5 \sqrt{f'c} b d$$

El factor F_R para diseño a cortante es igual a 0.75. Una vez obtenido el valor de V_{CR} , se debe corroborar que no sea mayor a la restricción que dicta la norma con la ecuación 5.3.4.

MPa

kg_f/cm²

$$V_{CR} \leq F_R 0.47 \sqrt{f'c} b d$$

$$V_{CR} \leq F_R 1.5 \sqrt{f'c} b d$$

Para el caso del cortante que aporta el acero de refuerzo (V_{SR}), basta con buscar la diferencia entre el cortante último y el cortante que aporta el concreto (V_{CR}).

$$V_{SR} = Vu - V_{CR}$$

Se debe tomar en cuenta que los estribos no serán de un acero con una fluencia f_y mayor a 420MPa (4200 kg/cm²). De igual forma, las varillas no deben tener un diámetro menor a 7.9 mm (número 2.5), y en caso de usarse, la separación no debe exceder la mitad del peralte efectivo ($d/2$).

En la norma NTC 2021 en la sección 5.3.5 se establece la separación entre estribos (S), siempre y cuando se cumplan ciertas condiciones.

a) si $V_{CR} < Vu$

$$S = \frac{F_R Av fy d (\text{sen } \theta + \text{cos } \theta)}{V_{SR}}$$

Donde:

Av = Área transversal del refuerzo para fuerza cortante comprendido en una distancia S

θ = Ángulo que dicho refuerzo forma con el eje de la pieza

V_{SR} = Fuerza cortante de diseño que toma el acero transversal calculada

$$b) \text{ si } \quad \underbrace{V_{CR} < Vu < F_R \cdot 0.47 \sqrt{f'c} b d}_{\text{MPa}} \quad \underbrace{V_{CR} < Vu < F_R \cdot 1.5 \sqrt{f'c} b d}_{\text{kg}_f/\text{cm}^2}$$

$$S = 0.5 d$$

$$b) \text{ si } \quad \underbrace{V_{CR} < Vu > F_R \cdot 0.47 \sqrt{f'c} b d}_{\text{MPa}} \quad \underbrace{V_{CR} < Vu > F_R \cdot 1.5 \sqrt{f'c} b d}_{\text{kg}_f/\text{cm}^2}$$

$$S = 0.25 d$$

Como se observó anteriormente, se debe considerar la separación (S) más pequeña entre el inciso a) y cualquiera de los otros dos incisos, y redondearse ligeramente aún más a un valor más práctico en la construcción. De esta forma se garantiza que el refuerzo de cortante por el acero sea aceptable.

Para el desarrollo de la aplicación móvil se toma este procedimiento y las posibles variables que se puedan presentar, dando como resultado final la comprobación de que la separación entre estribos del acero de refuerzo transversal pasa en función del área de las varillas y la fuerza de cortante que debe resistir.

CAPÍTULO III
DESARROLLO EN ANDROID

III.1. GENERALIDADES DE LA APLICACIÓN MÓVIL

En el capítulo anterior se explicó ampliamente todo lo relacionado al diseño de vigas de concreto reforzado usando el código ACI 318-19 y las normas NTC 2021, ahora se hace referencia a la programación de la aplicación móvil.

Para empezar, se debe explicar y justificar el motivo del porqué se elige el sistema operativo Android destacando el tema del costo, características del sistema operativo y la tecnología a emplear.

A comparación de su competencia directa, que es iOS de la empresa Apple, Android es el líder del mercado a nivel mundial con un porcentaje de usuarios de aproximadamente el 72% que lo consumen. Esto se debe a su amplia gama de modelos de diferentes marcas, dando versatilidad para diferentes gustos. En adición, el costo de un dispositivo Android varía de precios, en contraste con Apple es demasiado bajo, dando la oportunidad a personas de un estatus socioeconómico relativamente bajo poseer un dispositivo móvil.

La principal diferencia entre Android y iOS es su sistema de código. Al estar creado en Linux, el código abierto de Android permite que cualquier programador pueda tener acceso al código fuente y modificar este software de Google. Esto marca una gran diferencia al desarrollar aplicaciones. Por medio del Entorno de Desarrollo Integrado (IDE), se puede trabajar sobre el código con total libertad. El IDE oficial de Android es Android Studio y esta disponible para funcionar en computadoras de Windows y Mac sin ningún costo.

Además, el código abierto (*Open Source* en Inglés) del sistema operativo de Android suele ser más económico y duradero en comparación con otros softwares de código cerrado, como lo es iOS. Esto se debe a que su desarrollo no se capitaliza solo para un autor o empresa, sino que el software pertenece a la comunidad.

En consecuencia de que Android es de código abierto, es muy usual encontrar aplicaciones gratuitas en Google Play, muchas de las cuales están hechas por la misma comunidad. Esta es una de las mejores ventajas de Android desde el punto de vista de los usuarios. Sin embargo, al no haber tantas restricciones a la hora de crear una aplicación para el PlayStore, es posible encontrar aplicaciones de mala calidad, volviendo vulnerable al usuario de ser víctima de un fraude, extorsión, susceptible a exposición de material audio visual no apto para su edad o inclusive descargar un virus.

Ante esta desventaja del sistema operativo Android, los especialistas en control de calidad tienen que invertir más tiempo a probar a fondo el rendimiento y contenido de las aplicaciones Android. Esto se ve reflejado al momento de subir una aplicación a la Play Store, donde algunos casos llegan a tardar hasta dos semanas en ser publicadas en la plataforma.

No obstante, esto no significa que no hayan aplicaciones de Android que sean excelentes y que cumplan con las necesidades, intereses y requisitos de los usuarios. Para el presente proyecto, al estar enfocado a la pedagogía y uso profesional, además de que es tema de tesis a nivel maestría, se deben satisfacer las demandas que puedan exigirle a la aplicación móvil.

La forma en que se va a compartir la aplicación móvil es por medio de un APK (Android Application Package) no firmado, por lo que se puede compartir como cualquier archivo a través de cualquier plataforma digital de mensajería como lo es Whats App, Telegram, Outlook, Gmail, entre otras o por medio de conexión Bluetooth.

En cuanto a la estructura del proyecto, se siguen las buenas prácticas en desarrollo de software que es subdividir la lógica en fragmentos más pequeños para asegurar el mejor rendimiento del dispositivo, en especial por que se trabajan procesos matemáticos muy complejos en base a iteraciones.

Por último, se debe mencionar que toda la aplicación esta codificada en kotlin y empleando la tecnología de JetPack Compose que salió de manera oficial a mediados del mes de Julio del año 2021, y de forma estable a principios del año en curso (2022).

La razón por la cual se opta por esta tecnología es debido a que en un futuro muy cercano, Google dejará de darle soporte a todas la librerías que funcionen con la tecnología Legacy, la cual emplea una forma de programar imperativa por medio de archivos de tipo XML para la parte visual, a diferencia con Compose que es declarativa y todos los componentes visuales están hechos a base de código en el lenguaje Kotlin.

Por estas razones se eligió realizar la aplicación móvil en Android usando la nueva tecnología que es JetPack Compose. En cuestión de marca es más fácil para los estudiantes conseguir un dispositivo Android que un iOS por el costo de adquisición y mantenimiento.

Por el lado de la tecnología, al ser una herramienta de cálculo y diseño de vigas de concreto reforzado, la demanda de desempeño es mayor al hacerse cambios de variables en tiempo de ejecución, por lo que JetPack Compose con la arquitectura MVVM resuelve sin problemas este tema.

Por último, el sistema operativo de Android permite descargar, compartir e instalar aplicaciones provenientes de plataformas de terceros ajenos a la Play Store o de Google con la opción de ser sin costo, lo que permite que esta herramienta sea aún más fácil de adquirir entre la comunidad de ingenieros civiles.

III.2. ARQUITECTURA MVVM

Un buen desarrollo de software implica tener un código limpio y estructurado, dividiendo ciertas partes de la lógica en capas. Una de las principales razones de tener estas buenas prácticas, es que el proyecto queda bien organizado y permite una mejor escalabilidad y mantenimiento del programa; además, es indispensable en el caso de trabajar en equipo, ya que permite con facilidad la lectura y entendimiento del proyecto.

Al tener conocimiento de todos los módulos que conforman una aplicación, al igual de la lógica que hay detrás de cada una de ellas, es más fácil agregar y eliminar funciones al desarrollo. El crear un programa computacional no es una tarea sencilla, ya que dependiendo de la complejidad, esta puede crecer y volver confuso el como se acoplan los módulos que realizan ciertas tareas del programa. Por esta razón, es como surgen los patrones arquitectónicos, que sirven para dividir todo el proyecto por capas, indicando la estructura, funcionamiento e interacción entre las partes de un software.

Referente al desarrollo móvil, en específico Android, existen tres patrones arquitectónicos que son los más usados para crear aplicaciones funcionales y de calidad:

- MVVM → Model - View - ViewModel (*Modelo - Vista - Vista de Modelo*)
- MVC → Model - View - Controller (*Modelo - Vista - Controlador*)
- MVP → Model - View - Presenter (*Modelo - Vista - Presentador*)

Si bien todas estas arquitecturas varían un poco en sus detalles y funcionamiento, no dejan de ser muy similares. Todas tienen un mismo objetivo, la separación de

responsabilidades; cada una de ellas logra esta separación al dividir el software en capas o componentes.

Actualmente, la arquitectura recomendada por Google y que se acopla bien, en especial con JetPack Compose, es la arquitectura MVVM. Consiste prácticamente en dividir el modelo de negocio (Modelo) de la interfaz de usuario (Vista) usando como intermediario al ViewModel el cual, almacena la información durante el ciclo de vida de la aplicación. Figura III.1.

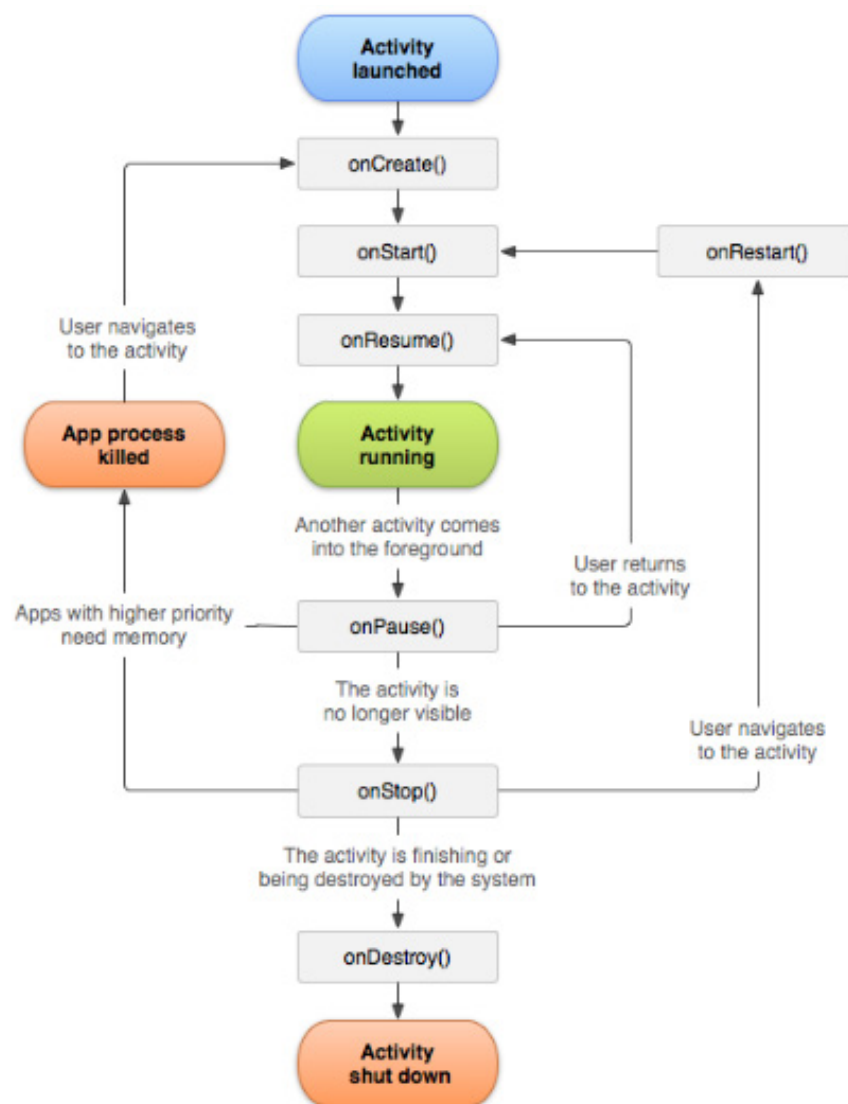


Figura.III.1. Ciclo de vida de una aplicación Android
Fuente: www.developer.android.com

Cabe mencionar que esta información sobrevive a cambios de configuración como lo es el idioma, rotación de pantalla, enviar la aplicación a segundo plano, entre otras; siempre y cuando el ViewModel sea empleado en la actividad donde se requiere persistir la información. Figura III.2.

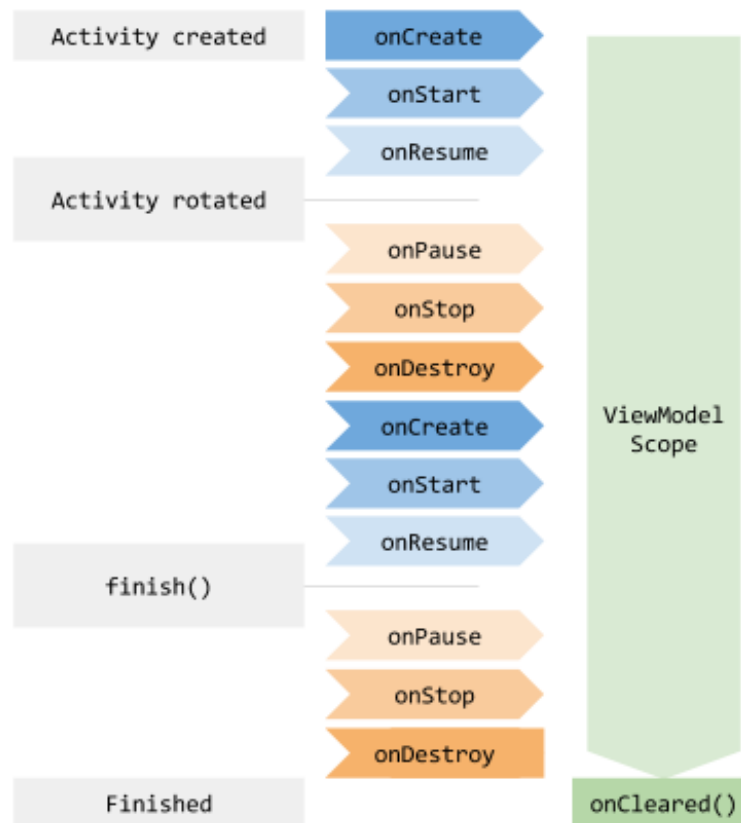


Figura.III.2. Ciclo de vida del ViewModel
Fuente: www.developer.android.com

A continuación se presentan las características de cada capa que conforma la arquitectura MVVM, así como su diagrama.

- **Model (Modelo):** Esta capa es la responsable de la abstracción de los objetos que se usan en la aplicación, ya que contiene los atributos y métodos

de dichos objetos. Se comunica directamente con el ViewModel para almacenar información de cada objeto que se esté creando.

- **View (Vista):** El objetivo de esta capa es informar al modelo de vista sobre la acción del usuario al interactuar con las diferentes vistas. Esta capa observa el ViewModel y no contiene ningún tipo de lógica de aplicación.
- **ViewModel (Vista de Modelo):** Expone los flujos de datos que son relevantes para la vista. En adición, sirve como enlace entre el Modelo y la Vista.

En general, la ventaja que tiene MVVM es que automatiza la comunicación entre las vistas (La capa View) y el Modelo. El ViewModel solo se encarga de almacenar la información proveniente del Modelo conforme la aplicación se está ejecutando sin que se pierda ante cualquier cambio de configuración, y las vistas solo observan esos cambios y actualizan sus estados. Figura III.3.

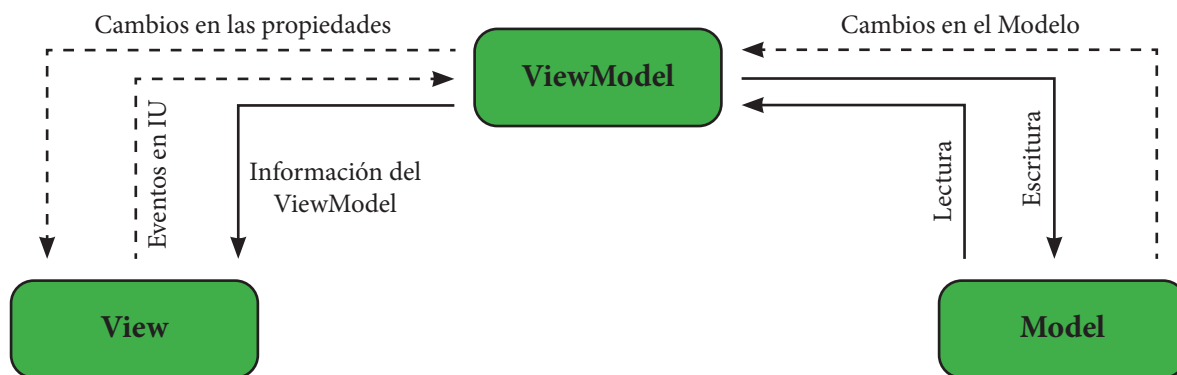


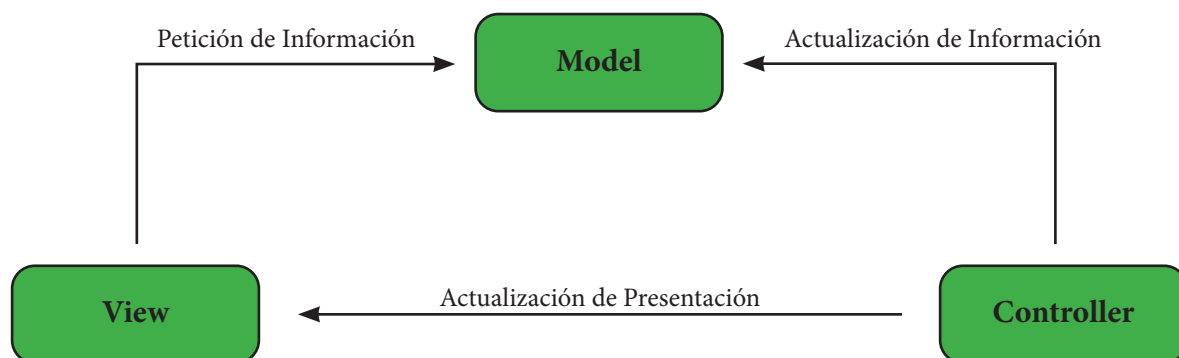
Figura.III.3. Arquitectura MVVM
Fuente: www.developer.android.com

III.2.1. COMPARATIVA CON MVC Y MVP

A pesar de que MVVM es el patrón de arquitectura recomendado por Google, anteriormente se creaban las aplicaciones empleando MVC, MVP y en el peor de los casos ninguno.

Uno de los primeros patrones de arquitectura en usarse en el desarrollo Android fue MVC (Model - View - Controller). Para proyectos pequeños resulta ser útil, pero presenta ciertas limitaciones al momento de escalar la aplicación. Similar a MVVM, separa la aplicación en tres responsabilidades, siendo la capa Model y View la misma que MVVM. La tercera capa es el Controller (Controlador), se encarga de estar al pendiente de las interacciones que tiene el usuario con diferentes vistas, por ejemplo un botón.

Cuando el usuario presiona un botón, el controlador decide cómo interactuar con el modelo correspondiente. Según el cambio de datos en el modelo, el controlador decide si actualizar el estado de la vista o no, según considere apropiado. En una aplicación Android, el controlador casi siempre está representado por una Activity o un Fragment. Figura III.4.

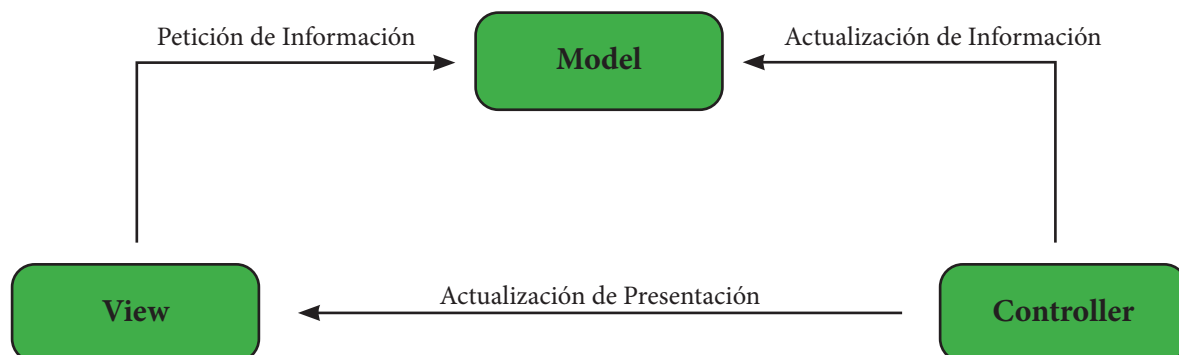


*Figura.III.4. Arquitectura MVC
Fuente: www.developer.android.com*

Algunos de los inconvenientes que tiene MVC son:

- Pruebas: El controlador está vinculado a las APIs de Android, por lo que resulta difícil hacer pruebas unitarias.
- Modularidad y Flexibilidad: Los controladores están acoplados a las vistas (por ejemplo un botón o un texto) que se consideran una extensión de las mismas. Si se cambia una vista, prácticamente se debe cambiar también el controlador.
- Mantenimiento: Conforme la aplicación crece, también la lógica. Por lo tanto, cuando la vista se extiende de igual forma lo hace el controlador. Esto dificulta el darle mantenimiento y puede provocar que ciertas líneas de código colapsen la aplicación.

Debido a estas limitaciones, surge la necesidad de crear un mejor patrón de arquitectura, en este caso MVP (Model - View - Presenter). Las capas de la Vista y el Modelo siguen siendo las mismas, la tercera capa que es el Presentador no está vinculado a una implementación específica de la Vista, solo a una interfaz. Esto resuelve los problemas de llevar a cabo pruebas unitarias, así como el tema de modularidad y flexibilidad que se tenía con MVC. Figura III.5.



*Figura.III.5. Arquitectura MVP
Fuente: www.developer.android.com*

Una desventaja que tiene MVP es la siguiente:

- **Mantenimiento:** Los Presenters, al igual que los Controllers, son propensos a recopilar lógica de negocio conforme la aplicación va creciendo. En proyectos de larga duración, se requieren de muchas interfaces lo que lo vuelve repetitivo y difíciles de separar cuando ya son muy extensos.

Por esta razón, se recomienda el uso de MVVM, porque resuelve los problemas de sus dos antecesores. MVVM automatiza las vistas en su comportamiento, permitiendo que solo observen los cambios que hay en ViewModel y solas deben de actualizarse.

Una de las técnicas para lograr este objetivo con MVVM, es usando DataBinding; sin embargo, esto era con la tecnología “antigua” que es Legacy (Usando archivos XML para las Vistas). Actualmente y como está desarrollada la aplicación, se emplean valores mutables en el ViewModel y se hace la referencia del ViewModel en el constructor de cada composable. Esto facilita el desarrollo ya que solo se debe hacer referencia del objeto, en este caso al ViewModel para poder acceder a sus variables y funciones.

La navegación de las pantallas con JetPack Compose es más sencilla, ya que solo se muestran los composables como pantallas completas y todas están dentro de un NavHost, por lo que solo se requiere de una Actividad. Esto significa que para el presente proyecto, solo se crea un ViewModel del cual se enlazan todos los composables que requieren compartir la información conforme se va ejecutando el cálculo estructural, los diagramas de cortante y momento, y el diseño de la sección, permitiendo el diseño de pestañas para mostrar los tres apartados.

III.2.2. MODEL DE LA APLICACIÓN

En cuanto a la capa Model de la aplicación móvil, se crean dos objetos los cuales son:

- Barra
- Nodo

Estos dos objetos tienen sus modelos por separado con los respectivos atributos y métodos que los conforman. Como buena práctica de programación en Android con el lenguaje Kotlin, se emplea un “data class” el cual sirve para introducir en el constructor todos los atributos que conforman al objeto. A continuación se muestra el pseudocódigo de como está construido cada objeto. Figura III.6.

```
data class Barra(  
    val variable_1 : MutableState<String>,   
    val variable_2 : APOYO,  
    .  
    .  
    .  
    val variable_3 : MutableState<Double>,  
) {  
    fun metodo_1(): Double {  
        val aux = coef_poisson.value.toDouble()  
        val aux1 = (2.0 * (aux + 1.0))  
        val G = elasticidad.value.toDouble() / aux1  
        return G  
    }  
  
    fun coefCortante() {  
        val G = moduloCortante()  
        val aux1 = (formula_del_auxiliar_1)  
        val aux2 = (formula_del_auxiliar_2)  
        val psi = aux1 / aux2  
    }  
}
```

Objeto Barra

Variable de tipo String. Estas variables almacenan atributos que dan esencia al objeto barra

Los métodos pueden retornar resultados que derivan de operaciones matemáticas donde las variables son los atributos del objeto Barra.

Los métodos pueden no retornar resultados. Se encargan de cambiar las variables del objeto barra conforme la aplicación se ejecuta en tiempo real.

Figura.III.6. Pseudocódigo del Model Barra
Fuente: Aplicación VigaLab RC. Autoría propia (2022).

Como se observa en la figura anterior las variables son de tipo mutable ya que, así lo requieren los elementos composables, como lo es un texto, caja de texto, etc, para poder modificar o mostrar el valor conforme se hacen cambios de parámetros en la viga. Estos atributos incluyen la elasticidad, área, longitud, inercia, entre otros.

Se crea el objeto barra ya que el elemento estructural puede estar conformado por una o más de estas, esto debido al procedimiento del método de rigideces. Se instancia una lista de tipo barra y cada uno de estos elementos tiene sus propios atributos y métodos. Los métodos del objeto incluyen la formación de la matriz de rigidez, código de barras, matriz de grados de libertad, y más.

El objeto nodo es esencial para subdividir la barra y llega a ser un papel importante en la creación de los diagramas de cortante y momento, esto debido a que definen los tramos de como esta segmentado cada diagrama. Además, para el cálculo estructural, cada uno de ellos cuenta con atributos correspondientes a los grados de libertad, dependiendo de estos se puede determinar el tipo de apoyo que tiene cada extremo de la barra.

A comparación con el objeto barra, el nodo solo cuenta con dos métodos referentes a definir el tipo de apoyo y se acopla con un “enum class” para clasificarlos. Figura III.7.

```

enum class APOYO {
    NINGUNO,
    SIMPLE_VERTICAL,
    SIMPLE,
    FIJO_DESPLAZABLE_ORTOGONAL,
    ARTICULACION,
    FIJO_DESPLAZAMIENTO_VERTICAL,
    FIJO_DESPLAZAMIENTO_HORIZONTAL,
    EMPOTRAMIENTO,
}

```

Enumerador para clasificar cada tipo de apoyo.

```

data class Nodo(
    var x: Double,
    var y: Double,
    val nodo_x: MutableState<Boolean>,
    val nodo_y: MutableState<Boolean>,
    val nodo_z: MutableState<Boolean>,
    val carga_x: MutableState<Double>,
    val carga_y: MutableState<Double>,
    val carga_z: MutableState<Double>
){

```

Objeto Nodo.

```

    fun tipoApoyo(x: Boolean, y: Boolean, z: Boolean): APOYO {
        var apoyo: APOYO = APOYO.NINGUNO
        if (!x && !y && !z) {
            apoyo = APOYO.NINGUNO
        }
        if (x && !y && !z) {
            apoyo = APOYO.SIMPLE_VERTICAL
        }
        .
        .
        .
        if (!x && y && z) {
            apoyo = APOYO.FIJO_DESPLAZAMIENTO_HORIZONTAL
        }
        if (x && y && z) {
            apoyo = APOYO.EMPOTRAMIENTO
        }
        return apoyo
    }

```

Método que recibe de parámetro un valor de tipo booleano para cada grado de libertad en X, Y y Z. Retorna el tipo de apoyo.

```

    fun codigoDesplazamiento(apoyo: APOYO): Array<Int> {
        var Cd: Array<Int> = emptyArray()
        when (apoyo) {
            APOYO.NINGUNO -> Cd = arrayOf(1, 1, 1)
            APOYO.SIMPLE_VERTICAL -> Cd = arrayOf(0, 1, 1)
            .
            .
            .
            APOYO.FIJO_DESPLAZAMIENTO_HORIZONTAL -> Cd = arrayOf(1, 0, 0)
            APOYO.EMPOTRAMIENTO -> Cd = arrayOf(0, 0, 0)
        }
        return Cd
    }
}

```

Método que devuelve un arreglo de tipo Integer para la formación de la matriz de grados de libertad.

Figura.III.7. Pseudocódigo del Model Nodo
Fuente: Aplicación VigaLab RC. Autoría propia (2022)

III.2.3. VIEW DE LA APLICACIÓN

La capa View se encarga de todos los elementos visuales con los cuales el usuario puede interactuar. Jetpack Compose permite crear objetos genéricos que pueden ser usados de forma repetitiva en diferentes partes de la aplicación, permitiendo que sea más fácil dar mantenimiento a la aplicación y escalarla en un futuro. Los composables genéricos que conforman a la aplicación son:

- **TextField (Caja de Texto):** Para ingresar los valores que va a tener cada atributo de los objetos previamente definidos.
- **Botón:** Con el parámetro `texto` para cambiar el contenido y su respectivo `onClick Listener` de tipo lambda.
- **CheckBox:** Para definir valores booleanos (falso o verdadero) en la definición de tipo de apoyo.
- **DropDown Menus:** Para escoger las unidades en cada procedimiento matemático.

Cabe destacar que por la simplicidad de cada uno de estos objetos composables en lo que se refiere a su función, no es necesario crear un model para cada uno de ellos y mucho menos aplicar principios SOLID, ya que su única tarea es recibir el valor ingresado por el usuario y asignarlo en el atributo correspondiente.

En la siguiente figura se muestra como se crea uno de estos componentes genéricos, el cual, se trata del componente de `CheckBox`. El procedimiento para programar cualquier componente genérico es la mismo. Figura III.8.

```

@Composable
fun TMCheckBox(
    checked: MutableState<Boolean>,
    label: String,
    modifier: Modifier = Modifier,
    enabled: Boolean = true,
    icon: Painter
) {
    val colors = CheckboxDefaults.colors(
        checkedColor = TMRed,
        uncheckedColor = TMGray,
        checkmarkColor = TMWhite,
        disabledColor = TMGray
    )

    Row(
        modifier = modifier.height(48.dp),
        verticalAlignment = Alignment.CenterVertically
    ) {
        Checkbox(
            checked = checked.value,
            onCheckedChange = {checked.value = it},
            enabled = enabled,
            colors = colors
        )
        Text(
            text = label,
            color = TMGray,
            lineHeight = 18.sp,
            style = MaterialTheme.typography.body2,
            textAlign = TextAlign.Justify
        )
        Spacer(modifier = Modifier.width(8.dp))
        Image(
            painter = icon,
            contentDescription = "grado de libertad",
            colorFilter = ColorFilter.tint(TMGray)
        )
    }
}

```

Figura.III.8. Función Composable Botón
Fuente: Aplicación VigaLab RC. Autoría propia (2022)

En cuanto a la interfaz de usuario, la aplicación móvil esta conformada principalmente por dos pantallas, las cuales también son composables dentro de un “NavHost”, que se idealiza como una ventana la cual va a mostrar los elementos que se le soliciten, dando la impresión de que se está navegando entre pantallas.

La primer pantalla está conformada por elementos genéricos en su mayoría por TextFields (cajas de texto) en donde se introducen las propiedades mecánicas y geométricas de cada barra. Figura III.9.



*Figura.III.9. Interfaz de usuario correspondiente a propiedades generales de una viga
Fuente: Aplicación VigaLab RC. Autoría propia (2022).*

La segunda pantalla representa un reto en programación, ya que se diseña para tener en la parte superior una barra de pestañas, conocido como TabWidget, del cual, las tres partes más importantes y que le dan sentido a la aplicación se conectan a un mismo ViewModel donde se comparten la información referente al cálculo y diseño de la viga. Figura III.10.

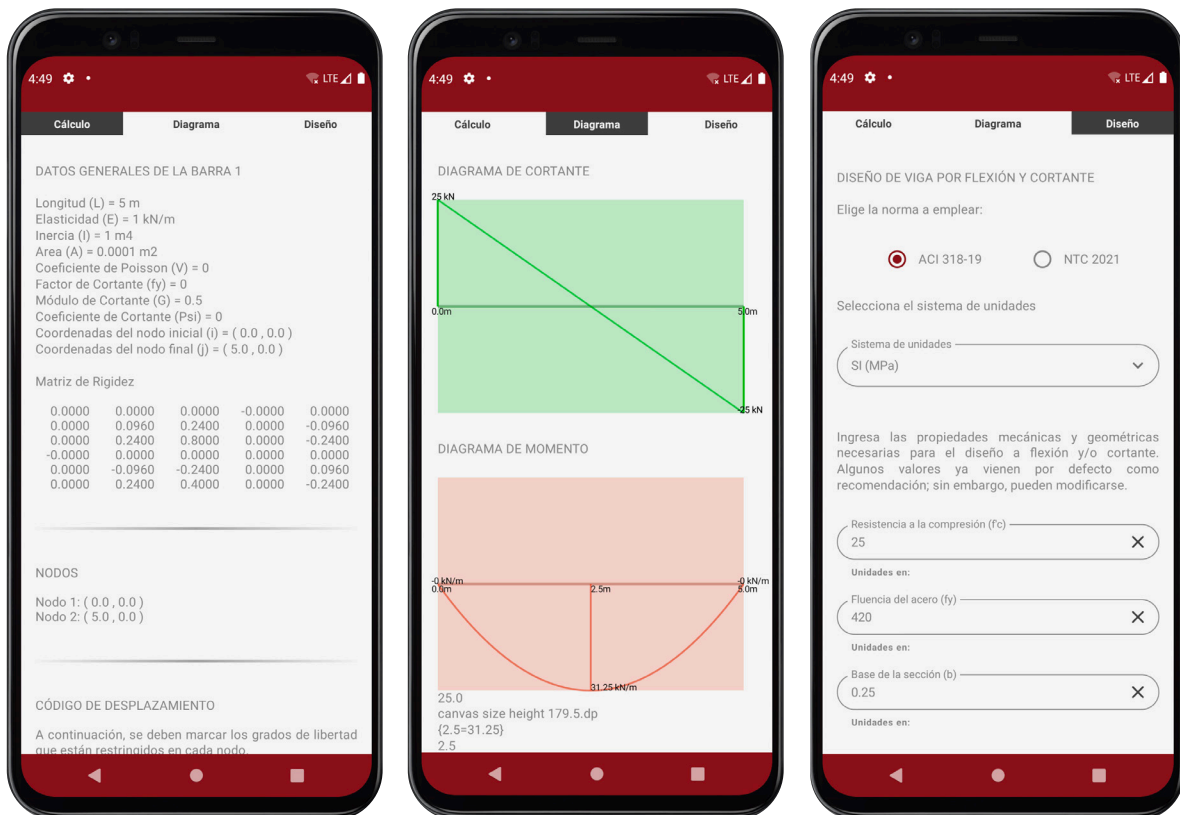


Figura.III.10. Interfaz de usuario correspondiente al cálculo y diseño de una viga
Fuente: Aplicación VigaLab RC. Autoría propia (2022).

III.2.4. VIEWMODEL DE LA APLICACIÓN

La capa del ViewModel funciona como un puente entre la vista (View) y el modelo de negocio (Model), se encarga de almacenar información y proveer la misma cuando sea necesario. Para el presente proyecto, el ViewModel posee datos que comparte entre las pestañas de cálculo, diagrama y diseño, de los cuales, los más relevantes son las matrices que se requieren en todo el proceso del cálculo estructural, la creación de la lista Barras y Nodos, y por último las variables para el diseño a flexión y cortante de la viga.

Al inicio del ViewModel se declaran las matrices bidimensionales que se requieren para el cálculo matricial. Como se puede observar en la siguiente imagen, algunas de ellas son de tipo Integer solo para especificar grados de libertad y otros son de tipo Double para hacer operaciones matriciales con números decimales de doble precisión. Figura III.11.

```
class MyViewModel : ViewModel() {  
  
    /**Este apartado corresponde al cálculo por el método de rigideces*/  
    lateinit var Cd: Array<IntArray>  
    lateinit var En: Array<IntArray>  
    lateinit var Cb: Array<IntArray>  
    lateinit var Eb: Array<IntArray>  
    lateinit var K: Array<DoubleArray>  
    lateinit var f_0_grados: Array<DoubleArray>  
    lateinit var f_c_grados: Array<DoubleArray>  
    lateinit var f_totales: Array<DoubleArray>  
    lateinit var K_inversa: Array<DoubleArray>  
    lateinit var K_inversa_compacta: Array<DoubleArray>  
    lateinit var desplazamientos: Array<DoubleArray>  
}
```

*Figura.III.11. ViewModel, matrices bidimensionales
Fuente: Aplicación VigaLab RC. Autoría propia (2022).*

Posteriormente, se instancian dos variables de tipo lista mutable para que se agreguen o remuevan barras y nodos a través de los métodos que le corresponde a cada objeto. Figura III.12.

```

/** Este apartado corresponde a la lista de barras que conforman a la viga completa */
private val _barras = mutableStateListOf<Barra>(
    Barra(
        mutableStateOf(""),
        mutableStateOf(""),
        APOYO.NINGUNO,
        APOYO.NINGUNO,
        .
        .
        .
        mutableStateOf(0.0),
        reac_x_i = mutableStateOf(0.0),
        reac_y_i = mutableStateOf(0.0),
        reac_z_i = mutableStateOf(0.0),
        reac_x_j = mutableStateOf(0.0),
        reac_y_j = mutableStateOf(0.0),
        reac_z_j = mutableStateOf(0.0),
        distributeLoadList = mutableListOf(0.0),
        punctualLoadList = mutableMapOf(0.0 to 0.0)
    )
)
val barras: List<Barra> = _barras

fun agregarBarra(nuevaBarra: Barra) {
    _barras.add(nuevaBarra)
}

fun removerBarra() {
    _barras.remove(_barras[_barras.size - 1])
}

/** Este apartado corresponde a la lista de Nodos que conforman cada sección de la viga completa */
private val _nodos = mutableList<Nodo>(
    Nodo(
        x = 0.0,
        y = 0.0,
        nodo_x = mutableStateOf(false),
        nodo_y = mutableStateOf(false),
        nodo_z = mutableStateOf(false),
        carga_x = mutableStateOf(0.0),
        carga_y = mutableStateOf(0.0),
        carga_z = mutableStateOf(0.0),
    )
)

```

Figura.III.12. ViewModel, lista de Barras y Nodos
Fuente: Aplicación VigaLab RC. Autoría propia (2022).

Por último, en la tercera parte del contenido del ViewModel se almacenan las propiedades necesarias para el diseño a flexión y cortante por medio del código ACI 318-19 y las normas NTC 2021.

De igual forma, se instancia la variable encargada de definir el sistema de unidades, dependiendo de la opción que se elija, se muestran las fórmulas correspondientes a dicho sistema. Figura III.13.

```
/** Propiedades para el diseño */  
val fc = mutableStateOf("0.0")  
val fy = mutableStateOf("0.0")  
val b = mutableStateOf("0.0")  
val d = mutableStateOf("0.0")  
val ety = mutableStateOf("0.002")  
val ecu = mutableStateOf("0.003")  
val Mu = mutableStateOf("0.0")  
val Vu = mutableStateOf("0.0")  
  
val normSelected = mutableStateOf(0)  
val selectedUnits = mutableStateOf( "" )  
  
}
```

*Figura.III.13. ViewModel, variables para diseño a flexión y cortante
Fuente: Aplicación VigaLab RC. Autoría propia (2022).*

III.3. LÓGICA

La aplicación implementa para el cálculo matricial el método de las rigideces, subdividiendo todo el procedimiento en módulos, cada uno de ellos dependiendo del anterior. Al estar en Compose, cada apartado del cálculo se actualiza ante cualquier cambio de variable en cuanto aparece en pantalla, obligando al usuario a llegar al final del cálculo para obtener los resultados definitivos.

Una vez obtenidos los resultados más actualizados, estos al estar almacenados en el ViewModel, se pueden compartir en otros composables, en este caso con el componente que muestra los diagramas de Cortante y Momento. Se genera un algoritmo que está en función de la cantidad de barras que conforman a la viga y con los resultados del cálculo matricial. Se emplea de manera nativa la librería de Canvas para generar los diagramas. Cabe destacar que toda la aplicación está hecha sin emplear ningún tipo de librería externa.

Los algoritmos responsables de generar los diagramas, también calculan los momentos y cortantes máximos. Esto es de utilidad, ya que se requieren estos resultados para el apartado correspondiente al diseño de la viga.

Ya que se calcula el cortante y momento máximo, se puede proceder al diseño de la viga, el cual, el objetivo es encontrar la cuantía de acero necesaria para que el miembro estructural pase por flexión y cortante.

En caso de requerir otro cálculo para una viga diferente, se puede regresar el usuario a la pantalla donde se ingresan los datos preliminares, que incluyen el módulo de elasticidad (E), inercia (I), longitud (L), área (A), etc. Y de nuevo realizar todo el procedimiento de cálculo y diseño.

III.3.1. APARTADO DEL CÁLCULO MATRICIAL

Para el cálculo estructural, se emplea el método de las rigideces dividiendo dicho método en secciones, cada una de ellas con un texto descriptivo de lo que se está haciendo.

Como se describió en el subcapítulo anterior, una vez que el usuario ingresa las propiedades mecánicas y geométricas principales, se procede a realizar el cálculo empezando con una breve descripción general de cada barra que entrega la viga con su respectiva matriz de rigidez. Figura III.14.

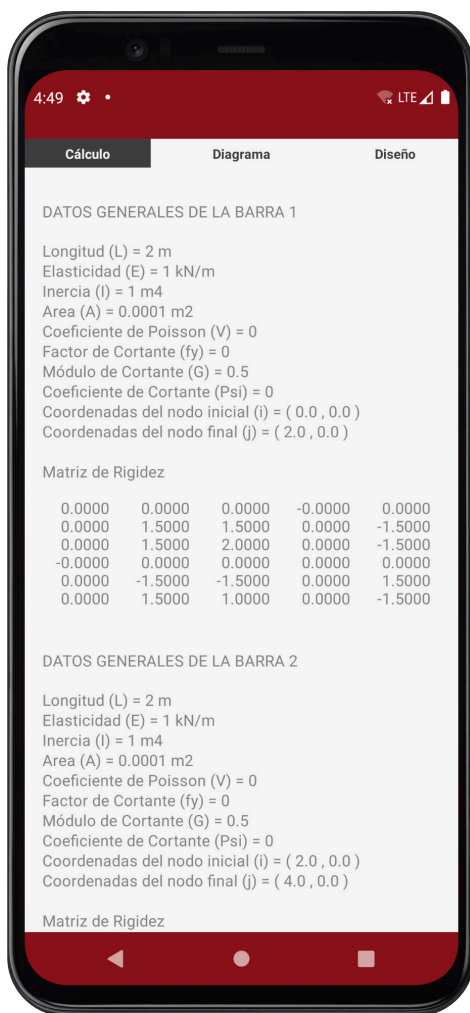


Figura.III.14. Datos Generales y Matriz de Rigidez
Fuente: Aplicación VigaLab RC. Autoría propia (2022).

El segundo paso consiste en mostrar los nodos totales a lo largo de la viga. En el caso de contar con dos o más barras, el nodo que une el extremo final e inicial de dos barras consecutivas se toma como uno solo. De esta forma se simplifica la información y los nodos se van apilando en una lista que lo administra el ViewModel.

Esta lista es de mucha utilidad para el cálculo ya que se le permite al usuario ingresar fuerzas en los nodos en caso de que sea necesario, y para el apartado de los diagramas donde se emplean para poder segmentar dichos diagramas. Figura III.15.



Figura.III.15. Nodos totales en la viga
Fuente: Aplicación VigaLab RC. Autoría propia (2022).

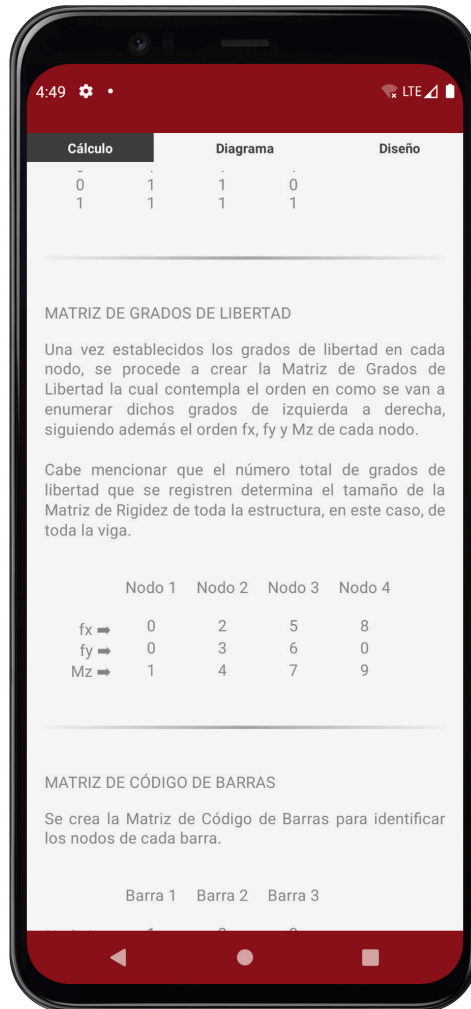
Una vez que se definen los nodos, se obtienen de la lista que se crea en el ViewModel y se emplean para generar una vista deslizable con las posibles combinaciones de restricciones en los grados de libertad que puede tener cada nodo.

En el mismo código, aprovechando la implementación de la lista de nodos, se construye la matriz de “Código de Desplazamiento”. El número de filas hace referencia a las grados de libertad en X, Y y Z, asignando el valor de cero cuando está restringido y uno cuando es libre; mientras que el número de columnas representa cada nodo. Figura III.16.



Figura.III.16. Código de desplazamiento
Fuente: Aplicación VigaLab RC. Autoría propia (2022).

En la siguiente parte del cálculo, dependiendo de las restricciones de grados de libertad de cada nodo, se enumeran y de esta forma se identifica la cantidad de grados de libertad que existen, y por consiguiente el tamaño de la matriz de rigidez de toda la viga. Figura III.17.



*Figura.III.17. Matriz de grados de libertad
Fuente: Aplicación VigaLab RC. Autoría propia (2022).*

Las dos matrices que continúan solo se usan como ayuda visual para identificar los grados de libertad que existen en cada barra. Como proceso final de la primera parte del cálculo, se despliega la matriz de rigidez de toda la viga. Figura III.18.

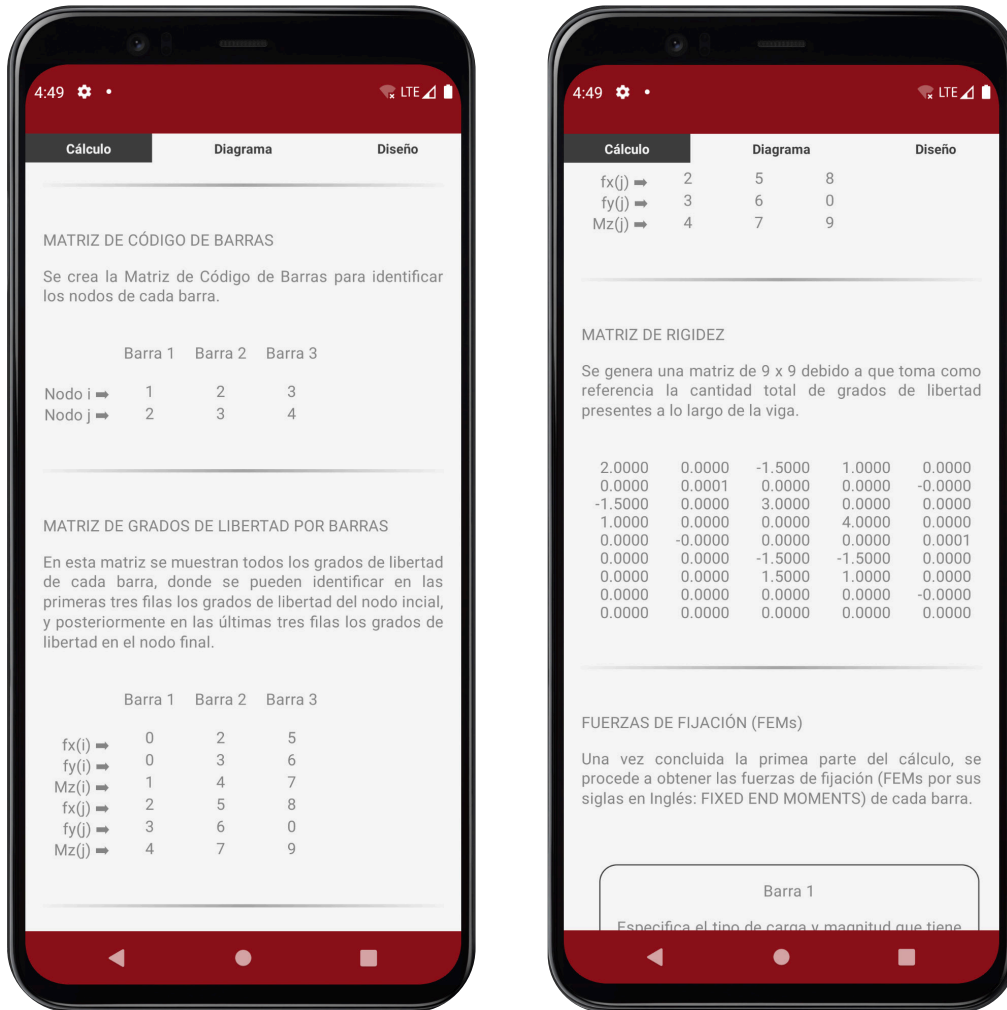


Figura.III.18. Matriz de código de barras y de grados de libertad por barras
 Fuente: Aplicación VigaLab RC. Autoría propia (2022).

La segunda parte del cálculo consiste en especificar las cargas puntuales y/o distribuidas de cada barra, estos se conocen en el cálculo matricial como “Fuerzas de Fijación” (Fixed-End Moments en Inglés). Se diseñó de manera muy intuitiva explicando lo que el usuario debe ingresar, y en caso de cometer algún error al registrar los datos, estos se pueden resetear limpiando toda carga que esté presente en el miembro.

De igual forma, como ayuda visual se muestra la suma de todas las reacciones que se van registrando en cada grado de libertad de la barra. Consecuentemente, se despliega el vector de reacciones de todos los grados de libertad en tiempo real cada vez que el usuario ingresa nuevas cargas en cualquiera de las barras. Figura III.19.

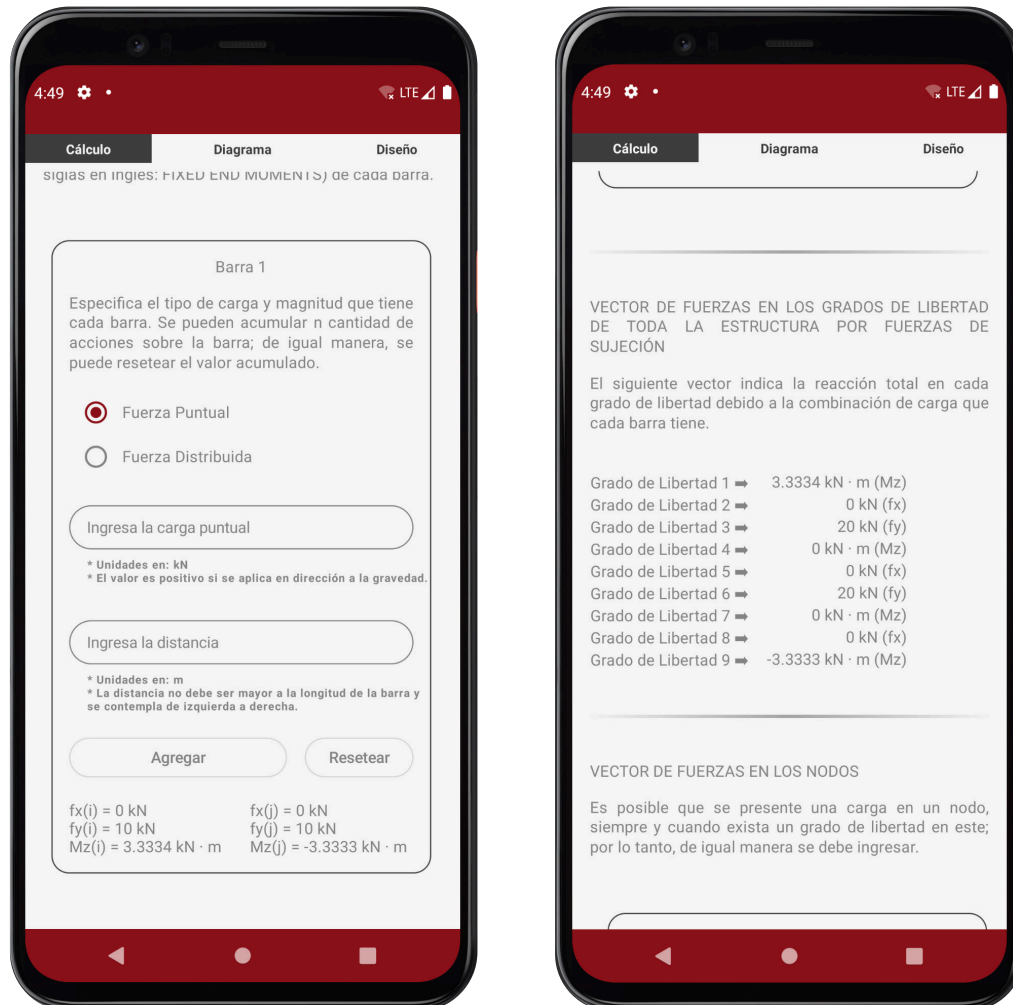


Figura.III.19. Fuerzas de fijación (Fixed-End Moments)
Fuente: Aplicación VigaLab RC. Autoría propia (2022).

También se le permite al usuario ingresar fuerzas en los nodos en caso de ser necesario, siguiendo el mismo método como en las barras. Figura III.20.



*Figura.III.20. Vector de fuerzas en los nodos
Fuente: Aplicación VigaLab RC. Autoría propia (2022).*

La tercera parte del método de rigideces consiste en tomar todos los valores ingresados por el usuario y proceder con las correspondientes operaciones matriciales y vectoriales para llegar a los resultados finales.

Lo primero es determinar la reacción total en cada grado de libertad, por lo que se suman las reacciones producidas por las fuerzas de fijación (FEM's) y las que están presentes en cada nodo. Figura III.21.

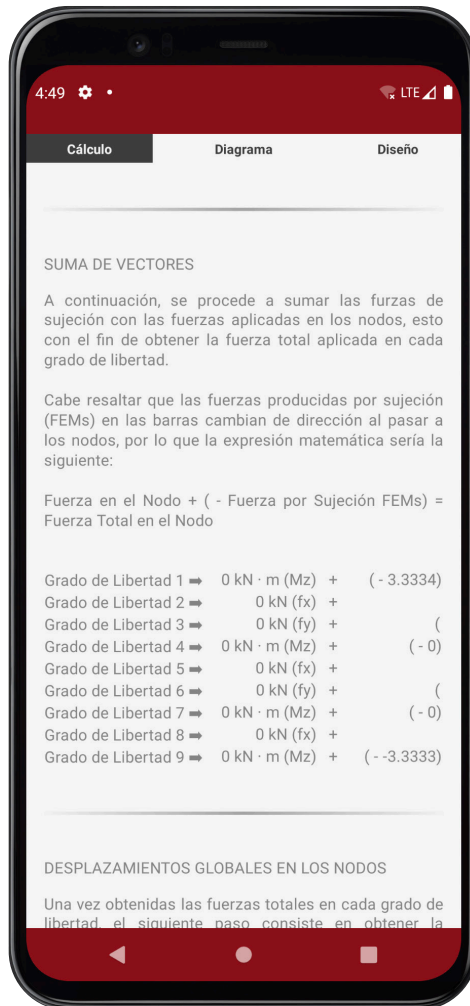


Figura.III.21. Suma de vectores
Fuente: Aplicación VigaLab RC. Autoría propia (2022).

Ya que se cuenta con la matriz de rigidez global y las fuerzas actuantes en los nodos, es posible encontrar los desplazamientos globales en cada uno de estos empleando la “Lay de Hooke”. Posterior a esto, se continua con los desplazamientos globales pero esta vez de cada barra. En la interfaz de usuario se explica con detalle el procedimiento mostrando fórmulas y breves explicaciones para que el usuario pueda guiarse, y hacer comprobaciones en caso de que así lo desee. Figura III.22.

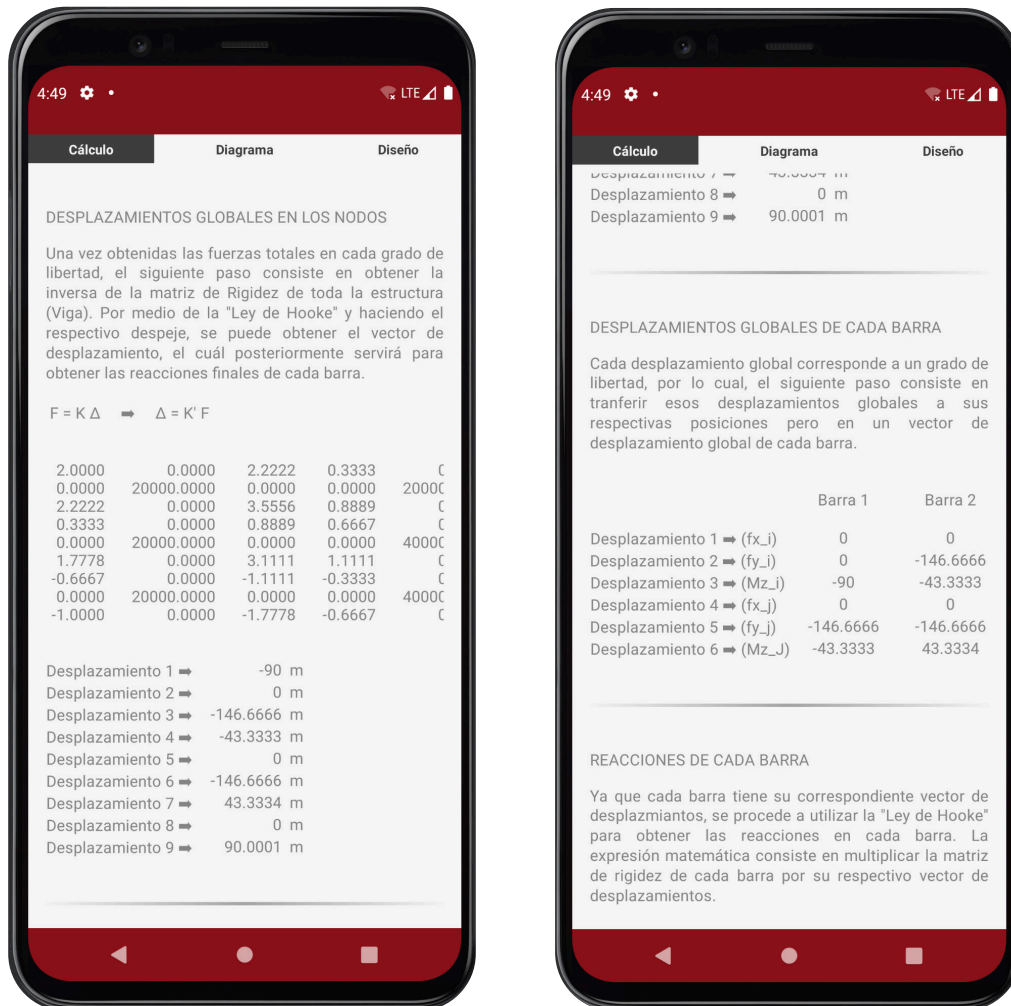


Figura.III.22. Desplazamientos globales en nodos y barras
 Fuente: Aplicación VigaLab RC. Autoría propia (2022).

Por último, se muestran los resultados finales de cada barra mostrando las reacciones en cada nodo. Se almacenan estos valores en cada una de las barras de la lista mutable que proviene del ViewModel.

El objetivo de esto, es que las tres pestañas están vinculadas al mismo ViewModel por lo que, para la creación de los diagramas, esta lista de barras es esencial, ya que con los resultados finales se puede construir un algoritmo que tome en cuenta cada uno de estos valores y trazar las líneas dentro del composable correspondiente al Canvas. Figura III.23.

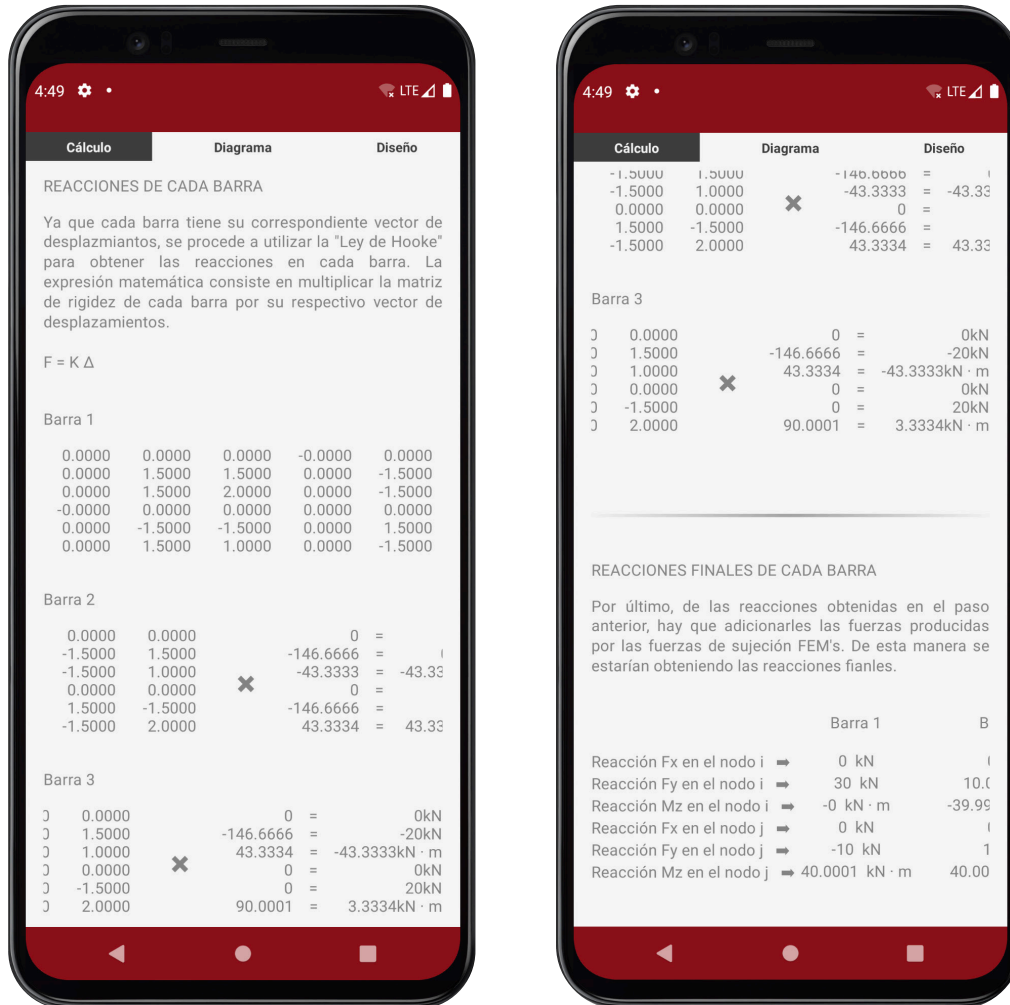


Figura.III.23. Reacciones finales de cada barra
 Fuente: Aplicación VigaLab RC. Autoría propia (2022).

Como buena práctica de programación, se deben segmentar los procesos computacionales en divisiones más pequeñas, en este caso, el método de rigideces se divide en quince partes, asignando en cada una de estas vistas composables los parámetros básicos para realizar el cálculo como son los objetos Barra y Nodo, y las unidades de Fuerza y Longitud. Todos estos módulos se integran en un composable general llamado LazyColumn, que no es más que una lista dinámica que renderiza toda vista que esté dentro de la pantalla, optimizando de esta manera los recursos internos del dispositivo móvil.

III.3.2. APARTADO DEL DIAGRAMA DE MOMENTO Y CORTANTE

Para la pestaña de “Diagrama”, se toma el tamaño de la ventana y los valores máximos de Momento y Cortante. Por medio de una regla de tres se puede establecer la longitud relativa de cada línea en base a los límites del Canvas.

La aplicación móvil permite graficar cualquier tipo de diagrama sin importar la cantidad de fuerzas distribuidas y puntuales que tenga el sistema estructural. De igual forma grafica el punto donde se encuentra el momento máximo y despliega los valores más relevantes tanto para el diagrama de cortante como el de momento. Figura III.24.

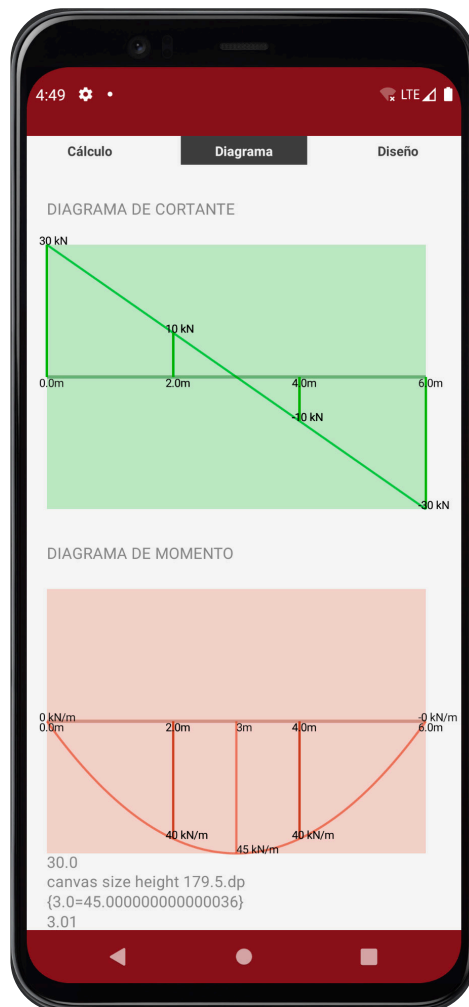


Figura.III.24. Reacciones finales de cada barra
Fuente: Aplicación VigaLab RC. Autoría propia (2022).

Para encontrar el momento máximo es necesario crear un algoritmo que por medio de iteraciones encuentre el mayor valor absoluto. Cabe destacar que para tener una buena precisión es necesario dividir en segmentos aun más pequeños cada barra. Figura III.25.

```

fun MaxMoment(
    claros: List<Barra>,
    nodos: List<Nodo>,
): MutableMap<Double, Double> {
    val momentDist = mutableMapOf<Double, Double>()

    val lengthDivisions = 500
    var maxMoment = 0.0
    var distance = 0.0
    var aux: Double
    var auxDist: Double

    claros.forEachIndexed { i, claro ->
        val distLoadList: MutableMap<Double, Double> = LinkedHashMap()
        claro.puntualLoadList.entries.sortedBy { it.key }
            .forEach { distLoadList[it.key] = it.value }

        val distancias = distLoadList.keys.toList()
        val cargas = distLoadList.values.toList()

        var variable = 0.0
        var variable2 = 0.0
        var variable3 = 0.0

        for (j in distancias.indices) {

            if (j != 0) {
                for (jj in 0 until j) {
                    variable += cargas[jj] * (distancias[j] - distancias[jj])
                }
            }

            if (j != distancias.lastIndex) {
                variable2 = (distancias[j + 1] - distancias[j]) / lengthDivisions
            }

            variable3 += cargas[j]

            /** Curva del diagrama de Momento **/
            for (k in 0 until lengthDivisions + 1) {

                auxDist = (nodos[i].x + distancias[j]) + (variable2 * k)
                aux = claro.vectorReacciones()[2] + (claro.vectorReacciones()[1] * (distancias[j] + . . .

                if (aux.absoluteValue > maxMoment.absoluteValue) {
                    maxMoment = aux
                    distance = auxDist
                }
            }
        }
    }
    momentDist[distance] = maxMoment
    return momentDist
}

```

*Figura.III.25. Algoritmo de momento máximo
Fuente: Aplicación VigaLab RC. Autoría propia (2022).*

Como punto final, se muestra también debajo de los dos diagramas información relevante de cada barra como las distancias a las cuales está aplicada cada fuerza puntual y la magnitud de la misma; además de la suma total de fuerzas distribuidas en caso de haber más de una.

Si el usuario requiere cambiar ciertos parámetros de la viga, basta con regresar a la pestaña de cálculo y hacer las modificaciones necesarias. Sin embargo, para actualizar los resultados que derivan de estos cambios, se debe deslizar la pantalla hasta el apartado de los resultados finales, solo de esta manera los diagramas pueden cambiar de forma.

III.3.3. APARTADO DEL DISEÑO DE LA SECCIÓN DE LA VIGA

Esta pestaña a pesar de estar vinculada con las otras dos por medio de un ViewModel, puede trabajar de manera independiente sin necesidad de realizar el cálculo y la revisión de los diagramas. La ventaja es que puede funcionar para diseñar vigas si se cuenta con el cortante y momento máximo desde otra fuente, además de hacer la revisión de la misma.

Como primer paso se debe especificar la normativa con la que se diseña y el sistema de unidades.



Figura.III.26. Selección de normativa y sistema de unidades
Fuente: Aplicación VigaLab RC. Autoría propia (2022).

Posteriormente, el usuario debe ingresar los valores necesarios para realizar el diseño por flexión y cortante. Todos estos parámetros son comunes entre las dos normativas, por consiguiente, la interfaz de usuario sigue siendo la misma. Figura III.27.



*Figura.III.27. Propiedades generales de la viga
Fuente: Aplicación VigaLab RC. Autoría propia (2022).*

Por último, el usuario puede llevar a cabo el diseño y la revisión de la viga completando parámetros adicionales en caso de ser requeridos. Algunos estos valores que deben ser ingresados tienen magnitudes predeterminadas que son recomendadas por la norma; sin embargo, se pueden sustituir por otros. Figura III.28.

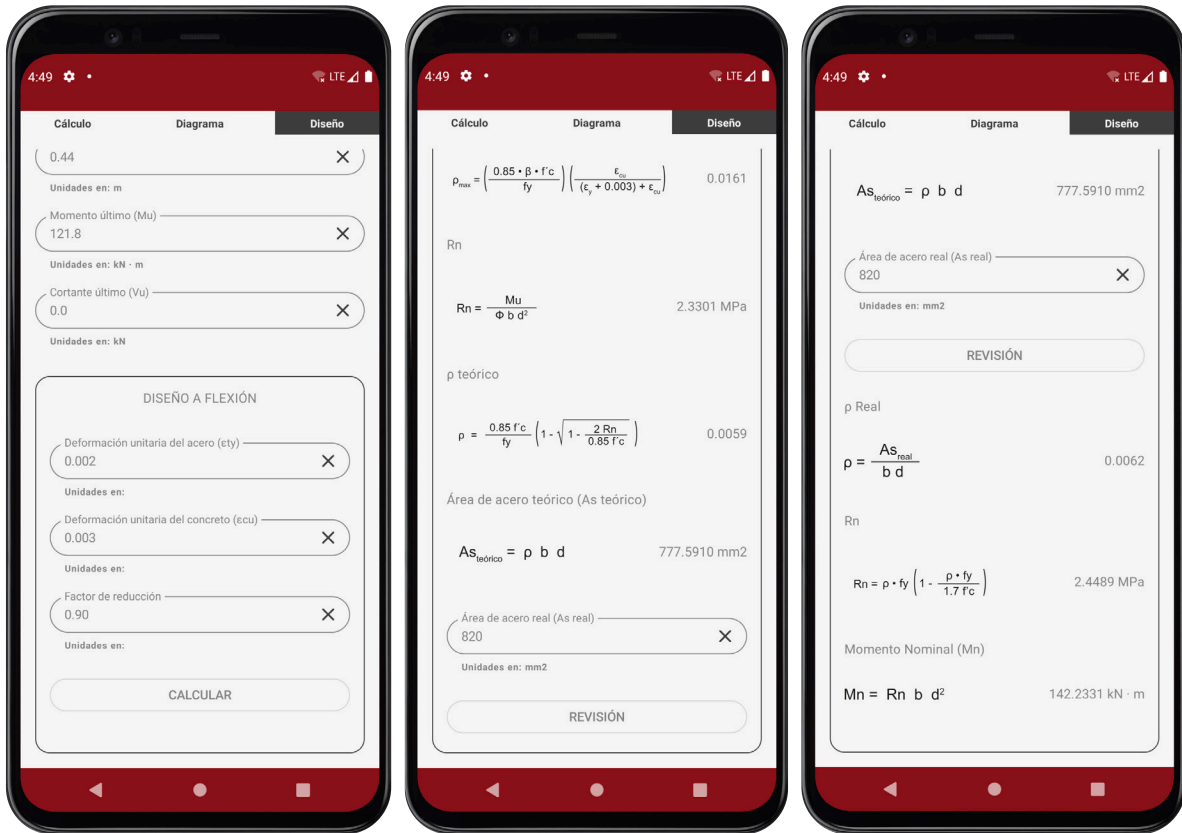


Figura.III.28. Diseño y revisión de una viga
Fuente: Aplicación VigaLab RC. Autoría propia (2022).

Para las dos normativas, se muestra tanto el diseño por flexión como el cortante en una vista deslizable horizontal, tal como está diseñado en el apartado de registro de propiedades de cada barra; por lo tanto, el usuario puede revisar los cálculos de manera más sencilla.

III.4. RESULTADOS

La aplicación móvil realiza dos funciones principales que no están ligadas, por una parte realiza el cálculo de una viga ante fuerzas puntuales y distribuidas por medio del método de rigideces, además de los diagramas de momento y cortante; y por otro lado, el diseño y revisión empleando dos normativas, el ACI 318-19 y las NTC 2021.

Para el presente subcapítulo se muestra evidencia de la precisión de la aplicación móvil empezando por el cálculo matricial. Se toma como referencia diversos ejercicios de fuentes bibliográficas serias que son usadas en la impartición de asignaturas referente al cálculo estructural y mecánica de materiales. Aunque, algunos ejercicios de estos libros usan otros métodos para encontrar las reacciones externas e internas en una viga, en teoría se debe llegar prácticamente al mismo resultado.

En las siguientes figuras se despliega la comparativa de las reacciones, diagrama de momento y cortante en una viga entre la aplicación VigaLab RC y un ejercicio del libro Análisis Estructural del autor R. C. Hibbeler. Cabe destacar que la aplicación VigaLab RC grafica el diagrama de momentos con los valores positivos por debajo del eje principal, razón por la cual parecería que el diagrama está al revés; sin embargo, son los mismo resultados. Figura III.29 y Figura III.30.

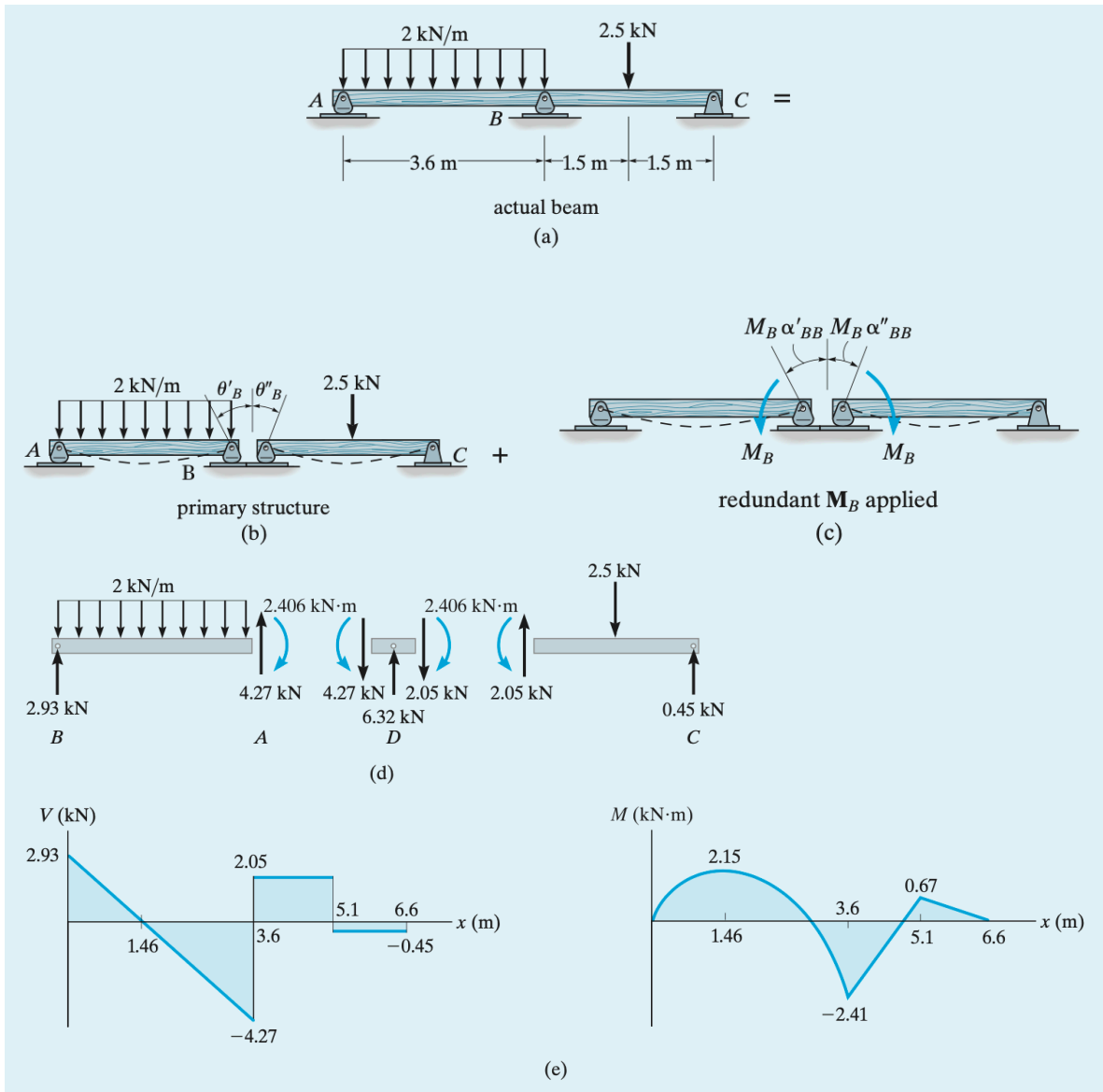


Figura.III.29. Reacciones y diagramas de cortante y momento de una viga
Fuente: Análisis Estructural, Hibbeler (2022).

	Barra 1	Barra 2
Reacción Fx en el nodo i →	0 kN	0 kN
Reacción Fy en el nodo i →	2.9316 kN	2.0522 kN
Reacción Mz en el nodo i →	-0 kN · m	2.4065 kN · m
Reacción Fx en el nodo j →	0 kN	0 kN
Reacción Fy en el nodo j →	4.2685 kN	0.4479 kN
Reacción Mz en el nodo j →	-2.4064 kN · m	-0 kN · m

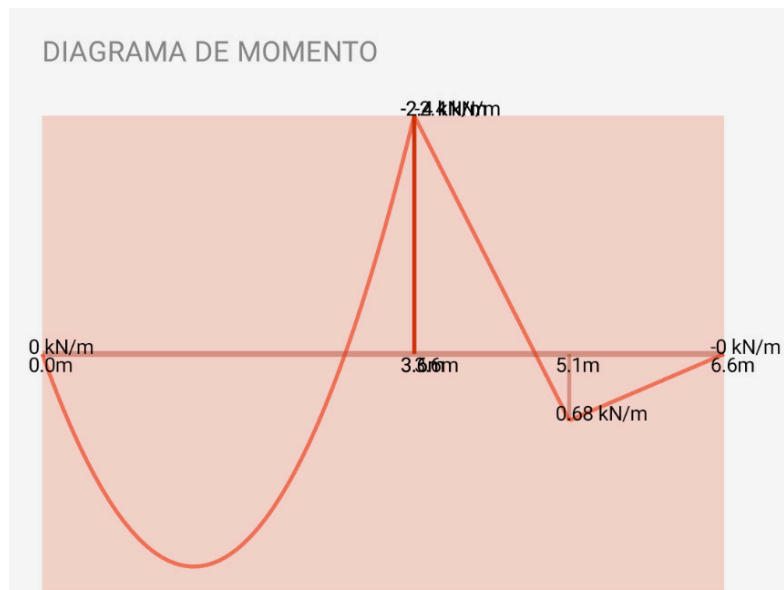
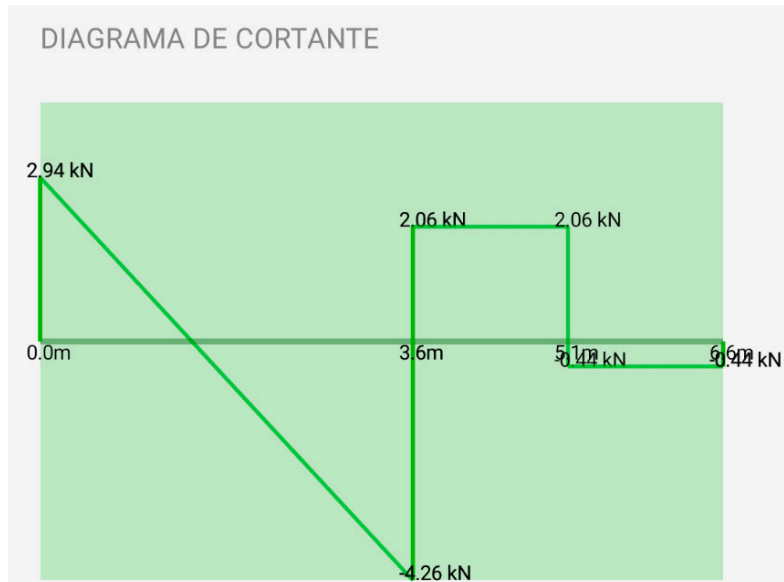


Figura.III.30. Resultados del ejercicio de Hibbeler
Fuente: Aplicación VigaLab RC. Autoría propia (2022).

Para el segundo ejemplo de demostración, se emplea un ejercicio del cálculo de una viga del libro Mecánica de Materiales del autor Ferdinand P. Beer. Para esta comparativa, se divide la viga en tres barras para ingresar el momento y la carga puntual en los dos grados de libertad que se encuentran en el nodo que une la segunda y tercera barra. Figura III.31 y Figura III.32.

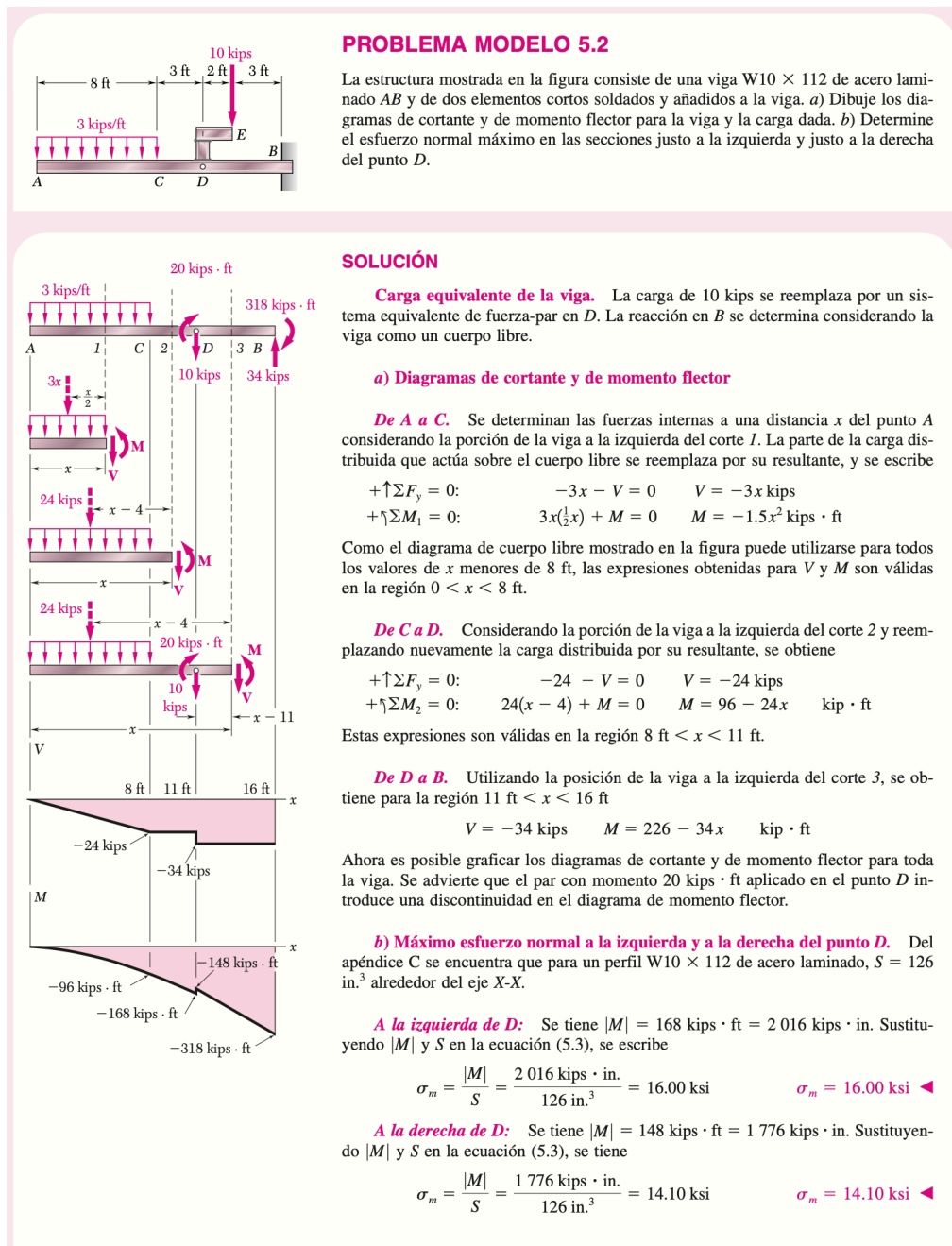


Figura.III.31. Reacciones y diagramas de cortante y momento de una viga
Fuente: Mecánica de materiales, Beer (2008).



Figura.III.32. Resultados del ejercicio de Beer
Fuente: Aplicación VigaLab RC. Autoría propia (2022).

Debido a la reciente actualización del ACI 318-19, por el momento no hay ejemplares que tomen como referencia este código. Sin embargo, existen expresiones matemáticas que prevalecen sin importar la versión; por ende, se puede emplear el libro Diseño de Concreto Reforzado del autor McCormac que se apoya en el código ACI 318-14 para demostrar la precisión de la aplicación al comparar el procedimiento de diseño a flexión de una viga. Figura III.33 y Figura III.34.

Las dimensiones de la viga mostrada en la Figura 4.5 han sido escogidas por razones arquitectónicas. Determinar el área del acero de refuerzo por medio de cada uno de los métodos descritos en esta sección.

Usando las tablas del apéndice

$$\frac{M_u}{\phi b d^2} = \frac{(12 \text{ plg/pie})(160,000 \text{ pie-lb})}{(0.9)(16 \text{ plg})(21 \text{ plg})^2} = 302.3 \text{ lb/plg}^2$$

ρ de la tabla A.12 apéndice A = 0.00538 (por interpolación)

$$A_s = (0.00538)(16 \text{ plg})(21 \text{ plg}) = 1.81 \text{ plg}^2$$

Use 6 varillas del #5 (1.84 plg²)

Usando la fórmula para ρ

$$R_R = \frac{M_u}{\phi b d^2} = 302.3 \text{ lb/plg}^2$$

$$\rho = \frac{(0.85)(3000 \text{ lb/plg}^2)}{60,000 \text{ lb/plg}^2} \left[1 - \sqrt{1 - \frac{(2)(302.3 \text{ lb/plg}^2)}{(0.85)(3000 \text{ lb/plg}^2)}} \right]$$

= 0.00538 (como el mismo valor obtenido del Apéndice A)

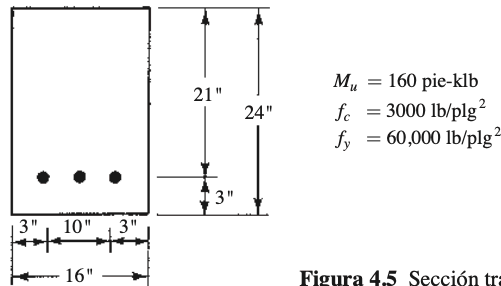


Figura 4.5 Sección transversal de la viga del ejemplo 4.5.

$$A_s = \frac{M_u}{\phi f_y \left(d - \frac{a}{2} \right)} = \frac{(12 \text{ plg/pie})(160,000 \text{ pie-lb})}{(0.9)(60,000 \text{ lb/plg}^2) \left(21 \text{ plg} - \frac{2 \text{ plg}}{2} \right)} = 1.78 \text{ plg}^2$$

$$a = \frac{A_s f_y}{0.85 f'_c} = \frac{(1.78 \text{ plg}^2)(60,000 \text{ lb/plg}^2)}{(0.85)(3000 \text{ lb/plg}^2)(16 \text{ plg})} = 2.62 \text{ plg}$$

Suponga $a = 2.6 \text{ plg}$

$$A_s = \frac{(12 \text{ plg/pie})(160,000 \text{ pie-lb})}{(0.9)(60,000 \text{ lb/plg}^2) \left(21 \text{ plg} - \frac{2.62 \text{ plg}}{2} \right)} = 1.81 \text{ plg}^2$$

$$a = \frac{(1.81 \text{ plg}^2)(60,000 \text{ lb/plg}^2)}{(0.85)(3000 \text{ lb/plg}^2)(16 \text{ plg})} = 2.66 \text{ plg} \text{ (valor bastante cercano)}$$

Basándose en este método, use un valor teórico de $A_s = 1.81 \text{ plg}^2$.
 Observe que los tres métodos conducen al mismo valor de $A_s = 1.81 \text{ plg}^2$.

Figura.III.33. Diseño a flexión de una viga
 Fuente: *Diseño de concreto reforzado, McCormac (2017).*

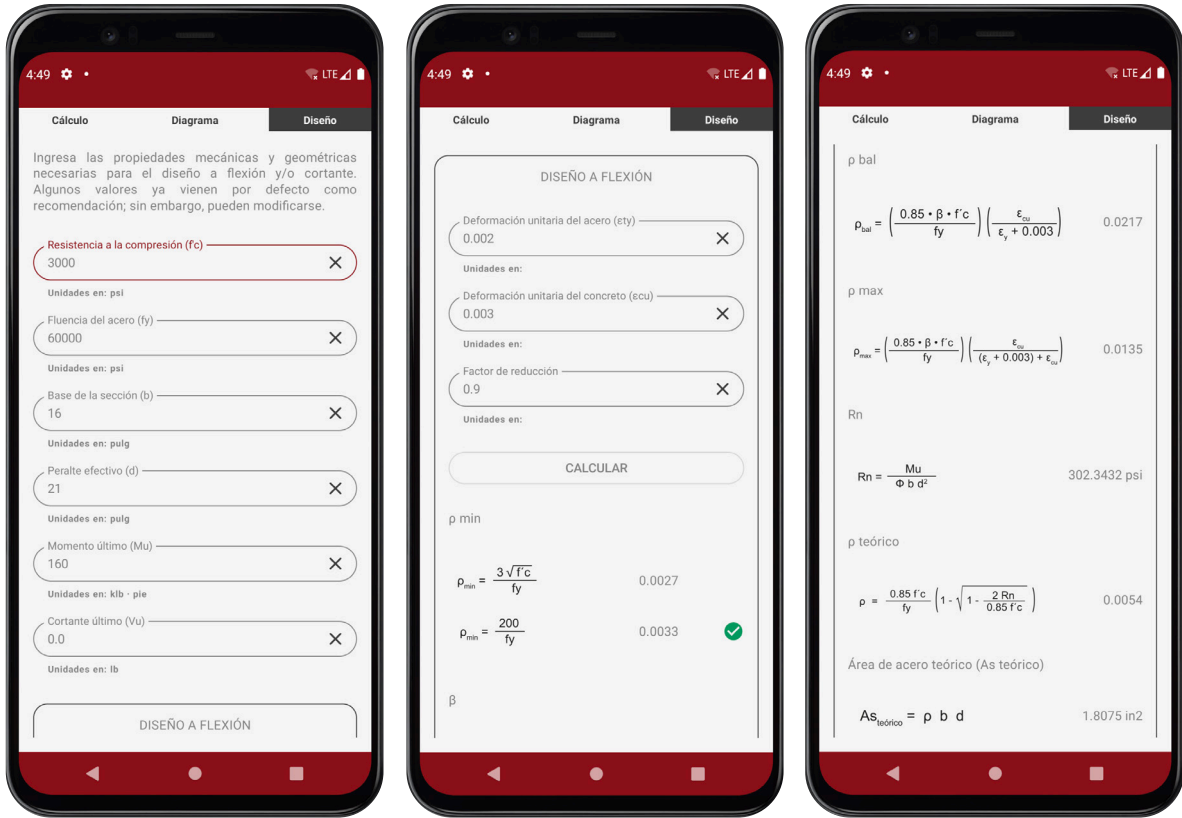


Figura.III.34. Resultados del ejercicio de McCormac
 Fuente: Aplicación VigaLab RC. Autoría propia (2022).

CONCLUSIÓN

En lo que respecta a las normativas, hay mucha diferencia en cuestión de condiciones que se deben tomar en cuenta. La norma americana ACI 318-19 es más meticulosa, mostrando diversas fórmulas dependiendo de las propiedades del concreto y del acero. Esto es bueno porque se garantiza un comportamiento idóneo y más cercano a la realidad cuando la viga ya está construida.

Según la norma, estos parámetros adicionales, actualización de fórmulas y tablas en comparación con su versión anterior ACI 318-14, se debe a la continua mejora del concreto y del acero que fabricantes realizan conforme la tecnología avanza con el fin de destacar en el mercado. Por otro lado, las normas técnicas complementarias NTC 2021 se apoyan en la norma americana, adaptando dichas fórmulas, tablas y consideraciones en base a la materia prima y materiales que tiene el país. No obstante, las NTC 2021 podrían haber tomado como referencia la norma ACI 318-19, que es la más actualizada, y no la versión anterior que ya no es recomendada usarse inclusive por su país de origen. La ventaja que tiene las NTC 2021, es que son más fáciles de usar y el diseño es más corto, traduciendo en un ahorro de tiempo.

Las dos normativas son válidas en nuestro país, y dependiendo del estado en donde se construya, una va a tener más influencia que la otra. De cualquier forma, el realizar el diseño ya sea por ACI 318-19 o NTC 2021, conlleva tiempo y esfuerzo; por esta razón, el uso de un dispositivo computacional es necesario y recomendado. El presente proyecto cubre esa necesidad al desarrollar una aplicación móvil que toma en cuenta las dos normativas más actualizadas.

El realizar la programación no fue tarea sencilla, ya que se debe diseñar una interfaz que se adapte a las dos normativas cuidando que no haya redundancia al momento de pedir o desplegar la información. Otro reto referente a la codificación,

fue la cantidad de condiciones que se deden escribir y como integrarlos a los algoritmos que realizan los cálculos. El uso de la arquitectura MVVM resuelve bien este problema pero se extiende demasiado la lógica al implementar todas las fórmulas que derivan de los diferentes sistemas de unidades que tienen las dos normas.

Referente al tiempo, se tarda más el diseñar los elementos visuales al asignar tamaños, colores y formas, que la programación de las expresiones matemáticas. Inclusive, todas las imágenes vectoriales realizadas en Adobe Illustrator consumió gran parte del desarrollo, al igual que el detallar los diagramas, usando la librería de Canvas nativo de Android.

En conclusión, a pesar del poco tiempo que se tuvo para desarrollar con amplitud esta aplicación móvil, se crea una herramienta útil tanto para estudiantes de ingeniería, como profesionistas, y no solo para mexicanos, sino para ingenieros de otros países. Se aprovecha la tecnología JetPack Compose y patrones de diseño más avanzadas que proporciona Android para desallorrar una aplicación de calidad que satisface las necesidades y demandas que puedan surgir en el aula o en el campo laboral.

RECOMENDACIONES

Se puede mejorar un poco más en lo que respecta al diseño y paleta de colores para que sea más vistosa la aplicación. También se puede sustituir la fuente por otra que sea más amigable.

De igual forma, para que sea aún más dinámica al devolver resultados en la pestaña de diagrama, se puede implementar un “Slider”, que es un componente que permite deslizar un elemento circular a lo largo de una barra. Esto podría permitir obtener el momento y cortante en cualquier punto de la viga.

Esta aplicación se puede actualizar cuando las normas presenten una mejora en su contenido o en su defecto cuando Google emita un cambio de tecnología; sin embargo, el programa ejecutable al subirse en una plataforma que no se la PlayStore, el usuario debe descargar la versión más reciente por su propia cuenta.

En relación al tema académico, se recomienda que el producto generado de este trabajo de investigación sea revisado y avalado por la o las academias competentes. Así mismo, se considere dentro del material didáctico que el estudiante utilice en las materias correspondientes debido a la practicidad, seguridad, versatilidad, rapidez y economía; en una era en donde la tecnología es indispensable en la educación.

En un futuro, inclusive se podrían añadir más objetos a la aplicación conforme surjan recomendaciones por parte de los usuarios académicos y profesionistas.

BIBLIOGRAFÍA

McCormac, J. & Brown R.. (2017). *Diseño de concreto reforzado*. Ciudad de México: Alfaomega.

Beer, F., Johnston, E. R., DeWolf, J., Mazurek, D., & Jr. Johnston, E. R. (2008). *Mechanics of Materials*. McGraw-Hill Education.

Hibbeler, R. C. (2019, 30 julio). *Structural Analysis in SI Units (10.a ed.)*. Pearson.

Reglamento de Construcciones del Distrito Federal. (2021). *Normas Técnicas Complementarias para Diseño y Construcción de Estructuras de Concreto*. México: NTC 2021.

American Concrete Institute. (2019). *Building Code Requirements for Structural Concrete*. Estados Unidos de America: ACI 318-19.

Android Developers. (2021), de Google Sitio web: <https://developer.android.com>

