



# MODELACIÓN NUMÉRICA DE UN CABEZAL DE ESTRIBO PARA PUENTE Y AUTOMATIZACIÓN DE SU MEMORIA DE CÁLCULO

NUMERICAL MODELING OF A HEAD FROM STIRUP TO BRIDGE  
AND AUTOMATION OF YOUR CALCULATION MEMORY

---

Fecha de recepción: 07/06/2024 | Fecha de aceptación: 28/06/2024

Echalar Flores Michael Willy<sup>1</sup>

<sup>1</sup>Ingeniero Civil, M. Sc. Docente Investigador de la Carrera de Ingeniería Civil, FCyT – U.A.J.M.S.

**Correspondencia del autor:** [michaelechalar@uajms.edu.bo](mailto:michaelechalar@uajms.edu.bo)<sup>1</sup>

Tarija - Bolivia

## RESUMEN

Este trabajo presenta la modelación numérica de un cabezal de estribo que soporta la superestructura de un puente, utilizando librerías de Python. Se describe un proceso automatizado que abarca desde el análisis estructural y el diseño del elemento de hormigón armado hasta la generación de la documentación técnica, incluyendo la memoria de cálculo y los esquemas de planos.

Se enfoca en la descripción del proceso, las herramientas utilizadas, y los resultados obtenidos con el fin de proporcionar un marco de referencia para otros investigadores, siendo este trabajo del tipo descriptivo.

La implementación se realiza completamente en un entorno de programación Python, utilizando diversas librerías especializadas que permiten optimizar el flujo de trabajo, evitando la necesidad de emplear múltiples herramientas de software. Se realiza el análisis estructural, el diseño de la sección, su detallamiento y la generación de su memoria de cálculo.

El estudio destaca las ventajas de Python, Latex y Quarto como un recurso accesible, eficiente y robusto para el análisis y diseño en ingeniería civil, demostrando su capacidad para reemplazar soluciones comerciales de software.

## ABSTRACT

This paper presents the numerical modeling of an abutment cap supporting a bridge superstructure, using the Python anaStruct library. An automated process is described that ranges from structural analysis and design of the reinforced concrete element to the generation of technical documentation, including the calculation report and drawing schemes.

It focuses on the description of the process, the tools used, and the results obtained in order to provide a reference framework for other researchers.

The implementation is carried out entirely in a Python programming environment, using various specialized libraries that allow the workflow to be optimized, avoiding the need to use multiple software tools. The study highlights the advantages of Python, Latex and Quarto as an accessible, efficient and robust resource for analysis and design in civil engineering, demonstrating their ability to replace commercial software solutions.

**Palabras Clave:** Python, Latex, Automatización, Diseño Estructural, FEM, Ingeniería Computacional.

**Keywords:** Python, Latex, Statics, FEM, Computational Engineering.

## 1. INTRODUCCIÓN

El proceso de análisis y diseño de elementos de hormigón armado es tedioso y requiere el paso por diferentes paquetes informáticos, iniciando por el empleado para el análisis del elemento; generalmente se emplean programas de elementos finitos como: Sap2000, CsiBridge, Midas y otros. A partir de los esfuerzos obtenidos se procede al diseño, para lo cual se suelen emplear planillas Excel, hojas de Mathcad y Matlab; se confeccionará posteriormente el plano constructivo que contiene el detallamiento de dimensiones y armaduras para la materialización del elemento. Lo cual se realiza generalmente en AutoCAD, se debe incluir también la memoria de cálculo la cual describe todo el proceso y permite la revisión de la idoneidad del trabajo, para lo cual se emplea casi siempre un documento de Word.

Lo descrito supone tráfico de información a través de varios paquetes y formatos. Este flujo de trabajo es ineficiente debido al tiempo necesario para realizar el cambio de formato, como a la posibilidad de la pérdida o corrupción de los datos durante el proceso.

El flujo óptimo es mantener todo el proceso dentro de una sola plataforma, en la cual se debe poder analizar, diseñar, dibujar y documentar. El proceso está descrito específicamente dentro del ámbito de la ingeniería estructural, pero de manera general lo que se está haciendo es emplear varios algoritmos y sistemas para solucionar un problema de ingeniería, esto último implica el uso de métodos computacionales y procesamiento de datos, lo que se está haciendo es Ingeniería Computacional como la define Fangohr (2018).

En la ingeniería actual debido a la complejidad de los problemas a los que se enfrenta y a la cantidad de datos existentes es necesario realizar cálculos numéricos, modelación, simulaciones y análisis de los datos obtenidos de estas (Huijse, 2022).

Es evidente que pocos recursos computacionales pueden cubrir los requerimientos para poder com-

pletar exitosamente el proceso, uno de ellos es el lenguaje de programación Python. De esta manera este trabajo se centra en la documentación y análisis de cómo se emplea el mismo para la automatización, de procesos de diseño y la generación de documentación técnica, se incluye la descripción detallada del algoritmo, las librerías empleadas y su modo de implementación. De esta forma la presente investigación puede considerarse como descriptiva.

Al tratarse este de un software libre y de código abierto ha sido acogido por la comunidad ingenieril en general, de modo que se cuenta con librerías para todas las tareas que deben ser realizadas en nuestro caso particular.

Las hay disponibles para la modelación numérica mediante elementos finitos en 2D y 3D como anaStruct, Pynite y Opensees; para el manejo de ecuaciones y cálculos de ingeniería como Sympy y Numpy, se puede pasar la notación de computadora a texto manual empleando Handcalc.

Es posible también manejar valores con unidades asignadas, de modo que, y al realizar los cálculos las mismas se compongan y simplifiquen, mediante Pint.

Están disponibles también librerías para la escritura de archivos dxf (formato universal para intercambio de información CAD) como Ezdxf o para la comunicación con AutoCad como Pycad. También se puede lograr la comunicación con Word mediante Python-docx. Añadiendo a lo mencionado existe Quarto, que es un sistema de publicación de documentación técnica de código abierto. El cual permite la creación de artículos informes y libros a partir de un archivo ipynb (interactive Python notebook) llamado cuaderno Jupyter, o de un archivo qm (quarto markdown); permitiendo la inserción de imágenes, tablas y otros.

Existen trabajos en los cuales se han realizado modelos numéricos para verificar la rigidez y el grado de deformación de elementos estructurales (Miran-

da et al., 2024), también se ha empleado en el cálculo de flechas inmediatas en elementos de hormigón armado (Déo, 2022), como en la optimización estructural de elementos sólidos como indican Sarma et al. (2023) y He et al. (2023); como también en cerchas (He et al., 2019).

El empleo de Python en la ingeniería civil no se reduce únicamente a la parte estructural, se ha empleado para el procesamiento de información y la predicción en 3D de investigaciones geotécnicas como mencionan Ngeno et al. (2024) y Ali & Shakir (2022). También se han empleado algoritmos para representar tridimensionalmente modelos hidrogeológicos (Mekonen et al., 2024).

El presente trabajo muestra la modelación numérica del cabezal de un estribo que soporta la superestructura de un puente, la cual incluye el análisis estructural, el diseño como sección de hormigón armado y el uso de esta información para la generación automática de documentación técnica que incluye la memoria de cálculo del elemento y esquemas para planos.

La integración de las librerías especializadas de Python para realizar el análisis, diseño, salida a planos y redacción de la memoria de cálculo se logra mediante Quarto; que es una implementación de la librería Pandoc, una versión moderna del lenguaje LaTeX. El cual siempre se ha empleado junto con Python para el manejo de expresiones matemáticas (Arroyo et al., 2024).

La generación de documentación técnica también ha empleado con bastante éxito Python y Latex (Poore, 2015). Los cuales permiten la generación automática de documentación. Respecto de esto Quarto supone un avance respecto de LaTeX, ya que no es necesario conocer el lenguaje a profundidad y se puede en gran parte de la generación de documentos evitar problemas de compilación. Sin embargo por sí mismo es

aún el estándar para la documentación matemática y de programación (Triantafyllidis & Papageorgiou, 2018).

Además de su uso en documentación técnica, Python y LaTeX son ampliamente usados en la generación de documentación educativa, como indica Orduz-Ducara (2013).

## 2. MATERIALES Y MÉTODOS

El presente trabajo se desarrolló en un sistema de las siguientes características:

OS: Windows 11  
IDE: VS Code

### Se usaron los siguientes recursos de software:

Lenguaje: Python v.3.12.3  
Librerías: Anastruct  
Pandas  
Handcalc  
Openpyxl  
Numpy  
Pint  
Ezfxf  
Quarto  
Pypdf

La estructura de la carpeta que contiene el proyecto es la siguiente:

```
infraestructura/  
    MdC_cabezal_estribo/  
    planos/  
    imágenes/  
    pdf/  
    src/
```

Python tiene diferentes tipos de archivo, que permiten diferentes grados de interactividad, iniciamos en el guion de extensión py los cuales no permite interacción, simplemente se ejecuta el guion y se ob-

serva el resultado de la ejecución, como muestra la figura 1

```
memoriaEstribo.ipynb  hacer.py x
hacer.py > ...
1 import os
2 import subprocess
3
4 file = "memoriaEstribo.ipynb"
5 render = f"quarto render {file} --to pdf"
6 subprocess.run(render)
7 # render2 = f"quarto render {file} --to docx"
8 # subprocess.run(render2)
9 print(os.path, file)
10
```

Figura 1: Archivo de extensión .py  
Fuente: Elaboración propia

Esta la terminal interactiva que permite la carga sucesiva de comandos y la presentación de resultados parciales. Esta como el documento más interactivo en ya mencionado ipynb, e cual es un documento con estructura html que contiene células de código, resultados y texto de tipo markdown, como se muestra en la figura 2.

```
memoriaEstribo.ipynb  Untitled-1.ipynb
+ Code + Markdown | Run All Restart Clear All Outputs

a=1
b=2
c=a+b
[1] ✓ 0.0s

se va imprimir la variable c

print(c)
[2] ✓ 0.0s
... 3
```

Figura 2: Archivo de extensión .ipynb  
Fuente: Elaboración propia

Para que Quarto pueda traducir el archivo de tipo ipynb a un pdf, se requiere de un encabezado del tipo yaml, en el cual se define el formato del documento a generar: tamaño de la hoja, orientación, márgenes, si se va mostrar el código de celdas o no, etc. Un pequeño ejemplo se tiene en la figura 3.

```
File Edit Selection View Go Run Terminal Help
memoriaEstribo.ipynb  Untitled-1.ipynb
memoriaEstribo.ipynb > M+ MATERIALES. -
+ Code + Markdown | Run All Clear All Outputs | Outline ...

---
title: Memoria de Calculo Cabezal de Estribo
format:
  pdf:
    keep-ipynb: true
    keep-tex: false
    documentclass: scrartcl
    papersize: letter
    toc: false
    shift-heading-level-by: 2
    fig-pos: "H"
    fig-cap-location: top
    geometry:
      - top=0.6in
      - right=.6in
      - bottom=0.8in
      - left=.6in
execute:
  echo: false
  warning: false
---
```

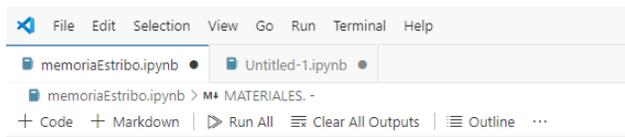
Figura 3: Encabezado de configuración en formato yaml  
Fuente: Elaboración propia

En la figura 3 se puede observar las líneas:

Execute:

Echo: false

Esto tendrá un efecto importante en el documento final ya que todas las celdas de código se escondrán, mas no su resultado. Por lo que la importación de librerías no será visible.



## NORMA DE DISEÑO. -

El puente se diseñó en su totalidad empleando la Norma de diseño: AASHTO LRFD Bridge Design Specifications (9th Edition, 2020)

## CARGAS.

De acuerdo a la normativa adoptada AASHTO LRFD 2020 las cargas aplicadas se ha considerado como peso específico de los materiales los siguientes:

- Hormigón armado: 23.544 KN/m<sup>3</sup>
- Hormigón pretensado: 23.544 KN/m<sup>3</sup>
- Hormigón simple: 21.750 KN/m<sup>3</sup>

Carga de vehículos:

Se ha considerado en el diseño el vehículo HL-93.

```
from openpyxl import load_workbook
from anastruct import SystemElements, Vertex
from anastruct import LoadCase, LoadCombination
import handcalcs.render
from handcalcs import handcalc
import numpy as np
import pandas as pd
from math import sqrt
from textwrap import dedent
from IPython.display import Markdown

from pint import UnitRegistry

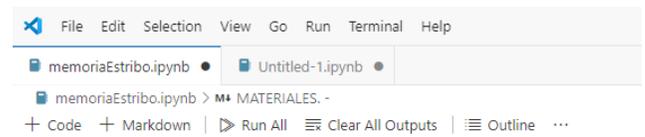
ureg = UnitRegistry()
ureg.default_format = "~"

pi=np.pi
```

Figura 4: Descripción de la norma, cargas e importación de librerías Fuente: Elaboración propia

Como se mencionó anteriormente se pueden emplear celdas con texto y código, en la figura 4, se puede observar el preámbulo que describe las normativas u condiciones del trabajo a realizar y también la importación de librerías de Python para las distintas tareas a realizar.

La parte siguiente del documento es la importación de valores, previamente calculados en una hoja de Excel, como se ve en la figura 5.



## Peso de la superestructura

```
tabla = pd.read_excel(r"D:/2024 proy/70 diseño final/03 pila/cabeza/TablaPesoSuperestructura.xlsx")

pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)
pd.set_option('display.width', 1000)
pd.set_option('display.colheader_justify', 'center')

tabla.head(8).style.format(precision=3).hide(axis="index")
```

Elemento	Area	Longitud	Volumen	Veces	Total
Tablero	1.777	30.600	54.364	1.000	54.364
Diafragma Int	2.393	0.250	0.598	6.000	3.589
Diafragma Ext	1.851	0.250	0.463	6.000	2.776
TOTAL LOSA					60.729
Sec I	0.524	27.400	14.366	4.000	57.463
Sec Ext	1.098	1.600	1.756	8.000	14.049
TOTAL VIGA					71.513

```
wb = load_workbook(r"D:/2024 proy/70 diseño final/03 pila/cabeza/TablaPesoSuperestructura.xlsx")
sh = wb["Hojas2"]
```

Figura 5: Lectura de los pesos de la superestructura Fuente: Elaboración propia

La tabla generada de la lectura de datos se presentará en el documento en el formato de LaTeX. La figura 6 muestra la definición de la estructura en la librería Anastruct. Solucionada la misma en el sentido longitudinal y luego transversal se procederá a recuperar el momento máximo actuante como se ve en a figura 7.

Posteriormente se realiza el diseño de la sección de hormigón armado, para lo cual se debe recuperar el momento máximo como se indica en la figura 8. Se observa que al momento actuante se le han asignado unidades de kN.m y a las resistencias de los materiales en MPa, se verá más adelante que la librería de unidades simplificará y compatibilizará los resultados.

```

File Edit Selection View Go Run Terminal Help
memoriaEstribo.ipynb • Untitled-1.ipynb
memoriaEstribo.ipynb > MATERIALES.
+ Code + Markdown | ▶ Run All | Clear All Outputs | Outline ...

Nod = [None] * 8
Nod[0] = [0, 0]
Nod[1] = [L - 2 * 4.3 - 15.24, 0]
Nod[2] = [L - 2 * 4.3 - 15.24 + 4.3, 0]
Nod[3] = [L - 2 * 4.3 - 15.24 + 4.3 + 4.3, 0]
Nod[4] = [L, 0]
Nod[5] = [L + 4.3, 0]
Nod[6] = [L + 2 * 4.3, 0]
Nod[7] = [2 * L, 0]

[91]

A = 1
I = 510
E = 29000

ss = SystemElements(EA=E * A, EI=E * I)

[92]

#ss.add_element(location=[Nod[0], Nod[1]])
#ss.add_element(location=[Nod[1], Nod[2]])
#ss.add_element(location=[Nod[2], Nod[3]])
#ss.add_element(location=[Nod[3], Nod[4]],spring=(2: 0))

ss.add_element(location=[Nod[4], Nod[5]])
ss.add_element(location=[Nod[5], Nod[6]])
ss.add_element(location=[Nod[6], Nod[7]])

ss.add_support_hinged(node_id=[1,4])

ss.q_load(element_id=[ele for ele in range(1,4)], q=-0. carril)
ss.point_load(node_id=[2,1], Fy=-P)
ss.point_load(node_id=[3], Fy=-P/4)
    
```

Figura 6: Definición de la geometría de la estructura  
Fuente: Elaboración propia

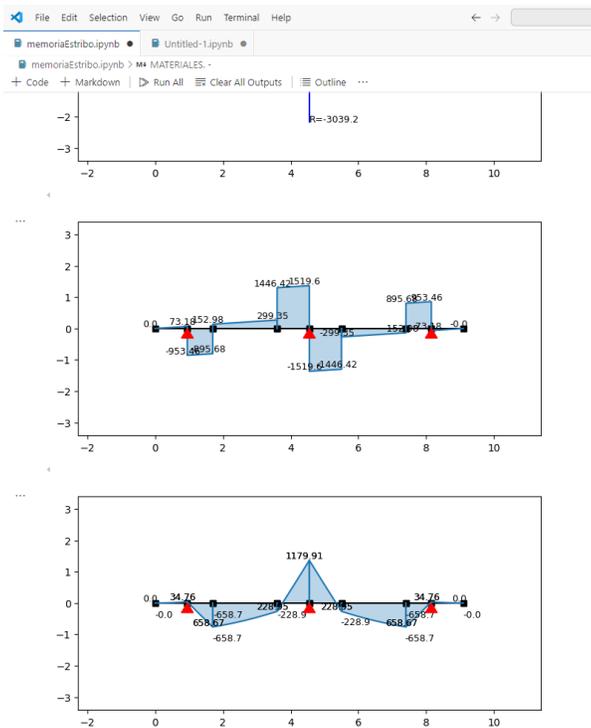


Figura 7: Diagrama de cortantes y momentos  
Fuente: Elaboración propia

```

File Edit Selection View Go Run Terminal Help
memoriaEstribo.ipynb • Untitled-1.ipynb
memoriaEstribo.ipynb > MATERIALES.
+ Code + Markdown | ▶ Run All | Clear All Outputs | Outline ...

M_u=ss.get_element_results(element_id=4)['Mmax']*ureg.kN*ureg.m

[121]

%%render
#Long

fc=21*ureg.MPa
fy=420*ureg.MPa

M_u
Phi=0.9
Mn= M_u/Phi
b=1.70*ureg.m
h=1.10*ureg.m
dc=0.025*ureg.m
d= h-dc
a= (d*(1-sqrt(1-Mn/(0.425*b*d**2*fc))))).to(ureg.cm)

[122]
    
```

Figura 8: Datos para el diseño de la sección de hormigón armado

Fuente: Elaboración propia

En este punto se debe aclarar que el código de Python recibe el valor de las variables y retorna el resultado de la evaluación de la fórmula, sin embargo, gracias a la librería Handcalc se puede realizar la presentación de la fórmula, la sustitución de valores y evaluación tal como se haría en un cálculo manual como se muestra en la figura 9.

$$\begin{aligned}
 f_c &= 21.000 \text{ megapascal} \\
 f_y &= 420.000 \text{ megapascal} \\
 M_u &= 1179.908 \text{ kilonewton} \cdot \text{meter} \\
 \Phi &= 0.900 \\
 M_n &= \frac{M_u}{\Phi} \\
 &= \frac{1179.908 \text{ kilonewton} \cdot \text{meter}}{0.900} \\
 &= 1311.009 \text{ kilonewton} \cdot \text{meter} \\
 b &= 1.700 \text{ meter} \\
 h &= 1.100 \text{ meter} \\
 d_c &= 0.025 \text{ meter} \\
 d &= h - d_c \\
 &= 1.100 \text{ meter} - 0.025 \text{ meter} \\
 &= 1.075 \text{ meter} \\
 a &= \left( d \cdot \left( 1 - \sqrt{1 - \frac{M_n}{0.425 \cdot b \cdot (d)^2 \cdot f_c}} \right) \right) \\
 &= \left( 1.075 \text{ meter} \cdot \left( 1 - \sqrt{1 - \frac{1311.009 \text{ kilonewton} \cdot \text{meter}}{0.425 \cdot 1.700 \text{ meter} \cdot (1.075 \text{ meter})^2 \cdot 21.000 \text{ megapascal}}} \right) \right) \\
 &= 4.097 \text{ centimeter}
 \end{aligned}$$

Figura 9: Calculo de la profundidad del bloque comprimido  
Fuente: Elaboración propia

En la figura 10 vemos que para un momento de 1179.9 kNm trabajando en una sección con hormigón de 21MPa y 420MPa de fluencia en el acero, son necesarias 9.42 barras de diámetro 20mm.

$$\begin{aligned} A_s &= \left( 0.85 \cdot a \cdot b \cdot \frac{f_c}{f_y} \right) \\ &= \left( 0.85 \cdot 4.097 \text{ centimeter} \cdot 1.700 \text{ meter} \cdot \frac{21.000 \text{ megapascal}}{420.000 \text{ megapascal}} \right) \\ &= 29.601 \text{ centimeter}^2 \end{aligned}$$

$$\phi = 20.000 \text{ millimeter}$$

$$\begin{aligned} A_b &= \left( \pi \cdot \frac{(\phi)^2}{4} \right) \\ &= \left( 3.142 \cdot \frac{(20.000 \text{ millimeter})^2}{4} \right) \\ &= 3.142 \text{ centimeter}^2 \end{aligned}$$

$$\begin{aligned} N &= \frac{A_s}{A_b} \\ &= \frac{29.601 \text{ centimeter}^2}{3.142 \text{ centimeter}^2} \\ &= 9.422 \text{ dimensionless} \end{aligned}$$

Figura 10: Determinación del área de armadura requerida y del número de barras

Fuente: Elaboración propia

Es evidente que el número de barras es adimensional (dimensionless).

Hasta este punto se ha realizado el análisis y diseño del elemento, y los comentarios de texto entre celdas de código completarían la memoria de cálculo. Para lograr la misma se debe compilar el cuaderno Jupyter usando Quarto en un pdf que sería el documento de la memoria de cálculo.

### 3. RESULTADOS Y DISCUSIÓN

El modelo de elementos finitos logrado con la librería anaStruct de Python, tiene la misma calidad del que se conseguiría empleando paquetes de pago como SAP2000 o Midas, de tal forma que a partir del análisis de la estructura; el diseño y detallamiento puede llevarse a cabo con total seguridad y lograr documentación técnica como planos y memorias de cálculo a nivel constructivo.

A continuación, se presenta el documento compilado. Lo óptimo aquí es que en ningún momento es necesario salir del entorno de Python y a la vez se realiza análisis, diseño y memoria de cálculo. En este trabajo no se envían datos automáticos a AutoCad para la generación de planos, pero las librerías mencionadas antes tienen toda la capacidad para la realización de la tarea.

La figura 11 y todas sus sub figuras muestran el informe en toda su extensión; partiendo de la página 1 hasta la página 22. Se recuerda al lector que las celdas de código han sido escondidas para la generación del pdf, estando presentes los resultados de la ejecución de las mismas y las figuras logradas. Las imágenes detallan todo el proceso de análisis y diseño, incluyendo las normativas y condiciones asumidas. Se muestra también la disposición final de la geometría y de las armaduras, las cuales ya fueron tomadas de los planos constructivos, los cuales son fiel reflejo de las armaduras asumidas.

### 4. CONCLUSIONES

Los resultados logrados con el modelo de elementos finitos del cabezal de estribo que soporta la superestructura de un puente, generado mediante la librería anaStruct en Python es lo suficientemente robusto para ser empleado en análisis y diseño de elementos a nivel constructivo.

La integración de Python en el flujo de trabajo de la ingeniería civil, específicamente en el análisis y diseño estructural, ha demostrado ser altamente eficiente. Python, con sus extensas librerías como anaStruct, Numpy, y Quarto, permite automatizar y unificar procesos que tradicionalmente requerían múltiples herramientas de software. Esto no solo reduce el tiempo necesario para completar el ciclo de diseño, sino que también minimiza la posibilidad de errores y pérdida de información. Esto valida a Python como una alternativa viable y económica para profesionales e investigadores en el campo de la ingeniería civil.

Esta automatización no solo ahorra tiempo, sino que también garantiza una mayor coherencia y precisión en la presentación de los resultados.

Dada la flexibilidad y potencia de Python, se anticipa un aumento en su uso dentro de la ingeniería civil, no solo en el área estructural, sino también en otras disciplinas como la geotecnia e hidrología. La capacidad de Python para integrarse con diferentes herramientas y formatos lo convierte en un recurso valioso para enfrentar los desafíos futuros de la ingeniería civil.

## 5. BIBLIOGRAFÍA

- Ali, H. M., & Shakir, R. R. (2022). Applying a Python script to predict the geotechnical properties of the Nasiriyah soil. *IOP Conference Series: Earth and Environmental Science*, 961(1), 012004. <https://doi.org/10.1088/1755-1315/961/1/012004>
- Arroyo, J., Mora, F., Eithel Rodríguez, & Porras, K. (2024). Parametrizando ejercicios con Python y LaTeX: una novedosa estrategia para generación de materiales de enseñanza y evaluación en matemáticas. *Revista Digital: Matemática, Educación e Internet*, 24(2), 1–15.
- Déo, P. F. G. (2022). Desenvolvimento de software para engenharia de estruturas: Calculadora de flechas imediatas em vigas biapoiadas de concreto armado. <https://repositorio.ufscar.br/handle/ufscar/16716>
- Fangohr, H. (2018). Introduction To Python For Computational Science And Engineering. Zenodo. <https://doi.org/10.5281/ZENODO.1411868>
- He, L., Gilbert, M., & Song, X. (2019). A Python script for adaptive layout optimization of trusses. *Structural and Multidisciplinary Optimization*, 60(2), 835–847. <https://doi.org/10.1007/s00158-019-02226-6>
- He, L., Schiantella, M., Gilbert, M., & Smith, C. C. (2023). A Python script for discontinuity layout optimization. *Structural and Multidisciplinary Optimization*, 66(7), 152. <https://doi.org/10.1007/s00158-023-03583-z>
- Huijse, P. (2022). *Computación Científica con Python*. <https://github.com/phuijse/Python-Book>
- Mekonen, S. S., Boyce, S. E., Mohammed, A. K., & Disse, M. (2024). Using an Open-Source Tool to Develop a Three-Dimensional Hydrogeologic Framework of the Kobo Valley, Ethiopia. *Geosciences*, 14(1), Article 1. <https://doi.org/10.3390/geosciences14010003>
- Miranda, Y. G. J., Pinheiro, R. M., Junior, H. G. D., & Xavier, C. R. (2024). Modelamento estrutural de elementos finitos: Estrutura do Auditório William Monachesi do Campus Universitário Olezio Galotti. *Cadernos UniFOA*, 19(54), Article 54. <https://doi.org/10.47385/cadunifoa.v19.n54.4811>
- Ngeno, F., Omuto, C., & Biamah, E. (2024). Contribution of 3D model representation in subsurface geotechnical investigations. *African Journal of Engineering and Innovation*, 2(1).
- Orduz-Ducuara, J. A. (2013). Técnicas en informática educativa (TIE): LaTeX y Python (herramientas para la enseñanza de las ciencias). *Revista Mexicana de Bachillerato a Distancia*, 8(15), Article 15.
- Poore, G. M. (2015). PythonTeX: reproducible documents with LaTeX, Python, and more. *Computational Science & Discovery*, 8(1), Article 1.
- Sarma, L. S., Mallikarachchi, C., & Herath, S. (2023). Design-Informed Generative Modelling using Structural Optimization (arXiv:2308.06734). arXiv. <https://doi.org/10.48550/arXiv.2308.06734>

Triantafyllidis, C. P., & Papageorgiou, L. G. (2018). An integrated platform for intuitive mathematical programming modeling using LaTeX. PeerJ Computer Science, 4, e161.

## ANEXOS

### Descripción

La siguiente memoria de cálculo resume el análisis y diseño del cabezal de estribo de hormigón armado, soportada por fundación profunda consistente en cabezal y tres pilotes. El elemento es sub estructura del Puente Tomatitas - San Lorenzo.

### MATERIALES. -

Las características de los materiales a utilizarse en este proyecto son las siguientes:

#### Hormigón:

- Tipo A de 21 MPa resistencia cilíndrica característica a los 28 días a utilizarse en toda la estructura.

#### Armadura Pasiva:

- Acero estructural a utilizar en todo el puente con límite de fluencia mínimo de 420 MPa

#### Otros:

- Neopreno compuesto

### NORMA DE DISEÑO. -

El puente se diseñó en su totalidad empleando la Norma de diseño:

AASHTO LRFD Bridge Design Specifications (9th Edition, 2020)

### CARGAS.

De acuerdo a la normativa adoptada AASHTO LRFD 2020 las cargas aplicadas son las siguientes: Peso propio:

Se ha considerado como peso específico de los materiales los siguientes valores:

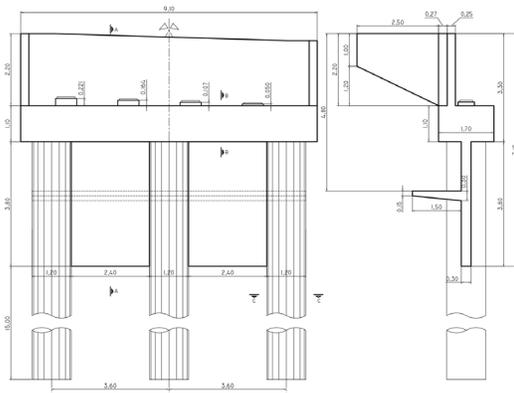
- Hormigón armado: 23,544 KN/m<sup>3</sup>
- Hormigón pretensado: 23,544 KN/m<sup>3</sup>
- Hormigón simple: 21,750 KN/m<sup>3</sup>

#### Carga de vehículos:

Se ha considerado en el diseño el vehículo HL-93.

*página 1 - materiales, normas y cargas*

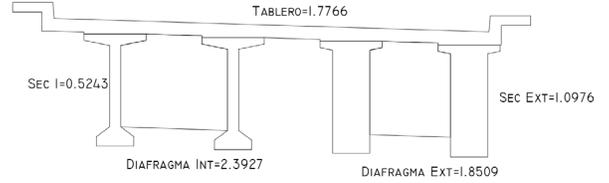
### Dimensiones estribo



*página 2 - geometría del elemento*

### Esquema de areas

Figure 1: Areas de elementos



### Peso de la superestructura

Table 1

Elemento	Area	Longitud	Volumen	Veces	Total
Tablero	1.777	30.600	54.364	1.000	54.364
Diafragma Int	2.393	0.250	0.598	6.000	3.589
Diafragma Ext	1.851	0.250	0.463	6.000	2.776
TOTAL LOSA					60.729
Sec I	0.524	27.400	14.366	4.000	57.463
Sec Ext	1.098	1.600	1.756	8.000	14.049
TOTAL VIGA					71.513

*página 3 – peso de la sección transversal*

$$loss_{volumen} = 60.729$$

$$viga_{volumen} = 71.513$$

$$total_{volumen} = loss_{volumen} + viga_{volumen} \\ = 60.729 + 71.513 \\ = 132.242$$

$$\gamma_{hormigon} = 23.544$$

$$total_{peso} = total_{volumen} \cdot \gamma_{hormigon} \\ = 132.242 \cdot 23.544 \\ = 3113.504$$

$$N_{apoyos} = 4$$

$$reaccion = \frac{total_{peso}}{N_{apoyos}} \cdot \frac{1}{2} \\ = \frac{3113.504}{4} \cdot \frac{1}{2} \\ = 389.188$$

### Reaccion longitudinal carga viva

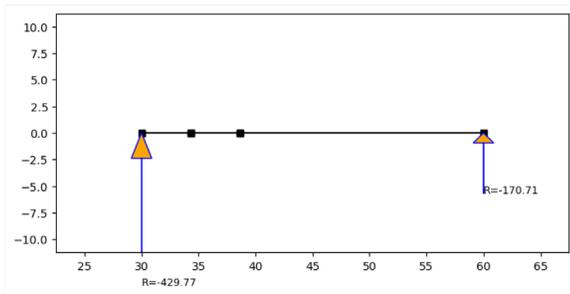
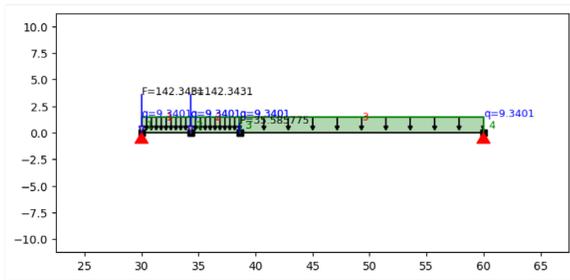
$$L = 30$$

$$P = 142.343$$

$$Q_{carril} = 9.340$$

*página 4- reacción longitudinal del peso propio y carga viva*

Memoria de Calculo Cabezal de Estribo

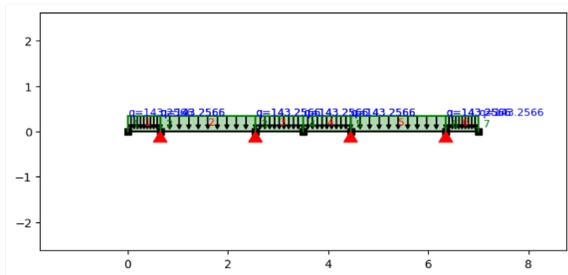
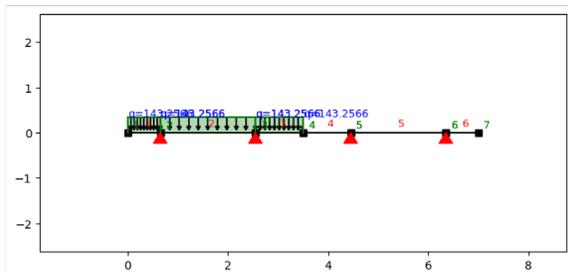


$$R = -429.770$$

$$\text{ancho}_{\text{carril}} = 3$$

página 5 – reacción longitudinal de carga viva

Memoria de Calculo Cabezal de Estribo



Solucion de la estructura

1\_carril

página 7- un carril cargado y dos carriles cargados

Memoria de Calculo Cabezal de Estribo

Distribucion transversal

$$\text{los}_{\text{ancho}} = 7$$

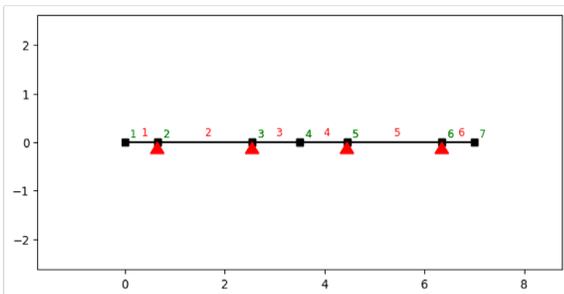
$$s = 1.900$$

$$N_{\text{vigas}} = 4$$

$$a = \frac{\text{los}_{\text{ancho}} - s \cdot (N_{\text{vigas}} - 1)}{2}$$

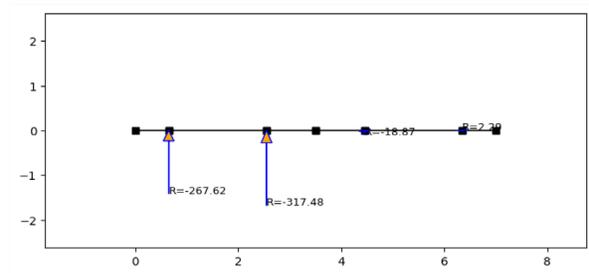
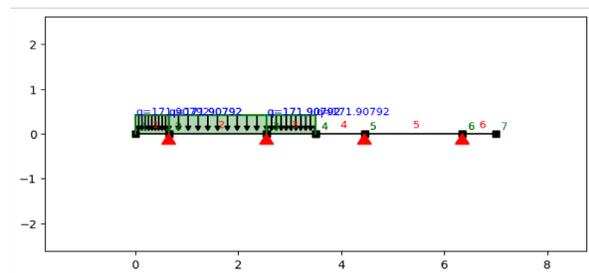
$$= \frac{7 - 1.900 \cdot (4 - 1)}{2}$$

$$= 0.650$$



página 6 – esquema para distribución transversal

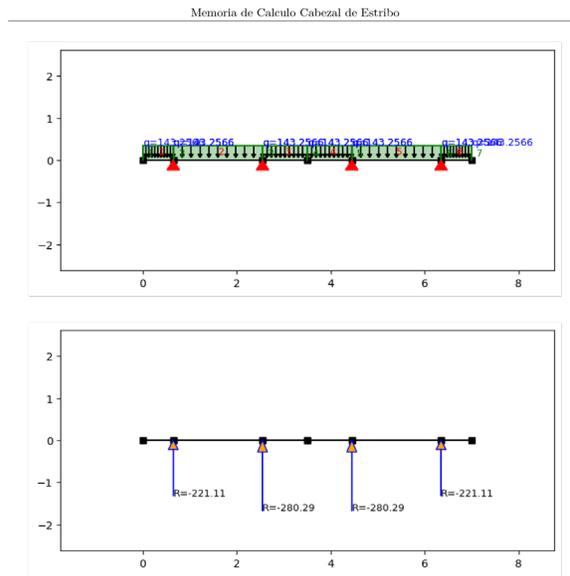
Memoria de Calculo Cabezal de Estribo



<IPython.core.display.Javascript object>

2\_carril

página 8 – reacciones de un carril cargado



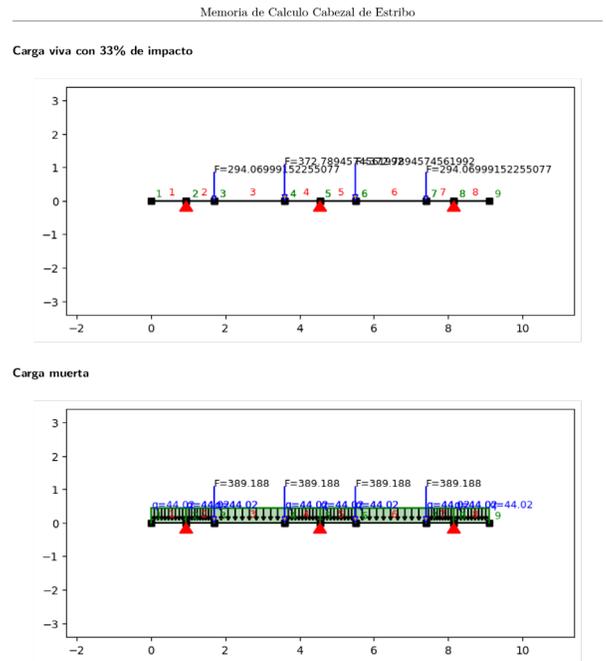
<IPython.core.display.Javascript object>

asumimos carga crítica de 2 carriles cargados, las reacciones a usar son:

Reaccion No 1 = -221.105  
lado izquierdo

Reaccion No 2 = -280.293  
Interior

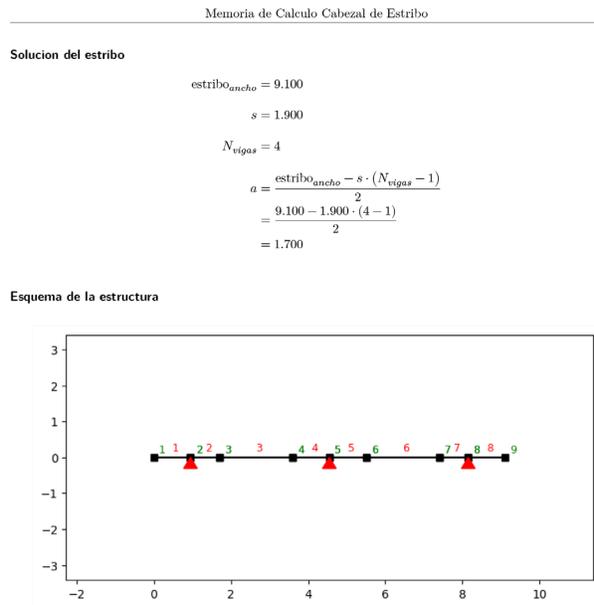
página 9 – reacciones de dos carriles cargados, es la que domina el diseño



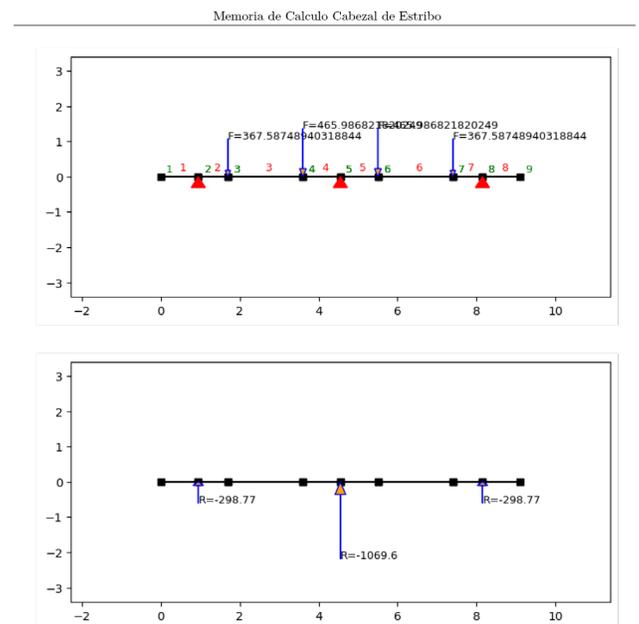
Solucion de los casos de carga y combinacion

1.25D + 1.75L

página 11- combinacion pésima

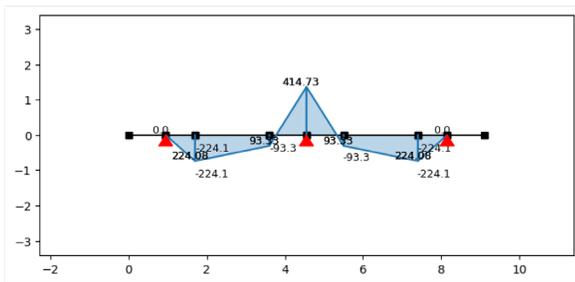
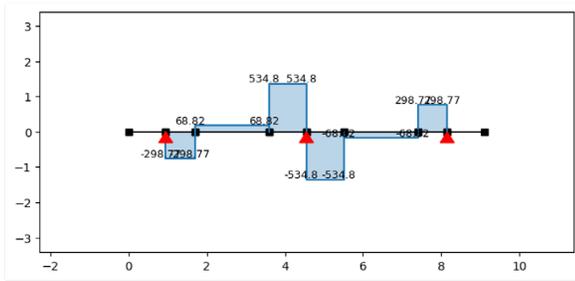


página 10- esquema del estribo que recibe las reacciones ya calculadas de la superestructura



página 12-reacciones en estribo carga muerta

Memoria de Calculo Cabezal de Estribo

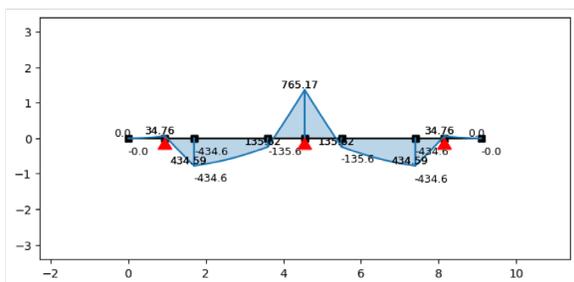
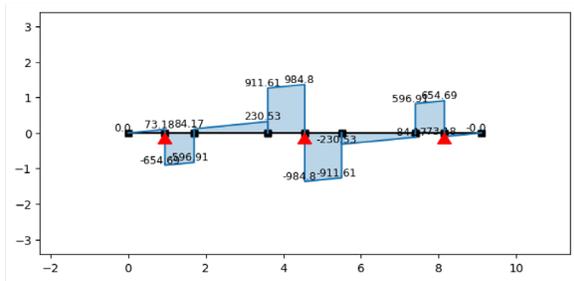


<IPython.core.display.Javascript object>

Dead

página 13-cortantes y momentos en el estribo por carga muerta

Memoria de Calculo Cabezal de Estribo

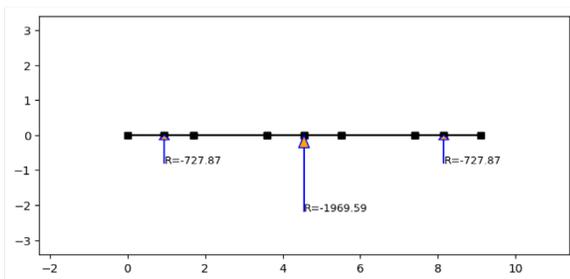
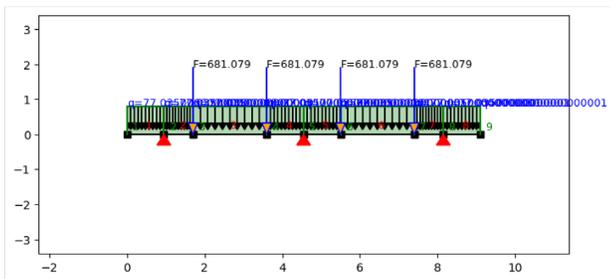


<IPython.core.display.Javascript object>

combination

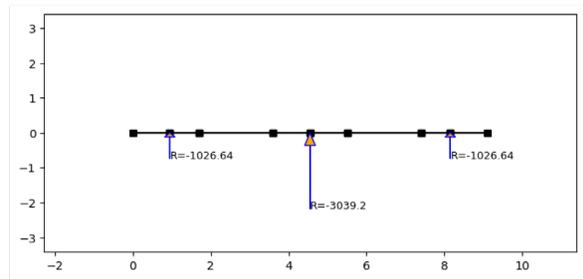
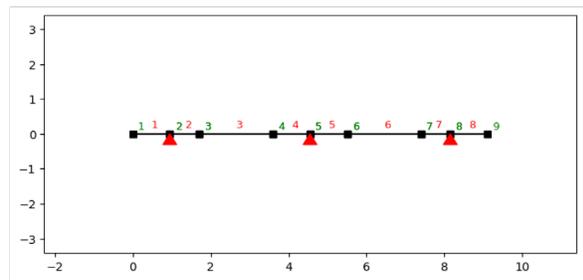
página 15 - cortantes y momentos de la carga viva

Memoria de Calculo Cabezal de Estribo



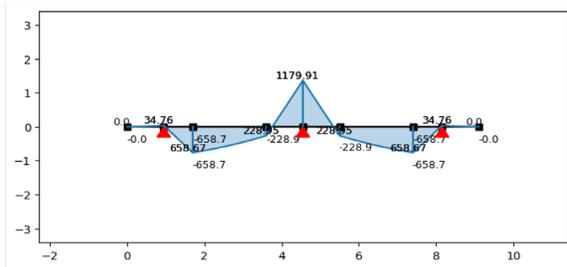
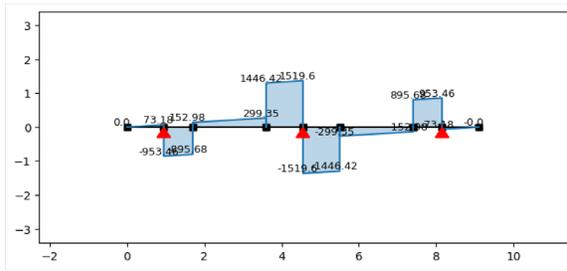
página 14-reacciones de la carga viva

Memoria de Calculo Cabezal de Estribo



página 16- reacciones combinación

Memoria de Calculo Cabezal de Estribo



<IPython.core.display.Javascript object>

página 17 – cortantes y momentos de la combinación

Memoria de Calculo Cabezal de Estribo

$$A_s = \left( 0.85 \cdot a \cdot b \cdot \frac{f_c}{f_y} \right) = \left( 0.85 \cdot 4.097 \text{ centimeter} \cdot 1.700 \text{ meter} \cdot \frac{21.000 \text{ megapascal}}{420.000 \text{ megapascal}} \right) = 29.601 \text{ centimeter}^2$$

$$\phi = 20.000 \text{ millimeter}$$

$$A_b = \left( \pi \cdot \frac{(\phi)^2}{4} \right) = \left( 3.142 \cdot \frac{(20.000 \text{ millimeter})^2}{4} \right) = 3.142 \text{ centimeter}^2$$

$$N = \frac{A_s}{A_b} = \frac{29.601 \text{ centimeter}^2}{3.142 \text{ centimeter}^2} = 9.422 \text{ dimensionless}$$

Asumimos 10φ20

Diseño a corte

$$V_u = 1519.599 \text{ kilonewton}$$

$$\beta = 0.500 \text{ megapascal}$$

$$c = 1.000 \text{ megapascal}$$

$$d_v = d - \frac{a}{2} = 1.075 \text{ meter} - \frac{4.097 \text{ centimeter}}{2} = 1.055 \text{ meter}$$

página 19- armadura para flexión e inicio de verificación a corte

Memoria de Calculo Cabezal de Estribo

Diseño de hormigon armado

Diseño a Flexion

$$f_c = 21.000 \text{ megapascal}$$

$$f_y = 420.000 \text{ megapascal}$$

$$M_u = 1179.908 \text{ kilonewton} \cdot \text{meter}$$

$$\Phi = 0.900$$

$$M_n = \frac{M_u}{\Phi} = \frac{1179.908 \text{ kilonewton} \cdot \text{meter}}{0.900} = 1311.009 \text{ kilonewton} \cdot \text{meter}$$

$$b = 1.700 \text{ meter}$$

$$h = 1.100 \text{ meter}$$

$$d_c = 0.025 \text{ meter}$$

$$d = h - d_c = 1.100 \text{ meter} - 0.025 \text{ meter} = 1.075 \text{ meter}$$

$$a = \left( d \cdot \left( 1 - \sqrt{1 - \frac{M_n}{0.425 \cdot b \cdot (d)^2 \cdot f_c}} \right) \right) = \left( 1.075 \text{ meter} \cdot \left( 1 - \sqrt{1 - \frac{1311.009 \text{ kilonewton} \cdot \text{meter}}{0.425 \cdot 1.700 \text{ meter} \cdot (1.075 \text{ meter})^2 \cdot 21.000 \text{ megapascal}}} \right) \right) = 4.097 \text{ centimeter}$$

página 18- momento ultimo y diseño de hormigón armado

Memoria de Calculo Cabezal de Estribo

$$V_{e1} = \left( \beta \cdot 0.083 \cdot d_v \cdot b \cdot \sqrt{\frac{f_c}{c}} \right) = \left( 0.500 \text{ megapascal} \cdot 0.083 \cdot 1.055 \text{ meter} \cdot 1.700 \text{ meter} \cdot \sqrt{\frac{21.000 \text{ megapascal}}{1.000 \text{ megapascal}}} \right) = 340.925 \text{ kilonewton}$$

$$V_{e2} = (0.25 \cdot f_c \cdot d_v \cdot b) = (0.25 \cdot 21.000 \text{ megapascal} \cdot 1.055 \text{ meter} \cdot 1.700 \text{ meter}) = 9411.547 \text{ kilonewton}$$

$$V_c = \min(V_{e1}, V_{e2}) = \min(340.925 \text{ kilonewton}, 9411.547 \text{ kilonewton}) = 340.925 \text{ kilonewton}$$

$$V_s = V_u - V_c = 1519.599 \text{ kilonewton} - 340.925 \text{ kilonewton} = 1178.674 \text{ kilonewton}$$

$$\phi = 12.000 \text{ millimeter}$$

$$A_b = \left( \pi \cdot \frac{(\phi)^2}{4} \right) = \left( 3.142 \cdot \frac{(12.000 \text{ millimeter})^2}{4} \right) = 1.131 \text{ centimeter}^2$$

$$A_s = 4 \cdot A_b = 4 \cdot 1.131 \text{ centimeter}^2 = 4.524 \text{ centimeter}^2$$

$$s = \left( A_s \cdot d_v \cdot \frac{f_y}{V_s} \right) = \left( 4.524 \text{ centimeter}^2 \cdot 1.055 \text{ meter} \cdot \frac{420.000 \text{ megapascal}}{1178.674 \text{ kilonewton}} \right) = 16.999 \text{ centimeter}$$

página 20 - determinación de la armadura a corte y su espaciamento

Asumimos 4φ12c/15

**Armadura de temperatura**

$$\begin{aligned}
 h &= 1.100 \text{ meter} \\
 b &= 1.700 \text{ meter} \\
 C &= 7500.000 \text{ megapascal} \\
 c &= 2.000 \frac{\text{meter}}{\text{centimeter}^2} \\
 A_{s_{temp}} &= \frac{C \cdot h \cdot b}{c \cdot (h + b) \cdot \gamma_y} \\
 &= \frac{7500.000 \text{ megapascal} \cdot 1.100 \text{ meter} \cdot 1.700 \text{ meter}}{2.000 \frac{\text{meter}}{\text{centimeter}^2} \cdot (1.100 \text{ meter} + 1.700 \text{ meter}) \cdot 420.000 \text{ megapascal}} \\
 &= 5.963 \text{ centimeter}^2
 \end{aligned}$$

φ = 12.000 millimeter

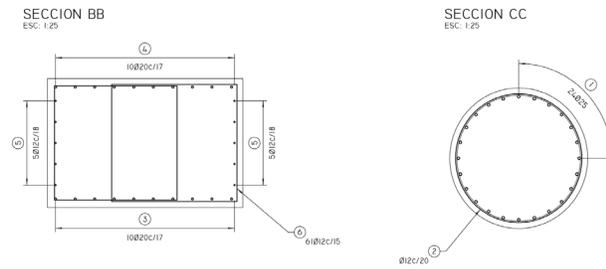
$$\begin{aligned}
 A_b &= \left( \pi \cdot \frac{(\phi)^2}{4} \right) \\
 &= \left( 3.142 \cdot \frac{(12.000 \text{ millimeter})^2}{4} \right) \\
 &= 1.131 \text{ centimeter}^2
 \end{aligned}$$

$$\begin{aligned}
 s &= \left( \frac{A_b}{A_{s_{temp}}} \cdot 1 \cdot \text{ureg.m} \right) \\
 &= \left( \frac{1.131 \text{ centimeter}^2}{5.963 \text{ centimeter}^2} \cdot 1 \cdot \text{ureg.m} \right) \\
 &= 18.966 \text{ centimeter}
 \end{aligned}$$

Asumimos 5φ12c/18

*página 21- armadura de temperatura de la sección*

**Disposicion final de armaduras**



*página 22-disposicion final de las armaduras, ya tomado de los planos constructivos*