

Universidad Simón Bolívar

**MODELADO DE ASCENSOR**

Elaborado por:

John Yanez 20-10667

Harold Cruz 20-10219

Angélica Tezara 20-10624

Sartenejas, 26 de marzo de 2025

**MODELADO DE UN ASCENSOR**

**Resumen**

Este informe presenta el desarrollo del proyecto "Modelado de un Ascensor", orientado a la simulación y optimización computacional de un sistema de elevación en un edificio residencial. Se implementaron tres módulos en Python—Caja-Motor, Contrapeso-Cabina y Capacidad-Intervalo—que calculan la potencia del motor, el equilibrio del sistema y la eficiencia operativa. Se integran fundamentos de física, mecánica y normativas (COVENIN 621-4) con el procesamiento de datos para validar el diseño

**1. Antecedentes**

El creciente número de edificios altos ha incrementado la importancia de contar con sistemas de ascensores seguros y eficientes. Tradicionalmente, el diseño de estos sistemas se basaba en modelos matemáticos simples y pruebas empíricas, pero el uso de la simulación computacional permite analizar escenarios complejos de forma rápida y precisa. La normativa COVENIN 621-4 establece los requisitos mínimos para el diseño de ascensores en Venezuela, abarcando aspectos como la capacidad de carga, la distribución de fuerzas y la seguridad en la transmisión de potencia.

El proyecto se apoya en las especificaciones de un motor Orona Next Rise, cuyos datos nominales incluyen:

* Capacidad: 8 personas.
* Carga: 630–1600 kg.
* Velocidad ajustable: 1.75 m/s, 2 m/s y 2.5 m/s.
* Recorrido máximo: 130 m.

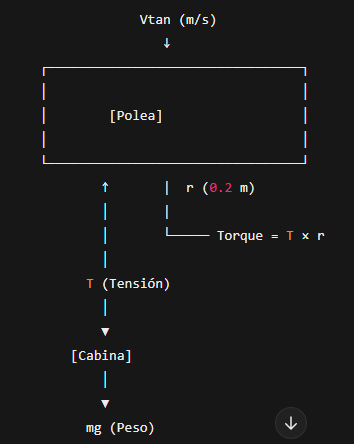
Se modela un edificio residencial con 20 pisos superiores y 2 sótanos, donde cada piso de sótano alberga 16 personas y cada piso superior 20. Para cumplir con la normativa y optimizar el funcionamiento, se propuso un sistema de ascensores distribuido en cuatro grupos (A, B, C para pasajeros y D para carga), con configuraciones de carga y velocidad ajustadas a cada función.

*[Sugerencia: Incluir un diagrama del edificio que muestre la distribución de pisos y la ubicación de los grupos de ascensores.]*

**2. Fundamentos Físicos del Modelo**

El modelo se fundamenta en varios principios de la física y la mecánica:

1. **Fuerzas y Equilibrio:** Las leyes de Newton determinan la tensión en los cables y la distribución de fuerzas en el sistema. El peso de la cabina y el contrapeso, cuando se combinan adecuadamente, permiten minimizar la carga sobre el motor, logrando un equilibrio mecánico.
2. **Torque y Potencia:** El torque, producto de la tensión por el radio de la polea, es la base para calcular la potencia necesaria para mover el ascensor. La potencia se ajusta considerando las pérdidas en la transmisión, con eficiencias establecidas (80% para la caja de cambios y 90% para el motor).
3. **Capacidad y Tiempos de Respuesta:** La capacidad de transporte y el intervalo entre viajes se calculan considerando el flujo de pasajeros y la configuración del edificio. Se aplican ecuaciones cinemáticas y de probabilidad para optimizar estos parámetros y minimizar los tiempos de espera.
4. **Aplicación Computacional de Ecuaciones Físicas:** La implementación de estos cálculos en Python permite automatizar y validar el modelo a través de simulaciones, facilitando el análisis de distintos escenarios y el ajuste de parámetros.



* **mg (Peso):** La fuerza ejercida por la cabina debido a la gravedad.
* **T (Tensión):** La fuerza en el cable, que en este modelo se asume igual a mg.
* **r (Radio de la polea):** La distancia desde el centro de la polea hasta el punto de aplicación de la tensión.
* **Torque:** Calculado como el producto de la tensión por el radio (T × r), es el giro que se aplica en la polea.
* **Vtan (Velocidad Tangencial):** La velocidad lineal en el borde de la polea, relacionada con la velocidad angular (ω) mediante la fórmula ω = Vtan / r.

*ugerencia: Incluir un diagrama que ilustre las fuerzas, el torque y la relación entre velocidad tangencial y angular.]*

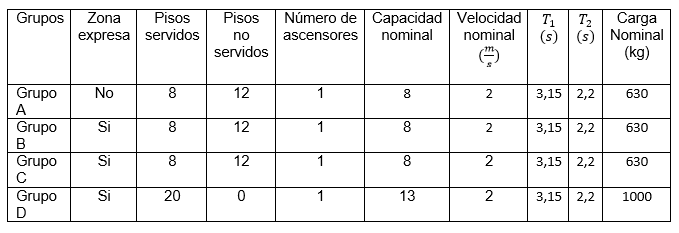
**3. Desarrollo del Sistema**

El proyecto se desarrolló mediante la creación de tres módulos en Python, cada uno responsable de un aspecto clave del modelo.

**3.1. Descripción General del Sistema**

El sistema de ascensores está diseñado para un edificio residencial con 20 pisos y 2 sótanos. Se implementaron cuatro grupos:

* **Grupos A, B y C:** Ascensores de pasajeros con una carga nominal de 630 kg y velocidad de 2 m/s. Estos tienen una cabina de aproximadamente 1.68 m² y dimensiones de 1.2 m × 1.4 m × 2.40 m.
* **Grupo D:** Ascensor de carga con una carga nominal de 1000 kg, adaptado a necesidades específicas de operación.

Cada grupo se evalúa en términos de capacidad de transporte, intervalo de espera, y tiempos de operación. Los resultados indican que los ascensores de pasajeros tienen una capacidad de aproximadamente 8.08 % y un intervalo de 135.56 s, mientras que el ascensor de carga muestra una capacidad de 6.65 % y un intervalo de 164.53 s. Aunque el grupo D no cumple totalmente con la normativa, es aceptable dada su función de uso aislado.

**3.2. Código de Caja-Motor**

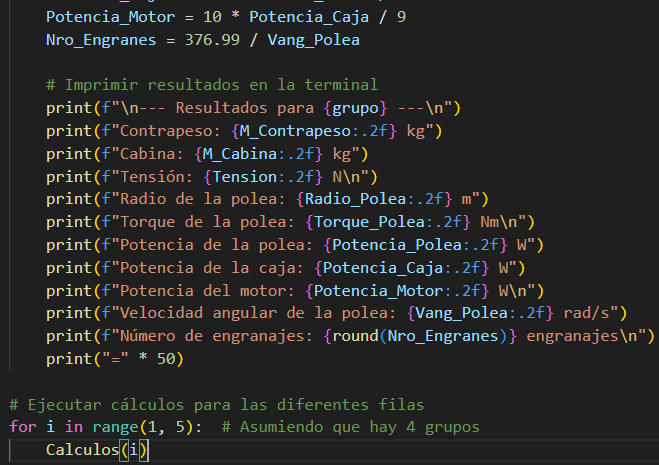
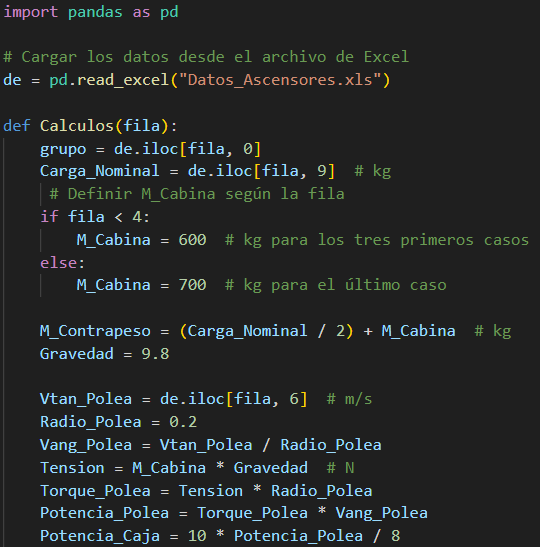
**3.2.1. Funcionalidad y Propósito**

El módulo *Caja\_Motor.py* se encarga de calcular los parámetros mecánicos fundamentales para la transmisión de energía en el sistema de ascensores. Su propósito es:

* Calcular la masa de la cabina y el contrapeso.
* Determinar la tensión en los cables y el torque en la polea.
* Estimar la potencia necesaria en la polea, en la caja de cambios y en el motor.
* Calcular el número de engranajes requeridos para reducir la velocidad del motor a la velocidad operativa del ascensor.

**3.2.2. Implementación**

El código extrae los datos de un archivo Excel y utiliza la librería Pandas para gestionar la información. La función principal, Calculos(fila), procesa cada grupo de ascensores. Se asignan diferentes valores de masa para la cabina dependiendo del grupo (600 kg para los ascensores de pasajeros y 700 kg para el de carga), y se calcula la masa del contrapeso sumando la mitad de la carga nominal a la masa de la cabina.

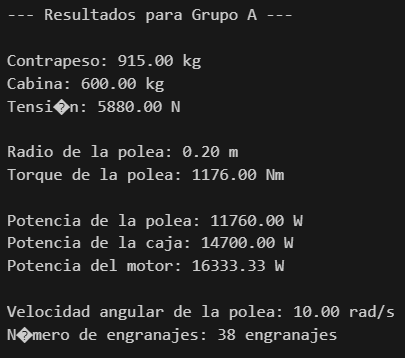
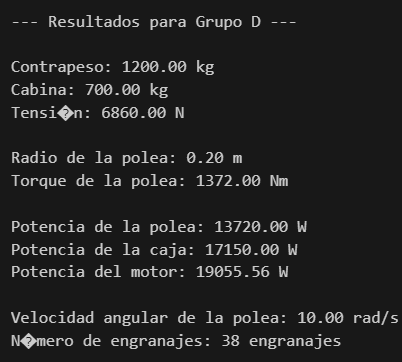
La velocidad angular se obtiene dividiendo la velocidad tangencial por el radio de la polea (0.2 m), y a partir de ahí se calcula la tensión y el torque. La potencia se deriva del producto del torque y la velocidad angular, ajustándose para las eficiencias del 80% y 90%.

**3.2.3. Resultados**

El módulo imprime en la terminal resultados como:Contrapeso y masa de la cabina.

* Tensión en los cables y torque de la polea.
* Potencia en la polea, caja de cambios y motor.
* Número de engranajes necesarios.

Estos resultados permiten validar que la transmisión de energía se realiza de manera óptima y que el diseño cumple con los requerimientos normativos.



**3.3. Código de Contrapeso y Cabina**

**3.3.1. Funcionalidad y Propósito**

El módulo *Contrapeso\_Cabina.py* se encarga de calcular la masa del contrapeso y definir las dimensiones de la cabina del ascensor, asegurando la estabilidad y seguridad del sistema. Es fundamental para equilibrar el ascensor y reducir la carga sobre el motor.

**3.3.2. Implementación**

Este módulo también utiliza la librería Pandas para extraer datos del archivo Excel. La función Carga(fila, grupo) calcula:

* La carga nominal.
* El peso de la cabina (definido como la mitad de la carga nominal).
* El contrapeso (1.5 veces el peso de la cabina).

Además, se asignan dimensiones a la cabina según el grupo; los ascensores de pasajeros tienen dimensiones menores, mientras que el ascensor de carga tiene dimensiones mayores. La superficie de la cabina se calcula para asegurar que cumpla con los requisitos normativos.

El módulo también evalúa el factor de seguridad de los cables de suspensión, utilizando parámetros como la carga de rotura y el número de cables.

**3.3.3. Resultados**

La salida del módulo muestra:

* La carga nominal y el peso de la cabina.
* El contrapeso calculado.
* Las dimensiones de la cabina y la superficie resultante.
* El factor de seguridad de los cables de suspensión.

Estos resultados son esenciales para garantizar que el ascensor opere de manera segura y cumpla con la normativa COVENIN.

**3.4. Código de Capacidad e Intervalo**

**3.4.1. Funcionalidad y Propósito**

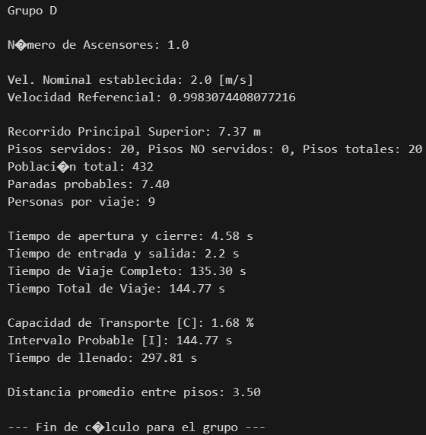
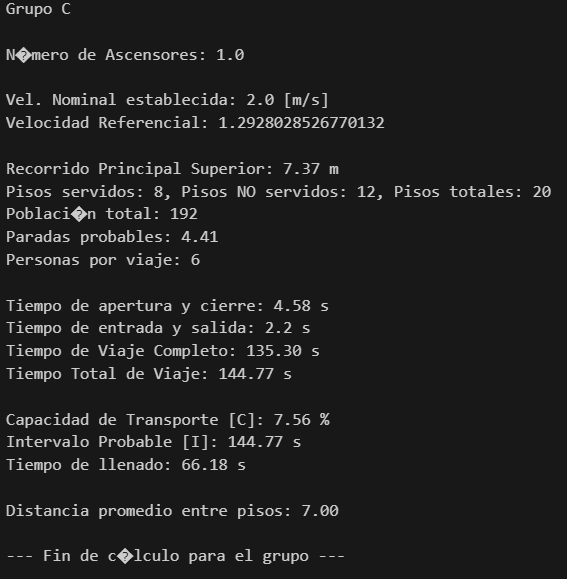
El módulo *Calculo\_Capacidad\_Intervalo.py* se centra en determinar la capacidad de transporte del ascensor y el intervalo probable entre viajes. Este análisis es crucial para optimizar el flujo de pasajeros y asegurar que el sistema responda de manera eficiente a la demanda en el edificio.

**3.4.2. Implementación**

El código importa las librerías numpy y pandas para realizar cálculos numéricos y gestionar datos. La función main(fila) extrae parámetros relevantes, como el número de pisos servidos, la capacidad nominal, la velocidad establecida y el número de ascensores. Además, define constantes importantes como la población por piso y la distancia promedio entre pisos.

El módulo calcula la distancia promedio entre pisos de manera dinámica, evalúa si el ascensor cuenta con zona expresa (una característica que afecta el tiempo de viaje) y determina la población total del edificio.

Se calculan las variables:

* Número de personas por viaje.
* Número de paradas probables.
* Tiempo de viaje completo, considerando diferentes escenarios (zona expresa o no).
* Capacidad de transporte en porcentaje.
* Intervalo probable entre ascensores y tiempo de llenado.

**3.4.3. Resultados**

El módulo imprime en la terminal:

* El tiempo total de viaje.
* La capacidad de transporte (en %).
* El intervalo probable entre ascensores.
* El tiempo de llenado.

Estos resultados permiten evaluar la eficiencia del sistema y ajustar los parámetros de operación para minimizar tiempos de espera y maximizar el flujo de pasajeros.

**4. Pruebas y Resultados Integrados**

La integración de los tres módulos permite simular el comportamiento global del sistema de ascensores. Las pruebas realizadas muestran que:

* **Caja-Motor:** Proporciona un cálculo preciso de la potencia y torque, validando el diseño de la transmisión.
* **Contrapeso-Cabina:** Garantiza el equilibrio del sistema mediante el cálculo correcto de las masas y dimensiones, y confirma la seguridad de los cables.
* **Capacidad-Intervalo:** Evalúa el rendimiento del ascensor en términos de capacidad y tiempos de operación, asegurando que se atiende de forma eficiente la demanda.

*[Sugerencia: Adjunte capturas de pantalla o gráficos de salida que resuman los resultados de cada módulo, y un diagrama de flujo que integre el funcionamiento de todo el sistema.]*

Los resultados integrados confirman que el modelo cumple en gran medida con las especificaciones normativas, aunque se observan pequeñas discrepancias en el ascensor de carga, lo cual es aceptable dada su función de uso esporádico.

**5. Conclusiones y Recomendaciones**

**5.1. Conclusiones**

El proyecto "Modelado de un Ascensor" demuestra la viabilidad de utilizar la programación en Python para simular sistemas complejos de elevación. Entre los logros alcanzados se destacan:

* **Automatización de Cálculos:** La implementación de los módulos permite realizar cálculos precisos de parámetros críticos como la potencia del motor, el torque, la capacidad de transporte y el equilibrio del sistema.
* **Integración Teórico-Práctica:** Los fundamentos físicos, basados en dinámica rotacional, equilibrio mecánico y cinemática, se aplican de manera efectiva en los algoritmos, lo que garantiza que el modelo sea robusto y representativo.
* **Eficiencia y Seguridad:** Los resultados obtenidos muestran que el diseño cumple en gran medida con la normativa COVENIN, optimizando el flujo de pasajeros y asegurando la estabilidad del sistema.
* **Escalabilidad:** El enfoque modular permite adaptar el modelo a diferentes configuraciones y escalas, lo que es ideal para futuras ampliaciones o modificaciones del sistema.

**5.2. Recomendaciones**

Para mejorar y expandir el modelo, se sugieren las siguientes acciones:

1. **Desarrollo de una Interfaz Gráfica:** Implementar una interfaz de usuario interactiva que facilite la entrada de datos y la visualización en tiempo real de los resultados, permitiendo un análisis más intuitivo.
2. **Simulaciones Dinámicas:** Integrar simulaciones avanzadas que muestren el comportamiento del ascensor en diferentes escenarios de carga y operación, lo que ayudará a validar y ajustar el modelo.
3. **Optimización del Código:** Revisar y mejorar la eficiencia computacional utilizando técnicas de vectorización y paralelización, especialmente para escenarios con grandes volúmenes de datos.
4. **Documentación Completa:** Mantener una documentación detallada del código y de los cálculos implementados para facilitar futuras modificaciones y el mantenimiento del sistema.
5. **Pruebas de Campo:** Realizar pruebas en entornos simulados o reales para comparar los resultados del modelo con datos operativos y ajustar los parámetros según sea necesario.
6. **Ampliación del Modelo:** Considerar la integración de variables adicionales, como la variabilidad en la demanda a lo largo del día, para hacer el modelo más robusto y adaptable a escenarios reales.

**6. Bibliografía de Soporte**

1. **Normativa COVENIN 621-4:** "Ascensores y montacargas. Requisitos de seguridad y diseño."
2. **Documentación Técnica del Motor Orona Next Rise.**
3. **Principios de Física Aplicada a Sistemas de Elevación y Transmisión Mecánica.**
4. **Documentación de Python y Pandas:**
   * <https://docs.python.org/3/>
   * https://pandas.pydata.org/pandas-docs/stable/
5. **Teoría de Tráfico Vertical y Modelado de Sistemas de Transporte en Edificios.**