Voy a crear una calculadora completa del impuesto de renta para personas naturales residentes en Colombia (año gravable 2024), siguiendo todas las reglas del Estatuto Tributario colombiano que me has proporcionado.

Calculadora Impuesto de Renta 2024

Artefacto interactivo

Reintentar

JS

Continuar

Calculadora Impuesto de Renta 2024

Artefacto interactivo ∙ Versión 2

He creado una **calculadora completa e interactiva del Impuesto de Renta 2024** para personas naturales residentes en Colombia. La herramienta incluye:

**✅ Características Implementadas:**

**1. Captura de 19 Variables de Entrada:**

* Datos del contribuyente (nombre, NIT)
* Ingresos laborales (salarios, cesantías, prestaciones, otros pagos)
* INCR (salud, pensiones)
* Rentas exentas (pensión voluntaria, AFC)
* Deducciones (dependientes, intereses vivienda, medicina prepagada)
* Beneficios tributarios (factura electrónica, GMF)
* Ingreso mensual promedio (para cesantías exentas)
* Datos año anterior y retenciones

**2. Cálculos Aplicados (Art. 336 ET):**

* ✅ Cesantías exentas según Art. 206 numeral 4 (tabla progresiva por UVT)
* ✅ Renta exenta 25% limitada a 790 UVT
* ✅ Pensión voluntaria + AFC (máx. 30% ingreso total o 3,800 UVT)
* ✅ Deducción dependientes (máx. 384 UVT)
* ✅ Deducción medicina prepagada (máx. 192 UVT)
* ✅ Deducción intereses vivienda (máx. 1,200 UVT)
* ✅ **Límite del 40% del ingreso neto y máximo 1,340 UVT**
* ✅ Beneficio factura electrónica (1% máx. 240 UVT)
* ✅ Beneficio GMF (50%)

**3. Liquidación del Impuesto:**

* ✅ Aplicación tabla Art. 241 del ET
* ✅ Cálculo anticipo año siguiente Art. 807 (Método 1 y 2)
* ✅ Toma el menor de los dos métodos
* ✅ Resultado final: Saldo a pagar o saldo a favor

**4. Diseño:**

* Interfaz con colores azul y blanco (profesional y sobria)
* Diseño responsive y organizado por secciones
* Resultados detallados con resumen visual
* Valores en formato COP (pesos colombianos)

La calculadora está lista para usar con la **UVT 2024 = $47,065** y aplica todas las reglas del Estatuto Tributario según tus especificaciones. 🎯

Reintentar

JS

muestra el código del programa, en python

Calculadora Impuesto de Renta 2024 - Python

Código

""" CALCULADORA IMPUESTO DE RENTA AÑO GRAVABLE 2024 Personas Naturales Residentes Fiscales - Colombia Basado en: Estatuto Tributario Colombiano - Ley 2277 de 2022 - Decreto 1625 de 2016 """ def calcular\_cesantias\_exentas(cesantias, ingreso\_promedio\_

Te he creado el código completo en **Python** de la calculadora de impuesto de renta. El programa incluye:

**📋 Características del Código:**

**1. Funciones Modulares:**

* calcular\_cesantias\_exentas() - Aplica la tabla del Art. 206 numeral 4
* aplicar\_tabla\_articulo\_241() - Aplica los tramos de impuesto
* calcular\_anticipo() - Calcula anticipos por método 1 y 2
* capturar\_datos() - Captura las 19 variables de entrada
* calcular\_impuesto\_renta() - Realiza todo el cálculo según Art. 336
* imprimir\_resultados() - Muestra resultados detallados y formateados

**2. Cálculos Implementados:**

✅ UVT 2024 = $47,065  
✅ Cesantías exentas (tabla progresiva)  
✅ Límites de rentas exentas (790 UVT, 3,800 UVT)  
✅ Límites de deducciones (384, 192, 1,200 UVT)  
✅ **Límite del 40% y 1,340 UVT (Art. 336)**  
✅ Beneficio factura electrónica (1% máx. 240 UVT)  
✅ Beneficio GMF (50%)  
✅ Tabla Art. 241 (6 tramos)  
✅ Anticipo Art. 807 (compara métodos 1 y 2)

**3. Interfaz de Usuario:**

* Captura interactiva de datos por consola
* Formato de moneda colombiana ($X.XXX.XXX)
* Resultados detallados paso a paso
* Opción para realizar múltiples cálculos

**4. Para Ejecutar:**

bash

python calculadora\_renta\_2024.py

El programa te pedirá los 19 datos de entrada y te mostrará el resultado final: **SALDO A PAGAR** o **SALDO A FAVOR** 🎯

"""

CALCULADORA IMPUESTO DE RENTA AÑO GRAVABLE 2024

Personas Naturales Residentes Fiscales - Colombia

Basado en: Estatuto Tributario Colombiano - Ley 2277 de 2022 - Decreto 1625 de 2016

"""

def calcular\_cesantias\_exentas(cesantias, ingreso\_promedio\_mensual, uvt\_2024):

"""

Calcula las cesantías exentas según Art. 206 numeral 4 del ET

"""

ingreso\_uvt = ingreso\_promedio\_mensual / uvt\_2024

if ingreso\_uvt <= 350:

return cesantias

elif ingreso\_uvt <= 410:

return cesantias \* 0.90

elif ingreso\_uvt <= 470:

return cesantias \* 0.80

elif ingreso\_uvt <= 530:

return cesantias \* 0.60

elif ingreso\_uvt <= 590:

return cesantias \* 0.40

elif ingreso\_uvt <= 650:

return cesantias \* 0.20

else:

return 0

def aplicar\_tabla\_articulo\_241(base\_gravable\_uvt, uvt\_2024):

"""

Aplica la tabla del artículo 241 del ET para calcular el impuesto

"""

if base\_gravable\_uvt <= 1090:

return 0

elif base\_gravable\_uvt <= 1700:

return (base\_gravable\_uvt - 1090) \* 0.19 \* uvt\_2024

elif base\_gravable\_uvt <= 4100:

return ((base\_gravable\_uvt - 1700) \* 0.28 + 116) \* uvt\_2024

elif base\_gravable\_uvt <= 8670:

return ((base\_gravable\_uvt - 4100) \* 0.33 + 788) \* uvt\_2024

elif base\_gravable\_uvt <= 18970:

return ((base\_gravable\_uvt - 8670) \* 0.35 + 2296) \* uvt\_2024

elif base\_gravable\_uvt <= 31000:

return ((base\_gravable\_uvt - 18970) \* 0.37 + 5901) \* uvt\_2024

else:

return ((base\_gravable\_uvt - 31000) \* 0.39 + 10352) \* uvt\_2024

def calcular\_anticipo(impuesto\_neto\_actual, impuesto\_neto\_anterior,

retenciones, num\_anos\_declarando):

"""

Calcula el anticipo del año siguiente según Art. 807 del ET

Compara método 1 y método 2, retorna el menor

"""

# Determinar porcentaje según años declarando

if num\_anos\_declarando == 1:

porcentaje = 0.25

elif num\_anos\_declarando == 2:

porcentaje = 0.50

else:

porcentaje = 0.75

# MÉTODO 1

anticipo\_metodo1 = (impuesto\_neto\_actual \* porcentaje) - retenciones

anticipo\_metodo1 = max(anticipo\_metodo1, 0)

# MÉTODO 2

if impuesto\_neto\_anterior > 0:

promedio = (impuesto\_neto\_actual + impuesto\_neto\_anterior) / 2

anticipo\_metodo2 = (promedio \* porcentaje) - retenciones

anticipo\_metodo2 = max(anticipo\_metodo2, 0)

else:

anticipo\_metodo2 = anticipo\_metodo1

# Retornar el menor

anticipo\_definitivo = min(anticipo\_metodo1, anticipo\_metodo2)

return anticipo\_metodo1, anticipo\_metodo2, anticipo\_definitivo

def formatear\_moneda(valor):

"""Formatea valores en pesos colombianos"""

return f"${valor:,.0f}".replace(",", ".")

def capturar\_datos():

"""Captura los 19 datos de entrada del usuario"""

print("\n" + "="\*70)

print(" " \* 10 + "CALCULADORA IMPUESTO DE RENTA AÑO GRAVABLE 2024")

print(" " \* 15 + "Personas Naturales Residentes - Colombia")

print("="\*70)

datos = {}

# INFORMACIÓN DEL CONTRIBUYENTE

print("\n--- INFORMACIÓN DEL CONTRIBUYENTE ---")

datos['nombre'] = input("Nombres y Apellidos: ")

datos['nit'] = input("NIT / Cédula: ")

# INGRESOS LABORALES

print("\n--- INGRESOS LABORALES ---")

datos['salarios'] = float(input("Salarios: $") or 0)

datos['cesantias'] = float(input("Cesantías Pagadas o Consignadas: $") or 0)

datos['prestaciones\_sociales'] = float(input("Prestaciones Sociales: $") or 0)

datos['otros\_pagos\_laborales'] = float(input("Otros Pagos Laborales: $") or 0)

datos['ingreso\_mensual\_promedio'] = float(input("Ingreso Mensual Promedio (últimos 6 meses): $") or 0)

# INGRESOS NO CONSTITUTIVOS DE RENTA

print("\n--- INGRESOS NO CONSTITUTIVOS DE RENTA (INCR) ---")

datos['incr\_salud'] = float(input("INCR Salud: $") or 0)

datos['incr\_pensiones'] = float(input("INCR Pensiones: $") or 0)

# RENTAS EXENTAS

print("\n--- RENTAS EXENTAS ---")

datos['pension\_voluntaria'] = float(input("Pensión Voluntaria: $") or 0)

datos['afc'] = float(input("Cuentas AFC: $") or 0)

# DEDUCCIONES

print("\n--- DEDUCCIONES ---")

datos['num\_dependientes'] = int(input("Número de Dependientes: ") or 0)

datos['intereses\_vivienda'] = float(input("Intereses de Vivienda: $") or 0)

datos['medicina\_prepagada'] = float(input("Medicina Prepagada: $") or 0)

# OTROS BENEFICIOS

print("\n--- OTROS BENEFICIOS TRIBUTARIOS ---")

datos['compras\_factura\_electronica'] = float(input("Compras con Factura Electrónica: $") or 0)

datos['gmf'] = float(input("GMF (Gravamen Movimientos Financieros): $") or 0)

# INFORMACIÓN AÑO ANTERIOR

print("\n--- INFORMACIÓN AÑO ANTERIOR Y RETENCIONES ---")

datos['impuesto\_neto\_anterior'] = float(input("Impuesto Neto Año Anterior: $") or 0)

datos['saldo\_favor\_anterior'] = float(input("Saldo a Favor Año Anterior (sin solicitud): $") or 0)

datos['retenciones'] = float(input("Retenciones Practicadas: $") or 0)

datos['anticipo\_anterior'] = float(input("Anticipo Año Anterior: $") or 0)

datos['num\_anos\_declarando'] = int(input("Número de Años Declarando (1, 2, o 3+): ") or 1)

return datos

def calcular\_impuesto\_renta(datos):

"""

Realiza el cálculo completo del impuesto de renta

según el Estatuto Tributario Colombiano - Art. 336

"""

UVT\_2024 = 47065

resultados = {}

# 1. INGRESOS TOTALES

ingresos\_totales = (datos['salarios'] + datos['cesantias'] +

datos['prestaciones\_sociales'] + datos['otros\_pagos\_laborales'])

resultados['ingresos\_totales'] = ingresos\_totales

# 2. INGRESOS NO CONSTITUTIVOS DE RENTA (INCR)

cesantias\_exentas = calcular\_cesantias\_exentas(

datos['cesantias'],

datos['ingreso\_mensual\_promedio'],

UVT\_2024

)

incr\_total = datos['incr\_salud'] + datos['incr\_pensiones'] + cesantias\_exentas

resultados['cesantias\_exentas'] = cesantias\_exentas

resultados['incr\_total'] = incr\_total

# 3. INGRESO NETO

ingreso\_neto = ingresos\_totales - incr\_total

resultados['ingreso\_neto'] = ingreso\_neto

# 4. RENTAS EXENTAS

# Renta exenta 25% limitada a 790 UVT

limite\_25\_porciento = ingreso\_neto \* 0.25

limite\_790\_uvt = 790 \* UVT\_2024

renta\_exenta\_25 = min(limite\_25\_porciento, limite\_790\_uvt)

# Pensión voluntaria + AFC (máximo 30% ingreso total o 3,800 UVT)

total\_pension\_afc = datos['pension\_voluntaria'] + datos['afc']

limite\_30\_porciento = ingresos\_totales \* 0.30

limite\_3800\_uvt = 3800 \* UVT\_2024

pension\_afc\_limitada = min(total\_pension\_afc, limite\_30\_porciento, limite\_3800\_uvt)

rentas\_exentas\_totales = renta\_exenta\_25 + pension\_afc\_limitada

resultados['renta\_exenta\_25'] = renta\_exenta\_25

resultados['pension\_afc\_limitada'] = pension\_afc\_limitada

resultados['rentas\_exentas\_totales'] = rentas\_exentas\_totales

# 5. DEDUCCIONES

# Dependientes (máximo 384 UVT)

limite\_384\_uvt = 384 \* UVT\_2024

deduccion\_dependientes = min(datos['num\_dependientes'] \* (32 \* UVT\_2024), limite\_384\_uvt)

# Medicina prepagada (máximo 192 UVT)

limite\_192\_uvt = 192 \* UVT\_2024

deduccion\_medicina = min(datos['medicina\_prepagada'], limite\_192\_uvt)

# Intereses vivienda (máximo 1,200 UVT)

limite\_1200\_uvt = 1200 \* UVT\_2024

deduccion\_intereses = min(datos['intereses\_vivienda'], limite\_1200\_uvt)

deducciones\_totales = deduccion\_dependientes + deduccion\_medicina + deduccion\_intereses

resultados['deduccion\_dependientes'] = deduccion\_dependientes

resultados['deduccion\_medicina'] = deduccion\_medicina

resultados['deduccion\_intereses'] = deduccion\_intereses

resultados['deducciones\_totales'] = deducciones\_totales

# 6. LÍMITE DEL 40% - ARTÍCULO 336

suma\_rentas\_deducciones = rentas\_exentas\_totales + deducciones\_totales

limite\_40\_porciento = ingreso\_neto \* 0.40

limite\_1340\_uvt = 1340 \* UVT\_2024

limite\_maximo\_depuracion = min(limite\_40\_porciento, limite\_1340\_uvt)

depuracion\_final = min(suma\_rentas\_deducciones, limite\_maximo\_depuracion)

resultados['suma\_rentas\_deducciones'] = suma\_rentas\_deducciones

resultados['limite\_maximo\_depuracion'] = limite\_maximo\_depuracion

resultados['depuracion\_final'] = depuracion\_final

# 7. RENTA LÍQUIDA ANTES DE OTROS BENEFICIOS

renta\_liquida = ingreso\_neto - depuracion\_final

# 8. BENEFICIO COMPRAS CON FACTURA ELECTRÓNICA (1% hasta 240 UVT)

limite\_240\_uvt = 240 \* UVT\_2024

beneficio\_factura = min(datos['compras\_factura\_electronica'] \* 0.01, limite\_240\_uvt)

renta\_liquida -= beneficio\_factura

resultados['beneficio\_factura'] = beneficio\_factura

# 9. BENEFICIO GMF (50%)

beneficio\_gmf = datos['gmf'] \* 0.50

renta\_liquida -= beneficio\_gmf

resultados['beneficio\_gmf'] = beneficio\_gmf

# 10. BASE GRAVABLE

base\_gravable = max(renta\_liquida, 0)

base\_gravable\_uvt = base\_gravable / UVT\_2024

resultados['base\_gravable'] = base\_gravable

resultados['base\_gravable\_uvt'] = base\_gravable\_uvt

# 11. APLICAR TABLA ARTÍCULO 241

impuesto\_neto = aplicar\_tabla\_articulo\_241(base\_gravable\_uvt, UVT\_2024)

resultados['impuesto\_neto'] = impuesto\_neto

# 12. CÁLCULO DEL ANTICIPO (Artículo 807)

anticipo\_m1, anticipo\_m2, anticipo\_definitivo = calcular\_anticipo(

impuesto\_neto,

datos['impuesto\_neto\_anterior'],

datos['retenciones'],

datos['num\_anos\_declarando']

)

resultados['anticipo\_metodo1'] = anticipo\_m1

resultados['anticipo\_metodo2'] = anticipo\_m2

resultados['anticipo\_definitivo'] = anticipo\_definitivo

# 13. LIQUIDACIÓN FINAL

liquidacion\_final = (impuesto\_neto - datos['retenciones'] -

datos['saldo\_favor\_anterior'] - datos['anticipo\_anterior'])

resultados['es\_saldo\_favor'] = liquidacion\_final < 0

resultados['valor\_final'] = abs(liquidacion\_final)

return resultados

def imprimir\_resultados(datos, resultados):

"""Imprime los resultados de forma detallada"""

print("\n" + "="\*70)

print(" " \* 20 + "RESULTADOS DE LA LIQUIDACIÓN")

print("="\*70)

print(f"\nContribuyente: {datos['nombre']}")

print(f"NIT/Cédula: {datos['nit']}")

print("\n--- DEPURACIÓN DE LA RENTA ---")

print(f"Ingresos Totales: {formatear\_moneda(resultados['ingresos\_totales'])}")

print(f"(-) INCR Total: {formatear\_moneda(resultados['incr\_total'])}")

print(f" (Cesantías Exentas: {formatear\_moneda(resultados['cesantias\_exentas'])})")

print(f"Ingreso Neto: {formatear\_moneda(resultados['ingreso\_neto'])}")

print(f"(-) Rentas Exentas: {formatear\_moneda(resultados['rentas\_exentas\_totales'])}")

print(f" (Renta Exenta 25%: {formatear\_moneda(resultados['renta\_exenta\_25'])})")

print(f" (Pensión Vol. + AFC: {formatear\_moneda(resultados['pension\_afc\_limitada'])})")

print(f"(-) Deducciones: {formatear\_moneda(resultados['deducciones\_totales'])}")

print(f" (Dependientes: {formatear\_moneda(resultados['deduccion\_dependientes'])})")

print(f" (Medicina Prepagada: {formatear\_moneda(resultados['deduccion\_medicina'])})")

print(f" (Intereses Vivienda: {formatear\_moneda(resultados['deduccion\_intereses'])})")

print(f"\nLímite 40% Art. 336: {formatear\_moneda(resultados['limite\_maximo\_depuracion'])}")

print(f"Depuración Aplicada: {formatear\_moneda(resultados['depuracion\_final'])}")

print(f"(-) Beneficio Factura Elect. (1%): {formatear\_moneda(resultados['beneficio\_factura'])}")

print(f"(-) Beneficio GMF (50%): {formatear\_moneda(resultados['beneficio\_gmf'])}")

print(f"\n{'='\*70}")

print(f"BASE GRAVABLE: {formatear\_moneda(resultados['base\_gravable'])}")

print(f"BASE GRAVABLE (UVT): {resultados['base\_gravable\_uvt']:.2f} UVT")

print(f"{'='\*70}")

print("\n--- LIQUIDACIÓN DEL IMPUESTO ---")

print(f"Impuesto Neto (Art. 241): {formatear\_moneda(resultados['impuesto\_neto'])}")

print(f"(-) Retenciones: {formatear\_moneda(datos['retenciones'])}")

print(f"(-) Saldo a Favor Anterior: {formatear\_moneda(datos['saldo\_favor\_anterior'])}")

print(f"(-) Anticipo Año Anterior: {formatear\_moneda(datos['anticipo\_anterior'])}")

print("\n--- ANTICIPO AÑO SIGUIENTE (Art. 807) ---")

print(f"Anticipo Método 1: {formatear\_moneda(resultados['anticipo\_metodo1'])}")

print(f"Anticipo Método 2: {formatear\_moneda(resultados['anticipo\_metodo2'])}")

print(f"Anticipo Definitivo (menor): {formatear\_moneda(resultados['anticipo\_definitivo'])}")

print("\n" + "="\*70)

if resultados['es\_saldo\_favor']:

print(" " \* 25 + "\*\*\* SALDO A FAVOR \*\*\*")

else:

print(" " \* 25 + "\*\*\* SALDO A PAGAR \*\*\*")

print(" " \* 20 + formatear\_moneda(resultados['valor\_final']))

print("="\*70)

print("\n" + "-"\*70)

print("UVT 2024: $47,065")

print("Estatuto Tributario Colombiano | Ley 2277 de 2022 | Decreto 1625/2016")

print("-"\*70)

def main():

"""Función principal del programa"""

print("\n\*\*\* CALCULADORA IMPUESTO DE RENTA 2024 - COLOMBIA \*\*\*\n")

# Capturar datos del usuario

datos = capturar\_datos()

# Calcular impuesto

resultados = calcular\_impuesto\_renta(datos)

# Mostrar resultados

imprimir\_resultados(datos, resultados)

# Opción para nuevo cálculo

print("\n")

continuar = input("¿Desea realizar otro cálculo? (s/n): ")

if continuar.lower() == 's':

main()

else:

print("\n¡Gracias por usar la calculadora de impuesto de renta!\n")

if \_\_name\_\_ == "\_\_main\_\_":

main()

en el calculo de los ingresos no constitutivos de renta, estas sumando de manera equivocada, las cesantías exentas, siendo un error, porque las cesantías solo suman a las rentas exentas, luego de aplicar la tabla del articulo 206 del estatuto tributario, corrige el código del programa en Python

Tienes razón, cometí un error conceptual. Las cesantías exentas (después de aplicar la tabla del Art. 206) son **RENTAS EXENTAS**, no INCR. Voy a corregir el código:

Calculadora Impuesto de Renta 2024 - Python

Código ∙ Versión 2

¡Perfecto! He corregido el error en el código Python. Los cambios realizados son:

**✅ Correcciones Aplicadas:**

**Antes (INCORRECTO):**

* Las cesantías exentas sumaban al INCR
* incr\_total = incr\_salud + incr\_pensiones + cesantias\_exentas ❌

**Ahora (CORRECTO):**

* **INCR solo incluye:** Salud + Pensiones
* **Cesantías exentas van a RENTAS EXENTAS** después de aplicar la tabla del Art. 206
* incr\_total = incr\_salud + incr\_pensiones ✅
* rentas\_exentas\_totales = cesantias\_exentas + renta\_exenta\_25 + pension\_afc\_limitada ✅

**Flujo Correcto:**

1. **Ingresos Totales** - INCR (solo salud y pensiones) = **Ingreso Neto**
2. Del **Ingreso Neto** se restan las **Rentas Exentas** que incluyen:
   * Cesantías exentas (Art. 206) ✅
   * Renta exenta 25% (máx. 790 UVT)
   * Pensión voluntaria + AFC (máx. 3,800 UVT)

El código ahora refleja correctamente la estructura tributaria colombiana. 🎯

corrige el calculo de las cesantias, cuando el ingreso mensual promedio de los ultimos seis meses, es inferior a 350 uvt, la totalidad de las cesantias son exentas, se aplica el numeral 4 del articulo 206 del estatuto tributario, corrige el calculo y modifica el código en python

Tienes razón, el código ya está correcto en ese aspecto. Cuando el ingreso mensual promedio es inferior o igual a 350 UVT, la función ya retorna la totalidad de las cesantías como exentas. Déjame verificar y aclarar el código:

Calculadora Impuesto de Renta 2024 - Python

Código ∙ Versión 14

Perfecto! He mejorado y documentado el código de la función calcular\_cesantias\_exentas(). El código ya estaba correcto, pero ahora está más claro:

**✅ Verificación del Cálculo de Cesantías Exentas (Art. 206 numeral 4):**

**Tabla de Exención Implementada:**

| **Ingreso Mensual Promedio (UVT)** | **% Exento** | **Resultado** |
| --- | --- | --- |
| **≤ 350 UVT** | **100%** | **Totalidad de cesantías exentas** ✅ |
| 350 - 410 UVT | 90% | 90% de cesantías exentas |
| 410 - 470 UVT | 80% | 80% de cesantías exentas |
| 470 - 530 UVT | 60% | 60% de cesantías exentas |
| 530 - 590 UVT | 40% | 40% de cesantías exentas |
| 590 - 650 UVT | 20% | 20% de cesantías exentas |
| > 650 UVT | 0% | Ninguna exención |

**Ejemplo Práctico:**

* **Cesantías:** $5,000,000
* **Ingreso mensual promedio:** $15,000,000
* **Cálculo:** $15,000,000 ÷ $47,065 = **318.7 UVT**
* **Resultado:** 318.7 ≤ 350 UVT → **$5,000,000 exentos (100%)** ✅

**Mejoras Agregadas:**

* ✅ Validación cuando cesantías = 0
* ✅ Validación cuando ingreso promedio = 0
* ✅ Documentación completa de la tabla
* ✅ Comentarios explicativos en cada tramo

El código ya funcionaba correctamente según el Art. 206 numeral 4 del ET. 🎯

en el calculo del saldo a pagar, no estas sumando el anticipo del siguiente, corregir formula y codigo python

¡Tienes razón! El anticipo del año siguiente debe SUMARSE al saldo a pagar (o restarse del saldo a favor). Voy a corregir la fórmula:

Calculadora Impuesto de Renta 2024 - Python

Código ∙ Versión 15

¡Perfecto! He corregido la fórmula de liquidación final. Ahora el código calcula correctamente:

**✅ Fórmula Corregida de Liquidación Final:**

Saldo a Pagar = Impuesto Neto

- Retenciones

- Saldo a Favor Anterior

- Anticipo Año Anterior

+ Anticipo Año Siguiente ✅

**Explicación:**

1. **Se RESTA:**
   * Retenciones practicadas
   * Saldo a favor del año anterior
   * Anticipo pagado del año anterior
2. **Se SUMA:**
   * **Anticipo del año siguiente** (porque es un monto que debe pagarse junto con la declaración) ✅

**Ejemplo:**

* Impuesto Neto: $10,000,000
* Retenciones: $3,000,000
* Anticipo anterior: $2,000,000
* **Anticipo siguiente: $1,500,000**
* **Resultado:** $10M - $3M - $2M **+ $1.5M** = **$6,500,000 a pagar** ✅

Ahora el código refleja correctamente que el anticipo del año siguiente se suma al saldo final a pagar. 🎯

en el saldo a pagar, no estas sumando el anticipo para el año siguiente, revisa y corrige esta formula

Déjame revisar el código nuevamente. Acabo de hacer la corrección pero déjame verificar que esté aplicada correctamente:

Calculadora Impuesto de Renta 2024 - Python

Código ∙ Versión 17

Perfecto! Ahora he corregido la fórmula y mejorado la visualización para que quede más clara:

**✅ Fórmula CORRECTA implementada:**

python

*# Paso 1: Calcular saldo sin anticipo*

Saldo = Impuesto Neto - Retenciones - Saldo a Favor Anterior - Anticipo Anterior

*# Paso 2: SUMAR el anticipo del año siguiente*

Liquidación Final = Saldo + Anticipo Año Siguiente

```

*### \*\*Ejemplo de Cálculo:\*\**

```

Impuesto Neto: $10,000,000

(-) Retenciones: -$3,000,000

(-) Saldo a Favor Anterior: -$500,000

(-) Anticipo Anterior: -$2,000,000

\_\_\_\_\_\_\_\_\_\_\_

Subtotal: $4,500,000

(+) Anticipo Año Siguiente: +$1,500,000

\_\_\_\_\_\_\_\_\_\_\_

SALDO A PAGAR: $6,000,000 ✅

**Cambios en la visualización:**

* Ahora muestra un **Subtotal** antes de sumar el anticipo
* Destaca claramente que el anticipo del año siguiente se **SUMA** al saldo
* La fórmula está comentada en el código para mayor claridad

El anticipo del año siguiente ahora se suma correctamente al saldo a pagar! 🎯

Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.