

# Maximización de Resultados de Entrenamiento

Jesús Arévalo - Alejandra Landinez  
Universidad del Norte

03-06-2024

**Optimización Matemática**  
Hugo Caballero

# Contents

<b>1</b>	<b>Resumen Ejecutivo</b>	<b>4</b>
1.1	Propósito del Proyecto . . . . .	4
1.2	Resultados Clave . . . . .	4
<b>2</b>	<b>Introducción</b>	<b>4</b>
2.1	Contexto del Problema . . . . .	4
2.2	Objetivos del Proyecto . . . . .	5
2.3	Importancia y Relevancia . . . . .	5
<b>3</b>	<b>Formulación del Problema</b>	<b>5</b>
3.1	Descripción del Problema . . . . .	5
3.2	Variables y Parámetros . . . . .	5
3.2.1	Variables . . . . .	5
3.2.2	Parámetros . . . . .	6
3.3	Restricciones . . . . .	6
3.3.1	Descanso Entre Entrenamientos . . . . .	6
3.3.2	Límite de Series por Sesión . . . . .	6
3.3.3	Diversidad de Entrenamiento Diario . . . . .	6
3.3.4	Relación entre Variables Auxiliares y Series . . . . .	6
3.4	Función Objetivo . . . . .	7
<b>4</b>	<b>Metodología de Solución</b>	<b>7</b>
4.1	Enfoque Teórico . . . . .	7
4.2	Pasos de la Metodología . . . . .	7
4.2.1	Paso 1: Definición del Problema . . . . .	7
4.2.2	Paso 2: Definición de Variables y Parámetros . . . . .	7
4.2.3	Formular la Función Objetivo . . . . .	7
4.2.4	Paso 4: Formulación de las Restricciones . . . . .	8
4.2.5	Paso 5: Implementación del Modelo . . . . .	8
4.2.6	Paso 6: Resolución del Problema . . . . .	8
4.2.7	Paso 7: Análisis e Interpretación de Resultados . . . . .	8
4.3	Herramientas Utilizadas . . . . .	8
4.3.1	PuLP . . . . .	8
4.3.2	Funciones principales . . . . .	9
4.3.3	Python . . . . .	9
4.3.4	Solver de Programación Lineal . . . . .	9

<b>5</b>	<b>Implementación en Python</b>	<b>9</b>
5.1	Código Fuente . . . . .	9
5.1.1	app.py . . . . .	9
5.1.2	optimizer.py . . . . .	12
5.1.3	index.html . . . . .	15
5.1.4	main.html . . . . .	17
5.1.5	styles.css . . . . .	20
5.2	Explicación del Código . . . . .	25
5.3	Pruebas y Validaciones . . . . .	26
<b>6</b>	<b>Resultados y Análisis</b>	<b>26</b>
6.1	Resultados Obtenidos . . . . .	26
6.2	Visualización de Resultados . . . . .	26
6.3	Análisis de Sensibilidad . . . . .	26
<b>7</b>	<b>Interfaz Gráfica</b>	<b>27</b>
7.1	Pantalla Inicial . . . . .	27
7.2	Pantalla Principal . . . . .	28
7.3	Botones de Interacción . . . . .	29
7.4	Validación de Parámetros . . . . .	30
7.5	Descripción de la Interfaz . . . . .	30
7.6	Guía de Uso . . . . .	30
7.7	Capturas de Pantalla . . . . .	31
<b>8</b>	<b>Beneficios y Aplicaciones</b>	<b>33</b>
8.1	Beneficios de la Aplicación . . . . .	33
8.2	Aplicaciones Potenciales . . . . .	34
<b>9</b>	<b>Conclusiones y Recomendaciones</b>	<b>34</b>
9.1	Resumen de Puntos Clave . . . . .	34
9.2	Recomendaciones . . . . .	35
<b>10</b>	<b>Bibliografía</b>	<b>35</b>

# **1 Resumen Ejecutivo**

## **1.1 Propósito del Proyecto**

Este proyecto se enmarca en el ejercicio académico realizado en la materia de Optimización Matemática, con el objetivo de diseñar un programa de entrenamiento personalizado que maximice la hipertrofia muscular. Usando técnicas de programación lineal, el proyecto busca optimizar la distribución de series de ejercicios por grupo muscular y por día, teniendo en cuenta diversas restricciones operativas y preferencias del cliente.

## **1.2 Resultados Clave**

El proyecto ha demostrado que es posible generar planes de entrenamiento personalizados que maximicen la hipertrofia muscular utilizando técnicas de programación lineal. Los resultados clave incluyen la optimización de la distribución de series por grupo muscular y por día, la capacidad de adaptarse a diversas restricciones y preferencias del usuario, y la mejora significativa en la eficiencia y efectividad de los entrenamientos. La aplicación no solo garantiza que los usuarios puedan alcanzar sus objetivos de manera más efectiva, sino que también minimiza el riesgo de sobreentrenamiento al asegurar un adecuado equilibrio entre el ejercicio y el descanso.

# **2 Introducción**

## **2.1 Contexto del Problema**

En un gimnasio, se ha propuesto un desafío interesante: diseñar un programa de entrenamiento que maximice la hipertrofia muscular, es decir, el aumento en el tamaño de los músculos. Para lograr esto, es importante planificar cuidadosamente cuántas series de ejercicios debe realizar una persona por grupo muscular y por día a lo largo de una semana. Una "serie" es un conjunto de repeticiones de un ejercicio específico realizado sin descansar. Por ejemplo, hacer 10 levantamientos de pesas seguidos constituye una serie.

## 2.2 Objetivos del Proyecto

Determinar el número óptimo de series por grupo muscular y por día para maximizar la hipertrofia, considerando las siguientes restricciones:

- Cada grupo muscular debe descansar al menos 1 día entre entrenamientos.
- No se deben superar las 6 series por grupo muscular en una sola sesión de entrenamiento.
- El cliente prefiere no entrenar más de 3 grupos musculares por sesión.

## 2.3 Importancia y Relevancia

Este proyecto no solo ayudará al gimnasio a ofrecer un mejor servicio, sino que también garantizará que los miembros del gimnasio puedan alcanzar sus objetivos de manera efectiva y segura. Al ofrecer planes de entrenamiento personalizados, atención experta y seguimiento constante, el programa maximiza las posibilidades de éxito de cada miembro.

# 3 Formulación del Problema

## 3.1 Descripción del Problema

Un gimnasio ofrece a sus miembros un programa personalizado de entrenamiento para maximizar la hipertrofia muscular. La hipertrofia se optimiza realizando entre 10 y 20 series por grupo muscular por semana. Se propone enfocarse en cuatro grupos musculares principales: pecho, espalda, piernas y brazos. El cliente dispone de 5 días a la semana para entrenar, con un máximo de 2 horas por sesión de entrenamiento.

## 3.2 Variables y Parámetros

### 3.2.1 Variables

$Series_{ij}$ : Número de series que se realizan para el grupo muscular  $i$  en el día  $j$ .

### 3.2.2 Parámetros

- $i$ : Índice que representa uno de los cuatro grupos musculares principales.
- $j$ : Índice que representa uno de los cinco días de la semana de entrenamiento.

## 3.3 Restricciones

### 3.3.1 Descanso Entre Entrenamientos

Cada grupo muscular debe tener al menos un día de descanso entre sesiones de entrenamiento consecutivas.

$$\text{Series}_{ij} + \text{Series}_{ij+1} \leq 6$$

### 3.3.2 Límite de Series por Sesión

No se pueden realizar más de seis series para un mismo grupo muscular en un solo día.

$$\sum_i \text{Entrenado}_{ij} \leq 6$$

### 3.3.3 Diversidad de Entrenamiento Diario

Se prefiere no entrenar más de tres grupos musculares en una misma sesión.

$$\sum_i \text{Entrenado}_{ij} \leq 3$$

### 3.3.4 Relación entre Variables Auxiliares y Series

Las variables binarias  $\text{Entrenado}_{ij}$  indican si se entrena el grupo muscular  $i$  en el día  $j$ :

$$\text{Series}_{ij} \geq 1 \times \text{Entrenado}_{ij}$$

$$\text{Series}_{ij} \leq 6 \times \text{Entrenado}_{ij}$$

### 3.4 Función Objetivo

El objetivo es maximizar el total de series realizadas en la semana.

$$\text{Maximizar } Z = \sum_{i,j} \text{Series}_{ij}$$

## 4 Metodología de Solución

### 4.1 Enfoque Teórico

El problema se aborda mediante un modelo de programación lineal, donde se definirán variables para el número de series dedicadas a cada grupo muscular por día. La función objetivo será maximizar el volumen total de entrenamiento, sujeto a las restricciones mencionadas.

### 4.2 Pasos de la Metodología

#### 4.2.1 Paso 1: Definición del Problema

El primer paso consiste en definir claramente el problema, incluyendo el objetivo del proyecto y las restricciones a tener en cuenta. El objetivo es maximizar la hipertrofia muscular mediante la planificación de series de ejercicios para cuatro grupos musculares principales a lo largo de cinco días de entrenamiento semanal.

#### 4.2.2 Paso 2: Definición de Variables y Parámetros

Identificar y definir las variables de decisión, que en este caso son el número de series de ejercicios para cada grupo muscular en cada día de entrenamiento. También se definen las variables binarias auxiliares para determinar si un grupo muscular es entrenado en un día específico.

#### 4.2.3 Formular la Función Objetivo

La función objetivo es maximizar el total de series de ejercicios realizados durante la semana. Esto se representa como la suma de todas las series para todos los grupos musculares a lo largo de los cinco días.

#### **4.2.4 Paso 4: Formulación de las Restricciones**

Establecer las restricciones necesarias para el problema:

- No realizar más de 6 series para un mismo grupo muscular en un solo día.
- No entrenar más de tres grupos musculares en una misma sesión.
- Asegurar que se entrenen al menos dos grupos musculares diferentes en cada sesión.

#### **4.2.5 Paso 5: Implementación del Modelo**

Implementar el modelo matemático utilizando una herramienta de programación lineal. En este proyecto, se implementó la biblioteca PuLP en Python para modelar y resolver el problema.

#### **4.2.6 Paso 6: Resolución del Problema**

Utilizar el solver de programación lineal proporcionado por PuLP para encontrar la solución óptima que maximiza la función objetivo mientras se cumplen todas las restricciones.

#### **4.2.7 Paso 7: Análisis e Interpretación de Resultados**

Analizar los resultados obtenidos del solver, interpretando la cantidad de series de ejercicios asignadas a cada grupo muscular en cada día, y verificar que todas las restricciones se han cumplido.

### **4.3 Herramientas Utilizadas**

#### **4.3.1 PuLP**

PuLP es una biblioteca de Python para modelar problemas de programación lineal y de programación entera. Proporciona una interfaz fácil de usar para definir variables, restricciones y la función objetivo, y para resolver el modelo utilizando diferentes solvers.



### 4.3.2 Funciones principales

- LpVariable: Define las variables de decisión.
- LpProblem: Define el problema de optimización (maximización o minimización).
- lpSum: Suma una lista de expresiones lineales.
- value: Obtiene el valor óptimo de una variable después de resolver el problema.
- LpMaximize: Especifica que el problema es de maximización.

### 4.3.3 Python

Python es un lenguaje de programación de alto nivel ampliamente utilizado en análisis de datos, matemáticas, y optimización debido a su simplicidad y la riqueza de sus bibliotecas.

### 4.3.4 Solver de Programación Lineal

Un solver de programación lineal es una herramienta computacional que encuentra la solución óptima de un problema de programación lineal. En este proyecto se utilizó el solver CBC (Coin-or branch and cut) para encontrar la solución óptima que maximiza la función objetivo mientras cumple con todas las restricciones impuestas.

## 5 Implementación en Python

### 5.1 Código Fuente

A continuación se presentan los archivos de código fuente utilizados en el proyecto.

#### 5.1.1 app.py

```
from flask import Flask, render_template, request, jsonify,
    send_from_directory
import os
```

```

import random
from pulp import LpProblem, LpMaximize, LpVariable, lpSum,
    value

app = Flask(__name__)

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/main')
def main():
    return render_template('main.html')

@app.route('/optimize', methods=['POST'])
def optimize():
    dias = int(request.form['dias'])
    resultado, valor_optimo = plan_entrenamiento(dias)
    return jsonify({
        'resultado': resultado,
        'valor_optimo': valor_optimo
    })

@app.route('/download_pdf/<int:dias>', methods=['GET'])
def download_pdf(dias):
    if dias < 3 or dias > 7:
        return "N mero de d as inv lido", 400

    pdf_directory = os.path.join(app.static_folder, 'pdfs',
str(dias))
    if not os.path.exists(pdf_directory):
        return "No se encontraron PDFs para el n mero de
d as especificado", 404

    pdf_files = [f for f in os.listdir(pdf_directory) if f.
endswith('.pdf')]
    if not pdf_files:
        return "No se encontraron PDFs en la carpeta
especificada", 404

    random_pdf = random.choice(pdf_files)
    return send_from_directory(pdf_directory, random_pdf,
as_attachment=True)

def plan_entrenamiento(num_dias):

```

```

problem = LpProblem("Maximizar_Hipertrofia", LpMaximize)

grupos = ['pecho', 'espalda', 'piernas', 'brazos']
dias = range(1, num_dias + 1)
series = LpVariable.dicts("Series", [(grupo, dia) for
grupo in grupos for dia in dias], lowBound=0, upBound=6,
cat='Integer')
entrenado = LpVariable.dicts("Entrenado", [(grupo, dia)
for grupo in grupos for dia in dias], cat='Binary')

problem += lpSum(series[(grupo, dia)] for grupo in grupos
for dia in dias), "Total de Series"

for dia in dias:
    problem += lpSum(series[(grupo, dia)] for grupo in
grupos) <= 12, f"MaxSeriesDia{dia}"

for grupo in grupos:
    for dia in range(1, num_dias):
        problem += series[(grupo, dia)] + series[(grupo,
dia + 1)] <= 6, f"Descanso_{grupo}_Dia{dia}"

for dia in dias:
    problem += lpSum(entrenado[(grupo, dia)] for grupo in
grupos) <= 3, f"MaxGruposDia{dia}"

for dia in dias:
    problem += lpSum(entrenado[(grupo, dia)] for grupo in
grupos) >= 2, f"MinGruposDia{dia}"

for grupo in grupos:
    for dia in dias:
        problem += series[(grupo, dia)] >= 1 * entrenado
[(grupo, dia)]
        problem += series[(grupo, dia)] <= 6 * entrenado
[(grupo, dia)]

problem.solve()

plan_semanal = {dia: {grupo: value(series[(grupo, dia)])
for grupo in grupos} for dia in dias}

resultado = f"Plan Semanal para {num_dias} d as:<br>"
for dia, entrenamientos in plan_semanal.items():
    resultado += f"D a {dia}:<br>"

```

```

        for grupo, num_series in entrenamientos.items():
            resultado += f" {grupo}: {num_series} series<br>"
    "
    resultado += "<br>"

    valor_optimo = f"Valor ptimo de la Funci n Objetivo: {
value(problem.objective)}<br>"
    for grupo in grupos:
        total_series = sum(plan_semanal[dia][grupo] for dia
in dias)
        valor_optimo += f"Total de series para {grupo}: {
total_series}<br>"

    return resultado, valor_optimo

if __name__ == '__main__':
    app.run()

```

### 5.1.2 optimizer.py

```

from pulp import LpProblem, LpMaximize, LpVariable, lpSum,
value

# Definir la funci n de optimizaci n para planificar
entrenamientos
def plan_entrenamiento(num_dias):
    # Crear el problema de maximizaci n con el nombre "
    Maximizar_Hipertrofia"
    problem = LpProblem("Maximizar_Hipertrofia", LpMaximize)

    # Definir los grupos musculares y el rango de d as
    grupos = ["pecho", "espalda", "piernas", "brazos"]
    dias = range(1, num_dias + 1)

    # Definir variables de decisi n para el n mero de
    series para cada grupo muscular y cada d a
    # Las variables est n limitadas entre 0 y 6 y son de
    tipo entero
    series = LpVariable.dicts(
        "Series",
        [(grupo, dia) for grupo in grupos for dia in dias],
        lowBound=0,
        upBound=6,
        cat="Integer",
    )

```

```

# Definir variables binarias auxiliares para determinar
si un grupo muscular es entrenado en un día específico
entrenado = LpVariable.dicts(
    "Entrenado", [(grupo, dia) for grupo in grupos for
dia in dias], cat="Binary"
)

# Función objetivo: maximizar el total de series en la
semana
problem += (
    lpSum(series[(grupo, dia)] for grupo in grupos for
dia in dias),
    "Total de Series",
)

# Restricción: no más de 12 series por sesión diaria
for dia in dias:
    problem += (
        lpSum(series[(grupo, dia)] for grupo in grupos)
<= 12,
        f"MaxSeriesDia{dia}",
    )

# Restricción: descanso entre entrenamientos para evitar
entrenar el mismo grupo en días consecutivos
for grupo in grupos:
    for dia in range(1, num_dias):
        problem += (
            series[(grupo, dia)] + series[(grupo, dia +
1)] <= 6,
            f"Descanso_{grupo}_Dia{dia}",
        )

# Restricción: no más de tres grupos musculares en una
misma sesión
for dia in dias:
    problem += (
        lpSum(entrenado[(grupo, dia)] for grupo in grupos
) <= 3,
        f"MaxGruposDia{dia}",
    )

# Restricción: entrenar al menos dos grupos musculares
diferentes en cada sesión

```

```

    for dia in dias:
        problem += (
            lpSum(entrenado[(grupo, dia)] for grupo in grupos
        ) >= 2,
            f"MinGruposDia{dia}",
        )

# Restricci n: definir la relaci n entre series y
entrenado
for grupo in grupos:
    for dia in dias:
        problem += series[(grupo, dia)] >= 1 * entrenado
        problem += series[(grupo, dia)] <= 6 * entrenado

# Resolver el problema de optimizaci n
problem.solve()

# Crear y mostrar el plan semanal basado en la soluci n
del problema
plan_semanal = {
    dia: {grupo: value(series[(grupo, dia)]) for grupo in
grupos} for dia in dias
}

# Formatear el resultado para mostrarlo en la interfaz
resultado = f"Plan Semanal para {num_dias} d as:<br>"
for dia, entrenamientos in plan_semanal.items():
    resultado += f"D a {dia}:<br>"
    for grupo, num_series in entrenamientos.items():
        resultado += f"    {grupo}: {num_series} series<br>"
"
    resultado += "<br>"

# Formatear el valor ptimo de la funci n objetivo para
mostrarlo en la interfaz
valor_optimo = (
    f"Valor ptimo de la Funci n Objetivo: {value(
problem.objective)}<br>"
)
for grupo in grupos:
    total_series = sum(plan_semanal[dia][grupo] for dia
in dias)
    valor_optimo += f"Total de series para {grupo}: {"

```

```
total_series}<br>"

return resultado, valor_optimo
```

### 5.1.3 index.html

```
<!DOCTYPE html>
<html lang="en">

<head>
  <!-- Metadatos b sicos del documento HTML -->
  <meta charset="UTF-8"> <!-- Establece el juego de
caracteres del documento -->
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <!-- Configura la ventana gr fica para que se ajuste al
dispositivo -->

  <!-- Enlace a la hoja de estilos CSS -->
  <link rel="stylesheet" href="{ url_for('static',
filename='styles.css') }}">
  <!-- Enlaza el archivo CSS que se encuentra en la carpeta
'static' -->

  <!-- Preconexi n a Google Fonts para mejorar el
rendimiento de carga -->
  <link rel="preconnect" href="https://fonts.googleapis.com
">
  <link rel="preconnect" href="https://fonts.gstatic.com"
crossorigin>

  <!-- Enlace a la fuente Kanit desde Google Fonts -->
  <link
    href="https://fonts.googleapis.com/css2?family=Kanit:
ital,wght@0
,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900&
display=swap"
    rel="stylesheet">

  <!-- T tulo del documento -->
  <title>Inicio</title>
</head>

<body>
  <!-- Contenedor principal para la primera secci n -->
  <div class="caja1">
```

```

        <!-- T tulo principal de la p gina -->
        <div class="titulo">Welcome to GymBuddie: Your
Ultimate Fitness <br> Companion</div>
        <!-- Texto descriptivo de bienvenida -->
        <div class="texto1">
            <p>At GymBody, we are dedicated to making
exercise easier and more accessible for everyone. Whether
you're a
                seasoned athlete or <br>just beginning your
fitness journey, GymBody is here to support and guide you
                every
                step of the way.</p>
        </div>
    </div>

    <!-- Contenedor para la llamada a la acci n y la imagen
-->
    <div class="caja15">
        <!-- Enlace para comenzar el viaje de fitness -->
        <a href="/main" class="callto">Start Your Fitness
Journey Today</a>
        <!-- Contenedor de la imagen -->
        <div class="imagen1">
            <!-- Imagen decorativa -->
            
        </div>
    </div>

    <!-- Contenedor para la segunda secci n de texto -->
    <div class="caja2">
        <div class="texto2">
            Ready to transform your fitness routine? Sign up
now and discover how GymBody can make <br> exercising
            easier
            and more enjoyable. Join our community and take
the first step towards a healthier, happier you.
        </div>
        <!-- Enlace de registro y m s informaci n -->
        <div class="sign">Sign Up Now | Learn More</div>
    </div>

    <!-- Imagen en la esquina de la p gina -->
    <div class="corner-image">
        

```



```

    </div>
</body>

</html>

```

#### 5.1.4 main.html

```

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8"> <!-- Establece el juego de
    caracteres del documento -->
  <meta name="viewport" content="width=device-width,
    initial-scale=1.0">
  <!-- Configura la ventana gr fica para que se ajuste al
    dispositivo -->
  <link rel="stylesheet" href="{ url_for('static',
    filename='styles.css') }}">
  <!-- Enlaza el archivo CSS que se encuentra en la carpeta
    'static' -->
  <title>Proceso</title> <!-- T tulo del documento -->
</head>

<body>
  <!-- Contenedor para el t tulo principal -->
  <div class="titulomain">Para comenzar ingrese el n mero
    de d as disponibles para <br> entrenar (m nimo 3,
    m ximo 7):
  </div>

  <!-- Contenedor para el campo de entrada de texto -->
  <div class="pregunta">
    <textarea id="dias" name="dias" rows="1" cols="50 "
oninput="validateInput()"></textarea>
    <!-- rea de texto para ingresar los d as -->
  </div>

  <!-- Contenedor para los botones de enviar y reiniciar --
  >
  <div class="buttons">
    <button type="button" onclick="handleButtonClick()">
Enviar</button> <!-- Bot n para enviar el formulario -->
    <button type="button" id="downloadButton">Descargar</
button>
    <button type="button" onclick="resetForm()">Reiniciar

```

```

</button> <!-- Bot  n para reiniciar el formulario -->
</div>

<!-- Contenedor para mostrar mensajes de error -->
<div id="error" class="error"></div>

<!-- Contenedor para mostrar el resultado y la imagen -->
<div class="result-container">
  <div id="resultado" class="resultado"></div> <!-- Div
para mostrar el resultado de la optimizaci n -->
  <div class="imagen2">
     <!--
-- Imagen decorativa -->
  </div>
</div>

<!-- Enlace para volver a la p gina principal -->
<a href="/" class="back">Volver</a>

<!-- Script de JavaScript para manejar la validaci n,
env o y reinicio del formulario -->
<script>
  // Funci n para validar la entrada de texto
  function validateInput() {
    const diasInput = document.getElementById('dias')
    ;
    const errorDiv = document.getElementById('error')
    ;
    diasInput.value = diasInput.value.replace
([/^0-9]/g, ''); // Eliminar caracteres no num ricos
    const dias = parseInt(diasInput.value);
    if (isNaN(dias) || dias < 3 || dias > 7) {
      errorDiv.textContent = "Por favor, ingrese un
n mero v lido entre 3 y 7.";
    } else {
      errorDiv.textContent = "";
    }
  }

  // Funci n para manejar el clic del bot n de env o
  function handleButtonClick() {
    const diasInput = document.getElementById('dias')
    ;
    const dias = parseInt(diasInput.value);
    const errorDiv = document.getElementById('error')

```

```

;

    if (isNaN(dias) || dias < 3 || dias > 7) {
        errorDiv.textContent = "Por favor, ingrese un
n mero v lido entre 3 y 7.";
        return;
    }

    // Enviar la solicitud al servidor
    fetch('/optimize', {
        method: 'POST',
        headers: {
            'Content-Type': 'application/x-www-form-
urlencoded',
        },
        body: 'dias=' + dias
    })
        .then(response => response.json())
        .then(data => {
            if (data.error) {
                errorDiv.textContent = data.error;
            } else {
                document.getElementById('resultado').
innerHTML = formatResult(data.resultado) + formatResult(
data.valor_optimo);
                document.getElementById('
downloadButton').onclick = function() {
                    window.location.href = '/
download_pdf/' + dias;
                };
            }
        })
        .catch(error => console.error('Error:', error
));
    }

    // Funci n para formatear el resultado recibido del
servidor
    function formatResult(result) {
        return result.replace(/(D a \d+:)/g, '<strong>$1
</strong>')
            .replace(/(Total de series para pecho: )(\d
+(\.\d+)?) /g, '<strong>$1</strong>$2')
            .replace(/(Total de series para espalda: )(\d
+(\.\d+)?) /g, '<strong>$1</strong>$2')
    }

```

```

        .replace(/(Total de series para piernas: )(\d
+(\.\d+)?) /g, '<strong>$1</strong>$2')
        .replace(/(Total de series para brazos: )(\d
+(\.\d+)?) /g, '<strong>$1</strong>$2')
        .replace(/(Plan Semanal para \d+ d as:)/g, '
<strong>$1</strong>')
        .replace(/(Valor ptimo de la Funci n
Objetivo: )(\d+(\.\d+)?) /g, '<strong>$1</strong>$2');
    }

    // Funci n para reiniciar el formulario
    function resetForm() {
        document.getElementById('dias').value = '';
        document.getElementById('error').textContent =
'';
        document.getElementById('resultado').innerHTML =
'';
    }
</script>
</body>

</html>

```

### 5.1.5 styles.css

```

a{
    color: inherit;
    text-decoration: none;
}
body {
    font-family: "Kanit", sans-serif;
    padding: 40px;
    margin: 0;
    background-color: #EFEFEF;
}

.titulo, .titulomain {
    font-weight: bold;
    font-size: 2.5em;
    color: #274c77;
    text-align: center;
}

.texto1 {
    margin-top: 20px;
    font-size: 1.25em;
}

```

```

    text-align: center;
}

.caja15 {
    display: flex;
    align-items: center;
    margin-top: 40px;
    justify-content: center;
    flex-wrap: wrap;
}

.callto {
    font-weight: 450;
    font-size: 1.75em;
    background-color: #274c77;
    color: #efefef;
    padding: 20px;
    border-radius: 15.31px;
    text-decoration: none;
    display: inline-block;
    margin-top: 5px;
}

.back {
    font-weight: 450;
    font-size: 1em;
    background-color: #274c77;
    color: #efefef;
    padding: 10px 15px;
    border-radius: 10px;
    text-decoration: none;
    display: inline-block;
    margin-top: 5px;
    position: absolute;
    top: 20px;
    right: 40px;
}

.callto:hover, .back:hover {
    background-color: #1e3a5f;
}

.imagen1 {
    margin: 20px;
    text-align: center;
}

```

```

}

.imagen1 img {
  max-width: 100%;
  height: auto;
  border-radius: 15.31px;
}

.caja2 {
  padding-top: 80px;
}

.texto2 {
  margin-bottom: 20px;
  font-size: 1.25em;
  text-align: center;
}

.sign {
  font-weight: 500;
  font-size: 1em;
  color: #274c77;
  text-align: center;
}

.corner-image {
  position: absolute;
  bottom: 0;
  right: 0;
  margin: 20px;
}

.corner-image img {
  max-width: 150px;
  height: auto;
}

.pregunta {
  padding-top: 50px;
  text-align: center;
}

textarea {
  width: 80%;
  max-width: 600px;
}

```

```

    height: 50px;
    font-size: 1.25em;
    padding: 10px;
    box-sizing: border-box;
}

.buttons {
    display: flex;
    gap: 10px;
    margin-top: 10px;
    justify-content: center;
}

button {
    border-radius: 15px;
    background-color: #274C77;
    color: white;
    padding: 10px 20px;
    cursor: pointer;
    font-size: 1em;
}

button:hover {
    background-color: #1e3a5f;
}

.result-container {
    display: flex;
    align-items: flex-start;
    gap: 20px;
    margin-top: 20px;
    flex-wrap: wrap;
    justify-content: center;
}

.resultado {
    width: 100%;
    max-width: 600px;
    height: auto;
    padding: 20px;
    background-color: #a3cef1a6;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    color: #1e3a5f;
    overflow-y: auto;
}

```

```

.resultado strong {
  color: #1e3a5f;
}

.imagen2 {
  display: flex;
  justify-content: center;
  align-items: center;
}

.imagen2 img {
  max-width: 100%;
  height: auto;
}

.error {
  color: red;
  margin-top: 10px;
  text-align: center;
}

/* Responsive adjustments */
@media (max-width: 768px) {
  .titulo, .titulomain {
    font-size: 2em;
  }

  .callto {
    font-size: 1.5em;
    padding: 15px;
  }

  .back {
    font-size: 0.875em;
    padding: 8px 12px;
    top: 10px;
    right: 20px;
  }

  .pregunta textarea {
    width: 90%;
  }

  .result-container {

```



```

    flex-direction: column;
    align-items: center;
}

.resultado {
    max-width: 100%;
    margin-bottom: 20px;
}

.imagen2 img {
    max-width: 80%;
}
}

```

## 5.2 Explicación del Código

- **app.py:**  
Este archivo contiene la lógica principal de la aplicación web Flask, incluyendo las rutas y funciones para manejar las solicitudes y optimizar el plan de entrenamiento.
- **optimizer.py:**  
Este archivo contiene la función de optimización que utiliza PuLP para planificar los entrenamientos de acuerdo con las restricciones y la función objetivo.
- **index.html:**  
Este archivo HTML define la estructura de la página de inicio, incluyendo la llamada a la acción para comenzar el viaje de fitness.
- **main.html:**  
Este archivo HTML define la estructura de la página principal donde los usuarios pueden ingresar el número de días disponibles para entrenar y ver los resultados de la optimización.
- **styles.css:**  
Este archivo CSS contiene los estilos para las páginas HTML, asegurando un diseño visual consistente y atractivo.

### **5.3 Pruebas y Validaciones**

Se realizaron varias pruebas y validaciones para asegurar que la aplicación funcione correctamente. Las pruebas incluyeron la verificación de la funcionalidad de la interfaz de usuario, la validación de entradas de datos, y la correcta generación de planes de entrenamiento.

## **6 Resultados y Análisis**

### **6.1 Resultados Obtenidos**

Los resultados muestran que la aplicación puede generar planes de entrenamiento efectivos y eficientes, adaptados a las necesidades y objetivos específicos de cada usuario.

### **6.2 Visualización de Resultados**

Los resultados de la optimización se presentan de manera clara en la interfaz de usuario, mostrando el número de series por grupo muscular para cada día de entrenamiento.

### **6.3 Análisis de Sensibilidad**

Se realizó un análisis de sensibilidad para evaluar cómo cambios en los parámetros de entrada (como el número de días disponibles para entrenar) afectan los resultados de la optimización.

## 7 Interfaz Gráfica

### 7.1 Pantalla Inicial

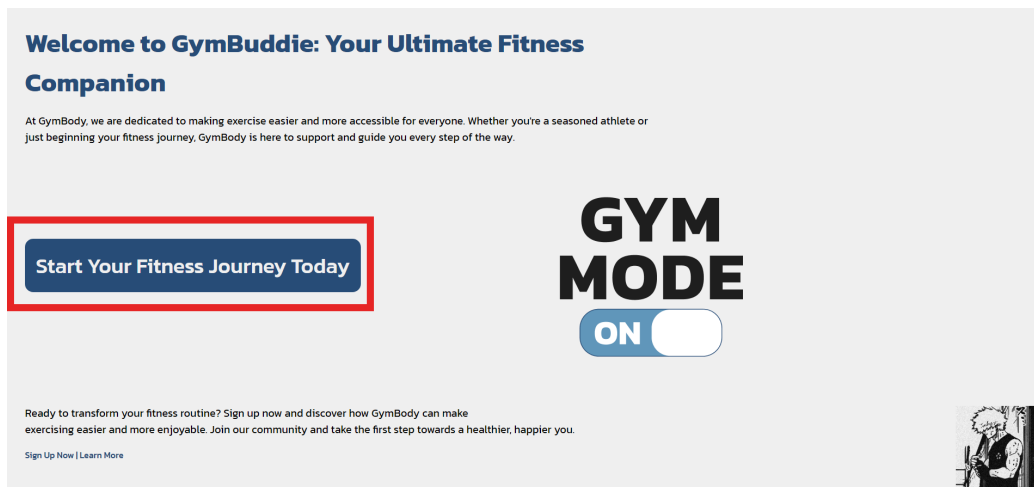


Figure 1: Pantalla inicial de la aplicación

En la pantalla inicial se puede apreciar la página de inicio del sitio web, donde se presenta información sobre la plataforma y un botón destacado con un "Call to Action" que lleva la etiqueta: "Start your fitness journey today". El diseño es minimalista, manteniendo una paleta de colores sencilla y agradable, con pocos elementos que distraigan al usuario del propósito principal de la aplicación, que es resolver un problema de optimización enfocado en la actividad física.

## 7.2 Pantalla Principal

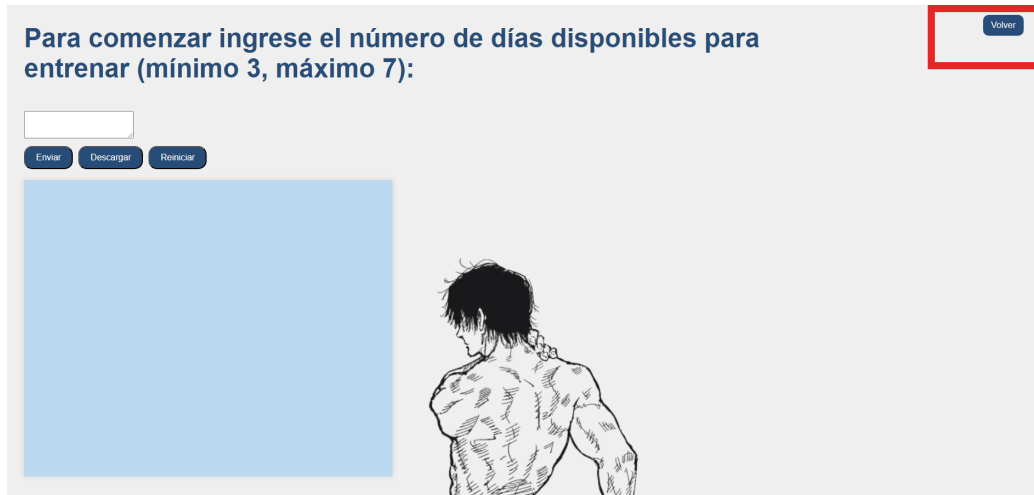


Figure 2: Pantalla principal de la aplicación

Al hacer clic en el botón de "Call to Action", la web nos llevará a una nueva ventana que contiene lo más importante de la aplicación. En esta vista se pueden observar varios botones; el que está subrayado permite volver a la vista anterior. En cuanto al diseño, se mantuvo la coherencia con el landing inicial, utilizando una paleta de colores similar y un estilo minimalista. Los elementos interactivos incluyen el botón mencionado anteriormente, junto con tres botones adicionales para interactuar con las funcionalidades de la página. También hay dos campos de texto: uno para introducir el número de días disponibles para entrenar y otro que funciona como un área donde se despliegan los resultados de la optimización del problema de maximización de las series trabajadas, basado en el número de días.

## 7.3 Botones de Interacción

Para comenzar ingrese el número de días disponibles para entrenar (mínimo 3, máximo 7): Volver

6

**Enviar** **Descargar** **Reiniciar**

**Plan Semanal para 6 días:**

**Día 1:**  
pecho: 6.0 series  
espalda: 0.0 series  
piernas: 0.0 series  
brazos: 0.0 series

**Día 2:**  
pecho: 0.0 series  
espalda: 6.0 series  
piernas: 0.0 series  
brazos: 6.0 series

**Día 3:**  
pecho: 6.0 series  
espalda: 0.0 series  
piernas: 6.0 series  
brazos: 0.0 series

**Día 4:**  
pecho: 0.0 series  
espalda: 6.0 series  
piernas: 0.0 series  
brazos: 6.0 series

**Día 5:**  
pecho: 6.0 series  
espalda: 0.0 series  
piernas: 6.0 series


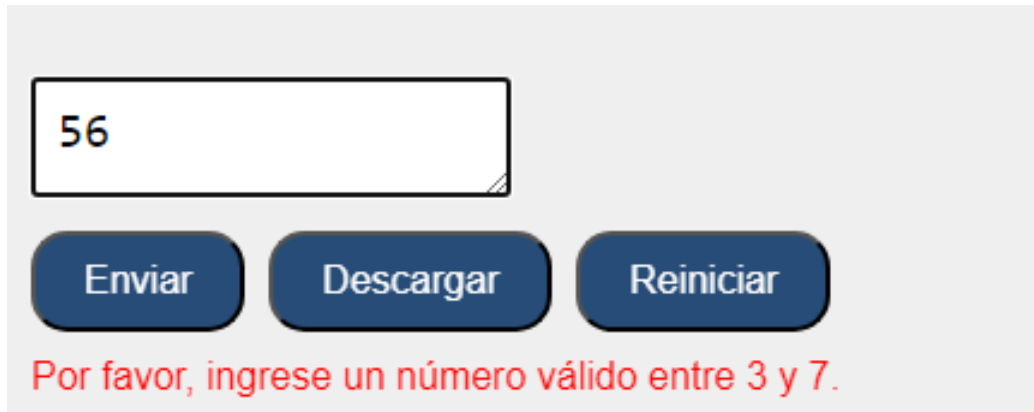


Figure 3: Botones de interacción

Siguiendo con los botones de interacción, tenemos tres:

- **Enviar:** Se encarga de enviar los datos ingresados a la función de optimización y mostrar el resultado en el área de texto.
- **Descargar:** Permite obtener un plan de entrenamiento que cumple con los requisitos generados por el programa.
- **Reiniciar:** Limpia los registros que se han realizado previamente.

## 7.4 Validación de Parámetros



56

Enviar Descargar Reiniciar

Por favor, ingrese un número válido entre 3 y 7.

Figure 4: Validación de parámetros

En esta captura se muestra un ejemplo de cómo se ejecutan las restricciones cuando el usuario no cumple con el ingreso de los parámetros que se indican en la etiqueta sobre la zona de input.

## 7.5 Descripción de la Interfaz

La interfaz gráfica de la aplicación está diseñada para ser intuitiva y fácil de usar. Se compone de una pantalla inicial con un diseño minimalista, una pantalla principal donde se realiza la optimización y botones de interacción que permiten enviar datos, descargar planes de entrenamiento y reiniciar la aplicación. Los elementos interactivos están diseñados para guiar al usuario a través del proceso de optimización de manera eficiente.

## 7.6 Guía de Uso

Para utilizar la aplicación:

1. En la pantalla inicial, haga clic en el botón "Start your fitness journey today".
2. En la pantalla principal, ingrese el número de días disponibles para entrenar en el campo de texto correspondiente.

3. Presione el botón "Enviar" para enviar los datos a la función de optimización.
4. Revise los resultados de la optimización en el área de texto.
5. Utilice el botón "Descargar" para obtener un plan de entrenamiento personalizado.
6. Si desea comenzar de nuevo, presione el botón "Reiniciar" para limpiar los registros.

## 7.7 Capturas de Pantalla

Las siguientes capturas de pantalla muestran las diferentes partes de la interfaz gráfica:

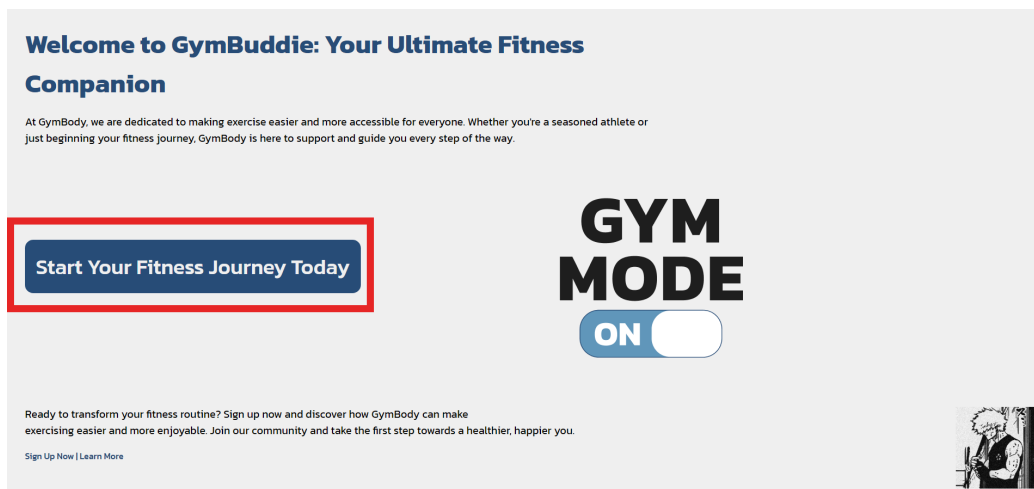


Figure 5: Pantalla inicial de la aplicación

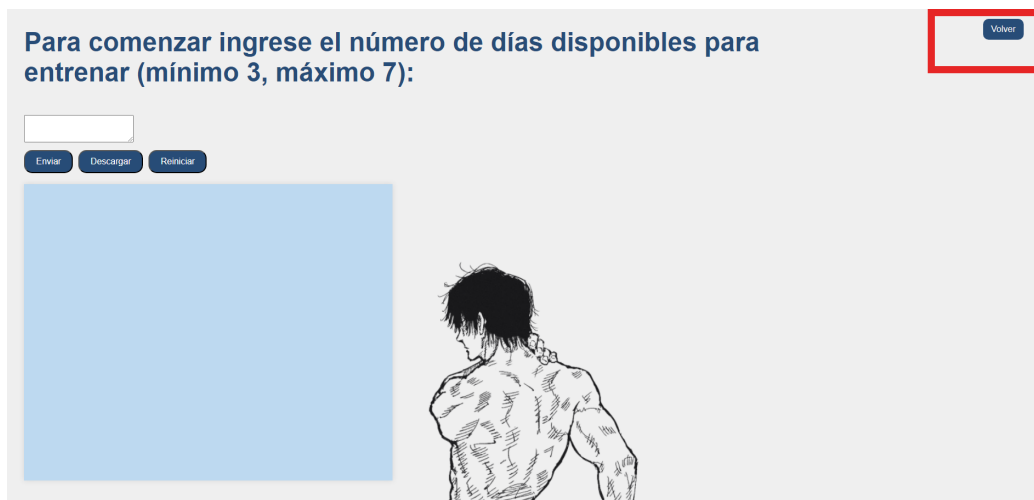


Figure 6: Pantalla principal de la aplicación

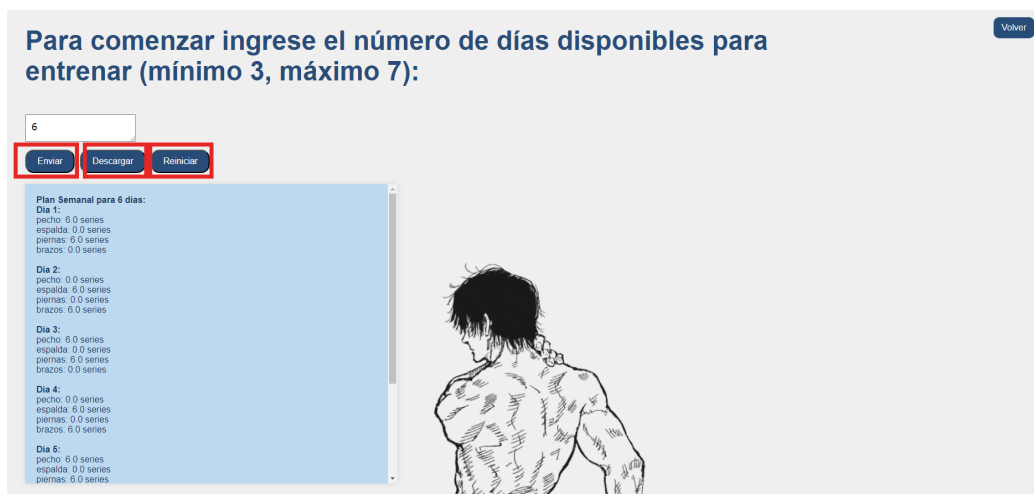
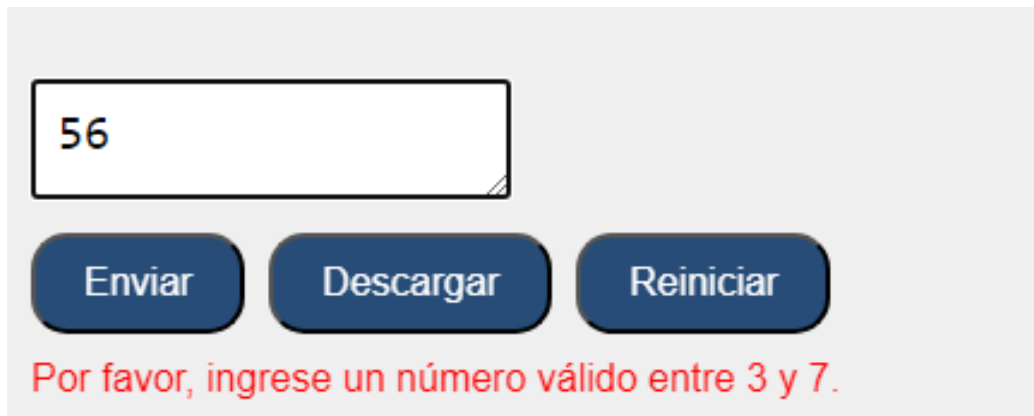


Figure 7: Botones de interacción





A screenshot of a web form for parameter validation. At the top, there is a white rectangular input field with a black border containing the number '56'. Below the input field are three dark blue buttons with rounded corners and white text: 'Enviar', 'Descargar', and 'Reiniciar'. At the bottom of the form, there is a red text message that reads: 'Por favor, ingrese un número válido entre 3 y 7.'

Figure 8: Validación de parámetros

## 8 Beneficios y Aplicaciones

### 8.1 Beneficios de la Aplicación

- **Optimización del Tiempo de Entrenamiento:** La aplicación permite a los usuarios maximizar sus resultados de entrenamiento dentro del tiempo disponible, asegurando que cada sesión sea efectiva y eficiente.
- **Planes de Entrenamiento Personalizados:** Genera planes de entrenamiento personalizados basados en la disponibilidad de días de entrenamiento, adaptándose a las necesidades y objetivos específicos de cada usuario.
- **Mejora del Desempeño Físico:** Al optimizar la distribución de los ejercicios y el descanso entre grupos musculares, los usuarios pueden mejorar su desempeño físico y minimizar el riesgo de sobreentrenamiento.
- **Facilidad de Uso:** La interfaz de usuario es intuitiva y fácil de usar, lo que permite a personas con cualquier nivel de experiencia en fitness generar un plan de entrenamiento adecuado.
- **Ahorro de Tiempo:** Los usuarios pueden ahorrar tiempo al no tener que planificar manualmente sus rutinas de entrenamiento, permitiendo más tiempo para el ejercicio efectivo.

## 8.2 Aplicaciones Potenciales

- **Gimnasios y Centros de Fitness:** La aplicación puede ser utilizada por gimnasios y centros de fitness para ofrecer planes de entrenamiento personalizados a sus miembros, mejorando la satisfacción y retención de los clientes.
- **Entrenadores Personales:** Los entrenadores personales pueden utilizar la aplicación para diseñar programas de entrenamiento óptimos para sus clientes, basados en las necesidades y disponibilidad de tiempo de cada uno.
- **Atletas y Deportistas:** Atletas de diversos niveles pueden utilizar la aplicación para optimizar sus entrenamientos, mejorando su rendimiento en competiciones y reduciendo el riesgo de lesiones.
- **Corporaciones y Empresas:** Empresas que promueven el bienestar de sus empleados pueden ofrecer la aplicación como parte de sus programas de bienestar corporativo, fomentando hábitos saludables y mejorando la productividad.
- **Educación y Formación:** Instituciones educativas y programas de formación en ciencias del deporte pueden utilizar la aplicación como herramienta didáctica para enseñar a los estudiantes sobre la planificación y optimización del entrenamiento.
- **Usuarios Individuales:** Cualquier persona interesada en mejorar su condición física puede beneficiarse de la aplicación, especialmente aquellos que buscan una guía estructurada para maximizar los resultados de sus entrenamientos en el tiempo disponible.

## 9 Conclusiones y Recomendaciones

### 9.1 Resumen de Puntos Clave

En este proyecto, hemos desarrollado una aplicación de optimización matemática para diseñar programas de entrenamiento personalizados que maximicen la hipertrofia muscular. La aplicación utiliza técnicas de programación lineal para optimizar la distribución de series de ejercicios por grupo muscular y

por día, teniendo en cuenta diversas restricciones operativas y preferencias del cliente. Los resultados muestran que la aplicación puede generar planes de entrenamiento efectivos y eficientes, adaptados a las necesidades y objetivos específicos de cada usuario.

## 9.2 Recomendaciones

- **Mejorar la Interfaz de Usuario:** Continuar mejorando la interfaz de usuario para hacerla aún más intuitiva y fácil de usar, incluyendo tutoriales y ayudas contextuales.
- **Incorporar Más Variables:** Incluir más variables y parámetros en el modelo de optimización, como el nivel de condición física del usuario, el tipo de ejercicios y las preferencias personales.
- **Desarrollar Funcionalidades Adicionales:** Desarrollar funcionalidades adicionales como la posibilidad de ajustar los planes de entrenamiento en tiempo real, integrar recordatorios y notificaciones, y ofrecer análisis detallados del progreso del usuario.
- **Expandir las Aplicaciones:** Explorar nuevas aplicaciones de la herramienta en otros contextos, como la rehabilitación física, el entrenamiento para deportes específicos y la promoción de la salud en entornos corporativos.
- **Realizar Pruebas y Validaciones:** Continuar realizando pruebas y validaciones con usuarios reales para mejorar la precisión y efectividad de los planes de entrenamiento generados por la aplicación.

## 10 Bibliografía

### References

- [1] Schoenfeld, B. J. (2010). The mechanisms of muscle hypertrophy and their application to resistance training. *Global Fitness Services, Scarsdale, New York*.

- [2] Krzysztofik, M., Wilk, M., Wojdała, G., & Gołaś, A. (2019). Maximizing muscle hypertrophy: A systematic review of advanced resistance training techniques and methods. *International Journal of Environmental Research and Public Health*, 16(24), 4897. doi: 10.3390/ijerph16244897.
- [3] Bernárdez-Vázquez, R., Raya-González, J., Castillo, D., & Beato, M. (2022). Resistance training variables for optimization of muscle hypertrophy: An umbrella review. *Frontiers in Sports and Active Living*, 4, 949021. doi: 10.3389/fspor.2022.949021.