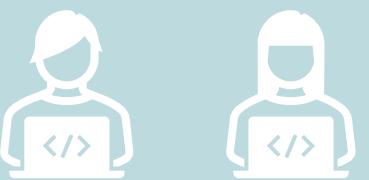


ANÁLISIS DE DATOS ACERCA DE LOS DISTINTOS SERVICIOS DE RENFE





OBTENCIÓN DE DATOS



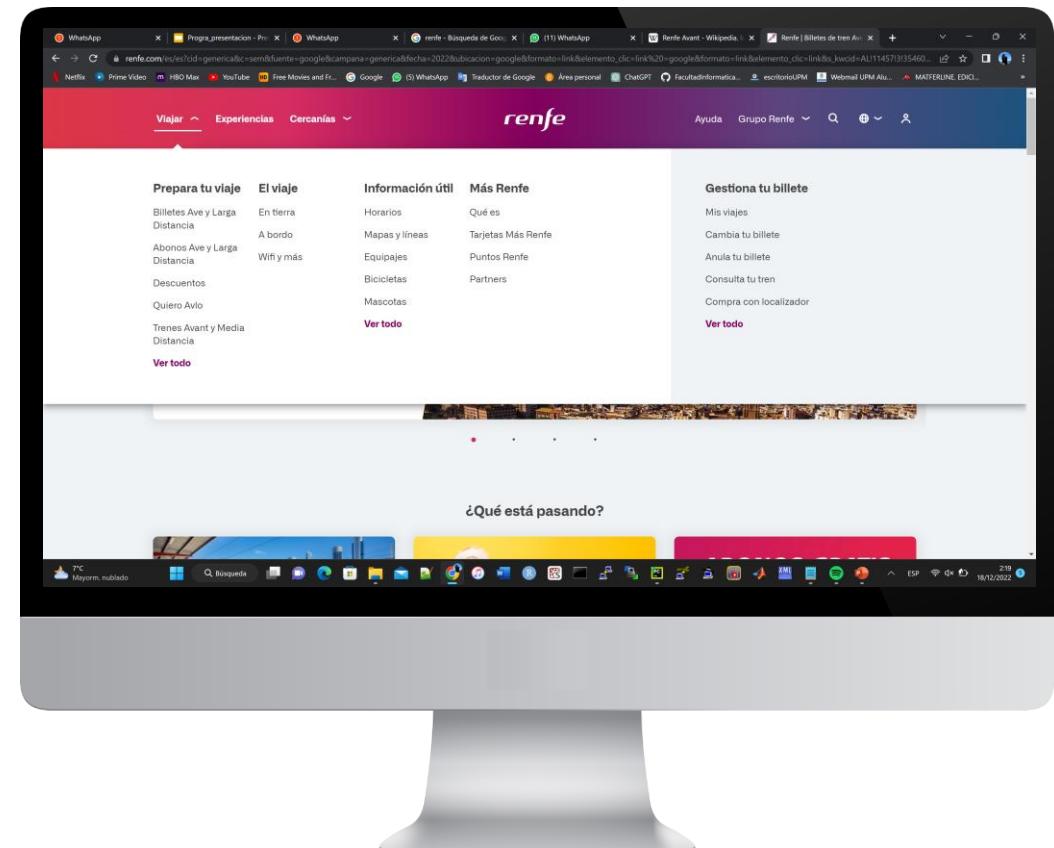
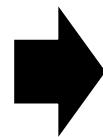
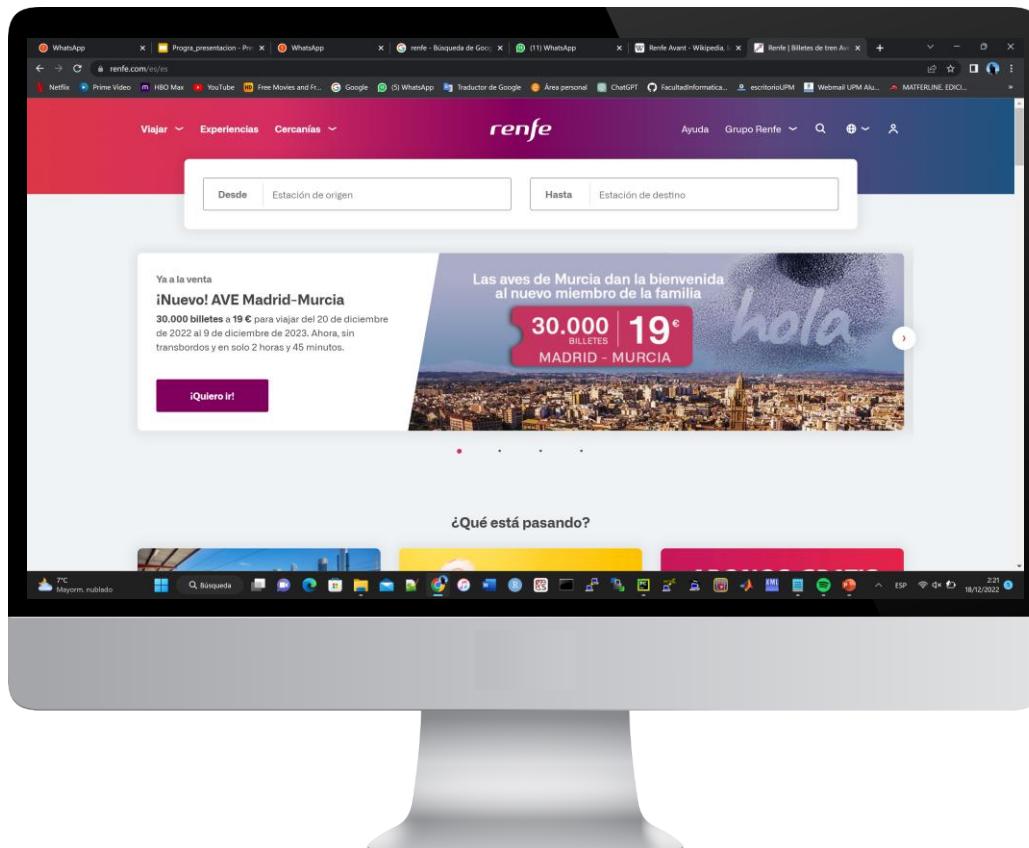
RECURSO:

**MOVIMIENTO DE TRENES DE LARGA Y MEDIA DISTANCIA
EN LA ESTACIÓN DE ATOCHA**



1. MOVIMIENTO DE TRENES DE LARGA Y MEDIA DISTANCIA EN LA ESTACIÓN DE ATOCHA

ENTRAMOS EN LA PÁGINA Y NOS DIRIGIMOS A LA SECCIÓN DESEADA



1. MOVIMIENTO DE TRENES DE LARGA Y MEDIA DISTANCIA EN LA ESTACIÓN DE ATOCHA

ENTRAMOS EN LA PÁGINA Y NOS DIRIGIMOS A LA SECCIÓN DESEADA

```
# ABRIMOS LA PÁGINA Y ACEPTAMOS LA COOKIES

driver = driver_init()
url = "https://www.renfe.com/es/es"

driver.maximize_window() # Abrimos en pantalla completa para que la página no tenga cambios respecto a la original.
time.sleep(1)
driver.get(url)

# Pulsamos el botón de las cookies
time.sleep(2)
WebDriverWait(driver, 10).until(EC.element_to_be_clickable((By.XPATH, "//button[@id='onetrust-reject-all-handler']"))).click()
time.sleep(random.uniform(0, 4))

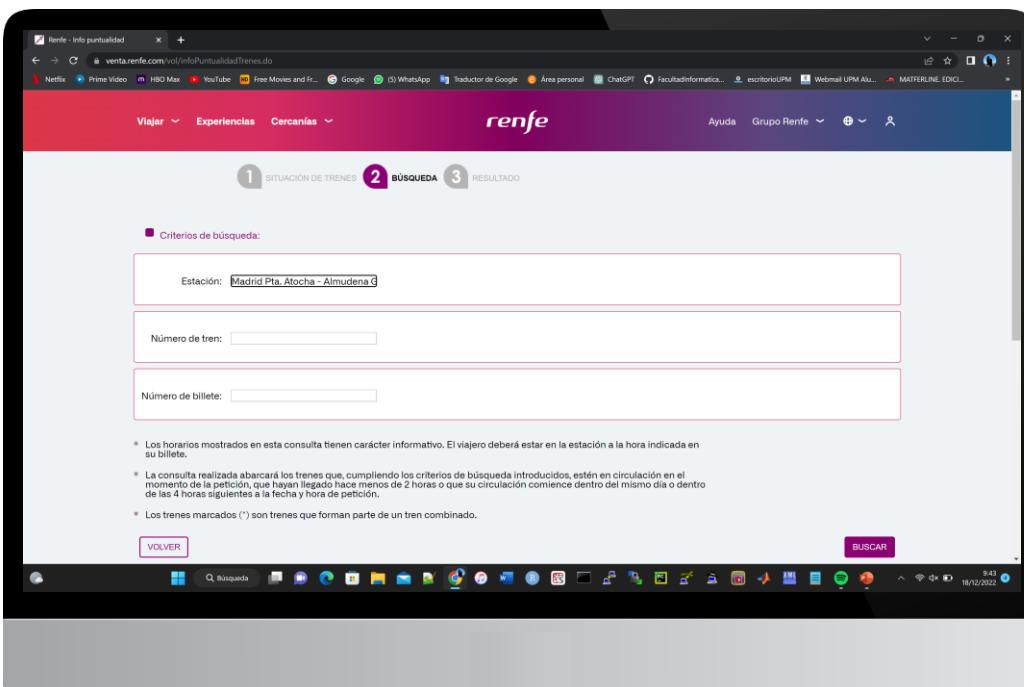
# INTERACCIONAMOS CON LA PÁGINA PARA DIRIGIRNOS A LA SECCIÓN DESEADA

for i in range(2):
    WebDriverWait(driver, 10).until(EC.element_to_be_clickable((By.XPATH, '//*[@id="contentPage"]/div/div/div[1]/div/div'\
                                                               '/div/div/div/rf-header/rf-header-top/div[1]/div/ul[1]/li[1]/rf-submenu/span'))).click()
    time.sleep(random.uniform(0, 4))
    WebDriverWait(driver, 10).until(EC.element_to_be_clickable((By.XPATH, '//*[@id="contentPage"]/div/div/div[1]/div/div'\
                                                               '/div/div/div/rf-header/rf-header-top/div[1]/div/ul[1]/li[1]/rf-submenu/div/div/ul[2]/li/ul/li[4]/a'))).click()

# Repetimos este proceso dos veces, ya que la página tiene un error y al darle la primera vez nos lleva a la página
# inicial de nuevo.
```

1. MOVIMIENTO DE TRENES DE LARGA Y MEDIA DISTANCIA EN LA ESTACIÓN DE ATOCHA

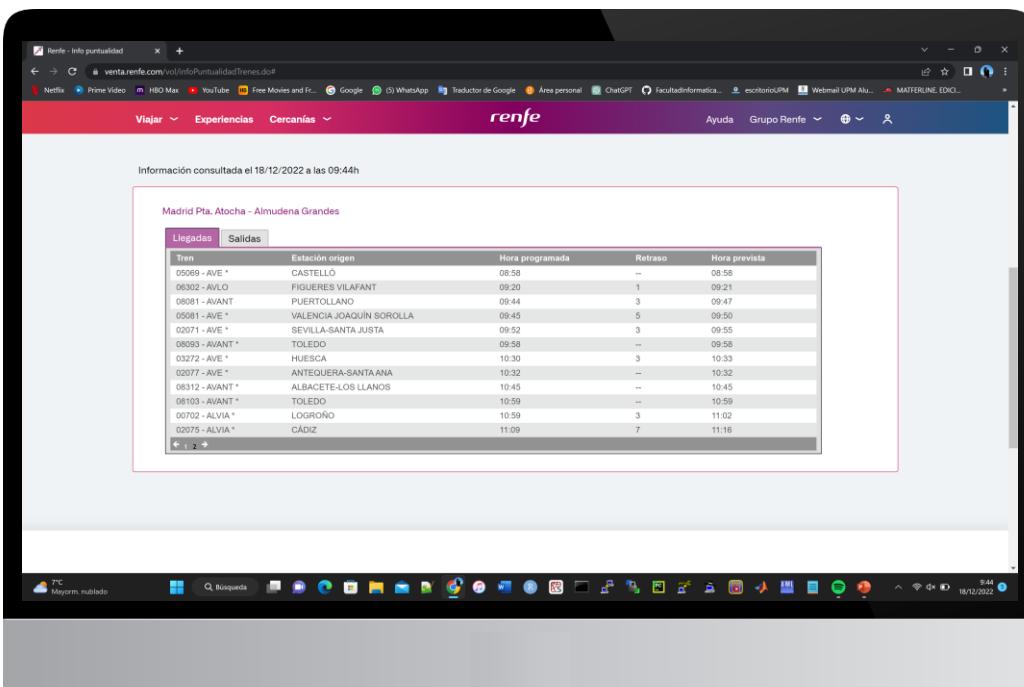
RELENAMOS LOS CAMPOS NECESARIOS



```
Estacion = driver.find_element('id', 'IdEstacion')
Estacion.send_keys('Madrid-Puerta De')
# Escribimos a medias la opción deseada para darle tiempo a exponer las opciones disponibles.
time.sleep(3)
Estacion.send_keys(' Atocha')
# Seleccionamos la opción deseada:
time.sleep(2)
WebDriverWait(driver, 10).until(EC.element_to_be_clickable((By.XPATH, "/html/body/div[11]/ul/li/strong"))).click()
time.sleep(3)
driver.execute_script("window.scrollTo(0, 100);")
time.sleep(3)
WebDriverWait(driver, 10).until(EC.element_to_be_clickable((By.XPATH, '//*[@id="divBotonBuscar1"]/a'))).click()
time.sleep(3)
# Nos movemos por la página, hacia abajo, para que la tabla esté en el dominio y podamos interaccionar con ella:
driver.execute_script("window.scrollTo(100, 600);")
time.sleep(2)
```

1. MOVIMIENTO DE TRENES DE LARGA Y MEDIA DISTANCIA EN LA ESTACIÓN DE ATOCHA

ANALIZAMOS LAS TABLAS OBTENIDAS



```
Estacion = driver.find_element('id', 'IdEstacion')
Estacion.send_keys('Madrid-Puerta De')
# Escribimos a medias la opción deseada para darle tiempo a exponer las opciones disponibles.
time.sleep(3)
Estacion.send_keys(' Atocha')
# Seleccionamos la opción deseada:
time.sleep(2)
WebDriverWait(driver, 10).until(EC.element_to_be_clickable((By.XPATH, "/html/body/div[11]/ul/li/strong"))).click()
time.sleep(3)
driver.execute_script("window.scrollTo(0, 100);")
time.sleep(3)
WebDriverWait(driver, 10).until(EC.element_to_be_clickable((By.XPATH, '//*[@id="divBotonBuscar1"]/a'))).click()
time.sleep(3)
# Nos movemos por la página, hacia abajo, para que la tabla esté en el dominio y podamos interaccionar con ella:
driver.execute_script("window.scrollTo(100, 600);")
time.sleep(2)
```

1. MOVIMIENTO DE TRENES DE LARGA Y MEDIA DISTANCIA EN LA ESTACIÓN DE ATOCHA

ANALIZAMOS LAS TABLAS OBTENIDAS

```
# PROCESAMOS LAS TABLAS DE LA SECCIÓN DE LLEGADAS

nTablas = len(driver.find_elements(By.XPATH, '//*[@id="botoneraidTablaTrenesLlegadas0"]/tbody/tr/td[2]/input'))
tablaLlegadas = []

for tabla in range(1, nTablas + 1):
    WebDriverWait(driver, 10).until(EC.element_to_be_clickable((By.XPATH, '//*[@id="botoneraidTablaTrenesLlegadas0"]/tbody/tr/td[2]/input["+str(tabla)+"")).click()
    time.sleep(2)
    nFilas = len(driver.find_elements(By.XPATH, "/html/body/form/div[3]/div[12]/div[5]/div[3]/div/div/div[1]/div[2]/table/tbody[2]/tr"))
    for x in range(1, nFilas+1):
        fila = []
        primerElemento = driver.find_element(By.XPATH, "/html/body/form/div[3]/div[12]/div[5]/div[3]/div/div/div[1]/div[2]/table/tbody[2]/tr["+ str(x) + "]/td[1]/span").text
        fila.append(primerElemento)
        for y in range(2, nColumnas+1):
            elemento = driver.find_element(By.XPATH, "/html/body/form/div[3]/div[12]/div[5]/div[3]/div/div/div[1]/div[2]/table/tbody[2]/tr["+str(x)+"]/td["+str(y)+"]").text
            fila.append(elemento)
        tablaLlegadas.append(fila)
```

```
# PROCESAMOS LAS TABLAS DE LA SECCIÓN DE SALIDAS

WebDriverWait(driver, 10).until(EC.element_to_be_clickable((By.XPATH, '//*[@id="linkSalidasTrenes"]'))).click()
nTablas2 = len(driver.find_elements(By.XPATH, '//*[@id="botoneraidTablaTrenesSalidas0"]/tbody/tr/td[2]/input'))
tablaSalidas = []

for tabla in range(1, nTablas2 + 1):
    WebDriverWait(driver, 10).until(EC.element_to_be_clickable((By.XPATH, '//*[@id="botoneraidTablaTrenesSalidas0"]/tbody/tr/td[2]/input["+str(tabla)+"")).click()
    time.sleep(2)
    nFilas2 = len(driver.find_elements(By.XPATH, "/html/body/form/div[3]/div[12]/div[5]/div[3]/div/div/div[2]/div[2]/table/tbody[2]/tr"))
    for x in range(1, nFilas2+1):
        fila = []
        primerElemento = driver.find_element(By.XPATH, "/html/body/form/div[3]/div[12]/div[5]/div[3]/div/div/div[2]/div[2]/table/tbody[2]/tr["+ str(x) + "]/td[1]/span").text
        fila.append(primerElemento)
        for y in range(2, nColumnas+1):
            elemento = driver.find_element(By.XPATH, "/html/body/form/div[3]/div[12]/div[5]/div[3]/div/div/div[2]/div[2]/table/tbody[2]/tr["+str(x)+"]/td["+str(y)+"]").text
            fila.append(elemento)
        tablaSalidas.append(fila)
```

1. MOVIMIENTO DE TRENES DE LARGA Y MEDIA DISTANCIA EN LA ESTACIÓN DE ATOCHA

ALMAZENAMOS LOS DATOS EN EL CSV: HorariosLargaDistancia.csv

```
with open('HorariosLargaDistancia.csv', 'w', newline='\n') as csvfile:
    csvtool = csv.writer(csvfile, delimiter=',')
    csvtool.writerow(["Información consultada el " + fechaConsulta])
    csvtool.writerow(["Estación: " + nombreEstacion])
    csvtool.writerow(titulosLLegadas)
    for elem in tablaLLlegadas:
        csvtool.writerow(elem)
    csvtool.writerow(titulosSalidas)
    for elem2 in tablaSalidas:
        csvtool.writerow(elem2)

driver.close()
```

Tren	Estación origen	Hora programada	Retraso	Hora prevista
08213	AVANT, TOLEDO	22:03	--	22:03
02191	AVE *, SEVILLA-SANTA JUSTA	22:17	--	22:17
00706	ALVIA, LOGROÑO	22:22	9	22:31
02175	ALVIA, CÁDIZ	22:22	--	22:22
05701	Intercity *, CASTELLÓ	22:34	--	22:34
02197	AVE *, ANTEQUERA-SANTA ANA	23:00	--	23:00
06511	AVL0, VALENCIA JOAQUÍN SOROLLA	23:06	--	23:06
03202	AVE *, BARCELONA-SANTS	23:12	--	23:12
02211	AVE *, SEVILLA-SANTA JUSTA	23:42	--	23:42
06310	AVL0, BARCELONA-SANTS	23:45	--	23:45
08220	AVANT *, PUERTOLLANO	22:15	--	22:15
01192	EUROMED *, ALICANTE/ALACANT	22:04	--	22:04
00437	ALVIA *, CASTEJÓN DE EBRO	22:20	--	22:20
15850	REGIONAL, PORTBOU	22:20	--	22:20
15238	REGIONAL, SALOU - PORT AVENTURA	22:32	--	22:32
18096	REG.EXP., TORTOSA	22:40	--	22:40
15626	REGIONAL, LLEIDA	22:42	1	22:43
06309	AVL0, MADRID PTA. ATOCHA - ALMUDENA GRANDES	22:46	--	22:46
15822	REGIONAL, PORTBOU	22:52	5	22:57
15138	REGIONAL, REUS	23:02	--	23:02
18262	REGIONAL, TORTOSA	23:32	--	23:32
15240	REGIONAL, SALOU - PORT AVENTURA	23:35	--	23:35
03203	AVE *, MADRID PTA. ATOCHA - ALMUDENA GRANDES	23:55	1	23:56
06309	AVL0, FIGUERES VILAFANT	22:56	--	22:56
05182	AVE *, MADRID-CHAMARTÍN-CLARA CAMPOAMOR	20:20	2	20:22
18181	MD *, CIUDAD REAL	20:32	6	20:38
00165	Intercity *, BARCELONA-SANTS	20:39	7	20:46
01161	EUROMED *, BARCELONA-SANTS	21:05	--	21:05
05382	AVE *, OURENSE	21:26	2	21:28
04140	ALVIA *, GIJÓN/XIXÓN	22:54	--	22:54
01181	EUROMED *, BARCELONA-SANTS	23:14	--	23:14
00165	Intercity *, CARTAGENA	20:51	7	20:58
05273	AVE *, OURENSE	07:20	--	07:20
01092	EUROMED *, BARCELONA-SANTS	07:20	--	07:20



ANÁLISIS Y REPRESENTACIÓN DE LOS DATOS



TIPOS DE TRENES DE LARGA DISTANCIA



AVE

- Alta Velocidad Española
- 310 km/h
- Ancho Internacional



AVLO

- Alta Velocidad Low Cost
- 300 km/h



Intercity

- Servicio híbrido entre larga y media distancia
- 300 km/h
- Muchas paradas.



ALVIA

- Se adapta a Ancho Internacional y Ancho Ibérico
- Mayor rango de acceso.
- 250 km/h

TIPOS DE TRENES DE MEDIA DISTANCIA



REGIONAL

- Radio de acción de 300 km
- Ámbito regional.
- 160 km/h



REG.EXP

- Radio de acción de 300 km
- Ámbito regional
- 160 km/h
- Menos paradas, menos tiempo



MD

- RENFE Media Distancia
- Ámbito regional.
- 160 km/h



AVANT

- Su trayecto suele incluir diversas provincias dentro de una misma comunidad autónoma
- 250 km/h



PREGUNTA 1

¿CUÁL ES LA FRECUENCIA DE ACTIVIDAD SEGÚN EL TIPO DE TREN?



MODIFICAMOS LOS DATOS PARA PODER ANALIZARLO

```
df = pd.read_csv('CSVAcumulativo.csv', index_col=0)

# Primero cambiamos los valores nulos por 0, necesario para después calcular la media del retraso.
df.replace('--', 0, inplace=True)

# Después, eliminamos los asteriscos de los tipos de trenes, ya que la distinción a la que hace referencia no nos hace falta.
df['Tren'] = df['Tren'].apply(lambda x: re.sub('\s*', '', x))

# Además, convertimos el tipo de tren, 'TORRE ORO' en Intercity ya que pese a que aparece como un tipo de tren a parte,
# pertenece a dicha clase. Lo mismo ocurre con 'EUROMED', que es del tipo 'ALVIA'.
df.replace('TORRE ORO', 'Intercity', inplace=True)
df.replace('EUROMED', 'ALVIA', inplace=True)

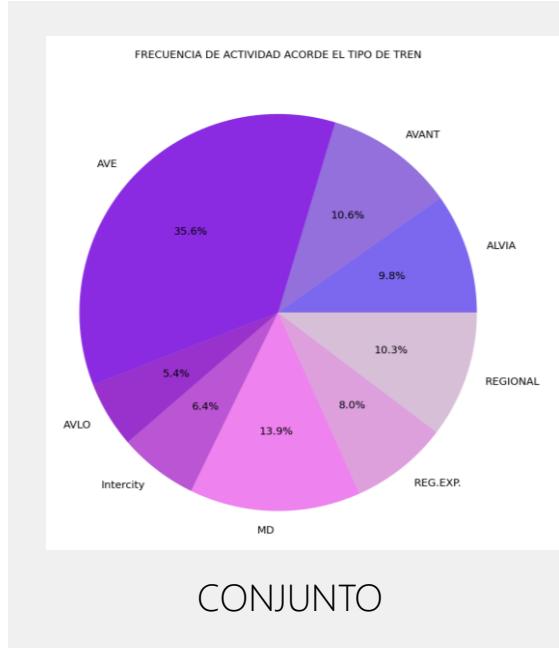
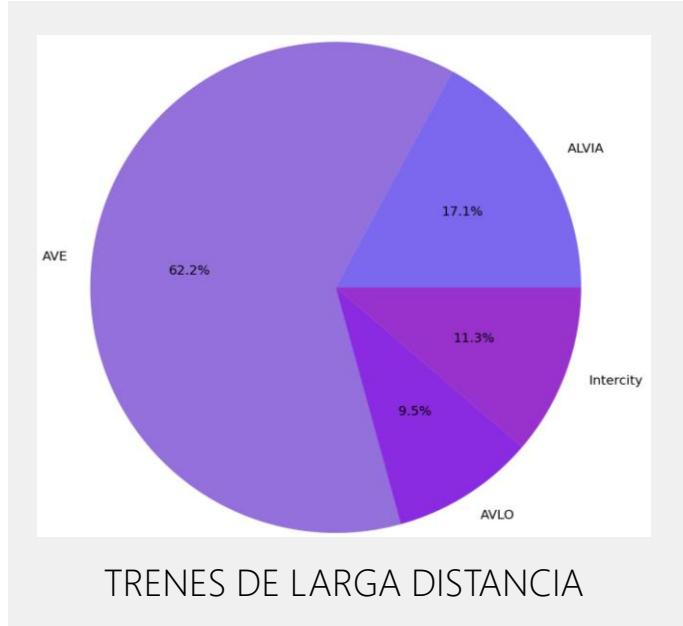
# Convertimos además los datos de la columna 'Retraso' en enteros, para poder operar con ellos.
df['Retraso'] = df['Retraso'].astype('int64')
```

¿CUÁL ES LA FRECUENCIA DE ACTIVIDAD SEGÚN EL TIPO DE TREN?

```
# Larga distancia:  
df_AVE = grouped_df.get_group('AVE')  
df_Avlo = grouped_df.get_group('AVLO')  
df_Alvia = grouped_df.get_group('ALVIA')  
df_Intercity = grouped_df.get_group('Intercity')
```

```
# Unimos los distintos dataframes:  
df_LDist = pd.concat([df_AVE, df_Avlo, df_Alvia, df_Intercity])  
df_MDist = pd.concat([df_MD, df_REGEXP, df_REGIONAL, df_AVANT])  
tipos_LD = df_LDist['Tren'].unique()  
tipos_MD = df_MDist['Tren'].unique()
```

```
# Media distancia:  
df_MD = grouped_df.get_group('MD')  
df_REGIONAL = grouped_df.get_group('REGIONAL')  
df_REGEXP = grouped_df.get_group('REG.EXP.')  
df_AVANT = grouped_df.get_group('AVANT')
```

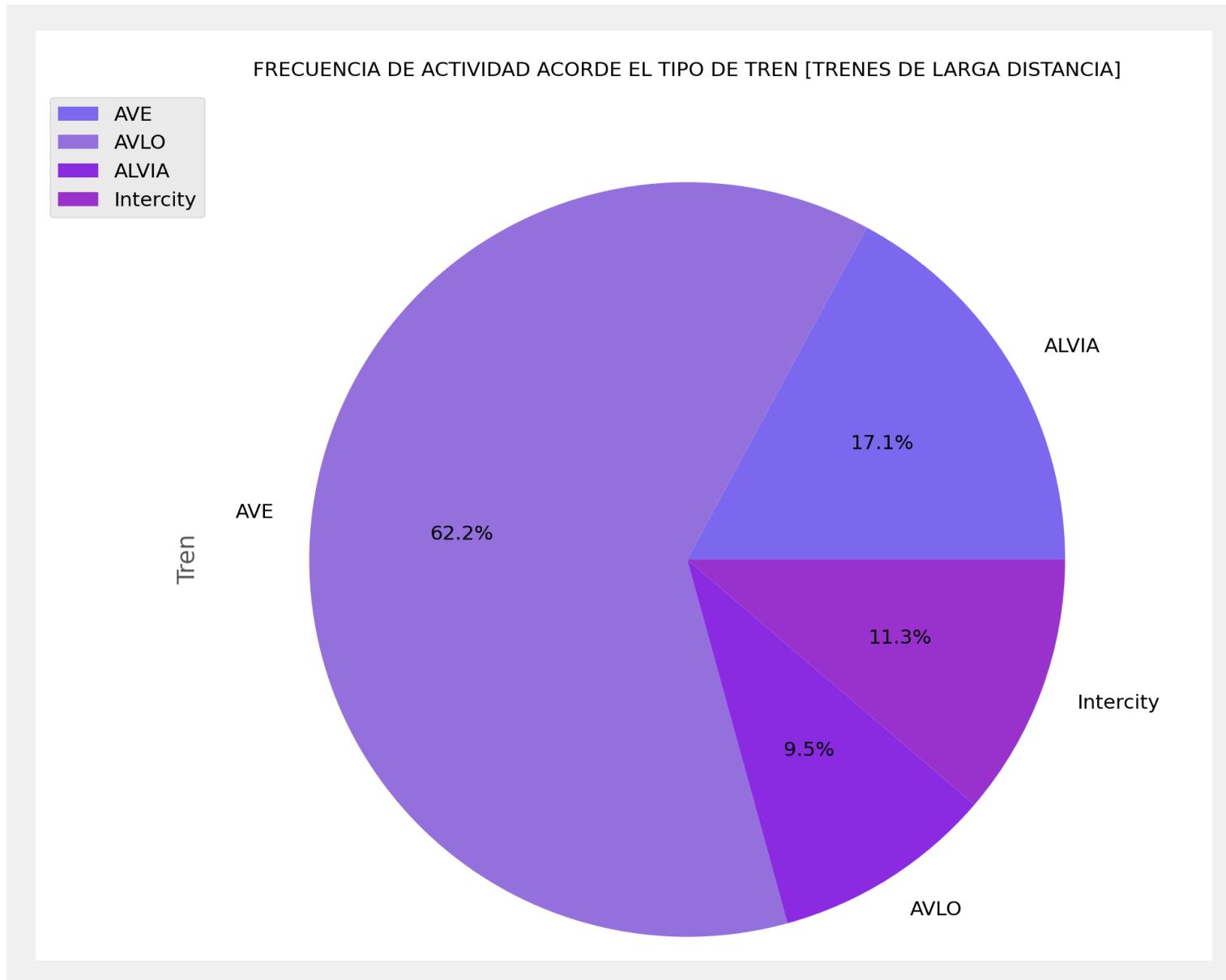


FRECUENCIA DE ACTIVIDAD: TRENES DE LARGA DISTANCIA

```
# Contamos las apariciones de cada tipo de tren en la columna que lo indica, para saber en forma de porcentaje, la
# cantidad de trenes que hay activos según su tipo correspondiente.
grouped_df_LDist = df_LDist.groupby('Tren')
er_distribution_LD = grouped_df_LDist['Tren'].count()
# Mostramos los datos obtenidos en un diagrama de tarta:
plt.style.use('ggplot')
plt.title('FRECUENCIA DE ACTIVIDAD ACORDE EL TIPO DE TREN [TRENES DE LARGA DISTANCIA]', fontdict={'fontsize': 10})
plt.ylabel('Porcentaje')
colores = ['mediumslateblue', 'mediumpurple', 'blueviolet', 'darkorchid']
er_distribution_LD.plot(kind='pie', autopct='%1.1f%%', colors=colores)
plt.legend(tipos_LD, bbox_to_anchor=(0, 1))

plt.show()
```

FRECUENCIA DE ACTIVIDAD: TRENES DE LARGA DISTANCIA

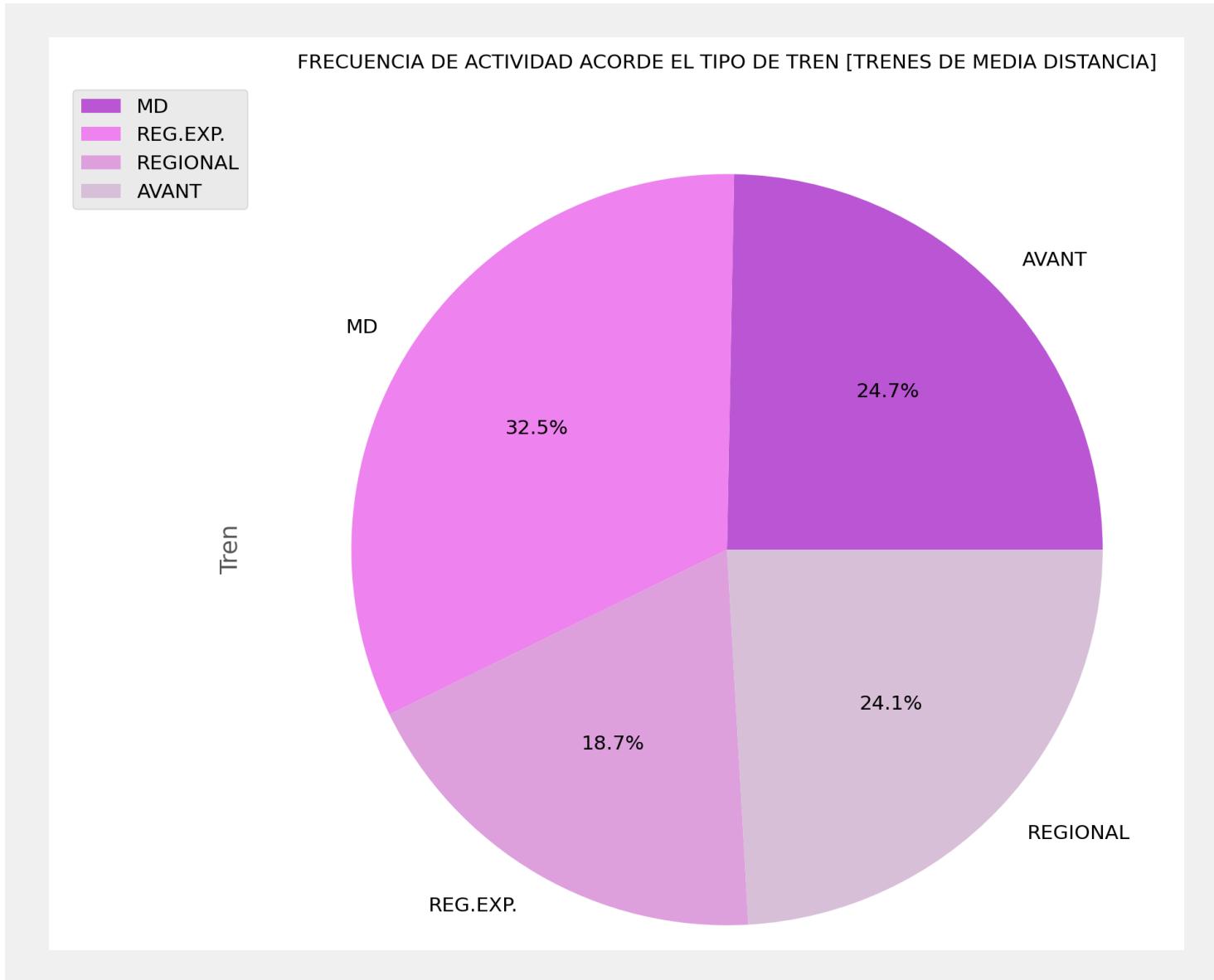


FRECUENCIA DE ACTIVIDAD: TRENES DE MEDIA DISTANCIA

```
# Contamos las apariciones de cada tipo de tren en la columna que lo indica, para saber en forma de porcentaje, la
# cantidad de trenes que hay activos según su tipo correspondiente.
grouped_df_MDist = df_MDist.groupby('Tren')
er_distribution_MD = grouped_df_MDist['Tren'].count()
# Mostramos los datos obtenidos en un diagrama de tarta:
plt.style.use('ggplot')
plt.title('FRECUENCIA DE ACTIVIDAD ACORDE EL TIPO DE TREN [TRENES DE MEDIA DISTANCIA]', fontdict={'fontsize': 10})
plt.ylabel('Porcentaje')
colores = ['mediumorchid', 'violet', 'plum', 'thistle']
er_distribution_MD.plot(kind='pie', autopct='%1.1f%%', colors=colores)
plt.legend(tipos_MD, bbox_to_anchor=(0, 1))

plt.show()
```

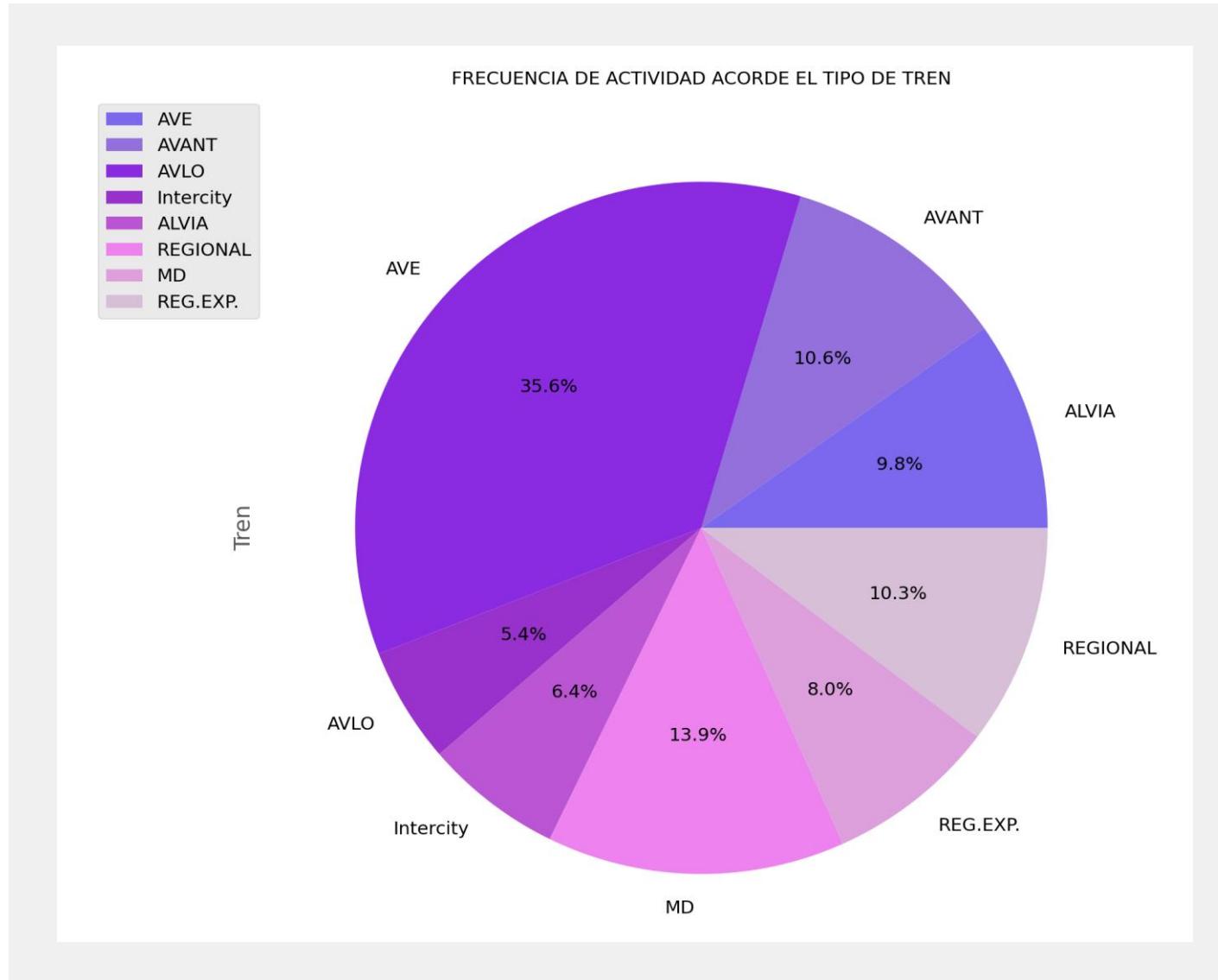
FRECUENCIA DE ACTIVIDAD: TRENES DE MEDIA DISTANCIA



FRECUENCIA DE ACTIVIDAD SEGÚN EL TIPO DE TREN

```
# Contamos las apariciones de cada tipo de tren en la columna que lo indica, para saber en forma de porcentaje, la  
# cantidad de trenes que hay activos según su tipo correspondiente.  
er_distribution2 = grouped_df['Tren'].count()  
# Mostramos los datos obtenidos en un diagrama de tarta:  
plt.style.use('ggplot')  
plt.title('FRECUENCIA DE ACTIVIDAD ACORDE EL TIPO DE TREN', fontdict={'fontsize': 10})  
plt.ylabel('Porcentaje')  
colores = ['mediumslateblue', 'mediumpurple', 'blueviolet', 'darkorchid', 'mediumorchid', 'violet', 'plum', 'thistle']  
er_distribution2.plot(kind='pie', autopct='%1.1f%%', colors=colores)  
tipos = df['Tren'].unique()  
plt.legend(tipos, bbox_to_anchor=(0, 1))  
  
plt.show()
```

FRECUENCIA DE ACTIVIDAD SEGÚN EL TIPO DE TREN





PREGUNTA 2

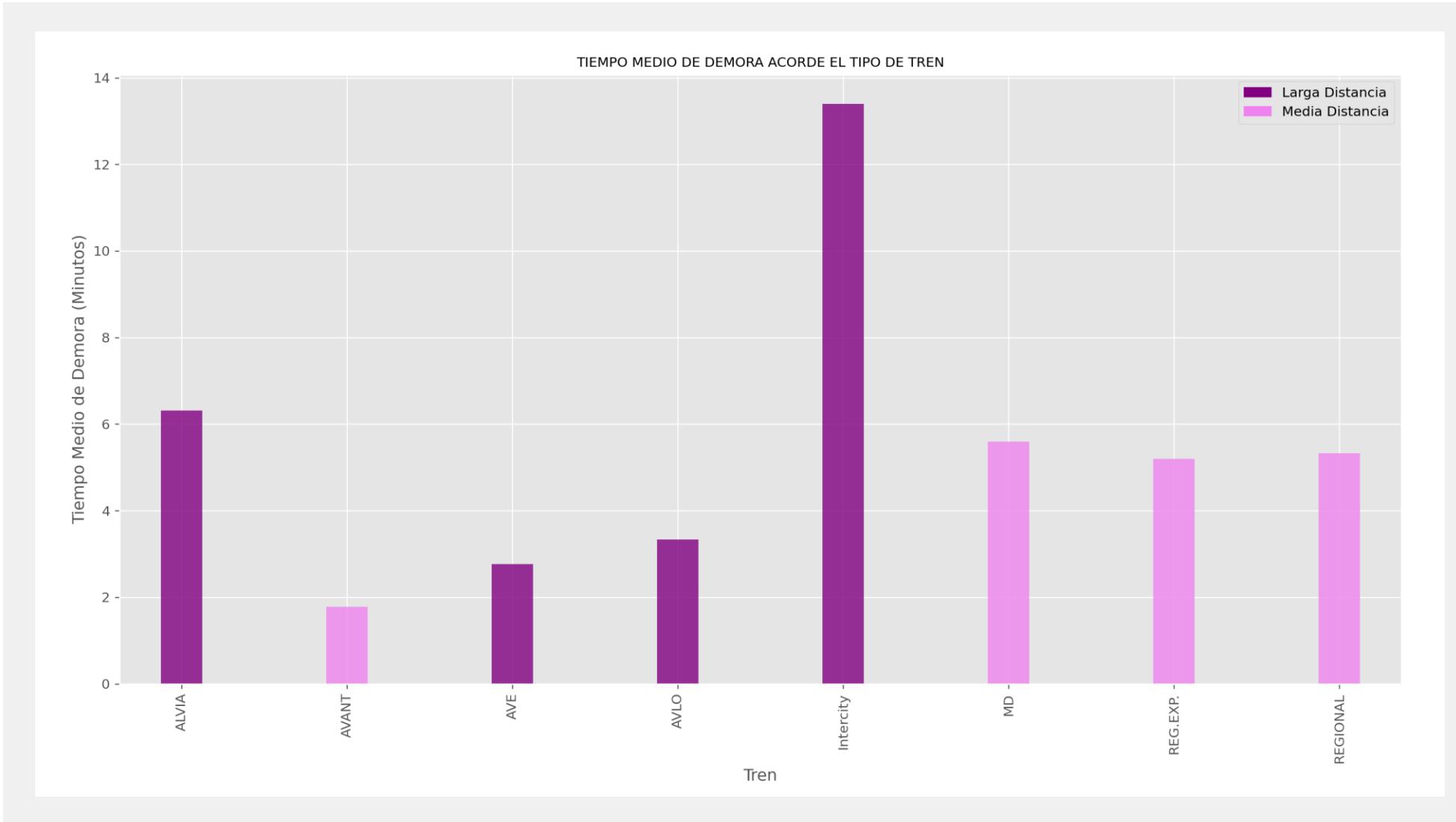
¿CUÁL ES EL TIEMPO MEDIO DE DEMORA SEGÚN EL TIPO DE TREN?



¿CUÁL ES EL TIEMPO MEDIO DE DEMORA SEGÚN EL TIPO DE TREN?

```
# Calculamos la media de los datos agrupados y le indicamos que solo opere las columnas con valores numéricos, en  
# nuestro caso, la columna donde se indican los retrasos.  
er_distribution = grouped_df.mean(numeric_only=True)[ 'Retraso' ]  
# Esto nos devuelve una serie.  
# Mostramos los datos obtenidos en un histograma:  
plt.style.use('ggplot')  
plt.title('TIEMPO MEDIO DE DEMORA ACORDE EL TIPO DE TREN', fontdict={'fontsize': 10})  
plt.ylabel('Tiempo Medio de Demora (Minutos)')  
  
colors = []  
for count in er_distribution.keys():  
    if count in tipos_LD:  
        colors.append('purple')  
    else:  
        colors.append('violet')  
  
er_distribution.plot(kind='bar', color=colors, alpha=0.8, width=0.25)  
  
First_patch = mpatches.Patch(color='purple', label='Larga Distancia')  
Second_patch = mpatches.Patch(color='violet', label='Media Distancia')  
plt.legend(handles=[First_patch, Second_patch])  
  
plt.show()
```

TIEMPO MEDIO DE DEMORA SEGÚN EL TIPO DE TREN





ANÁLISIS DE DATOS ACERCA DE LOS DISTINTOS SERVICIOS DE RENFE