

PROCESAMIENTO PYTHON

Generación y ajuste procesamiento python, panel EP reporte de aprendices no activos

```
Procesamiento_Aprendices_EP.py X Release Notes: 1.1200
C:\Users\LENOVO > OneDrive > Escritorio > Panel EP > Procesamiento_Aprendices_EP.py > ...
135 # & (df_principal['ESTADO_REGIS_ACADE_APREND'] != 'Por Certificar', 'Cancelado', 'Certificado', 'Retiro Voluntario', 'Traslado']])) # SE AGREGÓ EL 02-09-2025 PARA NO TRAER LOS OTROS ESTADOS DE
136 df_4 = df_4.iloc[:, [0,1,19,14,15,20,21,22,16,17,18,25,28,26,27,23,24,2,3,4,5,6,8,29,31,32,36,30,34,35,40,37,38,39,41,42,43,44,56,45,46,47,48,49,53,54,55,57,58,59,50,51,52]]] # SE AGREGÓ LA COLUMNA 41,44 y 47 PARA TRAER
137 #df_4.to_excel('5_Aprendices_Vencimiento_Terminos.xlsx', index=False)
138 exportar_excel_segmentado(df_4, '5_Aprendices_Vencimiento_Terminos.xlsx')
139 #print("*****Finalizado 5_Aprendices_Vencimiento_Terminos*****")
140
141 df_5 = df_principal.loc[(df_principal['ESTADO_RAP_EP'] != 'POR EVALUAR')
142 & (df_principal['ESTADO_CMP_EP'] != 'POR EVALUAR')
143 & ~(df_principal['ESTADO_REGIS_ACADE_APREND'].isin(['Cancelado', 'Certificado', 'Retiro Voluntario', 'Traslado']])) # SE AGREGÓ EL 02-09-2025 PARA NO TRAER LOS OTROS ESTADOS DE APRENDIZ]
144 df_5 = df_5.iloc[:, [0,1,19,14,15,20,21,22,16,17,18,25,28,26,27,23,24,2,3,4,5,6,8,29,31,32,34,35,36,37,38,39,40,41,42,43,44,56,45,46,47,48,49,53,54,55,57,58,59,50,51,52]]] # SE AGREGÓ LA COLUMNA 41,44 y 47 PARA TRAER
145 #df_5.to_excel('6_Aprendices_Situaciones_Atipicas.xlsx', index=False)
146 exportar_excel_segmentado(df_5, '6_Aprendices_Situaciones_Atipicas.xlsx')
147 #print("*****Finalizado 6_Aprendices_Situaciones_Atipicas*****")
148
149 df_6 = df_principal.loc[(df_principal['MESES_TRANSCURRIDOS DE INICIO EP'] >= 0)
150 & (df_principal['RAPS_NO_APROBADOS'] == 0) & (df_principal['RAPS_POR_EVALUAR'] == 0)
151 & (df_principal['ESTADO_CMP_EP'] == 'APROBADO')
152 & ~(df_principal['ESTADO_REGIS_ACADE_APREND'].isin(['Cancelado', 'Certificado', 'Retiro Voluntario', 'Traslado']])) # SE AGREGÓ EL 02-09-2025 PARA NO TRAER LOS OTROS ESTADOS DE APRENDIZ]
153 df_6 = df_6.iloc[:, [0,1,19,14,15,20,21,22,16,17,18,25,28,26,27,23,24,2,3,4,5,6,8,29,31,32,36,30,34,35,40,37,38,39,41,42,43,44,56,45,46,47,48,49,53,54,55,57,58,59,50,51,52]]] # SE AGREGÓ LA COLUMNA 41,44 PARA TRAER RAPS_
154 #df_6.to_excel('7_Aprendices_Por_Certificar.xlsx', index=False)
155 exportar_excel_segmentado(df_6, '7_Aprendices_Por_Certificar.xlsx')
156 #print("*****Finalizado 7_Aprendices_Por_Certificar*****")
157
158 df_7 = df_principal.loc[(df_principal['MESES_TRANSCURRIDOS DE INICIO EP'] > 0)
159 & (df_principal['ESTADO_REGIS_ACADE_APREND'].isin(['Cancelado', 'Certificado', 'Retiro Voluntario', 'Traslado']]))
160 df_7 = df_7.iloc[:, [0,1,19,14,15,20,21,22,16,17,18,25,28,26,27,23,24,2,3,4,5,6,8,29,31,32,36,30,34,35,40,37,38,39,41,42,43,44,56,45,46,47,48,49,53,54,55,57,58,59,50,51,52]]] # SE AGREGÓ LA COLUMNA 41,44 PARA TRAER RAPS_
161 #df_7.to_excel('8_Aprendices_Estados_NO_Activos.xlsx', index=False)
162 exportar_excel_segmentado(df_7, '8_Aprendices_Estados_NO_Activos.xlsx')
163 #print("*****Finalizado 8_Aprendices_Estados_NO_Activos.xlsx*****")
164
165 df_8 = df_principal.loc[
166 (
167 (df_principal['ESTADO_RAP_EP'] == 'POR EVALUAR') &
168 (df_principal['RAPS_POR_EVALUAR'] > 0) &
169 (df_principal['RAPS_NO_APROBADOS'] > 0)
170 )
171 &
172 (
173 df_principal['ESTADO_REGIS_ACADE_APREND'].isin([
174 'Formacion', 'Induccion', 'Condicionado', 'Apilazado'
175 ])
176 )
177 ]
178 }
```

Ajuste de procesamiento de python, Eliminar y modificar columnas nuevas creadas en SQL

```
Procesamiento_Aprendices_EP.py X Release Notes: 1.1200
C:\Users\LENOVO > OneDrive > Escritorio > Panel EP > Procesamiento_Aprendices_EP.py > ...
46 # Asignar los encabezados de hoja1 a hoja2
47 hoja2.columns = hoja1.columns
48 hoja3.columns = hoja1.columns
49 print("*****asignado los encabezados de las hojas de Aprendices*****")
50
51 # Unir ambas hojas en un solo DataFrame
52 df_principal = pd.concat([hoja1, hoja2, hoja3], ignore_index=True)
53 print("*****unidas las hojas de Aprendices*****")
54
55 df_principal['GRUPO'] = df_principal['GRUPO'].astype('Int64')
56 df_principal['NUM_DOCUMENTO'] = df_principal['NUM_DOCUMENTO'].astype('str')
57
58
59 df_principal['GRUPO_DOCUMENTO'] = (
60 df_principal['GRUPO'].astype('Int64').astype(str) + '-' +
61 df_principal['NUM_DOCUMENTO']
62 )
63
64 print(df_principal[['GRUPO', 'GRUPO_DOCUMENTO']].head(10))
65
66 # Mostrar primeras filas para verificar
67 print(df_principal.head())
68 print(f"Total de filas combinadas: {len(df_principal)}")
69
70
71 df_principal.rename(columns={'PSP_FCH_PRESENTACION': 'FCH PRESENTACIÓN P T&T'}, inplace=True)
72 df_principal.rename(columns={'APOYO DE SOSTENIMIENTO FIC': 'APOYO DE SOSTENIMIENTO PARA FIC'}, inplace=True)
73 df_principal.rename(columns={'24 MESES DE FECHA INICIO EP': 'FECHA FIN 24 MESES DESPUES DE FECHA INICIO EP'}, inplace=True)
74 df_principal.rename(columns={'PORC_RAPS_POR_EVALUAR': 'RAPS POR EVALUAR'}, inplace=True)
75 df_principal.rename(columns={'MESES_TRANSCURRIDOS': 'MESES TRANSCURRIDOS DE INICIO EP'}, inplace=True)
76
77
78 #SE AGREGAN COLUMNAS POR SOLICITUD DE LA VISTA NUEVA DE INDICADORES
79
80 print("*****se realizó el cambio de nombre de las columnas*****")
81
82 # Filtrando las columnas utilizando loc
83 df_1 = df_principal.loc[(df_principal['MESES_TRANSCURRIDOS DE INICIO EP'] >= -3)
84 & (df_principal['MESES_TRANSCURRIDOS DE INICIO EP'] < 0)
85 & (df_principal['PROGRAMA_ESPECIAL'].isnull() | (df_principal['PROGRAMA_ESPECIAL'] == ''))
86 & (df_principal['ESTADO_REGIS_ACADE_APREND'] != 'Por Certificar')]] #SE INHIBIÓ EL 02-09-2025 PARA HACER LA LINEA 52
87 & ~(df_principal['ESTADO_REGIS_ACADE_APREND'].isin(['Por Certificar', 'Cancelado', 'Certificado', 'Retiro Voluntario', 'Traslado']])) # SE AGREGÓ EL 02-09-2025 PARA NO TRAER LOS OTROS ESTADOS DE APRENDIZ]
88 #Generar el nuevo dataframe con el índice de las columnas requeridas con iloc.
89 df_1 = df_1.iloc[:, [0,1,19,14,15,20,21,22,16,17,18,25,28,26,27,23,24,2,3,4,5,6,8,29,31,32,36,30,34,35,40,37,38,39,41,42,43,44,56,45,46,47,48,49,53,54,55,57,58,59,50,51,52]]] # SE AGREGÓ LA COLUMNA 41,44 y 47 PARA TRAER
90 #print("*****Finalizado 2_Aprendices_Proximos_Iniciar_EP*****")
91 #df_1.to_excel('2_Aprendices_Proximos_Iniciar_EP.xlsx', index=False)
92 exportar_excel_segmentado(df_1, '2_Aprendices_Proximos_Iniciar_EP.xlsx')
```

Ajuste y generación de procesamiento python, exportar anexos por centro de formación

```
Procesamiento_Aprendices_EP.py  Convertir de word a pdf.py  Release Notes: 1.120.0
C:\Users\LENOVO > OneDrive > Escritorio > SENA > SENA > Python > Generación reportes de horas trimestrales > Paso 5 - Convertir de Word a PDF > Convertir de word a pdf.py > ...

1 from docx2pdf import convert
2 import os
3
4 # Ruta donde tienes los archivos .docx
5 carpeta_entrada = r"C:\Users\LENOVO\OneDrive\Escritorio\SENA\SENA\Python\Generación reportes de horas trimestrales\Paso 5 - Convertir de Word a PDF\Informes VF 29842826"
6 carpeta_salida = r"C:\Users\LENOVO\OneDrive\Escritorio\SENA\SENA\Python\Generación reportes de horas trimestrales\Paso 5 - Convertir de Word a PDF\Informes PDF"
7
8 # Crear carpeta de salida si no existe
9 os.makedirs(carpeta_salida, exist_ok=True)
10
11 # Recorrer todos los archivos de la carpeta
12 for archivo in os.listdir(carpeta_entrada):
13     if archivo.endswith(".docx"):
14         ruta_docx = os.path.join(carpeta_entrada, archivo)
15         ruta_pdf = os.path.join(carpeta_salida, archivo.replace(".docx", ".pdf"))
16
17         print(f"Convirtiendo: {archivo} a PDF")
18         convert(ruta_docx, ruta_pdf)
19
20 print("✓ Conversión completada.")
```

Generación y ajuste de procesamiento de python, extraer porcentajes de celdas de los anexos

```
Procesamiento_Aprendices_EP.py  Convertir de word a pdf.py  extraer porcentajes de celdas de los anexos.py  Release Notes: 1.120.0
C:\Users\LENOVO > OneDrive > Escritorio > SENA > SENA > Python > Generación reportes de horas trimestrales > Paso 2 - Copiar celdas del anexo > extraer porcentajes de celdas de los anexos.py > ...

1 import pandas as pd
2 from openpyxl import load_workbook
3 from pathlib import Path
4
5 # -----
6 # CONFIGURACIÓN
7 # -----
8
9 # Carpeta donde están los archivos de los centros
10 ruta_entrada = Path(r"C:\Users\LENOVO\OneDrive\Escritorio\SENA\SENA\Python\Generación reportes de horas trimestrales\Paso 2 - Copiar celdas del anexo\Reportes - Paso 1")
11
12 # Ruta donde se guardará el consolidado
13 ruta_salida = Path(r"C:\Users\LENOVO\OneDrive\Escritorio\SENA\SENA\Python\Generación reportes de horas trimestrales\Paso 2 - Copiar celdas del anexo\porcentajes para informes 23842826.xlsx")
14
15 # -----
16 # FUNCIONES
17 # -----
18
19 def formato_porcentaje(valor):
20     """Convierte un número o texto a formato porcentaje (ej: 0.45 -> 45%, 45 -> 45%)."""
21     if valor is None:
22         return None
23     try:
24         # Si el valor ya tiene '%', lo dejamos igual
25         if isinstance(valor, str) and '%' in valor:
26             return valor.strip()
27         # Si es decimal (ej: 0.45), convertir a porcentaje
28         elif isinstance(valor, (float, int)):
29             return f"{valor * 100:.0f}%"
30         else:
31             return str(valor)
32     except:
33         return valor
34
35 # -----
36 # PROCESAMIENTO
37 # -----
38
39 # Iterar por todos los archivos .xlsx en la carpeta
40 for archivo in ruta_entrada.glob("*.xlsx"):
41     try:
42         wb = load_workbook(archivo, data_only=True)
43
44         # Verificar si existe la hoja "Análisis Anual"
45         if "Análisis Anual" not in wb.sheetnames:
46             print(f"⚠ El archivo {archivo.name} no tiene hoja 'Análisis Anual'. Se omite.")
47             continue
48     except:
```

Ajuste y generación de procesamiento python, copiar tablas dinamicas completas al Word

```
Procesamiento_Aprendices_EP.py  Convertir de word a pdf.py  extraer porcentajes de celdas de los anexos.py  exportar centro y unir la tabla dinamica de otro archivo- con imagen de criterios.py  Telesse Notes: 1.120.0
C:\Users\LENOVO> OneDrive > Escritorio > SENA > SENA > Python > Generación reportes de horas trimestrales > Paso 1 - Exportar por centro de formación > exportar centro y unir la tabla dinamica de otro archivo- con imagen de criterios.py > ...

1 import pandas as pd
2 from openpyxl import load_workbook, Workbook
3 from openpyxl.styles import PatternFill, Alignment
4 from openpyxl.utils import get_column_letter
5 from openpyxl.drawing.image import Image
6 from openpyxl.worksheet.table import Table, TableStyleInfo
7 import win32com.client as win32
8 import os
9 import zipfile
10
11 # --- CONFIGURACIONES ---
12 ruta_anual = r"C:\Users\LENOVO\OneDrive\Escritorio\SENA\SENA\Python\Generación reportes de horas trimestrales\Paso 1 - Exportar por centro de formación\Centros anual prueba.xlsx"
13 ruta_mensual = r"C:\Users\LENOVO\OneDrive\Escritorio\SENA\SENA\Python\Generación reportes de horas trimestrales\Paso 1 - Exportar por centro de formación\Centros mensual.xlsx"
14 ruta_casos = r"C:\Users\LENOVO\OneDrive\Escritorio\SENA\SENA\Python\Generación reportes de horas trimestrales\Paso 1 - Exportar por centro de formación\Casos Atípicos.xlsx"
15 ruta_instructores_varios_centros = r"C:\Users\LENOVO\OneDrive\Escritorio\SENA\SENA\Python\Generación reportes de horas trimestrales\Paso 1 - Exportar por centro de formación\Instructores con varios centros.xlsx"
16 ruta_instructores = r"C:\Users\LENOVO\OneDrive\Escritorio\SENA\SENA\Python\Generación reportes de horas trimestrales\Paso 1 - Exportar por centro de formación\Instructores sin programación.xlsx"
17 ruta_plantilla = r"C:\Users\LENOVO\OneDrive\Escritorio\SENA\SENA\Python\Generación reportes de horas trimestrales\Paso 1 - Exportar por centro de formación\Anexo 1 Horas programadas 2026 Nacional 2204WV Documento de T
18 ruta_imagen_criterios = r"C:\Users\LENOVO\OneDrive\Escritorio\SENA\SENA\Python\Generación reportes de horas trimestrales\Paso 1 - Exportar por centro de formación\criterios.png"
19 carpeta_salida = r"C:\Users\LENOVO\OneDrive\Escritorio\SENA\SENA\Python\Generación reportes de horas trimestrales\Paso 1 - Exportar por centro de formación"
20 columna_clave = "DATOS CENTRO"
21
22 os.makedirs(carpeta_salida, exist_ok=True)
23
24 # --- CARGAR DATOS (con chequeo de existencia de archivos) ---
25 def leer_excel_si_existe(ruta, **kwargs):
26     if os.path.exists(ruta):
27         return pd.read_excel(ruta, **kwargs)
28     else:
29         print(f"⚠ No se encontró: {ruta} - se creará DataFrame vacío.")
30         return pd.DataFrame()
31
32 df_anual = leer_excel_si_existe(ruta_anual)
33 df_mensual = leer_excel_si_existe(ruta_mensual)
34 df_casos = leer_excel_si_existe(ruta_casos)
35 df_instructores = leer_excel_si_existe(ruta_instructores)
36 df_instructores_varios_centros = leer_excel_si_existe(ruta_instructores_varios_centros)
37
38 # --- OBTENER CENTROS DESDE REPORTE ANUAL (sin validar comunes) ---
39 if columna_clave not in df_anual.columns:
40     raise KeyError(f"La columna clave '{columna_clave}' no existe en el archivo anual.")
41 centros_reporte_anual = sorted(df_anual[columna_clave].dropna().unique())
42
43 # --- EXTRAER FORMATO DEL ENCABEZADO (si existe archivo mensual) ---
44 estilos_encabezado = []
45 if os.path.exists(ruta_mensual):
46     wb_mensual = load_workbook(ruta_mensual)
```

HTML EN POWER BI

MENÚ - PANEL EP



TARJTEAS INFORMATIVAS – SEGUIMIENTO CONTRATOS NUEVOS



MENÚ - PRUEBAS DE ORIENTACIÓN VOCACIONAL



PRUEBAS DE ORIENTACIÓN VOCACIONAL - GENERAL

DIRECCIÓN DE FORMACIÓN PROFESIONAL

BÚSQUEDA REGIONAL BÚSQUEDA CENTRO DE FORMACIÓN MES EJECUTADO

REGIONAL DATOS CENTRO ENERO FEBRERO MARZO ABRIL MAYO

TIPO PRUEBA

CARAEM IPO VISTAS

GENERAL

IPO CARAEM

RESUMEN NACIONAL

Pruebas aplicadas a nivel nacional

12.904

Usuarios únicos evaluados

Total general

PRUEBAS

PRUEBAS EJECUTADAS VS META - REGIONALES			PRUEBAS EJECUTADAS VS META - CENTROS DE FORMACIÓN		
REGIONAL	META	EJECUTADO	REGIONAL	DATOS CENTRO	META EJECUTADO
AMAZONAS	300	11	AMAZONAS 9517 - CENTRO PARA LA BIODIVERSIDAD Y EL TURISMO DEL AMAZONAS	300	8
ANTIOQUIA	4.800	2.202	AMAZONAS SIN CENTRO DE FORMACIÓN		3
ARAUCA	300	19	ANTIOQUIA 9101 - CENTRO DE LOS RECURSOS NATURALES RENOVABLES - LA SALADA	300	547
ATLÁNTICO	1.200	681	ANTIOQUIA 9127 - CENTRO DE FORMACIÓN MINERO AMBIENTAL	300	285
BOLÍVAR	1.200	369	ANTIOQUIA 9201 - CENTRO DEL DISEÑO Y MANUFACTURA DEL CUERO	300	490
BOYACÁ	1.200	222	ANTIOQUIA 9202 - CENTRO DE FORMACIÓN EN DISEÑO, CONFECCIÓN Y MODA	300	11
CALDAS	1.500	687	ANTIOQUIA 9203 - CENTRO PARA EL DESARROLLO DEL HABITAT Y LA CONSTRUCCIÓN	300	20
CAQUETÁ	300	29	ANTIOQUIA 9204 - CENTRO DE TECNOLOGÍA DE LA MANUFACTURA AVANZADA	300	28
CASANARE	300	58	ANTIOQUIA 9205 - CENTRO TECNOLÓGICO DEL MOBILIARIO	300	22
CAUCA	900	261	ANTIOQUIA 9206 - CENTRO TEXTIL Y DE GESTIÓN INDUSTRIAL	300	21
CESAR	900	139	ANTIOQUIA 9301 - CENTRO DE COMERCIO	300	44
Total	35.400	12.904	Total	35.400	12.904

EJECUTADO POR REGIONAL

Regional	Ejecutado
ANTIOQUIA	2360
CUNDINAMARCA	2202
LÓRQUO PORTOBELLO	1367
VALLÉ	1028
CAJAMÉ	717
ATLÁNTICO	687
SANTANDER	681
BOLÍVAR	427
QUINDÍO	369
PASTA	324
CAUCA	264
NORTE DE SANTANDER	261
BOLIVIA	239
BOGOTÁ	222
MAQUILLAS	205
MANIZO	175
COCHIBO	152
HUILA	149
CEJA	149
SAN ANDRÉS	139
GUAY	121
GUAY	95
GUAY	95
CASANARE	62
BUENAVISTA	58
CHOCO	48
CAQUETÁ	40
PASTORAL	29
AMAZONAS	28
VOCACIÓN	19
VALENCIA	11
VALPARAISO	6
VALPARES	3



