

# creacion\_dataset\_entrenamiento

July 11, 2025

## 1 CREACIÓN DE DATASET PARA ENTRENAMIENTO

```
[1]: # INSTALACIÓN DE LIBRERÍAS NECESARIAS
      %pip install pandas
```

```
Requirement already satisfied: pandas in
c:\users\carolina\documents\proyectos_programacion\predestu\.venv\lib\site-
packages (2.3.1)
Requirement already satisfied: numpy>=1.26.0 in
c:\users\carolina\documents\proyectos_programacion\predestu\.venv\lib\site-
packages (from pandas) (2.3.1)
Requirement already satisfied: python-dateutil>=2.8.2 in
c:\users\carolina\documents\proyectos_programacion\predestu\.venv\lib\site-
packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in
c:\users\carolina\documents\proyectos_programacion\predestu\.venv\lib\site-
packages (from pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in
c:\users\carolina\documents\proyectos_programacion\predestu\.venv\lib\site-
packages (from pandas) (2025.2)
Requirement already satisfied: six>=1.5 in
c:\users\carolina\documents\proyectos_programacion\predestu\.venv\lib\site-
packages (from python-dateutil>=2.8.2->pandas) (1.17.0)
Note: you may need to restart the kernel to use updated packages.
```

```
[2]: # IMPORTACIÓN DE LIBRERÍAS
      import pandas as pd
```

```
[59]: # CARGAR DATASETS
      df_estudiantes = pd.read_csv('../fuentes_datos/datasets_finales/CSV/estudiante.
      ↪csv', sep=';')
      df_tutorias = pd.read_csv('../fuentes_datos/datasets_finales/CSV/
      ↪tutorias_completo.csv', sep=';')
      df_cursos_tomados = pd.read_csv('../fuentes_datos/datasets_finales/CSV/
      ↪curso_tomado_completo.csv', sep=';')
      df_reprobaciones = pd.read_csv('../fuentes_datos/datasets_finales/CSV/
      ↪reprobacion_completo.csv', sep=';')
```

```
df_test_acra = pd.read_csv('../fuentes_datos/datasets_finales/CSV/test_acra.
↳csv', sep=';')
```

```
C:\Users\carolina\AppData\Local\Temp\ipykernel_13008\324333355.py:4:
DtypeWarning: Columns (1) have mixed types. Specify dtype option on import or
set low_memory=False.
    df_cursos_tomados =
pd.read_csv('../fuentes_datos/datasets_finales/CSV/curso_tomado_completo.csv',
sep=';')
```

```
[60]: # DATASET CON LA MITAD DE LOS DATOS DE LOS DATAFRAMES
df_tutorias2 = df_tutorias.sample(frac=0.5, random_state=42)
df_cursos_tomados2 = df_cursos_tomados.sample(frac=0.25, random_state=42)
df_reprobaciones2 = df_reprobaciones.sample(frac=0.25, random_state=42)
```

```
[61]: # UNIÓN DEL DATASET ESTUDIANTES CON TEST ACRA
df_union1 = pd.merge(df_estudiantes, df_test_acra, on='correo', how='inner')
```

```
[62]: print("DATASET UNIÓN ESTUDIANTES CON ACRA")
print(df_union1.head(5)) # Muestra los primeros 5 registros
```

DATASET UNIÓN ESTUDIANTES CON ACRA

	id_estudiante	id_escuela	id_plan	anio_ingreso	\
0	14200123	1	1	14	
1	20200147	1	3	20	
2	21200121	1	3	21	
3	19200300	2	7	19	
4	20200152	1	3	20	

	correo	edad	sexo	condicion_estudiante	\
0	alexandra.ajen@unmsm.edu.pe	26	F	2	
1	sergio.alejo@unmsm.edu.pe	21	M	1	
2	jose.apaza2@unmsm.edu.pe	21	M	1	
3	esmeralda.arango@unmsm.edu.pe	21	F	1	
4	anderson.araujo@unmsm.edu.pe	24	M	1	

	lectura_prev_estudio	anotacion_temas_clase	...	\
0	2		2	...
1	2		3	...
2	2		2	...
3	3		3	...
4	1		1	...

	resolucion_conflictos_previos	interaccion_social_estudio	\
0	3	1	
1	3	2	
2	2	2	
3	2	3	

	4	2	2
	motivacion_conocimiento	motivacion_evitar_consecuencias	\
0	3	3	
1	3	2	
2	2	2	
3	3	2	
4	2	3	

  

	adaptacion_estrategias_aprendizaje	dialogo_resolutivo_conflictos	\
0	2	1	
1	4	4	
2	2	2	
3	3	3	
4	2	1	

  

	motivacion_orgullo_personal	control_ambiente_estudio	Unnamed: 63	\
0	2	3	NaN	
1	4	3	NaN	
2	2	2	NaN	
3	3	3	NaN	
4	1	2	NaN	

  

	Unnamed: 64
0	NaN
1	NaN
2	NaN
3	NaN
4	NaN

[5 rows x 69 columns]

```
[63]: n_registros_union1 = df_union1.shape[0]
print(f"Número de registros en df_union1: {n_registros_union1}")
```

Número de registros en df\_union1: 126

```
[64]: # UNIÓN DEL DATASET UNION 1 CON TUTORÍAS
df_union2 = pd.merge(df_union1, df_tutorias2, on='id_estudiante', how='inner')
```

```
[65]: print("DATASET UNION 1 CON TUTORÍAS")
print(df_union2.head(5)) # Muestra los primeros 5 registros
```

	id_estudiante	id_escuela	id_plan	anio_ingreso	\
0	14200123	1	1	14	
1	14200123	1	1	14	
2	14200123	1	1	14	
3	14200123	1	1	14	

4	14200123	1	1	14
---	----------	---	---	----

  

	correo	edad	sexo	condicion_estudiante	\
0	alexandra.ajen@unmsm.edu.pe	26	F	2	
1	alexandra.ajen@unmsm.edu.pe	26	F	2	
2	alexandra.ajen@unmsm.edu.pe	26	F	2	
3	alexandra.ajen@unmsm.edu.pe	26	F	2	
4	alexandra.ajen@unmsm.edu.pe	26	F	2	

  

	lectura_prev_estudio	anotacion_temas_clase	...	\
0	2		2	...
1	2		2	...
2	2		2	...
3	2		2	...
4	2		2	...

  

	adaptacion_estrategias_aprendizaje	dialogo_resolutivo_conflictos	\
0	2	1	
1	2	1	
2	2	1	
3	2	1	
4	2	1	

  

	motivacion_orgullo_personal	control_ambiente_estudio	Unnamed: 63	\
0	2	3	NaN	
1	2	3	NaN	
2	2	3	NaN	
3	2	3	NaN	
4	2	3	NaN	

  

	Unnamed: 64	id_tutoria	id_semestre	tipo_autorizacion	num_res_autoriza
0	NaN	3435.0	30.0	AM	RD 448-D-FISI-30
1	NaN	3386.0	32.0	TO	RD 504-D-FISI-32
2	NaN	3431.0	16.0	AM	RD 323-D-FISI-16
3	NaN	3430.0	8.0	AM	RD 411-D-FISI-8
4	NaN	3381.0	13.0	AM	RD 700-D-FISI-13

[5 rows x 73 columns]

```
[66]: n_registros_union2 = df_union2.shape[0]
print(f"Número de registros en df_union2: {n_registros_union2}")
```

Número de registros en df\_union2: 2209

```
[67]: # UNIÓN DEL DATASET UNION2 CON CURSOS_TOMADOS
df_union3 = pd.merge(df_union2, df_cursos_tomados2, on='id_estudiante',
↳how='inner')
```

```
[68]: print("DATASET UNIÓN2 CON CURSOS_TOMADOS")
print(df_union3.head(5)) # Muestra los primeros 5 registros
```

DATASET UNIÓN2 CON CURSOS\_TOMADOS

	id_estudiante	id_escuela	id_plan	anio_ingreso	\
0	14200123	1	1	14	
1	14200123	1	1	14	
2	14200123	1	1	14	
3	14200123	1	1	14	
4	14200123	1	1	14	

  

	correo	edad	sexo	condicion_estudiante	\
0	alexandra.ajen@unmsm.edu.pe	26	F	2	
1	alexandra.ajen@unmsm.edu.pe	26	F	2	
2	alexandra.ajen@unmsm.edu.pe	26	F	2	
3	alexandra.ajen@unmsm.edu.pe	26	F	2	
4	alexandra.ajen@unmsm.edu.pe	26	F	2	

  

	lectura_prev_estudio	anotacion_temas_clase	...	Unnamed: 64	id_tutoria	\
0	2	2	...	NaN	3435.0	
1	2	2	...	NaN	3435.0	
2	2	2	...	NaN	3435.0	
3	2	2	...	NaN	3435.0	
4	2	2	...	NaN	3435.0	

  

	id_semestre_x	tipo_autorizacion	num_res_autoriza	id_ct	id_curso	\
0	30.0	AM	RD 448-D-FISI-30	488399	202SW0905	
1	30.0	AM	RD 448-D-FISI-30	488352	202W0606	
2	30.0	AM	RD 448-D-FISI-30	488380	202W0801	
3	30.0	AM	RD 448-D-FISI-30	488391	202SW0207	
4	30.0	AM	RD 448-D-FISI-30	488384	INE016	

  

	id_semestre_y	nota	id_ciclo
0	37	10	83
1	27	16	70
2	33	18	72
3	36	19	76
4	34	19	23

[5 rows x 78 columns]

```
[69]: n_registros_union3 = df_union3.shape[0]
print(f"Número de registros en df_union3: {n_registros_union3}")
```

Número de registros en df\_union3: 72347

```
[70]: # UNIÓN DEL DATASET UNION3 CON REPROBACIONES
```

```
df_union4 = pd.merge(df_union3, df_reprobaciones2, on='id_estudiante',
                    how='inner')
```

```
[71]: print("DATASET UNIÓN3 CON REPROBACIONES")
      print(df_union4.head(5)) # Muestra los primeros 5 registros
```

DATASET UNIÓN3 CON REPROBACIONES

	id_estudiante	id_escuela	id_plan	anio_ingreso	\
0	14200123	1	1	14	
1	14200123	1	1	14	
2	14200123	1	1	14	
3	14200123	1	1	14	
4	14200123	1	1	14	

	correo	edad	sexo	condicion_estudiante	\
0	alexandra.ajen@unmsm.edu.pe	26	F	2	
1	alexandra.ajen@unmsm.edu.pe	26	F	2	
2	alexandra.ajen@unmsm.edu.pe	26	F	2	
3	alexandra.ajen@unmsm.edu.pe	26	F	2	
4	alexandra.ajen@unmsm.edu.pe	26	F	2	

	lectura_prev_estudio	anotacion_temas_clase	...	tipo_autorizacion	\
0	2	2	...	AM	
1	2	2	...	AM	
2	2	2	...	AM	
3	2	2	...	AM	
4	2	2	...	AM	

	num_res_autoriza	id_ct	id_curso_x	id_semestre_y	nota	id_ciclo	\
0	RD 448-D-FISI-30	488399	202SW0905	37	10	83	
1	RD 448-D-FISI-30	488399	202SW0905	37	10	83	
2	RD 448-D-FISI-30	488399	202SW0905	37	10	83	
3	RD 448-D-FISI-30	488399	202SW0905	37	10	83	
4	RD 448-D-FISI-30	488399	202SW0905	37	10	83	

	id_curso_y	n_reprobaciones	id_semestre
0	202SW0605	1	9
1	20W0802	1	28
2	202SW0E09	1	40
3	20W0601	1	16
4	202SW0905	1	37

[5 rows x 81 columns]

```
[72]: n_registros_union4 = df_union4.shape[0]
      print(f"Número de registros en df_union4: {n_registros_union4}")
```

Número de registros en df\_union4: 886660

```
[73]: # Agregar columna con el número de tutorías por estudiante, sin perder las
      ↪demás columnas
df_union4["n_tutorias"] = df_union4.groupby("id_estudiante")["id_estudiante"].
      ↪transform("count")
```

```
[74]: n_registros_union4 = df_union4.shape[0]
      print(f"Número de registros en df_union4: {n_registros_union4}")
```

Número de registros en df\_union4: 886660

```
[75]: # MOSTRAR LAS COLUMNAS DEL DATASET FINAL
      print("Columnas del dataset final:")
      df_union4.columns.tolist()
```

Columnas del dataset final:

```
[75]: ['id_estudiante',
      'id_escuela',
      'id_plan',
      'anio_ingreso',
      'correo',
      'edad',
      'sexo',
      'condicion_estudiante',
      'lectura_prev_estudio',
      'anotacion_temas_clase',
      'subrayado_importante',
      'uso_signos_personales',
      'subrayado_memorizacion',
      'anotaciones_marginales',
      'repeticion_datos_relevantes',
      'lectura_en_voz_alta',
      'resumen_mental',
      'autoevaluacion_aprendizaje',
      'preguntas_entre_pares',
      'representacion_visual_ideas',
      'reorganizacion_personal_contenido',
      'transferencia_conocimiento',
      'discusion_colaborativa_contenido',
      'analogias_conocimiento_previos',
      'aplicacion_laboral_contenido',
      'preguntas_durante_clase',
      'preguntas_previas_estudio',
      'aprendizaje_palabras_propias',
      'obtencion_conclusiones',
      'clasificacion_personal_datos',
      'resumen_por_apartados',
      'resumen_post_tema',
```

'resumen\_basado\_subrayado',  
 'organizacion\_criterio\_logico',  
 'secuencia\_temporal\_historica',  
 'diagramas\_procedimentales',  
 'mapas\_conceptuales\_post\_estudio',  
 'diagramas\_en\_v',  
 'memorizacion\_ubicacion\_espacial',  
 'identificacion\_estructura\_texto',  
 'aplicacion\_practica\_contenido',  
 'recuerdo\_estructuras\_previas',  
 'recuperacion\_por\_asociacion',  
 'aprovechamiento\_observaciones\_profesor',  
 'esquema\_previo\_redaccion',  
 'cuidado\_presentacion\_examen',  
 'analisis\_datos\_previos',  
 'preparacion\_mental\_mensaje',  
 'planificacion\_solucion\_problemas',  
 'uso\_datos\_secundarios',  
 'conciencia\_estrategias\_memoria',  
 'uso\_herramientas\_organizacion',  
 'registro\_tareas\_asignatura',  
 'gestion\_tiempo\_estudio',  
 'planificacion\_estrategias\_examen',  
 'monitoreo\_estrategias\_estudio',  
 'evaluacion\_estrategias\_examen',  
 'autoregulación\_emocional\_examen',  
 'visualizacion\_para\_calma',  
 'resolucion\_conflictos\_previos',  
 'interaccion\_social\_estudio',  
 'motivacion\_conocimiento',  
 'motivacion\_evitar\_consecuencias',  
 'adaptacion\_estrategias\_aprendizaje',  
 'dialogo\_resolutivo\_conflictos',  
 'motivacion\_orgullo\_personal',  
 'control\_ambiente\_estudio',  
 'Unnamed: 63',  
 'Unnamed: 64',  
 'id\_tutoria',  
 'id\_semestre\_x',  
 'tipo\_autorizacion',  
 'num\_res\_autoriza',  
 'id\_ct',  
 'id\_curso\_x',  
 'id\_semestre\_y',  
 'nota',  
 'id\_ciclo',  
 'id\_curso\_y',



```
'n_reprobaciones',  
'id_semestre',  
'n_tutorias']
```

```
[76]: # ELIMINACIÓN DE COLUMNAS NO NECESARIAS  
df_union4.drop(columns=['id_estudiante', 'anio_ingreso', 'correo', 'edad',  
↳ 'sexo', 'Unnamed: 63', 'Unnamed: 64', 'id_tutoria', 'id_semestre_x',  
↳ 'tipo_autorizacion', 'num_res_autoriza', 'id_ct', 'id_curso_x',  
↳ 'id_semestre_y', 'id_curso_y'], inplace=True)
```

```
[77]: # MOSTRAR LAS COLUMNAS DEL DATASET FINAL  
print("Columnas del dataset final:")  
df_union4.columns.tolist()
```

Columnas del dataset final:

```
[77]: ['id_escuela',  
      'id_plan',  
      'condicion_estudiante',  
      'lectura_prev_estudio',  
      'anotacion_temas_clase',  
      'subrayado_importante',  
      'uso_signos_personales',  
      'subrayado_memorizacion',  
      'anotaciones_marginales',  
      'repeticion_datos_relevantes',  
      'lectura_en_voz_alta',  
      'resumen_mental',  
      'autoevaluacion_aprendizaje',  
      'preguntas_entre_pares',  
      'representacion_visual_ideas',  
      'reorganizacion_personal_contenido',  
      'transferencia_conocimiento',  
      'discusion_colaborativa_contenido',  
      'analogias_conocimiento_previos',  
      'aplicacion_laboral_contenido',  
      'preguntas_durante_clase',  
      'preguntas_previas_estudio',  
      'aprendizaje_palabras_propias',  
      'obtencion_conclusiones',  
      'clasificacion_personal_datos',  
      'resumen_por_apartados',  
      'resumen_post_tema',  
      'resumen_basado_subrayado',  
      'organizacion_criterio_logico',  
      'secuencia_temporal_historica',  
      'diagramas_procedimentales',  
      'mapas_conceptuales_post_estudio',
```

```

'diagramas_en_v',
'memorizacion_ubicacion_espacial',
'identificacion_estructura_texto',
'aplicacion_practica_contenido',
'recuerdo_estructuras_previas',
'recuperacion_por_asociacion',
'aprovechamiento_observaciones_profesor',
'esquema_previo_redaccion',
'cuidado_presentacion_examen',
'analisis_datos_previos',
'preparacion_mental_mensaje',
'planificacion_solucion_problemas',
'uso_datos_secundarios',
'conciencia_estrategias_memoria',
'uso_herramientas_organizacion',
'registro_tareas_asignatura',
'gestion_tiempo_estudio',
'planificacion_estrategias_examen',
'monitoreo_estrategias_estudio',
'evaluacion_estrategias_examen',
'autoregulación_emocional_examen',
'visualizacion_para_calma',
'resolucion_conflictos_previos',
'interaccion_social_estudio',
'motivacion_conocimiento',
'motivacion_evitar_consecuencias',
'adaptacion_estrategias_aprendizaje',
'dialogo_resolutivo_conflictos',
'motivacion_orgullo_personal',
'control_ambiente_estudio',
'nota',
'id_ciclo',
'n_reprobaciones',
'id_semestre',
'n_tutorias']

```

[78]: # CLASIFICACIÓN DEL RIESGO ACADÉMICO

```

def clasificar_riesgo(nota):
    if nota < 10:
        return 0 # Muy bajo
    elif nota < 12:
        return 1 # Bajo
    elif nota < 15:
        return 2 # Medio
    elif nota < 18:
        return 3 # Bueno
    else:

```

```
return 4 # Muy bueno
```

```
[79]: df_union5 = df_union4.copy()
```

```
[80]: df_union5['riesgo_academico'] = df_union5['nota'].apply(clasificar_riesgo)
```

```
[81]: # CONVERSIÓN DE N_TUTORIAS A BINARIAS (0 = No, 1 = Sí)
df_union5['n_tutorias'] = df_union5['n_tutorias'].apply(lambda x: 1 if x > 0
↳ else 0)
```

```
[82]: # CONVERSIÓN DE ESTRATEGIAS ACRA A USO_ACRA_ACTIVADO (NUMÉRICO)
acra_cols = [
    'condicion_estudiante', 'lectura_prev_estudio', 'anotacion_temas_clase',
    'subrayado_importante', 'uso_signos_personales', 'subrayado_memorizacion',
    'anotaciones_marginales', 'repeticion_datos_relevantes', 'lectura_en_voz_alta',
    'resumen_mental', 'autoevaluacion_aprendizaje', 'preguntas_entre_pares',
    'representacion_visual_ideas', 'reorganizacion_personal_contenido',
↳ 'transferencia_conocimiento',
    'discusion_colaborativa_contenido', 'analogias_conocimiento_previos',
↳ 'aplicacion_laboral_contenido',
    'preguntas_durante_clase', 'preguntas_previas_estudio',
↳ 'aprendizaje_palabras_propias',
    'obtencion_conclusiones', 'clasificacion_personal_datos',
↳ 'resumen_por_apartados',
    'resumen_post_tema', 'resumen_basado_subrayado',
↳ 'organizacion_criterio_logico',
    'secuencia_temporal_historica', 'diagramas_procedimentales',
↳ 'mapas_conceptuales_post_estudio',
    'diagramas_en_v', 'memorizacion_ubicacion_espacial',
↳ 'identificacion_estructura_texto',
    'aplicacion_practica_contenido', 'recuerdo_estructuras_previas',
↳ 'recuperacion_por_asociacion',
    'aprovechamiento_observaciones_profesor', 'esquema_previo_redaccion',
↳ 'cuidado_presentacion_examen',
    'analisis_datos_previos', 'preparacion_mental_mensaje',
↳ 'planificacion_solucion_problemas',
    'uso_datos_secundarios', 'conciencia_estrategias_memoria',
↳ 'uso_herramientas_organizacion',
    'registro_tareas_asignatura', 'gestion_tiempo_estudio',
↳ 'planificacion_estrategias_examen',
    'monitoreo_estrategias_estudio', 'evaluacion_estrategias_examen',
↳ 'autoregulación_emocional_examen',
    'visualizacion_para_calma', 'resolucion_conflictos_previos',
↳ 'interaccion_social_estudio',
    'motivacion_conocimiento', 'motivacion_evitar_consecuencias',
↳ 'adaptacion_estrategias_aprendizaje',
```

```

'dialogo_resolutivo_conflictos', 'motivacion_orgullo_personal',
↪ 'control_ambiente_estudio'
]

df_union5['uso_acra_total'] = df_union5[acra_cols].sum(axis=1)

```

```

[83]: # CONVERSIÓN DE uso_acra_total A BINARIAS (0 = SI ES MENOR A 147.5, 1 = SI ES
↪ IGUAL O MAYOR A 147.5)
df_union5['uso_acra'] = df_union5['uso_acra_total'].apply(lambda x: 1 if x >=
↪ 147.5 else 0)

```

```

[84]: # MOSTRAR LAS COLUMNAS DEL DATASET FINAL
print("Columnas del dataset final:")
df_union5.columns.tolist()

```

Columnas del dataset final:

```

[84]: ['id_escuela',
'id_plan',
'condicion_estudiante',
'lectura_prev_estudio',
'anotacion_temas_clase',
'subrayado_importante',
'uso_signos_personales',
'subrayado_memorizacion',
'anotaciones_marginales',
'repeticion_datos_relevantes',
'lectura_en_voz_alta',
'resumen_mental',
'autoevaluacion_aprendizaje',
'preguntas_entre_pares',
'representacion_visual_ideas',
'reorganizacion_personal_contenido',
'transferencia_conocimiento',
'discusion_colaborativa_contenido',
'analogias_conocimiento_previos',
'aplicacion_laboral_contenido',
'preguntas_durante_clase',
'preguntas_previas_estudio',
'aprendizaje_palabras_propias',
'obtencion_conclusiones',
'clasificacion_personal_datos',
'resumen_por_apartados',
'resumen_post_tema',
'resumen_basado_subrayado',
'organizacion_criterio_logico',
'secuencia_temporal_historica',

```

```

'diagramas_procedimentales',
'mapas_conceptuales_post_estudio',
'diagramas_en_v',
'memorizacion_ubicacion_espacial',
'identificacion_estructura_texto',
'aplicacion_practica_contenido',
'recuerdo_estructuras_previas',
'recuperacion_por_asociacion',
'aprovechamiento_observaciones_profesor',
'esquema_previo_redaccion',
'cuidado_presentacion_examen',
'analisis_datos_previos',
'preparacion_mental_mensaje',
'planificacion_solucion_problemas',
'uso_datos_secundarios',
'conciencia_estrategias_memoria',
'uso_herramientas_organizacion',
'registro_tareas_asignatura',
'gestion_tiempo_estudio',
'planificacion_estrategias_examen',
'monitoreo_estrategias_estudio',
'evaluacion_estrategias_examen',
'autoregulación_emocional_examen',
'visualizacion_para_calma',
'resolucion_conflictos_previos',
'interaccion_social_estudio',
'motivacion_conocimiento',
'motivacion_evitar_consecuencias',
'adaptacion_estrategias_aprendizaje',
'dialogo_resolutivo_conflictos',
'motivacion_orgullo_personal',
'control_ambiente_estudio',
'nota',
'id_ciclo',
'n_reprobaciones',
'id_semestre',
'n_tutorias',
'riesgo_academico',
'uso_acra_total',
'uso_acra']

```

```
[85]: df_union6 = df_union5.copy()
```

```
[86]: # ELIMINACIÓN DE COLUMNAS NO NECESARIAS
df_union6.drop(columns=acra_cols, inplace=True)
```

```
[87]: # MOSTRAR LAS COLUMNAS DEL DATASET FINAL
print("Columnas del dataset final:")
df_union6.columns.tolist()
```

Columnas del dataset final:

```
[87]: ['id_escuela',
       'id_plan',
       'nota',
       'id_ciclo',
       'n_reprobaciones',
       'id_semestre',
       'n_tutorias',
       'riesgo_academico',
       'uso_acra_total',
       'uso_acra']
```

```
[88]: # ELIMINACIÓN DE COLUMNAS NO NECESARIAS
df_union6.drop(columns=['uso_acra_total', 'nota'], inplace=True)
```

```
[89]: print("DATASET UNIÓN6")
print(df_union6.head(5)) # Muestra los primeros 5 registros
```

DATASET UNIÓN6

	id_escuela	id_plan	id_ciclo	n_reprobaciones	id_semestre	n_tutorias	\
0	1	1	83	1	9	1	
1	1	1	83	1	28	1	
2	1	1	83	1	40	1	
3	1	1	83	1	16	1	
4	1	1	83	1	37	1	

	riesgo_academico	uso_acra
0	1	0
1	1	0
2	1	0
3	1	0
4	1	0

```
[90]: # RENOMABRAR COLUMNA N_TUTORIAS A TUTORIAS
df_union6.rename(columns={'n_tutorias': 'tutorias'}, inplace=True)
```

```
[91]: # GUARDAR EL DATASET NUEVO PARA ENTRENAMIENTO
df_union6.to_csv('../fuentes_datos/datasets_finales/CSV/dataset_entrenamiento.
↪ csv', index=False, sep=';')
```

```
[92]: # MOSTRAR LAS COLUMNAS DEL DATASET FINAL
print("Columnas del dataset final:")
df_union6.columns.tolist()
```

Columnas del dataset final:

```
[92]: ['id_escuela',  
       'id_plan',  
       'id_ciclo',  
       'n_reprobaciones',  
       'id_semestre',  
       'tutorias',  
       'riesgo_academico',  
       'uso_acra']
```