



DATOS PERSONALES

TIPO DE PETICIONARIO: PERSONA NATURAL

TIPO DE IDENTIFICACIÓN: CÉDULA DE CIUDADANÍA

NÚMERO DE IDENTIFICACIÓN: 1010244119

NOMBRES: JUAN DIEGO

APELLIDOS: BARON RODRIGUEZ

EMAIL: JBARON@SERVICIOCIVIL.GOV.CO

TELEFONO CELULAR: 6013680038

DIRECCIÓN: CARRERA 30 NO. 25 - 90

MEDIO ENVIO RESPUESTA: CORREO ELECTRÓNICO

PAÍS: COLOMBIA

DEPARTAMENTO: BOGOTÁ, D.C. CIUDAD: BOGOTÁ, D.C.

DATOS DE LA SOLICITUD

TRÁMITE: CONSULTA

DESCRIPCIÓN: INFORME DE SUPERVISIÓN MES DE OCTUBRE CPS-P-031-2025 JUAN DIEGO BARON RODRIGUEZ

Doctora

Gina Paola Silva

Subdirectora Técnica de Desarrollo Organizacional y Empleo Público

Supervisora Contrato CPS-AG-031-2025

ASUNTO: Radicación cuenta de cobro No 8 de CPS-AG-031-2025

Respetada Doctora Paola:

Por medio del presente hago entrega del informe No. 8 y cuenta de cobro correspondiente al periodo comprendido entre el: 1 del mes de octubre y 31 de octubre de la vigencia 2025.

Cordialmente,

Nombre: Juan Diego Barón Rodríguez

Cedula de ciudadanía No. 1010244119

A handwritten signature in black ink, appearing to read 'Juan Barón', with a stylized, cursive script.

Firma



ALCALDÍA MAYOR
DE BOGOTÁ D.C.
GESTIÓN PÚBLICA
Departamento Administrativo del Servicio Civil

MACROPROCESOS DE APOYO
PROCESO DE APOYO A LA GESTIÓN
PROCESO GESTIÓN CONTRACTUAL
FORMATO INFORME DEL CONTRATISTA - SUPERVISOR
CÓDIGO: A-CON-FM-020
VERSIÓN: 13.0
VIGENCIA: JUNIO DE 2025



1. DATOS GENERALES

CONTRATO No.	CPS-AG-031-2025	FECHA DEL CONTRATO:	03/03/2025
NOMBRE DEL/LA SUPERVISOR(A):	Gina Paola Silva Velasquez	Nº CEDULA:	1026250419
NOMBRE IDENTITARIO DEL/LA SUPERVISOR(A):	Subdirectora Técnica de Desarrollo Organizacional y Empleo Público		
CARGO DEL/LA SUPERVISOR(A):	Juan Diego Baron Rodriguez	CEDULA/NIT:	1010244119
NOMBRE DEL/LA CONTRATISTA:			
NOMBRE IDENTITARIO			
OBJETO:	Prestar servicios de apoyo a la gestión, con autonomía técnica y administrativa, para la optimización de la gestión de la Subdirección Técnica de Desarrollo Organizacional y Empleo Público mediante la automatización de tareas, el procesamiento de datos con herramientas de análisis e Inteligencia Artificial (IA).		
VALOR INICIAL:	\$		22.100.000,00
VALOR ADICIÓN (Si Aplica):	\$		10.880.000,00
VALOR TOTAL:	\$		32.980.000,00
PLAZO:	6,0 MESES	15,0 DIAS	
PRORROGA (Si Aplica):	3 meses y 6 días		ADICIÓN
SUSPENSIONES (Fecha Inicial-Final):	00/00/0000 a 00/00/0000	00/00/0000 a 00/00/0000	
FECHA DE ACTA DE INICIO:	11/03/2025		
FECHA FINALIZACIÓN CONTRATO:	31/12/2025		
FECHA INFORME	01/11/2025		
PERIODO INFORME	01/10/2025 a 31/10/2025		
% AVANCE CONTRATO	78,0%	Avance físico acumulado del contrato	
CRP	79 de 2025		

2. FUENTE DE FINANCIACIÓN

TIPO DE INFORME:

PERSONA NATURAL
MENSUAL

x

x

PERSONA JURIDICA
PARCIAL

FINAL

Proceso al que pertenece (marcar con una x) se puede marcar mas de uno si se requiere:

Gerencia Estratégica

Sistemas de Gestión

Gestión del Conocimiento

Gestión del Talento Humano

Seguridad de la Información

Gestión de la Comunicación

Atención al Ciudadano

Organización del Trabajo

Bienestar, Desarrollo y Medición del Rendimiento

Gestión de la Información del Talento Humano

Asesoría y Conceptualización Jurídica

Gestión TIC's

Gestión de Recursos Físicos y Ambientales

Gestión Documental

x

Gestión Financiera

Gestión Contractual

Gestión Jurídica

Control y Seguimiento

Control Disciplinario Interno

* Si aplica

Código del proyecto de inversión que financia:

7971

* Si aplica - Cuando es con recursos de inversión

3. ESTADO FINANCIERO DEL CONTRATO

VALOR TOTAL DEL CONTRATO	\$	32.980.000,00
SUMA DE PAGOS EFECTUADOS	\$	22.666.666,00
FALTANTE POR PAGAR	\$	10.313.334,00
VALOR A PAGAR EN ESTE PERIODO	\$	3.400.000,00
NUMERO DEL PAGO A REALIZAR	8	



SC-CER431153



CO-SC-CER431153

4. GESTIÓN DEL CONTRATISTA

* ¿El/La contratista debe hacer entrega de productos?

SI

NO

x

Si la respuesta es afirmativa por favor detalle a continuación los productos que deben ser entregados por el/la contratista y los mismos fueron entregados a satisfacción, de lo contrario diligencie los campos relacionados con las obligaciones en el contrato.

ENTREGA A
SATISFACCIÓN

N°	PRODUCTO	%AVANCE	SI	NO
1	"Por favor ajustar el tamaño de la fila antes de imprimir, para que se pueda verificar toda la información contenida"			

* Si alguno de los productos no es entregado a satisfacción, por favor especifique en el campo de observaciones el motivo.

OBLIGACIONES		
N°	OBLIGACIÓN	DESCRIPCIÓN
1	Contribuir a la comprensión de los datos recopilados por la Subdirección Técnica de Desarrollo Organizacional y Empleo Público, a través del uso de técnicas de Inteligencia Artificial para generar análisis descriptivos, predictivos y prospectivos que impulsen el cumplimiento las metas a cargo de la dependencia	Para esta obligación, se ejecutó un modelo de procesamiento de lenguaje natural en Python basado en BETO-BERT, con el propósito de clasificar los contratos correspondientes al año 2025, optimizando así la identificación de patrones y categorías relevantes dentro de la información contractual.
2	Desarrollar y organizar bases de datos para los modelos de información del proceso de Organización del Trabajo. La Inteligencia Artificial será la base para implementar funciones de creación, lectura, actualización y borrado de datos, optimizando la gestión de la información	En cumplimiento de esta obligación, se recolectó y depuró la base de datos de contratos correspondientes al año 2025 hasta el mes de agosto, garantizando la calidad y consistencia de los registros. A partir de este insumo, se obtuvo una base de datos resultante del análisis realizado por el modelo BETO-BERT, la cual fue consolidada y sometida a revisión inicial para su análisis e interpretación posterior.
3	Apoyar a la Subdirección Técnica de Desarrollo Organizacional y Empleo Público, en el uso y generación de herramientas de Inteligencia Artificial para el procesamiento y visualización de datos, que sirvan de insumos para el cumplimiento de las funciones	En el marco de esta obligación, se afinó y reentrenó el modelo BETO-BERT utilizando los datos correspondientes al año 2025, con el objetivo de mejorar su capacidad de clasificación y comprensión semántica de los textos contractuales. Este proceso de ajuste fino (fine-tuning) permitió adaptar el modelo al contexto específico de la contratación distrital, aumentando la precisión en la identificación de tipologías contractuales, entidades contratantes y temáticas recurrentes. Adicionalmente, se implementaron herramientas de visualización interactivas que permiten explorar los resultados del modelo de forma dinámica, facilitando la interpretación de patrones y tendencias mediante paneles gráficos y dashboards de análisis descriptivo y predictivo.
4	Crear guías de usuario para los modelos desarrollados, que facilitarán el uso y la comprensión de las herramientas por parte de los usuarios.	En cumplimiento de esta obligación, se actualizó la guía de usuario del modelo BETO-BERT, incorporando las mejoras derivadas del proceso de afinamiento y entrenamiento con los datos de 2025. La nueva versión de la guía incluye instrucciones detalladas para la ejecución del modelo, interpretación de resultados, mantenimiento de la base de datos y visualización de la información procesada, garantizando así una mayor autonomía y comprensión por parte de los usuarios técnicos y analíticos de la Subdirección Técnica de Desarrollo Organizacional y Empleo Público.

5	Mantener un registro detallado del código de software empleado en el desarrollo de los módulos del proceso de Organización del Trabajo, de acuerdo con los procedimientos establecidos por el DASC para la documentación, cuando sea pertinente.	En el marco de esta obligación, se documentó, registró y almacenó el código fuente utilizado en el desarrollo de los modelos y procesos asociados al análisis de datos del año 2025, siguiendo las directrices establecidas por el DASC. La documentación incluye comentarios técnicos, versiones del código, dependencias, estructura de carpetas y procedimientos de ejecución, asegurando la trazabilidad, reproducibilidad y mantenimiento del software empleado en el proceso de Organización del Trabajo.
---	--	---

¿Adjunta Anexo?	CD		Documento	x
-----------------	----	--	-----------	---

5. DETALLE DE LA INFORMACIÓN REGISTRADA EN LA PLANILLA Y APORTES A SEGURIDAD SOCIAL

	NOMBRE ADMINISTRADORA	Nº DE PLANILLA	FECHA DE PAGO	MES DE APORTE
PAGO APORTES SALUD	Sanitas	90598024	14/10/2025	Octubre
PAGO APORTES PENSIÓN	Protección	90598024	14/10/2025	Octubre
PAGO APORTE ARL	Positiva	90598024	14/10/2025	Octubre

PERIODO CERTIFICADO					ARL	Clase de Riesgo
FECHA INICIAL	FECHA FINAL	DÍAS CERTIFICADOS	VALOR MENSUAL	PAGO SALUD FECHA DE PAGO	PAGO PENSIÓN FECHA DE PAGO	FECHA DE PAGO
01/10/2025	31/10/2025	30	3400000	14/10/2025 12,5%	14/10/2025 16,0%	14/10/2025 0,52%
				177900	227800	7400

* Recuerde que la planilla puede registrar un valor superior por razones de mora.

* SMLMV \$ 1.423.500,00

* Fondo de solidaridad NO (1% del valor base a cotizar en seguridad social cuando supera los 4 smlmv)

Valor Fondo solidaridad

6. SOPORTES DOCUMENTALES ENTREGADOS POR EL CONTRATISTA

Nº	DOCUMENTO	ENTREGA SI NO
1	Adjunta CD de las evidencias de las actividades desarrolladas en el periodo	x
2	Adjunta documentos impresos de las evidencias de las actividades desarrolladas en el periodo	x
3	Adjunta copia de la planilla de aportes a seguridad social por concepto del pago de (pensión, salud y ARL).	x
4	Se encuentra(n) vigente(s) la(s) garantía(s) suscrita(s) entre el/la contratista y el DASC.	x
5	En el caso de personas jurídicas anexa la certificación de pago de seguridad social y parafiscales por revisor fiscal y/o representante legal.	x

* El/la supervisor/a debe validar la información de la planilla con el siguiente campo.

7. NOVEDADES, SITUACIONES O INCONVENIENTES

* ¿Sucedio alguna novedad, situación o inconveniente durante la ejecución del contrato?

SI

NO

x

Si la respuesta es afirmativa por favor detalle las novedades, situaciones o inconvenientes suscitados en el campos observaciones del presente numeral.

OBSERVACIONES

De conformidad con la normatividad vigente, se adjunta la planilla de seguridad social del mes de octubre 2025.

8. FIRMAS

DATOS DEL SUPERVISOR(A)		FIRMA
NOMBRE:	Gina Paola Silva Velasquez	
CEDULA	1026250419	
CARGO	Subdirectora Técnica de Desarrollo Organizacional y Empleo Público	

DECLARACIÓN : Como supervisor(a) certifico que el/la contratista cumple con las obligaciones pactadas en el contrato a satisfacción y las mismas, cumplieron los fines perseguidos con la contratación, por lo cual debe procederse al pago.

Nota: Si conforme al plazo de ejecución y sus modificaciones es el último informe, proceda a evaluar el desempeño proveedor o contratista.

9. EVALUACIÓN DEL DESEMPEÑO DEL/LA PROVEEDOR O CONTRATISTA

Marque con una "X" el desempeño del/la proveedor/a o contratista, en una escala de 1 a 3, donde 1 es Malo, 2 es Regular y 3 Bueno.

1

2

3

DATOS DEL /LA CONTRATISTA		FIRMA
NOMBRE:	Juan Diego Baron Rodriguez	
NOMBRE IDENTITARIO:	(Si se identifica con un nombre diferente al documento de identidad)	
CEDULA:	1010244119	

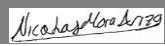
NOTA: Firmar el formato solo en el caso de personas naturales.

REVISADO

NOMBRE:

Jhoan Nicolas Mora Ariza

V.B



DATOS DEL APORTANTE						
TIPO	NÚMERO	NOMBRE APORTANTE		DIRECCIÓN	TELÉFONO	CORREO
CC	1010244119	JUAN DIEGO BARÓN RODRIGUEZ		carrera 34 n 1a 57	3022998353	jbaron@serviciocivil.gov.co
FORMA PRESENTACIÓN	CLASE APORTANTE	NOMBRE SUCURSAL	CÓDIGO	DEPARTAMENTO	CIUDAD / MUNICIPIO	
ÚNICA	I – Independiente			BOGOTÁ D. C.	BOGOTÁ, D.C.	
EXONERADO PAGO PARAFISCALES Y SALUD						NO

DATOS DE LA PLANILLA					
PLANILLA ASOCIADA	FECHA PAGO ASOCIADA (DÍA/MES/AÑO)	TIPO PLANILLA	FECHA PAGO (DÍA/MES/AÑO)	NÚMERO PLANILLA	CANTIDAD
					EMPLEADOS UPC
PERIODO SALUD	PERIODO PENSIONES				1 0
2025-10	2025-10	I	14/10/2025	90598024	\$413.300

TOTALES POR SUBSISTEMAS

TOTALES SALUD													
Código EPS	Nombre	NIT	Cotización Obligatoria	UPC Adicional	Incapacidades		Licencia Maternidad		Días Mora	Valor Mora Cotización	Valor Mora UPC	Total a Pagar	No. Afiliados
					No. Autorización	Valor	No. Autorización	Valor					
EPS005	Sanitas EPS	800251440-6	178.000	0		0		0	0	0	0	178.000	1

TOTALES PENSIÓN												
Código AFP	Nombre	NIT	Cotización Obligatoria	Aporte Voluntario Afiliado	Aporte Voluntario Aportante	Aporte FSP - Solidaridad	Aporte FSP - Subsistencia	Días Mora	Valor Mora Cotización	Valor Mora FSP	Total a Pagar	No. Afiliados
230201	Proteccion (ING + Proteccion)	800229739-0	227.800	0	0	0	0	0	0		227.800	1

TOTALES RIESGOS LABORALES															
Código ARL	Nombre	NIT	Cotización Obligatoria	Incapacidades		Aportes Otros Sistemas	Valor Neto Cotización	Días Mora	Valor Mora Cotización	Subtotal Cotización	No. Radicado Saldo a Favor	Valor Saldo a Favor	Fondo Solidaridad	Total a Pagar	No. Afiliados
				No. Autorización	Valor										
14-23	Positiva Seguros	860011153-6	7.500				7.500	0	0	7.500			75	7.500	1

TOTALES CAJAS							
Código CCF	Nombre	NIT	Valor Aporte	Días Mora	Valor Mora Aporte	Total a Pagar	No. Afiliados

TOTALES PARAFISCALES				
Valor Aporte	Días Mora	Valor Mora Aporte	Total a Pagar	No. Afiliados
SENA				
0	0	0	0	0
ICBF				
0	0	0	0	0
ESAP				
MEN				

TOTALES POR SUBSISTEMA			
Tipo Administradora	No. Administradoras Reportadas	Valor antes de IGE, LMA, IRP y Mora	Total a Pagar
Salud	1	178.000	178.000
Pensión	1	227.800	227.800
Riesgos Laborales	1	7.500	7.500
CCF	0	0	0
ESAP	0	0	0
ICBF	0	0	0
MEN	0	0	0
SENA	0	0	0
TOTALES	3	413.300	413.300

DATOS DEL APORTANTE						
TIPO	NÚMERO	NOMBRE APORTANTE		DIRECCIÓN	TELÉFONO	CORREO
CC	1010244119	JUAN DIEGO BARÓN RODRIGUEZ		carrera 34 n 1a 57	3022998353	jbaron@serviciocivil.gov.co
FORMA PRESENTACIÓN	CLASE APORTANTE	NOMBRE SUCURSAL	CÓDIGO	DEPARTAMENTO	CIUDAD / MUNICIPIO	
ÚNICA	I – Independiente			BOGOTÁ D. C.	BOGOTÁ, D.C.	
						EXONERADO PAGO PARAFISCALES Y SALUD
						NO

DATOS DE LA PLANILLA					
PLANILLA ASOCIADA	FECHA PAGO ASOCIADA (DIA/MES/AÑO)	TIPO PLANILLA	FECHA PAGO (DIA/MES/AÑO)	NÚMERO PLANILLA	CANTIDAD
					EMPLADOS
PERIODO SALUD	PERIODO PENSIONES				UPC
2025-10	2025-10	I	14/10/2025	90598024	\$413.300

DETALLE POR COTIZANTE																																														
INFORMACIÓN COTIZANTE										INFORMACIÓN NOVEDADES							PENSIÓN							SALUD			RIESGOS LABORALES			CCF			PARAFISCALES													
No.	Tipo	No. de Identificación	Apellidos y Nombres	Cotizante	Subjeto	Exonerado	Clasificación	Exonerado	ING	RET	TDE	TAE	TDP	TAP	VBP	VBT	SILN	IDE	MA	MA	MA	MA	MA	MA	MA	Cód. EPS	IBC EPS	Cotización / Valor UPC	Cód. ARL	IBC ARL	Clase de Riesgo	Cotización	Código CCF	IBC CCF	Aporte CCF	IBC otros parafiscales	Aporte SENA	Aporte ICBF	Aporte ESAP	Aporte MEN						
1	CC	1010244119	BARON RODRIGUEZ JUAN DIEGO	59	0		N																			230201	1.423.500	227.800	0	0	0	0	EPS005	1.423.500	178.000	14-23	1.423.500	1	7.500		0	0	0	0	0	0

PAGADA

Evidencias obligaciones octubre 2025

Contribuir a la comprensión de los datos recopilados por la Subdirección Técnica de Desarrollo Organizacional y Empleo Público, a través del uso de técnicas de Inteligencia Artificial para generar análisis descriptivos, predictivos y prospectivos que impulsen el cumplimiento las metas a cargo de la dependencia

```
[ ] # ✅ Importar librerías
import pandas as pd
import torch
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import classification_report, confusion_matrix
from transformers import AutoTokenizer, AutoModelForSequenceClassification, Trainer, TrainingArgument
from datasets import Dataset, DatasetDict
import numpy as np

[ ] # ✅ Entrenamiento optimizado
training_args = TrainingArguments(
    output_dir='./results',
    evaluation_strategy='epoch',
    per_device_train_batch_size=16,
    per_device_eval_batch_size=16,
    num_train_epochs=2,
    fp16=True,
    weight_decay=0.01,
    logging_steps=10,
    save_total_limit=1
)

/usr/local/lib/python3.11/dist-packages/transformers/training_args.py:1611: FutureWarning: `evaluat
warnings.warn(

[ ] # ✅ Cargar los datos en Google Colab
from google.colab import files
import pandas as pd

# Subir archivo manualmente
uploaded = files.upload()
```

```
# ✅ Codificar etiquetas
label_encoder = LabelEncoder()
df['label'] = label_encoder.fit_transform(df['clasificacion_del_contrato'])
num_labels = len(label_encoder.classes_)
```

```
# ✅ División train/val
train_texts, val_texts, train_labels, val_labels = train_test_split(
    df['objeto'].tolist(), df['label'].tolist(),
    test_size=0.1, stratify=df['label'], random_state=42
)
```

```
▶ # ✅ Tokenizador y datasets
model_name = 'dccuchile/bert-base-spanish-wwm-cased'
tokenizer = AutoTokenizer.from_pretrained(model_name)

def tokenize(batch):
    return tokenizer(batch['text'], truncation=True, padding='max_length', max_length=128)

train_ds = Dataset.from_dict({'text': train_texts, 'label': train_labels})
val_ds = Dataset.from_dict({'text': val_texts, 'label': val_labels})
ds = DatasetDict({'train': train_ds, 'validation': val_ds})
ds = ds.map(tokenize, batched=True)
ds = ds.remove_columns(['text'])
ds.set_format('torch')
```

🔄 /usr/local/lib/python3.11/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF\_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (<https://huggingface.co/settings/tokens>)
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets

warnings.warn(
tokenizer_config.json: 100% ██████████ 364/364 [00:00<00:00, 38.1kB/s]
config.json: 100% ██████████ 648/648 [00:00<00:00, 28.0kB/s]
vocab.txt: 100% ██████████ 242k/242k [00:00<00:00, 2.67MB/s]
tokenizer.json: 100% ██████████ 480k/480k [00:00<00:00, 5.68MB/s]

☒ Modelo

```
model = AutoModelForSequenceClassification.from_pretrained(model_name, num_labels=num_labels)
```

```
pytorch_model.bin: 100% ██████████ 440M/440M [00:03<00:00, 121MB/s]
```

model.safetensors: 100% 440M/440M [00:04<00:00, 143MB/s]

Some weights of BertForSequenceClassification were not initialized from the model checkpoint at dcc. You should probably TRAIN this model on a down-stream task to be able to use it for predictions and

☒ Entrenamiento optimizado

```
training_args = TrainingArguments(
    output_dir='./results',
    evaluation_strategy='epoch',
    per_device_train_batch_size=16,
    per_device_eval_batch_size=16,
    num_train_epochs=2,
    fp16=True,
    weight_decay=0.01,
    logging_steps=10,
    save_total_limit=1
```

```
)
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=ds['train'],
    eval_dataset=ds['validation'],
)
```

```
/usr/local/lib/python3.11/dist-packages/transformers/training_args.py:1611: FutureWarning: `evaluate` is deprecated and will be removed in a future version. Use `evaluator.evaluate` instead.
  warnings.warn(
```

```
# Ejecutar entrenamiento
trainer.train()

wandb: WARNING The `run_name` is currently set to the same value as `TrainingArguments.output_dir`.
wandb: Using wandb-core as the SDK backend. Please refer to https://wandb.me/wandb-core for more i
wandb: Logging into wandb.ai. (Learn how to deploy a W&B server locally: https://wandb.me/wandb-ser
wandb: You can find your API key in your browser here: https://wandb.ai/authorize
wandb: Paste an API key from your profile and hit enter: .....
wandb: WARNING If you're specifying your api key in code, ensure this code is not shared publicly.
wandb: WARNING Consider setting the WANDB_API_KEY environment variable, or running `wandb login` fr
wandb: No netrc file found, creating one.
wandb: Appending key for api.wandb.ai to your netrc file: /root/.netrc
wandb: Currently logged in as: jbaron (jbaron-dascd) to https://api.wandb.ai. Use `wandb login --re
Tracking run with wandb version 0.19.8
Run data is saved locally in /content/wandb/run-20250325_170833-ahcq6cx3
Syncing run ./results to Weights & Biases (docs)
View project at https://wandb.ai/jbaron-dascd/huggingface
View run at https://wandb.ai/jbaron-dascd/huggingface/runs/ahcq6cx3
[66512/66512 3:17:57, Epoch 2/2]

Epoch Training Loss Validation Loss
1 0.292200 0.331890
2 0.148400 0.256323

TrainOutput(global_step=66512, training_loss=0.405639834840918, metrics={'train_runtime':
11890.496, 'train_samples_per_second': 89.497, 'train_steps_per_second': 5.594, 'total_flos':
7.003163807651942e+16, 'train_loss': 0.405639834840918, 'epoch': 2.0})

# Guardar modelo y tokenizer en Google Drive
trainer.save_model('/content/drive/MyDrive/modelo_entrenado_beto')
tokenizer.save_pretrained('/content/drive/MyDrive/modelo_entrenado_beto')

# Evaluación
preds = trainer.predict(ds['validation'])
y_pred = np.argmax(preds.predictions, axis=1)
```

Desarrollar y organizar bases de datos para los modelos de información del proceso de Organización del Trabajo. La Inteligencia Artificial será la base para implementar funciones de creación, lectura, actualización y borrado de datos, optimizando la gestión de la información.

CONTROL Y SEGUIMIENTO A		
1013632801	PROCESOS/PROYECTOS/PROGRAMAS	PROFESIONAL
41790280	SERVICIOS DE APOYO A LA GESTIÓN ADMINISTRATIVA Y/O ASISTENCIAL	
1010173266	CONTROL Y SEGUIMIENTO A PROCESOS/PROYECTOS/PROGRAMAS	
1065642375	GESTIÓN CONTRACTUAL	PROFESIONAL
1019070983	GESTIÓN CONTRACTUAL	PROFESIONAL
1010181693	GESTIÓN POLICIVA	PROFESIONAL
7181544	NO CLASIFICADOS	
1020750697	CONTROL Y SEGUIMIENTO A PROCESOS/PROYECTOS/PROGRAMAS	
1013596336	GESTIÓN AMBIENTAL	
80724789	NO CLASIFICADOS	
1020763540	NO CLASIFICADOS	
1032459187	RELACIONAMIENTO CON LA CIUDADANÍA	
1018452269	NO CLASIFICADOS	
1000372101	NO CLASIFICADOS	

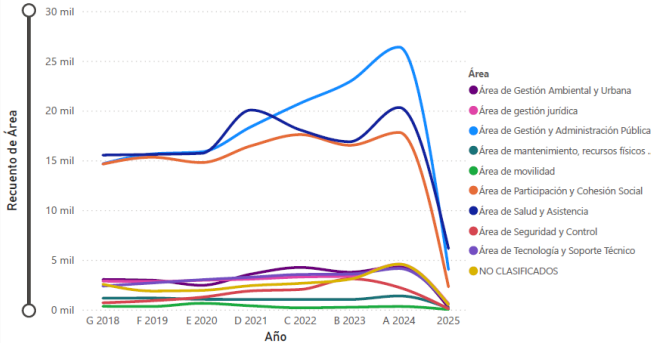
1020750697	SERVICIOS DE APOYO A LA GESTIÓN ADMINISTRATIVA Y/O ASISTENCIAL	PROFESIONAL
1013596336	CONTROL Y SEGUIMIENTO A PROCESOS/PROYECTOS/PROGRAMAS	PROFESIONAL
80724789	CONTROL Y SEGUIMIENTO A PROCESOS/PROYECTOS/PROGRAMAS	
1020763540	GESTIÓN JURÍDICA	PROFESIONAL
1032459187	CONTROL Y SEGUIMIENTO A PROCESOS/PROYECTOS/PROGRAMAS	PROFESIONAL
1018452269	SERVICIOS DE APOYO A LA GESTIÓN ADMINISTRATIVA Y/O ASISTENCIAL	PROFESIONAL
53043291	GESTIÓN CONTRACTUAL	PROFESIONAL
1018515558	RELACIONAMIENTO CON LA CIUDADANÍA	
1026267438	GESTIÓN JURÍDICA	PROFESIONAL
1010213205	GESTIÓN AMBIENTAL	PROFESIONAL
81717618	GESTIÓN CONTRACTUAL	PROFESIONAL
53032910	RELACIONAMIENTO CON LA CIUDADANÍA	PROFESIONAL
1022401679	CORRESPONDENCIA/GESTIÓN DOCUMENTAL	
53066743	NO CLASIFICADOS	
53050590	RELACIONAMIENTO CON LA CIUDADANÍA	PROFESIONAL
79416261	GESTIÓN JURÍDICA	PROFESIONAL
10779866	GESTIÓN CONTABLE Y FINANCIERA	PROFESIONAL
1032406798	CONTROL Y SEGUIMIENTO A PROCESOS/PROYECTOS/PROGRAMAS	PROFESIONAL
80797275	CONTROL Y SEGUIMIENTO A PROCESOS/PROYECTOS/PROGRAMAS	PROFESIONAL
79417249	CONTROL Y SEGUIMIENTO A PROCESOS/PROYECTOS/PROGRAMAS	PROFESIONAL
1030569086	CONTROL Y SEGUIMIENTO A PROCESOS/PROYECTOS/PROGRAMAS	PROFESIONAL
1030590897	CONTROL Y SEGUIMIENTO A PROCESOS/PROYECTOS/PROGRAMAS	PROFESIONAL
52436713	CONTROL Y SEGUIMIENTO A PROCESOS/PROYECTOS/PROGRAMAS	PROFESIONAL
52988711	RELACIONAMIENTO CON LA CIUDADANÍA	
45519331	GESTIÓN CONTABLE Y FINANCIERA	PROFESIONAL
52428963	RELACIONAMIENTO CON LA CIUDADANÍA	
1015441924	RELACIONAMIENTO CON LA CIUDADANÍA	
1020739257	RELACIONAMIENTO CON LA CIUDADANÍA	
1020785938	RELACIONAMIENTO CON LA CIUDADANÍA	
80421812	RELACIONAMIENTO CON LA CIUDADANÍA	
27738294	RELACIONAMIENTO CON LA CIUDADANÍA	
1015458217	RELACIONAMIENTO CON LA CIUDADANÍA	
80143120	GESTIÓN AMBIENTAL	PROFESIONAL
1020808644	GESTIÓN JURÍDICA	PROFESIONAL
80275145	ATENCIÓN ANIMAL Y VETERINARIA	PROFESIONAL
1010036367	COMUNICACIONES/PERIODISMO	PROFESIONAL
1073969872	GESTIÓN CONTABLE Y FINANCIERA	PROFESIONAL

	CONTROL Y SEGUIMIENTO A	
1094897594	PROCESOS/PROYECTOS/PROGRAMAS	PROFESIONAL
	SERVICIOS DE APOYO A LA GESTIÓN	
1070917500	ADMINISTRATIVA Y/O ASISTENCIAL	PROFESIONAL
1010228018	GESTIÓN CONTRACTUAL	PROFESIONAL
52707718	GESTIÓN JURÍDICA	PROFESIONAL
	SERVICIOS DE APOYO A LA GESTIÓN	
1020839818	ADMINISTRATIVA Y/O ASISTENCIAL	PROFESIONAL
1193034339	SERVICIOS DE APOYO A LA GESTIÓN ADMINISTRATIVA Y/O ASISTENCIAL	
80028120	GESTIÓN CONTRACTUAL	PROFESIONAL
79860852	CONDUCTOR	
53000759	GESTIÓN CONTABLE Y FINANCIERA	PROFESIONAL
1000948768	INFRAESTRUCTURA	
1032442625	SEGURIDAD Y SALUD EN EL TRABAJO	PROFESIONAL
1022398466	GESTIÓN CONTRACTUAL	PROFESIONAL
	GESTIÓN, ACOMPAÑAMIENTO Y DESARROLLO	
1020766924	COMUNITARIO	PROFESIONAL
	RECUPERACIÓN, DEFENSA Y SUPERVISIÓN DEL	
1018484018	ESPACIO PÚBLICO/URBANISMO	PROFESIONAL
52930030	GESTIÓN CONTRACTUAL	PROFESIONAL
1104805513	RELACIONAMIENTO CON LA CIUDADANÍA	
80134896	COMUNICACIONES/PERIODISMO	PROFESIONAL
1010173266	GESTIÓN CONTRACTUAL	PROFESIONAL
52691346	RELACIONAMIENTO CON LA CIUDADANÍA	
1020809533	CORRESPONDENCIA/GESTIÓN DOCUMENTAL	
	CONTROL Y SEGUIMIENTO A	
52356569	PROCESOS/PROYECTOS/PROGRAMAS	PROFESIONAL
	DESARROLLO TECNOLÓGICO, SOPORTE DE APLICATIVOS Y PLATAFORMA,	
80212092	BASES DE DATOS	
	DESARROLLO TECNOLÓGICO, SOPORTE DE APLICATIVOS Y PLATAFORMA,	
1143370139	BASES DE DATOS	
1007344351	RELACIONAMIENTO CON LA CIUDADANÍA	
1019140373	NO CLASIFICADOS	PROFESIONAL
	CONTROL Y SEGUIMIENTO A	
1003913722	PROCESOS/PROYECTOS/PROGRAMAS	PROFESIONAL
	CONTROL Y SEGUIMIENTO A	
1001199498	PROCESOS/PROYECTOS/PROGRAMAS	PROFESIONAL
1065814267	RELACIONAMIENTO CON LA CIUDADANÍA	PROFESIONAL
	SERVICIOS DE APOYO A LA GESTIÓN	
52410335	ADMINISTRATIVA Y/O ASISTENCIAL	PROFESIONAL
79769870	RELACIONAMIENTO CON LA CIUDADANÍA	
	CONTROL Y SEGUIMIENTO A	
1018464994	PROCESOS/PROYECTOS/PROGRAMAS	PROFESIONAL
1007426834	GESTIÓN JURÍDICA	
7141363	INFRAESTRUCTURA	
52419797	CORRESPONDENCIA/GESTIÓN DOCUMENTAL	
1018479951	GESTIÓN JURÍDICA	PROFESIONAL

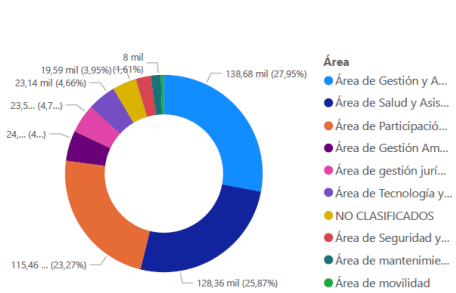
	CONTROL Y SEGUIMIENTO A	
52816659	PROCESOS/PROYECTOS/PROGRAMAS	PROFESIONAL
1020802719	NO CLASIFICADOS	PROFESIONAL
1020849033	CORRESPONDENCIA/GESTIÓN DOCUMENTAL	
1020767876	RELACIONAMIENTO CON LA CIUDADANÍA	
71674865	RECURSOS FÍSICOS, INVENTARIO Y ALMACÉN	
1018454192	RELACIONAMIENTO CON LA CIUDADANÍA	PROFESIONAL
1004347551	GESTIÓN JURÍDICA	PROFESIONAL
1032430328	GESTIÓN JURÍDICA	PROFESIONAL
80799927	GESTIÓN JURÍDICA	PROFESIONAL
1022329382	SERVICIOS DE APOYO A LA GESTIÓN ADMINISTRATIVA Y/O ASISTENCIAL	
	SERVICIOS DE APOYO A LA GESTIÓN	
1015471221	ADMINISTRATIVA Y/O ASISTENCIAL	PROFESIONAL
1070614675	SEGURIDAD Y SALUD EN EL TRABAJO	PROFESIONAL
1001217774	GESTIÓN AMBIENTAL	
80409994	GESTIÓN CONTABLE Y FINANCIERA	PROFESIONAL

Apoyar a la Subdirección Técnica de Desarrollo Organizacional y Empleo Público, en el uso y generación de herramientas de Inteligencia Artificial para el procesamiento y visualización de datos, que sirvan de insumos para el cumplimiento de las funciones.

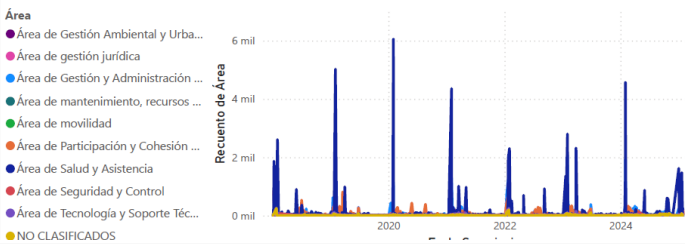
Incidencia en el tiempo por área



Clasificación por área

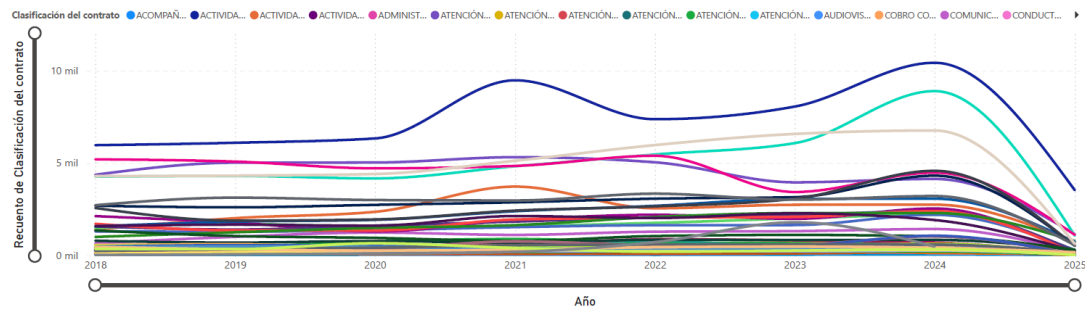


Estacionalidad de contratación

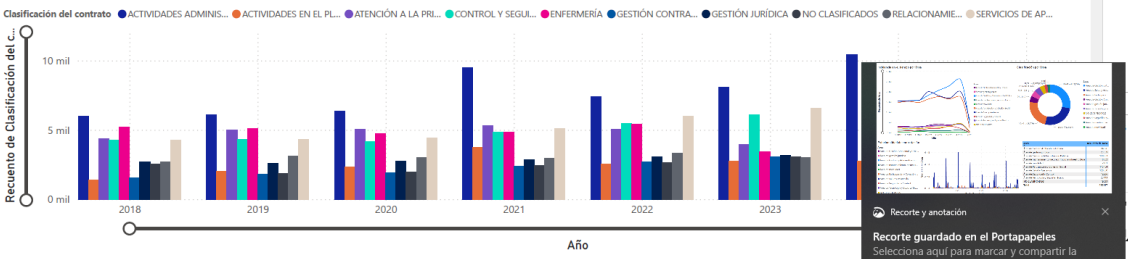


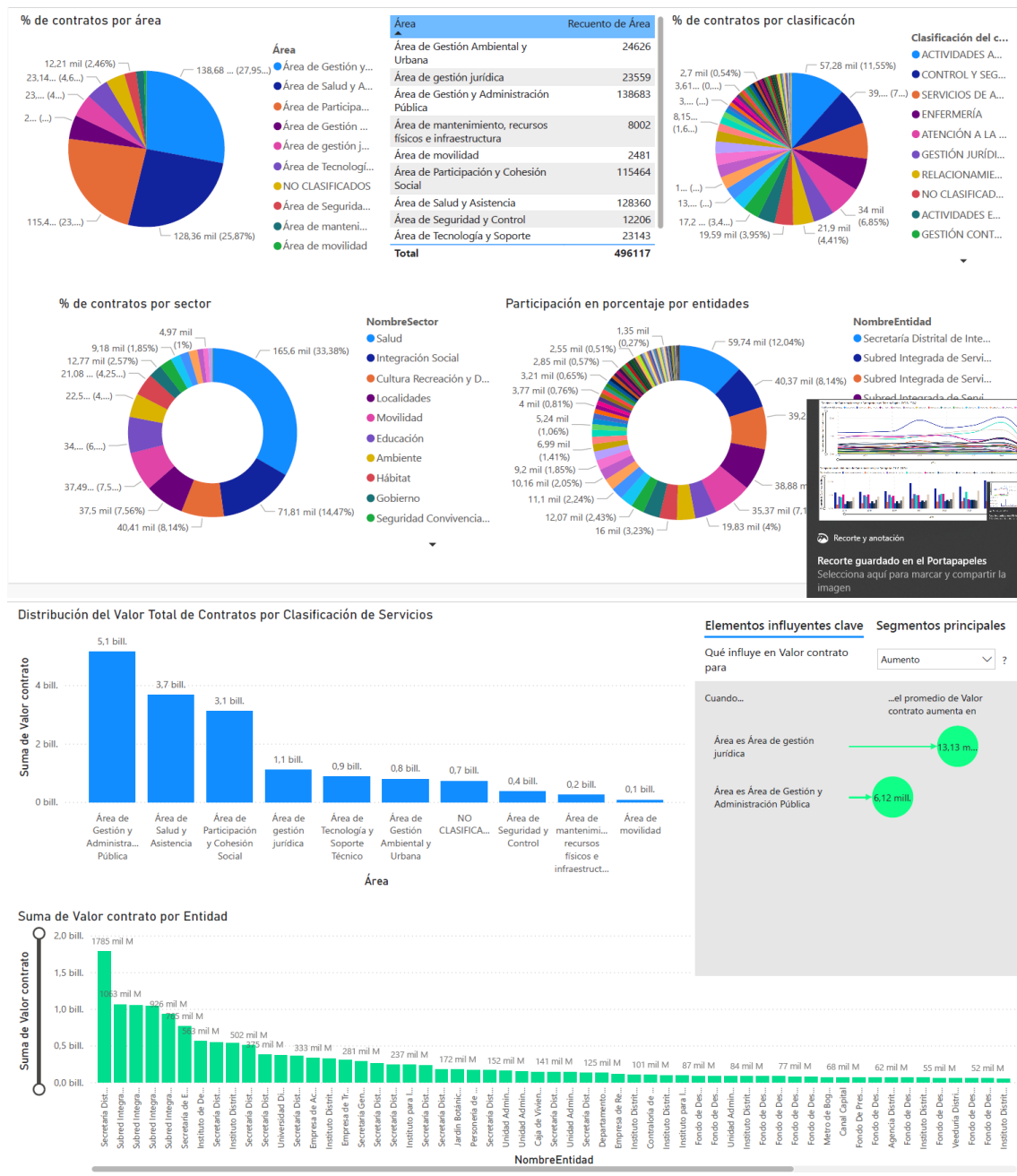
Área	Recuento de Área
Área de Gestión Ambiental y Urbana	24626
Área de gestión jurídica	23559
Área de Gestión y Administración Pública	138683
Área de mantenimiento, recursos físicos e infraestructura	8002
Área de movilidad	2481
Área de Participación y Cohesión Social	115464
Área de Salud y Asistencia	128360
Área de Seguridad y Control	12206
Área de Tecnología y Soporte Técnico	23143
NO CLASIFICADOS	19593
Total	496117

Tendencia de Contrataciones por Categoría en el Distrito Capital (2018-2024)



Comparativo de Volumen de Contrataciones por Categoría (2018-2024)





Crear guías de usuario para los modelos desarrollados, que facilitarán el uso y la comprensión de las herramientas por parte de los usuarios.

- coding: utf-8 -

=====

Proyecto: Clasificación de Contratos con BETO-BERT

Autor: [Nombre del responsable o equipo]

Versión: Octubre 2025

Descripción:

Este script implementa un pipeline completo para la clasificación de textos contractuales

usando el modelo BETO (BERT entrenado para español). El flujo incluye:

- Carga y limpieza de datos
- Codificación de etiquetas
- Entrenamiento del modelo
- Evaluación de desempeño
- Generación de predicciones para nuevos textos

"""

```
# =====
```

```
# 🛠 Instalación de librerías necesarias
```

```
# =====
```

```
!pip install transformers datasets scikit-learn
```

```
# =====
```

```
# 📖 Importación de librerías
```

```
# =====
```

```
import pandas as pd
```

```
import torch
```

```
from sklearn.model_selection import train_test_split
```

```
from sklearn.preprocessing import LabelEncoder
```

```
from sklearn.metrics import classification_report, confusion_matrix
```

```
from transformers import AutoTokenizer, AutoModelForSequenceClassification, Trainer,  
TrainingArguments
```

```
from datasets import Dataset, DatasetDict
```

```
import numpy as np
```

```
# =====
```

```
# ⚙ Configuración de los parámetros de entrenamiento
```

```
# =====
```

```
training_args = TrainingArguments(  
    output_dir='./results',
```

```
    # Carpeta donde se guardan los resultados
```

```

evaluation_strategy='epoch',    # Evalúa al final de cada época
per_device_train_batch_size=16, # Tamaño de lote para entrenamiento
per_device_eval_batch_size=16,  # Tamaño de lote para validación
num_train_epochs=2,             # Número de épocas de entrenamiento
fp16=True,                      # Usa precisión de 16 bits si hay GPU compatible
weight_decay=0.01,              # Regularización L2
logging_steps=10,               # Frecuencia de registro
save_total_limit=1              # Número máximo de checkpoints guardados
)

```

```

# =====

```

```

# 📁 Carga del archivo con los contratos clasificados

```

```

# =====

```

```

from google.colab import files

```

```

uploaded = files.upload() # Permite subir manualmente el archivo .csv

```

```

# Se carga el archivo y se seleccionan las columnas relevantes

```

```

df = pd.read_csv('datos_clasificados.csv', encoding='latin1', sep=';')

```

```

df = df[['objeto', 'clasificacion_del_contrato']].dropna()

```

```

# =====

```

```

# 🏷️ Codificación de las etiquetas (categorías de contratos)

```

```

# =====

```

```

label_encoder = LabelEncoder()

```

```

df['label'] = label_encoder.fit_transform(df['clasificacion_del_contrato'])

```

```

num_labels = len(label_encoder.classes_) # Total de categorías detectadas

```

```

# =====

```

```

# 🧠 División en datos de entrenamiento y validación

```

```

# =====

```

```

train_texts, val_texts, train_labels, val_labels = train_test_split(

```

```

df['objeto'].tolist(),
df['label'].tolist(),
test_size=0.1,
stratify=df['label'],
random_state=42
)

# =====

# 📄 Tokenización de los textos (preparación para BERT)
# =====

model_name = 'dccuchile/bert-base-spanish-wwm-cased' # Modelo BETO
tokenizer = AutoTokenizer.from_pretrained(model_name)

def tokenize(batch):
    """Convierte los textos en tokens para el modelo BERT"""
    return tokenizer(batch['text'], truncation=True, padding='max_length', max_length=128)

# Creación de datasets compatibles con HuggingFace
train_ds = Dataset.from_dict({'text': train_texts, 'label': train_labels})
val_ds = Dataset.from_dict({'text': val_texts, 'label': val_labels})
ds = DatasetDict({'train': train_ds, 'validation': val_ds})

# Aplicar tokenización a todo el dataset
ds = ds.map(tokenize, batched=True)
ds = ds.remove_columns(['text'])
ds.set_format('torch')

# =====

# 🧠 Carga del modelo BETO preentrenado
# =====

```

```
model = AutoModelForSequenceClassification.from_pretrained(model_name,
num_labels=num_labels)
```

```
# =====
```

```
# 🧠 Entrenamiento del modelo
```

```
# =====
```

```
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=ds['train'],
    eval_dataset=ds['validation'],
)
```

```
trainer.train()
```

```
# =====
```

```
# 💾 Guardado del modelo entrenado y tokenizador
```

```
# =====
```

```
from google.colab import drive
drive.mount('/content/drive')
trainer.save_model('/content/drive/MyDrive/modelo_entrenado_beto')
tokenizer.save_pretrained('/content/drive/MyDrive/modelo_entrenado_beto')
```

```
# =====
```

```
# 📊 Evaluación del modelo
```

```
# =====
```

```
preds = trainer.predict(ds['validation'])
y_pred = np.argmax(preds.predictions, axis=1)
print(classification_report(val_labels, y_pred, target_names=label_encoder.classes_))
```

```
# =====
```

```
# 🗨 Predicción de nuevos textos (sin clasificar)
```

```

# =====

from transformers import BertForSequenceClassification, BertTokenizer,
TextClassificationPipeline

# Cargar el modelo guardado desde Drive

model = BertForSequenceClassification.from_pretrained(
    '/content/drive/MyDrive/modelo_entrenado_beto',
    local_files_only=True
)

tokenizer = BertTokenizer.from_pretrained(
    '/content/drive/MyDrive/modelo_entrenado_beto',
    local_files_only=True
)

# Crear el pipeline de clasificación

pipeline = TextClassificationPipeline(
    model=model,
    tokenizer=tokenizer,
    device=0 if torch.cuda.is_available() else -1
)

# Preparar los textos para predecir (usando la columna objeto_a_contratar)

df_unlabeled = df.dropna(subset=['objeto_a_contratar'])
texts_to_predict = df_unlabeled['objeto_a_contratar'].tolist()

# Configurar las etiquetas reales del modelo

etiquetas_reales = [ ... lista completa de categorías ... ]

model.config.id2label = {i: etiqueta for i, etiqueta in enumerate(etiquetas_reales)}
model.config.label2id = {etiqueta: i for i, etiqueta in enumerate(etiquetas_reales)}

# Ejecutar las predicciones

```

```

predicciones = pipeline(texts_to_predict)

# Crear un DataFrame con los resultados
df_predicciones = pd.DataFrame({
    'texto': texts_to_predict,
    'prediccion': [p['label'] for p in predicciones],
    'score': [p['score'] for p in predicciones]
})

# Guardar predicciones
df_predicciones.to_csv('/content/drive/MyDrive/Defensoría_SECOP.csv', index=False)
print('✔ Predicciones guardadas correctamente en Drive.')

# =====
# 🌀 Predicción manual en GPU (opcional)
# =====
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
model.to(device)
tokenized_inputs = tokenizer(
    texts_to_predict,
    truncation=True,
    padding='max_length',
    max_length=128,
    return_tensors='pt'
)
tokenized_inputs = {k: v.to(device) for k, v in tokenized_inputs.items()}

# Evaluar y generar etiquetas
model.eval()
with torch.no_grad():
    outputs = model(**tokenized_inputs)

```

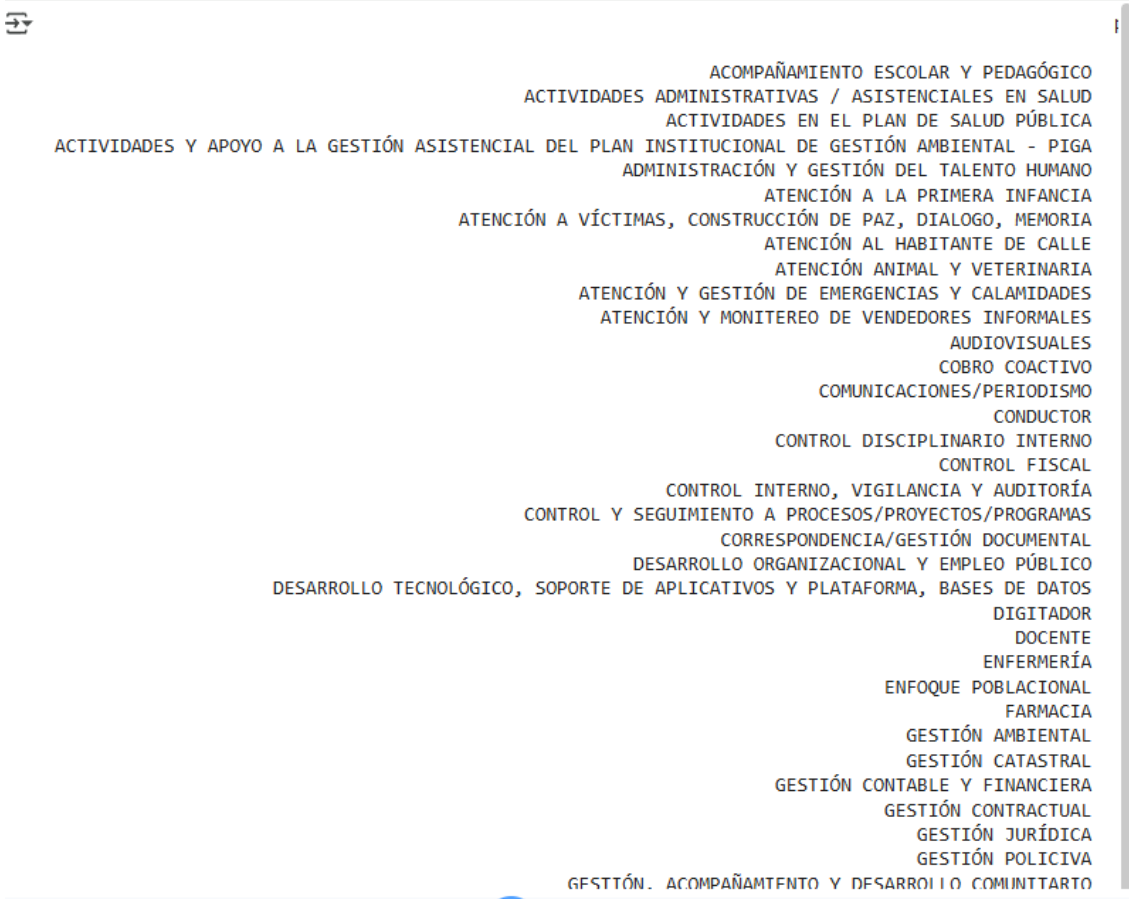
```
preds_unlabeled = torch.argmax(outputs.logits, axis=1).cpu().numpy()
```

```
pred_labels = label_encoder.inverse_transform(preds_unlabeled)
```

```
df_unlabeled['prediccion'] = pred_labels
```

6. Mantener un registro detallado del código de software empleado en el desarrollo de los módulos del proceso de Organización del Trabajo, de acuerdo con los procedimientos establecidos por el DASCD para la documentación, cuando sea pertinente.

```
#  Evaluación
preds = trainer.predict(ds['validation'])
y_pred = np.argmax(preds.predictions, axis=1)
print(classification_report(val_labels, y_pred, target_names=label_encoder.classes_))
```



```
from google.colab import drive
drive.mount('/content/drive')
```

Mounted at /content/drive

```
import os
print(os.listdir('/content/drive/MyDrive/modelo_entrenado_beto'))
```

```
['vocab.txt', 'tokenizer_config.json', 'training_args.bin', 'config.json', 'special_tokens_map.json']
```

```
import pandas as pd
```

```
from google.colab import files

# Subir archivo CSV manualmente
uploaded = files.upload()
```

Sin archivos seleccionados Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving procesar.csv to procesar (2).csv

```
import pandas as pd

df_unlabeled = pd.read_csv('procesar.csv', encoding='latin1', sep=';')
```

Predicción de textos sin clasificar

```
from transformers import BertForSequenceClassification, BertTokenizer, TextClassificationPipeline
import torch

model = BertForSequenceClassification.from_pretrained(
    '/content/drive/MyDrive/modelo_entrenado_beto',
    local_files_only=True
)
```

```
# ✅ Asignar etiquetas reales al modelo (id2label y label2id)
etiquetas_reales = [
    "ACOMPANAMIENTO ESCOLAR Y PEDAGÓGICO",
    "ACTIVIDADES ADMINISTRATIVAS / ASISTENCIALES EN SALUD",
    "ACTIVIDADES EN EL PLAN DE SALUD PÚBLICA",
    "ACTIVIDADES Y APOYO A LA GESTIÓN ASISTENCIAL DEL PLAN INSTITUCIONAL DE GESTIÓN AMBIENTAL - PI",
    "ADMINISTRACIÓN Y GESTIÓN DEL TALENTO HUMANO",
    "ATENCIÓN A LA PRIMERA INFANCIA",
    "ATENCIÓN A VÍCTIMAS, CONSTRUCCIÓN DE PAZ, DIÁLOGO, MEMORIA",
    "ATENCIÓN AL HABITANTE DE CALLE",
    "ATENCIÓN ANIMAL Y VETERINARIA",
    "ATENCIÓN Y GESTIÓN DE EMERGENCIAS Y CALAMIDADES",
    "ATENCIÓN Y MONITOREO DE VENDEDORES INFORMALES",
    "AUDIOVISUALES",
    "COBRO COACTIVO",
    "COMUNICACIONES/PERIODISMO",
    "CONDUCTOR",
    "CONTROL DISCIPLINARIO INTERNO",
    "CONTROL FISCAL",
    "CONTROL INTERNO, VIGILANCIA Y AUDITORÍA",
    "CONTROL Y SEGUIMIENTO A PROCESOS/PROYECTOS/PROGRAMAS",
    "CORRESPONDENCIA/GESTIÓN DOCUMENTAL",
    "DESARROLLO ORGANIZACIONAL Y EMPLEO PÚBLICO",
    "DESARROLLO TECNOLÓGICO, SOPORTE DE APLICATIVOS Y PLATAFORMA, BASES DE DATOS",
    "DIGITADOR",
    "DOCENTE",
    "ENFERMERÍA",
    "ENFOQUE POBLACIONAL",
    "FARMACIA",
    "GESTIÓN AMBIENTAL",
    "GESTIÓN CATASTRAL",
    "GESTIÓN CONTABLE Y FINANCIERA",
    "GESTIÓN CONTRACTUAL",
    "GESTIÓN JURÍDICA",
    "GESTIÓN POLÍTICA",
    "GESTIÓN, ACOMPANAMIENTO Y DESARROLLO COMUNITARIO",
    "GESTOR (PAZ, CONVIVENCIA, COMUNITARIO, ETC)",
    "GUARDIAN CICLOVÍA",
    "HÁBITAT"
```

```
1 # 🤖 Predicción usando pipeline con manejo automático de memoria
import pandas as pd

# Usa la variable correcta definida antes: texts_to_predict
id2label = model.config.id2label
predicciones = pipeline(texts_to_predict)

df_predicciones = pd.DataFrame({
    'texto': texts_to_predict,
    'prediccion': [p['label'] for p in predicciones],
    'score': [p['score'] for p in predicciones]
})

df_predicciones.to_csv('/content/drive/MyDrive/Defensoría SECOP.csv', index=False)
print('✅ Predicciones guardadas en Drive como predicciones_beto2.csv')
```

✅ Predicciones guardadas en Drive como predicciones_beto2.csv

```
1 id2label = model.config.id2label

# Para obtener la lista en orden:
lista_etiquetas = [id2label[i] for i in sorted(id2label.keys())]
```

```
df_predicciones.to_csv('predicciones_beto', index=False, encoding='utf-8')
df_predicciones.to_csv()
```

```
from google.colab import files
```

```
df predicciones.to_csv('predicciones beto.csv', index=False, encoding='utf-8')
```

```
# Descargar el archivo al equipo local
files.download('predicciones beto.csv')
```

```
df_predicciones.to_excel('predicciones_beto.xlsx', index=False)
from google.colab import files
files.download('predicciones_beto.xlsx')
```

John Brown

Cedula de ciudadanía No. 1010244119

Historial

El usuario 'GINA PAOLA SILVA VASQUEZ (1026250419 : gsilva@serviciocivil.gov.co)' firmo el documento involucrado en el flujo de firmas satisfactoriamente.

04/11/2025 02:30:15 PM

Email enviado a todos los participantes con el documento finalizado.

04/11/2025 02:30:15 PM

Flujo de firmas concluido

04/11/2025 02:30:15 PM

Flujo de firmas creado por 'JHOAN NICOLAS MORA ARIZA (1000122861 : jmora@serviciocivil.gov.co)'

04/11/2025 12:02:23 PM
