

# Universidad Central del Ecuador

---

**Título: Metodología CRISP DM para la  
aplicación de TisserIA**





# Programación para dispositivos móviles

## Metodología CRISP DM para la aplicación de TisserIA

[Balseca Pablo, Echeverría Yessenia, Sandoval Stalin y Tualombo Leonel]

Carrera Sistemas de Información, Facultad de Ingeniería y Ciencias Aplicadas

Décimo Semestre

Ing. Noguera Diego

Miércoles, 23 de febrero de 2024

## Contenido

1	Comprensión del negocio .....	3
1.1	Objetivos del negocio .....	3
1.2	Evaluación de la situación .....	3
1.3	Objetivos del Data Mining.....	4
1.4	Inventario de Recursos.....	4
1.5	Supuestos y Restricciones.....	5
1.6	Riesgos y Contingencias .....	6
1.7	Criterios de Éxito .....	7
2	Comprensión de los datos.....	7
2.1	Recolección de los datos.....	7
2.2	Descripción de los datos .....	8
2.3	Exploración de los datos .....	8
2.4	Verificación de la calidad de los datos .....	10
3	Preparación de los datos .....	11
3.1	Selección de los datos .....	12
3.2	Limpieza de los datos.....	12
3.3	Construir los datos .....	13
3.4	Integrar los datos.....	13
3.5	Formateo de los datos .....	14
4	Modelado.....	14
4.1	Escoger la técnica de modelado.....	14
4.2	Generar plan de prueba .....	16
4.3	Construir el modelo: .....	17
5	Evaluación .....	20
5.1	Evaluar los resultados .....	21
6	Implementación .....	22

## 1 Comprensión del negocio

### 1.1 Objetivos del negocio

- **Propósito principal:** Desarrollar una aplicación móvil que facilite la replicación de prendas de crochet mediante la identificación de patrones utilizando IA.
- **Audiencia objetivo:** Usuarios de todas las edades interesados en el crochet, desde principiantes hasta expertos.
- **Problemas por abordar:** Dificultad para encontrar y entender patrones de crochet, lo que limita la capacidad de replicar prendas de manera precisa.
- **Impacto esperado:** Fomentar el crecimiento de emprendimientos relacionados con el crochet y mejorar las habilidades profesionales de los usuarios.

### 1.2 Evaluación de la situación

En la actualidad, existe una creciente comunidad de aficionados al crochet que buscan nuevas herramientas y recursos para mejorar sus habilidades y proyectos. Sin embargo, la mayoría de las aplicaciones disponibles se centran en tutoriales en video cortos o imágenes estáticas, lo que limita la capacidad de los usuarios para comprender y replicar patrones complejos. Además, hay una clara demanda de una aplicación móvil que pueda analizar y proporcionar instrucciones detalladas sobre patrones de crochet, ya que muchos entusiastas enfrentan dificultades para encontrar recursos de calidad que se adapten a sus necesidades específicas.

El propósito principal de este proyecto es desarrollar una aplicación móvil que facilite la replicación de prendas de crochet mediante la identificación de patrones utilizando inteligencia artificial. La audiencia objetivo abarca desde principiantes hasta expertos en crochet, con la intención de ofrecer una herramienta accesible y útil para todos los niveles de habilidad.

Abordaremos la dificultad que enfrentan muchos entusiastas del crochet al intentar encontrar y comprender patrones complejos, lo que limita su capacidad para crear prendas con precisión. Se espera que esta aplicación no solo impulse el crecimiento de emprendimientos relacionados con el crochet, sino que también mejore las habilidades profesionales y la satisfacción de los usuarios al crear sus propias creaciones.

### 1.3 Objetivos del Data Mining

- Identificar y analizar patrones de crochet en imágenes proporcionadas por los usuarios.
- Desarrollar un modelo de IA que pueda reconocer y clasificar diferentes tipos de puntos y técnicas de crochet.
- Generar instrucciones paso a paso para replicar los patrones identificados, adaptadas al nivel de habilidad del usuario.

### 1.4 Inventario de Recursos

En cuanto a recursos de Software, se dispone de las siguientes herramientas:

Back y Front:

- CPU: Intel Core i3 8va generación
- RAM :16GB
- GPU: NVIDIA GTX 1660 super
- ROM: SSD 500gb

IA (Google Colab):

- CPU: Intel Xeon

## Programación para dispositivos móviles

- RAM: 13 GB – 83 GB
- GPU: A100 – V100
- VRAM: 16GB – 40GB

### 1.5 Supuestos y Restricciones

Supuestos:

Código	Supuestos	Descripción
<i>Sup_001</i>	1. Disponibilidad de recursos para la creación de la base de datos	Se asume que habrá recursos suficientes, como tiempo y disponibilidad, para recopilar, etiquetar y almacenar una cantidad significativa de imágenes de patrones de crochet.
<i>Sup_002</i>	2. Acceso a la variedad de patrones de crochet	Se supone que se podrá acceder a una amplia gama de patrones de crochet para incluir en la base de datos, lo que incluye diferentes estilos, niveles de dificultad y tipos de prendas.
<i>Sup_003</i>	3. Facilidad para obtener permisos de uso de imágenes	Se asume que será relativamente fácil obtener los permisos necesarios para utilizar las imágenes en la base de datos, ya sea a través de colaboraciones con diseñadores o mediante el uso de imágenes de dominio público.

Restricciones:

Código	Restricciones	Descripción
<i>Res_001</i>	1. Recursos limitados para la recopilación de datos	Dado que se requiere crear la base de datos desde cero, se enfrentarán a restricciones de tiempo, personal y presupuesto para recopilar y etiquetar las imágenes de manera efectiva.
<i>Res_002</i>	2. Limitaciones en la variedad y cantidad de datos	Es posible que la cantidad y variedad de imágenes disponibles sean limitadas, lo que podría afectar la capacidad del modelo de IA para identificar patrones con precisión.

## Programación para dispositivos móviles

<i>Res_003</i>	3. Calidad de las imágenes	Existe la restricción de que las imágenes recopiladas deben ser de alta calidad y tener una resolución suficiente para permitir una identificación precisa de los patrones de crochet.
<i>Res_004</i>	4. Consistencia en la etiquetación de datos	Se debe garantizar la consistencia en la etiquetación de las imágenes para entrenar el modelo de IA de manera efectiva, lo que puede ser un desafío debido a la subjetividad en la interpretación de los patrones de crochet.

### 1.6 Riesgos y Contingencias

<i>Código</i>	<b>Riesgo</b>	<b>Probabilidad</b>	<b>Impacto</b>	<b>Mitigación</b>
<i>RyC_001</i>	Es posible que los datos obtenidos para el proyecto no sean suficientes para concluir los resultados esperados	Alta	Medio	Generar la mayor cantidad de fotos de prendas de crochet
<i>RyC_002</i>	Tener una computadora que no cuente con GPU	Alta	Alto	Comprar una licencia de Google Colab el cual contendrá una GPU, Ram: 25GB y Rom: 166 GB
<i>RyC_003</i>	No detectar a los patrones de crochet	Media	Medio	Repetir los pasos para ajustar el modelo y poder obtener mejores resultados.
<i>RyC_004</i>	No cumplir con el plazo de entrega	Bajo	Alto	Trabajar más horas de las planificadas, incluyendo horario fuera de lo establecido

## 1.7 Criterios de Éxito

Código	Criterio de Éxito	Descripción
CE_001	Precisión del modelo de IA	Lograr una alta precisión en la identificación y clasificación de patrones de crochet.
CE_002	Usabilidad de la aplicación	Diseñar una interfaz de usuario intuitiva y proporcionar instrucciones claras para todos los usuarios.
CE_003	Retroalimentación de los usuarios	Valorar y utilizar la retroalimentación de los usuarios para mejorar continuamente la aplicación.
CE_004	Impacto en la comunidad de crochet	Aumentar la participación en la comunidad, estimular el crecimiento de emprendimientos relacionados y elevar el estándar de calidad en las creaciones de crochet.

## 2 Comprensión de los datos

### 2.1 Recolección de los datos

Inicialmente no se cuenta con una base de datos, por lo cual surge la necesidad de crear una. Esta base de datos creada desde cero usa imágenes referentes a tejidos de crochet y se encontrará alojada en Google Drive.

Para la recolección de datos, utilizamos una variedad de fuentes, incluyendo la implementación de Scrapy para recopilar imágenes de patrones de crochet de sitios web específicos. Configuramos y ejecutamos archivos de Scrapy en Google Colab para extraer automáticamente imágenes de alta calidad de sitios web de crochet populares y repositorios en línea. Además, complementamos esta estrategia con la búsqueda manual en redes sociales y la captura de fotografías propias de patrones de crochet.



dataset\_images\_pdf.zip

## 2.2 Descripción de los datos

VARIABLE (Columnas)	DESCRIPCIÓN
Imagen	Contiene una imagen de una prenda de crochet
Etiqueta	Imagen_numero.jpg
PDF	Contiene el patrón a seguir
Etiqueta	Imagen_numero.pdf

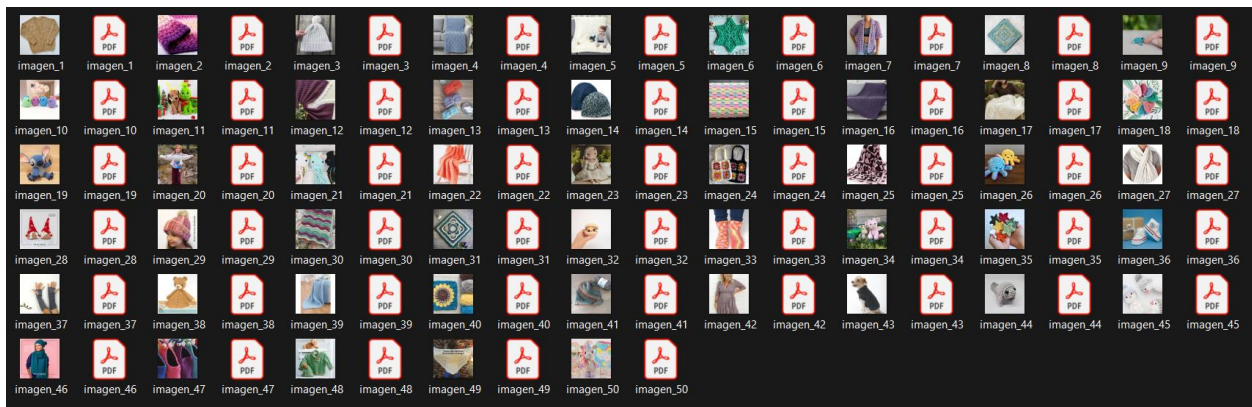
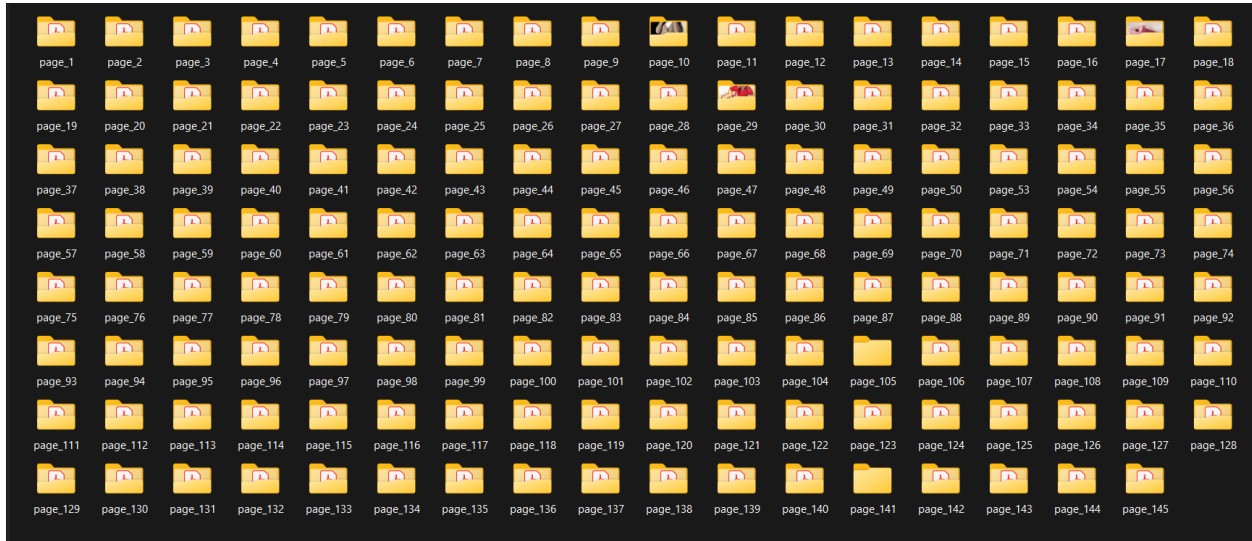
CONTENIDO (Filas)	DESCRIPCIÓN
6321	Número total de imágenes de prendas en crochet.
6321	Número total de archivos PDF con los patrones.



## 2.3 Exploración de los datos

Para explorar los datos, realizaremos una inspección visual de una muestra aleatoria de imágenes. Esto nos permitirá familiarizarnos con el contenido y la calidad de las imágenes en nuestro conjunto de datos. Durante esta inspección visual, estaremos atentos a cualquier patrón o tendencia preliminar que podamos observar, como estilos de tejido comunes, colores predominantes, o tipos de prendas representadas.

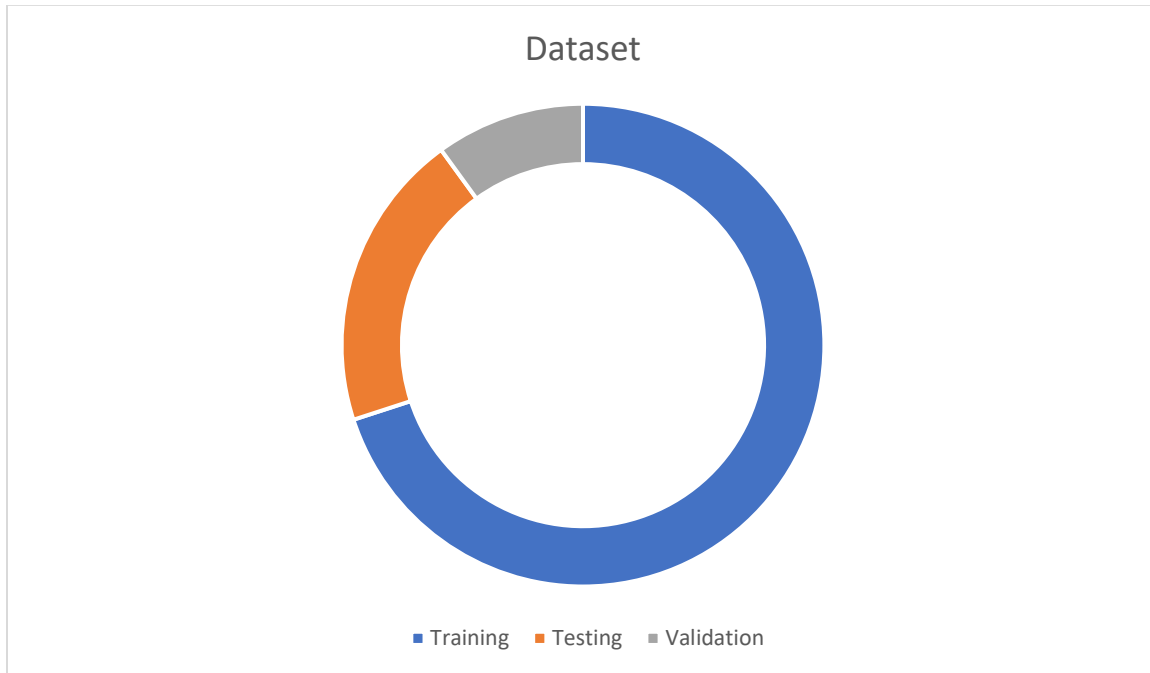
## Programación para dispositivos móviles



Una vez que se han descrito los datos, se procede a explorarlos, esto implica aplicar pruebas estadísticas básicas que revelarán propiedades de los datos, y crear un gráfico de pastel para determinar cómo se distribuirán las imágenes.

Número de imágenes: 6321

- Training: 70% - 4424
- Testing: 20% - 1265
- Validation: 10% - 632



### 2.4 Verificación de la calidad de los datos

Finalmente, verificaremos la calidad de los datos, evaluando cada imagen en busca de posibles problemas. Esto incluirá verificar si hay imágenes borrosas, con baja resolución, problemas de iluminación u otros problemas de calidad que puedan afectar la capacidad del modelo de IA para identificar los patrones de crochet de manera precisa.

Además, revisaremos la consistencia en el etiquetado de las imágenes, si están etiquetadas. Esto implica verificar si hay consistencia en los términos utilizados para describir los patrones y si las etiquetas son precisas y útiles para el propósito del proyecto.

Después de hacer la exploración inicial de los datos se puede afirmar que estos son completos. Los datos cubren los casos requeridos para la obtención de los resultados necesarios para poder cumplir los objetivos del proyecto.



*Ilustración 1: Imagen no apta, patrón no valido*



*Ilustración 2: Imagen valida, patrón valido*

### 3 Preparación de los datos

Durante esta fase, realizaremos una serie de tareas para preparar nuestros datos para su uso en el entrenamiento del modelo de identificación de patrones de crochet. Estas tareas incluirán la limpieza, la integración y el formateo de los datos según sea necesario.

### 3.1 Selección de los datos

En términos de imágenes, se utilizarán todas las imágenes recopiladas de patrones de crochet para nuestro proyecto, ya que hemos creado específicamente esta base de datos para cumplir con los objetivos del proyecto.

En nuestra base de datos de imágenes de patrones de crochet, hemos recopilado una variedad de imágenes que representan diferentes estilos, diseños y niveles de dificultad en el crochet. Esto nos proporciona una muestra diversa que nos permitirá entrenar nuestro modelo para identificar una amplia gama de patrones de crochet.

### 3.2 Limpieza de los datos

La base de datos utilizada para nuestro proyecto de identificación de patrones de crochet ha sido construida específicamente para cumplir con los objetivos de la minería de datos en este caso práctico. Esta base de datos se compone de una amplia variedad de imágenes de patrones de crochet recopiladas a través de web scraping de fuentes confiables y relevantes.

Durante el proceso de web scraping, se encontró que aproximadamente el 5% de los datos recopilados no eran adecuados para nuestro modelo. Estos datos fueron excluidos de nuestro conjunto de datos debido a diversos problemas, como:

<i>Código</i>	<i>Tipo de Problema</i>	<i>Descripción</i>	<i>Solución</i>
<i>LD_001</i>	<i>Registros mal Etiquetados</i>	Las imágenes no tenían la etiqueta correcta.	A las imágenes que tenían mal la etiqueta se las volvió a etiquetar para que se ajusten a las demás imágenes.

LD_002	Contenido irrelevante, archivos corruptos o baja definición	Las imágenes no sirven para el entrenamiento del modelo	Tratar de reemplazar esos archivos con imágenes que aporten con nuevas técnicas de tejidos.
--------	---	---	---

Es importante destacar que, debido al cuidadoso proceso de selección y preparación de datos, la base de datos restante se considera limpia y completa, sin necesidad de una limpieza adicional. Esto garantiza que nuestro modelo de identificación de patrones de crochet esté respaldado por datos de alta calidad y relevancia para cumplir con nuestros objetivos.

### 3.3 Construir los datos

Una vez que hemos seleccionado y limpiado nuestros datos, procedemos a construir nuestro conjunto de datos de entrenamiento, validación y prueba. Organizamos las imágenes en directorios separados según su función en el entrenamiento del modelo.

- Dataset clothes filter:  
<https://goo.su/SD8TC>
- Dataset crochet filter:  
<https://goo.su/5NUJfcP>
- Dataset crochet to pattern:  
<https://lc.cx/SAR6hk>

### 3.4 Integrar los datos

En el proyecto de identificación de patrones de crochet, no fue necesario crear nuevas estructuras ni fusionar distintas tablas de la base de datos. Dado que la base de datos consiste

principalmente en un conjunto de imágenes de patrones de crochet recopiladas mediante web scraping, no hubo necesidad de realizar fusiones o integraciones complejas entre conjuntos de datos.

La naturaleza del proyecto nos permitió trabajar con un solo conjunto de datos coherente y unificado, lo que simplificó significativamente el proceso de integración. Esto nos permitió centrarnos en la preparación y formateo de los datos de manera más eficiente, asegurando que estén listos para su uso en el entrenamiento del modelo de identificación de patrones de crochet.

### 3.5 Formateo de los datos

Para el proyecto, las imágenes de los patrones de crochet disponibles están en formatos .jpg y .png. Sin embargo, el formato de entrada requerido para el entrenamiento del modelo es exclusivamente .jpg. Por lo tanto, fue necesario convertir las imágenes en formato .png a .jpg antes de poder utilizarlas en el proceso de entrenamiento del modelo. Además, todas las imágenes fueron redimensionadas a una resolución de 255x255 para estandarizar el tamaño y facilitar el procesamiento durante el entrenamiento del modelo.

## 4 Modelado

### 4.1 Escoger la técnica de modelado

Modelos utilizados:

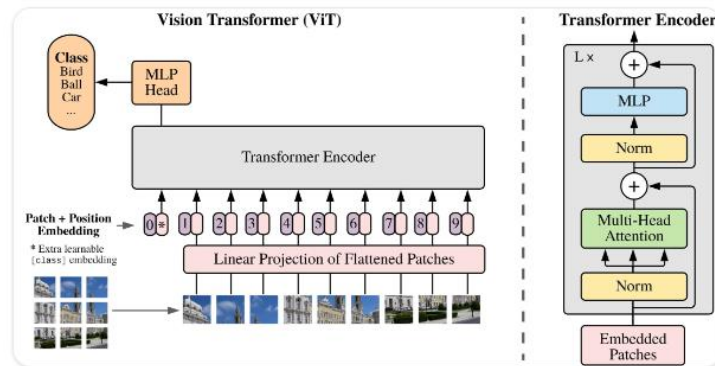
1. ViT (Clothes – Crochet - Filter):

El modelo Vision Transformer (ViT) fue propuesto en "Una imagen vale 16x16 palabras: Transformadores para el reconocimiento de imágenes a gran escala" por Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai,



## Programación para dispositivos móviles

Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby. Es el primer documento que entrena con éxito un codificador Transformer en ImageNet, logrando resultados muy buenos en comparación con arquitecturas convolucionales familiares.



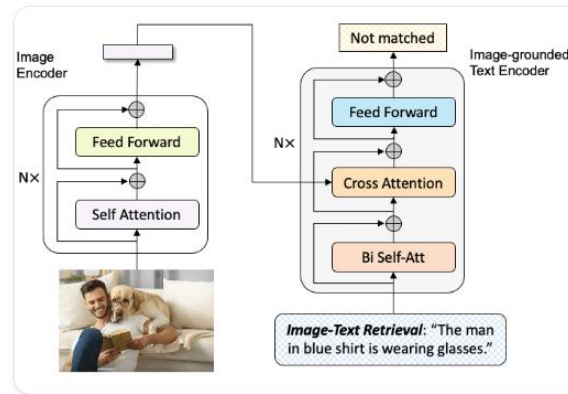
### 2. BLIP (Crochet2Pattern):

El modelo BLIP fue propuesto en "BLIP: Preentrenamiento de lenguaje-imagen de arranque para comprensión y generación unificadas de visión-lenguaje" por Junnan Li, Dongxu Li, Caiming Xiong, Steven Hoi.

BLIP es un modelo que puede realizar varias tareas multimodales, incluyendo:

- Respuesta a preguntas visuales
- Recuperación de imagen-texto (coincidencia de imagen-texto)
- Generación de subtítulos de imágenes





## 4.2 Generar plan de prueba

<i>Clothes Filter – Google/vit-base-patch16-224</i>	
<i>Fuente de datos</i>	<a href="https://goo.su/isuFYn">https://goo.su/isuFYn</a>
<i>Conjunto de entrenamiento</i>	Porcentaje De Datos Para Entrenamiento: 80%
<i>Conjunto de Testeo</i>	Porcentaje De Datos Para Testeo: 20%
<i>Modo de generar los conjuntos</i>	VitImageProcessor
<i>Medir Precisión</i>	Precisión Global, Precisión Positiva, Acurracy

<i>Crochet Filter – Google/vit-base-patch16-224</i>	
<i>Fuente de datos</i>	<a href="https://goo.su/V1EyYbk">https://goo.su/V1EyYbk</a>
<i>Conjunto de entrenamiento</i>	Porcentaje De Datos Para Entrenamiento: 80%
<i>Conjunto de Testeo</i>	Porcentaje De Datos Para Testeo: 20%
<i>Modo de generar los conjuntos</i>	VitImageProcessor

## Programación para dispositivos móviles

### *Medir Precisión*

Precisión Global, Precisión Positiva, Accuracy

### ***Crochet2Pattern – Salesforce/blip-image-captioning-large***

#### *Fuente de datos*

<https://goo.su/mJE6Bt>

#### *Conjunto de entrenamiento*

Porcentaje De Datos Para Entrenamiento: 80%

#### *Conjunto de Testeo*

Porcentaje De Datos Para Testeo: 20%

#### *Modo de generar los conjuntos*

BLIPProcessor

#### *Medir Precisión*

Precisión Global, Precisión Positiva, Accuracy

### 4.3 Construir el modelo:

- Clothes filter: Google/vit-base-patch16-224

```
args = TrainingArguments(
    f"{model_name}-clothes-filter",
    remove_unused_columns=False,
    evaluation_strategy = "epoch",
    save_strategy = "epoch",
    learning_rate=5e-5,
    per_device_train_batch_size=batch_size,
    gradient_accumulation_steps=4,
    per_device_eval_batch_size=batch_size,
    num_train_epochs=5,
    warmup_ratio=0.1,
    logging_steps=10,
    load_best_model_at_end=True,
    metric_for_best_model="accuracy",
    push_to_hub=True,
)
```

Ilustración 3 Ajuste de parámetros

```
model_checkpoint = "google/vit-base-patch16-224"
batch_size = 32
```

Ilustración 4: Construcción del modelo 1

## Programación para dispositivos móviles

```
from torchvision.transforms import (
    CenterCrop,
    Compose,
    Normalize,
    RandomHorizontalFlip,
    RandomResizedCrop,
    Resize,
    ToTensor,
)

normalize = Normalize(mean=image_processor.image_mean, std=image_processor.image_std)
if "height" in image_processor.size:
    size = (image_processor.size["height"], image_processor.size["width"])
    crop_size = size
    max_size = None
elif "shortest_edge" in image_processor.size:
    size = image_processor.size["shortest_edge"]
    crop_size = (size, size)
    max_size = image_processor.size.get("longest_edge")
```

Ilustración 5: Construcción del modelo 2

```
from transformers import AutoModelForImageClassification, TrainingArguments, Trainer

model = AutoModelForImageClassification.from_pretrained(
    model_checkpoint,
    label2id=label2id,
    id2label=id2label,
    ignore_mismatched_sizes = True,
)
```

Ilustración 6: Construcción del modelo 3

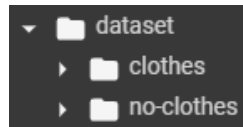


Ilustración 7: Estructura de directorio

- Crochet filter: Google/vit-base-patch16-224

```
args = TrainingArguments(
    f"{model_name}-crochets-clothes-classification",
    remove_unused_columns=False,
    evaluation_strategy = "epoch",
    save_strategy = "epoch",
    learning_rate=5e-5,
    per_device_train_batch_size=batch_size,
    gradient_accumulation_steps=4,
    per_device_eval_batch_size=batch_size,
    num_train_epochs=3,
    warmup_ratio=0.1,
    logging_steps=10,
    load_best_model_at_end=True,
    metric_for_best_model="accuracy",
    push_to_hub=True,
)
```

Ilustración 8 Ajuste de parámetros

## Programación para dispositivos móviles

```
model_checkpoint = "google/vit-base-patch16-224"  
batch_size = 32
```

Ilustración 9: Construcción del modelo 1

```
from torchvision.transforms import (  
    CenterCrop,  
    Compose,  
    Normalize,  
    RandomHorizontalFlip,  
    RandomResizedCrop,  
    Resize,  
    ToTensor,  
)  
  
normalize = Normalize(mean=image_processor.image_mean, std=image_processor.image_std)  
if "height" in image_processor.size:  
    size = (image_processor.size["height"], image_processor.size["width"])  
    crop_size = size  
    max_size = None  
elif "shortest_edge" in image_processor.size:  
    size = image_processor.size["shortest_edge"]  
    crop_size = (size, size)  
    max_size = image_processor.size.get("longest_edge")
```

Ilustración 10: Construcción del modelo 2

```
from transformers import AutoModelForImageClassification, TrainingArguments, Trainer  
  
model = AutoModelForImageClassification.from_pretrained(  
    model_checkpoint,  
    label2id=label2id,  
    id2label=id2label,  
    ignore_mismatched_sizes = True,  
)
```

Ilustración 11: Construcción del modelo 3

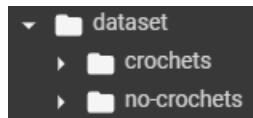


Ilustración 12 Estructura del directorio

- Crochet to pattern: Salesforce/blip-image-captioning-large

```
from torch.utils.data import DataLoader  
  
train_dataloader = DataLoader(preprocessed_data, shuffle=True, batch_size=4)
```

Ilustración 13: Definición de parámetros 1

```
import torch  
  
optimizer = torch.optim.AdamW(model.parameters(), lr=5e-5)  
  
device = "cuda" if torch.cuda.is_available() else "cpu"  
model.to(device)
```

Ilustración 14: Definición de parámetros 2

# Programación para dispositivos móviles

```
from transformers import AutoProcessor, BlipForConditionalGeneration

processor = AutoProcessor.from_pretrained("Salesforce/blip-image-captioning-large")
model = BlipForConditionalGeneration.from_pretrained("Salesforce/blip-image-captioning-large")
```

Ilustración 15: Construcción del modelo 1

```
import torch
from torch.nn.utils.rnn import pad_sequence

def preprocess_data(dataset, processor):
    preprocessed_data = []
    count = 1
    for item in dataset:
        print(count)
        count += 1
        encoding = processor(images=item["image"], text=item["additional_feature"], padding="longest", return_tensors="pt")
        # remove batch dimension
        encoding = {k: v.squeeze() for k, v in encoding.items()}

        # Truncate input_ids and attention_mask to max_length 512
        max_length = 512
        if len(encoding["input_ids"]) > max_length:
            encoding["input_ids"] = encoding["input_ids"][:max_length]
            encoding["attention_mask"] = encoding["attention_mask"][:max_length]
        else:
            padding_length = max_length - len(encoding["input_ids"])
            padding_tensor = torch.zeros(padding_length, dtype=torch.long)
            padding_mask = torch.zeros(padding_length, dtype=torch.long)
            encoding["input_ids"] = torch.cat((encoding["input_ids"], padding_tensor), dim=0)
            encoding["attention_mask"] = torch.cat((encoding["attention_mask"], padding_mask), dim=0)

        preprocessed_data.append(encoding)

    return preprocessed_data

# Preprocess your data
preprocessed_data = preprocess_data(dataset, processor)
```

Ilustración 16: Construcción del modelo 2

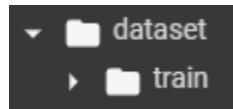


Ilustración 17: Estructura del directorio

## 5 Evaluación

- Clothes filter: Google/vit-base-patch16-224

```
***** eval metrics ***** [317/317 00:26]
epoch                =          5.0
eval_accuracy        =         0.9992
eval_loss            =         0.0019
eval_runtime         = 0:00:27.01
eval_samples_per_second = 93.613
eval_steps_per_second  = 11.734
```

Ilustración 18 Accuracy

- Crochet filter: Google/vit-base-patch16-224

```
[60/60 00:08]
**** eval metrics ****
epoch           =      2.99
eval_accuracy   =     0.9958
eval_loss       =     0.0168
eval_runtime    = 0:00:08.82
eval_samples_per_second = 214.848
eval_steps_per_second  =    6.795
```

*Ilustración 19 Accuracy*

- Crochet to pattern: Salesforce/blip-image-captioning-large

```
Loss: 1.8331432342529297
Epoch: 4 Accuracy: 28.4547598989048
```

*Ilustración 20 Accuracy*

## 5.1 Evaluar los resultados

Desde la perspectiva del negocio, el criterio de éxito principal para el proyecto de identificación de patrones de crochet radica en la capacidad de realizar predicciones con un nivel de fiabilidad considerado "aceptable". Sin embargo, reconocemos que este criterio puede ser subjetivo, por lo que resulta esencial respaldarnos en criterios de éxito más específicos y precisos desde el ámbito de la minería de datos.

Para determinar la aceptabilidad de las predicciones, es necesario contar con una base objetiva, como los indicadores estadísticos obtenidos al ejecutar los modelos.

- Clothes filter: Google/vit-base-patch16-224
  - Precisión (Accuracy): 0.9992. Esto significa que el modelo predice correctamente el 99.92% de los casos.
  - Pérdida (Loss): 0.0019. Este es un indicador de cuánto se desvían las predicciones del modelo de los valores reales. Cuanto más cercano a cero, mejor es el modelo.

- Crochet filter: Google/vit-base-patch16-224
  - Precisión (Accuracy): 0.9958. Esto significa que el modelo predice correctamente el 99.58% de los casos.
  - Pérdida (Loss): 0.0168. Este es un indicador de cuánto se desvían las predicciones del modelo de los valores reales. Cuanto más cercano a cero, mejor es el modelo.
- Crochet to pattern: Salesforce/blip-image-captioning-large
  - Precisión (Accuracy): 0.2845. Esto significa que el modelo predice correctamente el 28.45% de los casos.
  - Pérdida (Loss): 0.7155. Este es un indicador de cuánto se desvían las predicciones del modelo de los valores reales. Cuanto más cercano a cero, mejor es el modelo.

## 6 Implementación

En esta fase, llevaremos a cabo la implementación del modelo de identificación de patrones de crochet y lo integraremos en una aplicación móvil. Los usuarios podrán utilizar la aplicación para cargar una imagen desde su galería de fotos o tomar una foto en tiempo real utilizando la cámara de su dispositivo móvil.

Una vez que el usuario haya seleccionado o capturado la imagen que desea analizar, nuestro modelo realizará la identificación del patrón de crochet presente en la imagen. Este proceso se llevará a cabo de manera rápida y eficiente dentro de la aplicación móvil.

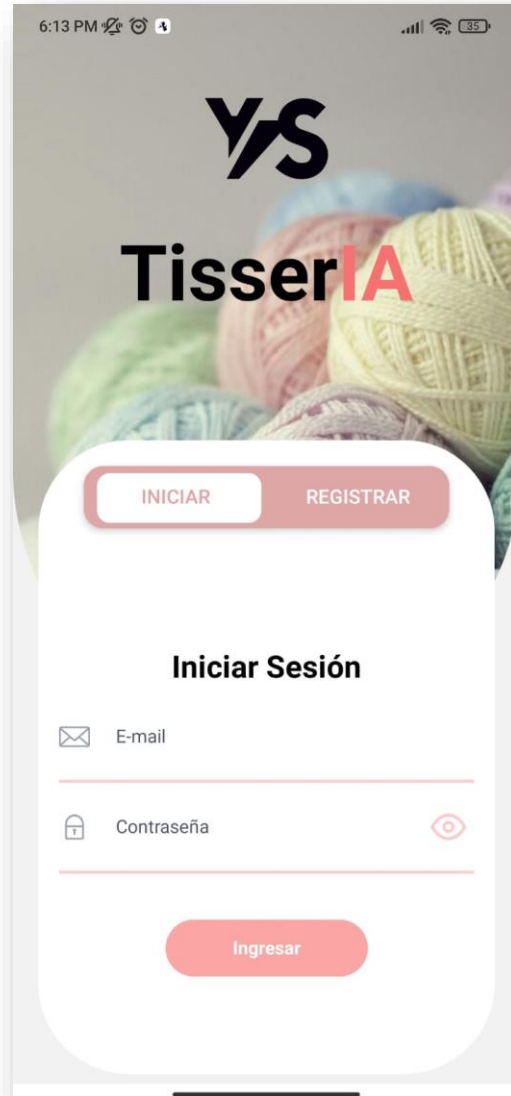
A continuación se muestra el funcionamiento del aplicativo móvil.

### Anexos

Pantalla principal



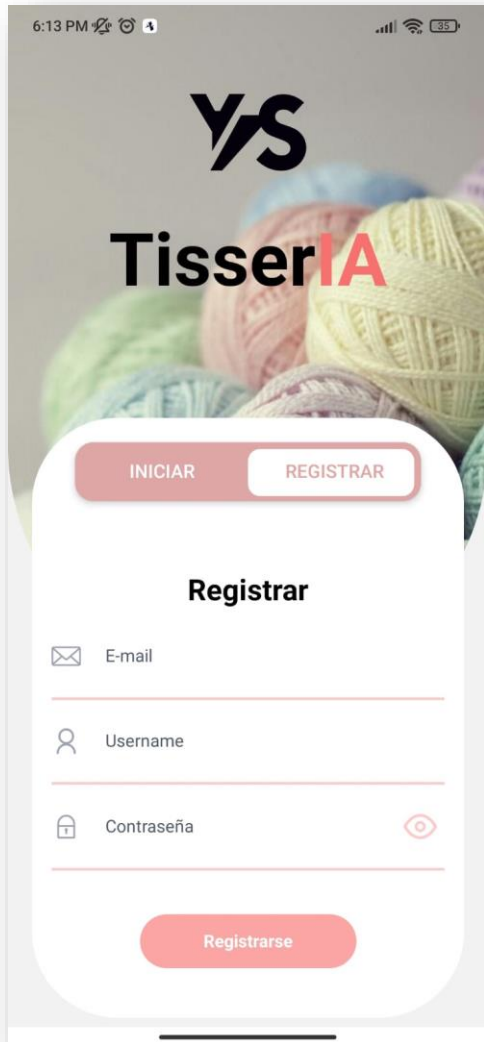
Pantalla de inicio de Sesión





## Programación para dispositivos móviles

Pantalla de registro



6:13 PM 6:13 PM 6:13 PM

**YS**  
**TisserIA**

INICIAR REGISTRAR

**Registrar**

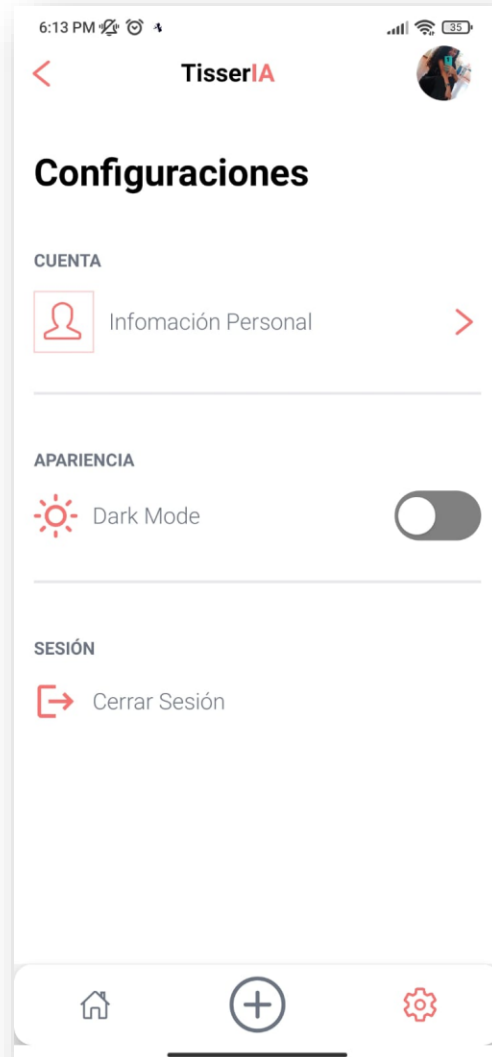
E-mail

Username

Contraseña

Regístrate

Pantalla configuraciones



6:13 PM 6:13 PM 6:13 PM

**TisserIA**

**Configuraciones**

CUENTA

Infomación Personal

APARIENCIA

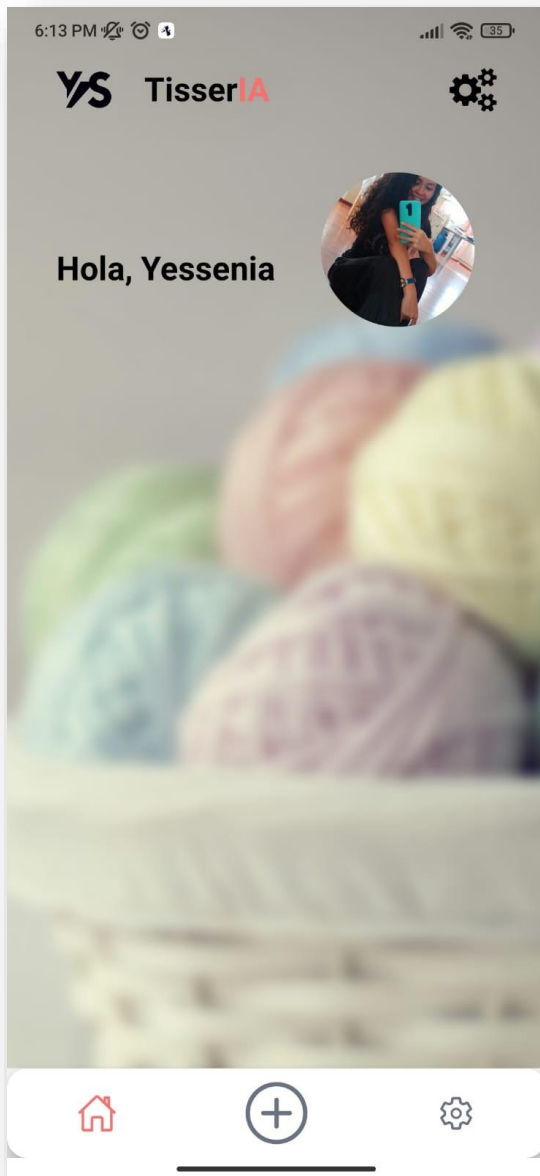
Dark Mode

SESIÓN

Cerrar Sesión

## Programación para dispositivos móviles

Pantalla de Home

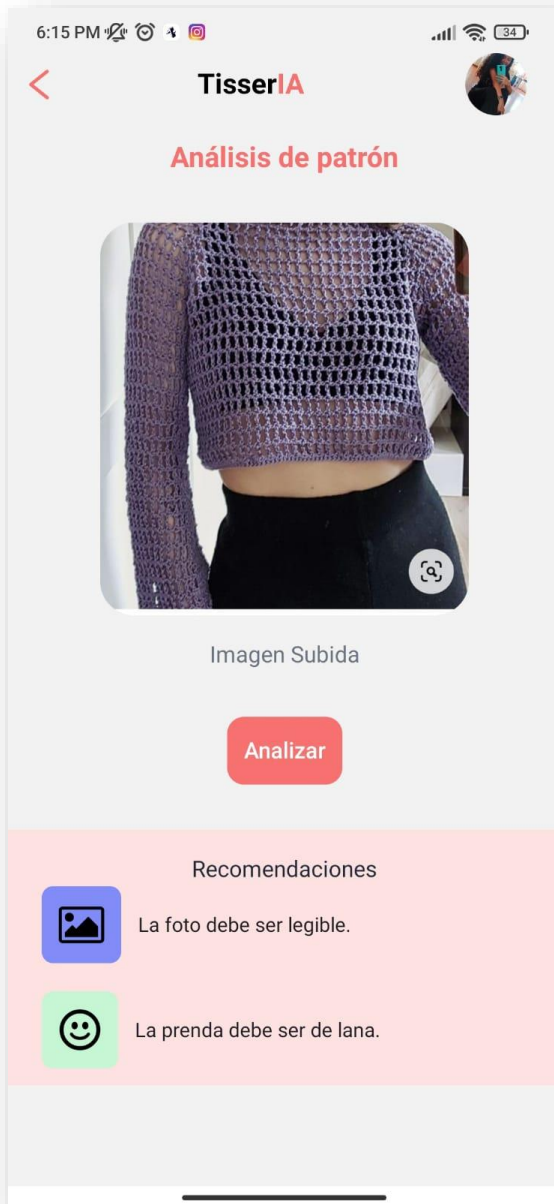


Pantalla de camapra

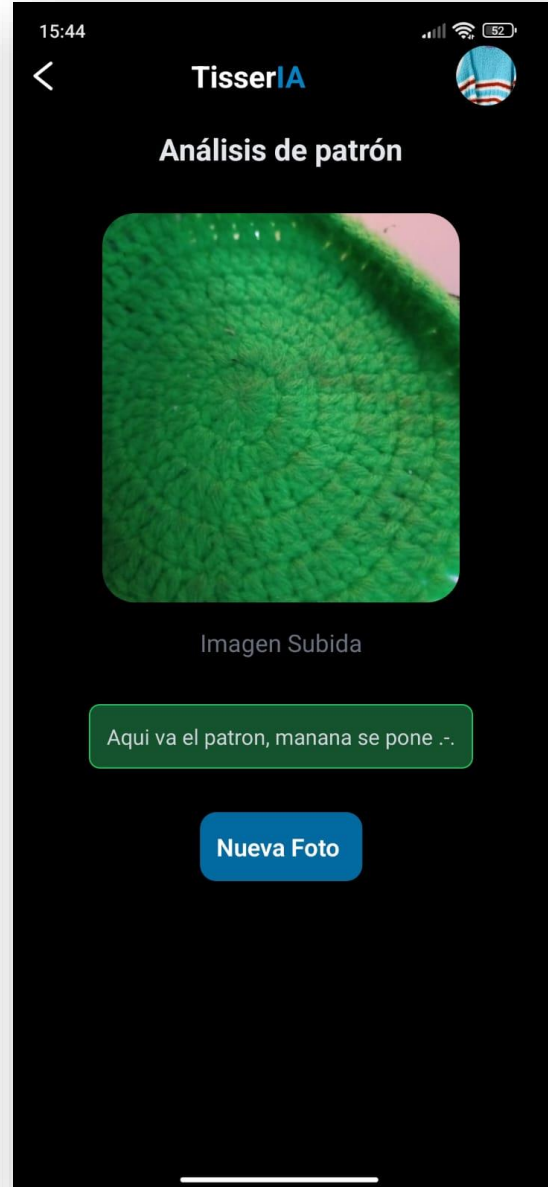


## Programación para dispositivos móviles

Pantalla de subir imagen



pantalla de patrón



Pantalla de guardar patrón

