

# Bajar imágenes de la web usando Selenium con Python y convertirlas en un Dataset de PyTorch

El aprendizaje automático está impulsando los desarrollos tecnológicos actuales, como los automóviles sin conductor, los vuelos espaciales, la imagen y el reconocimiento de voz. Sin embargo, un profesional de la ciencia de datos necesita un gran volumen de datos para crear un modelo de aprendizaje automático sólido y confiable para tales problemas comerciales.

La extracción o recopilación de datos es el primer paso en el ciclo de vida de la ciencia de datos. Según los requisitos comerciales, es posible que deba recopilar datos de fuentes como servidores SAP, registros, bases de datos, API, repositorios en línea o web.

Las herramientas para web scraping como Selenium pueden extraer un gran volumen de datos, como texto e imágenes, en un tiempo relativamente corto.

Posteriormente a la extracción de los datos estos se deben almacenar en un formato que sea compatible con los Dataset 's de PyTorch

## 1.0 ¿Qué es el Web Scrapping?

Web Scrapping, también llamado *Crawling* o *Spidering*, es la técnica para recopilar datos automáticamente de una fuente en línea, generalmente de un sitio web. Si bien Web Scrapping es una manera fácil de obtener un gran volumen de datos en un período de tiempo relativamente corto, agrega estrés al servidor donde se aloja la fuente.

Esta es también una de las razones principales por las que muchos sitios web no permiten raspar todo en su sitio web. Sin embargo, siempre que no interrumpa la función principal de la fuente en línea, es bastante aceptable.

## 1.1 ¿Qué es Selenium?

Selenium es una herramienta de automatización basada en web de código abierto. El Selenium se usa principalmente para pruebas en la industria, pero también se puede usar para web scraping. Usaremos el navegador Chrome pero puedes probar en cualquier navegador, es casi lo mismo.

## 1.2 Instalar GoogleDriver para utilizar Selenium con Google Chrome

- Identifica tu versión de Chrome. Normalmente se encuentra haciendo clic en Acerca de Google Chrome. Actualmente tengo la versión 106.0.5249.103 (mi versión principal es 106, el número antes del primer punto).
- Descargue el ChromeDriver correspondiente desde [aquí](#) para su versión principal y coloque la ubicación del ejecutable en el `$PATH` de la computadora
- Instale el paquete de Python Selenium a través de `pip` o `conda`

```
In [11]: # Instalar selenium usando pip
!pip install selenium
```

```

Requirement already satisfied: selenium in /Users/wladimir/mambaforge/lib/python3.9/site-packages (4.5.0)
Requirement already satisfied: urllib3[socks]~=1.26 in /Users/wladimir/mambaforge/lib/python3.9/site-packages (from selenium) (1.26.11)
Requirement already satisfied: trio~=0.17 in /Users/wladimir/mambaforge/lib/python3.9/site-packages (from selenium) (0.22.0)
Requirement already satisfied: trio-websocket~=0.9 in /Users/wladimir/mambaforge/lib/python3.9/site-packages (from selenium) (0.9.2)
Requirement already satisfied: certifi>=2021.10.8 in /Users/wladimir/mambaforge/lib/python3.9/site-packages (from selenium) (2022.9.24)
Requirement already satisfied: attrs>=19.2.0 in /Users/wladimir/mambaforge/lib/python3.9/site-packages (from trio~=0.17->selenium) (22.1.0)
Requirement already satisfied: sortedcontainers in /Users/wladimir/mambaforge/lib/python3.9/site-packages (from trio~=0.17->selenium) (2.4.0)
Requirement already satisfied: outcome in /Users/wladimir/mambaforge/lib/python3.9/site-packages (from trio~=0.17->selenium) (1.2.0)
Requirement already satisfied: async-generator>=1.9 in /Users/wladimir/mambaforge/lib/python3.9/site-packages (from trio~=0.17->selenium) (1.10)
Requirement already satisfied: idna in /Users/wladimir/mambaforge/lib/python3.9/site-packages (from trio~=0.17->selenium) (3.4)
Requirement already satisfied: sniffio in /Users/wladimir/mambaforge/lib/python3.9/site-packages (from trio~=0.17->selenium) (1.3.0)
Requirement already satisfied: exceptiongroup>=1.0.0rc9 in /Users/wladimir/mambaforge/lib/python3.9/site-packages (from trio~=0.17->selenium) (1.0.0rc9)
Requirement already satisfied: wsproto>=0.14 in /Users/wladimir/mambaforge/lib/python3.9/site-packages (from trio-websocket~=0.9->selenium) (1.2.0)
Requirement already satisfied: PySocks!=1.5.7,<2.0,>=1.5.6 in /Users/wladimir/mambaforge/lib/python3.9/site-packages (from urllib3[socks]~=1.26->selenium) (1.7.1)
Requirement already satisfied: h11<1,>=0.9.0 in /Users/wladimir/mambaforge/lib/python3.9/site-packages (from wsproto>=0.14->trio-websocket~=0.9->selenium) (0.14.0)

```

```

In [12]: # Instalar selenium usando conda
!conda install -c conda-forge selenium

```

```

Collecting package metadata (current_repodata.json): done
Solving environment: done

```

```
# All requested packages already installed.
```

```
Retrieving notices: ...working... done
```

## 2.0 Formato para almacenar las imágenes de manera que se puedan utilizar con el Dataset de PyTorch

El formato de clasificación de imágenes contiene clases separadas de imágenes en directorios separados titulados con un nombre de clase particular.

Por ejemplo, todas las imágenes de `oso_hormiguero` están contenidas en el

directorio `oso_hormiguero/`.

Este formato es popular en muchos puntos de referencia de clasificación de imágenes diferentes, incluido [ImageNet](#) (de los conjuntos de datos de punto de referencia de visión artificial más populares).

Puede ver un ejemplo del formato de almacenamiento a continuación, los números de las imágenes son arbitrarios.

```
perezoso-hormiguero/ <- carpeta del conjunto de datos general
  entrenamiento/ <- imágenes de entrenamiento
    oso_hormiguero/ <- nombre de clase como nombre de
    carpeta
      imagen01.jpeg
      imagen02.jpeg
      ...
    oso_perezoso/
      imagen24.jpeg
      imagen25.jpeg
      ...
  prueba/ <- imágenes de prueba
    oso_hormiguero/
      imagen101.jpeg
      imagen102.jpeg
      ...
    oso_perezoso/
      imagen154.jpeg
      imagen155.jpeg
      ...
```

El objetivo será **tomar esta estructura de almacenamiento de datos y convertirla en un conjunto de datos utilizable con PyTorch**.

**Nota:** La estructura de los datos con los que trabaja variará según el problema en el que esté trabajando. Pero la premisa sigue siendo: buscar los datos y luego encontrar la mejor manera de convertirlos en un conjunto de datos compatible con PyTorch.

## Importar las librerías necesarias

```
In [10]: import selenium
import os
import time
import io
import hashlib
import requests
from PIL import Image
from selenium import webdriver
from selenium.webdriver.common.by import By

In [14]: wd = webdriver.Chrome(executable_path='/Users/wladimir/Documents/Cursos/chrome
/var/folders/x4/___tc8h0d0rs9324_8_zvtysh0000gn/T/ipykernel_5115/1509332257.p
y:1: DeprecationWarning: executable_path has been deprecated, please pass in
a Service object
wd = webdriver.Chrome(executable_path='/Users/wladimir/Documents/Cursos/chrome
romedriver')
```

```
In [17]: wd.get('https://google.com')
```

```
In [19]: caja_de_busqueda = wd.find_element(By.CSS_SELECTOR, 'input.gLfyf')
caja_de_busqueda.send_keys('Dogs')
```

### 3.0 Definir una función para buscar los URL de las imágenes

```
In [11]: def buscar_imagen_urls(consulta:str, max_enlaces_a_buscar:int, wd:webdriver,

# construir la consulta en google
consulta_url = "https://www.google.com/search?safe=off&site=&tbm=isch&sc

# cargar la página
wd.get(consulta_url.format(q=consulta))

imagen_urls = set()
imagen_contador = 0
resultados_inicio = 0
while imagen_contador < max_enlaces_a_buscar:
    # Ir al final de la página
    wd.execute_script("window.scrollTo(0, document.body.scrollHeight);")
    time.sleep(pausa_entre_interacciones)

    # obtener todos los resultados de las miniaturas de las imágenes
    miniatura_resultados = wd.find_elements(By.CSS_SELECTOR, "img.Q4LuWd
    total_resultados = len(miniatura_resultados)

    print(f"Encontrados: {total_resultados} resultados consulta. Extrae

    for imagen in miniatura_resultados[resultados_inicio:total_resultado
        # tratar de hacer click en cada miniatura para obtener la imagen
        try:
```

```

        imagen.click()
        time.sleep(pausa_entre_interacciones)
    except Exception:
        continue

    # extraer el url de las imágenes
    imagenes_actuales = wd.find_elements(By.CSS_SELECTOR, 'img.n3VNC')
    for imagen_actual in imagenes_actuales:
        if imagen_actual.get_attribute('src') and 'http' in imagen_a
            imagen_urls.add(imagen_actual.get_attribute('src'))

    imagen_contador = len(imagen_urls)

    if len(imagen_urls) >= max_enlaces_a_buscar:
        print(f"Encontrados: {len(imagen_urls)} enlaces a imagenes,
        break
    else:
        print(f"Encontrados: {len(imagen_urls)} enlaces a imagenes, busco
        time.sleep(30)
        #return
        boton_cargar_mas = wd.find_element(By.CSS_SELECTOR, ".mye4qd")
        if boton_cargar_mas:
            wd.execute_script("document.querySelector('.mye4qd').click()

    # mover el punto de partida más abajo
    resultados_inicio = len(miniatura_resultados)

    return imagen_urls

```

### 3.1 Definir una función para guardar las imágenes

```

In [12]: def guardar_image(ruta_directorio:str, url:str):
    try:
        contenido_imagen = requests.get(url).content

    except Exception as e:
        print(f"ERROR - No se pudo bajar {url} - {e}")

    try:
        archivo_imagen = io.BytesIO(contenido_imagen)
        imagen = Image.open(archivo_imagen).convert('RGB')
        ruta_archivo = os.path.join(ruta_directorio, hashlib.sh1(contenido_i
        with open(ruta_archivo, 'wb') as f:
            imagen.save(f, "JPEG", quality=85)
        print(f"EXITO - Almacenado {url} - como {ruta_archivo}")
    except Exception as e:
        print(f"ERROR - No se pudo almacenar {url} - {e}")

```

### 3.2 Definir una función para buscar y bajar las imágenes

```
In [13]: def busqueda_y_bajada(termino_busqueda:str,ruta_driver:str,ruta_destino='../
        directorio_destino = os.path.join(ruta_destino,'_').join(termino_busqueda

        if not os.path.exists(directorio_destino):
            os.makedirs(directorio_destino)

        with webdriver.Chrome(executable_path=ruta_driver) as wd:
            resultado = buscar_imagen_urls(termino_busqueda, numero_imagenes, wd

        for elemento in resultado:
            guardar_image(directorio_destino,elemento)
```

### 3.3 Buscar 50 imagenes de entrenamiento y 10 de prueba para la clase Oso Hormiguero

```
In [14]: RUTA_DRIVER = '/Users/wladimir/Documents/Cursos/chromedriver'
termino_busqueda = 'Oso hormiguero'

busqueda_y_bajada(
    termino_busqueda=termino_busqueda,
    ruta_driver=RUTA_DRIVER,
    ruta_destino='../datos/perezoso-hormiguero/entrenamiento',
    numero_imagenes=50)

busqueda_y_bajada(
    termino_busqueda=termino_busqueda,
    ruta_driver=RUTA_DRIVER,
    ruta_destino='../datos/perezoso-hormiguero/prueba',
    numero_imagenes=10)
```

/var/folders/x4/\_\_\_tc8h0d0rs9324\_8\_zvtysh0000gn/T/ipykernel\_7757/1618089782.py:7: DeprecationWarning: executable\_path has been deprecated, please pass in a Service object

```
with webdriver.Chrome(executable_path=ruta_driver) as wd:
Encontrados: 100 resultados consulta. Extrayendo enlaces desde 0:100
Encontrados: 50 enlaces a imagenes, listo!
EXITO - Almacenado https://www.lanacion.com.ar/resizer/HTJ6j30sr2LMtxS9xVOv212bEHo=/309x206/smart/filters:format(webp):quality(80)/cloudfront-us-east-1.images.arcpublishing.com/lanacionar/P2AU56UUZJF7LGJOWIYWYR37PY.jpg - como ../datos/perezoso-hormiguero/entrenamiento/oso_hormiguero/c5094bc7f3.jpg
EXITO - Almacenado https://www.zoomadrid.com/content/dam/zoo/images/animals/oso-hormiguero/Oso-hormiguero-Zoo-Madrid-main.jpg - como ../datos/perezoso-hormiguero/entrenamiento/oso_hormiguero/0b0d587f7e.jpg
EXITO - Almacenado https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSjd9NcFQ4yk7lJLFr1ATST9q05BIOODknTA&usqp=CAU - como ../datos/perezoso-hormiguero/entrenamiento/oso_hormiguero/937cfaeff8.jpg
EXITO - Almacenado https://image.shutterstock.com/image-vector/anteater-on-white-background-illustration-260nw-1208724931.jpg - como ../datos/perezoso-hormiguero/entrenamiento/oso_hormiguero/a41949d4d1.jpg
EXITO - Almacenado https://img.freepik.com/vector-gratis/oso-hormiguero-esce-na-bosque-muchos-arboles_1308-60021.jpg?w=2000 - como ../datos/perezoso-horm
```

iguero/entrenamiento/oso\_hormiguero/3467a0ebee.jpg  
EXITO - Almacenado <https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSq5feSb8QwcbSREe-qffUn9bFltWFe3COAvQ&usqp=CAU> - como ../datos/perezoso-hormiguero/entrenamiento/oso\_hormiguero/0b7f4feb04.jpg  
EXITO - Almacenado [https://static.vecteezy.com/system/resources/previews/003/513/798/non\\_2x/yurumi-cartoon-giant-anteater-illustrations-free-vector.jpg](https://static.vecteezy.com/system/resources/previews/003/513/798/non_2x/yurumi-cartoon-giant-anteater-illustrations-free-vector.jpg) - como ../datos/perezoso-hormiguero/entrenamiento/oso\_hormiguero/e7868fbb39.jpg  
EXITO - Almacenado [https://static.nationalgeographicla.com/files/styles/image\\_3200/public/giant-anteater\\_01\\_0.jpg?w=1600&h=1065](https://static.nationalgeographicla.com/files/styles/image_3200/public/giant-anteater_01_0.jpg?w=1600&h=1065) - como ../datos/perezoso-hormiguero/entrenamiento/oso\_hormiguero/55869cac37.jpg  
EXITO - Almacenado [https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSyN7fMq5prZtI-p6\\_qt-yh3kz0Cl1HFo\\_YPW&usqp=CAU](https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSyN7fMq5prZtI-p6_qt-yh3kz0Cl1HFo_YPW&usqp=CAU) - como ../datos/perezoso-hormiguero/entrenamiento/oso\_hormiguero/770489c890.jpg  
EXITO - Almacenado [https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTW9KIj-MECBwJwGjH5soUxLdN6HXaBU2\\_T5Q&usqp=CAU](https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTW9KIj-MECBwJwGjH5soUxLdN6HXaBU2_T5Q&usqp=CAU) - como ../datos/perezoso-hormiguero/entrenamiento/oso\_hormiguero/d63c98d345.jpg  
EXITO - Almacenado [https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQgDlK5ilgN27BWIm68yhIf\\_7KOKlzxUBQYPW&usqp=CAU](https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQgDlK5ilgN27BWIm68yhIf_7KOKlzxUBQYPW&usqp=CAU) - como ../datos/perezoso-hormiguero/entrenamiento/oso\_hormiguero/c7fc720be8.jpg  
EXITO - Almacenado [https://e.rpp-noticias.io/xlarge/2021/11/10/203320\\_1172157.jpg](https://e.rpp-noticias.io/xlarge/2021/11/10/203320_1172157.jpg) - como ../datos/perezoso-hormiguero/entrenamiento/oso\_hormiguero/7d86facb11.jpg  
EXITO - Almacenado <https://cloudfront-us-east-1.images.arcpublishing.com/inforbae/YBS6PEGTDNGPDKQEW2ZJ6C24FI.jpg> - como ../datos/perezoso-hormiguero/entrenamiento/oso\_hormiguero/d090de86f7.jpg  
EXITO - Almacenado [https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTQLhdCT5AN7mwOqjgOWaZQuCp7V6\\_6r6L9tg&usqp=CAU](https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTQLhdCT5AN7mwOqjgOWaZQuCp7V6_6r6L9tg&usqp=CAU) - como ../datos/perezoso-hormiguero/entrenamiento/oso\_hormiguero/0e9952a722.jpg  
EXITO - Almacenado [https://upload.wikimedia.org/wikipedia/commons/8/80/Dirkvdm\\_tamandua.jpg](https://upload.wikimedia.org/wikipedia/commons/8/80/Dirkvdm_tamandua.jpg) - como ../datos/perezoso-hormiguero/entrenamiento/oso\_hormiguero/7e17e9a7f5.jpg  
EXITO - Almacenado [https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRuSfBLdF3-O\\_ZRiyprasnSFPuN9T6x9r5w6A&usqp=CAU](https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRuSfBLdF3-O_ZRiyprasnSFPuN9T6x9r5w6A&usqp=CAU) - como ../datos/perezoso-hormiguero/entrenamiento/oso\_hormiguero/93875a7e0c.jpg  
EXITO - Almacenado <https://i.pinimg.com/originals/5f/b8/69/5fb869301ae9da4579f645616eb7e7fd.png> - como ../datos/perezoso-hormiguero/entrenamiento/oso\_hormiguero/f2a073abf0.jpg  
EXITO - Almacenado <https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTIGH-0aYrnSBnwTHRiJ3pCR674lyMDzpyc-g&usqp=CAU> - como ../datos/perezoso-hormiguero/entrenamiento/oso\_hormiguero/7d16009d1a.jpg  
EXITO - Almacenado [https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSw9HkTszOf1hxNPND0jT\\_n-2\\_r-Y6D0qz4RQ&usqp=CAU](https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSw9HkTszOf1hxNPND0jT_n-2_r-Y6D0qz4RQ&usqp=CAU) - como ../datos/perezoso-hormiguero/entrenamiento/oso\_hormiguero/5e30f7cecb.jpg  
EXITO - Almacenado [https://t1.ea.ltmcdn.com/es/posts/0/3/6/donde\\_vive\\_el\\_oso\\_hormiguero\\_20630\\_orig.jpg](https://t1.ea.ltmcdn.com/es/posts/0/3/6/donde_vive_el_oso_hormiguero_20630_orig.jpg) - como ../datos/perezoso-hormiguero/entrenamiento/oso\_hormiguero/288e657840.jpg  
EXITO - Almacenado <https://upload.wikimedia.org/wikipedia/commons/thumb/3/3b/Myresluger2.jpg/800px-Myresluger2.jpg> - como ../datos/perezoso-hormiguero/entrenamiento/oso\_hormiguero/e5d8c83f7b.jpg  
EXITO - Almacenado [https://static.wikia.nocookie.net/planetzoo/images/5/51/Oso\\_Hormiguero\\_Gigante.jpg/revision/latest?cb=20200408204208&path-prefix=es](https://static.wikia.nocookie.net/planetzoo/images/5/51/Oso_Hormiguero_Gigante.jpg/revision/latest?cb=20200408204208&path-prefix=es) - como ../datos/perezoso-hormiguero/entrenamiento/oso\_hormiguero/1fa9024461.jp



g

EXITO - Almacenado <https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSu1-g6gf3n4NxnqIi0k0lZBczSQLNXoAsDqw&usqp=CAU> - como ../datos/perezoso-hormiguero/entrenamiento/oso\_hormiguero/70ee8922ff.jpg

EXITO - Almacenado <https://especiesenextincionenvenezuela.files.wordpress.com/2018/09/oso-hormiguero-2.jpg?w=1400> - como ../datos/perezoso-hormiguero/entrenamiento/oso\_hormiguero/ad16e38e6d.jpg

EXITO - Almacenado [https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTFgZSLoI\\_Mvo9tDoz0AxnedgJYPgwJB8I2RQ&usqp=CAU](https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTFgZSLoI_Mvo9tDoz0AxnedgJYPgwJB8I2RQ&usqp=CAU) - como ../datos/perezoso-hormiguero/entrenamiento/oso\_hormiguero/5c30218187.jpg

EXITO - Almacenado [https://t1.ea.ltmcdn.com/es/razas/0/7/8/oso-hormiguero-gigante\\_870\\_0\\_orig.jpg](https://t1.ea.ltmcdn.com/es/razas/0/7/8/oso-hormiguero-gigante_870_0_orig.jpg) - como ../datos/perezoso-hormiguero/entrenamiento/oso\_hormiguero/6c28aa5165.jpg

EXITO - Almacenado [https://media.cnnchile.com/sites/4/2020/12/131979025\\_3617254051630205\\_7734630945272557401\\_n-667x400.jpg](https://media.cnnchile.com/sites/4/2020/12/131979025_3617254051630205_7734630945272557401_n-667x400.jpg) - como ../datos/perezoso-hormiguero/entrenamiento/oso\_hormiguero/9e98621f6c.jpg

EXITO - Almacenado [https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcR7QfTijceg50zF\\_WoogmrGluwMCOF3IKrk8w&usqp=CAU](https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcR7QfTijceg50zF_WoogmrGluwMCOF3IKrk8w&usqp=CAU) - como ../datos/perezoso-hormiguero/entrenamiento/oso\_hormiguero/7e0c4ef187.jpg

EXITO - Almacenado [https://www.museocostarica.go.cr/wp-content/uploads/articulos\\_educativos/historia\\_natural/oso\\_caballo/myrmecophaga-trydactila-lisa-sanchez-aguilar-mncr.jpg](https://www.museocostarica.go.cr/wp-content/uploads/articulos_educativos/historia_natural/oso_caballo/myrmecophaga-trydactila-lisa-sanchez-aguilar-mncr.jpg) - como ../datos/perezoso-hormiguero/entrenamiento/oso\_hormiguero/3355e9f41e.jpg

EXITO - Almacenado [https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQ114V8Eq133AQsUDIVB\\_FZ0zG5tInHMPnvZw&usqp=CAU](https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQ114V8Eq133AQsUDIVB_FZ0zG5tInHMPnvZw&usqp=CAU) - como ../datos/perezoso-hormiguero/entrenamiento/oso\_hormiguero/b5d6dfef1c.jpg

EXITO - Almacenado <https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSvN69NEnXJaOwU69srCB1MggboTjps1SmfMQ&usqp=CAU> - como ../datos/perezoso-hormiguero/entrenamiento/oso\_hormiguero/0f69c37a6c.jpg

EXITO - Almacenado [https://estaticos-cdn.prensaiberica.es/clip/87669bcb-7be6-4de2-824e-267a67ef36b6\\_16-9-aspect-ratio-default\\_0.jpg](https://estaticos-cdn.prensaiberica.es/clip/87669bcb-7be6-4de2-824e-267a67ef36b6_16-9-aspect-ratio-default_0.jpg) - como ../datos/perezoso-hormiguero/entrenamiento/oso\_hormiguero/fb422ff978.jpg

EXITO - Almacenado <https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQtmzj11ONmDTsxgH5LyeCMLcdTwoYppp0lKg&usqp=CAU> - como ../datos/perezoso-hormiguero/entrenamiento/oso\_hormiguero/75081d3dc3.jpg

EXITO - Almacenado <https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTvVLu4P2dnqNa7lDbNh-mG0vwMIWJk8mRYQA&usqp=CAU> - como ../datos/perezoso-hormiguero/entrenamiento/oso\_hormiguero/6d3db35e82.jpg

EXITO - Almacenado [https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRxRPCY7LaDJ3vo80qtCR1Jziz\\_IV-oUy-mng&usqp=CAU](https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRxRPCY7LaDJ3vo80qtCR1Jziz_IV-oUy-mng&usqp=CAU) - como ../datos/perezoso-hormiguero/entrenamiento/oso\_hormiguero/56e066b084.jpg

EXITO - Almacenado [https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQEIJw7IfNKUTR2QnIvLsuZhkpFUYbEYI\\_i3A&usqp=CAU](https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQEIJw7IfNKUTR2QnIvLsuZhkpFUYbEYI_i3A&usqp=CAU) - como ../datos/perezoso-hormiguero/entrenamiento/oso\_hormiguero/0690358cc3.jpg

EXITO - Almacenado <https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRPXM00nXZnknFUEg4Hrzn0mlUSw9Y4ngagwg&usqp=CAU> - como ../datos/perezoso-hormiguero/entrenamiento/oso\_hormiguero/21d9fbfbf8.jpg

EXITO - Almacenado [https://cdn.shopify.com/s/files/1/2137/3313/products/el-oso-hormiguero-1\\_475x700.jpg?v=1571876570](https://cdn.shopify.com/s/files/1/2137/3313/products/el-oso-hormiguero-1_475x700.jpg?v=1571876570) - como ../datos/perezoso-hormiguero/entrenamiento/oso\_hormiguero/06f848f385.jpg

EXITO - Almacenado <https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQM2jyqLCRBkdH6pbsam3rhC-8s8eBEX1voHw&usqp=CAU> - como ../datos/perezoso-hormiguero/entrenamiento/oso\_hormiguero/2d8a6d1e09.jpg

EXITO - Almacenado <https://us.123rf.com/450wm/ganchaonan/ganchaonan1908/ganchaonan190800114/132603393-oso-hormiguero-gigante-aislado-sobre-fondo-blanco-oso-hormiguero-lindo-animal-de-brasil-oso-hormigue.jpg?ver=6> - como ../datos/perezoso-hormiguero/entrenamiento/oso\_hormiguero/5c34fcd392.jpg

EXITO - Almacenado <https://imgs.mongabay.com/wp-content/uploads/sites/25/2021/11/30031948/Oso-Hormiguero-2-1200x800.jpg> - como ../datos/perezoso-hormiguero/entrenamiento/oso\_hormiguero/09150bc7b5.jpg

EXITO - Almacenado [https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GCr-fzuU-1f\\_fCekkCEZK8CG42J45M3ghsVlvQ&usqp=CAU](https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GCr-fzuU-1f_fCekkCEZK8CG42J45M3ghsVlvQ&usqp=CAU) - como ../datos/perezoso-hormiguero/entrenamiento/oso\_hormiguero/4c20c50d48.jpg

EXITO - Almacenado [https://cdn.elimpulso.com/media/2021/07/210731\\_Oso\\_Hormiguero\\_web-696x696.jpg](https://cdn.elimpulso.com/media/2021/07/210731_Oso_Hormiguero_web-696x696.jpg) - como ../datos/perezoso-hormiguero/entrenamiento/oso\_hormiguero/caafcc95f1.jpg

EXITO - Almacenado <https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRF1EKp0WLGLE9iK0OV4RTuMgQyltCot053ag&usqp=CAU> - como ../datos/perezoso-hormiguero/entrenamiento/oso\_hormiguero/38cbbadadf.jpg

EXITO - Almacenado <https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTl0GCugU064YSubaA47KJfTKgIEl5RB-1y-g&usqp=CAU> - como ../datos/perezoso-hormiguero/entrenamiento/oso\_hormiguero/c03142d4d6.jpg

EXITO - Almacenado <https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRScgBIgMx14Hc9ZRhPFvguUri57m363CWexQ&usqp=CAU> - como ../datos/perezoso-hormiguero/entrenamiento/oso\_hormiguero/bc52ceeb87.jpg

EXITO - Almacenado [https://upload.wikimedia.org/wikipedia/commons/thumb/4/43/Oso\\_hormiguero\\_%28Myrmecophaga\\_tridactyla%29\\_%288697865538%29.jpg/1200px-Oso\\_hormiguero\\_%28Myrmecophaga\\_tridactyla%29\\_%288697865538%29.jpg](https://upload.wikimedia.org/wikipedia/commons/thumb/4/43/Oso_hormiguero_%28Myrmecophaga_tridactyla%29_%288697865538%29.jpg/1200px-Oso_hormiguero_%28Myrmecophaga_tridactyla%29_%288697865538%29.jpg) - como ../datos/perezoso-hormiguero/entrenamiento/oso\_hormiguero/8a9d24ea56.jpg

EXITO - Almacenado [https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRVFzSBqZ\\_hojQDrrQLL84Wr3I7Dy0NxZWoaQ&usqp=CAU](https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRVFzSBqZ_hojQDrrQLL84Wr3I7Dy0NxZWoaQ&usqp=CAU) - como ../datos/perezoso-hormiguero/entrenamiento/oso\_hormiguero/8d589d2748.jpg

EXITO - Almacenado <https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTQ2SdvAAVUmsb2iS6WkHC8Qf48QqeUJlAVeg&usqp=CAU> - como ../datos/perezoso-hormiguero/entrenamiento/oso\_hormiguero/d513b431cc.jpg

EXITO - Almacenado <https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSTxjUPmx0yQ-Vvc-klbMwUVgbuJJBnJqEzng&usqp=CAU> - como ../datos/perezoso-hormiguero/entrenamiento/oso\_hormiguero/539ff656df.jpg

Encontrados: 100 resultados consulta. Extrayendo enlaces desde 0:100

Encontrados: 10 enlaces a imagenes, listo!

EXITO - Almacenado <https://especiesenextincionenvenezuela.files.wordpress.com/2018/09/oso-hormiguero-2.jpg?w=1400> - como ../datos/perezoso-hormiguero/prueba/oso\_hormiguero/ad16e38e6d.jpg

EXITO - Almacenado <https://www.lifeder.com/wp-content/uploads/2018/10/oso-hormiguero-1.jpg> - como ../datos/perezoso-hormiguero/prueba/oso\_hormiguero/0ba57844a9.jpg

EXITO - Almacenado [https://t1.ea.ltmcdn.com/es/razas/0/7/8/oso-hormiguero-gigante\\_870\\_0\\_orig.jpg](https://t1.ea.ltmcdn.com/es/razas/0/7/8/oso-hormiguero-gigante_870_0_orig.jpg) - como ../datos/perezoso-hormiguero/prueba/oso\_hormiguero/6c28aa5165.jpg

EXITO - Almacenado <https://www.zoomadrid.com/content/dam/zoo/images/animals/oso-hormiguero/Oso-hormiguero-Zoo-Madrid-main.jpg> - como ../datos/perezoso-hormiguero/prueba/oso\_hormiguero/0b0d587f7e.jpg

EXITO - Almacenado <https://www.equilibriumx.com/wp-content/uploads/2020/07/oso-hormiguero-1200x800.png?v=1635822443> - como ../datos/perezoso-hormiguero/prueba/oso\_hormiguero/3d8d033ae7.jpg

EXITO - Almacenado <https://upload.wikimedia.org/wikipedia/commons/thumb/4/43>

/Oso\_hormiguero\_%28Myrmecophaga\_tridactyla%29\_%288697865538%29.jpg/1200px-Oso\_hormiguero\_%28Myrmecophaga\_tridactyla%29\_%288697865538%29.jpg - como ../datos/perezoso-hormiguero/prueba/oso\_hormiguero/8a9d24ea56.jpg  
 EXITO - Almacenado https://t1.ea.ltmcdn.com/es/posts/0/3/6/donde\_vive\_el\_oso\_hormiguero\_20630\_orig.jpg - como ../datos/perezoso-hormiguero/prueba/oso\_hormiguero/288e657840.jpg  
 EXITO - Almacenado https://static.nationalgeographicla.com/files/styles/image\_3200/public/giant-anteater\_01\_0.jpg?w=1600&h=1065 - como ../datos/perezoso-hormiguero/prueba/oso\_hormiguero/55869cac37.jpg  
 EXITO - Almacenado https://upload.wikimedia.org/wikipedia/commons/thumb/3/3b/Myresluger2.jpg/800px-Myresluger2.jpg - como ../datos/perezoso-hormiguero/prueba/oso\_hormiguero/e5d8c83f7b.jpg  
 EXITO - Almacenado https://static.wikia.nocookie.net/planetzoo/images/5/51/Oso\_Hormiguero\_Gigante.jpg/revision/latest?cb=20200408204208&path-prefix=es - como ../datos/perezoso-hormiguero/prueba/oso\_hormiguero/1fa9024461.jpg

### 3.4 Buscar 50 imagenes de entrenamiento y 10 de prueba para la clase Oso Perezoso

In [15]: termino\_busqueda = 'Oso perezoso'

```
busqueda_y_bajada(
    termino_busqueda=termino_busqueda,
    ruta_driver=RUTA_DRIVER,
    ruta_destino='../datos/perezoso-hormiguero/entrenamiento',
    numero_imagenes=50)
```

```
busqueda_y_bajada(
    termino_busqueda=termino_busqueda,
    ruta_driver=RUTA_DRIVER,
    ruta_destino='../datos/perezoso-hormiguero/prueba',
    numero_imagenes=10)
```

/var/folders/x4/\_\_\_tc8h0d0rs9324\_8\_zvtysh0000gn/T/ipykernel\_7757/1618089782.py:7: DeprecationWarning: executable\_path has been deprecated, please pass in a Service object

with webdriver.Chrome(executable\_path=ruta\_driver) as wd:

Encontrados: 100 resultados consulta. Extrayendo enlaces desde 0:100

Encontrados: 51 enlaces a imagenes, listo!

EXITO - Almacenado https://s1.eestatic.com/2021/08/30/ciencia/medio-ambiente/608201208\_203739946\_1024x576.jpg - como ../datos/perezoso-hormiguero/entrenamiento/oso\_perezoso/d0a9cacf5e.jpg

EXITO - Almacenado https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTa6fUVAKh3cN2YaAcJ5103Vdfg2LS4gJmLcg&usqp=CAU - como ../datos/perezoso-hormiguero/entrenamiento/oso\_perezoso/e71cad3afc.jpg

EXITO - Almacenado https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcR\_uMBqrYUScgRUDmiFtfp41gUZvOfSwSz6ow&usqp=CAU - como ../datos/perezoso-hormiguero/entrenamiento/oso\_perezoso/5821eb90c2.jpg

ERROR - No se pudo bajar data:image/jpeg;base64,/9j/4AAQSkZJRgABAQAAQABAAD/2wCEAAoHCBYWFrgWFhYZGBgZHB4cHBwcGh0fHxwhJR4hIR4eIx4hIS4lHiErHxwcJjgmKy8xNTU1HyU7QDs0Py40NTEBDawMEA8QHxISHzQsJSs0NDQ0NP/AABEIAKgBKwMBIqACEQEDEQH/xAABAAACAwEBAQAAAAAAAAAA

AAAEQBQIDBgABB//EADoQAAIBAgQEBAUDAwMEAwEAAAAECEQAhAwQSMQVBUWEicYGREzKhscEG0fBC  
UuEUcvEVI2KSgrLCov/EABkBAAMBAQEAAAAAAAAAAAAAAAAECAwAEBf/EACURAAICAgMBAAIBBQAA  
AAAAAAAAABaHEhMQMSQVEycWETUoGRof/aAAwDAQACEQMRAD8Ae8WzCwNtwJ+UDx8t+VKvjIrogUd  
T4j+1QynDi7fEd9bkksTy7Acq8ZGDs025DLXkykm7Z0J0EY+eYQDyHL6C38vVTYzSYJFjaedUvj  
xC/iT39KLW8tsxFwIoN2PbBEzLlJcHpeDM87b7VNQRuk8ib9I+1EFew/WeVUjOqJUSQepHtfnWW  
QEjlyU1LNjtz/mlVPky4kWM87Hr61bh5jEXYqydVMHfmNrUdnc2mpIHUEEWN7UaM0BY2MVUHSwQe  
FvDN7Rtt35UK+ECQQANO7A3O29rbfWrV4+UYyngm+/v1mKtfGw3IdNtWogGJHMEd7+1ZqgC4hRmk  
kkBgw9pIvluKqzeGGzAosqFF+hFT40oGKhRyELzbkGW4+htVPwwzlgwGmCJ3YChJVLtfhCTqVhed  
gF3CxHWbSPL8VTlCiviIyGAluo6Wop8cN/3Q1wbj70PmcyBmS4EgoDHekfZpv+P+jSblHsaPXiNg  
BkSchlm2Nt5HlSJ/1Bi4jhAQWWRpAie59KZpmAyMiuRrQjEEFv6Z/qjLWXywCYivN10iSPmEHST  
6WqjgprTlkEk2kDYZkyRM/emuSSBtS3DxwDGmfFM8r8opzhKzHUBHYRFmx4oozIBs0AEKXrOZ/Mr  
cLYGwim/6mxygWFltRkHy5lmgCdWsLuQNX5d6aEV+TJSXi03AMAYuEZPK1toql8IHHbBNhoU7+c  
1f8Ap/ShIR5W29mHYiqMvhs2exRAA0j26ig45bC4qrLP09ksMI7yV0FtQJ5copzkmbQHw2hWbs3T  
/NDcS4ew0vhAsWIDiwmRHtS5+MIh+EpKRFjc6toHrU5cb5GmwOKbyPs3mXVQWCGByYG3kaTnNeaQ  
y61NoAn3obH4gqQ2OPiKWIMWbTNzPWR0CJiQPEjjUkHZSLAnrSrioRoZcJxMJ00yAGAIuPTpUOKO  
iEamQOegEQKry2Pr1Ii2IMED87V4nDQoOoglqQTsfIVJxUXTW/8ALE/aBMTMa0IF4043os5tvg6C  
gZtgxB1AedCYWKIME+HC/wB0iP8AmrscOHDq+pCI0j1VF2ukq/YylbopGAwQkAfc0HhA6wCZU+Hy  
m1MsvnSbQCwPMkW/NFZ+QBIMhIJHjC2n02p71f1Q2mILJYCdUCZgGqyCbqDa8c+81dneLMuoPEBi  
BbeicrxFHwX8L89Ip3KXqWFXf8CZwjGRm4HQf8ANW5HPuhKnXKeQ2j969zGVUEjDe4kjuCIHSh+  
F5MrqVjEAKHeem1VfVoDQ6ynHMFV0hU0j1G/WjJnsJrnBSSBptSfL8NcrqVWPpRDo4t8M2A/p7VF  
8MLuhkbV8NQYEA2Jjy5n1+lKMR1awKCTAI3j6xzphlss7FzaSBfkDymrMPKqol9M9du/W1Io278O  
lCwKFKkggnsfr0q/MYrESCnyLzsPLnvRqsJnwlSFsTBJ9Cft2rs1gKotAAvP+aMk0hmKHJvP22iI  
I3EmfzRSIJL4T4ovBBivdBIJKkdjvHI2PQ1dw/KrqkOyneAZn3BtSJ269JXmgfDaRGwPW0XjaoKn  
iM2IiNr3/wAU/GS06CAGgcxI8496tbJKs+ERF5/c7VSMJJGutMYzOCGLWGmxkHtfbyqnLIVSCIA7  
biefajsxkheBBELSCfIekDyqCKFUodxzPOazfg1CXOup2MFSO/ODHlJt2oHAyrgTqliLMDY/zpRm  
dyelixWwuOlXNALXwkJTE0M99ESrA+c0/LEZ6phuQwjoYMYLXtE+1JXzmIjgpEkEXvaaY5HQ4K6t  
D/2nwnzDXH0r08NXWA5uLwlie4b5T70Vi0x1bjRHL4mO+MzhmeTL6RYWibbVHDZkzC4eIhIcEEkz  
fl/O9PMvmsTCQIihJ0lh/Ux8opVxaurKXUg7Ger6fY1lJvFCJNYZRl0h+wJ9ga0GVQWidqWuwYpG  
5v78veaa5VTMaYtSS0UiqFX6hQlB/urP4R0kEiVNjPSeXetP+ppXDUgE+OirPPlcViCMNgp5tb6U  
YJ1ZGcX2wHnATDx4w8RWEcx3vynatHiz8ItYMN7eKORnpWSy/BcySSMGR/vW/TnWnw4wmXGV1IX  
wnl0+YWPvTyxkpG0soDfOIqHDeJhZQfee9VthYcsCFDsLM4FiOh60LxnF0NoaAdAKuqjxei0Wv50  
u4Rj6w4xCToBIJ/qjLSKLeUSk7eDs/1lLqhZWIjYzfnTHAw1BtACz/VM9qC4UinFd3QMq3VSevSn  
HwEnVGm0Wv6zzN6MrWDKLsDyWcxMJGRV1Ame1dmWfUhbY2tt3Hnarcfn4aqUQlr3JG3WDzqzA44j  
4L4WgO4ICtHLMsa107YKBzmlcqjQNIJA0gAt6+VE6zAPw9KiNj1EmxtMV6ugYRDAGTIt8vkaGTPs  
gRdKECRPPeRPI0kk9oVprKCM1iuzr4Anpf309aX4mjBdSJGg25G1Jm4ijKEZQW97/tRuW4iCpV1E  
Ed47Dt2NBckovKGSx2aybl2AUgWIIbVInYede80loxwytyNiset63WDxKEZMJQmlZ1nSWPeDWLbi  
au5Lu7OTdoAqim5LQ3oJnMwa09DE9f4K7CXQRFSZ738jROYyKOpGGx1TIVrEnnBNB4+X0gLctZTH  
lIHnePSnTTRmeZbHdiGLMDajUdx2q7OYbl2Idt/7zUslgsphkIuACR9KvzTy7RAv0rN3oyZulxHV  
G0nxR02vYR5UImYYrLGSNyQPxtalEr4gtjBBPQDL++1B/C0nSviIkwe87CIiubs9HSnk5XYmLGat  
wxAg3Emx9KrXDNrRtv3qKrrgGCJ7isngbwLfmZMAXsd/KatyJAYMeR85HX60pORDHcHtttli/PrR  
uVymnxXkjaTHsfvR9BVs0K42rpv528uVe5vNBEZmBIUTa9Z3MiFJkqbXULW9IvVeHitiAqcUsOhJ  
36d/8VTu6ElEsHEEXAWVAhG9r32He1U4uPqElbXuLz7XhtVPw3R+QEwOm3UfmmGF13ZYAt350mXs  
0dzO4q6vlGIgEC/pt67e5rIYUOqO6wwEAjmbIFvStLmcNWBQgggPaOQPtWawsDWvwbySAdhc8/T  
bvVVMiKo4Z2KmE6MWlMTan72BkbVRw3iLoQrr8VB6sojcvZgcPLnSu/OT71HHGwnhhz3mxpIt6W  
SEXK8Gi4YqOgbCxs4U2BjUkiI8q7OZHwrgGGaLm4kfbbSrPZDQ+rCbQ4GoMDY9iJvTbh+aZ0Bez7  
MBtPP339adfSkrrWZgIyEBhBWR6Hp9aZcPe9z97VfiAEUsbFYnSkhmNGPP8AnlodW2Pxysa5jiSo  
LKCQd9z6dN6QZria06lgbm88o7b38q0WQ/Th0+PeOd496pzP6ZosWXT2ERbzbq0Ulos38BOHZon8j  
aT/aDv8AiYvWgyWYaOv0n05Ghst+ldJVxZhazHb7Uxfhj4aaJ4wLxzHrWaQUxXx7hKYiFlQsw+QC  
2kze3TttWOPCHwi5chBEXPqbVvXzUAPcXAkElx9I7ljP1flFDq+tn1hnAkQCY5ch/mppNPGiU4p0  
0J2zUTA3PvTnhucUnRF+U7Gs3gDYsbglOSbyd5FGUVJUTu0X53GCO8rJDHy3qvK57FwW04aBwKRB  
APloPMkkkkMxJmW/PWi+GIDquAwg9Pan6pRyTGeNmbAMdDXmIM/715eleZdGY+FLYcmm3r0qnMYL  
oEdk1LJLTZjy896Ez2XcenFCCJ8LEHreTSKKaoZPBOP+n4oJZakMoKktewg+1TR2KMMQXMD/AGmb  
GRyilmWOIyIysuqDMj118qLBbDVwyypELGwPY8xvUnBgx4FPhozh3Ui1tJB5RB3oDPPJGqNa2B06

S3+4fml/F8fSWAEAX7jYiKM4dx0lAuMq4qnafmA7HnTw4nFXyVfpg7EzBMFhHl0r3L8QfWCN43N7  
jb6VdxjLqgro+G4fDcSs/Mu0qfKgWPjILAwZA+tU62sjPIfluLYr4mgsCJ6CPtVkObhQZ59aVZXEL  
7bwTPpRGFmcUADb1pXBLSQMfDcPmQXII1AN8sXFzB+29VY2OxJM6B0Av0N4k3586Ey4YPKQSzGWJ  
iNxYc79bV2IXBJdlm+51n0AgA22rkSfo6cvSS4TESSTBBBJNWrlD2VSbiRf89xQeJiAz/wB15ttC  
gbchBofGwhZtbA9S7c/M/SqLKodNhrZVgxa5n+kC32tVyGLkFSJtPnVGRd4+cm8Ac+x3qzNhwssa  
wF5te9xH1pX2s1tFuNhKfFLSt5Bb7fzevfitpOkzB5ib+1KmYBmtaJXl5HeNqc4WKCoYCQYJI5c4  
PpRdmuy3CxnZTqAsREH8elMsLOYQANJEgi03vs09KczjHAMQoWG8C1iofe83qODjo5+INSG40sV  
tYWN+/KnSdWa3dE84SxkXtvY7bE85tWfUupMxZgZt088yBT/AAsBYc3sepPLeZ2rMcQLBrSRv2/1  
qMG7M3R7mCrX4l1IbXGmoZvEJVA4brqteeopNh54lyGLGbagbD6bVZhZxT4GktyP71VQdEUmOERA  
mqSvc/4JNEcELKTJDI1tSmQDyPUdD6UixcrNy7SbBZNqb8Ix1GGNII0iCetDqvASwmaDEgG/8FK0  
4kmWdsw6lgDoUDrJ+wFC5ziLC3K3saR/qjNyMNBsWL/QAfSnjG3QeN1Z9N4F+rsHMtoGpX3KsOXU  
HbnWhfMp08xXyHgKf6ZHxZBcrAg7C0jzr0/rv5skm/n53MbVRRXhW/p9D/U/6o/0qAjCL15C7gAx  
zP4rC4H6/wAyGDsUfDPzppFhMRtPlo3LcaXHw2XF06WvDbClWd/TuEUJw2Yzex1Afn3rXH01Pw3W  
OuG2CGwzOG6TE7Wnff8A4rPtIU2WfQssrkBjPEcec7i5pP8ApjN4uGWwH+UjS03Qg+VP+FYJGUOH  
Yu6sNue4Mc46VCSpBlmJlWwtIPxFGqCdSxHrFqVtmIcEDUFO1xNHvkcRdWkJEcZMbj/cpuKFR0ct  
rYqDniBz6Wp4l+yFkczjq7wzsik2nZek9aGRG1kMYyJwHYNCZ2iKu04QU6pZt1I5HoakwcwjqVx  
HiB4VEST07U9lpGQzKMqgcR2YKSSlp/+LGxrx2V0dMRwyKRpgQYMW0jmOYonLZk4mAqMQmhvEWWx  
WCCL771SMXQACEcSIKGSrYMGox/s0q8E3GHbCfDRHlQgutizJn6Vuf0vj4eLhFGE1BDahfreszmc  
dy+nQ0A2aFI/+tT4fiBS5RyjtAdSIBg3IO01TtjJvAjMsQ4hUdJKlWF7G0H/AGkUt4vi5cnSoKlQ  
DAjwHmJ7/ing00qTIBiAw6jeR5GsviDFkh1BDSJUD3oQyGndjTgj4DEI91bY3BDde3SrjwB3xGfC  
AZBYEkAmbH4rP5XCKOJJBXYciKtGaxQxc6lF4JmIvsNqPRqTae0ZJphqZLFwzodIsxFwbxbamuV4  
UdC6nvF9qG/T+XLicw8swNscBH0orDyWDiAYnxGXVeDNqWTUdjJIdlGE6XCAC5IEAHvO+53oNsUL  
8g1k7vP8+1NshwzWGBBcksBAmOp7VHM5QDZCItLAjnymuOM1thT9YlwSzeACDlue/wAxmiv9HJGt  
je4BP860bhKASBvfl+a9TDYwQbjsPvTSlWh2kiOFkkWFkie+1+v1TLHRVS5WIsW2360M4kXE9e9B  
4yQAumwkDt+1NGVoasA2aZFYqjXHMxHnMH+GistnfcFZDPVVMH81Ti61HzKoI6XMRavQwLgAkTew  
Mz5UdoXWRoMbCCOhZQW3DjSte2/kKHGC2HqKRpcQf8e9ChmchXpK8iYgdt95+9FpglgIteevf2NB  
2jbKy7qoAYet579TQOawFdgXBC3JAmbDYc4/emGJgmDOW5dvPfakvEXdfkZgNovY9ibiek00MSDT  
rJ6vDi6wmEwXlqt5eda4HAHL+GFjdieY351cm87FA2uVAg3ghLYfej8hw5MMs2IWPmCgDva96sml  
6SuhZjopEkkONtNy0dutMeFZFlSwNUSb7wPsSapTgZxmXEDkLJg7W3sOfSm+YcJh6EnwjfmY6mlc  
vjyXjB03L0zOfY69LAhtiCI8qXvhywlbrJA6loczxBXdf68z1j1XmVw10F3AMXuBysB9ftTdmtoi  
uOnaZmsTPh/CwiNiPzQb5Yk7j70xz+GriV+bew3pUUCda6IvGAsa5HFCL4oB5RMnrvtrQ4oxICQo  
g2H3796z7hj8u9EZXBxAQQdv2gz6HahKKcmzYcNzBILncDpz2tTpVP8Ap2gwQAY9QRfONJOEYDBF  
Ui035PfetXk8raDbz+ornkiyVqjEHi7YjbtYTE1SSFA1MYhta4obiGXV1Z9IDneLISNyB1NUcdyB  
XMvhhAgDQOQI5NO0Gd6hm+HosKc8ryZKqrkf+0QarlimqOSUaYG+EQhbWEEKdMTczjzn50flfhKod  
11E/KoMgdyA4n6YV8PXhM2IAQNY27ggb351nHyj4eI2Fq0nVF9j0rKSmmkxW7GSZxcV1DMSS7AX  
XoADY1PEYjAwYtspuNvzQGCzFjCDUSNthG9PcAsBDG29zIk7n2pZLroZJs0PDuHkLqZ1A0DkDJ70  
r4rw/AKNiI+1lIDr002mKIyeWUAFcdizX0kypINx2tep8S4OmLmCzmG03C2DR1IpFXbi0dgoHlyj  
IreIOsTy8QtHfa9LXcaSuspEXYT2I7itWmUw8bDR0lGQAJq5Rs09L8zwnDkM5Aknw6oDTuK3amNK  
SukJc3wWUXFV2ZtiFFvMU3yP6cZcDRiNOu5EyQIsI5GrcLMukqmCAq7AST7xApjg5osoZ10E2IJm  
Pas5S1YrTRB8IooRMIEBdMzy/egcLh5AA0taruKZrFRVbDQmpnxEmLb7VR/lodvesoyMoyNdLcPw  
AMjm0+AgMscwCLnnXuWzLsSuHmUxgN8LMLpcdpPOPKq2UgjQDq84H88qXZricmMxhDEtbX8uIt/7  
139ahwtOJWLwP0ySNOvBfBbmVYat5XMelQXI4JgKHHk3+KrymMhwLOHiYjLyD7jtEVNX5HfnEmpX  
aq9I/CtJnr8FR18OI6nl035xE0JmeD4iRHjvbiN/YmTRKSLqRlvJ+1qv+04iIA77nyAmIFF8cWhW  
jOfiJlGBvZhH1vahizG5UqOx+vWtVmMP4nhdQw2Oobc/MW70nzf6eGoNhqYPUnb1pHxUBrAuOX1J  
pgsx3tYjp05qWSyb4SlbaTfxMTznmbD/ADRGDwXMggRI5nwgDnl6UY3AMdrkr6kCfvPrQ6yqgIA  
x8e5B0mLHuI5XvSrj2XD5dinzdDoI2pjnuH46C+GYHMR+Deg8zhttpYysmxJn+cq0YuLsDuhdwrM  
OERVdAWuWzTwnTI2HSnGiYOIQWakgVInc9eVJMDLrhrKkqzEhrdzYdKapilkAAaltjJsR+KabzZL  
CyMMY5kAfLyAFvIUPjZHFxCUAVF2LMPsBcmg8DHZX0FjOk6BpIBYCDiabyJ9qIfN6kLM8qPmg7dj  
F6FPwMuSTsbM7xHIDaxUnGR4Ia0yDO0XM/SmWfhstilBFz9v3FE5fJo8oiIqWYaJJ9Db1NeZKNDg  
lidQsYiR5AU7t1/A0ZdnVGU4bwtlTViWziFCnkOc96PweGoQQBc8yNv5FMjhaidWwuPS5+5ojL4A  
BHQ/f9u1Vcm8lVBIW4XA1g22EfX9quy/CIMASbW9tzH2p+EjflP88uXej+E5eXBilATJ+vtSuTGp  
IVYeEyoFIECIH5pwxKhWPPke4t/zR3EcBCw/uUift9KJxcv/ANsCx0g26g3EpoUEzfGsor4HxNQ  
RgQrOs6o2A0kibedYHiuXy6CFbEZ43KAD63r6flVJTFXcalj1A5cqUBFxcPRj4LLIIGpZYR/UCNv

Wl7dXbOf1TTs+ecM+O5GCuI6IAWADEAHfYdTtf8AS+MmO64OZOpdcq5s2oAwRNvpIke1aDK5ZMat  
8JS7RPjHiYWsvKlfd0xMzjuhTDwCpBgoVZgTtMwx503dSi3RHNmgfCyxchMJbWLKIHuN6GznDFZd  
YECfl6dqfYGS+Agd2DIgJMLeAOOnOkuPxrDIBELAJcxPM7L5xU4JydjZSE44i2A40kkRpgLIHn0q8  
Zc4rkk6UEeKZ3vA6UFitiMdZU6XMgDmK7M8UCK6QVIEgbGSKbN0thcqHWYz6YCBZJtYbnlpZmOKY  
0K6Jht4j4d2PQg8rdqWZBsJyVxXKuWfpuOldZnF+G4ZRghiBuZAAAnzpowskyjeWaLG4piPhNoAR  
0uw3gDcVNcyThLjtAk6HXCg0yOludQ4M2HilcTCXQfEMRCLmQRI/uEnah3TGOAcPDwpRjN5BXy5R  
3o1FKtBVUaHgwDI5SSIKoQIPSOlqt4/DAWJCLf8AnSgeF42Pggh7sLAEgeH03501/wBcWvp37/4q  
TnOLpZEc5RwaBmaAp3Yct7cp7ikWdcH+krExtAFvpTvMYbaDqbUwMgx0pVm8ZgPlkX353/xXPDZd  
Oti3B4iUYpqhJsbyDub9OvpcU6wc7YarT3t1nuYuKQ5hG3A7dx3+lAa3SLllvY/iuuMvAqTWzf40  
aDW5XJJ63/yavTNSJiDcgeQ5/esXwviJZWlvfJn16e5rQZbMSFIG9rf7ace7HmRxYQAm8+Ik99/U  
xR+C8je8C/0Ples/lSG08t/Sbfb70blczYGZlW5dNv53rGGaYoJLTikgwZII8JFu4NDZl2Q6gCV  
5sCbdPDMXrxGAU3Hjk9pN/vNUYgYqVVocNKzz3kdwB270xgtM4n9bW5HxQZvvUcxg4Di4kDb94/x  
WexeIPhtp06iTZSYJkGNLGxPh08fkmpmiQdLMjLurgKEESCH0zEHe3TCUKsFEOIcAdj/ALZ0+dh9  
FP3pBj8FzSMNGkiRrggyOYA3mOdq1OC7wNukEXKywB/3apBonExQFJ3i5kSR3iJ+9DqvRHCLM2yN  
qGsDQviIYXtf0g3pFg5XWWXVGoEBLXeb+LtWlGmMIDpVXBEEjsfTrUHYiYaeFFQDuAPcnekUKMo1  
hGay/B8REGH8QsJmFW45ke/oPWvcdSAEW+0xf3PpTHNviHwoFRd2LMR5yQpEdjVGDlX+Z0Jk2YEN  
af8Axmev7GqUUjFI9w8AaQ0SQfoeo6lcmUAYmDztvt5SKIXAhd5Aht1m3IGp40nfYHeeX8NAYrw  
cFmWVE29R6c6c8Mw10hogx7H9v50qPDsMB2InrH3HnNeYWMUc28MzHY9B6+9KzEcW2Ix3v8Az6De  
naqGtTSDNXckHew7jce23pTDhWOChEyR9qWGePlQmIzDZ0W3KVgD6VlflBxDM4OJKkhFbxMQCrCZ  
AiJBgitfmcY68McmDfj+elZTjHEUTM4mFieLCcLq38BAABn2NKJPRHPcRR0DFVeD4NBIA2BteoZ  
fDZyAUsOTG4n67UKuSK/I6sm4YLHoTsDSzW7FyHYGSCSTB8jSVK6Wjnbzk+gY2VjCZFbVKsAWPUG  
xr4yODYiFg0gobqdie09udbr9I59wr4eKxYKfASZMcfWjOOZzmdMRAhZGJYMYBXSQJsaaE+snEZ  
01ZmOI/qDWgRERFEKniuO8AQPEGcXKYOCnFSUvJMG/JpHLauz/DsVsRnbD02DeGNJ66TsfvU8bEx  
XCBHZtAI0EbCOnOrUrwJl+HcMyLO2rSjBBdR8wHY8qJwkAcaHCGfACJlCImYIBFAMmIFUI6KQsXk  
kXPY7j9p6Z5XPMEYAqGAftzteLbUssKxm6RLCXHXFIJBDGdURHKKezZnHfliSbADqaQ5Z3LkzXYA  
2J2t0rhiquITiMwY/KvIfvU3G3YqlSLuK4a4phEb4nhJBYYB6Dr6VDCyOMAPl/8A6qjOeMqFcnTz  
0/Sau0N/efcUvAf7Xs2oQhSCZBkm1/L+e9KM0gdGFk+vpTDNPZVCyAfOOV73oHO4lwR/UvL6ly3a  
R0S8Fj7d5P70vzWXYryuOtHICT5+VdiIZg2AH8vVYyZlctmcGF8NtSss8+9/OnXCeKgtPne8cwf8  
0uz+XgzI78qUYuCAQUaGHQ/muqLUthvro+iYecCtvPhPkYuI+9MOHZNwQJtqj/2BH5+tYXI8UD6V  
azAAE8iRb6i0U9y2d0LC33F/ceWwj/NaqHUrNMmYuCNtvyDHvRGWxFZVc22Y/ePSfpWZTPBiAAez  
jzAMft7UamaIOsLg7wd95t60QjXN4SsQpvcEWHI29efnVbPrAV5tdWUkXG5nrcyDl9ao+NcAk2gq  
R5f8ewqeYxpWbfbwTbof52oWYkmZ0PBYDcLqlTz8LG6nt9hTUPSQJ6qdxSZZXV0F5EG/rsR15VfgZm  
FENEWvf0M38v8VrNRdmuGIX14cAn5oI8XUQbTee/3p+AFkaokzBSJ67SPpXJmgR000Tcj/8AQPL0  
rlseZUeIHZEdx/yD+Qajx10iUtztch/4nr712ER80BS24E6T3v8AzyqzDDaYkKd+X7/mp4WVvcwG  
i24Ppy89qNhIrgjZRbmN/frS0pp8Dxa3sPvH0rTJh6V21el/LzisvxN4Y8559CBP1ifWlNY64e2l  
CZvFvIWH1kULjZkBX5pWNPNS0ikud4iW/7abCCx8iCV+hFSRY7t5T63j6xWMhlmX3K8jPnBI+1E  
8GcFwRvcGhcJSQ02uSfi7/Ufer8q64ZU73/bnQYRhxFCCjAtoafQidWT/VuOBiWQNqRdZPl9dhW5  
x0lowG5FqxnGMPxI5IA0AMivKkrvz2pXS2S5fxMsnEDhQrBWw3EOkk6fI/mn+WwUOGLjQICkx/5  
d/vVf+jwHA1rpnblvb0oTHypy7SjasIjxKtT/j7UXnRHsmqZTiYhR20IYk2UXItf6U6ymfQopIKu  
LQ3TvS/MYqKMPFw7qTDHpyv0NHYiAi4kGpOKu2I7i8ijjWixYfCWFkFlmGd/AMl5HuK5crgu2pcz  
oe4ExpI5im2DpGoETK6R0HTyobK8OYFw4RcP+kKLnl7VX+oooKml0zXGuE42GusurjVHUGedefp  
5yMRVxABrIW4I32N6licJwwZA226e21DcV4R8VlZ3ssRFiCDivWXNGSpgr2JuM5d8tj60Xuu5BH  
MXrs/gYmMUZMKABJlgCT060zz2K7hEK6gDzIkHrPSgXXEKt8NtDC51f/AJOxp1JAAtC9+GZsMCqkC  
RPOP3pj/AKgiwmltVf60IuNjo6F3aNA4EQw9L0R/1TE5fas8jKkaw45IvHtVbpIHbbtUENdms0EWT  
zMC8CYm/tXiKUPSSITxlKTqxccOGixI630luVVxrIiRCT06kb73oknXpYwCd4G9SwsFQJEK9zYfz  
tXpQtRzs74RxTAHwBHyme5BJ/alGPhAEmCb37+1aBsJpIkwLyR7AQKV5ggMVG/lyU8ZNCyaQozWX  
sYgCvclxFKEPJjY/g0Xi4JJE370JiZcXIqsZrTJ9qeBzls0jQ6HlB/n19Kb4eaiSiVfjsbfQ73rC  
qHVgV5dNjTDLcUKkhxYj2/5p6+FFP6bhGNO20348rmI7mrCZGoX6idj/AD61lsnxRYgGfWDWiTNK  
yl0IkjrZ8xtP3FKx07KXcp4hseXI/gVHDzIJMSwJgjnPl2otswpgOAQefMdj3sb0rz4QEFDPyddj  
Gx7HvWCaAkhVOnUpo/hyBpLDUOX5B5TtcVm+GcaLAIwK729Zjlg+tHYffigBuQB19N6DYUzWYSLE  
Wjl3HKe/7d68dkAgACJt0rI5rjJ0Qranmw9In3Aqa559AOqCGkk7EdPX80GY0bzSqb269D/BWK4n  
nZZglzefaF/A9aJxOIBFbcFpsf52pFlFaZ7ye/8AcPKxoxQGw7h6i5Jkn6yQfua2GRwF0KIG0Soc  
3B9iKz2Wym36H67g/wCaf5TM+gv7T+LUQnmNhlYBF7g9wQCPqKhJqdbHSSARz3/ABbzmiczjgkc

5+xuKKy2CFIJ8/59PrSmsJR9JjtfSRWH/XAZGUglQWMQQJsDfsINbV2M7XJ37f8AFZr9a5ZXw0Zg  
CJK3N78rUnK51ldmK4RxVGlMUKAkajs3cVPL0cUFImP6ojymvWCosQFHKqsDjAUHUhAG0byaZr+0  
5lnSDcLLNgGZ14REOnTv5UZgZAodeHilsI7Kbx2M7edVYXEUInVA72BocZ7/AExloVbDcwYt9u32  
pMjpt4YV/q4JDAKQJ338hvXZbPer/UVN5I/lqF4xoVsPFEI GnxbhOek/WrcDOiYwhnTApelkpRaw  
TXOY2o/KF5CL/mpkljLMD9qghJuaHzWfVBbxHoP35Vld4BF/EXARiX0k2n9qyeGuJrhCugnaSYPQ  
dKs4hlSBHxQSSKAAfEC06kVa3AIJOgelzV4xjFbHSS2UZ3M46ko6iV2mL9/K1K8TOtJ3/wDan2Zw  
20DV8yCFuDPbrS34oN90/lVEl8H6o+ho80Nxo/8AbT/ePtFcArTxcRacu5sSpUwfmV4fEqmmc/F+  
RblmSB/VHTy26V5iYwX5FIHcd+gHaqMtnvAI0rYWak7Gp4mNCkhmPoBXDR6CurBcVsZ9pCx0I95v  
0ofHyzf10BI/uv6Ci3ckQR56mNBuZPKT0H5p7FkkDBov5/gqgkMYHLeimTVteedDHDN6KJOJUydv  
80Ni4cn7CjmfFNgbftVGKk/imi2tgWNgv+m338+9SwcTET5HYdtXv6LJA6UWUvSb3pnI1ZsGTImK  
Imp39KIY2K7yd03IEW9DBiqMXAjbXuUzIwYsXgG16N3oaTkkMcvjhPmVlflY3NCZrMMzEspvvMzT  
EQ8RBWA07/WpZ9MNCMMSZmCsir5Tv+1LlMlcn6B5YoLglT0m/wDIn3pq+aCp4mAFjJ25c/Ss78SR  
KzHI+sfioI5cFZtVOtlIyd0zuIZsuQqMYWRM7zWm4UnhBPODA+v871mMngDVpJjotWu4DjKh0Pty  
O9F/EVivpoMniACDcD7UVjYawHUwZ+vQ+Yt6lQ+XgSu3Lt/g0F8bdDYzz+xFIUCsJmnURESPI71d  
i8WAYKTO31G/a9Ca2UbEW26mLHzH1oN8qrkMx0hrx0gzA9zQMPMHYcWtex69ftST9bZgDLgsHMG  
SVAEWsTPSf5amuUyyq5csNA+RdotH4rE/r/ieHiOgStoLSV5be+59qMvkSbwZfiOYbEuJIgDuBsN  
quxcynw7DSIhgBme/qKGfGZVBVoG9oBjlqvBzCsRjnnJ3v171XrgiUY+YEjTyv5U8yuKcTD0YkEA  
SDsQeVcTnMPQyQCTYuRz6f8AFD5PNkrGTZQSY8961X4GLH2RxiilcVQCj7MpMx+1A5/KMmIWBlGk  
pFpHQ9xzqGJhnFUOrAKfmvyHemWQxgR8LEup+VuhqbaTwZSi3RRls4QdtJw2tN60eWwss4jSJPI2  
P+azuNlVAZCsQT/UZI5GeYiKafpJCUo2oE6PCNUbRM1LlS63YnIqC83kVQeH5SDHagy4kDKBTrHw  
1KkWA+1ZLiud+HiaEZXBZeNtp2JtS8LclXpNK9DJ0Q3dQQL36edVtgcPa979GaPtWexOIM4JItG  
w2pN/q3/ALo7Sa6owl9HUA9PouexHVzDWvEgftXiKXwcSTf5ffY+9dXV56iqM0juDYiFFX+qOm1W  
Y6iCASRM1ldVns6l+ILpjcKBznnVwxFOQoAHl+9dXUBSkN/bseleYiALIHn0HeurqPoAM4J6yaqG  
Xg3/AJ+9dXUG8kZfkCMPxCltqtRzME1ldWYDjMuV4rhkgqLncAXP8g1ldVOPY60F8IRkQhhEQYms  
BclNuIqdSlgRqkqRMdDXV1VebEeHGVzVOMDoQAajZdgN4ovLZR0QlZJjny5VldTLsgj+RPAQMSOf  
KtHkcElZAmNx+a6upZHREdcOxivhNxtVmcyoN5rq6lY5RjNjW0w6g2IvbpFW5nIF4LE8iVA6d66u  
pWAo4rnlCaLyVKkzLLI286+e4GScsZeQu6+VdXU8SO9lWJib5RG0fSl78POog2HKbVxldvYgcUUY  
uEyW002NDKh29+ldXVRaJvYxyGDilWVJKGJ6U6yzE4YBlXfU89a6uqHI9iPDGPDcc4gOG8fEQeAn  
+odJqrM8rDeKoShElhHinpXV1JS7DS8FOU42+tdRlBjBlYwJyneneazHIKoogf0yZPXtbpXldR5F  
lqjOTS6laYaxBW9Vf9Lwead3Ne1lTUMTP//Z - No connection adapters were found for  
'data:image/jpeg;base64,/9j/4AAQSkZJRgABAQAAQABAAD/2wCEAAoHCBYWFRRgWFhYZGBgZ  
HB4CHBwcGh0fHxwhJR4hIR4eIx4hIS4lHiErHxwcJjgmKy8xNTU1HyU7QDs0Py40NTEBDawMEA8Q  
HxISHzQsJSs0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0  
NP/AABEIAKqBKwMBIGACEQEDEQH/xAAbAAACAwEBAQAAAAAAAAAAAEQBQIDBgABB//EADoQAAIB  
AgQEBAUDAwMEAWEAAAECEQAHAwQSMQVBUWEicYGREzKhscEG0fBCUUEUcvEVI2KSgrLCov/EABkBA  
AAMBAQEAAAAAAAAAAAECAwAEbf/EACURAAICAgMBAAIBBQAAAAAAAAABAHEhMQMSQVEYcWET  
UoGRof/aAAwDAQACEQMRAD8Ae8WzCwNTWj+UDx8t+VKvj1rogUdT4j+1QynDi7fEd9bkksty7Ac  
q8ZGDS025DlXkykm7Z0J0EY+eYQDYHL6C38vVTYzSYJfjaedUvjqxC/iT39KLW8tsxFwIoN2PbBE  
zLlJcHpeDM87b7VNQRuk8ib9I+1EFew/WeVUj0qJUSQepHtfnWWQEjlyU1LNjtz/mlVPky4kWM8  
7Hr61bh5jEXYqydVMHfmNrUdnc2mpIHUEEWN7UaM0BY2MVUHSwQeFvDN7Rtt35UK+ECQQANO7A3O  
29rbfWrV4+UYyngm+/v1mKtfGw3IdNtWogGJHMed7+1ZqgC4hRmkkkBgw9pIvluKqzeGGZAosqFF  
+hFT40oGKhRyELzbkGW4+htVPwzlgwGmCJ3YChJVLtfhCTqVheDgF3CxHwBSPL8VTlCiviIyGA1  
uo6Wop8cN/3Q1wbj70PmcyBms4EgoDHeKfZpv+P+jSblHsaPXiNgBkSchlm2Nt5HlSJ/1Bi4jhaQ  
WWRPaie59KZpmAyMiuRrQjEExFv6Z/qjLWxyCYivNlOisPmEHST6WqjgprTlkeK2kDYZkyRM/em  
uSSbtS3DxdWgmFM8r8opzhKzHUBHYRFmx4oozIBS0AEkXrOZ/MrclyGwim/6mxygWfltrkHy5ln  
gGcdWsLuQNX5d6aEV+TJSxIO3AMAYuEZPK1toql8IHhBBNhoU7+c1f8Ap/ShIR5W29mHYiqMvhs2  
exRAA0j26ig45bC4qrLP09ksMI7yV0FtQJ5copzkmbQHw2hWBs3T/NDcS4ew0vhAsWIDiwmRHtS5  
+MIh+EpKRFjc6toHrU5cb5GmwOKbyPs3mXVQWCGByYG3kaTnNEaQy61NoAn3obH4gqQ2OPiKWIMW  
bTNzPwr0CJiQPEjjUkHZSLAnrSrioRoZcJxMJ00yAGAIuPTpUOKOieAmQOegEQKry2Prlii2IMED  
87V4ndQoOoglgQTsfIVJxUXTW/8ALE/abMTMa0IF4043os5tvG6CgZtgxB1AedCYWKIME+HC/wB0  
iP8AmrscOHDq+pCI0jlVF2ukq/YylbopGAWQkAfc0HhA6wCZU+Hym1msvnSBQcWPMkW/NFZ+QBim  
hIJHjC2n02p7lF1Q2mIlJYCdUCZgGqYCbqDa8c+8ldneLMuopeBiBbeicrxFHwx8L89Ip3KXqWFX  
f8CZwjGRM4HQf8ANW5HPuhKnxKeQ2j969zGVUEjDe4kjUCIHSh+F5MrqVjEAKHeem1VfVoDQ6yn



HMVF0hU0j1G/WjjnsJrnBSSBptSfL8NcrqVWPpRDo4t8M2A/p7VF8MLuhkbv8NQYEA2Jjy5n1+lK  
MR1awKCTAI3j6xzphlss7FZaSBfkDymrMPKqol9M9du/WlIo2780lCwKFKkggnsfr0q/MYrESCNY  
LzsPLnvRqsjnwLSFsTBJ9Cft2rs1gKotAAVP+aMk0hmKHJVpZ2iII3EmfzRSIjL4T4ovBBivdBIJ  
KkdjvHI2PQldw/KrqkOyneAZn3BtSJ269JXmgfDaRGwPW0XjaoKniM2IiNr3/wAU/GS06CAGgcxI  
8496tbJKs+ERF5/c7VSMJJGutmYzOCGLWgmXkHtfbyqnLIVSCIA7biefajsxkheBBELSCfIekDyq  
CKFUodxzPOazfglCXOup2MFSO/ODHlJt2oHAyrgTqliLMDY/zpRmdyelixWwuOlXNALXwkJTE0M9  
9ESrA+c0/LEZ6phuQwjoYMYLXtE+1JXzmIjgpEkEXvaaY5HQ4K6tD/2nwnzDXH0r08NXWA5uLwli  
e4b5T70Vi0x1bjRHL4mO+MzhmeTL6RYWibbVHDZkzC4eIhIcEEkzfl/O9PMvmsTCQIihJOlh/Ux8  
opVxAurKXUg7Ger6fy1lJvFCJNYZRL0h+wJ9ga0GVQWidqWuwYpG5v78veaa5VTMaYtSS0UiQFX6  
hQlB/urP4R0kEiVNjPSeXetP+ppXDUgE+OIrPPlcViCMNgp5tb6UYJlZGcX2wHnATDx4w8RWEcx3  
vynatHiz8ItYMN7eKORnpWSy/BcySSMGR/vW/TnWnww4wmXGVlIXwnl0+YWPvTyxkpG0soDfOIQh  
DeJhZQfee9VthYcsCFDsLM4FiOh60LxnF0NoaAdAKuqjxEi0Wv50u4Rj6w4xCToBIJ/qjlSKLeUS  
k7eDs/1lLqhZWIjYzfntHAWlBtACz/VM9qC4UinFd3QMq3VSevSnHwEnVGm0Wv6zzN6MrWDKLSdy  
WcxMJGRVlAmeIdmWfUhbY2tt3HnarcfN4aqUQlr3JG3WDzqzA44j4L4WgO4ICtHLmSalO7YKBzm1  
cqjQNIJA0gAt6+VE6zAPw9KiNJlEmxtMV6ugYRDAGTIt8vkaGTPsgRdKECRPPeRPI0kk9oVprKCM  
liuzr4Anpf309aX4mjBdSJGg25G1Jm4iJKEZQW97/tRuW4iCpVlEE47Dt2NBckovKGSx2aybl2A  
UgWIIbVInYede80loxwytyNiset63WDxKEZMJQmlZlnSWPeDWLbiau5Lu7OTdoAqim5LQ3oJnMwA  
09DE9f4K7CXQRFSZ738jROYyKOPGGx1TIVrEnnBNB4+X0gLctZTHlIHnePSnTTRmeZbHdiGLMDaj  
Udx2q7OYbl2Idt/7zUslgsphkIuACR9KvzTy7RAv0rN3oyZulxHVG0nxR02vYR5UImYYrLGSNyQP  
XtalEr4gtjBBPQDL++1B/C0nSviIkwe87CIiubs9HSnk5XYmLGatwXAg3Emx9KrXDNrRtv3qKrrg  
GCJ7isngbwLfMzMAXsd/KatyJAYMeR85HX60pORDHcHtttli/PrRuVymnxXkjaTHsfvR9BVs0K42  
rpv528uVe5vNBEZmBIUTa9Z3MiFJkqbXULW9IvVeHitiAqcUsOhJ36d/8VTu6ElEsHEEXAWVAhG9  
r32He1U4uPqElbXuLz7XHtVPw3R+QEwOm3UfmmGF13ZYAt350mXs0dZ04q6v1GIgEC/pt67e5rIY  
UOqO6wwEAjmBIFvStLmcNWBQggqPaOQPtWawsDWvwbySAdhc8/TbvVVMiko4Z2KmE6MWlmtAn72  
BkbVRw3iLoQrr8VB6sojcvZgcPLnSu/OT7lHHwGwnhhz3mxpIt6WSEXK8Gi4YqOgbCXS4U2BjUki  
I8q7OZHWRgGGaLm4kfbpSrPZDQ+rCbQ4GoMDY9iJvTbh+aZ0Bez7MBtPP339adfSkrWRZgIyEBhB  
WR6Hp9aZcPe9z97VfiaEUsbFYnSkhmNgPP8AnlodW2Pxysa5jiSoLKCQd9z6dN6QZria06lgbm88  
o7b38q0WQ/Th0+PeOd496pzP6ZOSWXT2ERbzq0Ulos38BOHZoN8jaT/aDv8AiYvWgyWYaOv0n05G  
hst+ldJVxZhazHb7Uxfhj4aaJ4wLxzHrWaQUxXx7hKYiFlQsw+QC2kze3TttWOPCHwi5chBEXPqb  
VvXzUAPcXAkex9I71jP1flFDq+tn1hnAkQCY5ch/mppNPGiU4p00J2zUTA3PvTnhucUnRF+U7Gs  
3gDYsbg1OSbyd5FGUVJUTu0X53GC08rJDHY3qvK57FxW04aBwKRBAPl0PMkkkkMxJmW/PWi+GIDq  
uAwg9Pan6pRyTGeNmbAMdDXmIM/7l5eleZdGY+FLYcmm3r0qnMYLoEdklLJLTzjy896Ez2XcenFC  
CJ8LEHreTSKKaoZPBoP+n4oJZAKMoKktewg+lTR2KMMQXMD/AGmbGRyilmWOIyIysuqDMjl18qLB  
bdVwyypELGwPY8xvUnBgx4FPhoZh3Ui1tJB5RB3oDPpJGqNa2B06S3+4fml/F8fSWAEAX7jYiKM4  
dx01AuMq4qnaFMA7HnTw4nFXyVfpg7EzBMFhHl0r3L8QfWCN43N7jb6VdxjLqro+G4fDcSs/Mu0q  
fKgWPjILAwZA+tu62sjPiFluLYr4mgsCJ6CPTvKObhQZ59aVZXEL7bwTPpRGfmcUADblpXBLSQMf  
DcPmQXII1AN8sXFzB+29VY2OxJM6B0Av0N4k3586Ey4YPKQSZGWJiNxYc79bv2IXBJdlm+51n0Ag  
A22rksf06cvSS4TESSTBBBJNWrl2VSbiriF89xQeJiAz/wB15ttCgcbhBofGwhZtbA9S7c/M/SqL  
KodNhrZVgxa5n+kC32tVyGLkFSJtPnVGRd4+cm8Ac+x3qzNhwssAwF5te9xHlpX2s1tFuNhKfFLS  
t5Bb7fzevfitpOkzB5ib+1KmYBmtaJXl5HeNqc4WKCoYCQYJI5c4PpRdmuy3CxnZTqAsREH8elMs  
LOYyQANJEgi03vs09KczjhAMQoWG8C1iofe83qODjo5+INSG40svtYWN+/KnSdWa3dE84SxkXtvY  
7bE85tWfUupMxZgZtO88yBT/AASBYc3sepPLeZ2rMcQLBrSRv2/lqMG7M3R7mcRx41lIbXGmoZvE  
JVA4brqteeopNh54lyGLGbagbD6bVZhZxT4GktyP71VQdEUmOERAmqSvc/4JNEcELKTJDI1tSmQD  
yPUdD6UixcrNy7SbBZNqb8IxlgGNIi0iCetDqvASwmaDEgG/8FK04kmWdsw6lgDoUDrJ+wFC5ziL  
C3K3saR/qjNyMNBSWL/QAfSnjG3QeN1Z9N4F+rSHMtoGpX3KsOXUHbnWhfMp08xXyHgKf6ZHxZBc  
rAg7C0jzr0/rV5skm/n53MbVRRXhW/p9D/U/6o/0qAjCL15C7gAxzP4rC4H6/wAyGDsUfDPzppFh  
MRtPl03LcaXHw2XF06WvDbClWd/TuEUJw2YzexlAfn3rXH01Pw3W0uG2CGwzOG6TE7Wnff8A4rPt  
iu2WfQssrkBjpeCec7i5pP8ApjN4uGwWH+UjSO3Qg+VP+FYJGUOHYU6sNue4Mc46VCSpBlmJlWwt  
IPxFGqCdSxHrFqVtmIcEDUF01xNHvkcRdWkjECzMBj/cpuKFR0ctrYqdNiBz6Wp41+yFkczj7wz  
sik2nZek9aGRG1kMYyJwHYNcCZ2iKu04QU6pZt1I5HoaKwcwjQvXhIB4VEST07U9lpGQzKMqqcR2  
YKSSlp/+LGxrx2V0dMRwyKRpgQYMW0jmOYonLZk4mAqMQmhvEWWxWCCL771SMXQACEcSIKGSRYMG  
oX/s0q8E3GHbCfDRHlQgutiZJn6Vuf0vj4eLhFGE1BDahfreszmcdy+nQ0A2aFI/+tT4fiBS5Ryj  
tAdSIBg3IO01TtjJvAjMsQ4hUdJKlWF7G0H/AGkUt4vi5cnSoKlQDAjwHmJ7/inGO0qTIBiAw6je



R5GsviDFkh1BDSJUD3oQyGndjTgj4DEI9lby3BDde3SrjwB3xGfCAZBYEkAmBH4rP5XCKOJJBXYc  
iKtGaxQxc6lF4JmIvsNqPRqTae0ZJphqZLFwzodIsxFwbxbamuV4UdC6nvF9qG/T+XLicw8swN  
bH0orDyWDiAYnxGXVeDNqWTUdjJIdlGE6XCAC5IEAHvO+53oNsUL8g1k7vP8+1NshwzWGBBcksBA  
mOp7VHM5QDZCItLAjnymuOM1thT9YlWszEACD1ue/wAxmiv9HJGtje4BP860bhKASBvfl+a9TDYw  
QbjsPvTslWh2kiOfkkWfkie+1+v1TLHRVS5WIsW2360M4kXE9e9B4yQAumwkDt+1NGVoasA2aZFY  
qjXHMxHnMH+GistnfCFZDPVVMH81Ti6lHzKoI6XMRAvQwLgAkTewMz5UdoXWRoMbCCOhzQW3DjST  
e2/kKHGC2HqKRpcQf8e9ChmcHXpK8iYgdt95+9FpglgIteevf2NB2jbKy7qoAYet579TQOawFdGx  
BC3JAmbDYc4/emGJgmdOW5dvPfakvEXdfkZgNovY9ibiek00MsDTrJ6vDi6wmEwXlqt5edA4HAHL  
+GFjdieY351cm87FA2uVAg3gHlyfej8hw5MMS2IWPMcgDva96sm16SuhZjopEkkONTNy0dutMeFZ  
FlSwnUSB7wPsSapTgZxmXEDkLJg7W3sOfSm+YcJh6EnwjfmY6mlcvjyXjB03L0zOfY69LAhtiCi8  
qXvhyw1brJA6loczxBXdf68z1j1XmVw10F3AMXuBysB9ftTdmtoiuOnaZmsTPh/CwiNiPzQb5Yk7  
j70xz+GriV+bew3pUUCda6IvGAsa5HFCL4oB5RMnrvtRQ4oxICQog2H3796z7hj8u9EZXBxAQQdv  
2gz6HahKKCmzYcNzBILncDpz2tTpVP8Ap2gwQAY9QRf0NJOEYDBFUiO35PfetXk8raDbz+ornkiy  
VqjEH17YjBtYTE1SSFA1MYhta4obiGXV1Z9IDneLISNyB1NUcdyBXMvhhAgDQOQI5NO0Gd6hm+Ho  
sKc8ryZKqrkf+0QarlimQSUaYG+EQhbWEEKDMTCzzjn50flfhKod11E/KoMgdyA4n6YV8PXhM2I  
AQNY27gg351nHyj4eI2Fq0nVF9j0rKSmmkxW7GSZxcV1DMSS7AXXoADY1PEYjawYtspuNvzQGCz  
FjCDUSNthG9PcAsBDG29zIk7n2pZLroZJs0PDuHkLqZ1A0DkDJ70r4rw/AKNiI+1lIDrO02mKIye  
WUAfCdizX0kypINx2tep8S4OmLmCzmG03C2DR1IpFXbI0dgOHlyjIreIOsTy8QtHfa9LXcaSuspE  
XYT2I7itWmUw8bDR0lGQAJq5Rs09L8zwnDkM5Aknw6oDTuK3amNKSukJc3wWUXFV2ZtiFFvMU3yP  
6czCDRiNOu5EyQIsI5GrcLMukqmCAq7AST7xApjg5osoZ10E2IJmPas5S1YrTRB8IooRMIEBdmzy  
/egcLh5AA0taruKZrFRVbDQmpnxEmLb7VR/1odvesoyMoyNdIcPwAMjm0+AgMscwCLnnXuWzLsSu  
HmUxgN8LMLpcdpPOPkq2UgjqDq84H88qXZricmMxhDEtbX8uIt/7139ahwtOJWLWp0ySNOvBfBbm  
VYaT5XMelQXI4JgKHHk3+KrymMhwLOHiYjLyD7jtEVNX5HfnEmPXAq9I/CtJnr8FR18OI6nl035x  
E0JmeD4iRHjVbiN/YmTRKSLqRlvJ+1qv+04iIA77nyAmIFF8cWhWjOFiJlGBvZhH1vahizG5UqOx  
+vWtVmMP4nhdQw2Oobc/MW70nzf6eGoNhqYPUnblPhxUBrAuOX1Jpgsx3tYjp05qWSyb4S1baTfx  
MTznmbD/ADRGDwXMggRI5nwgDn1n6UY3AMdrkr6kCfvPrQ6yqgIAx8e5B0mLHuI5XvSrj2XD5din  
zddoI2pjnuH46C+GYHMR+Deg8zhttpYysmxJn+cq0YuLsDuhdwrMOERVdAWuWZtWntI2HSnGiYOI  
QWakgGVInc9eVJMDLrhrKkqzEhrdzYdKapilkAAaltjJsR+KabzZLCyMMY5kAflYAfvIUPjZHFxCU  
AVF2LMPsBcmg8DHZX0FjOk6BpIBYCdIabyJ9qIfN6kLM8qPmg7djf6FPwMuSTsbm7xHIDAxUnGR4  
Ia0yDO0XM/SmWfhstilBFz9v3FE5fJo8OiIqWYaJJ9Db1NeZKNDglidQsYiR5AU7t1/A0ZdnVGU4  
bwtlTViWZiFCnkOc96PweGoQQBC8yNv5FMjhaidWwuPS5+5ojL4ABHQ/f9u1Vcm8lVBIW4XA1g22  
EfX9quy/CIMASbW9tzH2p+EjflP88uXej+E5eXBI1ATJ+vtSuTGpIVYeEYOFIECIH5pwxKhWPPke  
4t/zR3EcBCw/uUift9KJxcv/ANsCx0g26g3EepoUEzfGsor4HxNQrGqR0s6o2A0kibedYHiuXy6C  
FbEZ43KAD63r6flVJTFXcalj1A5cqUBFxcPRj4LLIIGpZYR/UCNvWl7dXbOf1TTs+ecM+O5GCUi6  
IAWADEAHfydTtF8AS+MmO64OZOpdcq5s2oAwRnvpIkelADK5ZMAT8JS7RPjHiYwsvKlfd0xMzjuh  
TDwCpBgovZgTtMwx503dSi3RHNmgfCyxchmJbWLKIHuN6GznDFZdYECfl6dqfYGS+Agd2DIgJMLE  
AOnOkuPxrDIbELAJcxPM7L5xU4JyDjZSE44i2A4OkkRpgLIHn0q8Zc4rk6UEEKZ3vA6UFitImDz  
U6XMgDmK7M8UCK6QVIEgbGSKbN0thcQHWYz6YCBZJtYbnlpZmOKY0K6Jht4j4d2PQg8rdqWZBsJy  
VxXKuWfpu0lDZnF+G4ZRqhiBuZAAAnzpowskyjeWaLG4piPhNoAR0uw3gDcVNcyThLjtAk6HXCg0  
y0ludQ4M2HilcTCXQfEMRCLmQRI/uEnah3TGOAcPDwpRjN5BXy5R3o1FKtBVUaHgwDI5SSIKoQIP  
Solqt4/DAWJCLf8AnSgeF42Pggh7sLAEgeH03501/wBcWvp37/4qTnOLpZEc5RwaBmaAp3Yct7cp  
7ikWdcH+krExtAFvpTvMYbaDqbUwMgx0pVm8ZgPlkX353/xXPDZdOti3B4iUYpqhJsbyDub9Ovpc  
U6wc7YarT3t1nuYuKQ5hG3A7dx3+1Aa3SLllvY/iuuMvAqTWzf4OaDW5XJJ63/yavTNSJiDcgeQ5  
/esXwviJZWlvFJn16e5rQZbMSFIG9rf7ace7HmRxYQAm8+Ik99/UxR+C8je8C/0Ples/lSG08t/  
Sbfb70blczYGZlW5dNv53rGGaYoJLTikgwZII8JFu4NDZl2Q6gCV5sCbdPDMXrxGAU3Hjk9pN/vN  
UYgYqVVocNKzz3kdwb270xgtM4n9bW5HxQZvvUcxg4Di4kDb94/xWexeIPhtp06iTZSYJkGNLGxP  
h08fkmpmiQdLMjLurgkeESCH0zEHe3TcUKsFE0IcADj/ALZ0+dh9FP3pBj8FzSMNGkiRrggyOYA3  
mOdq1OC7wNUkEXKywB/3apBonExQFJ3i5kSR3iJ+9DqvRHCLM2yNqGsDQviIYXtf0g3pFg5XWWXV  
GoEB1Xeb+LtWlGMmIDpVXBEEjSfTrUHyiYaeFFQDuAPcnekUKMo1hGay/B8REGH8QsJmFW45ke/o  
PWvcdSAEW+0xf3PpTHNviHwoFRd2LMR5yQpEdjVGDlX+Z0Jk2YENaf8Axmev7GqUUjFI9w8AaQ0S  
Qfoeo61cmUAYmDztvt5SKIXAhd5Aht1m3IGp4OnfYHeeX8NAYrwcFmWVE29R6c6c8Mw10hogx7H  
9v50qPDsMB2InrH3HnNeYWMUC28MzHY9B6+9KzEcW2Ix3v8Az6DenaqGTtSDNXckHew7jce23pTD  
hWOChEyR9qwGePlQmIzDZ0W3KVgD6VlflBxDM40JKkhFbXMQCrCZAIJBgitfmcY68McmDfj+elZT

jHEUTM4mFieLCcLq38BAABn2NKJPRHPcRR0DFVeD4NBIaD2BteoZfDZyAUsOTG4n67UKuSK/I6sm  
4YLHoTsDSzW7FyHYGSCSTB8jSVK6Wjnbzk+gY2VjCZFbVKsAWPUGxr4yODYiFg0gobqdie09udbr  
9I59wr4eKxYKfASZMcwfWjOOZzmdMRAhZGJYMYBXSQJsaaE+snEZ01ZmOI/qDwGReRFEKniu08AQ  
PegcXKYOCnFSUvJMG/JpHLauz/DsVsRnbD02DeGNJ66TsfvU8bExXCBHZtAI0EbCOnOrUrwJl+Hc  
MyLO2rSjBBdR8wHY8qJwkAcaHCGfACJlCImYIBFAMmIFUI6KQsXkkXPY7jp96Z5XPMEYAqGAftzt  
eLbUssKxm6RLCXHXFIJBDGdURHKKeZzNhFliSbADqaQ5Z3LKzXYA2J2t0rhiqUITiMwY/KvIfvU3  
G3Yq1SLuK4a4phEb4nhJBYBY6Dr6VDCyOMAP1/8A6qjOeMqFcnTz0/Sau0N/efcUVaF7Xs2oQhSC  
ZBkm1/L+e9KM0gDGFk+vpTDNPZVCyAfOOV73oHO4lwR/UvL61y3aR0S8Fj7d5P70vzWXYryuOtHI  
CT5+VdiIZg2AH8vVYyZlctmcGF8NtSss8+9/OnXCeKgtPne8cwf80uz+XgzI78qUYuCAQUaGHQ/m  
uqLUthvro+iYecCtvPhPkYuI+9MOHZnwQJtqj/2BH5+tyXI8UD6VazAAE8iRb6i0U9y2d0LC33F/  
ceWwj/NaqHURnMmYuCNtvyDHvRGWxFZVc22Y/ePSfpWZTPBiAAeZjzAMft7UamaI0sLg7wd95t60  
QjXN4SsQpvcEWHI29efnVbPrAV5tdWUkXG5nrcyDl9ao+NcAk2gqR5f8ewqeYxpWbfbWtbof52oWY  
kmZ0PBYDcLq1Tz8LG6nt9hTUPsQJ6qdxSZXV0F5EG/rsR15VfgZmFENEWvf0M38v8VrNRdmuGIX1  
4cAn5oI8XUQbTee/3p+AFkaokzBSJ67SPpXJmgR000Tcj/8AQPL0r1seZUeIHxEdx/yD+Qajx10i  
Utztch/4nr712ER80BS24E6T3v8AzyqzDDaYkkd+X7/mp4WVvcwGi24Ppy89qNhIrgjZRbmN/frS  
0pp8Dxa3sPvH0rTJh6V21el/LzisvxN4Y8559CBP1ifWlNY64e2lCZvFvIWH1kULjZkBXb5wPWNP  
s0ikud4iW/7abCCx8iCV+hFSRy7t5T63j6xWMhlmX3K8jPnBI+1E8GcFwRvcGhcJSQ02uSfI7/Uf  
er8q64ZU73/bnQYRhxFCCjAtoafQiDWT/VuOBiWQNgRdZPl9dhW5x01owG5FqxnGMPxI5IA0AMiv  
Kkrvz2pXS2S5fxMsnEDhQrBWw3EOkk6fI/mn+WwUOGLjQICKbx/5d/vVf+jwHA1rpnblvb0oTHyp  
y7SjasIjxKtT/j7UXnRHsmqZTiYhR20IYk2UXItf6U6ymfQopIKuLQ3TvS/MYqKMPFw7qTDHpyv0  
NHYiAi4kGpOku2I7i8ijjWixYfCWfKfLMgD/AM15HuK5crgu2pczoe4ExpI5im2DpGoETK6R0HTy  
obK8OYFw4RcP+kKLnq17VX+oooKml0zXGuE42GusurjVHUGedefp5yMRVxABrIW4I32N61icJwwZ  
A226e21DcV4R8VLZ3ssRFiCDivWXNGSpgr2JuM5d8tj60Xuu5BHMxrs/gYmMUZMKABJlgCT060z  
z2K7hEK6gDzIkHrPSgXXEKt8NtDC51f/AJOxp1JAAtC9+GZsMCqkCRPOP3pj/AKgiwmltvtf60IuNj  
o6F3a4EQw9L0R/1TE5fas8jKkaw45IvHtVbPihbttUENDms0EWTzMC8CYm/tXiKUpSSTIx1KTqx  
ccOGixI630luVVxrIiRCT06kb73oknXpYwCd4G9SwsFQJek9zYfztXpQtRzs74RxTAHwBHyme5BJ  
/alGPhAEmCb37+1aBsJpIkWLyR7AQKV5ggMVG/lyU8ZNCyaQozWXsYgCvclxFkEPJjY/g0Xi4JJE  
370JiZcXIqsZrTJ9qeBzls0jQ6HlB/n19Kb4eaisIvFjsbfQ73rCqHVgV5dnjTDLcUKkhxYj2/5p  
6+FFP6bhGNo20348rmI7mrCZGoX6idj/AD61lsnxRYgGfWDWiTNKy10IkjrZ8xtP3FKx07KXcp4h  
seXI/gVHDZIJMSwJgjnPl2otswpgOAQefmdj3sb0rz4QEFDPyddjGx7HvWCaAkhVOnUpo/hyBpLD  
UOX5B5TtcVm+GcaLAIwk729Zjlg+thYffFiGbUQB19N6DYUzWYSLEWj13HKe/7d68dkAgACJt0rI5  
rjJ0Qranmw9In3Aqa559AOqCGkk7EdPX80GY0bZsQb269D/BWK4nnZZglzeFaF/A9aJxOIBFbcFp  
sf52pFlFaZ7ye/8AcPKxoxQGw7h6i5Jkn6yQfua2GRwF0KIG0S0c3B9iKz2WyM36H67g/wCaf5TM  
+gv7T+LUQnmNhlyBF7g9wQCPqKhhJqdbHSSARz3/ABbzmiczjgkc5+xuKKy2CFIj8/59PrSmsJR9  
JjtfSRWH/XAZGUglQWMQQJsDfsINbV2M7XJ37f8AFZr9a5ZXw0ZgCJk3N78rUnK51ldmK4RxVGLm  
UkAkajs3cVPLOcUFImP6ojymvWCosQFHKqsDjAUHUhAG0byaZr+05lnSDcLLNGZ14REOnTv5UZg  
ZAodeHilsI7Kbx2M7edVYXEUInVA72BocZ7/AEx1oVbDcwYt9u32pMjpt4YV/q4JDAKQJ338hvXZ  
bPER/UVN5I/lqF4xoVsPFELGnxbhOek/WrcDOiywhnTApelkpRawTXOY2o/KF5CL/mpkljLMD9qg  
hJuaHzWfVBbxHoP35Vld4BF/EXARiX0k2n9qyeGuJrhcUgnaSYPQdKs4hlshXQSSKAAFECo6kVa  
3AIJOqelzV4xjFbHSS2UZ3M46ko6iV2mL9/KlK8T0tJ3/wDan2Zw20DV8yCFuDPbrS34oN90/lVE  
l8H6o+ho80Nxo/8AbT/ePtFcARtXcRacu5sSpUwFMV4fEqmmc/F+RblmSB/VHTy26V5iYwX5FIHc  
d+gHaqMtnvAI0rYWAK7Gp4mNCkhmPoBXdr6CurBcVsZ9pCx0I95v0ofHyzf10BI/uv6Ci3ckQR56  
mNBuZPKT0H5p7FkkDBOv5/gqgkMYHLeimTVteedDHDN6KJOJUydv80Ni4cn7CjmfFNgbftVGKk/i  
mi2tgWNgv+m338+9SwcTET5HYdtXV6LJA6UWuVsb3pni1ZsGTImKImp39KIY2K7yd03IEW9DBiqM  
XajvXuUzIwySxgG16N3oaTkkMcvjhPmVlflY3NCZrMMZEspvvMzTEQ8RBWA07/WpZ9MNCMSZmCs  
iR5Tv+1LlMlcn6B5YoLglT0m/wDIn3pq+aCp4mAFjj25c/Ss78SRKzHI+sfioI5cFZtV0tliYd0z  
uIZsuQqMYWRM7zWm4UnhBPODA+v871lmMngDVpJjoTWu4DjKh0Pty09F/EVivpoMniACDcd7UVjYa  
wHUWz+vQ+Yt61Q+XgSu3Lt/g0F8bdDYzz+xFIUCsJmnURESPI7ldi8WAYKTO31G/a9Ca2UbeW26m  
LHzH1oN8qrkMx0hrx0gzA9zQMPMHYiCwtex69ftST9bzgDLgSHMGsVAEWsTPSf5amuYyyq5csNA+  
RdotH4rE/r/ieHiOgSTolSV5be+59qMVkSbwZfioYbEuJigDuBsNquxcynw7DSIhgbme/qKGfGZV  
BVoG9oBj1qvBzCsRJnnJ3v171XrgiUY+YEjTyv5U8yuKcTD0YkEASDsQeVcTnMPQyQCTYurZ6f8A  
FD5PNKrGTZQSY8961X4GLH2RxiilcVQcJ7MpMx+1A5/KMmIWBlGkpFpHQ9xzqGJhnFUOrAKfmvyH  
emWQxgR8LEup+VuhqbaTwZSi3RRls4QdTjw2tN60eWwss4jSJPI2P+azuNlVAZCsQT/UZI5GeYiK

afpjcUo2oE6PCNUbRM1LLS63YnIqC83kVQeH5SDHagy4kDkBTTrHw1KkWA+1ZLiud+HiaEZXBZeN  
tp2JtS8LclXpNK9DJ0Q3dQQL36edVtgcPa979GaPtWexOIM4JItGw2pN/q3/ALo7Sa6owl9HUa9P  
ouexHVzDWvEgftXiKXwcSTf5ffY+9dXV56iqM0juDYiFFX+qOm1WY6iCASRM1ldVns6l+ILpjCkB  
znnVwxF0QoAhl+9dXUBSKn/bseleYiALIHn0HeurqPoAM4J6yaqGXg3/AJ+9dXUG8kzfkcMPxClt  
qtRzME1ldWYDjMUv4rhkgqLncAXP8g1ldVOPY60F8IRkQhheQYmSBc1NuIqdSlgRqkqRMdDXV1Ve  
bEeHgVZvOMDOQAajZdgN4ovLZR0QlZJJny5VldTLsgj+RPAQMSOfKtHkcElZAmNx+a6upZHREdcO  
xivhNxtVmcyoN5rq6lY5RjNjW0w6g2IvbpFW5nIF4LE8iVA6d66upWAO4rnlCaLyVKkzLLI286+e  
4GScsZeQu6+vdXU8SO9lWJib5RG0fSl78POog2HKBvXldVYgcUUyUyEWO02NDKh29+ldXVRaJvYx  
yGDilWVJKGJ6U6yzE4YBlXFu89a6uqHI9iPDGPDcc4gOG8fEQeAn+odJqrM8RdEkoShElhHinpXV  
1JS7DS8FOU42+tdRlBjBlywjyneneazHIKooGf0yZPXtbpXldR5F1qjOTS6laYaxBW9Vf9Lwead3  
Ne11TUmTP//Z'

ERROR - No se pudo almacenar data:image/jpeg;base64,/9j/4AAQSkZJRgABAQAAQAB  
AAD/2wCEAAoHCBYWFRGWFhYZGBgZHB4cHBwcGh0fHxwhJR4hIR4eIx4hIS4lHiErHxwcJjgmKy8x  
NTU1HyU7QDs0Py40NTEBDawMEA8QHxISHzQsJSS0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0  
NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0NP/AABEIAKgBKwMBIgaCEQEDEQH/xAABAAACAwEBAQAAAAA  
AAAAAAEBQIDBgABB//EADoQAAIBAgQEBAUDAwMEAWEAAACEQEAhAwQSMQVBUWEicYGREzKhscEG  
0fBCUuEUcvEVI2KSgrLCov/EABkBAAMBAQEAAAAAAAAAAAAAAAAECAwAEBf/EACURAAICAgMBAABIB  
BQAAAAAAAAABAAhEhMQMSQVEycWETUoGRof/aAAwDAQACEQMRAD8Ae8WzCwNtWJ+UDxE8t+VKvjlr  
ogUdT4j+1QynDi7fEd9bkksTy7Acq8ZGDs025DLXkykm7Z0J0EY+eYQDyHL6C38vVTYzSYJfjaed  
UvjqxC/iT39KLw8tsxFwIoN2PbBEZLlJChpeDM87b7VNQRuk8ib9I+1EFew/WeVUjOqJUsQepHtf  
bnWWQEjlyU1LNjtz/mlVPky4kWM87Hr6lhb5jEXYqydVMHfmNrUdnc2mpIHUEEWN7UaM0BY2MVUH  
SwQeFvDN7Rtt35UK+ECQQANO7A3O29rbfWrv4+UYyngm+/v1mKtfGw3IdNtWogGJHMEd7+1ZqgC4  
hRmkkkBgw9pIvluKqzeGGzAosqFF+hFT40oGKhRyELzbkGW4+htVPwzlgwGmCJ3YChJVLtfhCTq  
VheDgF3C3xHWbSPL8VTlCiviIyGAluo6Wop8cN/3Q1wbj70PmcyBmS4EgoDHekfZpv+P+jSblHsaP  
XiNgBkSchlm2Nt5HlSJ/1Bi4jhaQWWRpAie59KZpmAyMiuRrQjEEFv6Z/qjLWXYwCYivN1OisPm  
EHST6WqjgprTlEk2kDYZkyRM/emuSSBtS3DxwDGmFFM8r8opzhKzHUBHYRFmx4oozIBS0AEKXrO  
Z/MrclyGwim/6mxygWFltRkHy5lNgGcDwsLuQNX5d6aEV+TJSxI03AMAYuEZPK1toql8IHHbBNho  
U7+c1f8Ap/ShIR5W29mHYiqMvhs2exRAA0j26ig45bC4qrLP09ksMI7yV0FtQJ5copzkmbQhW2hW  
Bs3T/NDcS4ew0vhAsWIDIwmRHtS5+MIh+EpKRFjc6toHrU5cb5GmwOKbyPs3mXVQWCGByYG3kaTN  
nEaQy6lNoAn3obH4gqQ2OPiKWIMWbTNzPWR0CJiQPEjjUkHZSLAnrSrioRoZcJxMJ0OyAGAIuPTp  
UOKOiEamQOegEQKry2Pr1Ii2IMED87V4nDQoOoglqQTsfIVJxUXTW/8ALE/aBMTMa0IF4043os5t  
vg6CgZtgxB1AedCYWKIME+HC/wB0iP8AmrscOHDq+pCI0j1VF2ukq/YylbopGAwQkAfc0HhA6wCZ  
U+Hym1MsvnSbQCwPMkW/NFZ+QBimhIJHjC2n02p71F1Q2mILJYCdUCZgGqyCbqDa8c+81dneLMuo  
pEBiBbeicrxFHWX8L89Ip3KXqWFXf8CZwjGRmm4HQf8ANW5HPuhKnXKeQ2j969zGVUEjDe4kjUCI  
HSh+F5MrqVjEAKHeem1VfVoDQ6ynHMFV0hU0j1G/WjjnsJrnBSSBptSfL8NcrqVWPpRDO4t8M2A/  
p7VF8MLuhkbV8NQYEA2Jjy5n1+lKMR1awKCTAI3j6xzphlss7FZaSBfkDymrMPKqol9M9du/W1Io  
278OlCwKFKkggnsfr0q/MYrESCnyLzsPLnvRqsjnw1SFstBJ9CfT2rs1gKotAAvP+aMk0hmKHJVp  
Z2iII3EmfzRSIjL4T4ovBBivdBIJKkdjvHI2PQldw/KrQkOyneAZn3BtSJ269JXmgfDaRGwPW0Xj  
aoKniM2IiNr3/wAU/GS06CAGgcxI8496tbJKs+ERF5/c7VSMJJGutMYzOCGLWGmxkHtfbyqnLIVS  
CIA7biefajsxkheBBELSCfIekDyqCKFUodxzPOazfg1CXOup2MFSO/ODHLJt2oHAyrgTqliLMDY/  
zpRmdyelixWwuOlXNALXwkJTE0M99ESrA+c0/LEZ6phuQwjoYMYLXtE+1JXzmIjgpEkEXvaaY5HQ  
4K6tD/2nwnzDXH0r08NXWA5uLwlie4b5T70Vi0x1bjRHL4mO+MzhmeTL6RYWibbVHDZkzC4eIhIc  
EEkzfl/09PMvmsTCQIihJ0lh/Ux8opVxaurKXUg7GeR6fY11JvFCJNYZR10h+wJ9ga0GVQWidqWu  
wYpG5v78veaa5VTMaYtSS0UiqFX6hQlB/urP4R0KEiVNjPSeXetP+ppXDUgE+OIrPPlcViCMNgp5  
tb6UYJ1ZGcX2wHnATDx4w8RWEcx3vynatHiz8ItYMN7eKORnpWSy/BcySSMGR/vW/TnWnw4wmXG  
V1IXwnl0+YWPvTyxkpG0soDfOIqHDeJhZQfee9VthYcsCFDsLM4FiOh60LxnF0NoaAdAKuqxjEi0  
Wv50u4Rj6w4xCToBIJ/qjLSKLeUSk7eDs/1lLqhZWIjYzfnTHAw1BtACz/VM9qC4UinFd3QMq3VS  
evSnHwEnVgm0Wv6zzN6MrWDKLsDyWcxMJGRV1Ame1dmWfUhbY2tt3HnarcfN4aqUQlr3JG3WDzqz  
A44j4L4WgO4ICtHLmSalO7YKBzmlcqjQNIJA0gAt6+VE6zAPw9KiNj1EmxtMV6ugYRDAGTIt8vka  
GTPsgRdKECRPPeRPI0kk9oVprKCM1iuzr4Anpf3O9ax4mjBdSJGg25G1Jm4ijKEZQW97/tRuW4ic  
pV1EEd47Dt2NBckovKGsx2aybl2AUgWIIbVInYede80loxwytyNiset63WDxKEZMJQmlZ1nSWPeD  
WLbiau5Lu7OTdoAqim5LQ3oJnMwa09DE9f4K7CXQrFSZ738jROYyKOPGGx1TIVrEnnBNB4+X0gLc  
tZTHlIHnePSnTTRmeZbHdiGLmDAjUdx2q7OYbl2Idt/7zUslgsphkIuACR9KvzTy7RAv0rN3oyZu

1xHVG0nxR02vYR5UImYYrLGSNyQPXta1Er4gtjBBPQdL++1B/C0nSviIkwe87CIiubs9HSnk5XYm  
LGatwxAg3Emx9KrXDNrRtv3qKrrgGCJ7isngbwLfmZMAXsd/KatyJAYMeR85HX60pORDHcHtttli  
/PrRuVymnxXkjaTHsfvR9BV50K42rpv528uVe5vNBEZmBIUTa9Z3MiFJkqbXULW9IvVeHitiaQcU  
sOhJ36d/8VTu6ElEsHEEXAWVAhG9r32He1U4uPqElbXuLz7XHTVPw3R+QEwOm3UfmmGF13ZYAt35  
0mXs0dZ04q6v1GIgEC/pt67e5rIYUOq06wwEAjmbIFvStLmcNWBQggqPaOQPtWawsDWvwbySAdh  
c8/TbvVvmIko4Z2KmE6MWlmTan72BkbVRw3iLoQrr8VB6sojcVZgcPLnSu/OT71HHwGwnhhz3mxp  
It6WSEXK8Gi4YqOgbCxS4U2BjUkiI8q7OZHWRgGGaLm4kfbpSrPZDQ+rCbQ4GoMDY9iJvTbh+aZ0  
Bez7MBtPP339adfSkrWRZgIyEBhBWR6Hp9aZcPe9z97VfiAEUsbFYnSkhmNgPP8AnlodW2Pxysa5  
jiSoLKQCd9z6dN6QZria06lgbm88o7b38q0WQ/Th0+PeOd496pzP6ZOSWXT2ERbzbq0Ulos38BOHZ  
oN8jaT/aDv8AiYvWgyWYAov0n05Ghst+ldJVxZhazHb7Uxfhj4aaJ4wLxzHrWaQUxXx7hKYiFlQs  
w+QC2kze3TttWOPCHwi5chBEXPqbVvXzUAPcXake1x9I71jP1flFDq+tn1hnAkQCY5ch/mppNPGi  
U4p00J2zUTA3PvTnhucUnRF+U7Gs3gDYsbglOSbyd5FGUVJUTu0X53GCO8rJDHy3qvK57FwW04aB  
WkRBAP1oPMkkkkMxJmW/PWi+GIDquAwg9Pan6pRyTGeNmbAMdDXmIM/715eleZdGY+FLYcmm3r0q  
nMYLoEdk1LJLTZjy896Ez2XcenFCCJ8LEHreTSKKaoZPBoP+n4oJZakMoKktewg+1TR2KMMQXMD/  
AGmbGRyilmWOIyIysuqDMj118qLbbDVwyypELGwPY8xvUnBgx4FPhoZh3Ui1tJB5RB3oDPPJGqNa  
2B06S3+4fml/F8fSWAEAX7jYiKM4dx01AuMq4qnaFMA7HnTw4nFXyVFPg7EzBMFhHl0r3L8QfWCN  
43N7jb6VdxjLqro+G4fDcSs/Mu0qfKqWPjILAwZA+tU62sjPIfluLYr4mgsCJ6CpTvKObhQZ59aV  
ZXE17bwTPpRGfmcUADb1pXBLSQmfDcPmQXII1AN8sXFzB+29VY2OxJM6B0Av0N4k3586Ey4YPKQS  
zGWJiNxYc79bV2IXBJdlm+51n0AgA22rkSfo6cvSS4TESSTBBBJNWrl2VSbIRf89xQeJiAz/wB1  
5ttCgbchBofGwhZtbA9S7c/M/SqLKodNhrZVgxa5n+kC32tVyGLkFSJtPnVGRd4+cm8Ac+x3qzNh  
wssAwF5te9xHlpX2s1tFuNhKfFLSt5Bb7fzevfitpOkzB5ib+1KmYBmtaJX15HeNqc4WKCoyCQYJ  
I5c4PpRdmuy3CxnZTqAsREH8elMsLOYQANJEgi03vs09KczjhAMQoWG8C1iofe83qODjo5+INSG  
40svtYWN+/KnSdWa3de84SxkXtvY7bE85tWfUupMxZgZt088yBT/AASBYc3sepPLeZ2rMcQLBrSR  
v2/lqMG7M3R7mcRx411IbXGmoZvEJVA4brqteeopNh54lyGLGbagbD6bVZhZxT4GktyP71VQdEUm  
OERAmqSvc/4JNEcELKTJDI1tSmQDyPUdD6UixcrNy7SbBZNqb8Ix1LGGNII0iCetDqvASwmaDEgG/  
8FK04kmWdsw6lgDoUDrJ+wFC5ziLC3K3saR/qjNyMNBsWL/QAfSnjG3QeN1Z9N4F+rsHmtoGpX3K  
sOXUHbnWhfMp08xXyHgKf6ZHxZBcrAg7C0jzr0/rV5skm/n53MbVRRXhW/p9D/U/6o/0qAjCL15C  
7gAxzP4rC4H6/wAyGDsUfDPzppFhMRtPlo3LcaXHw2XF06WvDbClWd/TuEUJw2Yzex1Afn3rXH01  
Pw3WouG2CGwzOG6TE7Wnff8A4rPtIU2WfQssrkBjPEcec7i5pP8ApjN4uGWwH+UjSO3Qg+VP+FYJ  
GUOHYU6sNue4Mc46VCSpBlmJlWwtIPxFGqCdSxHrFqVtmIcEDUF01xNHvkcRdWkjECzMbJ/cpuKF  
R0ctrYqdNiBz6Wp41+yFkczjq7wzsik2nZek9aGRG1kMYyjwHYNcCZ2iKu04QU6pZt1I5Hoakwcw  
jqVxHiB4VEST07U9lpGQzKMqqcR2YKSSlp/+LGxrx2V0dMRwyKRpgQYmW0jm0YonLZk4mAQMqmhv  
EWWxWCCl771SMXQACEcSIKgsRyMGox/s0q8E3GHbCfDRHlQgutizJn6Vuf0vj4eLhFGE1BDahfre  
szmcdy+nQ0A2aFI/+tT4fiBS5RyjtAdSIBg3IO01TtjJvAjMsQ4hUdJKlWF7G0H/AGkUt4vi5cns  
oKlQDAjwHmJ7/ing00qTIBiAw6jeR5GsvidFkh1BDSJUD3oQyGndjTgj4DEI91bY3BDde3SrjwB3  
xGfCAZBYEkAmBH4rP5XCKOJJBXYciKtGaxQxc6lF4JmIvsNqPRqTae0ZJphqZLFwzodIsxFwbxba  
muV4UdC6nvF9qG/T+XLicw8swNscBH0orDyWDiAYnxGXVeDNqWTUdjJIdlGE6XCAC5IEAHvO+53o  
NsUL8g1k7vP8+lNshwzWGBBcksBAMOp7VHM5QDZCItLAjnymuOM1thT9YlwSzeACDlue/wAxmiv9  
HJGtje4BP860bhKASBvfl+a9TDYwQbjsPvTSlWh2kiOfkKwFkie+1+v1TLHRVS5WIsW2360M4kXE  
9e9B4yQAumwkDt+1NGVoasA2aZFYqjXHMxHnMH+GistnfcFZDPVVMH81Ti61HzKoI6XMRavQwLgA  
kTewMz5UdoXWRoMbCCOhZQW3DjStE2/kKHGC2HqKRpcQf8e9ChmcHXpK8iYgdt95+9FpplgIteev  
f2NB2jbKy7qoAYet579TQOawFdgXBC3JAmbDYc4/emGJgmDOW5dvPfakvEXdfkZgNovY9ibiek00  
MsDTrJ6vDi6wmEwXlqt5eda4HAHL+GFjdieY351cm87FA2uVAg3gHlyfej8hw5Mms2IWPmCgDva9  
6sm16SuhZjopEkkONtNy0dutMeFZF1SwnUSb7wPsSapTgZxmXEDkLJg7W3sOfSm+YcJh6EnwjfmY  
6mlcvjyXjB03L0zOfY69LAhtiCI8qXvhyw1brJA6loczxBXdf68z1j1XmVw10F3AMXuBysB9ftTd  
mtoiuOnaZmsTPh/CwiNiPzQb5Yk7j70xz+GriV+bew3pUUCda6IvGAsa5HFCL4oB5RMnrvtRQ4ox  
ICQog2H3796z7hj8u9EZXBxAQQdv2gz6HahKKcmzYcNzBILncDpz2tTpVP8Ap2gWQAY9QRf0NJOE  
YDBFUi035PfetXk8raDbz+ornkiyVqjEhi7YjBtYTE1SSFA1MYhta4obiGXV1Z9IDneLISNyB1NU  
cdyBXMvhhAgDQOQI5NO0Gd6hm+HosKc8ryZKqrkf+0Qar1imqOSUaYG+EQhbWekdMTczzn50flf  
hKod11E/KoMgdyaY4n6YV8PXhM2IAQNY27ggb351nHyj4eI2Fq0nVF9j0rKSmmkxw7GSZxcv1DMS  
s7AXXoADY1PEyjawYtspuNvzQGCzFjCDUSNthG9PcASBDG29zIk7n2pZLroZJs0PDuHkLqZ1A0Dk  
DJ70r4rw/AKNiI+1lIDr002mKIyeWUAFcdizX0kypINx2tep8S4OmLmCzmG03C2DR1IPfXbI0dgO  
HlyjIreIOsTy8QtHfa9LXcaSuspEXYT2I7itWmUw8bDR01GQAJq5Rs09L8zwnDkM5Aknw6oDTuK3

amNKSukJc3wWUXFV2ZtiFFvMU3yP6cZcDRiNOu5EyQIsI5GrcLMukqmCAq7AST7xApjg5osoZ10E  
2IJmPas5S1YrTRB8IooRMIEBdMzy/egcLh5AA0taruKZrFRVbDQmpnxEmLb7VR/lodvesoyMoyNd  
lcPwAMjm0+AgMscwCLnnXuWzLsSuHmUxgN8LMLpcdpPOPKq2UgjQDq84H88qXZricmMxhDEtBx8u  
It/7139ahwtOJWLwP0ySNOvBfBbmVYAT5XMelQXI4JgKHHk3+KrymMhw1OHiiYjLyD7jtEVNX5Hfn  
EmPXaq9I/CtJnr8FR18OI6nl035xE0JmeD4iRHjVbiN/YmTRKSLqRlvJ+1qv+O4iIA77nyAmIFF8  
cWhWjOFiJlGBvZhHlvahizG5UqOx+vWtVmMP4nhdQw2Oobc/MW70nzf6eGoNhqYPUnblpHxUBrAu  
OX1Jpgsx3tYjp05qWSyb4SlbaTfxMTznmbD/ADRGDwXMggRI5nwgDn1n6UY3AMdrkr6kCfvPrQ6y  
qgIAX8e5B0mLHuI5XvSrj2XD5dinzdDoI2pJnuH46C+GYHMR+Deg8zhhttpYysmxJn+cq0YuLsDuh  
dwrMOERVdAWuWZtWntI2HSnGiYOIQWakgGVInc9eVJMDLrhrKkqzEhrdzYdKapilkAAAltjJsR+Ka  
bzZLCyMMY5kAfLyAFvIUPjZHFxCUAVF2LMPsBcmg8DHZX0FjOk6BpIBYCdIabyJ9qIfn6kLM8qPm  
g7djf6FPwMuSTSBm7xHIDAXUnGR4Ia0yDO0XM/SmWfhstilBFz9v3FE5fJo80iIqwYaJJ9Db1NeZ  
KNDglidQsYiR5AU7t1/A0ZdnVGU4bwtlTViWziFCnkOc96PweGoQQBc8yNv5FMjhaidWwuPS5+5o  
jL4ABHQ/f9ulVcm8lVBIW4XAlg22EfX9quy/CIMASbW9tzH2p+EjflP88uXej+E5eXBI1ATJ+vtS  
uTGpIVYeEYOFIECIH5pwxKhWPPke4t/zR3EcBCw/uUift9KJxcv/ANsCx0g26g3EepoUEzfGsor4  
HxNQRgQrOs6o2AOkibedYHiuXy6CFbEZ43KAD63r6flVJTFXcalj1A5cqUBFxcPrj4LLIIGpZYR/  
UCNwVl7dXboflTTs+ecM+O5GCuI6IAWADEAHfydTtF8AS+MmO64OZOpdcq5s2oAwRnvpIkeladK5  
ZMAT8JS7RPjHiYwsvKlfD0xmZjuhtDwCpBgoVZgTtMwx503dSi3RHNmgfCyxchMJbWLKIHuN6Gzn  
DFZdYECfl6dqfYGS+Agd2DIgJMLEaOnOkuPxrDIbELAjcxPM7L5xU4JydjZSE44i2A4OkkRpgLIH  
n0q8Zc4rk6UEEKZ3vA6UFitimidZU6XMgDmK7M8UCK6QVIEgbGSKbN0thcqHWYz6YCBZJtYbnlpZ  
mOKY0K6Jht4j4d2PQg8rdqWZBsJyVxXKuWfpuOldZnF+G4ZRghiBuZAAAnzpowskyjeWalG4piPh  
NoAR0uw3gDcVNcyThLjtAk6HXcG0yOludQ4M2HilcTCXQfEMRCLmQRI/uEnah3TGOAcPDwprjN5B  
Xy5R3olFKtBVUaHgWdI5SSikoQIPSOlqT4/DAWJCLf8AnSgef42Pggh7sLAEgeH03501/wBcWvp3  
7/4qTnOLpZEc5RwaBmaAp3Yct7cp7ikWdcH+krExtAFvpTvMYbaDqbUwMgx0pVm8ZgPlkX353/xx  
PDZd0ti3B4iUYpqhJsbyDub9OvpcU6wc7YarT3t1nuYuKQ5hG3A7dx3+lAa3SLllvY/iuuMvAqTW  
zf40aDW5XJJ63/yavTNSJiDcgeQ5/esXwviJZWlvFJn16e5rQZbMSFIG9rf7ace7HmRxYQAm8+Ik  
99/UxR+C8je8C/0Ples/lSG08t/Sbfb70blczYGZlW5dNv53rGGaYoJLTikgwZII8JFu4NDZl2Q  
6gCV5sCbdPDMXrxGAU3Hjk9pN/vNUYgYqVVocNKzz3kdwb270xgtM4n9bW5HxQZvvUcxg4Di4kDb  
94/xWexeIPhtp06iTZSYJkGNLGxPh08fkmpmiQdLMjLurgKEESCH0zEHe3TcUKsFEOIcADj/ALZ0  
+dh9FP3pBj8FzSMNGkiRrggyOYA3mOdq1OC7wNUkEXKywB/3apBonExQFJ3i5kSR3iJ+9DqvRHCL  
M2yNqGsDQviIYXtf0g3pFg5XWWXVGoEBLXeb+LtWlGMMIDpVXBEEjsfTrUHYiYaeFFQDuAPcnekU  
KMolhGay/B8REGH8QsJmFW45ke/opWvcdSAEW+0xf3PpTHNviHwoFRd2LMR5yQpEdjVGDlX+Z0Jk  
2YENaf8Axmev7GqUUjFI9w8AaQ0SQfoeo6lcmUAYmDztvt5SKIXAhd5Ahtlm3IGp4OnfYHeeX8N  
AYrwcFmWVE29R6c6c8Mwl0hogx7H9v50qPDsMB2InrH3HnNeYWMUc28MzHY9B6+9KzEcW2Ix3v8A  
z6DenaqGtTSDNXckHew7jce23pTDhWOChEyR9qwGePlQmIzDZ0W3KVgD6VlflBxDM4OJKkhFbxMQ  
CrCZAIJBgitfmcY68McmDfj+elZTjHEUTM4mFieLCcLq38BAABn2NKJPRHPcRR0DFVeD4NBIaD2B  
teoZfDZyAusOTG4n67UKuSK/I6sm4YLHoTsDsZw7FyHYGSCSTB8jSVK6Wjnbzk+gY2VjCZfBVksA  
WPUGxr4yODYiFg0gobqdie09udbr9I59wr4eKxYKfASZMcfWjOOZzmdMRAhZGJYMYBXSQJsaE+  
snEZ01ZmOI/qDWgRERFEKniuO8AQpegcXKYOCnFSUvJMG/JpHLauz/DsVsRnbD02DeGNJ66TsfvU  
8bExXCBHZtAI0EbCOnOrUrwJ1+HcMyLO2rSjBBdR8wHY8qJwkAcaHCGfACJ1CIImYIBFAMmIFUI6K  
QSXkkXPY7j96Z5XPMEYAqGAftztelbUssKxm6RLCXHXFIJBDGdURHKKeZzNhFlisbADqaQ5Z3LK  
zXYA2J2t0rhiquITiMwY/KvIfvU3G3YqlSLuK4a4phEb4nhJBYY6Dr6VDCyOMAP1/8A6qjOemqF  
cntZ0/Sau0N/efcUVaF7Xs2oQhSCZBkm1/L+e9KM0gDGfK+vpTDNPZVCyAf0OV73oHO4lwr/UvL6  
ly3aR0S8Fj7d5P70vzWXYryuOthICT5+VdiIZg2AH8vVYyZlctmcGF8NtSss8+9/OnXCeKgtPne8  
cwf80uz+XgzI78qUYuCAQUaGHQ/muqLUthvro+iYecCtvPhPkYuI+9MOHZNwQJtqj/2BH5+tYXI8  
UD6VazAAE8iRb6i0U9y2d0LC33F/ceWwj/NaqHUrNMmYuCNtvyDHvRGWxFZVc22Y/ePsfPwZTPBi  
AAezjzAmft7UamaIOsLg7wd95t60QjXN4SsQpvcEWHI29efnVbPrAV5tdWUKXG5nrcyDl9ao+Nca  
k2gqR5f8ewqeYxpWbfbWtbof52oWYkmZ0PBYDcLq1Tz8LG6nt9hTUPSQJ6qdxSZXV0F5EG/rsR15V  
fgZmFENEWvf0M38v8VrNRdmuGIX14cAn5oI8XUQbTee/3p+AFkaokzBSJ67SPpXJmgR000Tcj/8A  
QPL0r1seZUeIHZEdx/yD+Qajx10iUtztch/4nr712ER80BS24E6T3v8AzyqzDDaYkkd+X7/mp4WV  
vcwGi24Ppy89qNhIrgjZRbmN/frS0pp8Dxa3sPvH0rTJh6V21el/LzisvxN4Y8559CBPlifwLNY6  
4e2lCZvFvIWH1kULjZkBxB5wPWNPs0ikud4iW/7abCCx8iCV+hFSRY7t5T63j6xWMhlmX3K8jPnB  
I+1E8GcFwRvcGhcJSQ02uSfi7/Ufer8q64ZU73/bnQYRhxFCCjAToafQidWT/VuOBiWQNqRdZPl9  
dhW5x01owG5FqxnGMPxI5IA0AMivKkrvz2pXS2S5fxMsnEDhQrBWw3EOkk6fi/mn+WwUOGLjQICK

bx/5d/vvf+jwHA1rpnblvb0oTHypy7SjasIjxKTt/j7UXnRHsmqZTiYhR20IYk2UXItf6U6ymfQo  
 pIKuLQ3TvS/MYqKMPFw7qTDHpyv0NHYiAi4kGpOKu2I7i8ijjWIXYfCWFkFlMgD/AM15HuK5crgu  
 2pczoe4ExpI5im2DpGoETK6R0HTyobK8OYFw4RcP+kKLnq17VX+oooKml0zXGuE42GusurjVHUGe  
 defp5YMRVxABrIW4I32N61icJwwZA226e21DcV4R8VlZ3ssRFiCDiVWXNGSpgrc2JuM5d8tj60Xu  
 u5BHMxrs/gYmMUZMKABJlgCT060zz2K7hEK6gDzIkHrPSgXXEkt8NtDC51f/AJOxp1JAtC9+GZsM  
 CqkCRPOP3pj/AKgiwmltvf60IuNjo6F3aNEQw9L0R/1TE5fas8jKkaw45IvHtVbpIHbbtUENdms  
 0EWTzMC8CYm/tXiKUpSSTIxLKTqxccOGixI630luVVxrIiRCT06kb73oknXpYwCd4G9SwsFQJEk9  
 zYfztXpQtRzs74RxTAHwBHyme5BJ/alGPhAEmCb37+laBsJpIkwLyR7AQKV5ggMVG/lyU8ZNCyaQ  
 ozWXsYgCvclxXfKEPJjY/g0Xi4JJE370JiZcXIqsZrTJ9qeBzls0jQ6HlB/n19Kb4eaisIvFjsbfQ  
 73rCqHVgV5dnJTDLcUKkhxYj2/5p6+FFP6bhGNo20348rmI7mrCZGoX6idj/AD61lsnxRYgGfWDW  
 iTNKyl0IkjrZ8xtP3FKx07KXcp4hseXI/gVHDzIJMSwJgjnPl2otswpgOAQefMdj3sb0rz4QEFDP  
 YddjGx7HvWCaAkhVOnUpo/hyBpLDUOX5B5TtcVm+GcaLAIwk729Zjlg+tHYfFiGbUQB19N6DYUzW  
 YSLEWj13HKe/7d68dkAgACJt0rI5rjJ0Qranmw9In3Aqa559AOqCGkk7EdPX80GY0bzSqb269D/B  
 WK4nnZZglzefaF/A9aJxOIBFbcFpsf52pFlFaZ7ye/8AcPKxoxQGw7h6i5Jkn6yQfua2GRwF0KIG  
 0Soc3B9iKz2WyM36H67g/wCaf5TM+gv7T+LUQnmNhlYBF7g9wQCPqKhhJqdbHSSARz3/ABbzmicz  
 jgkc5+xuKKy2CFIJ8/59PrSmsJR9JjtfsRWH/XAZGUglQWMQJsdFsINbV2M7XJ37f8AFZr9a5ZX  
 w0ZgCJJK3N78rUNk51ldmK4RxVGlMUKAkajs3cVPLocUFImP6ojymvWCosQFHKqsDjAUHUhAG0bya  
 Zr+05lnSDcLLNgGZ14REOnTv5UZgZAodeHilsI7Kbx2M7edVYXEUInVA72BocZ7/AEx1oVbDcwYT  
 9u32pMjpt4YV/q4JDAKQJ338hvXZbPer/UVN5I/lqF4xoVsPFE1GnxbhOek/WrcDOiYwhnTApelk  
 pRawTXOY2o/KF5CL/mpkljLMD9qghJuaHzWfVBbxHoP35VlD4BF/EXARiX0k2n9qyeGuJrhcUgna  
 SYPQdKs4hlSBHxQSSKAAfEC06kVa3AIJOgelzV4xjFbHSS2UZ3M46ko6iV2mL9/K1K8TOtJ3/wDa  
 n2Zw20DV8yCFuDPbrS34oN90/lVE18H6o+ho80Nxo/8AbT/ePtFcArtXcRacu5sSpUwfMV4fEqmm  
 c/F+RblmSB/VHTy26V5iYwX5FIHcd+gHaqMtnvAI0rYWak7Gp4mNCKhmpOBXdR6CurBcVsZ9pCx0  
 I95v0ofHyzf10BI/uv6Ci3ckQR56mNBuZPKT0H5p7FkkDBov5/gqgkMYHLeimTVteedDHDN6KJOJ  
 UydV80Ni4cn7CjMfFNgBftVGKk/imi2tgWNgv+m338+9SwcTET5HYdtXv6LJA6UWUvSb3pnI1ZsG  
 TimKImp39KIY2K7yd03IEW9DBiqMXAjjvXuUzIwYsXgG16N3oaTtkMcVjhPmVlflY3NCZrMMzEspv  
 vMzTEQ8RBWA07/WpZ9MNCmMSZmCsir5Tv+1LlMlcn6B5YoLglT0m/wDIn3pq+aCp4mAFjj25c/Ss  
 78SRKZHI+sfioI5cFztVotLIyd0zuIZsuQqMYWRM7zWm4UnhBPODA+v871mMngDVpJjotWu4DjKh  
 0PtY09F/EVivpoMniACDcd7UVjYawHUwZ+vQ+Yt6lQ+XgSu3Lt/g0F8bdDYzz+xFIUCsJmnURESP  
 I71di8WAYKTO31G/a9Ca2UbEW26mLHzH1oN8qrkMx0hrx0gzA9zQMPMHYcWtex69ftST9bZgDLg  
 sHMGsVAEWsTPSf5amuUyyq5csNA+RdotH4rE/r/ieHiOgSTolSV5be+59qMvKsbwZfioYbEuJIgD  
 uBsNquxcynw7DSIhgbme/qKGfGZVBVoG9oBjlqvBzCsRJnnJ3v171XrgiUY+YEjTyv5U8yuKcTD0  
 YkEASDsQeVcTnMPQyQCTYURz6f8AFD5PNKRGTZQSY8961X4GLH2RxiilcVQCj7MpMx+1A5/KMmIW  
 BlGkpFpHQ9xzqGJhnFUOrAKfmvyHemWQxgR8LEup+VuhqbaTwZSi3RRls4Qdtjw2tN60eWwss4jS  
 JPI2P+azuN1VAZCsQT/UZI5GeYiKafpjCUo2oE6PCNUbRM1LlS63YnIqC83kVQeH5SDHagy4kDkB  
 TrHw1KkWA+1ZLiud+HiaEZXBZeNtp2JtS8LclXpNK9DJ0Q3dQQL36edVtgcPa979GaPtWexOIM4  
 JItGw2pN/q3/ALo7Sa6owl9HUA9PouexHVzDWvEgftXiKXwcSTf5ffY+9dXV56iqM0juDYiFFX+q  
 Om1WY6iCASRM1ldVns6l+ILpjckBznVwxFOQoAHl+9dXUBskN/bseleYiALIHn0HeurqPoAM4J6  
 yaqGXg3/AJ+9dXUG8kZfkMPxCltqtRzME1ldWYDjMUv4rhkgqLncAXP8g1ldVOPY60F8IRkQhhe  
 QYmSBclNuIqdSlgRqkqRMdDXV1VebEeHGVZvOMDoQAajZdgN4ovLZR0QlZJjny5V1dTLsgj+RPAQ  
 MSOfKtHkcElZAmNx+a6upZHREdcOxivhNxtVmcyoN5rq6lY5RjNjW0w6g2IvbpFW5nIF4LE8iVA6  
 d66upWAo4rnlCaLyVKkzLLI286+e4GScsZeQu6+VdXU8SO9lWJib5RG0fSl78POog2HKBvXldVYg  
 cUUYuEyW002NDKh29+ldXVRaJvYxyGDilWVJKGJ6U6yzE4YBlXFu89a6uqHI9iPDGPDcc4gOG8fE  
 QeAn+odJqrM8RdEkoShElhHinpXV1JS7DS8FOU42+tdR1BjBlywjyneneazHIKoogf0yZPxtbpXl  
 dR5F1qjOTS6laYaxBW9Vf9Lwead3Ne11TUMTP//Z - local variable 'contenido\_imagen'  
 referenced before assignment  
 EXITO - Almacenado [https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcrKi  
 ct9-QmE8TRXZnxTxYlSX3EEvIH-r1xFwA&usqp=CAU](https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcrKi<br/>
  ct9-QmE8TRXZnxTxYlSX3EEvIH-r1xFwA&usqp=CAU) - como ../datos/perezoso-hormigue  
 ro/entrenamiento/oso\_perezoso/c1a7e028c7.jpg  
 EXITO - Almacenado [https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSfc  
 70hAfREG4eWW6EwzNE1WNdrnFW12PEtMA&usqp=CAU](https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSfc<br/>
  70hAfREG4eWW6EwzNE1WNdrnFW12PEtMA&usqp=CAU) - como ../datos/perezoso-hormigue  
 ro/entrenamiento/oso\_perezoso/f31c131675.jpg  
 EXITO - Almacenado <https://i.ytimg.com/vi/g84asJsXS-U/hqdefault.jpg> - como .

./datos/perezoso-hormiguero/entrenamiento/oso\_perezoso/e7169b9681.jpg  
EXITO - Almacenado <https://us.123rf.com/450wm/kovalova/kovalova1703/kovalova170300001/74375113-pereza-feliz-descansando-sobre-un-fondo-blanco-limpio.jpg?ver=6> - como ../datos/perezoso-hormiguero/entrenamiento/oso\_perezoso/4cd22a8202.jpg  
EXITO - Almacenado <https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcThPzdlEVUCP4uhj9y5bvN3ZQMnpttz-NWh6Q&usqp=CAU> - como ../datos/perezoso-hormiguero/entrenamiento/oso\_perezoso/9b6a1471f1.jpg  
EXITO - Almacenado <https://thumbs.dreamstime.com/z/oso-perezoso-de-tres-puntas-movi%C3%A9ndose-en-los-%C3%A1rboles-costa-rica-animal-caracter%C3%ADstico-del-bosque-tropical-172814761.jpg> - como ../datos/perezoso-hormiguero/entrenamiento/oso\_perezoso/4ecb96f8d3.jpg  
EXITO - Almacenado <https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRbiOylxCo4rwGyIUxHAYX4tBISQeGauszn-w&usqp=CAU> - como ../datos/perezoso-hormiguero/entrenamiento/oso\_perezoso/1fd237f6b8.jpg  
EXITO - Almacenado [https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQVWJjZS\\_c3lHSGE65-WKf\\_bMcBr2Hhe730Q&usqp=CAU](https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQVWJjZS_c3lHSGE65-WKf_bMcBr2Hhe730Q&usqp=CAU) - como ../datos/perezoso-hormiguero/entrenamiento/oso\_perezoso/345244b41a.jpg  
EXITO - Almacenado <https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTfcPaL6qKSgRyMkhyKvzFUSGDpoYdhg79Mhw&usqp=CAU> - como ../datos/perezoso-hormiguero/entrenamiento/oso\_perezoso/30d3138055.jpg  
EXITO - Almacenado <https://fotografias.lasexta.com/clipping/cmsimages02/2016/07/05/4454BC8F-57AF-4873-8EA0-DB2117C964C0/58.jpg> - como ../datos/perezoso-hormiguero/entrenamiento/oso\_perezoso/bbe8998206.jpg  
EXITO - Almacenado [https://st2.depositphotos.com/3729457/5241/i/600/depositphotos\\_52410571-stock-photo-sloth-in-puerto-viejo-costa.jpg](https://st2.depositphotos.com/3729457/5241/i/600/depositphotos_52410571-stock-photo-sloth-in-puerto-viejo-costa.jpg) - como ../datos/perezoso-hormiguero/entrenamiento/oso\_perezoso/7c09a159e7.jpg  
EXITO - Almacenado <https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSWf5GDdvykC0FixkSpvtEdX5ChxzpsbhdRQ&usqp=CAU> - como ../datos/perezoso-hormiguero/entrenamiento/oso\_perezoso/ef224ae8af.jpg  
EXITO - Almacenado [https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSq\\_mwt6d3VknGfCVMLd8ZsXWklOFTADaP-hg&usqp=CAU](https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSq_mwt6d3VknGfCVMLd8ZsXWklOFTADaP-hg&usqp=CAU) - como ../datos/perezoso-hormiguero/entrenamiento/oso\_perezoso/3ea750951c.jpg  
EXITO - Almacenado <https://lanotapositiva.com/wp-content/uploads/2019/11/aoisdnoa-homadas.jpg> - como ../datos/perezoso-hormiguero/entrenamiento/oso\_perezoso/c5c433fbbd.jpg  
EXITO - Almacenado [https://ichef.bbci.co.uk/news/640/amz/worldservice/live/assets/images/2016/01/26/160126012442\\_oso\\_perezoso\\_ecuador\\_624x351\\_comisiónde transitodeecuador\\_nocredit.jpg](https://ichef.bbci.co.uk/news/640/amz/worldservice/live/assets/images/2016/01/26/160126012442_oso_perezoso_ecuador_624x351_comisiónde transitodeecuador_nocredit.jpg) - como ../datos/perezoso-hormiguero/entrenamiento/oso\_perezoso/a004271f43.jpg  
EXITO - Almacenado <https://static.wikia.nocookie.net/mascotas-inusuales-info/rmate-aqui/images/6/67/Oso-perezoso-colgado-228x300.jpg/revision/latest?cb=20130407095909&path-prefix=es> - como ../datos/perezoso-hormiguero/entrenamiento/oso\_perezoso/91f9f78503.jpg  
EXITO - Almacenado <https://thumbs.dreamstime.com/b/caricatura-oso-perezoso-en-divertida-pose-feliz-sonriendo-personaje-animal-aislado-sobre-fondo-blanco-estilo-escandinavo-207586430.jpg> - como ../datos/perezoso-hormiguero/entrenamiento/oso\_perezoso/ca38b697f9.jpg  
EXITO - Almacenado [https://pbs.twimg.com/profile\\_images/670330843978530816/d5VwzZwF\\_400x400.jpg](https://pbs.twimg.com/profile_images/670330843978530816/d5VwzZwF_400x400.jpg) - como ../datos/perezoso-hormiguero/entrenamiento/oso\_perezoso/c36e71c7eb.jpg  
EXITO - Almacenado [https://www.heraldousa.com/u/fotografias/m/2021/11/10/f608x342-18877\\_48600\\_15.jpg](https://www.heraldousa.com/u/fotografias/m/2021/11/10/f608x342-18877_48600_15.jpg) - como ../datos/perezoso-hormiguero/entrenamiento/o



so\_perezoso/b077b7d8f2.jpg

EXITO - Almacenado <https://fotografias-compromiso.atresmedia.com/clipping/cmimages02/2015/06/10/FED2B774-C946-4F22-84B6-19BF47CA15F8/58.jpg> - como ../datos/perezoso-hormiguero/entrenamiento/oso\_perezoso/a8e712d21e.jpg

EXITO - Almacenado <https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcR5t4iaTpEcxMT-AkQ2qwSOemuC7tMZKBCNg&usqp=CAU> - como ../datos/perezoso-hormiguero/entrenamiento/oso\_perezoso/26bd6c727f.jpg

EXITO - Almacenado <https://efeagro.com/wp-content/uploads/2022/06/5426912.jpg> - como ../datos/perezoso-hormiguero/entrenamiento/oso\_perezoso/957ce654bd.jpg

EXITO - Almacenado <https://cloudfront-us-east-1.images.arcpublishing.com/semana/EGZ2QN5UMRDWPHXYLNIZW3DFQA.jpg> - como ../datos/perezoso-hormiguero/entrenamiento/oso\_perezoso/07f32a6f62.jpg

EXITO - Almacenado [https://st2.depositphotos.com/5909840/8643/i/600/depositphotos\\_86439650-stock-photo-happy-sloth.jpg](https://st2.depositphotos.com/5909840/8643/i/600/depositphotos_86439650-stock-photo-happy-sloth.jpg) - como ../datos/perezoso-hormiguero/entrenamiento/oso\_perezoso/571811a577.jpg

EXITO - Almacenado <https://i.pinimg.com/550x/de/cf/2a/decf2ae44b336a0109206eaa6622e95d.jpg> - como ../datos/perezoso-hormiguero/entrenamiento/oso\_perezoso/a51b728d63.jpg

EXITO - Almacenado <https://www.fundacionaquae.org/wp-content/uploads/2019/10/PEREZOSO.jpg> - como ../datos/perezoso-hormiguero/entrenamiento/oso\_perezoso/4c68ea36e7.jpg

EXITO - Almacenado [https://t2.ev.ltmcdn.com/es/posts/0/1/6/el\\_oso\\_perezoso\\_esta\\_en\\_peligro\\_de\\_extincion\\_3610\\_orig.jpg](https://t2.ev.ltmcdn.com/es/posts/0/1/6/el_oso_perezoso_esta_en_peligro_de_extincion_3610_orig.jpg) - como ../datos/perezoso-hormiguero/entrenamiento/oso\_perezoso/845fclld7f1.jpg

EXITO - Almacenado [https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQmfqmgsl49cIt8xFKTydzNrp-\\_QUS-TZujEg&usqp=CAU](https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQmfqmgsl49cIt8xFKTydzNrp-_QUS-TZujEg&usqp=CAU) - como ../datos/perezoso-hormiguero/entrenamiento/oso\_perezoso/cce611c566.jpg

EXITO - Almacenado <https://static.vecteezy.com/system/resources/thumbnails/000/125/628/small/sloth-in-jungle-rainforest.jpg> - como ../datos/perezoso-hormiguero/entrenamiento/oso\_perezoso/1da23de39f.jpg

EXITO - Almacenado <https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcROP43YxV2e0yZdhUBdyZp0g53H91RNsY70g&usqp=CAU> - como ../datos/perezoso-hormiguero/entrenamiento/oso\_perezoso/456e824030.jpg

EXITO - Almacenado <https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcR81lFKb6ErihhWPePGCgIWPCWV6fsbm8L9Jg&usqp=CAU> - como ../datos/perezoso-hormiguero/entrenamiento/oso\_perezoso/3c28daf479.jpg

EXITO - Almacenado <https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRHZTjP9JG1bfGCFmw1FpCmrXZ-dawugnojja&usqp=CAU> - como ../datos/perezoso-hormiguero/entrenamiento/oso\_perezoso/398714c518.jpg

EXITO - Almacenado [https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQ52HTztjl7oNfmGdzvW\\_m\\_eSfUdP3bZdLLZw&usqp=CAU](https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQ52HTztjl7oNfmGdzvW_m_eSfUdP3bZdLLZw&usqp=CAU) - como ../datos/perezoso-hormiguero/entrenamiento/oso\_perezoso/b58ba2591f.jpg

EXITO - Almacenado <https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQNUmSL1T8joCLjXzZwfPgF5N9zCi5vsQOVYQ&usqp=CAU> - como ../datos/perezoso-hormiguero/entrenamiento/oso\_perezoso/2d75584fed.jpg

EXITO - Almacenado [https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSjMbgs6VG7yWskKDJkZ\\_XZCajo3mMGdJ05ug&usqp=CAU](https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSjMbgs6VG7yWskKDJkZ_XZCajo3mMGdJ05ug&usqp=CAU) - como ../datos/perezoso-hormiguero/entrenamiento/oso\_perezoso/b3dab7ebbd.jpg

EXITO - Almacenado <https://www.prensa-latina.cu/wp-content/uploads/2021/07/Oso-Perezoso.jpg> - como ../datos/perezoso-hormiguero/entrenamiento/oso\_perezoso/d9a18ald06.jpg

EXITO - Almacenado <https://www.tintiando.com/wp-content/uploads/2019/09/OSO->



Page 25 of 53

gF3CxHWbSPL8VTlCiviIyGA1uo6Wop8cN/3Q1wbj70PmcyBmS4EgoDHekfZpv+P+jSblHsaPXiNg  
BkSchlm2Nt5HlSJ/1Bi4jhAQWWRpAie59KZpmAyMiuRrQjEEExFv6Z/qj1WXYwCYivN10iSPmEHST  
6WqjggrTlKek2kDYZkyRM/emuSSBtS3DxdWgmFm8r8opzhKzHUBHYRFmx4oozIBs0AEkXrOZ/Mr  
clYGwim/6mxygWFltRkHy5lmgCdWsLuQNX5d6aEV+TJSxI03AMAYuEZPK1toql8IHHbBNhoU7+c  
1f8Ap/ShIR5W29mHYiqMvhs2exRAA0j26ig45bC4qrLP09ksMI7yV0FtQJ5copzkmbQhW2hWBs3T  
/NDcS4ew0vhAsWIDiwmRHtS5+MIh+EpKRFjc6toHrU5cb5GmwOKbyPs3mXVQWCGByYG3kaTnNeAQ  
y61NoAn3obH4gqQ2OPiKWIMWbTNzPWR0CJiQPEjjUkHZSLAnrSrioRoZcJxMJ00yAGAIuPTpUOKO  
iEaMQOegEQKry2Pr1Ii2IMED87V4nDQoOoglqQTsfIVJxUXTW/8ALE/aBMTMa0IF4043os5tvg6C  
gZtgxB1AedCYWKiME+HC/wB0iP8AmrscOHDq+pCI0jlVF2ukq/YylbopGAwQkAfc0HhA6wCZU+Hy  
mlMsvnSbQCwPMkW/NFZ+QBihIjHjC2n02p71f1Q2mIlJYCdUCZgGqyCbqDa8c+81dneLMuoPEBi  
BbeicrxFHwx8L89Ip3KXqWFXf8CZwjGRmm4HQf8ANW5HPuhKnXkeQ2j969zGVUEjDe4kjuCIHSh+  
F5MrqVjEakHeem1VfVoDQ6ynHMFV0hU0j1G/WjjnsJrnBSSBptSfL8NcrqVWPpRDo4t8M2A/p7VF  
8MLuhkbV8NQYEA2Jjy5n1+lKMRlawkCTAI3j6xzhplss7FzaSBfkDymrMPKqol9M9du/W1Io2780  
lCwKFKkggnsfr0q/MYrESCnyLzsPLnvRqsjnwlsfSTBJ9Cft2rs1gKotAAvP+aMk0hmKHJVpZ2iI  
I3EmfzRSIjL4T4ovBBivdBIJKkdjvHI2PQldw/KrqkOyneAZn3BtSJ269JXmgfDaRGwPW0XjaoKn  
iM2IiNr3/wAU/GS06CAGgcxI8496tbJKs+ERF5/c7VSMJJGutMYzOCGLWGmxkHtfbyqnLIVSCIA7  
biefajsxkheBBELSCfIekDyqCKFUodxzP0azfglCXOup2MFSO/ODHlJt2oHAyrgTqliLMDY/zpRm  
dyelixWwuOlXNALXwkJTE0M99ESrA+c0/1EZ6phuQwjoYMYLXtE+1JXzmIjgpEkEXvaaY5HQ4K6t  
D/2nwnzDXH0r08NXWA5uLwlie4b5T70Vi0x1bjRHL4mO+MzhmeTL6RYWibbVHDZkzC4eIhIcEEKz  
fl/O9PMvmsTCQIihJOlh/Ux8opVxAurKXUg7Ger6fY11JvFCJNYZRL0h+wJ9ga0GVQWidqWuwYpG  
5v78veaa5VTMaYtSS0UiqFX6hQlB/urP4R0kEiVNjPSeXetP+ppXDUGe+OIrPPlcViCMNgp5tb6U  
YJ1ZGcX2wHnATDx4w8RWEcX3vynatHiz8ItYMN7eKORnpWSy/BcySSMGR/vW/TnWnw4wmXGV1IX  
wnl0+YWPvTyxkpG0soDfOIqHDeJhZQfee9VthYcsCFDsLM4FiOh60LxnF0NoaAdAKuqjxEi0Wv50  
u4Rj6w4xCToBIJ/qj1SKLeUSk7eDs/11LqhZWIjYzfnTHAw1BtAcZ/VM9qC4UinFd3QMq3VSevSn  
HwEnVGm0Wv6zzN6MrWDKLsDyWcxMJGRV1Ame1dmWfUhbY2tt3Hnarcfn4aqUQlr3JG3WDzqzA44j  
4L4WgO4ICtHLMsa107YKBzmlcqjJqNIJA0gAt6+VE6zAPw9KiNj1EmxtMV6ugYRDAGTIt8vkaGTPs  
gRdKECRPPeRPI0kk9oVprKCM1iuzr4Anpf309aX4mjBdSJGg25G1Jm4ijKEZQW97/tRuW4iCpV1E  
Ed47Dt2NBckovKGSx2aybl2AUgWIIbVInYede80loxwytyNiset63WDxKEZMJQmlz1nSWPeDWLbi  
au5Lu7OTdoAqim5LQ3oJnMwA09DE9f4K7CXQRFSZ738jROyYKOpGGx1TIVrEnnBNB4+X0gLctZTH  
liHnePSnTTRmeZbHdiGLMDAjuDx2q7OYbl2Idt/7zUslgsphkIuACR9KvzTy7RAV0rN3oyZulxHV  
G0nXR02vYR5UImYyRlGSNyQPXtalEr4gtjBBPQDL++1B/C0nSviIkwe87CIiubs9HSnk5XYmLGat  
wxAg3Emx9KrXDNrRtv3qKrrgGCJ7isngbwLfmZMAXsd/KatyJAYMeR85HX60pORDHcHtttli/PrR  
uVymnxXkjaTHsfvR9BVs0K42rpv528uVe5vNBEZmBIUTa9Z3MiFJkqbXULW9IvVeHitiaQcUsOhJ  
36d/8VTu6ElEsHEEXAWVAhG9r32He1U4uPqElbXuLz7XhTVPw3R+QEwOm3UfmmGF13ZYAt350mXs  
0dzO4q6v1GIgEC/pt67e5rIYUOqO6wwEAjmbIFvStLmcNWBQggqPaOQPtWawsDWvwbySAdhc8/T  
bvVVMiKo4Z2Kme6MWlmtAn72BkbVRw3iLoQrr8VB6sojczVZgcPLnSu/OT71HHwGwnhHz3mxpIt6W  
SEXK8Gi4YqOgbCxs4U2BjUkiI8q7OZHwrgGGA1m4kfbpSrPZDQ+rCbQ4GoMDY9iJvTbh+aZ0Bez7  
MBtPP339adfskrWRZgIyEBhBWR6Hp9aZcPe9z97VfiAEUsbFYnSkhmNgPP8AnlodW2Pxysa5jiSo  
LKCQd9z6dN6QZriaO6lgbm88o7b38q0WQ/Th0+PeOd496pzP6ZosWXT2ERbZq0Ulos38BOHZon8j  
aT/aDv8AiYvWgyWYaOv0n05Ghst+ldJVxZhazHb7Uxfhj4aaJ4wLxZhrWaQUxXx7hKYiFlQsw+QC  
2kze3TttWOPCHwi5chBEXPqbVvXzUAPcXAkex9I71jPlf1FDq+tn1hnAkQCY5ch/mpnPNGiU4pO  
0J2zUTA3PvTnhucUnRF+U7Gs3gDYsbglOSbyd5FGUVJUTu0X53GCO8rJDHy3qvK57FxW04aBwKRB  
APloPMkKKkMxJmW/PWi+GIDquAwg9Pan6pRyTGeNmbAMdXmIM/715eleZdGY+FlYcmm3r0qnMYL  
oEdk1LJLTZjy896Ez2XceNFCCJ8LEHreTSKKaoZPBoP+n4oJZAKMoKktewg+1TR2KMMQXMD/AGmb  
GRyilmWOIyIysuqDMj118qLBbDVwyypELGwPY8xvUnBgx4FPhoZh3Ui1tJB5RB3oDPPJGqNa2B06  
S3+4fml/F8fSWAEAX7jYiKM4dx01AuMq4qnaFMA7HnTw4nFXyVfpg7EzBMFhHl0r3L8QfWCN43N7  
jb6VdxjLqgro+G4fDcSs/Mu0qfKGPwJiLAWZA+tU62sjPiFluLYr4mgsCJ6CpTvkObhQZ59aVZXEL  
7bwTPpRGFmcUADb1pXBLSQMfDcPmQXII1AN8sXFzB+29VY2OxJM6B0Av0N4k3586Ey4YPKQSGWJ  
iNxYc79bV2IXBJdlm+51n0AgA22rkSfo6cvSS4TESSTBBBJNWrld2VSbIRf89xQeJiAz/wB15ttC  
gbchBofGwhZtbA9S7c/M/SqLKodNhrZVgxa5n+kC32tVyGLkFSJtPnVGRd4+cm8Ac+x3qzNhwssa  
wF5te9xH1pX2s1tFuNhKfFLSt5Bb7fzevfitpOkzB5ib+1KmYBmtaJX15HeNqC4WKCoYCQYJI5c4  
PpRdmuy3CxnZTqAsREH8elMsLOYQANJEgi03vs09KczjHAMQoWG8C1iofe83qODjo5+INSG40sv  
tYWN+/KnSdWa3dE84SxkXtvY7bE85tWfUupMxZgZtO88yBT/AASBYc3sepPLeZ2rMcQLBrSRv2/1

qMG7M3R7mcRx411IbXGmoZvEJVA4brqteeopNh54lyGLGbagbD6bVzhZxT4GktyP71VQdEUmOERA  
mqSvc/4JNEcELKTJDilTsmQDyPUdD6UixcrNy7SbBZNqb8IxlGGNIIOiCetDqvASwmaDEgG/8FK0  
4kmWdsw6lgDoUDrJ+wFC5ziLC3K3saR/qjNyMNBsWL/QAfSnjG3QeN1Z9N4F+rsHMtoGpX3KsOXU  
HbnWhfMpO8xXyHgKf6ZHxZBcrAg7C0jzr0/rv5skm/n53MbVRRXhW/p9D/U/6o/0qAjCL15C7gAx  
zP4rC4H6/wAyGDsUfDPzppFhMRtPlO3LcaXHw2XF06WvDbClWd/TuEUJw2Yzex1Afn3rXH01Pw3W  
OuG2CGwzOG6TE7Wnff8A4rPtIU2WfQssrkBjPEcec7i5pP8ApjN4uGWwH+UjS03Qg+VP+FYJGUOH  
Yu6sNue4Mc46VCSpBlmJlWwtIPxFGqCdSxHrFqVtmIcEDUFO1xNHvkcRdWkjECzMbJ/cpuKFR0ct  
rYqdNiBz6Wp41+yFkczjQ7wzsik2nZek9aGRG1kMYjwHYNCZ2iKu04QU6pZt1I5HoakwcwjQVx  
HiB4VEST07U9lpGQzKMqgcR2YKSS1p/+LGxrx2V0dMRwyKRpgQYMW0jmOYonLZk4mAqMQmhvEWWx  
WCCL771SMXQACEcSIKGSrYMGox/s0q8E3GHbCfDRHlQgutizJn6Vuf0vj4eLhFGE1BDahfreszmc  
dy+nQ0A2aFI/+tT4fiBS5RyjtAdSIBg3IO01TtjJvAjMsQ4hUdJKlWF7G0H/AGkUt4vi5cnSoKlQ  
DAjwHmJ7/inG00qTIBiAw6jeR5GsviDFkh1BDSJUD3oQyGndjTgj4DEI91bY3BDde3SrwB3xGfC  
AZBYEkAmBH4rP5XCKOJJBXYciKtGaxQxc6lF4JmIvsNqPRqTae0ZJphqZLFwzodIsxFWbxbamuV4  
UdC6nvF9qG/T+XLicw8swNscBH0orDyWDiAYnxGXVeDNqWTUdjJIdlGE6XCAC5IEAHvO+53oNsUL  
8g1k7vP8+1NshwzWGBBcksBAMOp7VHM5QDZCItLAjnymuOM1thT9YlwsZEACDlue/wAxmiv9HJGt  
je4BP860bhKASBvfl+a9TDYwQbjsPvTSLwH2kiOFkkWFkie+1+v1TLHRVS5WIsW2360M4kXE9e9B  
4yQAumwkDt+1NGVoasA2aZFYqjXhMXHnMH+GistnfcFZDPVVMH81Ti61HzKoI6XMRaVQwLgAkTew  
Mz5UdoXWRoMbCCOhZQW3DjStE2/kKHGC2HqKRpcQf8e9ChmcHXpK8iYgdt95+9FpglgIteevf2NB  
2jbKy7qoAYet579TQOawFdGxBC3JAmbDYc4/emGJgmDOW5dvPfakvEXdfkZgNovY9ibiek00MSDT  
rJ6vDi6wmEwXlqt5eda4HAHL+GFjdieY351cm87FA2uVAg3ghLYfej8hw5MMs2IWPmCgDva96sm1  
6SuhZjopEkkONTny0dutMeFZFlSwNUSb7wPsSapTgZxmXEDkLJg7W3sOfSm+YcJh6EnwjfmY6mlc  
vjyXjB03L0zOfY69LAhtiCI8qXvhyw1brJA61oczxBXdf68z1j1XmVw10F3AMXuBysB9ftTdmtoi  
uOnaZmsTPh/CwiNiPzQb5Yk7j70xz+GriV+bew3pUUCda6IvGAsa5HFCL4oB5RMnrvtrQ4oxICQo  
g2H3796z7hj8u9EZXBxQQdv2gz6HahKKcmzYcNzBILncDpz2tTpVP8Ap2gWQAY9QRfONJOEYDBF  
UiO35PfetXk8raDbz+ornkiyVqjEHi7YjBTYTE1SSFA1MYHTa4obiGXV1Z9IDneLISNyB1NUcdyB  
XMvhhAgDQOQI5NO0Gd6hm+HosKc8ryZKqrkf+0QarlimqOSUaYG+EQhbWEEKdMTczzn50flfhKod  
11E/KoMgdyay4n6YV8PXhM2IAQNY27ggB351nHyj4eI2Fq0nVF9j0rKSmmkxw7GSZxcV1DMSS7AX  
XoADY1PEYjawYtspuNvzQGCzFjCDUSNthG9PcAsBDG29zIk7n2pZLroZJs0PDuHkLqZ1A0DkDJ70  
r4rw/AKNIi+1lIDrO02mKIyeWUafCdizX0kypINx2tep8S4OmLmCzmG03C2DR1IpFXbI0dgOHlyj  
IreIOsTy8QtHfa9LXcaSuspEXYT2I7itWmUw8bDR0lGQAJq5Rs09L8zwnDkM5Aknw6oDTuK3amNK  
SukJc3wWUXFV2ZtiFFvMU3yP6cZcDRiNOu5EyQIsI5GrcLMukqmCAq7AST7xApjg5osoZ10E2IJm  
Pas5S1YrTRB8IooRMIEBdMzy/egcLh5AA0taruKZrFRVbDQmpnxEmLb7VR/lodvesoyMoyNdlcPw  
AMjm0+AgMscwCLnnXuWzLsSuHmUxgN8LMLpcdpPOPKq2UgJQDq84H88qXZricmMxhDEtbX8uIt/7  
139ahwtOJWLwP0ySNOvBfBbmVYat5XMelQXI4JgKHHk3+KrymMhwloHiYjLyD7jtEVNX5HfnEmpX  
aq9I/CtJnr8FR18OI6nl035xE0JmeD4iRHjVbiN/YmTRKSLqRlvJ+1qv+O4iIA77nyAmIFF8cWhW  
jOfiJlGBvZhh1vahizG5UqOx+vWtVmMP4nhdQw2Oobc/MW70nzf6eGoNhqYPUnblpHxUBrAuOX1J  
pgsx3tYjpO5qWSyb4SlbaTfxMTznmbD/ADRGDwXMggRI5nwgDnl6UY3AMdrkr6kCfvPrQ6yqgIA  
x8e5B0mLHuI5XvSrj2XD5dinzddoI2pjnuH46C+GYHMR+Deg8zhttpYysmxJn+cq0YuLsDuhdwrM  
OERVdAWuWZtWntI2HSnGiYOIQWakgGVInc9eVJMDLrhrKkqzEhrdzYdKapilkAAltjJsR+KabzZL  
CyMMY5kAfLyAFvIUPjZHFxCUAVF2LMPsBcmg8DHZX0FjOk6BpIBYCDiabyJ9qIfN6kLM8qPmg7dj  
F6FPwMuSTsbm7xHIDaxUnGR4Ia0yDO0XM/SmWfhstilBFz9v3FE5fJo80iIqWYaJJ9Db1NeZKNDg  
lidQsYiR5AU7t1/A0ZdnVGU4bwtlTViWziFCnkOc96PweGoQQBc8yNv5FMjhaidWwuPS5+5ojL4A  
BHQ/f9ulVcm8lVBIW4XA1g22EfX9quy/CIMASbW9tzH2p+EjflP88uXej+E5eXBilATJ+vtSuTGp  
IVYeEYOfIECIH5pwxKhWPPke4t/zR3EcBCw/uUift9KJxcv/ANsCx0g26g3EepoUEzfGsor4HxNQ  
RgQrOs6o2A0kibedYHiuXy6CFbEZ43KAD63r6flVJTFXcalj1A5cqUBFxcPrj4LLIIGpZYR/UCNv  
Wl7dXboflTTs+ecM+O5GCuI6IAWADEAHfydTtF8AS+MmO64OZOpdcq5s2oAwRnvpIkeladK5ZMat  
8JS7RPjHiYWsvKlfd0xMzjuhTDwCpBgoVZgTtMwx503dSi3RHNmgfCyxchMJbWLKIHuN6GznDFZd  
YECfl6dqfYGS+Agd2DIgJMLEaOnOkuPxrDIbELAJcxPM7L5xU4JydjZSE44i2A40kkRpgLIHn0q8  
Zc4rkK6UEEKZ3vA6UFitImdZU6XMgDmK7M8UCK6QVIEgbGSKbN0thcqHWYz6YCBZJtYbnlpZmOKY  
0K6Jht4j4d2PQg8rdqWZBsJyVxXKuWfpuOldZnF+G4ZRqhiBuZAAAnzpowskyjeWaLG4piPhNoAR  
0uw3gDcVNcyThLjtAk6HXcG0yOludQ4M2HilcTCXQfEMRCLmQRI/uEnah3TGOAcPDwpRjN5BXY5R  
3o1FKtBVUaHgWdI5SSiKoQIPSOlqt4/DAWJCLf8AnSgeF42Pggh7sLAEgeH03501/wBcWvp37/4q  
TnOLpZEc5RwaBmaAp3Yct7cp7ikWdcH+krExtAFvpTvMYbaDqbUwMgx0pVm8ZgPlkX353/xXPDZd

Oti3B4iUYpqhJsbyDub9OvpcU6wc7YarT3t1nuYuKQ5hG3A7dx3+lAa3SL1lvY/iuuMvAqTWzf4O  
aDW5XJJ63/yavTNSJiDcgeQ5/esXwviJZWlvFJn16e5rQZbMSFIG9rf7ace7HmRxYQAm8+Ik99/U  
xR+C8je8C/0Ples/lSG08t/Sbfb70blczYGZlW5dNv53rGGaYoJLTikgwZII8JFu4NDZl2Q6gCV  
5sCbdPDMXrxGAU3Hjk9pN/vNUYgYqVvocNkz3kdwb270xgtM4n9bW5HxQZvvUcxg4Di4kDb94/x  
WexeIPhtp06iTZSYJkGNLGxPhO8fkmpmiQdLMjLurgkEESCH0zEHe3TcUKsFEOIcADj/ALZ0+dh9  
FP3pBj8FzSMNGKiRrggyOYA3mOdq1OC7wNUkEXKywB/3apBonExQFJ3i5kSR3iJ+9DqvRHCLM2yN  
qGsDQviIYXtf0g3pFg5XWWXVGoEBLXeb+LtWlGMMIDpVXBEEjSfTrUHYiYaeFFQDuAPcnekUKMo1  
hGay/B8REGH8QsJmFW45ke/oPWvcdSAEW+0xf3PpTHNviHwoFRd2LMR5yQpEdjVGDlX+Z0Jk2YEN  
af8Axmev7GqUUjFI9w8AaQ0SQfoeo6lcmUAYmDztvt5SKIXAhd5Aht1m3IGp40nfYHeeX8NAYrw  
cFmWVE29R6c6c8Mw10hogx7H9v50qPDsMB2InrH3HnNeYWMUc28MzHY9B6+9KzEcW2Ix3v8Az6De  
naqGTtSDNXckHew7jce23pTDhWOChEyr9qWGePlQmIzDZ0W3KVgD6VlflBxDM40JKkhFbxMQCrCZ  
AiJBgitfmcY68McmDfj+elZTjHEUTM4mFieLCcLq38BAABn2NKJPRHPcRR0DFVeD4NBiAd2BteoZ  
fDzyAUsOTG4n67UKuSK/I6sm4YLHoTsDSzW7FyHYGSCSTB8jSVK6Wjnbzk+gY2VjCzFbVKsAWPUG  
xr4yODYiFg0gobqdie09udbr9I59wr4eKxYKfASZMcfWjOOZmdMRAH2GJYMYBXSQJsaE+snEZ  
01ZmOI/qDWgRERFEKniuO8AQPEGcXKYOCnFSUvJMG/JpHLauz/DsVsRnbD02DeGNJ66TsfvU8bEx  
XCBHZtAI0EbCONOrUrwJl+HcMyLO2rSjBBDr8wHY8qJwkAcaHCGfACJlCImYIBFAMmIFUI6KQsXk  
kXPY7j96Z5XPMEYAqGAftzteLbUssKxm6RLCXHXFIJBDGdURHKKezZnHFlisBAdqaQ5Z3LkzXYA  
2J2t0rhiquITiMwY/KvIfvU3G3Yq1SLuK4a4phEb4nhJBYY6Dr6VDCyOMAP1/8A6qjOeMqFcnTz  
0/Sau0N/efcUvAf7Xs2oQhSCZBkm1/L+e9KM0gDGFk+vpTDNPZVCyAfOOV73oHO4lwR/UvL61y3a  
R0S8Fj7d5P70vzWXYryuOtHICT5+VdiIZg2AH8vVYyZlctmcGF8NtSss8+9/OnXCeKgtPne8cwf8  
0uz+XgzI78qUYuCAQUaGHQ/muqLUthvro+iYecCtvPhPkYuI+9MOHZNwQJtqj/2BH5+tYXI8UD6V  
azAAE8iRb6i0U9y2d0LC33F/ceWwj/NaqHUrNMmYuCNtvyDHvRGWxFZVc22Y/ePsfPwZTPBiAAeZ  
jzAMft7UamaI0sLg7wd95t60QjXN4SsQpvcEWHI29efnVbPrAV5tdWUkXG5nrcyDl9ao+NcAk2gq  
R5f8ewqeYxpWbfbWtbof52oWYkmZ0PBYDcLq1Tz8LG6nt9hTUPSQJ6qdxSZXV0F5EG/rsR15VfgZm  
FENEWvf0M38v8VrNRdmuGIX14cAn5oI8XUQbTee/3p+AFkaokzBSJ67SPpXJmgR000Tcj/8AQPL0  
rlseZUeIHZeDx/yD+Qajx10iUtztch/4nr712ER80BS24E6T3v8AzyqzDDaYkKd+X7/mp4WVvcwG  
i24Ppy89qNhIrgjZRbmN/frS0pp8Dxa3sPvH0rTJh6V21el/Lzisvxn4Y8559CBP1ifwLNY64e2l  
CZvFvIWH1kULjZkBXb5wPWNPs0ikud4iW/7abCCx8iCV+hFSRY7t5T63j6xWMhlmX3K8jPnBI+1E  
8GcFwRvcGhcJSQ02uSfi7/Ufer8q64ZU73/bnQYRhx FCCjAToafQiDWT/VuOBiWQNqRdZPl9dhW5  
x0lowG5FqxnGMPxI5IA0AMivKkrvz2pXS2S5fxMsnEDhQrBWw3EOkk6fI/mn+WwUOGLjQICkxb/5  
d/vVf+jwHA1rpnblvb0oTHypy7SjasIjxKtT/j7UXnRHsmqZTiYhR20IYk2UXItf6U6ymfQopIKu  
LQ3TvS/MYqKMPFw7qTDHpyv0NHYiAi4kGpOKu2I7i8ijjWiXyfcWFkFlmGd/AM15HuK5crgu2pcz  
oe4ExpI5im2DpGoETK6R0HTyobK8OYFw4RcP+kKLnq17VX+oooKm1ozXGUe42GusurjVHUGedefp  
5yMRVxAbRiW4I32N6licJwWZA226e21DcV4R8VlZ3ssRFiCDiVWXNGSpgrc2JuM5d8tj60Xuu5BH  
MXrs/gYmMUZMKABJlgCT060zz2K7hEK6gDzIkHrPSgXxEKt8NtDC51f/AJOxp1JAAtC9+GZSMCqkC  
RPOP3pj/AKgiwmltvmf60IuNjo6F3aNA4EQw9LOR/1TE5fas8jKkaw45IvHtVbpIHbbtUENdms0EWT  
zMC8CYm/tXiKUpSSTIx1KTqxccOGixI630luVVxrIiRCT06kb73oknXpYwCd4G9SwsFQJEK9zyfz  
tXpQtRzs74RxTAHwBHyme5BJ/alGPhAEmCb37+1aBsJpIkwLyR7AQKV5ggMVG/lyU8ZNCyaQozWX  
sYgCvclxKfEPJjy/g0Xi4JJE370JizcXIqsZrTJ9qeBzls0jQ6HlB/n19Kb4eaisIvFjsbfQ73rC  
qHVgV5dNjTDLcUKkhxYj2/5p6+FFP6bhGNo20348rmI7mrCZGoX6idj/AD61lsnxRYgGfWDWiTNK  
yl0IkjrZ8xtP3FKx07KXcp4hseXI/gVHDzIJMSwJgjnPl2otswpgOAQefMdj3sb0rz4QEFDPyddj  
Gx7HvWCaAkhVOnUpo/hyBpLDUOX5B5TtcVm+GcaLAIwk729Zjlg+tHYffIGbUQB19N6DYUzWYSLE  
Wjl3HKe/7d68dkAgACJt0rI5rjJ0Qranmw9In3Aqa559AOqCGkk7EdPX80GY0bZsQb269D/BWK4n  
nZZglzefaF/A9aJxOIBFbcFpsf52pFlFaZ7ye/8AcPKxoxQGw7h6i5Jkn6yQfua2GRwF0KIG0Soc  
3B9iKz2WyM36H67g/wCaf5TM+gv7T+LUQnmNhLYBF7g9wQCPqKhHJqdbHSSARz3/ABbzmiczjgkc  
5+xuKKy2CFIj8/59PrSmsJR9JjtfSRWH/XAZGUglQWMQJSDfsINbV2M7XJ37f8AFZr9a5ZXW0Zg  
CJK3N78rUnk51ldmK4RxVGLmUkAkaJs3cVPL0cUFImP6ojymvWCosQFHKqsDjAUHUhAG0byaZr+0  
5lnSDcLLNgGZ14REOnTv5UZgZAodeHilsI7Kbx2M7edVYXEUInVa72BocZ7/AExloVbDcwYt9u32  
pMjpt4YV/q4JDAKQJ338hvXZbPer/UVN5I/lqF4xoVsPFEI GnxbhOek/WrcDOiYwhnTApelkpRaw  
TXOY2o/KF5CL/mpkljLMD9qghJuaHzWfVBbxHoP35Vld4BF/EXARiX0k2n9qyeGuJrhcUgnaSYPQ  
dKs4hlshbXsQSSKAaFECo6kVa3AIJOgelzV4xjFbHSS2UZ3M46ko6iV2mL9/K1K8TOtJ3/wDan2Zw  
20DV8yCFuDPbrS34oN90/lVEl8H6o+ho80Nxo/8AbT/ePtFcArtXcRacu5sSpUwfmV4fEqmmc/F+  
RblmSB/VHTy26V5iYwX5FIHcd+gHaqMtnvAI0rYWAK7Gp4mNCKhmPoBXDR6CurBcVsZ9pCx0I95v

0ofHyzf10BI/uv6Ci3ckQR56mNBuZPKT0H5p7FkkDBOv5/gqgkMYHLeimTVteedDHDN6KJOJUydv  
80Ni4cn7CjmfFNgBftVGKk/imi2tgWNgv+m338+9SwcTET5HYdtxv6LJA6UWuVsb3pnI1ZsGTImK  
ImP39KIY2K7yd03IEW9DBiqMXAjbXuUzIwysXgG16N3oaTkkMcVjhPmVlflY3NCZrMMzEspvVzT  
EQ8RBWA07/WpZ9MNCMMSZmCsir5Tv+1LlMlcn6B5YoLglT0m/wDIn3pq+aCp4mAFjJ25c/Ss78SR  
KzHI+sfioI5cFZtVOtliYd0zuIZsuQqMYWRM7zWm4UnhBPODA+v871mMngDVpJjoTWu4DjKh0Pty  
O9F/EVivpoMniACDcD7UVjYawHUWZ+vQ+Yt61Q+XgSu3Lt/g0F8bdDYzz+xFIUCsJmnURESPI71d  
i8WAYKTO31G/a9Ca2UbEW26mLHzH1oN8qrkMx0hrx0gzA9zQMPMHYcWtex69ftST9bZgDLgshMG  
SVAEWsTPSf5amuUyyq5csNA+RdotH4rE/r/ieHiOgSToLSV5be+59qMVkSbwZfiOYbEuJIgDuBsN  
quxcynw7DSIhgbme/qKGfGZVBVoG9oBj1qvBzCsRjnnJ3v171XrgiUY+YEjTyv5U8yuKcTD0YkEA  
SDsQeVCtnMPQyQCTYurZ6f8AFD5PNKRGTZQSY8961X4GLH2RxiilcVQCJ7MpMx+1A5/KMmIWBlGk  
pFpHQ9xzqGJhnFUOrAKfmvyHemWQxgR8LEup+VuhqbaTwZSi3RRls4QdTjw2tN60eWwss4jSJPI2  
P+azuNlVAZCsQT/UZI5GeYiKafpjCUo2oE6PCNUbRM1LlS63YnIqC83kVQeH5SDHagy4kDKBTrHw  
lKkWA+1ZLiud+HiaEZXBZeNtp2JtS8LclXpNK9DJ0Q3dQQL36edVtgcPa979GaPtWexOIM4JItG  
w2pN/q3/ALo7Sa6owl9HUA9PouexHVzDWvEgftXiKXwcSTf5ffY+9dXV56iqM0juDYiFFX+qOm1W  
Y6iCASRM1ldVns6l+ILpjckBznVwxFOQoAHl+9dXUBSkN/bseleYiAlIhN0HeurqPoAM4J6yaqG  
Xg3/AJ+9dXUG8kZfkCMPxC1tqtRzME1ldWYDJMuV4rhkgqLncAXP8g1ldVOPY60F8IRkQhheQYMs  
BclNuIqdSlgRqkqRMdXV1VebEeHGVzVOMDoQAajZdgN4ovLZR0QlZJjny5VldTLsgj+RPAQMSOf  
KtHkcElZAmNx+a6upZHREdcOxivhNxtVmcyoN5rq61Y5RjNjW0w6g2IvbpFW5nIF4LE8iVA6d66u  
pWAo4rnlCaLyVKkzLLI286+e4GScsZeQu6+VdXU8SO91WJib5RG0fS178POog2HKBvXldVYgcUUY  
uEyWO02NDKh29+ldXVRaJvYxyGDilWVJKGJ6U6yzE4YBlXfu89a6uqHI9iPDGPDcc4gOG8fEQeAn  
+odJqrM8RdeKoShElhHinpXV1JS7DS8FOU42+tdR1BjBlywjyneneazHIKoogf0yZPXtbpXldr5F  
lqjOTS6laYaxBW9Vf9Lwead3Ne11TUMTP//Z - No connection adapters were found for  
'data:image/jpeg;base64,/9j/4AAQSkZJRgABAQAAQABAAD/2wCEAAoHCBYWFRRgWFhYZGBGz  
HB4CHBwcGh0fHxwhJR4hIR4eIx4hIS4lHiErHxwcJjgmKy8xNTU1HyU7QDs0Py40NTEBDawMEA8Q  
HxISHZQsJSs0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0  
NP/AABEIAKgBKwMBigACEQEDEQH/xAAaAAACAwEBAQAAAAAAAAAAAEQBQIDBgABB//EADoQAAIB  
AgQEBAUDAwMEAWEAAAECEQAhAwQSMQVBUWEicYGREzKhscEG0fBCUuEUcvEVI2KSgrLCov/EABkB  
AAMBAQEAAAAAAAAAAAECAwAEbf/EACURAAICAgMBAAIBBQAAAAAAAAAABAhEhmMQMSQVEycWET  
UoGRof/aAAwDAQACEQMRAD8Ae8WzCwNTWj+UDxE8t+VKvj1rogUdT4j+1QynDi7fEd9bkksTy7Ac  
q8ZGDs025DlXkykm7Z0J0EY+eYQDYHL6C38vVTYzSYJfjaedUvjgxC/iT39KLW8tsxFwIoN2PbBE  
zLlJcHpeDM87b7VNQRuk8ib9I+1EFew/WeVUjOqJUsQepHtfnWWQEjlyU1LNjtz/mlVPky4kWM8  
7Hr61bh5jEXYqydVMHfmNrUdnc2mpIHUEEWN7UaM0BY2MVUHSwQeFvDN7Rtt35UK+ECQQANO7A30  
29rbfWrV4+UYyngm+/v1mKtfGw3IdNtWogGJHMed7+1ZqgC4hRmkkkBgw9pIvluKqzeGGZAosqFF  
+hFT40oGKhRyELzbkGW4+htVPwzlgwGmCJ3YChJVLtfhCTqVheDgF3CxBWBSPL8VTlCiviIyGA1  
uo6Wop8cN/3Q1wbj70PmcyBmS4EgoDhekfZpv+P+jSblhsaPXiNgBkSchlm2Nt5HlSJ/1Bi4jhaQ  
WWRpAie59KZpmAyMiuRrQjEEFv6Z/qjLWxywCYivN1OiSPmEHST6Wqjgprt1kEk2kDYZkyRM/em  
uSSBtS3DxdGmfFM8r8opzhKzHUBHYRFmx4oozIBs0AEkXrOZ/MrclyGwim/6mxygWfltrKhY51n  
gGcDwsLuQNX5d6aEV+TJSxI03AMAYuEZPK1toql8IHhBNhoU7+c1f8Ap/ShIR5W29mHYiqMvhs2  
exRAA0j26ig45bC4qrLP09ksMI7yV0FtQJ5copzkmbQHw2hWBs3T/NDcS4ew0vhAsWIDiwmRHtS5  
+MIh+EpKRFjc6toHrU5cb5GmwOKbyPs3mXVQWCGByYG3kaTNNeAQy61NoAn3obH4gqQ2OPiKWIMW  
bTNzPwr0CJiQPEjjUkHZSLAnrSrioRoZcJxMJ00yAGAIuPTpUOKoiEamQOegEQKry2Prlii2IMED  
87V4ndQoOoglqQTsfIVJxUXTW/8ALE/aBMTMa0IF4043os5tv6GgZtgxB1AedCYWKIME+HC/wB0  
iP8AmrscOHDq+pCI0jlVF2ukq/YylbopGAWQkAfc0HhA6wCZU+Hym1MsvnSbQCwPMkW/NFZ+QBim  
hIJHjC2n02p7lF1Q2mIlJYCdUCZgGqYCbqDa8c+8ldneLMuopeBiBbeicrxFHWX8L89Ip3KXqWFX  
f8CZwjGRMM4HQf8ANW5HPuhKnxKeQ2j969zGVUEjDe4kjUCIHSh+F5MrqVjEAKHeem1VfVoDQ6yn  
HMFV0hu0j1G/WjjnsJrnBSSBptSfL8NcrqVWPpRDo4t8M2A/p7VF8MLuhkbv8NQYEA2Jjy5n1+lK  
MRlawkCTAI3j6xzphlss7FZaSBfkDymrMPKqol9M9du/WlIo2780lCwKFKkggnsfr0q/MYrESCny  
LzsPLnvRqsjnw1SfSTBJ9Cft2rs1gKotAAvP+aMk0hmKHJVpZ2iII3EmfzRSijL4T4ovBBivdBIJ  
KkdjvHI2PQldw/KrqkOyneAZn3BtSJ269JXmgfDaRGwPW0XjaoKniM2iInr3/wAU/GS06CAGgcxI  
8496tbJKs+ERF5/c7VSMJJGutmYzOCGLWgmXkHtfbyqnLIVSCIA7biefajsxkheBBELSCfiekDyq  
CKFUodxzP0azfglCXOup2MFSO/ODHlJt2oHAYrgTqliLMDY/zpRmdyelixWwuOlXNALXwkJTE0M9  
9ESRa+c0/LEZ6phuQwjoYMYLXtE+1JXzmIjgPeKEXvaaY5HQ4K6tD/2nwnzDXH0r08NXWA5uLwli  
e4b5T70Vi0x1bjRHL4mO+MzhmeTL6RYWibbVHDZkzC4eIhIceekzf1/O9PMvmsTCQIihJ0lh/Ux8

opVxAurKXUg7GeR6fY1lJvFCJNYZRl0h+wJ9ga0GVQWidqWuwYpG5v78veaa5VTMaYtSS0UiqFX6  
hQlB/urP4R0kEiVnJPSeXetP+ppXDUgE+OIrPPlcViCMNgp5tb6UYJlZGcX2wHnATDx4w8RWEcX3  
vynatHiz8ItYMN7eKORnpWSy/BcySSMGR/vW/TnWnww4wmXGVlIXwnl0+YWPvTyxkpG0soDfOIQh  
DeJhZQfee9VthYcsCFDsLM4FiOh60LxnF0NoaAdAKuqjxEi0Wv50u4Rj6w4xCToBIJ/qjlSKLeUS  
k7eDs/1lLqhZWIjYzfnTHAwlBtACz/VM9qC4UinFd3QMq3VSevSnHwEnVGm0Wv6zzN6MrWDKLSdy  
WcxMJGRVlAmeldmWfUhbY2tt3HnarcfN4aqUQlr3JG3WDzqzA44j4L4WgO4ICtHLMsa107YKBzm1  
cqjqNIJA0gAt6+VE6zAPw9KiNJlEmxtMV6ugYRDAGTIt8vkaGTPsgRdKECRPPeRPI0kk9oVprKCM  
liuzr4Anpf309aX4mjBdSJGg25G1Jm4ijKEZQW97/tRuW4iCpVlEE47Dt2NBckovKGSx2aybl2A  
UgWIIbVInYede80loxwytyNiset63WDxKEZMJQmlZlnSWPeDWLbiau5Lu70TdoAqim5LQ3oJnMwA  
09DE9f4K7CXQRFSZ738jROYyKOPGGx1TIVrEnnBNB4+X0gLctZTHlIHnePSnTTRmeZbHdiGLMDaj  
Udx2q70Ybl2Idt/7zUslgsphkIuACR9KvzTy7RAv0rN3oyZulxHVG0nxR02vYR5UImYYrLGSNyQP  
XtalEr4gtjBBPQdL++1B/C0nSviIkwe87Ciiubs9HSnk5XYmLGatwxAg3Emx9KrXDNrRtv3qKrrg  
GCJ7isngbwLfMzMAXsd/KatyJAYMeR85HX60pORDHCHtttli/PrRuVymnxXkjaTHsfvR9BV50K42  
rpv528uVe5vNBEZmBIUTa9Z3MiFJkqbXULW9IvVeHitiAqcUsOhJ36d/8VTu6ElEsHEEXAWVAhG9  
r32He1U4uPqElbXuLz7XHtVPw3R+QEwOm3UfmmGF13ZYAt350mXs0dZ04q6v1GIgEC/pt67e5rIY  
UOqO6wwEAjmbIFvStLmcNWBQggqPaOQPtWawsDWvwbySAdhc8/TbvVVMiko4Z2KmE6MWlmtAn72  
BkbVRw3iLoQrr8VB6sojcVZgcPLnSu/OT7lHHwGwnhhz3mXPt6WSEXXK8Gi4YqOgbCxs4U2BjUki  
I8q7OZHWRgGgaLm4kfbbSrPZDQ+rCbQ4GoMDY9iJvTbh+aZ0Bez7MBtPP339adfSkrWRZgIyEBhB  
WR6Hp9aZcPe9z97VfiAEUSbFYnSkhmNgPP8AnlodW2Pxysa5jiSoLKQCd9z6dN6QZria06lgbm88  
o7b38q0WQ/Th0+PeOd496pzP6ZOSWXT2ERbzq0Ulos38BOHZoN8jaT/aDv8AiYvWgyWYAov0n05G  
hst+ldJVxZhazHb7Uxfhj4aaJ4wLxzHrWaQUxXx7hKYiFlQsw+QC2kze3TttWOPCHwi5chBEXPqb  
VvXzUAPcXAkex9I7ljp1flFDq+tn1hnAkQCY5ch/mppNPGiU4p00J2zUTA3PvTnhucUnRF+U7Gs  
3gDYsbglOSbyd5FGUVJUTu0X53GC08rJDHY3qvK57FxW04aBwKRBAP1oPMkkkkMxJmW/PWi+GIDq  
uAwg9Pan6pRyTGeNmbAMdDXmIM/7l5eleZdGY+FlYcmm3r0qnMYLoEdklLJLTZjy896Ez2XcenFC  
CJ8LEHreTSKKaoZPBoP+n4oJZAKMoKktewg+lTR2KMMQXMD/AGmbGRyilmW0IyIysugDMjll8qLB  
bdVwyypELGwPY8xvUnBgx4FPhoZh3Uil1tJB5RB3oDPpJGqNa2B06S3+4fml/F8fSWAEAX7jYiKM4  
dx0lAuMq4qnaFm7HnTw4nFXyVfpg7EzBMfHl0r3L8QfWCN43N7jb6VdxjLqro+G4fDcSs/Mu0q  
fKgWPjILAWZA+tu62sjPiFluLYr4mgsCJ6CptVkoBhQZ59aVZXE17bwTPpRGfmcUADblpXBLSQMf  
DcPmQXII1AN8sXFzB+29VY2OxJM6B0Av0N4k3586Ey4YPKQSZGWJiNxyC79bv2IXBJdlm+51n0Ag  
A22rksf06cvSS4TESSTBBBJNWrlD2VSbirf89xQeJiAz/wB15ttCgbchBofGwhZtbA9S7c/M/SqL  
KodNhrZVgxa5n+kC32tVyGLkFSJtPnVGRd4+cm8Ac+x3qzNhwssAwF5te9xH1pX2s1tFuNhKfFLS  
t5Bb7fzevfitpOkzB5ib+1KmYBmtaJXl5HeNqc4WKCoYCQYJI5c4PpRdmuy3CxnZTqAsREH8elMs  
LOYyQANJEgi03vs09KczjhAMQoWG8C1iofe83qODjo5+INSG40svtYWN+/KnSdWa3de84SsxXtvY  
7be85tWfUupMxZgZtO88yBT/AASBYc3sepLeZ2rMcQLBrSrv2/lqMG7M3R7mcRx41lIbXGmoZvE  
JVA4brqteeopNh54lyGLGbagbD6bVzhZxT4GktyP7lVQdEUmOERAmqSvc/4JNEcELKTJDI1tSmQD  
yPUdD6UixcrNy7SbBZNqb8IxlGGNII0iCetDqvASwmaDEgG/8FK04kmWdsw6lgDoUDrJ+wFC5ziL  
C3K3sar/qjNyMNBsWL/QafSnjG3QeN1Z9N4F+rSHMtoGpX3KsOXUHbnWhfMp08xXyHgKf6ZHxZBc  
rAg7C0jzr0/rV5skm/n53MbVRRXhW/p9D/U/6o/0qAjCL15C7gAxzP4rC4H6/wAyGDsUfDPzppFh  
MRtPlo3LcaXhW2XF06WvDbClWd/TuEUJw2Yzex1Afn3rXH01Pw3W0uG2CGwzOG6TE7Wnff8A4rPt  
iu2WfQssrkBjpECec7i5pP8ApjN4uGwW+UjSO3Qg+VP+FYJGUOHYU6sNue4Mc46VCSpBlmJlWwt  
IPxFGqCdSxHrFqVtmIcEDUF01xNHvkcRdWkjECzMbj/cpuKFR0ctrYqdNiBz6Wp41+yFkczj7wz  
sik2nzek9aGRG1kMYYjwHYNcCZ2iKu04QU6pZt1I5HoakwcwjQVxHiB4VEST07U9lpGQzKMqqcR2  
YKSSlp/+LGxrx2V0dMRwyKRpgQYmW0jmOYonLZk4mAqMQmhvEWWxWCCL771SMXQACEcSIKgsRyMG  
oX/s0q8E3GHbCfDRhlQgutiZJn6Vuf0vj4eLhFGelBDahfreszmcDy+nQ0A2aFI/+tT4fiBS5Ryj  
tAdSIBg3IO01TtjJvAjMsQ4hUdJKlWF7G0H/AGkUt4vi5cnSoKlQDAjwHmJ7/inGO0qTIBiAw6je  
R5GsviDFkh1BDSJUD3oQyGndjTgj4DEI91bY3BDde3SrjwB3xGfCAZBYEkAmBH4rP5XCKOJJBXYc  
iKtGaxQxc6lF4JmIvsNqPRqTae0ZJphqZLFwzodIsxFwbxbamuV4UdC6nvF9qG/T+XLicw8swNsc  
bH0orDyWDiAYnxGXVeDNqWTUdjJid1GE6XCAC5IEAHvO+53oNsUL8g1k7vP8+1NshwzWGBBcksBA  
mOp7VHM5QDZCItLAjnymuOM1thT9YlwsZEACD1ue/wAxmiv9HJGtje4BP860bhKASBvfl+a9TDYw  
QbjsPvTslWh2kiOfkkWfkie+1+v1TLHRVS5WIsW2360M4kXE9e9B4yQAumwkDt+1NGVoasA2aZFY  
qjXHMxHnMH+GistnfcFZDPVVMH81Ti61HzKoI6XMRAvQwLgAkTewMz5UdoXWRoMbCCOhzQW3DjST  
e2/kKHGC2HqKRpcQf8e9ChmchXpK8iYgdt95+9FpglgIteevf2NB2jbKy7qoAYet579TQOawFdGx  
BC3JambDYc4/emGJgmDOW5dvPfakvEXdfkZgNovY9ibiek00MsDTrJ6vDi6wmEwXlqt5edA4HAHL

+GFjdieY351cM87FA2uVAg3gHlyfej8hw5MMs2IWPMcgDva96sm16SuhZjopEkkONtNy0dutMeFZ  
FlSwnUSB7wPsSapTgZxmXEDkLJg7W3sOfSm+YcJh6EnwjfmY6mlcvjyXjB03L0zOfY69LAhtiCI8  
qXvhyw1brJA6loczxBXdf68z1j1XmVw10F3AMXuBysB9ftTdmtoiuOnaZmsTPh/CwiNiPzQb5Yk7  
j70xz+GriV+bew3pUUCda6IvGAsa5HFCL4oB5RMnrvtRQ4oxICQog2H3796z7hj8u9EZXBxAQQdv  
2gz6HahKKcmzYcNzBILncDpz2tTpVP8Ap2gwQAY9QRf0NJOEYDBFUi035PfetXk8raDbz+ornkiy  
VqjEHi7YjBtYTE1SSFA1MYHta4obiGXV1Z9IDneLISNyB1NUcdyBXMvhhAgDQOQI5NO0Gd6hm+Ho  
sKc8ryZKqrkf+0QarlimqOSUaYG+EQhbWEkdMTczzn50flfhKod11E/KoMgdyA4n6YV8PXhM2I  
AQNY27gg351nHyj4eI2Fq0nVF9j0rKSmmkxW7GSZxcV1DMSS7AXXoADY1PEYjawYtspuNvzQGCz  
FjCDUSNthG9PcAsBDG29zIk7n2pZLroZJs0PDuHkLqZ1A0DkDJ70r4rw/AKNiI+llIDr002mKIye  
WUAFcdizX0kypINx2tep8S4OmLmCzmG03C2DR1IpFXbI0dgOHlyjIreIOsTy8QtHfa9LXcaSusPE  
XYT2I7itWmUw8bDR0lGQAJq5Rs09L8zwnDkM5Aknw6oDTuK3amNKSukJc3wWUXFV2ZtiFFvMU3yP  
6czcDRiNOu5EyQIsI5GrcLMukqmCAq7AST7xApjg5osoZ10E2IJmPas5S1YrTRB8IooRMIEBdMzy  
/egcLh5AA0taruKZrFRVbDQMpnxEmLb7VR/1odvesoyMoyNdlcPwAMjm0+AgMscwCLnnXUwZLsSu  
HmUxgN8LMLpcdpPOPKq2UgjQDq84H88qXZricmMxhDEtbX8uIt/7139ahwtOJWLWp0ySNObfBbm  
VYaT5XMelQXI4JgKHHk3+KrymMhwLOHiYjLyD7jtEVNX5HfnEmPXAq9I/CtJnr8FR18OI6n1035x  
E0JmeD4iRHjVbiN/YmTRKSLqRlvJ+1qv+04iIA77nyAmIFF8cWhWjOFiJlGBvZhH1vahizG5UqOx  
+vWtVmMP4nhdQw2Oobc/MW70nzf6eGoNhqYPUnblPhxUBrAuOX1Jpgsx3tYjp05qWSyb4S1baTfx  
MTznmbD/ADRGDwXMggRI5nwgDn1n6UY3AMdrkr6kCfvPrQ6yqgIAx8e5B0mLHuI5XvSrj2XD5din  
zddoI2pJnuH46C+GYHMR+Deg8zhhttpYysmxJn+cq0YuLsDuhdwrMOERVdAWuWZtWntI2HSnGiYOI  
QWAKgGVInc9eVJMDLrhrKkqzEhrdzYdKapilkaAltjJSr+KabzZLCyMMY5kaFLyAFvIUjZHFxCU  
AVF2LMPsBcmg8DHZXFjOk6BpIBYCdIabyJ9qIfN6kLM8qPmg7djf6FPwMuSTsbm7xHIDAXUnGR4  
Ia0yDO0XM/SmWfhstilBFz9v3FE5fJo8OiIqWYaJJ9Db1NeZKNDglidQsYiR5AU7t1/A0ZdnVGU4  
bwtlTViWZiFCnkOc96PweGoQQBc8yNv5FMjhaidWwuPS5+5ojL4ABHQ/f9u1Vcm8lVBIW4XA1g22  
EfX9quy/CIMASbW9tzH2p+EjflP88uXej+E5eXBI1ATJ+vtSuTGpIVYeYOFIECIH5pwxKhWPPke  
4t/zR3EcBCw/uUift9KJxcv/ANsCx0g26g3EepoUEzfGsor4HxNQRgQrOs6o2A0kibedYHiuXy6C  
FbEZ43KAD63r6flVJTFXcalj1A5cqUBFXcPRj4LLIIGpZYR/UCNwVl7dXb0flTTs+ecM+O5GCUi6  
IAWADEAHfydTtF8AS+MmO64OZOpdcq5s2oAwRNvpIkelaDK5ZMAT8JS7RPjHiYWsvKlfd0xMzjuh  
TDwCpBgoVZgTtMwx503dSi3RHNmgfCyxchMJbWLKIHuN6GznDFZdYECfl6dqfYGS+Agd2DIgJML  
AOnOkupxrDIBELAJcxPM7L5xU4JyDjZSE44i2A40kkRpgLIHn0q8Zc4rkk6UEEKZ3vA6UFitImDz  
U6XMgDmK7M8UCK6QVIEgbGSKbN0thcQHWYz6YCBZJtYbnlpZmOKY0K6Jht4j4d2PQg8rdqWZBsJy  
VxXKuWfpuOldZNF+G4ZRqhiBuZAAAnzpowskyjeWaLG4piPhNoAR0uw3gDcVNcyThLjtAk6HXCg0  
yOludQ4M2HilcTCXQfEMRCLmQRI/uEnah3TGOAcPDwpRjN5BXy5R3o1FKtBVUaHgwDI5SSIKoQIP  
SOlqt4/DAWJCLf8AnSgeF42Pggh7sLAEgeH03501/wBcWvp37/4qTnOLpZEc5RwaBmaAp3Yct7cp  
7ikWdcH+krExtAFvpTvMYbaDqbUwMgx0pVm8ZgPlkX353/xXPDZdOtI3B4iUYpqhJsbyDub9Ovpc  
U6wc7YarT3t1nuYuKQ5hG3A7dx3+1Aa3SL1lvY/uuMvAqTWzf4OaDW5XJJ63/yavTNSJiDcgeQ5  
/esXwviJZWlvFJn16e5rQZbMSFIG9rf7ace7HmRxYQAm8+Ik99/UxR+C8je8C/0Ples/lSG08t/  
Sbfb70blczYGZlW5dNv53rGGaYoJLTikgwZII8JFu4NDZl2Q6gCV5sCbdPDMXrxGAU3Hjk9pN/vN  
UYgYqVVocNKzz3kdwB270xgtM4n9bW5HxQZvvUcxg4Di4kDb94/xWexeIPhtp06iTZSYJkGNLGxP  
h08fkmpmiQdLMjLurgKEESCH0zEHE3TcUKsFEOIcADj/ALZ0+dh9FP3pBj8FzSMNGkiRrggyOYA3  
mOdq1OC7wNUKEXKyWb/3apBonExQFJ3i5kSR3iJ+9DqvRHCLM2yNqGsDQviIYXtf0g3pFg5XWWXV  
GoEB1Xeb+Ltw1GMmIDpVXBEEjsfTrUHyiYaeFFQDuAPcnekUKMo1hGay/B8REGH8QsJmFW45ke/o  
PWvcdSAEW+0xf3PpTHNviHwoFRd2LMR5yQpEdjVGD1X+Z0Jk2YENaf8Axmev7GqUUjFI9w8AaQ0S  
Qfoeo61cmUAYmDztvt5SKIXAhd5Aht1m3IGp4OnfyHeeX8NAYrwcFmWVE29R6c6c8Mw10hogx7H  
9v50qPDsMB2InrH3HnNeYWMUC28mZHY9B6+9KzEcW2Ix3v8Az6DenaqGTtSDNXckHew7jce23pTD  
hWOChEyR9qwGePlQmIzDZ0W3KVgD6Vl1f1BxDM4OJkKhFbXMQCrCZAIJBgitfmcY68McmDfj+elZT  
jHEUTM4mFieLCcLq38BAABn2NKJPRHPcRR0DFVeD4NBIA2BteoZfDZYAuSOTG4n67UKuSK/I6sm  
4YLHoTsDSzW7FyHYGSCSTB8jSVK6Wjnbzk+gY2VjCZFBvKsAWPUGxr4yODYiFg0gobqdie09udbr  
9I59wr4eKxYKfASZMcwfWjOOZzmdMRAhZGJYMYBXSQJsaaE+snEZ01ZmOI/qDwGRERFEKniu08AQ  
PegcXKYOCnFSUvJMG/JpHLauz/DsVsRnbD02DeGNJ66TsfvU8bExxCBHZtAI0EbCOnOrUrwJ1+Hc  
MyLO2rSjBBdR8wHY8qJwkAcaHCGfACJ1CIImYIBFAMmIFUI6KQsXkkXPY7jp96Z5XPMEYAqGAftzt  
eLbUssKxm6RLCXHXFIJBDGdURHKKEzZnHfliSbADqaQ5Z3LKzXYA2J2t0rhiquITiMwY/KvIfvU3  
G3Yq1SLuK4a4phEb4nhJBYY6Dr6VDCyOMAP1/8A6qjOemqFctZ0/Sau0N/efcUvAf7Xs2oQhSC  
ZBkm1/L+e9KM0gDGFk+vpTDNPZVCyAfOOV73oHO4lwR/UvL61y3aR0S8Fj7d5P70vzWXYryuOtHI



CT5+VdiIZg2AH8vVYyZlctmcGF8NtSss8+9/OnXCeKgtPne8cwf80uz+XgzI78qUYuCAQUaGHQ/m  
uqLUthvro+iYecCtvPhPkYuI+9MOHZnwQJtqj/2BH5+tyXI8UD6VazAAE8iRb6i0U9y2d0LC33F/  
ceWwj/NaqHURnMmYuCNtvyDHvRGWxFZVC22Y/ePSfPwZTPBiAAezjzAMft7UamaIOsLg7wd95t60  
QjXN4SsQpvcEWHI29efnVbPrAV5tdWUkXG5nrcyD19ao+NcAk2gqR5f8ewqeYxpWbfbwTbof52oWY  
kmZ0PBYDcLqLTz8LG6nt9hTUPsQJ6qdxSZXV0F5EG/rsR15VfgZmFENEWvf0M38v8VrNRdmuGIX1  
4cAn5oI8XUQbTee/3p+AFkaokzBSJ67SPpXJmgR000Tcj/8AQPL0rlseZUeIHZEdx/yD+Qajx10i  
Utztch/4nr712ER80BS24E6T3v8AzyqzDDaYkKd+X7/mp4WVvcwGi24Ppy89qNhIrgjZRbmN/frS  
0pp8Dxa3sPvH0rTJh6V21el/LzisvxN4Y8559CBP1ifWlNY64e2lCZvFvIWH1kULjZkBX5wPWNP  
s0ikud4iW/7abCCx8iCV+hFSRy7t5T63j6xWMhlmX3K8jPnBI+1E8GcFwRvcGhcJSQ02uSfI7/Uf  
er8q64ZU73/bnQYRhxFCCjAToafQiDWT/Vu0BiWQNqRdZPl9dhW5x0lowG5FqxnGMPxI5IA0AMiv  
Kkrvz2pXS2S5fxMsnEDhQrBWw3EOkk6fI/mn+WwUOGLjQICKbx/5d/vVf+jwHA1rpnblvb0oTHyp  
y7SjasIjxKtT/j7UXnRHsmqZTiYhR20IYk2UXItf6U6ymfQopIKuLQ3TvS/MYqKMPFw7qTDHpyv0  
NHYiAi4kGpOKu2I7i8ijjWixYfCWfKfLMgD/AM15HuK5crgu2pczoe4ExpI5im2DpGoETK6R0HTy  
obK8OYFw4RcP+kKLnq17VX+oooKml0zXGuE42GusurjVHUGedefp5yMRVxABrIW4I32N6licJwwZ  
A226e21DcV4R8VlZ3ssRFiCDivWXNGSpgr2JuM5d8tj60Xuu5BHMxrs/gYmMUZMKABJlgCT060z  
z2K7hEK6gDzIkHrPSgXXEKt8NtDC51f/AJOxp1JAAtC9+GZsMCqkCRPOP3pj/AKgiwmltVf60IuNj  
o6F3aN4EQw9L0R/1TE5fas8jKkaw45IvHtVbpIHbttUENDms0EWTzMC8CYm/tXiKUpSSTIxLKTqx  
ccOGixI630luVVxrIiRCT06kb73oknXpYwCd4G9SwsFQJek9zYfztXpQtRzs74RxTAHwBHyme5BJ  
/alGPhAEmCb37+1aBsJpIkWLyR7AQKV5ggMVG/lyU8ZNCyaQozWXsYgCvclxKfEPJjY/g0Xi4JJE  
370JizcXIqsZrTJ9qeBzls0jQ6HlB/n19Kb4eaisIvFjsbfQ73rCqHVgV5dNjTDLcUKkhxYj2/5p  
6+FFP6bhGNo20348rmI7mrCZGoX6idj/AD61lsnxRYgGfWDWiTNKy10IkjrZ8xtP3FKx07KXcp4h  
seXI/gVHDZIJMSwJgjnPl2otswpgOAQefMdj3sb0rz4QEFDPyddjGx7HvWCaAkhVOnUpo/hyBpLD  
UOX5B5TtcVm+GcaLAIwk729Zjlg+thYfFiGbUQB19N6DYUzWYSLEWj13HKe/7d68dkAgACJt0rI5  
rjJ0Qranmw9In3Aqa559AOqCGkk7EdPX80GY0bZsQb269D/BWK4nnZZglzeFaF/A9aJxOIBFbcFp  
sf52pFlFaZ7ye/8AcPKxoxQGw7h6i5Jkn6yQfua2GRwF0KIG0Soc3B9iKz2Wym36H67g/wCaf5TM  
+gv7T+LUQnmNhlyBF7g9wQCPqKhhJqdbHSSARz3/ABbzmiczjgkc5+xuKKy2CFIJ8/59PrSmsJR9  
JjtfSRWH/XAZGUGlQWMQQJsDfsINbV2M7XJ37f8AFZr9a5ZXw0ZgCJk3N78rUnK51ldmK4RxVGLm  
UkAkajs3cVPLocUFImP6ojymvWCosQFHKqsDjAUHUhAG0byaZr+05lnSDcLLNgGZ14REOnTv5UZg  
ZAodeHilsI7Kbx2M7edVYXEUInVA72BocZ7/AEx1oVbDcwyT9u32pMjpt4YV/q4JDAKQJ338hvXZ  
bPER/UVN5I/lqF4xoVsPFELGnxbhOek/WrcDOiywhnTApelkpRawTXOY2o/KF5CL/mpkljLMD9qg  
hJuaHzWfVBbxHoP35Vld4BF/EXARiX0k2n9qyeGuJrhcUgnaSYPQdKs4hlshXQSSKAAFECo6kVa  
3AIJOqelzV4xjFbHSS2UZ3M46ko6iv2mL9/KlK8T0tJ3/wDan2Zw20DV8yCFuDPbrS34oN90/lVE  
l8H6o+ho80Nxo/8AbT/ePtFcArTxcRacu5sSpUwFMV4fEqmmc/F+RblmSB/VHTy26V5iYwX5FIHc  
d+gHaqMtnvAI0ryWak7Gp4mNCkhmPoBXDR6CurBcVsZ9pCx0I95v0ofHyzf10BI/uv6Ci3ckQR56  
mNBuZPKT0H5p7FkkDBOv5/gggkMYHLeimTVteedDHDN6KJOJUydv80Ni4cn7CjmfFNgBftVGKk/i  
mi2tgWNgv+m338+9SwcTET5HYdtXV6LJA6UWuVsb3pnI1ZsGTimKImp39KIY2K7ydO3IEW9DBiqM  
XajvXuZiWysXgG16N3oaTkkMcvjhPmVlflY3NCZrMMzEspvMzTEQ8RBWA07/WpZ9MNCmMSZmCs  
iR5Tv+1LlMlcn6B5YoLglT0m/wDIn3pq+aCp4mAFjJ25c/Ss78SRKzHI+sfioI5cFZtV0tliYd0z  
uIZsuQqMYWRM7zWm4UnhBPODA+v871mMngDVPjjoTWu4DjKh0PtyO9F/EVivpoMniACDcd7UVjYa  
wHUWz+vQ+Yt61Q+XgSu3Lt/g0F8bdDYzz+xFIUCsJmnURESPI7ldi8WAYKTO31G/a9Ca2UbeW26m  
LHzH1oN8qrkMx0hrx0gzA9zQMPMHYiCwtex69ftST9bZgDLgsHMGsVAEWsTPSf5amuUyyq5csNA+  
RdotH4rE/r/ieHiOgStolSV5be+59qMVkSbwZfiOYbEuJIgDuBsNquxcynw7DSihgme/qKGfGZV  
BVoG9oBj1qvBzCsRJnnJ3v171XrgiUY+YEjTyv5U8yuKcTD0YkEASDsQeVctnMPQyQCTYurZ6f8A  
FD5PNKrGTZQSY8961X4GLH2RxiilcVQcJ7MpMx+1A5/KMmIWBlGkpFpHQ9xzqGJhnFUOrAKfmvyH  
emWQxgR8LEup+VuhqbaTwZSi3RRls4QdTjw2tN60eWwss4jSJPI2P+azuNlVAZCsQT/UZI5GeYiK  
afpJCUo2oE6PCNUbRM1LLS63YnIqC83kVQeH5SDHagy4kDkBTTrHw1KkWA+1ZLiud+HiaeZXBZeN  
tp2JtS8LclXpNK9DJ0Q3dQQL36edVtgcPa979GaPtWexOIM4JItGw2pN/q3/ALo7Sa6owl9HUa9P  
ouexHVzDWvEgftXiKXwcSTf5ffY+9dXV56iqM0juDYiFFX+qOm1WY6iCASRM1ldVns61+ILpjckB  
znnVwxf0QoAhl+9dXUBSkN/bseleYiAlIHn0HeurqPoAM4J6yaqGXg3/AJ+9dXUG8kZfkcMPxc1t  
qtRzME1ldWYdJMUv4rhkgqLncAXP8g1ldVOPY60F8IRkQhheQYmSBc1NuIqdSlgRqkqRMdDXV1Ve  
bEeHgVZvOMDOQAajZdgN4ovLZR0QlZJJny5VldTLsgj+RPAQMSOfKtHkclZAmNx+a6upZHREdcO  
xivhNxtVmcyon5rq61Y5RjNjW0w6g2IvbpFW5nIF4LE8iVA6d66upWAO4rnlCaLyVKkzLLI286+e  
4GScsZeQu6+VdXU8SO91WJIB5RG0fS178POog2HKBvXldVYgcUUyUEyWO02NDKh29+ldXVRaJvYx



yGDilWVJKGJ6U6yzE4YBlXFu89a6uqHI9iPDGPDcc4gOG8fEQeAn+odJqrM8RdEKoShElhHinpXV  
1JS7DS8FOU42+tdRlBjBlywjyneneazHIKooGf0yZPxtbPxlDR5F1qjOTS6laYaxBW9Vf9Lwead3  
Ne11TUmtP//Z'

ERROR - No se pudo almacenar data:image/jpeg;base64,/9j/4AAQSkZJRgABAQAAQAB  
AAD/2wCEAAoHCBYWFRGWFhYZGBgZHB4cHBWcGh0fHxwhJR4hIR4eIx4hIS4lHiErHxwcJjgmKy8x  
NTU1HyU7QDs0Py40NTEBDawMEA8QHxISHzQsJSs0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0  
NDQ0NDQ0NDQ0NDQ0NDQ0NDQ0NP/AABEIAKgBKwMBIgACEQEDEQH/xAABAAACAwEBAQAAAAAA  
AAAAAAAEbQIDBgABB//EADoQAAIBAgQEBAUDAwMEAWEAAAECEQAhAwQSMQVBUWEicYGREzKhscEG  
0fBCUuEUcvEVI2KSgrLCov/EABkBAAMBAQEAAAAAAAAAAAAAAAAEACAwAEbf/EACURAAICAgMBAAIB  
BQAAAAAAAAABAHEhMQMSQVEycWETUoGRof/aAAwDAQACEQMRAD8Ae8WzCwNtWJ+UDxE8t+VKvjlr  
ogUdT4j+1QynDi7fEd9bkksTy7Acq8ZGDS025DlXkykm7Z0J0EY+eYQDyHL6C38vVTYzSYJfjaed  
UvjxqC/iT39KLw8tsxFwIoN2PbBEzLlJChpeDM87b7VNQRuk8ib9I+1EFew/WeVUjOqJUSQepHtf  
bnWWQEjlyU1LNjtZ/m1VPky4kWM87Hr6lhb5jEXYqydVMHfmNrUdnc2mpIHUEEWN7UaM0BY2MVUH  
SwQeFvDN7Rtt35UK+ECQQANO7A3O29rbfWrv4+UYyngm+/v1mKtfGw3IdNtWogGJHMEd7+1ZggC4  
hRmkkkBgw9pIvluKqzeGGzAosqFF+hFT40oGKhRyELzbkGW4+htVPwzlwGmGmCJ3YChJVLtfhCTq  
VheDgF3CxHwBSPL8VTlCiviIyGAluo6Wop8cN/3Q1wbj70PmcyBmS4EgoDHEkfzpv+P+jSblHsaP  
XiNgBkSchlm2Nt5HlSJ/1Bi4jhAQWWRpAie59KZpmAyMiuRrQjEEFv6Z/qjLWXyWYivN1OiSPm  
EHST6WqjgprTlKek2kDYZkyRM/emuSSBtS3DxwDGmfFM8r8opzhKzHUBHYRFMx4oozIBS0AEkXrO  
Z/MrclyGwim/6mxygWFltRkHy51ngGcDWSLuQNX5d6aEV+TJSXi03AMAYuEZPK1toql8IHHbBNho  
U7+c1f8Ap/ShIR5W29mHYiqMvhs2exRAA0j26ig45bC4qrLP09ksMI7yV0ftQJ5copzkmbQHW2hW  
Bs3T/NDcS4ew0vhAsWIDIwmRHtS5+MIh+EpKRFjc6toHrU5cb5GmwOKbyPs3mXVQWCGByYG3kaTN  
nEaQy61NoAn3obH4gqQ2OPiKWIMWbTNzPWR0CJiQPEjjUkHZSLAnrSrioRoZcJxMJ0OyAGAIuPTp  
UOKOiEamQOegEQKry2Pr1Ii2IMED87V4nDQoOoglqQTsfIVJxUXTW/8ALE/aBMTMa0IF4043os5t  
vg6CgZtgxB1AedCYWKIME+HC/wB0iP8AmrscOHDq+pCI0jlVF2ukq/YylbopGAwQkAfc0HhA6wCZ  
U+Hym1MsvnSBQcWPMkW/NFZ+QBimHIJHjC2n02p7lF1Q2mILJYCdUCZgqyCbqDa8c+81dneLMuo  
pEBiBbeicrxFHwx8L89Ip3KXqWFXf8CZwjGRm4HQf8ANW5HPuhKnXkeQ2j969zGVUEjDe4kjUCI  
HSh+F5MrqVjEakHeem1VfVoDQ6ynHMFV0hU0jlg/WjjnsJrnBSSBptSfL8NcrqVWPpRDo4t8M2A/  
p7VF8MLuhkbV8NQYEA2Jjy5n1+lKMR1awKCTAI3j6xzphlss7FzaSBfkDymrMPKqol9M9du/W1Io  
278OlCwKFKkggnsfr0q/MyRESCNyLzsPLnvRqsjnw1SFstBJ9Cft2rs1gKotAAvP+aMk0hmKHJVp  
Z2iII3EmfzRSIjL4T4ovBBivdBIJkKdjvHI2PQldw/KrqkOyneAZn3BtSJ269JXmgfDaRGwPW0Xj  
aoKniM2IiNr3/wAU/GS06CAGgcXI8496tbJKs+ERF5/c7VSMJJGutMYzOCGLWGmxkHtfbyqnLIVS  
CIA7biefajsxkheBBELSCfIekDyqCKFUodxzPOazfglCXOup2MFSO/ODHlJt2oHAyrgTqliLMDY/  
zpRmdyelixWwuOlXNALXwkJTE0M99ESrA+c0/LEZ6phuQwjoYMYLXtE+1JXzmIjgpEkEXvaaY5HQ  
4K6tD/2nwnzDXH0r08NXWA5uLwlie4b5T70Vi0x1bjRHL4mO+MzhmeTL6RYWibbVHDZkzC4eIhIc  
EEkzfl/O9PMvmsTCQIihJ0lh/Ux8opVxAurKXUg7Ger6fy11JvFCJNYZrL0h+wJ9ga0GVQWidqWu  
wYpG5v78veaa5VTMaYtSS0UiqFX6hQlB/urP4R0kEiVNjPSeXetP+ppXDUgE+OIrPPlcViCMNgp5  
tb6UYJ1ZGcX2wHnATDx4w8RwECx3vynatHiz8ItYMN7eKORnpWSy/BcySSMGR/vW/TnWnw4wmXG  
V1IXwnl0+YWPvTyxkpG0soDfOIqHDeJhZQfee9VthYcsCFDsLM4FiOh60LxnF0NoaAdAKuqxjEi0  
Wv50u4Rj6w4xCToBIJ/qjLSKLeUSk7eDs/1lLqhZWIjYzfnTHAw1BtACz/VM9qC4UinFd3QMq3VS  
evSnHwEnVGm0Wv6zzN6MrWDKLSdyWcxMJGRV1Ame1dmWfUhbY2tt3Hnarcfn4aqUQlr3JG3WDzqz  
A44j4L4WgO4ICtHLmSalO7YKbZmlcqjgNIJA0gAt6+VE6zAPw9KiNj1EmxtMV6ugYRDAGTIt8vka  
GTPsgRdKECRPPeRPI0kk9oVprKCM1iuzr4Anpf309ax4mjBdSJGg25G1Jm4ijKEZQW97/tRuW4ic  
pV1EE47Dt2NBckovKGSx2aybl2AUgWIIbVInYede80loxwytyNiset63WDxKEZMJQmlZlnSWPeD  
WLbiau5Lu7OTdoAqim5LQ3oJnMwA09DE9f4K7CXQRFSZ738jROyYKOpGGx1TIVrEnnBNB4+X0gLc  
tZTHlIHnePSnTTRmeZbHdiGLMDajUdx2q7OYbl2Idt/7zUslgsphkIuACR9KvzTy7RAv0rN3oyZu  
lxHVG0nxR02vYR5UImYYrLGSNyQPxtalEr4gtjBBPQdL++1B/C0nSviIkwe87CIiubs9HSnk5XYm  
LGatwXag3Emx9KrXDNrRtv3qKrrgGCJ7isngbwLfmZMAXsd/KatyJAYMeR85HX60pORDHcHtttli  
/PrRuVymnxXkjaTHsfvR9BV50K42rpv528uVe5vNBEZmBIUTa9Z3MiFJkqbXULW9IvVeHitiAqcU  
sOhJ36d/8VTu6ElEsHEEXAWVAhG9r32He1U4uPqElbXuLz7XhtVPw3R+QEwOm3UfmmGF13ZYAt35  
0mXs0dZ04q6v1GIgEC/pt67e5rIYUOqO6wwEAjmbIFvStLmcNWBQggqPaOQPtWawsDWvwvbySadh  
c8/TbvVVMiKo4Z2KmE6MWlmTan72BkbVRw3iLoQrr8VB6sojczVZgcPLnSu/OT71HHwGwnhhz3mxp  
It6WSEXK8Gi4YqOgbCxS4U2BjUkiI8q7OZHWRgGGaLm4kfbpSrPZDQ+rCbQ4GoMDY9iJvTbh+aZ0  
Bez7MBtPP339adfSkrWRZgIyEBhBWR6Hp9aZcPe9z97VfiAEUsbFYnSkhmNgPP8AnlodW2Pxysa5

jiSoLKQCd9z6dN6QZriaO6lgbm88o7b38q0WQ/Th0+PeOd496pzP6ZOSWXT2ERbzq0Ulos38BOHZ  
oN8jaT/aDv8AiYvWgyWYaOv0n05Ghst+ldJVxZhazHb7Uxfhj4aaJ4wLxZhrWaQUxXx7hKYiFlQs  
w+QC2kze3TttWOPCHwi5chBEXPqbVvXzUAPcXAke1x9I71jP1flFDq+tn1hnAkQCY5ch/mpnPNGi  
U4p00J2zUTA3PvTnhucUnRF+U7Gs3gDYSbg1OSbyd5FGUVJUtu0X53GCO8rJDHy3qvK57FwW04aB  
WkRBAP1oPMkkkkMxJmW/PWi+GIDquAwg9Pan6pRyTGeNmbAMdDXmIM/715eleZdGY+FlYcmm3r0q  
nMYLoEdkl1LJLTzjy896Ez2XceNFCCJ8LEHreTSKKaoZPBoP+n4oJZAKMoKktewg+lTR2KMMQXMD/  
AGmbGRyilmWoiYIysuqDMj118qLBbDVwyypELGwPY8xvUnBgx4FPhoZh3Ui1tJB5RB3oDPpJGqNa  
2B06S3+4fml/F8fSWAEAX7jYiKM4dx01AuMq4qnaFMA7HnTw4nFXyVFPg7EzBMFhH10r3L8QfWCN  
43N7jb6VdxjLqro+G4fDcSs/Mu0qfKqWPjILAwZA+tU62sjPIflULYr4mgsCJ6CptVkObhQZ59aV  
ZXE17bwTPpRGfmcUADblpXBLSQMfDcPmQXII1AN8sXFzB+29VY2OxJM6B0Av0N4k3586Ey4YPKQS  
zGWJiNxYc79bV2IXBJdlm+51n0AgA22rkSfo6cvSS4TESSTBBBJNWrl2VSbIRf89xQeJiAz/wB1  
5ttCgbchBofGwhZtbA9S7c/M/SqLKodNhrZVgxa5n+kC32tVyGLkFSJtPnVGRd4+cm8Ac+x3qzNh  
wssAwF5te9xH1pX2s1tFuNhKfFLSt5Bb7fzevfitpOkzB5ib+1KmYBmtaJX15HeNqC4WKCoYCYJ  
I5c4PpRdmuy3CxnZTqASREH8elMsLOYQANJEgi03vs09KczjhAMQoWG8C1iofe83qODjo5+INSg  
40svtYWN+/KnSdWa3de84SxkXtvY7bE85tWfUupMxZgZt088yBT/AASBYc3sepPLeZ2rMcQLBrSR  
v2/lqMG7M3R7mcRx411IbXGmoZvEJVA4brqteeopNh54lyGLGbagbD6bVZhZxT4GktyP71VQdEUm  
OERAmqSvc/4JNEcELKTJDI1tSmQDyPUdD6UixcrNy7SbBZNqb8Ix1GGNII0iCetDqvASwmaDEgG/  
8FK04kmWdsw6lgDoUDrJ+wFC5ziLC3K3saR/qjNyMNBsWL/QAfSnjG3QeN1Z9N4F+rsHMTogPx3K  
sOXUHbnWhfMp08xXyHgKf6ZHxZBcrAg7C0jzr0/rv5skm/n53MbVRRXhW/p9D/U/6o/0qAjCL15C  
7gAxzP4rC4H6/wAyGDSUfDPzppFhMRtP1o3LcaXHw2XF06WvDbClWd/TuEUJw2Yzex1Afn3rXH01  
Pw3W0uG2CGwzOG6TE7Wnff8A4rPtIU2WfQssrkBjpECec7i5pP8ApjN4uGWwH+UjSO3Qg+VP+FYJ  
GUOHYU6sNue4Mc46VCSpBlmJlWwtIPxFGqCdSxHrFqVtmIcEDUFO1xNHvkcRdWkJECzMBj/cpuKF  
R0ctrYqdNiBz6Wp41+yFkczjq7wsik2nZek9aGRG1kMYjwHYNcCZ2iKu04QU6pZt1I5Hoakwcw  
jqVxHiB4VEST07U9lpGQzKMqgcR2YKSSlp/+LGxrx2V0dMRwyKRpgQYmW0jm0YonLZk4mAQMqmhv  
EWWxWCL771SMXQACEcSIKgsRyMGoX/s0q8E3GHbCfDRHlQgutizJn6Vuf0vj4eLhFGE1BDahfre  
szmcdy+nQ0A2aFI/+tT4fiBS5RyjtAdSIBg3IO01TtjJvAjMsQ4hUdJKlWF7G0H/AGkUt4vi5cns  
oKlQDAjwHmJ7/inGO0qTIBiAw6jeR5GsvidFkh1BDSJUD3oQyGndjTgj4DEI91bY3BDde3SrjwB3  
xGfCAZBYEkAmBH4rP5XCKOJBXYciKtGaxQxc6lF4JmIvsNqPRqTae0ZJphqZLFwzodIsxFwbxba  
muV4UdC6nvF9qG/T+XLicw8swNscBH0orDyWDiAYnxGXVeDnqWTUdjJIdlGE6XCAC5IEAHvO+53o  
NsUL8g1k7vP8+lNshwzWGBBcksBAMOp7VHM5QDZCItLAjnymuOM1thT9YlwSzeACDlue/wAxmiv9  
HJGtje4BP860bhKASBvfl+a9TDYwQbjsPvTSlWh2kiOFkkWfkie+1+vlTLHRVS5WIsW2360M4kXE  
9e9B4yQAumwkDt+lNGVoasA2aZFYqjXHMxHnMH+GistnfcFZDPVVMH81Ti61HzKoI6XMRavQwLgA  
kTewMz5UdoXWRoMbCCOhZQW3DjSte2/kKHGC2HqKRpcQf8e9ChmcHXpK8iYgdt95+9FpglgIteev  
f2NB2jbKy7qoAYet579TQOawFdgXBC3JAmbDYc4/emGJgmDow5dvPfakvEXdfkZgNovY9ibiek00  
MsDTrJ6vDi6wmEwXlqt5eda4HAHL+GFjdieY351cm87FA2uVAg3ghLYfej8hw5MMs2IWPmCgDva9  
6sm16SuhZjopEkKONtNy0dutMeFZF1SwnUSB7wPsSapTgZxmXEDkLJg7W3sOfSm+YcJh6EnwjfmY  
6mlcvjyXjB03L0zOfY69LAhtiCI8qXvhyw1brJA6loczxBXdf68z1jLXmVw10F3AMXuBysB9ftTd  
mtoiuOnaZmsTPh/CwiNiPzQb5Yk7j70xz+GriV+bew3pUUCda6IvGAsa5HFCL4oB5RMnrvtrQ4ox  
ICQog2H3796z7hj8u9EZXBxAQQdv2gz6HahKKcmzYcNzBILncDpz2tTpVP8Ap2gWQAY9QRf0NJOE  
YDBFUi035PfetXk8raDbz+ornkiyVqjEHi7YjBTyTE1SSFA1MYHta4obiGXV1Z9IDneLISNyB1NU  
cdyBXMvhhAgDQOQI5NO0Gd6hm+HosKc8ryZKqrkf+0QarlimqOSUaYG+EQhbwEKdMTczzjn50flf  
hKod11E/KomGdyaY4n6YV8PXhM2IAQNY27ggb351nHyj4eI2Fq0nVF9j0rKSmmkxW7GSZxcV1DMS  
s7AXXoADY1PEYjawYtspuNvzQGCzFjCDUSNthG9PcAsBDG29zIk7n2pZLroZJs0PDuHkLqZ1A0Dk  
DJ70r4rw/AKNiI+llIDrO02mKIyeWUafCdizX0kypINx2tep8S4OmLmCzmG03C2DR1IPfXbI0dgO  
HlyjIreIOsTy8QtHfa9LXcaSuspEXYT2I7itWmUw8bDR0LGQAjQ5Rs09L8zwnDkM5Aknw6oDTuK3  
amNKSukJc3wWUXFV2ZtiFFvMU3yP6cZcDRiNOu5EyQIsI5GrcLMukqmCAq7AST7xApjg5osoZ10E  
2IJmPas5S1YrTRB8IooRMIEBdMzy/egcLh5AA0taruKZrFRVbDQMpnxEmLb7VR/lodvesoyMoyNd  
lcPwAmjm0+AgMscwCLnnXuWzLsSuHmUxgN8LMLpcdpPOPKq2UgjQDq84H88qXZricmMxhDEtBx8u  
It/7139ahwtOJWLwP0ySNOvBfBbmVYat5XMelQXI4JgKHHk3+KrymMhw1OHiiYjLyD7jtEVNX5Hfn  
EmPXaq9I/CtJnr8FR18OI6nl035xE0JmeD4iRHjVbiN/YmTRKSLqR1vJ+1qv+04iIA77nyAmIFF8  
cWhWjOFiJLGBvZhHlvahizG5UqOx+vWtVmMP4nhdQw2Oobc/MW70nzf6eGoNhqYPUnblpHxUBrAu  
OX1Jpgsx3tYjp05qWSyb4SlbaTfxMTznmbD/ADRGDwXMggRI5nwgDn1n6UY3AMdrkr6kCfvPrQ6y  
qgIAX8e5B0mLHuI5XvSrj2XD5dinzddoI2pjnuH46C+GYHMR+Deg8zhhttpYysmxJn+cq0YuLsDuh

dwrMOERVdAWuWZtWntI2HSnGiYOIQWakgGVInc9eVJMDLrhrKkqzEhrdzYdKapilKAAAltjJsR+Ka  
bzZLCyMMY5kAfLyAFvIUPjZHFxCUAVF2LMPsBcmg8DHZX0FjOk6BpIBYCdIabyJ9qIfN6kLM8qPm  
g7djf6FPwMuSTSBM7xHIDAxUnGR4Ia0yDO0XM/SmWfhstILBFz9v3FE5fJo80iIqWYaJJ9Db1NeZ  
KNDglidQsYiR5AU7t1/A0ZdnVGU4bwtlTViWZiFCnKOC96PweGoQQBc8yNv5FMjhaidWwuPS5+5o  
jL4ABHQ/f9u1Vcm8lVBIW4XA1g22EfX9quy/CIMASbW9tzH2p+EjflP88uXej+E5eXBI1ATJ+vtS  
uTGpIVYeEYOFIECIH5pwxKhWPPke4t/zR3EcBCw/uUift9KJxcv/ANsCx0g26g3EpoUEzfGsor4  
HxNQRgQrOs6o2AOkibedYHiuXy6CFbEZ43KAD63r6flVJTFXcalj1A5cqUBFxcPRj4LLIIGpZYR/  
UCNwVl7dXbOfLTts+ecM+O5GCUi6IAWADEAHfydTtf8AS+MmO64OZOpdcq5s2oAwRnvpIke1aDK5  
ZMAT8JS7RPjHiYwsvKlfd0xmZjuHTDwCpBgoVZgTtMwx503dSi3RHNmgfCyxchMJbWLKIHuN6Gzn  
DFZdYECfl6dqfYGS+Agd2DIgJMLeAOnOkuPxrDIbELAJcxPM7L5xU4JydzjZSE44i2A4OkkRpGLIH  
n0q8Zc4rkk6UEeKZ3vA6UFitiMdZU6XMgDmK7M8UCK6QVIEgbGSKbN0thcqHWYz6YCBZJtYbnlpZ  
mOKY0K6Jht4j4d2PQg8rdqWZBsJyVxXKuWfpu0LDznF+G4ZRqhiBuZAAAnzpowskyjeWaLG4piPh  
NoAR0uw3gDcVNcyThLjtAk6HXCg0y0ludQ4M2HilcTCXQfEMRCLmQRI/uEnah3TGOAcPDwpRjN5B  
Xy5R3o1FKtBVUaHgwDI5SSiKoQIPSOlqT4/DAWJCLf8AnSgef42Pggh7sLAEgeH03501/wBcWvp3  
7/4qTnOLpZEc5RwaBmaAp3Yct7cp7ikWdcH+krExtAFvpTvMYbaDqbUwMgx0pVm8ZgPlkX353/xX  
PDZdOtI3B4iUYpqhJsbyDub9OvpcU6wc7YarT3t1nuYuKQ5hG3A7dx3+lAa3SLllvY/iuuMvAqTW  
zf40aDW5XJJ63/yavTNSJiDcgeQ5/esXwviJZWlvFJn16e5rQZbMSFIG9rf7ace7HmRxYQAm8+Ik  
99/UxR+C8je8C/0Ples/lSG08t/Sbfb70blczYGZlW5dNv53rGGaYoJLTikgwZII8JFu4NDZl2Q  
6gCV5sCbdPDMXrxGAU3Hjk9pN/vNUYgYqVvocNKzz3kdwb270xgtM4n9bW5HxQZvvUcxg4Di4kDb  
94/xWexeIPhtp06iTZSYJkGNLGxPhO8fkmpmiQdLMjLurgKEESCH0zEHe3TcUKsFEOIcADj/ALZ0  
+dh9FP3pBj8FzSMNGkiRrggyOYA3mOdq1OC7wNUKEXKywB/3apBonExQFJ3i5kSR3iJ+9DqvRHCL  
M2yNqGsDQviiYXtf0g3pFg5XWWXVGoEBLXeb+LtlWlGmMIDpVXBEEjsfTrUHYiYaeFFQDuAPcnekU  
KMo1hGay/B8REGH8QsJmFW45ke/opWvcdSAEW+0xf3PpTHNviHwoFRd2LMR5yQpEdjVGDlX+Z0Jk  
2YENaf8Axmev7GqUUjFI9w8AaQ0SQfoeo6lcmUAYmDztvt5SKIXAhd5Ahtlm3IGp40nfYHeeX8N  
AYrwcFmWVE29R6c6c8Mw10hogx7H9v50qPDsMB2InrH3HnNeYWMUc28MzHY9B6+9KzEcW2Ix3v8A  
z6DenaqGTtSDNXckHew7jce23pTDhWOChEyR9qWGePlQmIzDZ0W3KVgD6VlflBxDM4OJKkhFbxMQ  
CrCZAiJBgitfmcY68McmDfj+elZTjHEUTM4mFieLCClq38BAABn2NKJPRHPcRR0DFVeD4NBiaD2B  
teoZfDzyAusOTG4n67UKuSK/I6sm4YLHoTsDSzW7FyHYGSCSTB8jSVK6Wjnbzk+gy2VjCZfBVksA  
WPUGxr4yODYiFg0gobqdie09udbr9I59wr4eKxYKfASZMcwfWjOOZzmdMRAH2GJYMYBXSQJsaE+  
snEZ01ZmOI/qDWgRERFEKniuO8AQpegcXKYOCnFSUvJMG/JpHLauz/DsVsRnbD02DeGNJ66TsFvU  
8bExXCbHZtAI0EbCOnOrUrwJl+HcMyLO2rSjBBdR8wHY8qJwkAcaHCGfACJ1CImYIBFAMmIFUI6K  
QSXkkXPY7j9p6Z5XPMEYAqGAftzteLbUssKxm6RLCXHXFIJBDGdURHKKeZzNhFlisbADqaQ5Z3LK  
zXYA2J2t0rhiquITiMwY/KvIfvU3G3YqLSLuK4a4phEb4nhJBYYBY6Dr6VDCyOMAP1/8A6qjOeMqF  
cnTz0/Sau0N/efcUVaF7Xs2oQhSCZBkm1/L+e9KM0gDGFk+vpTDNPZVCyAf0OV73oHO41wR/UvL6  
ly3aR0S8Fj7d5P70vzWXyryuOthICT5+VdiIZg2AH8vVYyZlctmcGF8NtSss8+9/OnXceKgtPne8  
cwf80uz+XgzI78qUYuCAQUaGHQ/muqLUthvro+iYecCtvPhPkYuI+9MOHZNwQJtqj/2BH5+tYXI8  
UD6VazAAE8iRb6i0U9y2d0LC33F/ceWwj/NaqHUrNMmYuCNTvyDHvRGWxFZVc22Y/ePSfpWZTPBi  
AAezjzAMft7UamaI0sLg7wd95t60QjXN4SsQpvcEWHI29efnVbPrAV5tdWUKXG5nrcyDl9ao+Nca  
k2gqR5f8ewqeYxpWbfbWtbof52oWYkmZ0PBYDcLq1Tz8LG6nt9hTUPSQJ6qdxSZXV0F5EG/rsR15V  
fgZmFENEWvf0M38v8VrNRdmuGIX14cAn5oI8XUQbTee/3p+AFkaokzBSJ67SPpXJmgR000Tcj/8A  
QPL0r1seZUeIHZEdx/yD+Qajx10iUtztch/4nr712ER80BS24E6T3v8AzyqzDDaYkkd+X7/mp4WV  
vcwGi24Ppy89qNhIrgjZRbmN/frS0pp8Dxa3sPvH0rTJh6V21el/LzisvxN4Y8559CBPlifWlNY6  
4e2lCZvFvIWH1kULjZkBXb5WPWNPs0ikud4iW/7abCCx8iCV+hFSRY7t5T63j6xWMhlmX3K8jPnB  
I+1E8GcFwRvcGhcJSQ02uSfi7/Ufer8q64ZU73/bnQYRhxFCCjAToafQidWT/VuOBiWQNqRdZPl9  
dhW5x0lowG5FqxnGMPxI5IA0AMivKkrvz2pXS2S5fxMsnEDhQrBWw3EOkk6fi/mn+WwUOGLjQICK  
bx/5d/vvf+jwHA1rpnblvb0oTHyp7SjasIjxKTt/j7UXNRHsmqZTiYhr20IYk2UXItf6U6ymfQo  
pIKuLQ3TvS/MYqKMPFw7qTDHpyv0NHYiAi4kGpOKu2I7i8ijjWIXYfCWFkFlMgD/AM15HuK5crgu  
2pczoe4ExpI5im2DpGoETK6R0HTyobK8OYFw4RcP+kKLnq17VX+oooKml0zXGuE42GusurjVHUGe  
defp5YMRVxAbriW4I32N6licJwwZA226e21DcV4R8VlZ3ssRFiCDiVWXNGSpgr2Jum5d8tj60Xu  
u5BHMxrs/gYmMUZMKABJlgCT060zz2K7hEK6gDzIkHrPSGxXEKt8NtDC51f/AJOxp1JatC9+GZsM  
CqkCRPOP3pj/AKgiwmltvf60IuNjo6F3aNEQw9L0R/1TE5fas8jKkaw45IvHtVbpIHbbtUENdms  
0EWTzMC8CYm/tXiKUpSSTIx1KTqxccOGixI630luVVxrIiRCT06kb73oknXpYwCd4G9SwsFQJEK9  
zyfztXpQtRzs74RxTAHwBHyme5BJ/alGPhAEmCb37+laBsJpIkwLyR7AQKV5ggMVG/lyU8ZNCyAQ

ozWXsYgCvclxFkEPJjY/g0Xi4JJE370JiZcXIqsZrTJ9qeBzls0jQ6HlB/n19Kb4eaiSIvFjsbfQ  
 73rCqHVgV5dnJTDLcUKkhxYj2/5p6+FFP6bhGNo20348rmI7mrCZGoX6idj/AD61lsnxRYgGfWDW  
 iTNKyl0IkjrZ8xtP3FKx07KXcp4hseXI/gVHDzIJMSwJgjnPl2otswpgOAQefMdj3sb0rz4QEFDP  
 YddjGx7HvWCaAkhVOnUpo/hyBpLDUOX5B5TtcVm+GcaLAIwk729Zjlg+tHYfFiGbUQB19N6DYUzW  
 YSLEWjl3HKe/7d68dkAgACJt0rI5rjJ0Qranmw9In3Aqa559AOqCGkk7EdPX80GY0bZsQb269D/B  
 WK4nnZZglzefaF/A9aJxOIBFbcFpsf52pFlFaZ7ye/8AcPKxoxQGw7h6i5Jkn6yQfua2GRwF0KIG  
 0SOc3B9iKz2Wym36H67g/wCaf5TM+gv7T+LUQnmNh1YBF7g9wQCPqKhhJqdbHSSARz3/ABbzmicz  
 jgkc5+xuKKy2CFIJ8/59PrSmsJR9JjtfSRWH/XAZGUglQWMQJSDfsINbV2M7XJ37f8AFZr9a5ZX  
 w0ZgCJK3N78rUNK51ldmK4RxVGlMUKAkajs3cVPLocUFImP6ojymvWCosQFHKqsDjAUHUhAG0bya  
 Zr+05lnSDcLLNgGZ14REOnTv5UZgZAodeHilsI7Kbx2M7edVYXEUInVA72BocZ7/AExl0vBdcwyT  
 9u32pMjpt4YV/q4JDAKQJ338hvXZbPer/UVN5I/lqF4xoVsPFElGnxhbOek/WrcDOiywhnTApelk  
 pRawTXOY2o/KF5CL/mpkljLMD9qghJuaHzWfVBbxHoP35Vld4BF/EXARiX0k2n9qyeGuJrhcUgna  
 SYPQdKs4hlshBxQSSKAAfEC06kVa3AIJOgelzV4xjFbHSS2UZ3M46ko6iV2mL9/K1K8TOtJ3/wDa  
 n2Zw20DV8yCFuDPbrS34oN90/lVEl8H6o+ho80Nxo/8AbT/ePtFcARtXcRacu5sSpUwfMV4fEqmm  
 c/F+RblmsB/VHTy26V5iYwX5FIHcd+gHaqMtnvAI0rYWak7Gp4mNCKhmpOBXdR6CurBcVsZ9pCx0  
 I95v0ofHyzf10BI/uv6Ci3ckQR56mNBuZPKT0H5p7FkkDBov5/gqgkMYHLeimTVteedDHDN6KJOJ  
 Uydv80Ni4cn7CjMfFNgbftVGKk/imi2tgWNgv+m338+9SwcTET5HYdtxv6LJA6UWuVsb3pnI1ZsG  
 TimKImp39KIy2K7yd03IEW9DBiqMXAjbvXuUzIwysXgG16N3oaTkkMcvjhPmVlflY3NCZrMMzEspv  
 vMzTEQ8RBWA07/WpZ9MNCmMSZmCsir5Tv+1LlMlcn6B5YoLglT0m/wDIn3pq+aCp4mAFjj25c/Ss  
 78SRKzHI+sfioI5cFZtVOtlIyd0zuIZsuQqMYWRM7zWm4UnhBPODA+v871mMngDVpJjoTWu4DjKh  
 0Pty09F/EVivpoMniACDcD7UVjYawHUWZ+vQ+Yt6lQ+XgSu3Lt/g0F8bdDYzz+xFIUCsJmnURES  
 P17ldi8WAYKTO3lG/a9Ca2UbEW26mLHzH1oN8qrkMx0hrx0gzA9zQMPMHYcWtex69ftST9bZgDLg  
 sHMGsVAEWsTPSf5amuUyyq5csNA+RdotH4rE/r/ieHiOgSTolSV5be+59qMVkSbwZfioYbEuJIgD  
 uBsNquxcynw7DSIhgbme/qKGfGZVBVoG9oBj1qvBzCsRJnnJ3v171XrgiUY+YEjTyv5U8yuKcTD0  
 YkEASDsQeVCtnMPQyQCTYurZ6f8AFD5PNKRGTZQSY8961X4GLH2RxiilcVQcJ7MpMx+1A5/KMmIW  
 BlGkpFpHQ9xzqGJhnFUOrAKfmvyHemWQxgR8LEup+VuhqbaTwZSi3RRls4Qdtjw2tn60eWwss4jS  
 JPI2P+azuNlVAZCsQT/UZI5GeYiKafpjCUo2oE6PCNUbRM1LlS63YnIqC83kVQeH5SDHagy4kDkB  
 TrHw1KkWA+1ZLiud+HiaEZXBZeNtp2JtS8LclXpNK9DJ0Q3dQQL36edVtgcPa979GaPtWexOIM4  
 JItGw2pN/q3/ALo7Sa6owl9HUA9PouexHVzDWvEgftXiKXwcSTf5ffY+9dXV56iqM0juDYiFFX+q  
 Om1WY6iCASRM1ldVns6l+ILpjckBznVwxFOQoAHl+9dXUBSkN/bseleYiAlIHn0HeurqPoAM4J6  
 yaqGXg3/AJ+9dXUG8kZfkCMPxCltqtRzME1ldWYDjMUv4rhkgqLncAXP8g1ldVOPY60F8IRkQhHE  
 QYmSBclNuIqdSlgRqkqRMdDXV1VebEeHGVZvOMDoQAajZdgN4ovLZR0QlZJjny5VldTLsgj+RPAQ  
 MSOfKtHkcElZAmNx+a6upZHREdcOxivhNxtVmcyoN5rq6lY5RjNjW0w6g2IvbpFW5nIF4LE8iVA6  
 d66upWAo4rnlCaLyVKkzLLI286+e4GScsZeQu6+VdXU8SO9lWJIB5RG0fSl78POog2HKbVxldVYg  
 cUUYuEyWO02NDKh29+ldXVRaJvYxyGDilWVJJKGJ6U6yze4YBlXFu89a6uqhI9iPDGPDcc4gOG8fE  
 QeAn+odJqrM8RdEKOShElhHinpXV1JS7DS8FOU42+tdR1BjBlywjyneneazHIKoogf0yZPxtbpxl  
 dR5F1qjOTS6laYaxBW9Vf9Lwead3Ne11TUmTP//Z - local variable 'contenido\_imagen'  
 referenced before assignment

EXITO - Almacenado <https://lanotapositiva.com/wp-content/uploads/2019/11/aoi-sdnoa-homadas.jpg> - como ../datos/perezoso-hormiguero/prueba/oso\_perezoso/c5c433fbbd.jpg

EXITO - Almacenado <https://i.ytimg.com/vi/g84asJsXS-U/hqdefault.jpg> - como ../datos/perezoso-hormiguero/prueba/oso\_perezoso/e7169b9681.jpg

EXITO - Almacenado [https://st2.depositphotos.com/5909840/8643/i/600/depositphotos\\_86439650-stock-photo-happy-sloth.jpg](https://st2.depositphotos.com/5909840/8643/i/600/depositphotos_86439650-stock-photo-happy-sloth.jpg) - como ../datos/perezoso-hormiguero/prueba/oso\_perezoso/571811a577.jpg

EXITO - Almacenado <https://i.pinimg.com/550x/de/cf/2a/decf2ae44b336a0109206eaa6622e95d.jpg> - como ../datos/perezoso-hormiguero/prueba/oso\_perezoso/a51b728d63.jpg

EXITO - Almacenado [https://t2.ea.ltmcdn.com/es/posts/8/5/6/por\\_que\\_el\\_oso\\_perezoso\\_es\\_tan\\_lento\\_20658\\_orig.jpg](https://t2.ea.ltmcdn.com/es/posts/8/5/6/por_que_el_oso_perezoso_es_tan_lento_20658_orig.jpg) - como ../datos/perezoso-hormiguero/prueba/oso\_perezoso/f5234c0860.jpg

EXITO - Almacenado <https://www.hogarmania.com/archivos/202110/oso-perezoso-2>

```
-sacando-lengua-XxXx80.jpg - como ../datos/perezoso-hormiguero/prueba/oso_perezoso/f1174523bd.jpg  
EXITO - Almacenado https://t2.ev.ltmcdn.com/es/posts/0/1/6/el_oso_perezoso_esta_en_peligro_de_extincion_3610_orig.jpg - como ../datos/perezoso-hormiguero/prueba/oso_perezoso/845fc1d7f1.jpg
```

## 2.1 Visualizar una imagen

Después de buscar y almacenar las imagenes a utilizar.

Vamos a visualizar una imagen:

1. Obtenga todas las rutas de la imagen usando `pathlib.Path.glob()` para encontrar todas las archivos que terminan en `.jpg`.
2. Elija una ruta de imagen aleatoria utilizando `random.choice()` de Python.
3. Obtenga el nombre de la clase de imagen usando `pathlib.Path.parent.stem`.
4. Y dado que estamos trabajando con imágenes, abriremos la ruta de la imagen aleatoria usando `PIL.Image.open()` (PIL significa Librería de imágenes de Python).
5. Luego mostraremos la imagen e imprimiremos algunos metadatos.

```
In [18]: import random
from PIL import Image
from pathlib import Path

# Setup path to data folder
ruta_datos = Path("../datos/")
ruta_imagenes = ruta_datos / "perezoso-hormiguero"

# Inicializar generador de números aleatorios
random.seed(42)

# 1. Obtener todas las rutas de imágenes (* significa "cualquier combinación"
lista_rutas_imagenes = list(ruta_imagenes.glob("*/*/*.jpg"))

# 2. Obtener una ruta de imagen aleatoria
ruta_imagen_aleatoria = random.choice(lista_rutas_imagenes)

# 3. Obtener la clase de la imagen del nombre de la ruta (la clase es el nombre de la imagen)
clase_imagen = ruta_imagen_aleatoria.parent.stem

# 4. Abrir imagen
imagen = Image.open(ruta_imagen_aleatoria)

# 5. Imprimir metadata
print(f"Ruta imagen aleatoria: {ruta_imagen_aleatoria}")
print(f"Clase imagen: {clase_imagen}")
print(f"Image height: {imagen.height}")
print(f"Image width: {imagen.width}")
imagen

../datos/perezoso-hormiguero directory exists.
Ruta imagen aleatoria: ../datos/perezoso-hormiguero/entrenamiento/oso_perezoso/ae6c78e31d.jpg
Clase imagen: oso_perezoso
Image height: 283
Image width: 178
```

Out [18]:



```
In [19]: # Setup train and testing paths
directorio_entrenamiento = ruta_imagenes / "entrenamiento"
directorio_prueba = ruta_imagenes / "prueba"

directorio_entrenamiento, directorio_prueba
```

```
Out [19]: (PosixPath('../datos/perezoso-hormiguero/entrenamiento'),
          PosixPath('../datos/perezoso-hormiguero/prueba'))
```

## 4. Transformación de datos

Ahora, ¿qué pasaría si quisiéramos cargar nuestros datos de imagen en PyTorch?

Antes de que podamos usar nuestros datos de imagen con PyTorch, necesitamos:

1. Convertirlo en tensores (representaciones numéricas de nuestras imágenes).
2. Conviértalo en `torch.utils.data.Dataset` y, posteriormente, en `torch.utils.data.DataLoader`; los llamaremos `Dataset` y `DataLoader` para abreviar.

Hay varios tipos diferentes de conjuntos de datos prediseñados y cargadores de conjuntos de datos para PyTorch, según el problema en el que esté trabajando.

Espacio de problemas	Conjuntos de datos y funciones preconstruidos
<b>Visión</b>	<code>torchvision.datasets</code>
<b>Audio</b>	<code>torchaudio.datasets</code>
<b>Texto</b>	<code>torchtext.datasets</code>
<b>Sistema de recomendaciones</b>	<code>torchrec.datasets</code>

Como estamos trabajando con un problema de visión, buscaremos `torchvision.datasets` para nuestras funciones de carga de datos, así como `torchvision.transforms` para preparar nuestros datos.

Importemos algunas librerías.

```
In [20]: import torch
from torch.utils.data import DataLoader
from torchvision import datasets, transforms
```



## 4.1 Transformación de datos con `torchvision.transforms`

Tenemos carpetas de imágenes, pero antes de que podamos usarlas con PyTorch, debemos convertirlas en tensores.

Una de las formas en que podemos hacer esto es usando el módulo `torchvision.transforms`.

`torchvision.transforms` contiene muchos métodos preconstruidos para formatear imágenes, convertirlas en tensores e incluso manipularlas para el propósito de **aumento de datos** (la práctica de alterar datos para que sea más difícil de aprender para un modelo)

Para obtener experiencia con `torchvision.transforms`, escribamos una serie de pasos de transformación que:

1. Cambie el tamaño de las imágenes usando `transforms.Resize()` (de aproximadamente 512x512 a 64x64, la misma forma que las imágenes en el [sitio web de CNN Explainer](#)).
2. Voltear nuestras imágenes al azar horizontalmente usando `transforms.RandomHorizontalFlip()` (este podría considerarse una forma de aumento de datos porque cambiará artificialmente nuestros datos de imagen).
3. Convertir nuestras imágenes de una imagen PIL a un tensor PyTorch usando `transforms.ToTensor()`.

Podemos compilar todos estos pasos usando `torchvision.transforms.Compose()`.

```
In [21]: # Escribir el transformador de imágenes
data_transform = transforms.Compose([
    # Cambiar tamaño de las imágenes a 64x64
    transforms.Resize(size=(64, 64)),
    # Voltear horizontalmente de manera aleatoria las imágenes
    transforms.RandomHorizontalFlip(p=0.5), # p = probabilidad de voltear, 0
    # Convertir la imagen a un torch.Tensor
    transforms.ToTensor() # esto también convierte los valores de los píxeles
])
```

## 4.2 Carga de datos de imagen mediante ImageFolder

Muy bien, es hora de convertir nuestros datos de imagen en un 'Conjunto de datos' capaz de usarse con PyTorch.

Dado que nuestros datos están en formato de clasificación de imágenes estándar, podemos usar la clase `torchvision.datasets.ImageFolder`

Donde podemos pasarle la ruta del archivo de un directorio de imágenes de destino, así como una serie de transformaciones que nos gustaría realizar en nuestras imágenes.

Probémoslo en nuestras carpetas de datos `directorio_entrenamiento` y `directorio_prueba` pasando `transform=data_transform` para convertir nuestras imágenes en tensores.

```
In [22]: # Usar ImageFolder para crear los conjuntos de datos
from torchvision import datasets
data_entrenamiento = datasets.ImageFolder(root=directorio_entrenamiento, # c
                                         transform=data_transform, # transformacion
                                         target_transform=None) # transformaciones

data_prueba = datasets.ImageFolder(root=directorio_prueba,
                                   transform=data_transform)

print(f"Data entrenamiento:\n{data_entrenamiento}\nData prueba:\n{data_prueba}")
```

```
Data entrenamiento:
Dataset ImageFolder
  Number of datapoints: 100
  Root location: ../datos/perezoso-hormiguero/entrenamiento
  StandardTransform
Transform: Compose(
  Resize(size=(64, 64), interpolation=bilinear, max_size=None,
  antialias=None)
  RandomHorizontalFlip(p=0.5)
  ToTensor()
)
Data prueba:
Dataset ImageFolder
  Number of datapoints: 19
  Root location: ../datos/perezoso-hormiguero/prueba
  StandardTransform
Transform: Compose(
  Resize(size=(64, 64), interpolation=bilinear, max_size=None,
  antialias=None)
  RandomHorizontalFlip(p=0.5)
  ToTensor()
)
```

```
In [23]: # Obtener los nombres de las clases como una lista
nombre_clases = data_entrenamiento.classes
nombre_clases
```

```
Out[23]: ['oso_hormiguero', 'oso_perezoso']
```

```
In [24]: # Obtener los nombres de las clases como un diccionario
diccionario_clases = data_entrenamiento.class_to_idx
diccionario_clases
```

```
Out[24]: {'oso_hormiguero': 0, 'oso_perezoso': 1}
```

## 4.3 Convertir las imágenes cargadas en `DataLoader`'s

Tenemos nuestras imágenes como PyTorch `Dataset`'s pero ahora vamos a convertirlas en `DataLoader`'s.

Lo haremos usando `torch.utils.data.DataLoader`.

Convertir nuestros `Dataset` en `DataLoader` los hace iterables para que un modelo pueda aprender las relaciones entre muestras y objetivos (características y etiquetas).

Para simplificar las cosas, usaremos `batch_size=1` y `num_workers=1`.

¿Qué es `num_workers`?

Define cuántos subprocessos se crearán para cargar sus datos.

Piénselo de esta manera, cuanto mayor sea el valor de `num_workers`, más potencia informática utilizará PyTorch para cargar sus datos.

Personalmente, generalmente lo configuro en la cantidad total de CPU en mi máquina a través de Python `os.cpu_count()`.

Esto asegura que `DataLoader` reclute tantos núcleos como sea posible para cargar datos.

**Nota:** Hay más parámetros con los que puede familiarizarse usando `torch.utils.data.DataLoader` en la [documentación de PyTorch](#).

```
In [25]: # Convertir los conjuntos de datos de entrenamiento y prueba en DataLoaders
from torch.utils.data import DataLoader
cargador_entrenamiento = DataLoader(dataset=data_entrenamiento,
                                   batch_size=1, # cuantos ejemplos por lote?
                                   num_workers=1, # cuantos subprocesos usar para
                                   shuffle=True) # barajar la data?

cargador_prueba = DataLoader(dataset=data_prueba,
                             batch_size=1,
                             num_workers=1,
                             shuffle=False) # usualmente no es necesario bar

cargador_entrenamiento, cargador_prueba
```

```
Out[25]: (<torch.utils.data.dataloader.DataLoader at 0x16a330e20>,
          <torch.utils.data.dataloader.DataLoader at 0x16a330f70>)
```

Ahora nuestros datos son iterables.

Vamos a probarlo y comprobar las formas.

```
In [26]: imagen, etiqueta = next(iter(cargador_entrenamiento))

# Tamaño lote es 1, puedes cambiar el parametro batch_size parámetro arriba
print(f"Forma imagen: {imagen.shape} -> [batch_size, canales_color, altura,
print(f"Forma etiqueta: {etiqueta.shape}")

Forma imagen: torch.Size([1, 3, 64, 64]) -> [batch_size, canales_color, altura, ancho]
Forma etiqueta: torch.Size([1])
```

## 5. Crear clase de modelo TinyVGG

En el notebook 23-RedneuronalConvolutacional\_2 usamos el modelo TinyVGG del [sitio web de CNN Explainer] (<https://poloclub.github.io/cnn-explainer/>).

Recreemos el mismo modelo, excepto que esta vez usaremos imágenes en color en lugar de escala de grises ( `in_channels=3` en lugar de `in_channels=1` para píxeles RGB).

```
In [32]: from torch import nn, optim

class TinyVGG(nn.Module):
    """
    Arquitectura del modelo basada en TinyVGG:
    https://poloclub.github.io/cnn-explainer/
    """
    def __init__(self, forma_entrada: int, unidades_ocultas: int, forma_salida: int):
        super().__init__()
        self.bloque_1 = nn.Sequential(
            nn.Conv2d(in_channels=forma_entrada,
                      out_channels=unidades_ocultas,
                      kernel_size=3,
                      stride=1,
                      padding=1),
            nn.ReLU(),
            nn.Conv2d(in_channels=unidades_ocultas,
                      out_channels=unidades_ocultas,
                      kernel_size=3,
                      stride=1,
                      padding=1),
            nn.ReLU(),
            nn.MaxPool2d(kernel_size=2,
                          stride=2)
        )
        self.bloque_2 = nn.Sequential(
            nn.Conv2d(unidades_ocultas, unidades_ocultas, 3, padding=1),
            nn.ReLU(),
            nn.Conv2d(unidades_ocultas, unidades_ocultas, 3, padding=1),
            nn.ReLU(),
            nn.MaxPool2d(2)
        )
        self.clasificador = nn.Sequential(
            nn.Flatten(),
            nn.Linear(in_features=unidades_ocultas*16*16,
                      out_features=forma_salida)
        )

    def forward(self, x: torch.Tensor):
        x = self.bloque_1(x)
        x = self.bloque_2(x)
        x = self.clasificador(x)
        return x
```

```
In [72]: # Fijar dispositivo GPU o CPU
if torch.backends.mps.is_available():
    dispositivo = 'mps'
elif torch.cuda.is_available():
    dispositivo = "cuda"
else: "cpu"
dispositivo
```

Out[72]: 'mps'

```
In [73]: torch.manual_seed(42)
modelo = TinyVGG(forma_entrada=3, # número de canales de colores (3 para RGB)
                  unidades_ocultas=10,
                  forma_salida=len(data_entrenamiento.classes)).to(dispositivo)
modelo
```

```
Out[73]: TinyVGG(
  (bloque_1): Sequential(
    (0): Conv2d(3, 10, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU()
    (2): Conv2d(10, 10, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (3): ReLU()
    (4): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (bloque_2): Sequential(
    (0): Conv2d(10, 10, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU()
    (2): Conv2d(10, 10, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (3): ReLU()
    (4): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (clasificador): Sequential(
    (0): Flatten(start_dim=1, end_dim=-1)
    (1): Linear(in_features=2560, out_features=2, bias=True)
  )
)
```

## 5.1 Usar `torchinfo` para tener una idea de las formas que atraviesan nuestro modelo

Imprimir nuestro modelo con `print(model)` nos da una idea de lo que está pasando con nuestro modelo.

Y podemos imprimir las formas de nuestros datos a través del método `forward()`.

Sin embargo, una forma útil de obtener información de nuestro modelo es usar `torchinfo`.

`torchinfo` viene con un método `summary()` que toma un modelo PyTorch así como `input_shape` y devuelve lo que sucede cuando un tensor se mueve a través de su modelo.

**Nota:** Si está utilizando Google Colab, deberá instalar `torchinfo`.

```
In [35]: # Instalar torchinfo isi no esta disponible, importar si esta disponible
try:
    import torchinfo
except:
    !pip install torchinfo
    import torchinfo

from torchinfo import summary
summary(modelo, input_size=[1, 3, 64, 64]) # hacer un paso de prueba de un e
```

```
Out[35]: =====
=====
Layer (type:depth-idx)                   Output Shape              Param #
=====
=====
TinyVGG                                  [1, 2]                    --
├─Sequential: 1-1                        [1, 10, 32, 32]           --
│   └─Conv2d: 2-1                        [1, 10, 64, 64]           280
│       └─ReLU: 2-2                      [1, 10, 64, 64]           --
│           └─Conv2d: 2-3                 [1, 10, 64, 64]           910
│               └─ReLU: 2-4               [1, 10, 64, 64]           --
│                   └─MaxPool2d: 2-5      [1, 10, 32, 32]           --
├─Sequential: 1-2                        [1, 10, 16, 16]           --
│   └─Conv2d: 2-6                        [1, 10, 32, 32]           910
│       └─ReLU: 2-7                      [1, 10, 32, 32]           --
│           └─Conv2d: 2-8                 [1, 10, 32, 32]           910
│               └─ReLU: 2-9               [1, 10, 32, 32]           --
│                   └─MaxPool2d: 2-10     [1, 10, 16, 16]           --
├─Sequential: 1-3                        [1, 2]                    --
│   └─Flatten: 2-11                      [1, 2560]                 --
│       └─Linear: 2-12                   [1, 2]                    5,122
=====
=====
Total params: 8,132
Trainable params: 8,132
Non-trainable params: 0
Total mult-adds (M): 6.74
=====
=====
Input size (MB): 0.05
Forward/backward pass size (MB): 0.82
Params size (MB): 0.03
Estimated Total Size (MB): 0.90
=====
=====
```

## 5.2 Función para realizar paso de entrenamiento del modelo



```

In [83]: def paso_entrenamiento(modelo,
            dataloader,
            perdida_fn,
            optimizador):
    # Colocar modelo en modo entrenamiento
    modelo.train()

    # Inicializar valores de perdida y exactitud del entrenamiento
    perdida_entrenamiento, exactitud_entrenamiento = 0, 0

    # Iterar sobre los lotes del DataLoader
    for imagenes, etiquetas in dataloader:

        # Enviar datos al dispositivo
        imagenes, etiquetas = imagenes.to(dispositivo), etiquetas.to(dispositivo)

        # 1. Propagar hacia adelante los datos de entrenamiento usando el método forward
        salida = modelo(imagenes)

        # 2. Calcule la pérdida (qué tan diferentes son las predicciones de las etiquetas)
        perdida = perdida_fn(salida, etiquetas)
        perdida_entrenamiento += perdida.item()

        # 3. Colocar a cero los gradientes del optimizador
        optimizador.zero_grad()

        # 4. Propagación hacia atrás
        perdida.backward()

        # 5. Realizar paso de optimización
        optimizador.step()

        # Calcular y acumular la exactitud sobre todos los lotes
        clase_predicha = torch.argmax(torch.softmax(salida, dim=1), dim=1)
        exactitud_entrenamiento += (clase_predicha == etiquetas).sum().item()

    # Ajustar métricas para obtener un promedio de la perdida y la exactitud
    perdida_entrenamiento /= len(dataloader)
    exactitud_entrenamiento /= len(dataloader)

    return perdida_entrenamiento, exactitud_entrenamiento

```

## 5.3 Función para realizar paso de validación del modelo

```
In [84]: def paso_validacion(modelo,
        dataloader,
        perdida_fn):

    # Colocar el modelo en modo evaluación
    modelo.eval()

    # Inicializar valores de perdida y exactitud de la validación
    perdida_validacion, exactitud_validacion = 0, 0

    # Iniciar el manejador de contexto para inferencia
    with torch.inference_mode():
        # Iterar sobre los lotes del DataLoader
        for imagenes, etiquetas in dataloader:
            # Enviar datos al dispositivo
            imagenes, etiquetas = imagenes.to(dispositivo), etiquetas.to(dispositivo)

            # 1. Propagar hacia adelante los datos de entrenamiento usando el modelo
            salida = modelo(imagenes)

            # 2. Calcular y acumular la pérdida
            perdida = perdida_fn(salida, etiquetas)
            perdida_validacion += perdida.item()

            # 3. Calcular y acumular la exactitud
            clases_predicha = salida.argmax(dim=1)
            exactitud_validacion += ((clases_predicha == etiquetas).sum()).item()

    # Ajustar métricas para obtener un promedio de la perdida y la exactitud
    perdida_validacion /= len(dataloader)
    exactitud_validacion /= len(dataloader)

    return perdida_validacion, exactitud_validacion
```

## 5.4 Función para entrenar el modelo

```

In [85]: from tqdm.auto import tqdm

def entrenar(modelo: torch.nn.Module,
             cargador_entrenamiento: torch.utils.data.DataLoader,
             cargador_evaluacion: torch.utils.data.DataLoader,
             optimizador: torch.optim.Optimizer,
             perdida_fn: torch.nn.Module = nn.CrossEntropyLoss(),
             epocas: int = 5):

    # 2. Crear diccionario vacio para los resultados
    resultados = {"perdida_entrenamiento": [],
                  "exactitud_entrenamiento": [],
                  "perdida_evaluacion": [],
                  "exactitud_evaluacion": []
    }

    # 3. Iterar sobre los pasos de entrenamiento y prueba por un número de e
    for epoca in tqdm(range(epocas)):
        perdida_entrenamiento, exactitud_entrenamiento = paso_entrenamiento(

        perdida_evaluacion, exactitud_evaluacion = paso_validacion(modelo=modelo,
                                                                    dataloader=cargador_evaluacion,
                                                                    perdida_fn=perdida_fn)

    # 4. Imprimir que esta pasando
    print(
        f"Epoca: {epoca+1} | "
        f"perdida_entrenamiento: {perdida_entrenamiento:.4f} | "
        f"exactitud_entrenamiento: {exactitud_entrenamiento:.4f} | "
        f"perdida_evaluación: {perdida_evaluacion:.4f} | "
        f"exactitud_evaluación: {exactitud_evaluacion:.4f}"
    )

    # 5. Actualizar el diccionario de resultados
    resultados["perdida_entrenamiento"].append(perdida_entrenamiento)
    resultados["exactitud_entrenamiento"].append(exactitud_entrenamiento)
    resultados["perdida_evaluacion"].append(perdida_evaluacion)
    resultados["exactitud_evaluacion"].append(exactitud_evaluacion)

    # 6. Retornar el diccionario de resultados al final de cada epoca
    return resultados

```

## 5.5 Definir función de perdida y optimizador

```

In [86]: torch.manual_seed(42)
         perdida_fn = nn.CrossEntropyLoss()
         optimizador = optim.Adam(modelo.parameters(), lr=3e-4)

```

## 5.6 Lazo de entrenamiento.

```
In [87]: # Fijar el número de épocas
NUMERO_EPOCAS = 10

# Inicializar el temporizador
from timeit import default_timer as timer
tiempo_inicial = timer()

# Entrenar modelo
modelo_resultados = entrenar(modelo=modelo,
                              cargador_entrenamiento=cargador_entrenamiento,
                              cargador_evaluacion=cargador_prueba,
                              optimizador=optimizador,
                              perdida_fn=perdida_fn,
                              epocas=NUMERO_EPOCAS)

# Finalizar el temporizador e imprimir cuanto tardo el entrenamiento
tiempo_final = timer()
print(f"Tiempo total de entrenamiento: {tiempo_final-tiempo_inicial:.3f} seg

0%|          | 0/10 [00:00<?, ?it/s]
Epoca: 1 | perdida_entrenamiento: 0.4676 | exactitud_entrenamiento: 0.8100 |
perdida_evaluación: 0.5356 | exactitud_evaluación: 0.6842
Epoca: 2 | perdida_entrenamiento: 0.4393 | exactitud_entrenamiento: 0.7500 |
perdida_evaluación: 0.4557 | exactitud_evaluación: 0.8421
Epoca: 3 | perdida_entrenamiento: 0.4166 | exactitud_entrenamiento: 0.8100 |
perdida_evaluación: 0.4488 | exactitud_evaluación: 0.8421
Epoca: 4 | perdida_entrenamiento: 0.3783 | exactitud_entrenamiento: 0.8200 |
perdida_evaluación: 0.4502 | exactitud_evaluación: 0.7895
Epoca: 5 | perdida_entrenamiento: 0.3830 | exactitud_entrenamiento: 0.7700 |
perdida_evaluación: 0.4181 | exactitud_evaluación: 0.7895
Epoca: 6 | perdida_entrenamiento: 0.3369 | exactitud_entrenamiento: 0.8500 |
perdida_evaluación: 0.4961 | exactitud_evaluación: 0.7368
Epoca: 7 | perdida_entrenamiento: 0.3211 | exactitud_entrenamiento: 0.8900 |
perdida_evaluación: 0.4193 | exactitud_evaluación: 0.8421
Epoca: 8 | perdida_entrenamiento: 0.3050 | exactitud_entrenamiento: 0.8500 |
perdida_evaluación: 0.4155 | exactitud_evaluación: 0.8947
Epoca: 9 | perdida_entrenamiento: 0.3090 | exactitud_entrenamiento: 0.8900 |
perdida_evaluación: 0.4402 | exactitud_evaluación: 0.7895
Epoca: 10 | perdida_entrenamiento: 0.2746 | exactitud_entrenamiento: 0.8600
| perdida_evaluación: 0.3995 | exactitud_evaluación: 0.7895
Tiempo total de entrenamiento: 125.262 segundos
```

## 5.7 Función para graficar las curvas de perdida y exactitud

```
In [88]: import matplotlib.pyplot as plt

def graficar_curvas_perdida(resultados):
    """Graficar las curvas de entrenamiento del diccionario de resultados.

    Argumentos:
        resultados (dict): diccionario conteniendo una lista de valores cont
        {"perdida_entrenamiento": [...],
         "exactitud_entrenamiento": [...],
         "perdida_evaluacion": [...],
         "exactitud_evaluacion": [...]}
    """

    # Obtener los valores de la perdida del diccionario de resultados (entre
    perdida = resultados['perdida_entrenamiento']
    perdida_validacion = resultados['perdida_evaluacion']

    # Obtener los valores de la exactitud del diccionario de resultados (ent
    exactitud = resultados['exactitud_entrenamiento']
    exactitud_validacion = resultados['exactitud_evaluacion']

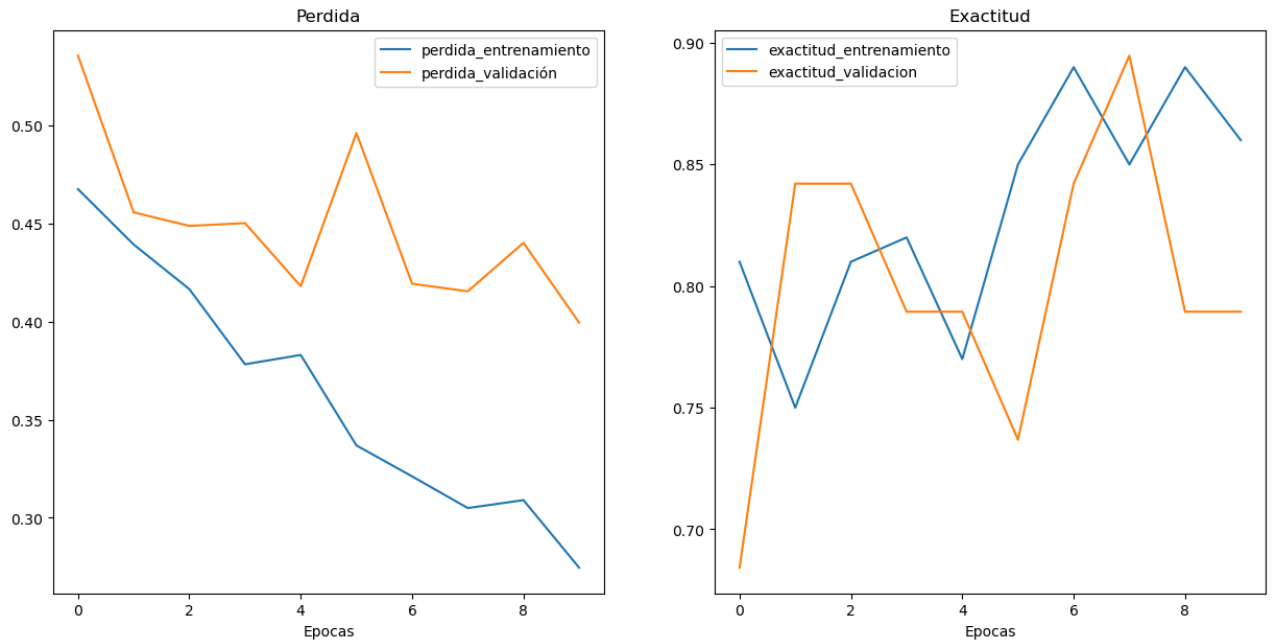
    # Determinar el número de épocas
    epocas = range(len(resultados['perdida_entrenamiento']))

    # Configurar la gráfica
    plt.figure(figsize=(15, 7))

    # Graficar la perdida
    plt.subplot(1, 2, 1)
    plt.plot(epocas, perdida, label='perdida_entrenamiento')
    plt.plot(epocas, perdida_validacion, label='perdida_validación')
    plt.title('Perdida')
    plt.xlabel('Epocas')
    plt.legend()

    # Graficar la exactitud
    plt.subplot(1, 2, 2)
    plt.plot(epocas, exactitud, label='exactitud_entrenamiento')
    plt.plot(epocas, exactitud_validacion, label='exactitud_validacion')
    plt.title('Exactitud')
    plt.xlabel('Epocas')
    plt.legend();
```

```
In [89]: graficar_curvas_perdida(modelo_resultados)
```



## 6. Guardar el modelo

```
In [16]: torch.save(modelo_1.state_dict(), '../modelos/hormiguero_perezoso')
```

```
In [ ]:
```