



IMAGO-ARG. IO

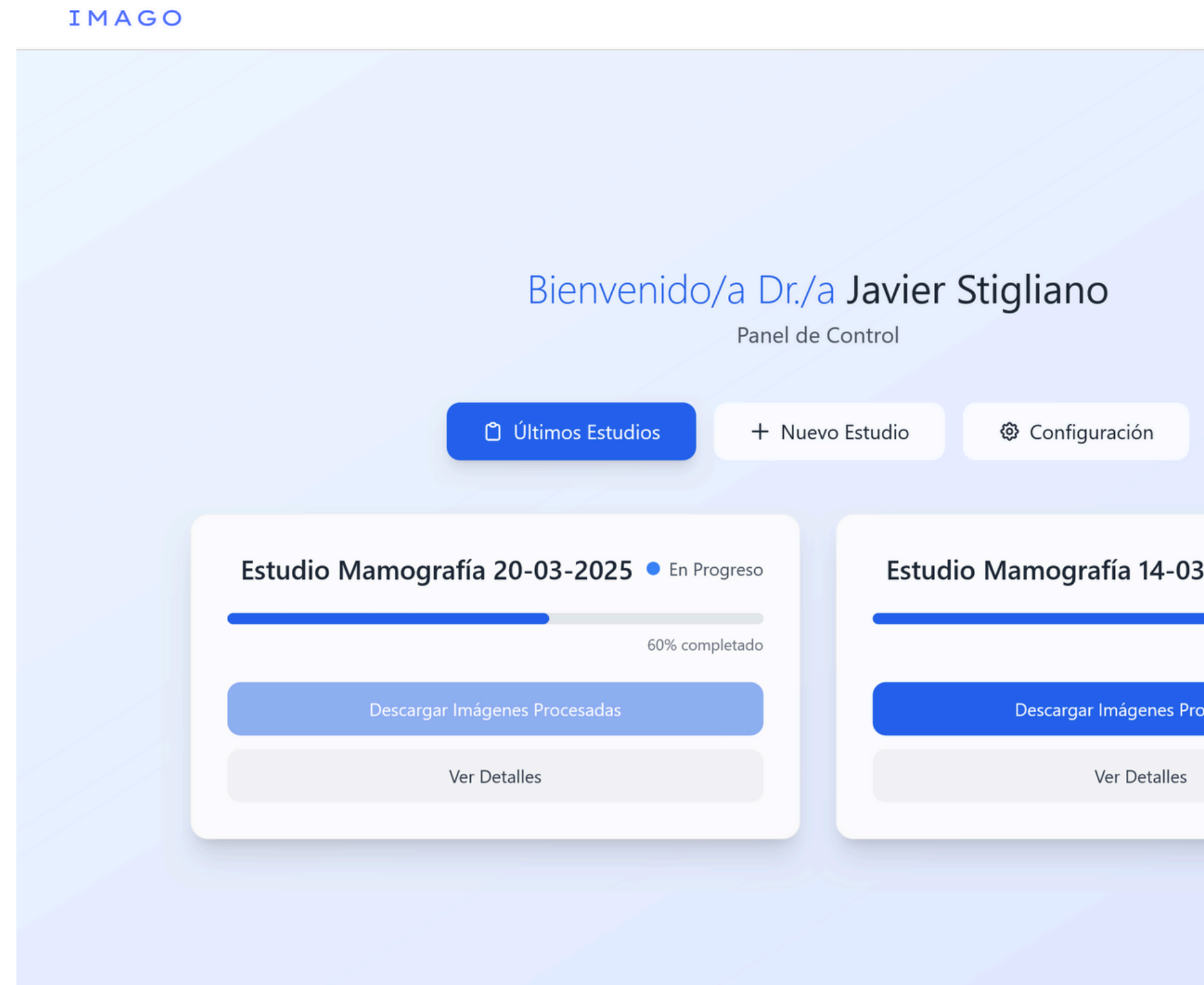
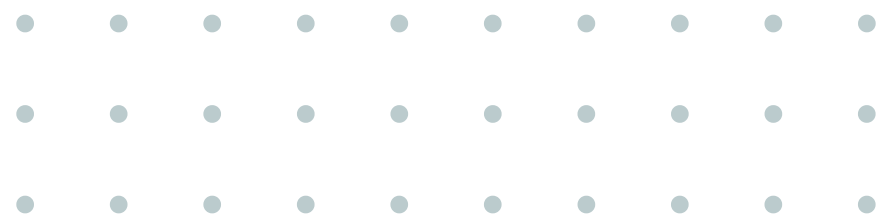
WEB INTERFACE FOR MANAGING
MAMMOGRAMS PROCESSED WITH
IMAGO ICE REVEAL



The Product

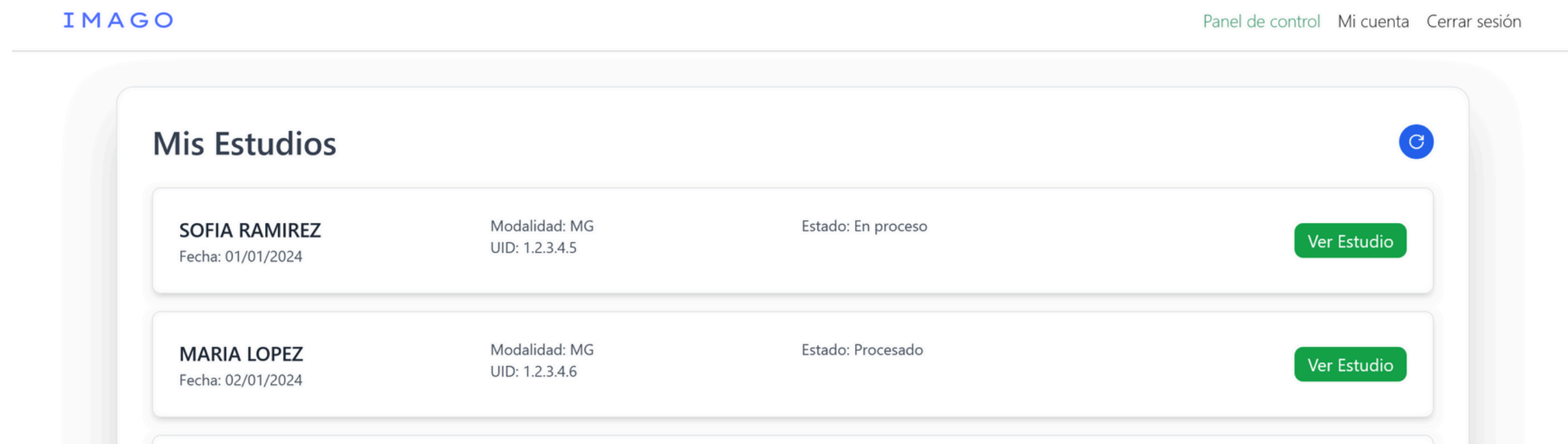
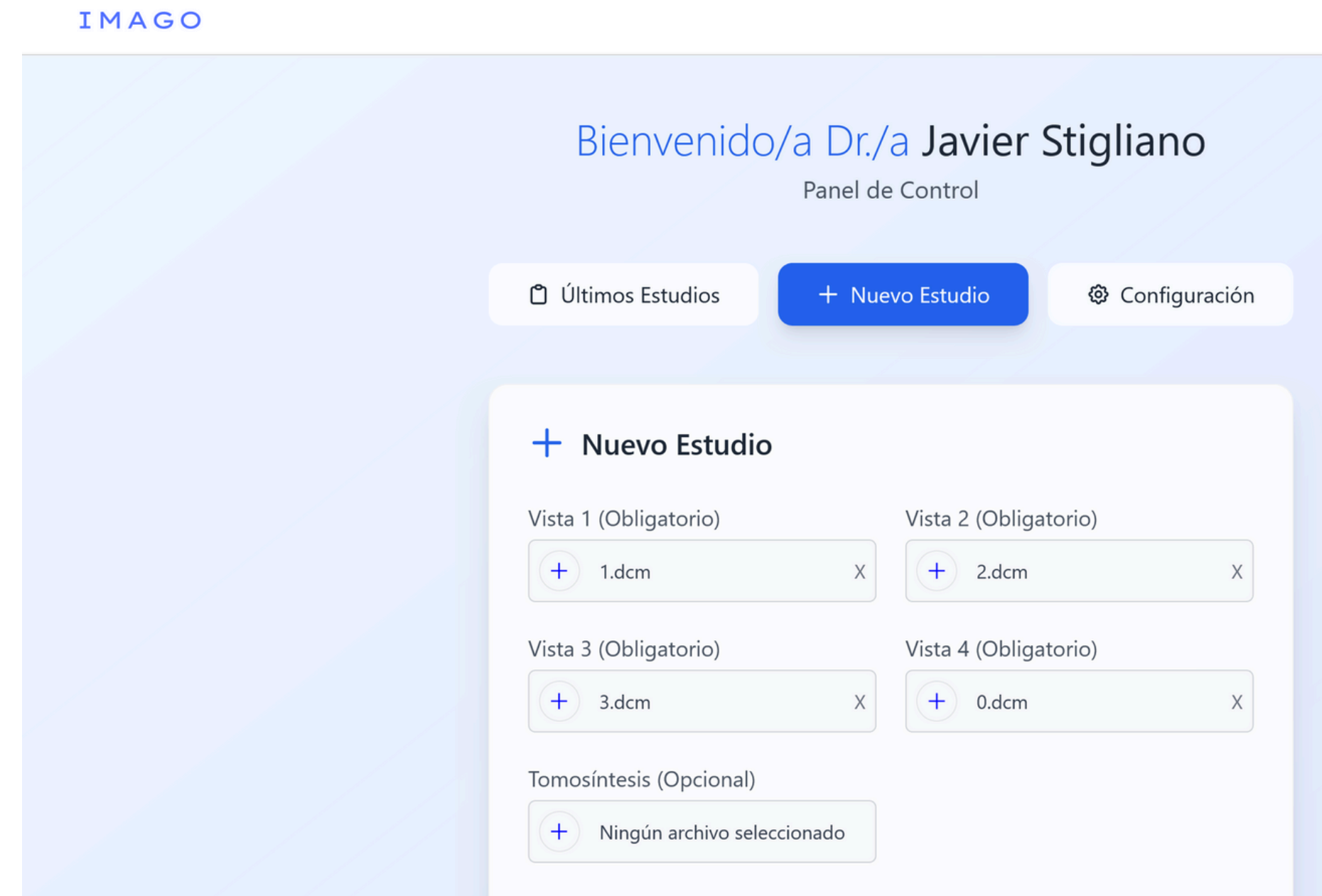
IMAGO-ARG.io is an intuitive web interface specifically designed for healthcare professionals, enabling the uploading, automatic processing, reporting, and downloading of digital mammograms processed using the groundbreaking Imago ICE Reveal* technology.

*FDA 510K on-going



Data Workflow


- Easy Upload: User-friendly interface allowing secure and rapid image uploads.
- Efficient Management: Clear visualization of processing status.
- Secure Download: Generation of encrypted download links ensuring secure access.



Authentication and Authorization

- Secure multi-factor authentication system powered by Firebase.
- Granular and centralized user administration with role-based permissions.

IMAGO



Imago Web Interface

Home

Settings

Users

Roles

Groups

Security

Integrations

Help



Authentication

Usuarios | Método de acceso | Plantillas | Uso | Configuración | Extensions

Las siguientes funciones de autenticación dejarán de estar disponibles cuando se cierre Firebase Dynamic Links el 1 de 2025: la autenticación a través de vínculos de correo electrónico en apps para dispositivos móviles, así como la autenticación con OAuth de Cordova para apps web.

Buscar por dirección de correo electrónico, número de teléfono o UID de usuario

Agregar usuario

Identificador	Proveedores	Fecha de creación	Fecha de acceso	UID de usuario
javier@mail.com		24 mar 2025	24 mar 2025	111111111111111111
guarinolucia@icloud.co...		23 mar 2025	24 mar 2025	111111111111111111

Filas por página: 50 1 - 2 of 2



Acceso Restringido

Por razones de seguridad y confidencialidad médica, esta sección requiere credenciales de acceso específicas.

Si cree que esto es un error, por favor contacte al administrador del sistema o al departamento de soporte técnico.

Volver al Panel Principal

Contactar Soporte

Cuenta en proceso de verificación

Tu cuenta está en proceso de verificación. La confirmación de la misma puede tardar hasta 48 horas.

IMAGO

Control and Administration

- New user registrations exclusively managed by administrators.
- Mandatory verification and authorization.

Registro

Título

Selecione...

▼

Nombre

Apellido

Nombre del profesional

Apellido del profesional

DNI

Domicilio

Documento de identidad

Domicilio

Centro de Salud

Centro de salud donde desarrolla su actividad


Correo electrónico

correo@ejemplo.com

Registrar

IMAGO


[Panel de control](#) [Panel de administración](#) [Mi cuenta](#) [Cerrar sesión](#)

 **Panel de Administración**

Bienvenido, Lucia Guarino


Usuarios Totales

2




Usuarios Aprobados

1



Pendientes de Aprobación

1



Gestión de Usuarios

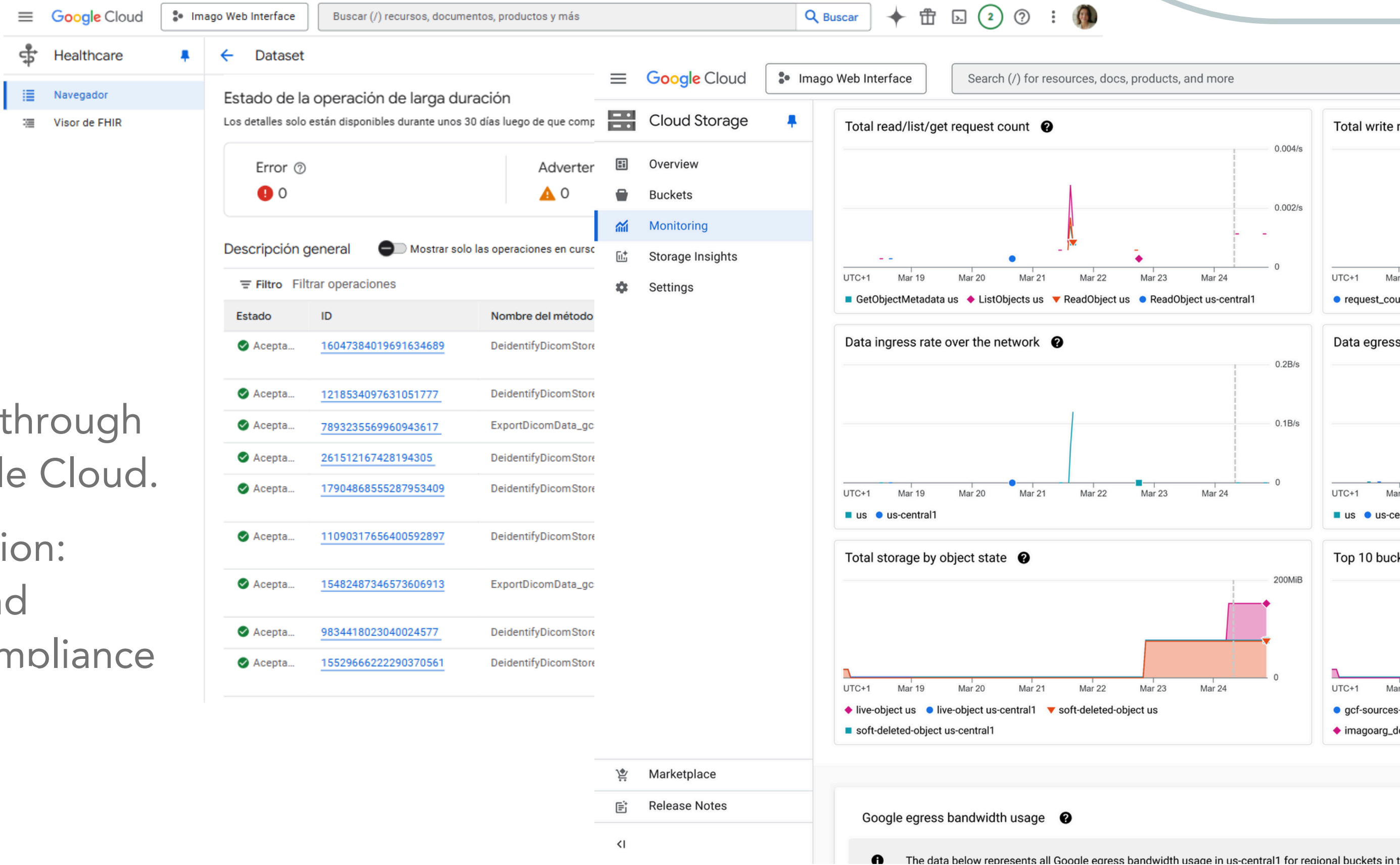
USUARIO	DNI	EMAIL	ESTADO	ACCIONES
Dr. Javier Stigliano Hospital Posadas	12345678	javier@mail.com	Pendiente	Aprobar Restringir
Lucia Guarino Administrador	12345678	guarinolucia@icloud.com	Aprobado	

Registrar Usuario

- Automated email notifications.
- Comprehensive security and traceability of platform access and permissions.

Integration with Google Cloud and Healthcare API

- Robust protection and simplified compliance through integration with Google Cloud.
- Automatic anonymization: Guaranteed privacy and ensured regulatory compliance (GDPR/HIPAA).



Robustness from the First Line of Code

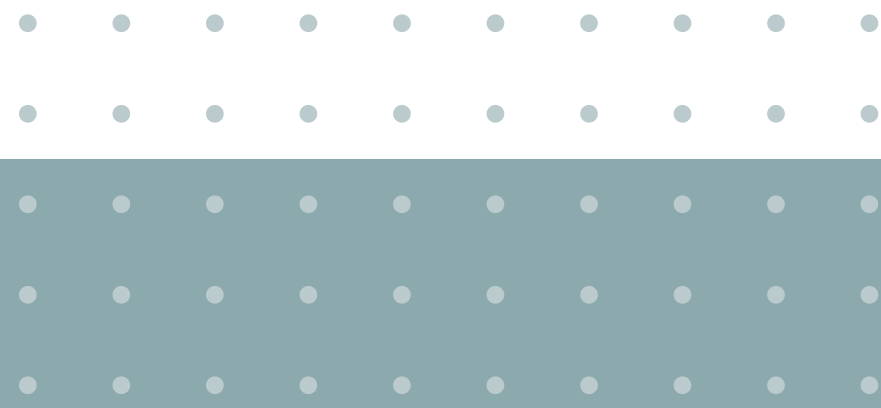


- Automatic Backups: Continuous protection and immediate recovery capabilities.
- Performance and Scalability: Modular, robust, and integrated architecture.
- Continuous Auditing: Constant monitoring and complete traceability.
- End-to-End Encryption: Maximum security during data transfer and storage.

```
dicom_service.py X
dicom-utils > dicom_service.py > ...
1 from flask import Flask, request, send_file, jsonify
2 import pydicom
3 from pydicom.dataset import Dataset
4 from io import BytesIO
5 import json
6 import os
7
8 app = Flask(__name__)
9 app.config['MAX_CONTENT_LENGTH'] = 100 * 1024 * 1024
10
11 def load_custom_tag_config():
12     config_path = os.path.join(os.path.dirname(os.path.dirname(__file__)), 'metadata_files', 'cus
13     try:
14         with open(config_path, 'r') as f:
15             return json.load(f)
16     except Exception as e:
17         print(f"Error loading custom tag config: {e}")
18         return {"customTagConfig": {"tags": []}}
19
20 def apply_custom_tags(ds, doctor_uid):
21     # Usar el tag estándar DICOM para Referring Physician Name
22     ds.add_new(0x00080090, 'PN', doctor_uid)
23     print(f"Added Referring Physician Name tag with value: {doctor_uid}")
24
25     # Verificar que el tag fue agregado
26     if 0x00080090 in ds:
27         print(f"Verified Referring Physician Name tag exists with value: {ds[0x00080090].value}")
28     else:
29         print("Warning: Referring Physician Name tag was not added successfully")
30
31 def extract_metadata(ds):
32     metadata = {}
33     fields = [
34         'SOPInstanceUID', 'SeriesInstanceUID', 'StudyInstanceUID',
35         'PatientID', 'PatientName', 'StudyDate', 'Modality',
36         'ImageLaterality', 'ViewPosition'
37     ]
38     for field in fields:
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

```
info: [object Object] {"timestamp": "2025-03-24T19:41:53.936Z"}
info: [object Object] {"timestamp": "2025-03-24T19:41:54.288Z"}
info: [object Object] {"timestamp": "2025-03-24T19:41:54.334Z"}
```



Thanks!

