

16.22 Informática Médica



Trabajo Práctico
N°6

Grupo 12

Integrantes:

De la Llave, Tomas

Erbin, Lola

Legajo:

61670

62417

Fecha de entrega:
18 de Junio de 2025

Actividad 1

Se creó un recurso de tipo *Patient* en el servidor HAPI FHIR, a partir del ejemplo “Example Patient/example (JSON)” provisto por HL7 FHIR. Se personalizó el recurso modificando el identificador de la organización responsable.

Recurso creado correctamente:

[illegible]

Para acceder al recurso creado, se utilizó el identificador "id": 47884583.

Read an individual resource instance given its ID (and optionally a version ID to retrieve a specific version of that instance to **vread** that instance)

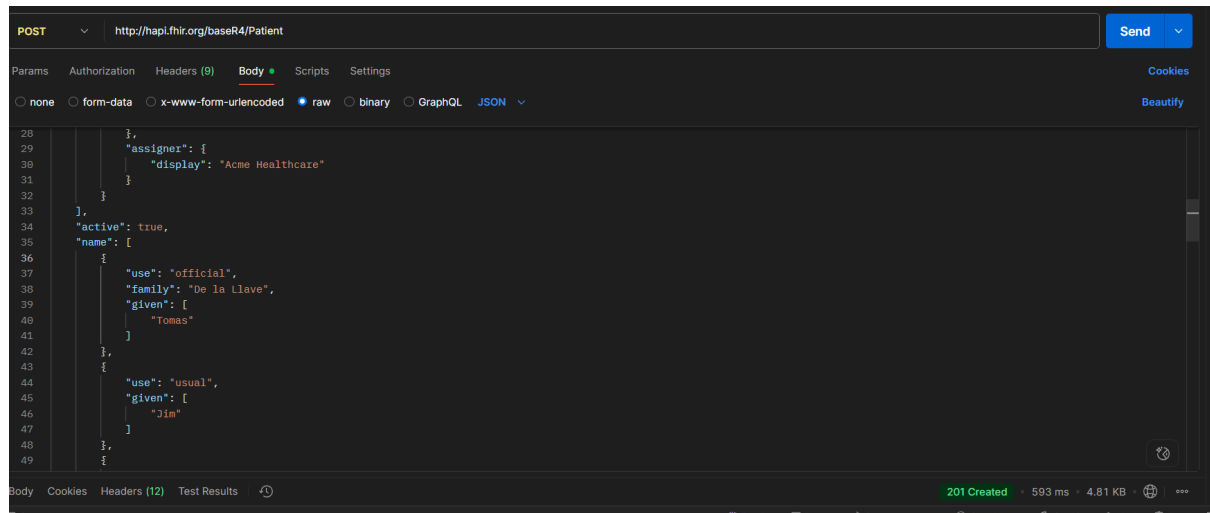
 Read ID* 47884583 Version ID (add for vread)

El recurso Patient creado incluye las siguientes propiedades:

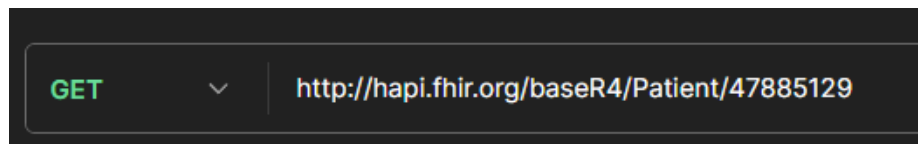
- id: Identificador del recurso en el sistema.
 - "example".
- identifier: Identificador del paciente asignado por una organización.
- active: Indica que el registro del paciente está activo.
 - "true"
- name: Contiene tres nombres registrados.
 - Oficial: Peter James Chalmers
 - Usual: Jim
 - De nacimiento (maiden): Peter James Windsor (válido hasta 2002)
- telecom: Indica las posibles formas de contacto:
 - Teléfonos laboral, móvil y uno antiguo con periodo de uso hasta 2014.
- gender: Indica el género
 - "male"
- birthDate: Indica la fecha de nacimiento.
 - "1974-12-25"
- deceasedBoolean: Indica que el paciente se encuentra con vida al momento del registro
 - "false"
- address: Indica la dirección
 - "Domicilio en 534 Erewhon St, PleasantVille, Rainbow, Vic 3999, uso "home""
- contact: Indica el contacto de emergencia:
 - Nombre: Bénédicte du Marché, mujer
 - Relación: código "N" (pariente)
 - Teléfono y dirección coincidente con la del paciente
 - Periodo de contacto: desde 2012 en adelante
- managingOrganization: Hace referencia a la organización que gestiona al paciente.
 - Servicio de gastroenterología.

Actividad 2

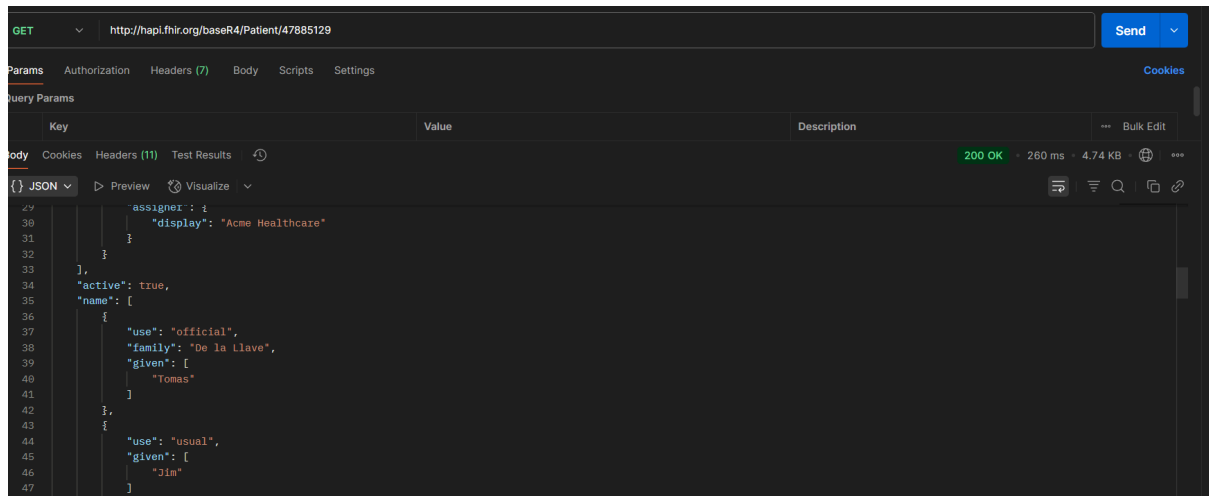
En esta etapa se replicó la creación del recurso Patient utilizando la herramienta Postman. Se modificó el campo "name" para personalizar el recurso y permitir su posterior reconocimiento con el nombre “Tomas de la Llave”.



El servidor devuelve como identificador el ID: 47885129, el cual se utilizó luego para leer el recurso mediante una solicitud GET.

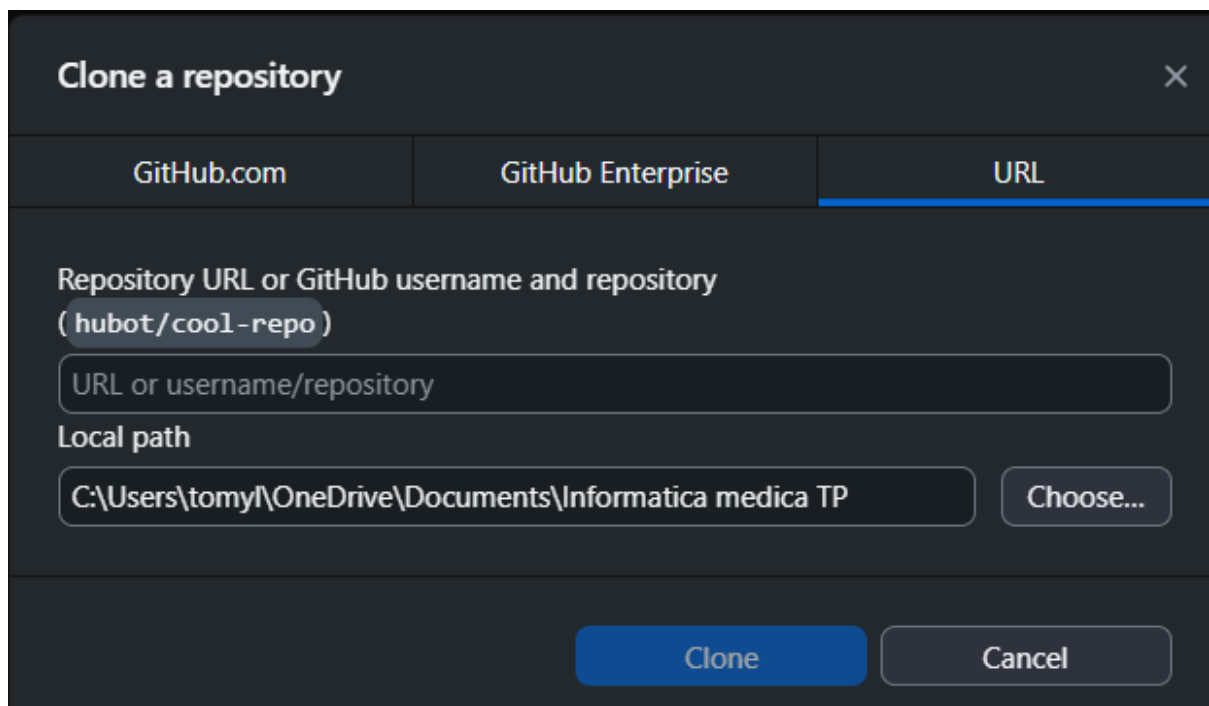


Se puede observar en la siguiente imagen que devuelve el paciente creado previamente, es decir que, en la respuesta del servidor se confirmó la correcta creación del recurso.

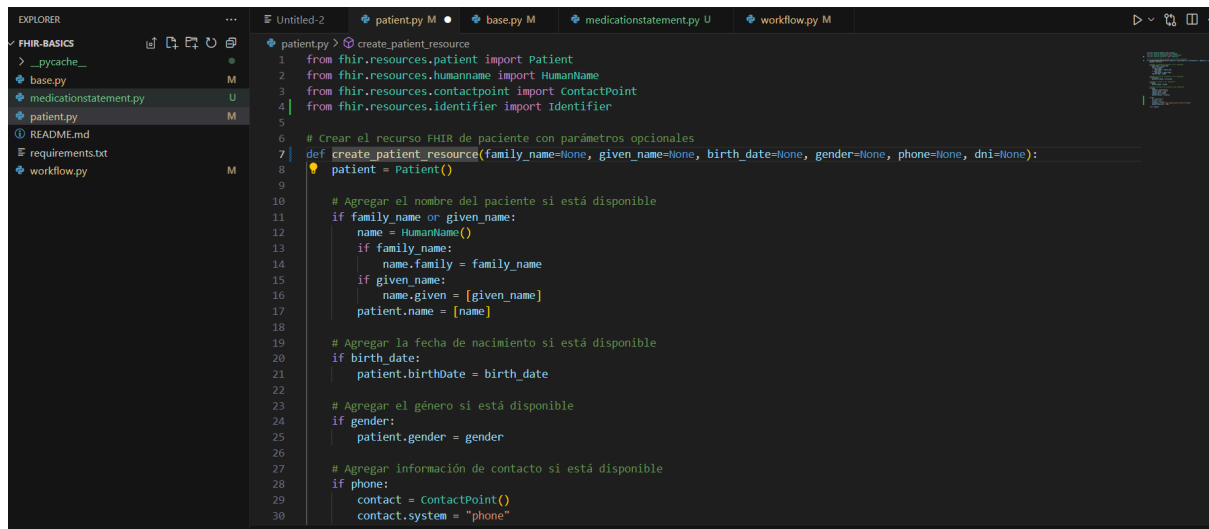


Actividad 3

a) Se clonó el repositorio utilizando la aplicación “github desktop”



De esta forma se clonó en un entorno local en Visual Studio Code.



Se importó el Identifier en el archivo [Patient.py](#).

```
from fhir.resources.identifier import Identifier
```

Se modificó la función `create_patient_resource`, para incluir un campo "identifier" con el número de documento del paciente, al agregar las siguientes líneas de código:

```
if dni:
    id = Identifier()
    id.value = dni
    id.use = "official"
    id.system = "https://www.argentina.gob.ar/interior/renaper"
    patient.identifier = [id]
```

Este fragmento de código define un identificador oficial para el paciente utilizando su DNI. Si se proporciona un valor de DNI, se crea un objeto Identifier al que se le asigna el número de documento, se establece su uso como "official" y se especifica que proviene del sistema RENAPER de Argentina mediante una URL. Este identificador se asocia luego al recurso Patient, permitiendo que dicho paciente pueda ser buscado en el servidor FHIR a través de su DNI.

Con la función modificada creamos y enviamos a happy fhir el recurso paciente creado, utilizando las funciones otorgadas por la cátedra.

```
# Crear y enviar el recurso de paciente
paciente = create_patient_resource("De la Llave", "Tomas", "2002-04-22", "male", None, "43992337")
patient_id = send_resource_to_hapi_fhir(paciente, 'Patient')
```

Se puede visualizar y obtener el recurso creado de la siguiente manera:

```
# Ver el recurso de paciente creado
if patient_id:
    get_resource_from_hapi_fhir(patient_id, 'Patient')
```

b) Se modificó la función `get_resource_from_hapi_fhir` para que devuelva el DNI. Se agregó una búsqueda dentro del campo `identifier` del recurso, filtrando por el sistema correspondiente a RENAPER. Si se encuentra un identificador con ese sistema, se extrae su valor y se retorna como el DNI del paciente.

```
# Buscar el recurso por ID
def get_resource_from_hapi_fhir(resource_id, resource_type):
    url = f"http://hapi.fhir.org/baseR4/{resource_type}/{resource_id}"
    response = requests.get(url, headers={"Accept": "application/fhir+json"})

    if response.status_code == 200:
        resource = response.json()
        print(resource)

        # Buscar el DNI en el campo identifier
        identifiers = resource.get("identifier", [])
        for identifier in identifiers:
            if identifier.get("system") == "https://www.argentina.gob.ar/interior/renaper":
                dni = identifier.get("value")
                print(f"DNI del paciente: {dni}")
                return dni # Retorna el DNI para usarlo en otras funciones

        print("DNI no encontrado en los identificadores.")
        return None

    else:
        print(f"Error al obtener el recurso: {response.status_code}")
        print(response.json())
```

Se creó la siguiente función para buscar pacientes por dni. Esta función permite buscar un paciente en el servidor HAPI FHIR utilizando su DNI como identificador. Realiza una consulta GET filtrando por el sistema RENAPER y, si encuentra coincidencias, devuelve el recurso Patient correspondiente.

```
# Buscar paciente por DNI
def search_patient_by_dni(dni):
    url = "http://hapi.fhir.org/baseR4/Patient"
    params = {
        "identifier": f"https://www.argentina.gob.ar/interior/renaper|{dni}"
    }
    response = requests.get(url, params=params, headers={"Accept": "application/fhir+json"})

    if response.status_code == 200:
        bundle = response.json()
        entries = bundle.get("entry", [])
        if not entries:
            print(f"No se encontró ningún paciente con DNI {dni}.")
            return None

        # Se encontró al menos un paciente
        patient = entries[0]["resource"]
        print(f"Paciente encontrado: {patient.get('name', [{}])[0].get('given', [{}])[0]} {patient.get('name', [{}])[0].get('family')}")
        return patient
    else:
        print(f"Error al buscar el paciente: {response.status_code}")
        print(response.json())
        return None
```

En la siguiente imagen se observa que se logra encontrar el recurso del paciente creado previamente.

```
21 |         get_resource_from_hapi_fhir(patient_id, patient)
22 |
23 |
24 | # Busco paciente por dni
25 | print(f"\n Buscamos paciente por dni \n")
26 | search_patient_by_dni(43992337)
27 |
28 |
29 |
30 |
31 | # Ahora, crear un recurso MedicationStatement asociado al paciente
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

DNI del paciente: 43992337

Buscamos paciente por dni

Paciente encontrado: Tomas De la Llave

c) Se creó la función `create_medication_statement`. Esta función crea un recurso `MedicationStatement` en FHIR, asociando un medicamento a un paciente previamente creado mediante su identificador. Permite registrar el código y nombre del medicamento, el estado del tratamiento (por defecto activo), la fecha en que fue administrado y una nota opcional. La estructura generada cumple con los estándares FHIR y puede ser enviada al servidor para su almacenamiento o consulta.


```

from fhir.resources.medicationstatement import MedicationStatement
from fhir.resources.codeableconcept import CodeableConcept
from fhir.resources.coding import Coding
from fhir.resources.reference import Reference
from fhir.resources.annotation import Annotation

def create_medication_statement(patient_id, medication_code, medication_display, status="active", effective_date=None, note_text=None):
    med_statement = MedicationStatement(
        status=status,
        medicationCodeableConcept=CodeableConcept(
            coding=[Coding(
                system="http://snomed.info/sct"
            )],
            code=medication_code,
            display=medication_display
        ),
        subject=Reference(reference=f"Patient/{patient_id}")
    )

    if effective_date:
        med_statement.effectiveDateTime = effective_date

    if note_text:
        med_statement.note = [Annotation(text=note_text)]

    return med_statement

```

Enviamos el recurso a hapi fhir y luego lo recuperamos.

```

# Actividad 3.c

# Ahora, crear un recurso MedicationStatement asociado al paciente
med_statement = create_medication_statement(
    patient_id=patient_id,
    medication_code="322236009", # código snomed Paracetamol
    medication_display="Paracetamol 500mg tableta",
    status="active",
    effective_date="2024-06-01",
    note_text="Paciente refiere automedicación ocasional por dolor."
)

# Enviar el recurso MedicationStatement
med_statement_id = send_resource_to_hapi_fhir(med_statement, 'MedicationStatement')

if med_statement_id:
    print(f"MedicationStatement creado con ID: {med_statement_id}")
    get_resource_from_hapi_fhir(med_statement_id, 'MedicationStatement')
else:
    print("No se pudo crear el paciente.")

```

Resultados del recurso medication statement creado.

```

Recurso creado exitosamente
MedicationStatement creado con ID: 47885563
{'resourceType': 'MedicationStatement', 'id': '47885563', 'meta': {'versionId': '1', 'lastUpdated': '2025-06-13T22:41:45.274+00:00', 'source': '#ZPeqPIkPrjFPWcYZ'}, 'status': 'active', 'medicationCodeableConcept': {'coding': [{'system': 'http://www.nlm.nih.gov/research/umls/rxnorm', 'code': '860975', 'display': 'Paracetamol 500mg tablet'}]}, 'subject': {'reference': 'Patient/47885562'}, 'effectiveDateTime': '2024-06-01', 'note': [{'text': 'Paciente refiere automedicación ocasional por dolor.'}]}
DNI no encontrado en los identificadores.
PS C:\Users\tomy\OneDrive\Documents\Informatica medica TP\FHIR-basics>

```

MedicationStatement se relaciona con Patient mediante una referencia directa que indica a quién pertenece el registro del medicamento.

Los codigos se cargaron en el repositorio:

https://github.com/tomasdelak/TP_6_INFO_MED.git