

# Medical Sapiens Rest API

## Métodos

Lista de métodos que ofrece el API para integrarse con Medical Sapiens

- LoginUser ..... 2
- Practitioner/List ..... 3
- Patient/Add..... 4
- Patient/List..... 5
- Episode/Add..... 6
- Patient/PreConsult ..... 7
- Episode/List..... 8
- Episode/Get..... 9
- Episode/Analyze ..... 10
- Códigos de Error ..... 11
- Llaves privada y pública..... 11

---

## LoginUser

Inicia una sesión en Medical Sapiens para un usuario del API e institución registrada. El código de la institución, el usuario del API y su clave son entregados por Medical Sapiens.

URI: /api\_v1/User/Login

Method: Post

Request: (JSon)

```
{
  "CodInstit": "<código de Institución>",
  "User": "<usuario del API>",
  "Pasw": "<clave del usuario>",
  "Sign64": "<firma en base 64>"
}
```

Para la firma en base 64 se debe generar el string “[<código de Institución>|<usuario>|<clave>]”, luego firmarlo con la llave privada (pem) de la institución y codificarla en base 64. La llave pública debe estar en poder de Medical Sapiens para poder hacer la verificación.

Response: (JSon)

```
{
  "Erc": "<código de error>",
  "ErcText": "<descripción del error>",
  "User": "<usuario del API>",
  "SessionID": "<token a ser usado con los otros métodos>"
}
```

El token tiene una duración de 4 horas y debe ser usado en los otros métodos. Este token restringe el uso de los métodos a la institución indicada en el Request..

La lista de códigos de error están más adelante.

Para generar la firma (Sign64) puede usar la siguiente página. Esta página no almacena ninguno de los parámetros que pide para generar la firma.

[www.medicalsapiens.com/sapiens/GenSign.php](http://www.medicalsapiens.com/sapiens/GenSign.php)

---

## Practitioner/List

Permite obtener la lista de los usuarios asociados a la institución. Si se especifica el usuario, sólo trae sus datos.

URI: /api\_v1/Practitioner/List

Method: Get

Parameters:

- User: <usuario del API>
- SessionID: <token devuelto por LoginUser>
- PractitionerUser: <Usuario del médico> (*opcional, si no se especifica los obtiene todos*)

Response: (JSon)

```
{
  "Erc":          <código de error>,
  "ErcText":     "<descripción del error>",
  "List" : [ {
    "identifier ": "<Id del doctor>",
    "user" :      "<usuario>",
    "name" :      "<nombre>",
    "email":      "<email>",
    "DocumentID" : "<RUT del médico>"
  } ... ]
}
```

---

## Patient/Add

Permite agregar un paciente en Medical Sapiens. El identificador del paciente debe ser único dentro de la institución. Se podría usar el RUT del paciente.

URI: /api\_v1/Patient/Add

Method: Post

Request: (JSoN)

```
{
  "User":      "<usuario del API>",
  "SessionID": "<token devuelto por LoginUser>",
  "Patient" : {
    "country":  "<código país>",           // CL: Chile, ISO 3166-1
    "identifier": "Identificador del paciente", // char(20), ej: 11222333-8
    "identifierType": "1: RUT, 2: otro ID",
    "gender":   "<género: male o female>"
    "birthdate": "<yyyy-mm-dd>",
    "name" : { // optional
      "family":   "<apellido paterno>",
      "given":    "<nombres>",
      "lastSurname" : "<apellido materno>"
    }
  }
}
```

Response: (JSoN)

```
{
  "Erc":      <código de error>,
  "ErcText":  "<descripción del error>"
}
```

---

## Patient/List

Permite obtener la lista de pacientes. Si se especifica el identificador del paciente, sólo trae sus datos.

URI: /api\_v1/Patient/List

Method: Get

Parameters:

- User: <usuario del API>
- SessionID: <token devuelto por LoginUser>
- identifier: <Id del paciente> (*opcional*)

Response: (JSon)

```
{
  "Erc": <código de error>,
  "ErcText": "<descripción del error>",
  "List": [ {
    "identifier": "<Identificador del paciente>",
    "identifierType": "<tipo del Id del paciente>",
    "country": "<código del país>",
    "gender": "male | female"
    "birthdate": "yyyy-mm-dd",
    "name": {
      "family": "<apellido paterno>",
      "given": "<nombres>",
      "lastSurname": "<apellido materno>"
    }
    "creationdate": "<fecha de creación del paciente>"
  }, ...]
}
```

---

## Episode/Add

Permite generar una URL que puede ser utilizada para abrir Médica Sapiens y permitir que el médico pueda agregar los síntomas y signos del paciente

URI: /api\_v1/Episode/Add

Method: Post

Request: (JSON)

```
{
  "User":           "<usuario del API>",
  "SessionID":     "<token devuelto por LoginUser>",
  "Episode" : {
    "PractitionerUser": "<usuario del médico>",
    "PatientId":       "<Identificador del paciente>",
    "EpisodeToken":   "<token para ser usado en GetEpisode>" opcional char(35), ej:
                      cód consulta
  }
}
```

Response: (JSON)

```
{
  "Erc":           <código de error>,
  "ErcText":       "<descripción del error>",
  "EpisodeUrl":    "<URL para invocar a MS y que el médico pueda ingresar datos>"
}
```

El episodio se crea cuando el médico graba algún dato.

El EpisodeToken debe ser único a nivel de paciente.

---

## Patient/PreConsult

Permite generar una URL que puede ser enviada al paciente para que pueda ingresar en la Pre Consulta y responder algunas preguntas.

El médico al ingresar a Médical Sapiens encontrará un episodio bajo este paciente con el motivo "PreConsulta-n".

URI: /api\_v1/Patient/PreConsult

Method: Post

Request: (JSON)

```
{
  "User":           "<usuario del API>",
  "SessionID":     "<token devuelto por LoginUser>",
  "PractitionerUser": "<usuario del médico>",
  "PatientId":     "<Identificador del paciente>",
  "EpisodeToken":  "<token ser usado en GetEpisode>" opcional char(35), ej: código consulta
}
```

Response: (JSON)

```
{
  "Erc":           <código de error>,
  "ErcText":       "<descripción del error>",
  "PreConsultUrl" : "<URL para que el paciente pueda invocar la Pre Consulta>",
  "iFrameUrl" :    "<URL para incluir la Pre Consulta en un iframe>"
}
```

ejemplo para agregar la Url en un <iframe>

```
<iframe src="...iFrameUrl..." title="Pre Consulta" height="720" width="50%" style="border:0;" allowfullscreen="" loading="lazy"></iframe>
```

El episodio de la preconsulta se crea en el momento que el paciente invoca a la url.

El EpisodeToken debe ser único a nivel de paciente.

Cuando el paciente presiona el botón Finalizar, se invoca la función FinalizarConsulta que debe estar en el padre del iframe. Esta función recibe el EpisodeId y el EpisodeToken, de modo que el que invoca al iframe de la PreConsulta pueda hacer lo que estime conveniente.

```
<script>
function FinalizarConsulta(EpisodeId, EpisodeToken)
{
  window.location.href = "https://www.medicalsapiens.com";
}
</script>
```

---

## Episode/List

Permite obtener la lista de los episodios de un paciente. Se puede filtrar por la fecha y/o el médico.

URI: /api\_v1/Episode/List

Method: Get

Parameters:

- User: <usuario del API>
- SessionID: <token devuelto por LoginUser>
- PatientId : <Id del paciente>
- Date : <fecha del episodio yyyy-mm-dd> *(opcional)*
- PractitionerUser: <usuario del médico> *(opcional)*

Response: (JSon)

```
{
  "Erc": <código de error>,
  "ErcText": "<descripción del error>",
  "List" : [ {
    "EpisodeId" : "<identificador del episodio>",
    "date" : "<fecha del episodio yyyy-mm-dd hh:mm:ss>",
    "reason": "<motivo de la consulta>",
    "PractitionerUser": "<usuario del médico>",
    "EpisodeToken": "<token del episodio>",
    "EpisodeUrl": "<URL para invocar a MS y ver los datos>"
  }, ...]
}
```



## Episode/Get

Permite obtener los datos de un episodio. Si el médico seleccionó la enfermedad que cree es la correcta, la respuesta incluye su nombre y sus códigos Cie10 y Snomed.

Se debe indicar el EpisodeToken o el Episodeld.

URI: /api\_v1/Episode/Get

Method: Get

Parameters:

- User: <usuario del API>
- SessionID: <token devuelto por LoginUser>
- PractitionerUser: <usuario del médico>
- PatientId: <Identificador del paciente>,
- EpisodeToken: <token enviado en AddEpisode>
- Episodeld: <identificador del episodio>

Response: (JSON)

```
{
  "Erc": <código de error>,
  "ErcText": "<descripción del error>",
  "Episode" : {
    "Episodeld" : <número identificador del episodio>,
    "EpisodeToken": <token del episodio>,
    "date" : <fecha del episodio yyyy-mm-dd hh:mm:ss>,
    "reason": <motivo de la consulta>,
    "diagnosis": <diagnóstico>,
    "disease": <enfermedad seleccionada por el médico en MS>,
    "cie10": <código Cie10 de la enfermedad seleccionada>.
    "snomed": <código Snamed de la enfermedad seleccionada>,
    "observations": <observaciones>,
    "anamnesis": <texto de la anamnesis próxima>,
    "physical_exam": <texto del examen físico>,
    "Weight": Peso en kg.
    "Height": Estatura en cm.
    "BMI": IMC
    "PreSystolic": Presión Sistólica
    "PreDiastolic": Presión Diastólica
    "Pressure": Presión
    "HeartRate": Frecuencia cardíaca
    "BreathFreq": Frecuencia respiratoria
    "history": <texto de la anamnesis remota>,
    "EpisodeUrl": <URL para invocar a MS y ver los datos>
  }
}
```

Si ingresa el mismo médico que creó el episodio antes de 24 horas de la creación del episodio, puede modificar los datos del episodio

## Episode/Analyze

Ejecuta el cálculo para el episodio y devuelve las posibles enfermedades que podría tener el paciente, de acuerdo a la información ingresada. El porcentaje indica la cercanía del episodio con la enfermedad. Si sólo se dispone de los datos de la preconsulta, los porcentajes serán bajos porque sólo tiene parte de los síntomas y faltarían los signos. Además se entrega el código CIE10 índice ESI y las especialidades asociadas a cada enfermedad.

Se debe indicar el EpisodeToken o el Episodeld.

URI: /api\_v1/Episode/Analyze

Method: Get

Parameters:

- User: <usuario del API>
- SessionID: <token devuelto por LoginUser>
- PractitionerUser: <usuario del médico>
- PatientId: <Identificador del paciente>
- EpisodeToken: <token enviado en AddEpisode>
- Episodeld: <identificador del episodio>

Response: (JSon)

```
{
  "Erc": <código de error>,
  "ErcText": "<descripción del error>",
  "PossibleDiseases" : [{
    "Disease" : "<nombre de la enfermedad>",
    "ClosenessPercentage" : "<porcentaje de cercanía>",
    "Cie10": "<código CIE10>",
    "idxESI": <índice ESI> // 1 a 5
    "ESI": <texto índice ESI>
    "Speciality1": <especialidad asociada>
    "Speciality2": <especialidad asociada>
    "Speciality3": <especialidad asociada>
  }, ...]
}
```

---

## Códigos de Error

Lista de códigos de error entregados por el sistema en el campo Erc.

- 0 No hay error
- 5000 el usuario del API no existe o es inválido
- 5001 la institución no existe o no está habilitada para operar con el API
- 5002 la firma no corresponde
- 5003 el SessionID no existe o está expirado
- 5004 el método no existe o el usuario del API no puede usarlo
- 5005 falta algún parámetro, en ErcText se indica lo que falta.
- 5006 hay algún parámetro con un valor inválido, en ErcText se indica el parámetro.
- 5007 la fecha está fuera de rango
- 5008 registro duplicado
- 5009 el médico no existe
- 5010 el paciente no existe
- 5011 el episodio no existe
- 5012 método no soportado

---

## Llaves privada y pública

- Para generar las llaves públicas y privadas de 2048 bits se pueden usar alguno de los siguiente sitios: (hay otros en internet)
  - <https://cryptotools.net/rsagen>
  - <https://travistidwell.com/jsencrypt/demo/>

- Otra forma de generar una llave privada es usar el siguiente comando en Linux

```
openssl genrsa -out rsa.private 2048
```

La llave privada queda almacenada en el archivo `rsa.private`.

Esta llave debe ser almacenada en la institución y no debe entregarse a ningún tercero.

La llave privada es un código que tiene una componente aleatoria que cambia cada vez que se ejecuta el comando, por lo tanto no se puede reproducir.

- Para generar la llave publica se puede usar el siguiente comando en Linux

```
openssl rsa -in rsa.private -out rsa.public -pubout -outform PEM
```

Este comando genera la llave pública basado en la llave privada obtenida antes y queda almacenada en el archivo `rsa.public`

La llave pública debe enviarse a Médical Sapiens para que pueda validar la autenticidad de la firma usada en el método Login.