**MUSEO VIRTUAL**

**Existen dos maneras para utilizar el prototipo:**

* + 1. Ingresando directamente a la siguiente dirección ip: <http://18.236.190.10:8000/>​, la cual corresponde a la máquina virtual en AWS, donde está alojado el prototipo del sistema, para presentarlo en la nube.
    2. Corriendo el código de manera local en cualquier maquina con Linux, siguiendo paso a paso el manual de usuario que se adjunta en la carpeta Anexos, es un archivo en pdf, con el nombre: Manual Sistema Recomendador.

**Descripción general:**

La aplicación entregada, es un prototipo de un sistema recomendador de recorridos en museos de arte virtualizado, su implementación se realizó utilizando el lenguaje de programación Python con el intérprete Django, bajo el sistema operativo Linux, la base de datos esta implementada en MySql, y un servidor de aplicaciones ngnix, el servidor de aplicaciones debe soportar la ejecución de ambientes virtuales, este debe generar un socket de comunicación con gunicorn.

Los datos que se utilizaron para las pruebas, son datos reales, extraídos de una base de datos del banco de la república, sobre obras de arte.

**Entregables**

En la carpeta prototipo, se va a encontrar una carpeta con nombre system\_recommender\_final y dentro de ella, existen las carpetas y archivos que se muestran en la imagen 1:

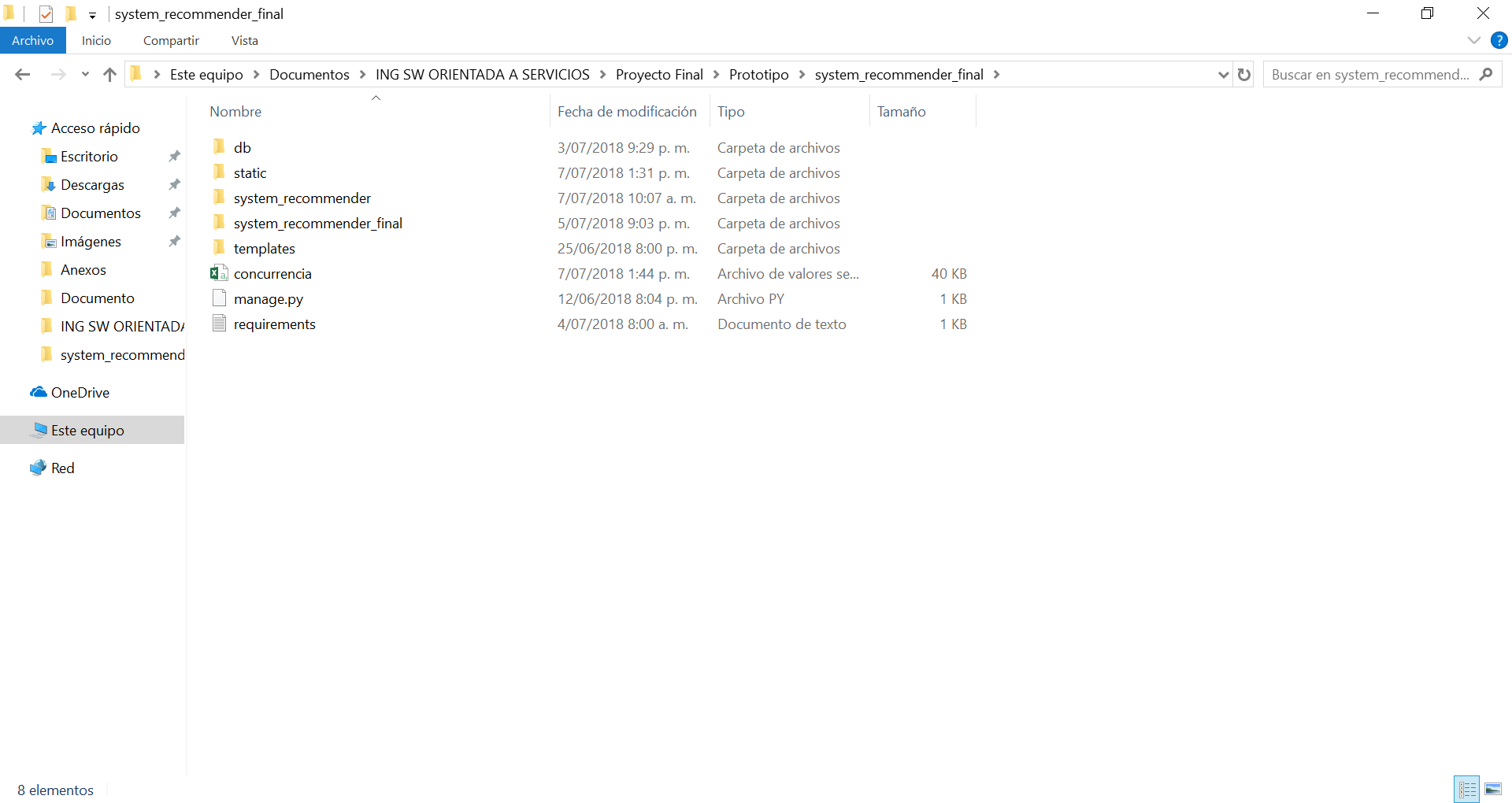


Imagen 1. Detalle carpeta Prototipo

A continuación, se detallan las carpetas entregadas, con la función de cada una de ellas dentro del prototipo:

* + - 1. Db: dentro de esta carpeta se encuentra el archivo Dump20180703.sql, este es el buckup de la base de datos.
      2. Static: corresponde a todos los archivos estáticos, los que no cambian durante el proyecto, está dividido en dos carpetas:
* Resources: Recursos propios que se diseñaron con el fin de mostrar la interface, como lo muestra la imagen 2.

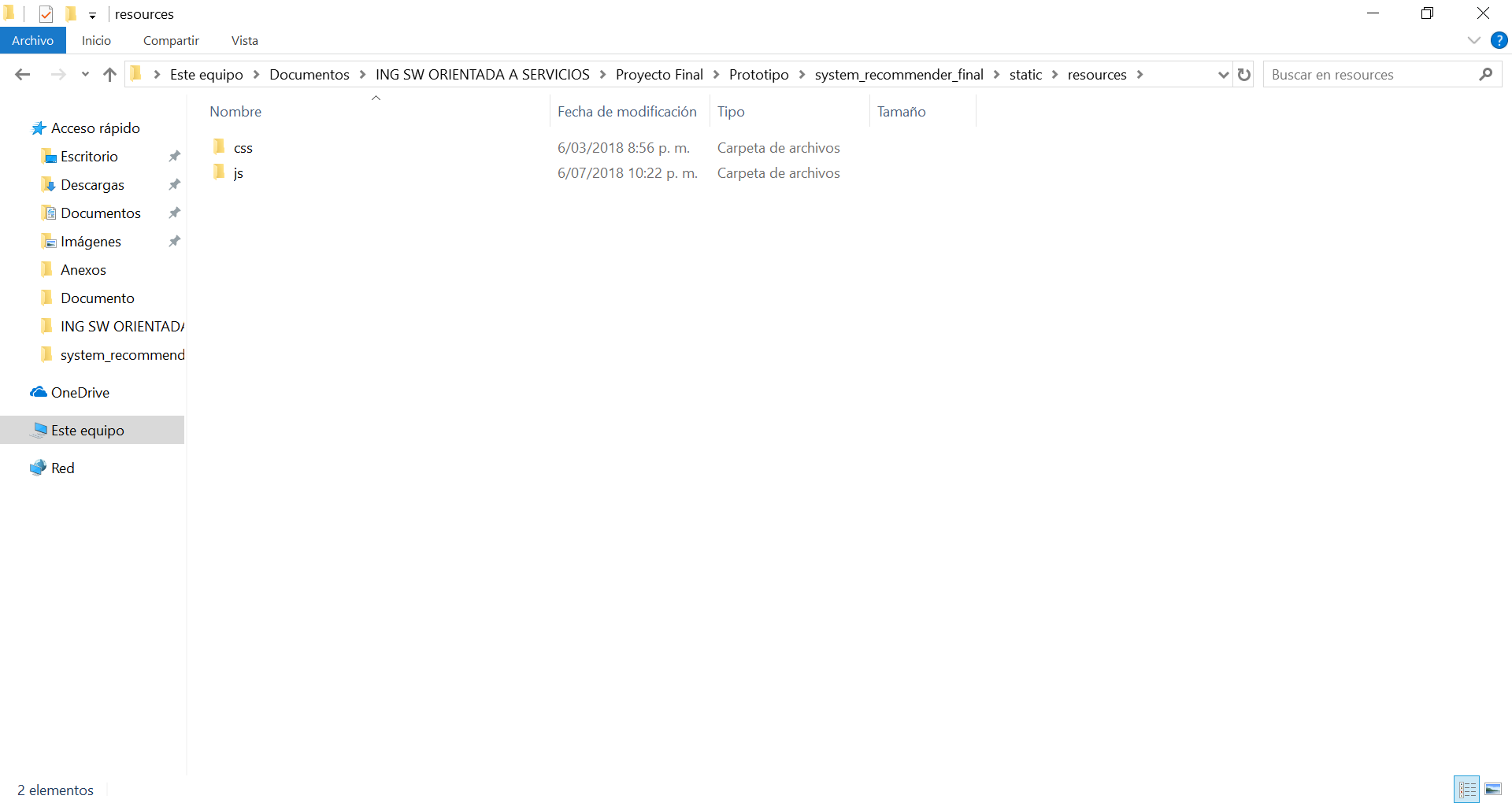


Imagen 2 Recursos

* Vendors: Librerías utilizadas para el desarrollo del proyecto a nivel de frontend, las cuales se muestran en la imagen 3.

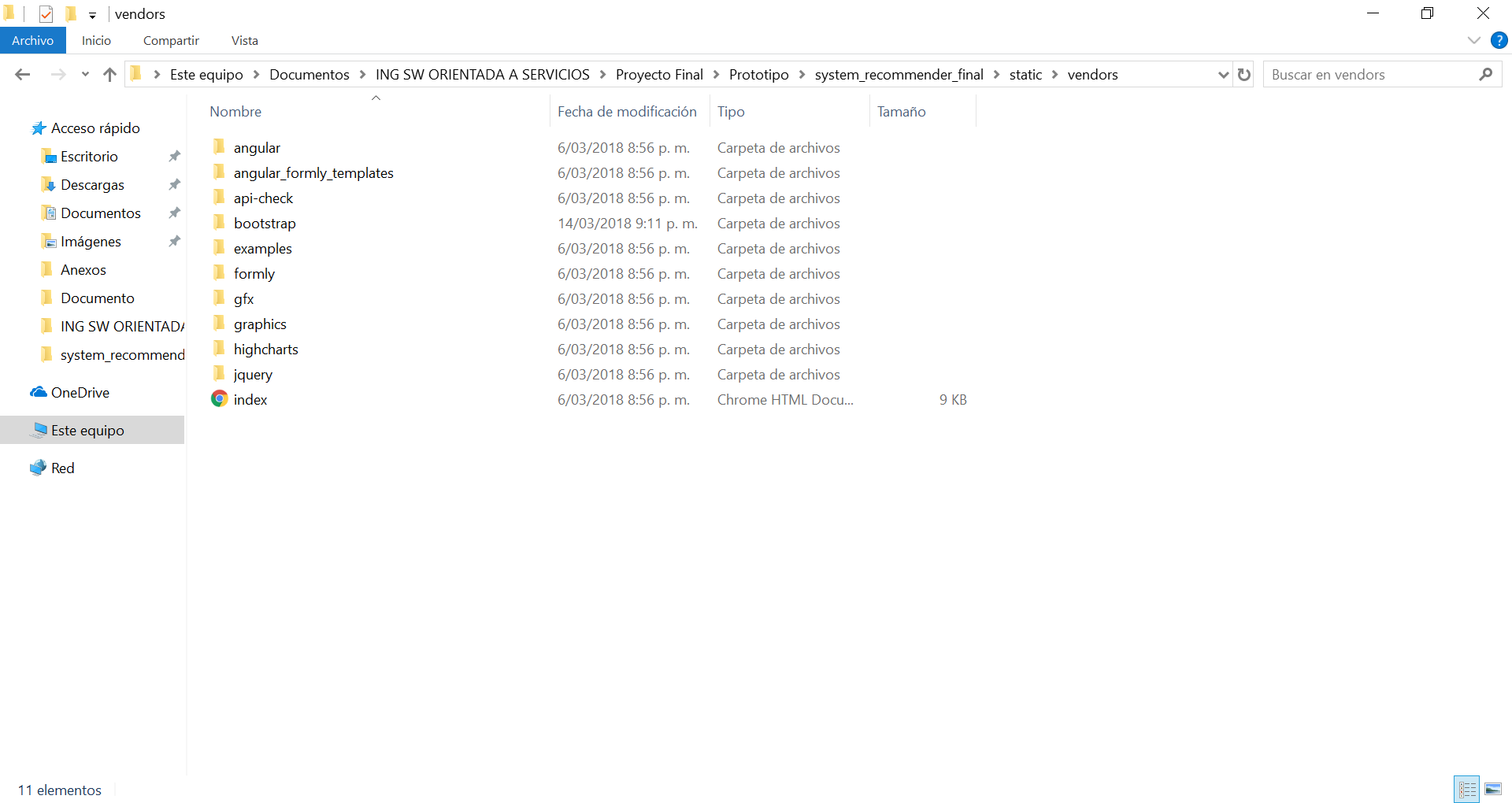


Imagen 3. Librerías

* + - 1. En Django se deben generar unas aplicaciones, en este caso la aplicación generada es system\_recomender, los archivos que se encuentran en esa carpeta se gestionan a través del backend, esto lo hace directamente Django (imagen 4)

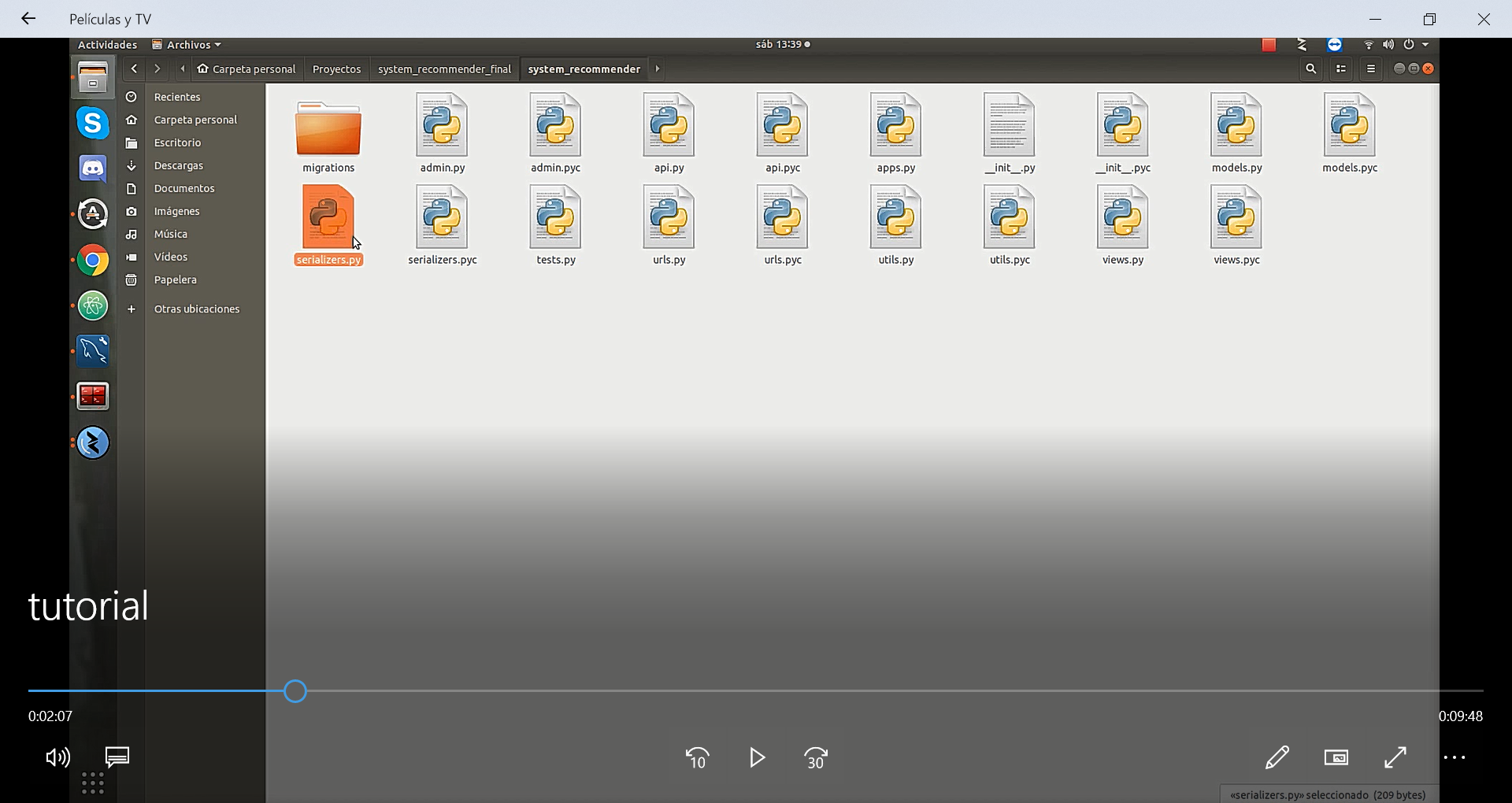


Imagen 4. system\_recomender

Dentro de los archivos, se observa por ejemplo el de models, que contiene los modelos que se van haciendo para la base de datos, también se tienen serializadores, ya que la idea es que esos modelos se puedan serializar para ser mostrados luego como respuestas de json, esos servicios están montados dentro del archivo de api, este archivo dispone todos los modelos que están desarrollados y que están siendo serializados, el archivo url define cuales son las direcciones de acceso para cada uno de los servicios, el archivo de utilidades, donde se encuentra todo lo referente al sistema recomendador.

* + - 1. En la carpeta system\_recomender\_final (Imagen 5), se encuentra toda la configuración del proyecto.

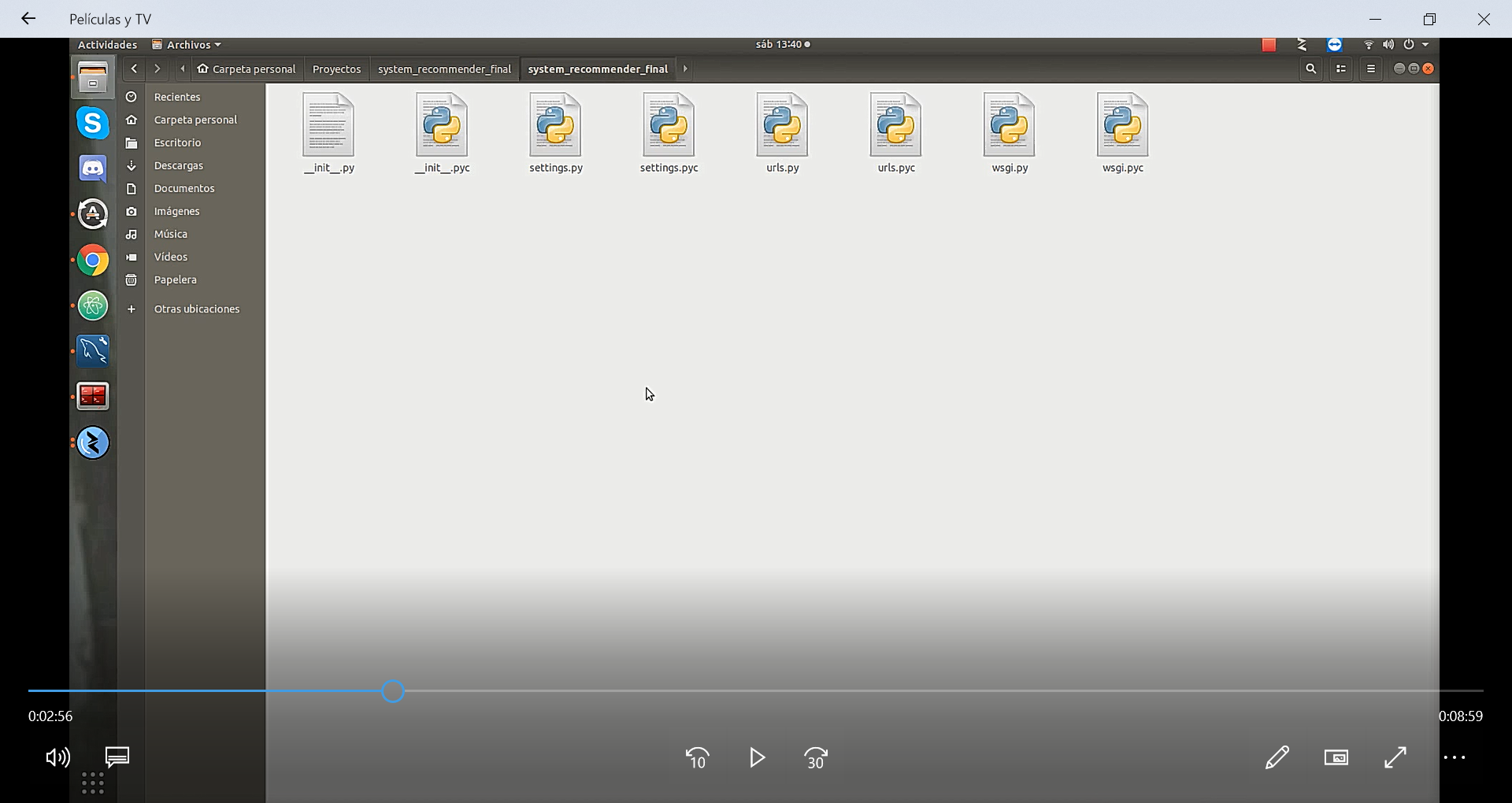


Imagen 5. system\_recomender\_final

* + - 1. En la carpeta de Templates, está alojada la interface gráfica que va visualizar el usuario.
      2. El archivo manage.py sirve para hacer la administración del servicio.
      3. El archivo requirements.txt, contiene las librerías y las versiones de las librerías que se están utilizando.
      4. El archivo concurrencia.csv, contiene la matriz de concurrencia, que es la base del sistema recomendador, para realizar las respectivas recomendaciones al usuario.
      5. La carpeta system\_recomender\_final\_env, contiene el ambiente virtual y esta debe ser generada con los pasos que se explican en el manual adjunto, que se encuentra en la carpeta Anexos, con el nombre Manual sistema recomendador.pdf.

**Base de Datos:**

Como se mencionó anteriormente, la base de datos esta implementada en MySql 5.7, dentro de las carpetas entregadas del código, en la carpeta System\_recommender\_final, existe un archivo llamado settings.py, el cual tiene la configuración de los parámetros, de acuerdo a lo que cada usuario maneje acerca de la base de datos, como se muestra en la Imagen 6:

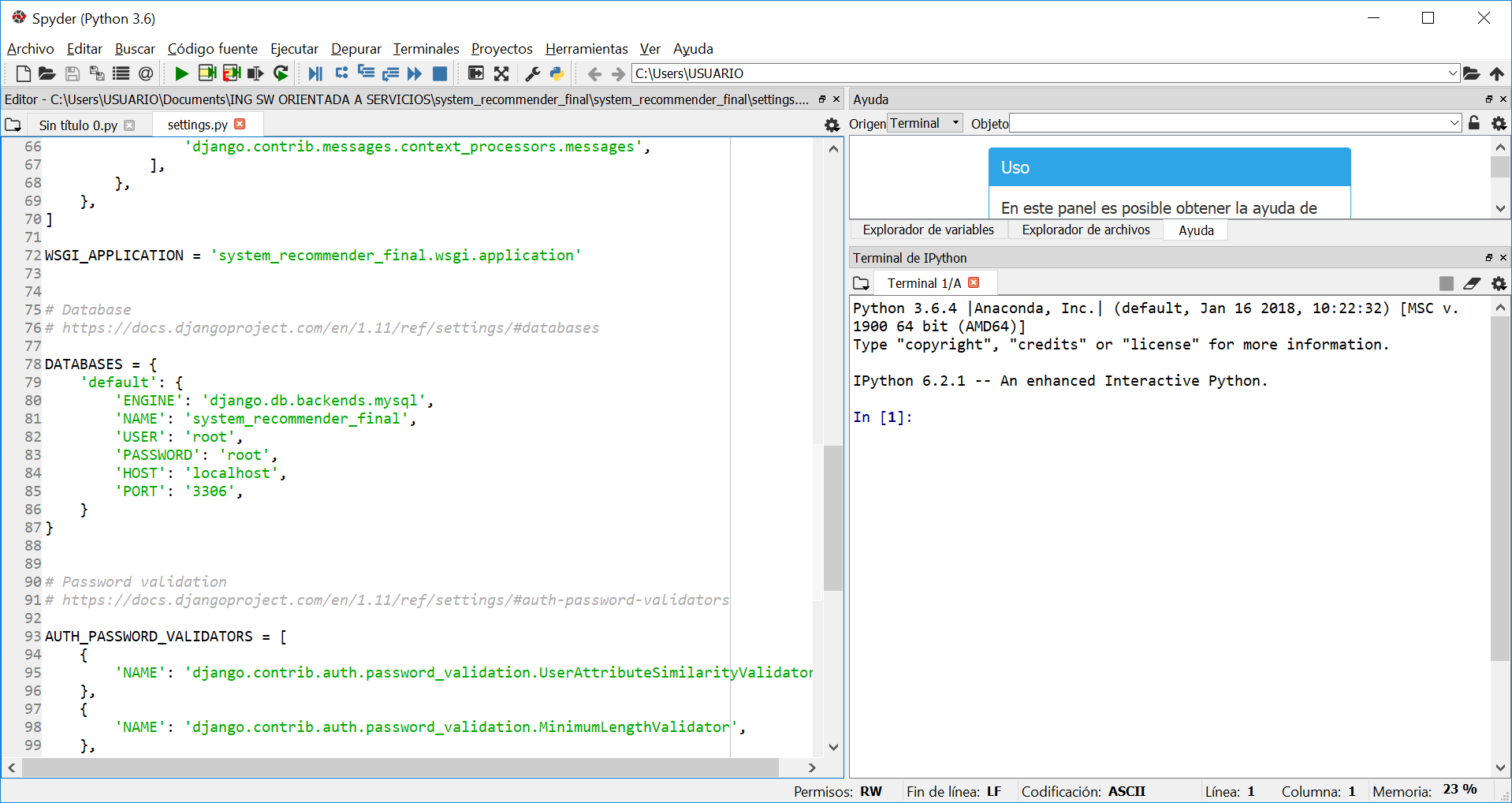


Imagen 6. Parámetros Base de Datos

El modelo de base de datos se muestra en el diagrama 1:

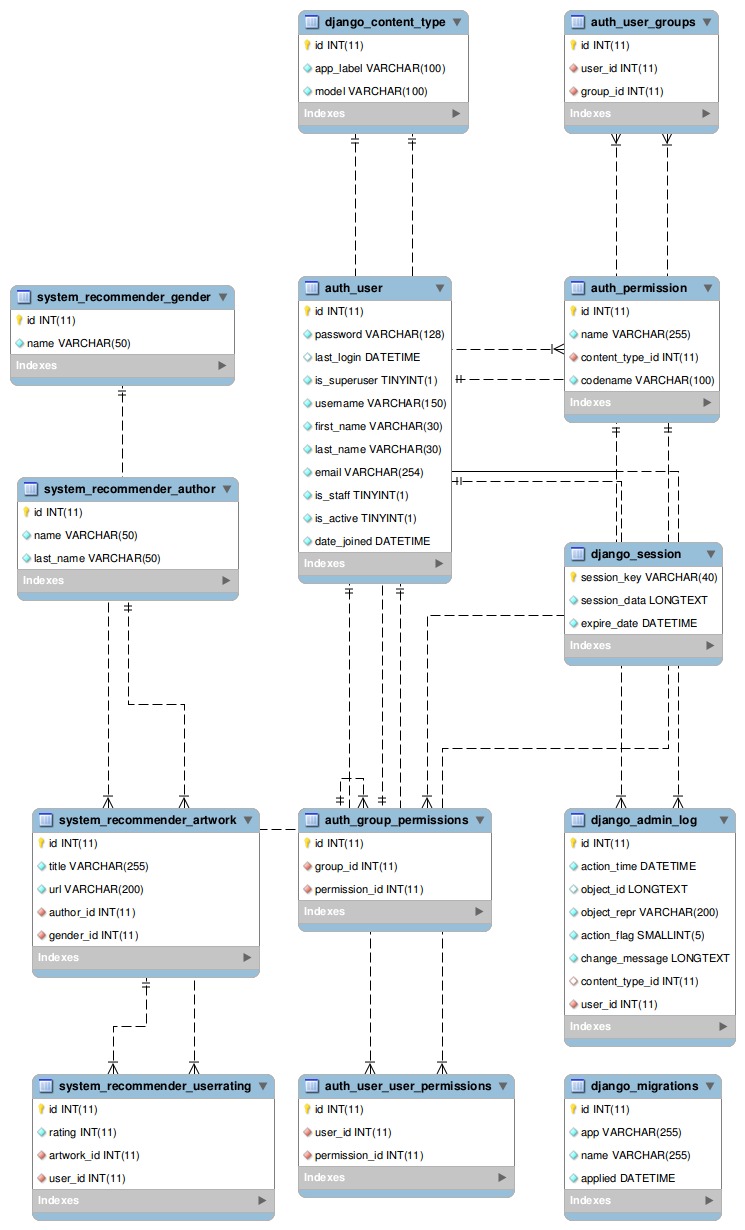


Diagrama 1. Modelo de Base de datos

**Funcionalidad del prototipo y pruebas**

En un sistema de recomendación, el objetivo es que la recomendación sea lo más precisa posible, hay varias técnicas de filtrado que identifican el tipo de sistema que se puede utilizar, entre ellas se destacan:

**Comentarios explícitos:** la retroalimentación explícita se desarrolla a través de un proceso de encuesta, el usuario da una calificación de contenido Esta retroalimentación permite con precisión las preferencias del usuario y permite dar una calificación de contenido.

**Comentarios implícitos:** la retroalimentación implícita evalúa, sin la intervención del usuario, las posibles opciones de recomendación, ya sea a través de películas, artículos web, libros, programas de televisión y otros. Estos eventos nos permiten analizar y comprender los gustos y preferencias del usuario.

El prototipo es un sistema recomendador de recorridos en museos de arte virtualizado, se diseñó con la técnica de recomendación de comentarios explícitos y se basó en una matriz de concurrencia.

Los sistemas recomendadores basan el contenido en la matriz de concurrencia para calcular las obras que serán recomendadas a los usuarios. Cada una de las obras presentadas se asocia a una característica que toma un valor de 0 a 1, estas características indican que tanto pertenece una obra a un género, siendo 1 que pertenece totalmente al género y 0 que no pertenece.

Posteriormente, las obras son calificadas por cada uno de los usuarios, no es necesario que los usuarios den calificación de todas las obras, ya que el sistema recomendador intentara predecir los valores para cada uno de los usuarios.

En la tabla, los signos de interrogación (?) representan las calificaciones no dadas por los usuarios sobre una pieza de arte.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Piezas de Arte** | **Usuarios** | | | **Caracteristicas** | |
| José | Sergio | Julian | x1 (Abstracto) | x2 (Expresionismo) |
| Sin titulo | 5 | 5 | 0 | 0,9 | 0 |
| Las mariposas | 5 | ? | ? | 1 | 0,01 |
| Casa blanca | ? | 4 | 0 | 0,99 | 0 |

Para las características se construye la matriz de características X como se muestra a continuación:

Cada una de las filas de las columnas de la matriz corresponde a una fila de la anterior tabla. Posteriormente se define la matriz que corresponde a las calificaciones realizadas por los usuarios.

Para cada usuario , el parámetro de aprendizaje . Predecir la calificación del usuario con la obra con . Este cálculo se realiza por cada uno de los signos de interrogación presentes en la matriz.

Finalmente, el modelo de aprendizaje está dado por:

Donde:

Corresponde a 1 si el usuario ha calificado la obra o 0 en otro caso

Calificación dada por el usuario a la pieza de arte

Número de obras calificadas por el usuario.

variables involucradas en el cálculo.

Parámetro de aprendizaje.

El modelo matemático involucra cada una de las calificaciones dadas por el usuario para predecir las calificaciones de un usuario a una obra. Para entregar las obras que al usuario más le gustaría, basta con ordenar de mayor a menor las calificaciones arrojadas por el sistema recomendador.

Para las pruebas se utilizaron imágenes públicas del Banco de la Republica (Colombia).

La aplicación funciona por medio de servicio web, que recibe un json con el usuario (imagen 7), el cual puede ser un usuario nuevo un usuario registrado, el proceso de registro es automático, cuando el usuario se ingresa; y las obras cuyas urls se encuentran almacenadas en la tabla Artworks (imagen 8).

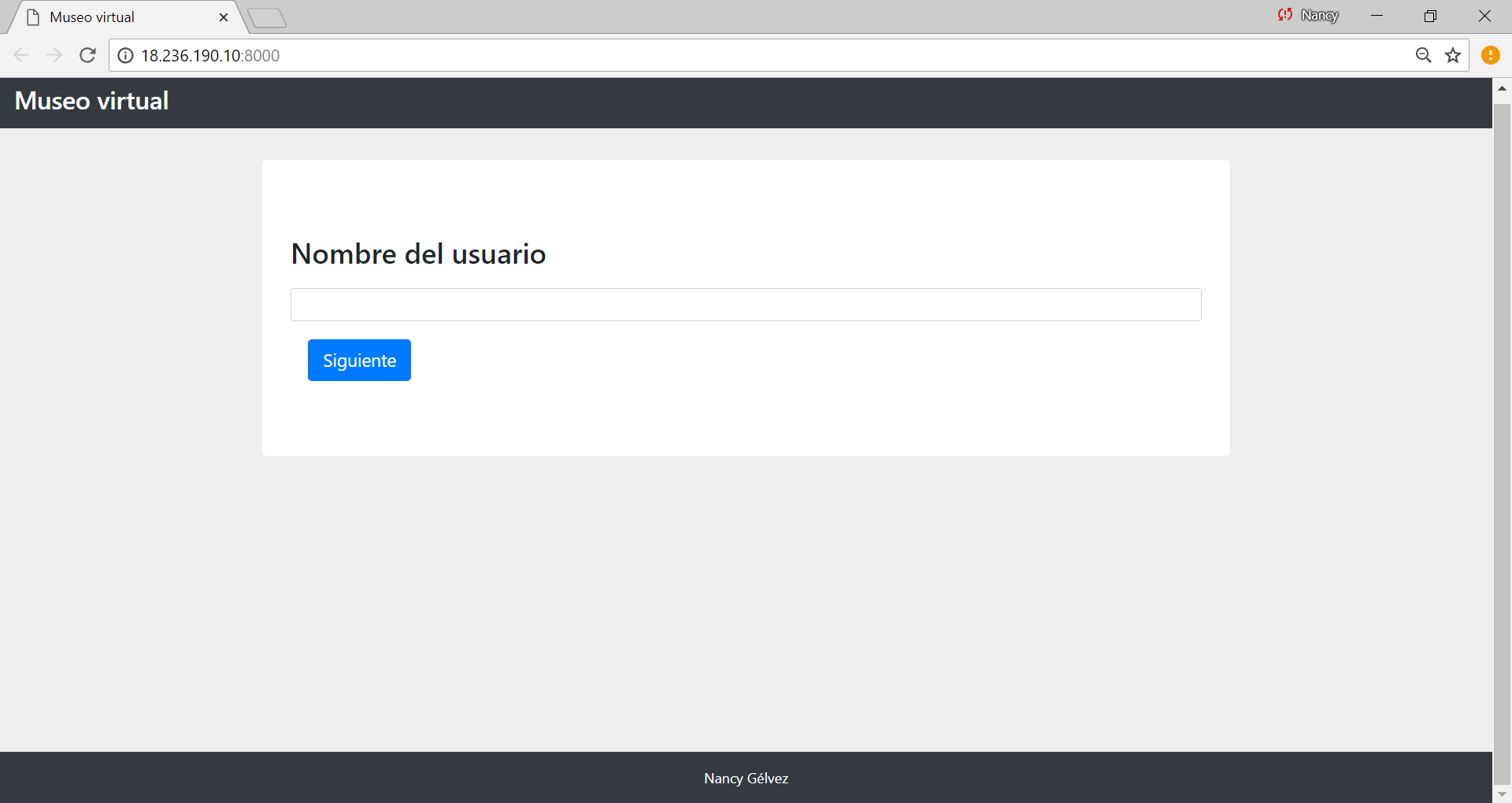


Imagen 7. Ingreso de usuario

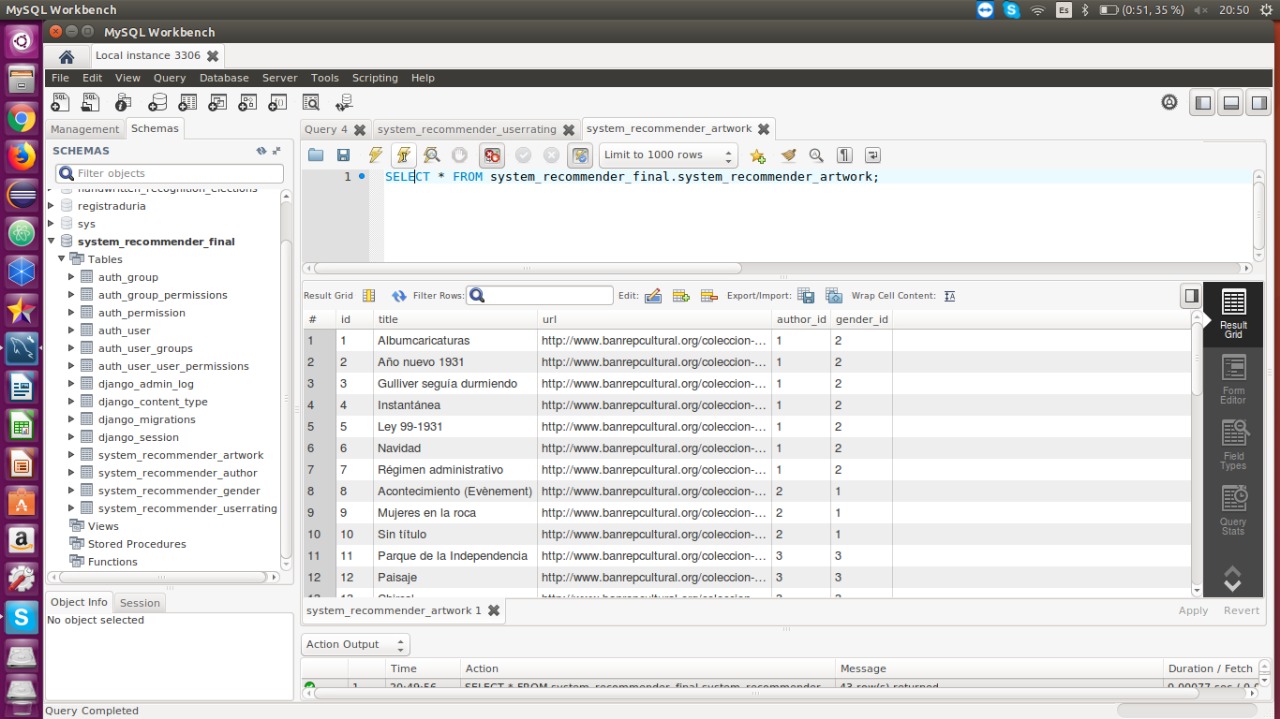


Imagen 8. Obras Almacenadas en BD

Paso siguiente, el servicio devuelve las obras que se encuentran almacenadas por medio de su url, y las muestra al usuario (imagen 9)

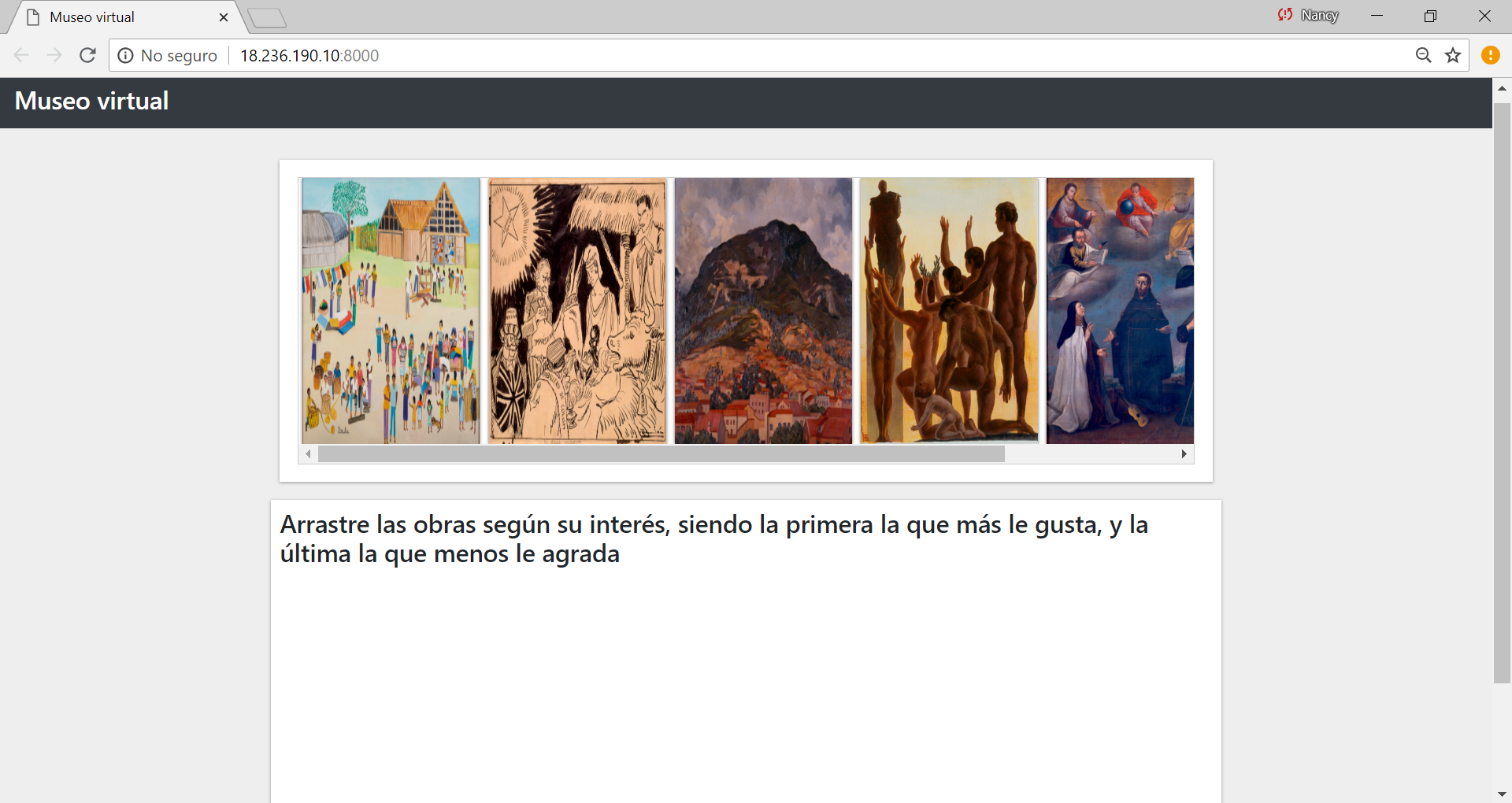


Imagen 9. Cargue de obras de arte

Posteriormente se irán mostrando 5 encuestas compuestas de seis imágenes para que el usuario las clasifique de la manera que considere identifica sus preferencias, las encuestas mencionadas se verán de la siguiente forma (imágenes 10-14).

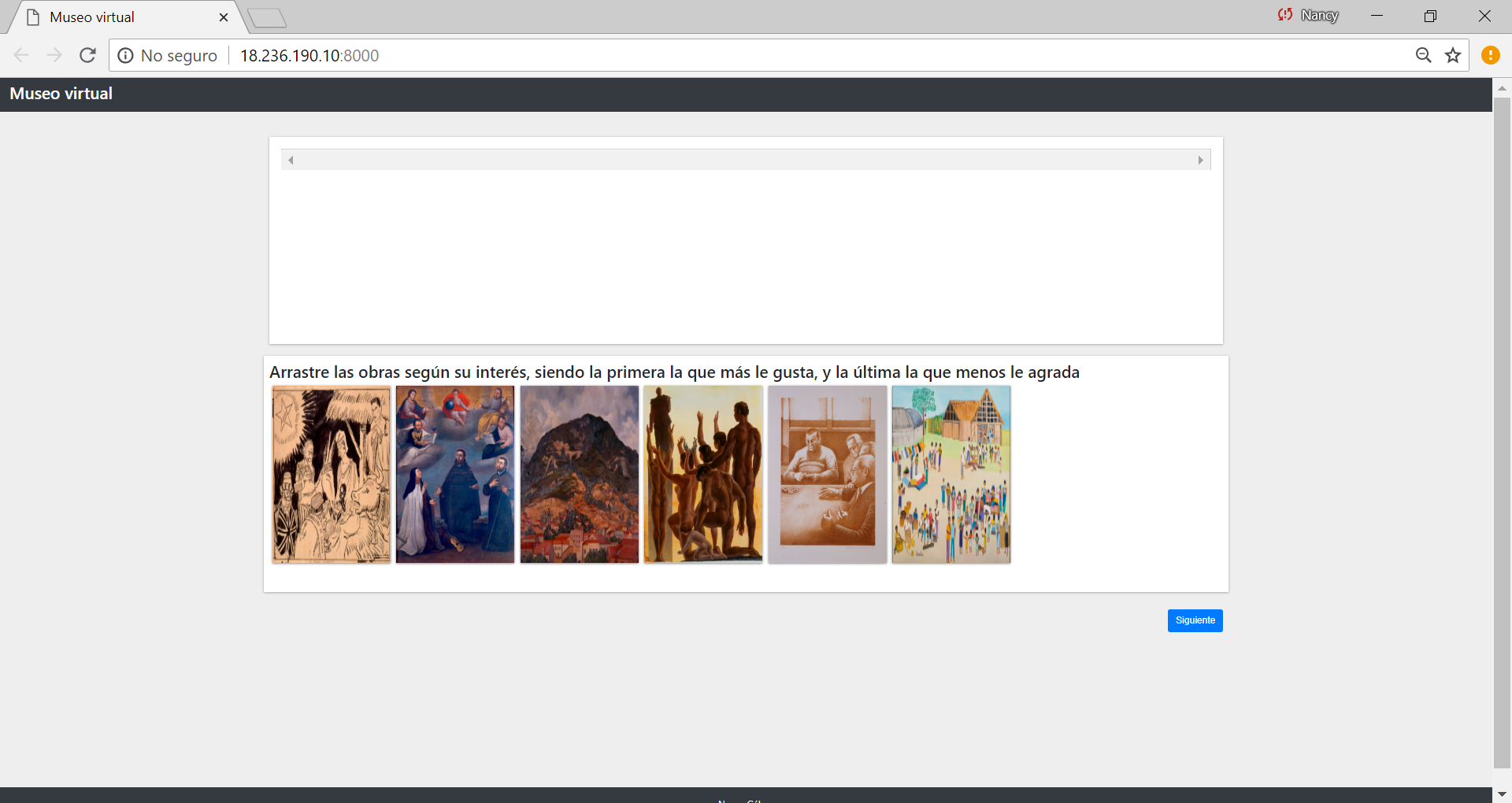


Imagen 10. Primer grupo de imágenes

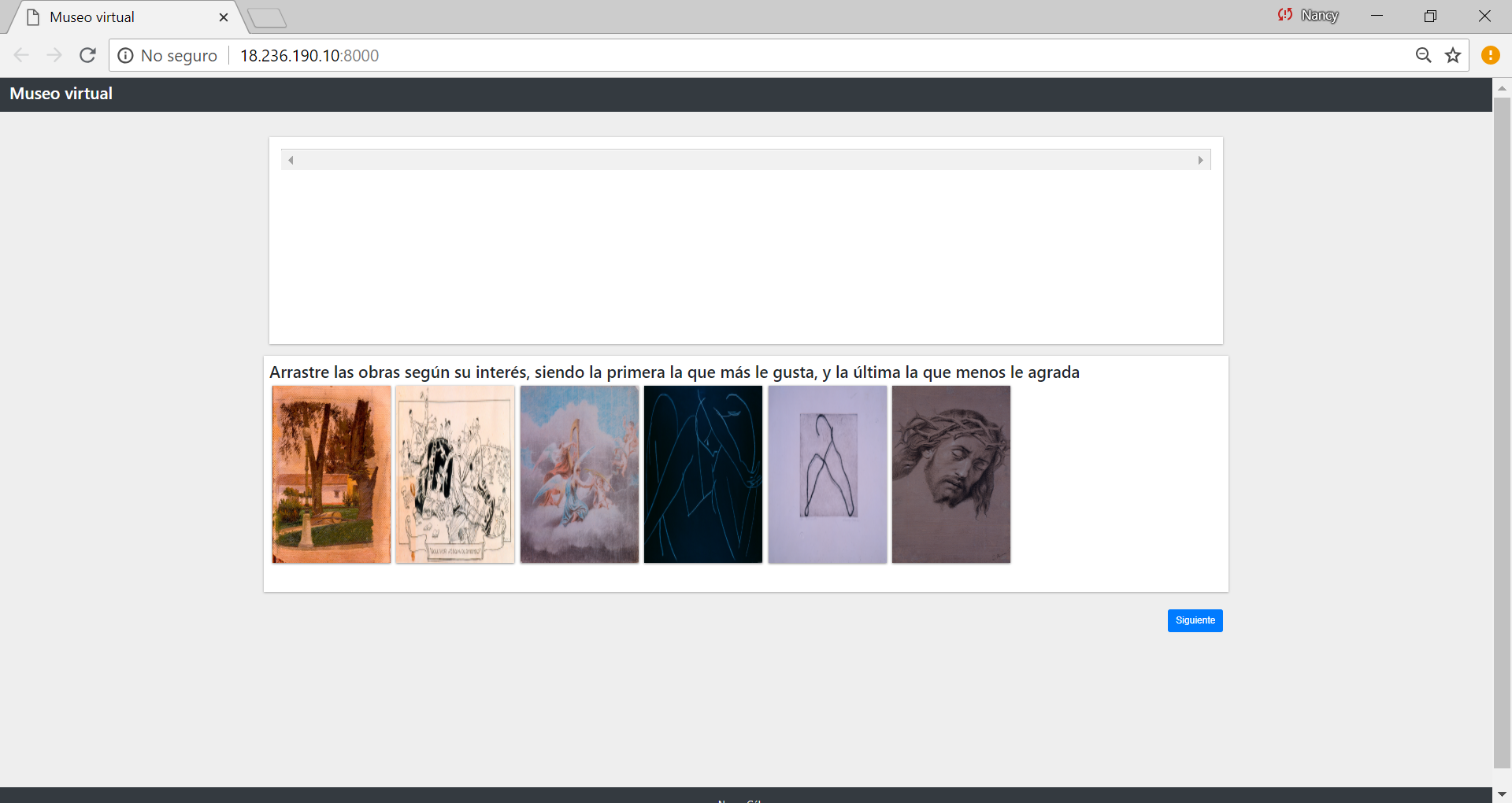


Imagen 11. Segundo grupo de imágenes

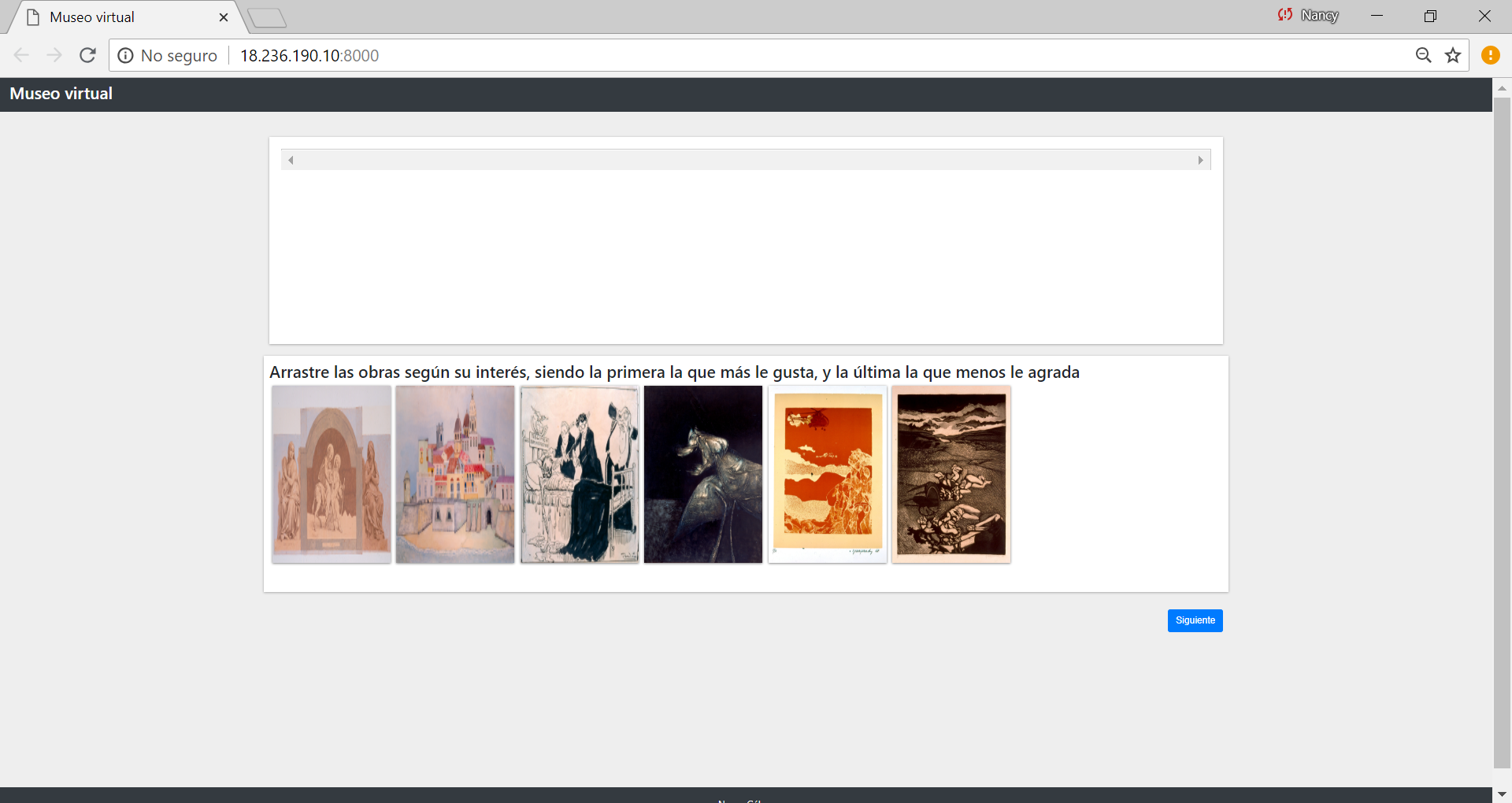


Imagen 12. Tercer grupo de imágenes

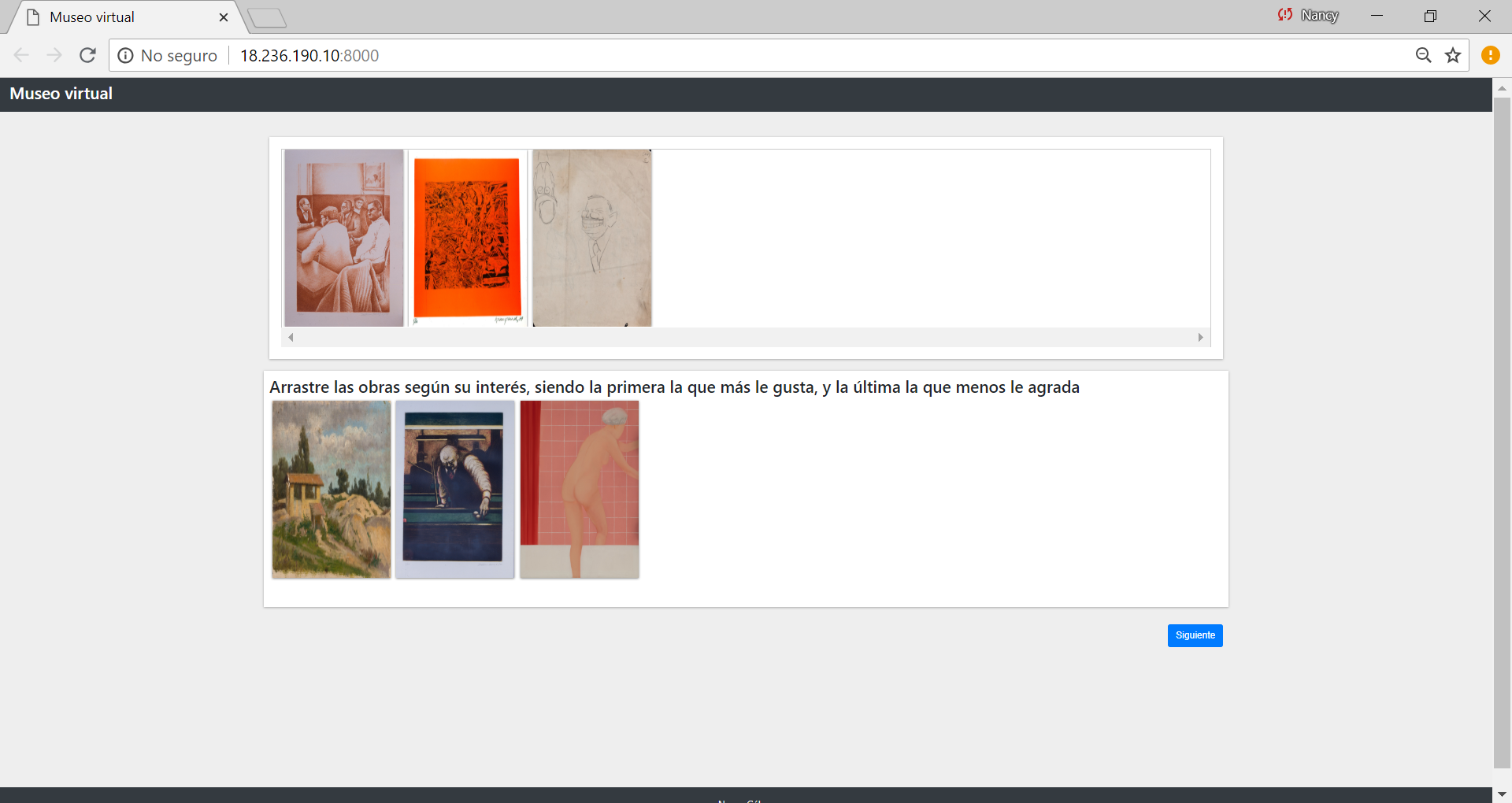


Imagen 13. Cuarto grupo de imágenes

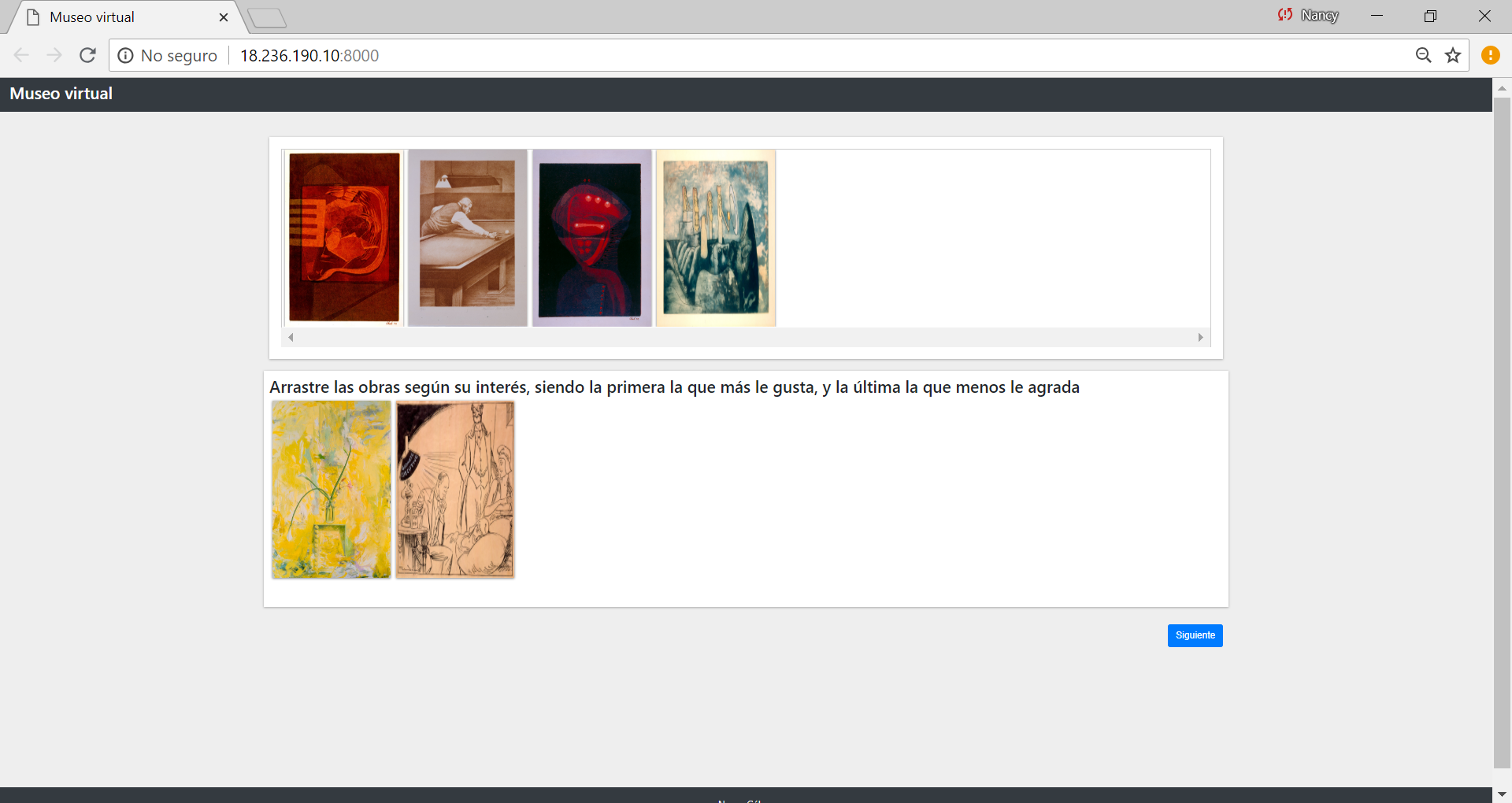
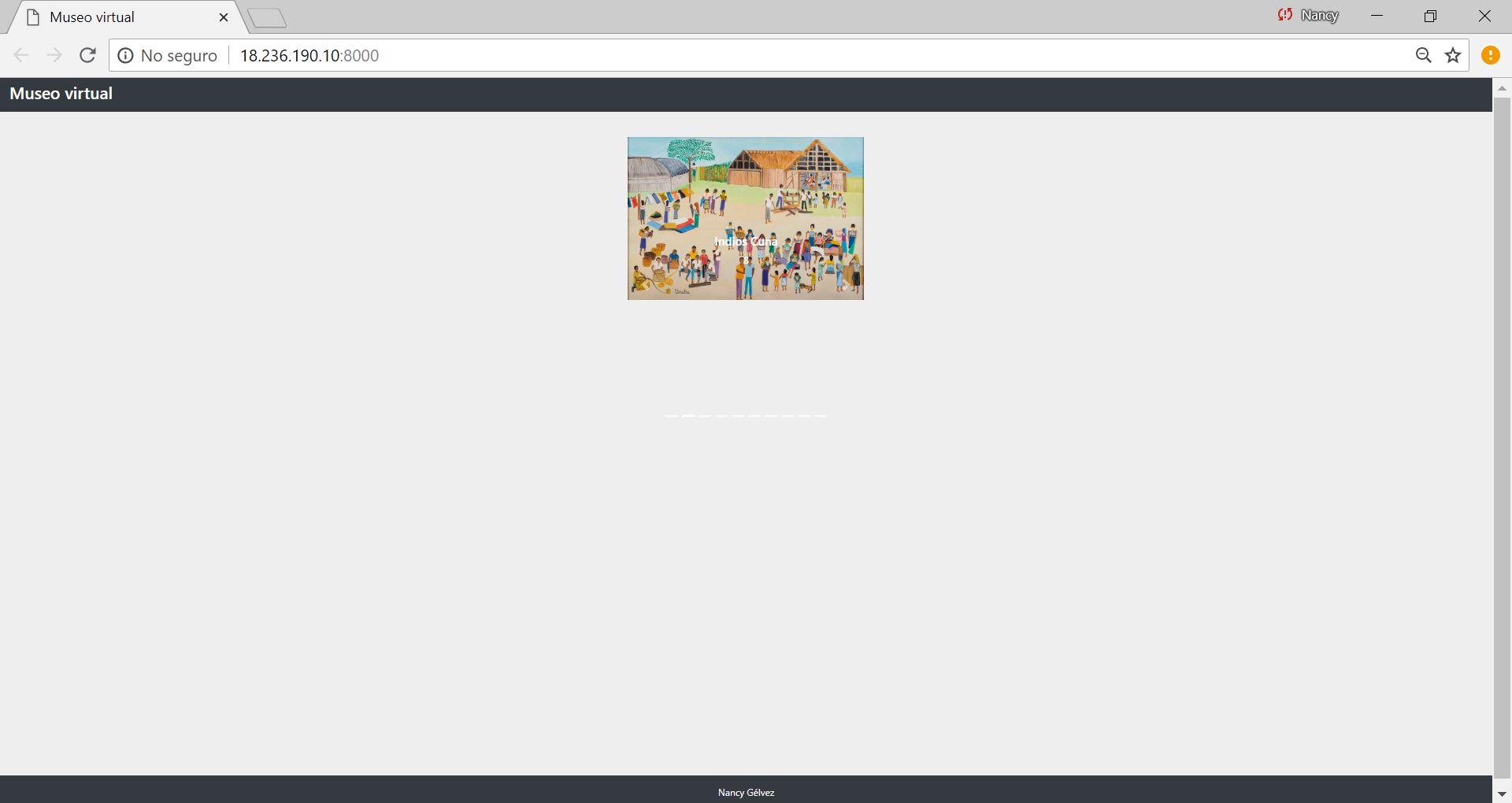


Imagen 14. Quinto grupo de imágenes

A continuación el servicio retorna un json con las obras de arte, para tal fin, el sistema calcula la matriz de concurrencia, basada en las preferencias del usuario y de los demás usuarios que hayan ingresado a la aplicación y cuyas preferencias sean parecidas o similares a las del usuario que está utilizando la aplicación en ese instante de tiempo, el sistema califica las obras de acuerdo al orden en el que el usuario las haya colocado, obteniendo 10 puntos la obra que haya colocado en primer lugar en cada grupo de obras, hasta 0, la obra que haya colocado en último lugar, de esta manera se calcula las obras que el usuario va a visualizar, Al final el sistema arrojará la ruta que el usuario puede recorrer por medio de slides, según las obras escogidas, cómo se muestra a en la imagen 15.



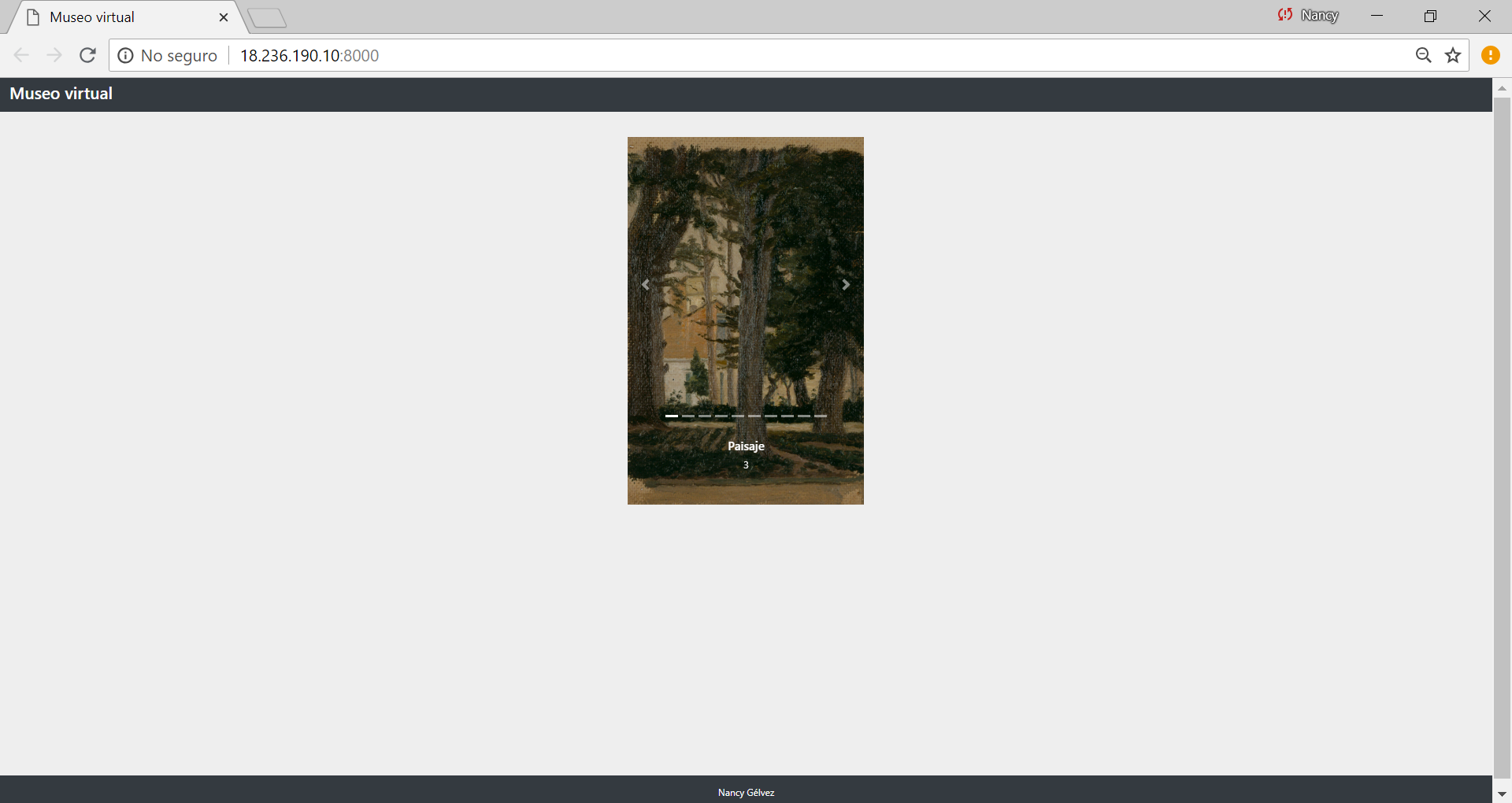


Imagen 15. Recorrido virtual

Finalmente se puede observar que de acuerdo al sistema recomendador, el prototipo muestra el recorrido virtual de un museo de obras, de acuerdo a las preferencias tanto del usuario que esté utilizando la aplicación, como de otros usuarios con preferencias similares.