Diseño de Infraestructura Inteligente para el IoT

Práctica 4: Integración de Sistemas Heterogéneos

Jiali Zheng

Jorge Vicente Camuñas Heredia

2 / 12 / 2020

Introducción

En esta práctica se pide desarrollar un cliente y un servidor valiéndose de la integración de sistemas heterogéneos. En concreto, para la parte del cliente se pide un programa en java que permita realizar varias operaciones con datos a través de una interfaz de usuario. Para la parte de servidor, se pide recibir por parte del cliente una serie de datos para realizar una predicción. Todos los datos serán proporcionados por el cliente.

A continuación, se van a desarrollar las partes que forman este proyecto: cliente y servidor.

Cliente

La parte de cliente está desarrollada en Java y debido a la extensión del código se van a mencionar brevemente las partes más relevantes:

- `MainFrame.java`: en este archivo se definen las diferentes partes de la interfaz de usuario. Esta interfaz consta de los siguientes botones: `Add`, `Delete`, `Send` y `Predict`. Además de una tabla para albergar los datos introducidos y varios cuadros de diálogos para los distintos botones.
- `Integration.java` y `IntegrationImp.java`: en estos archivos se define lo necesario para la interacción con el servidor python por parte del cliente.
- `Main.java`: en este archivo se establecen los valores generales usados en todo el programa.

Servidor

El servidor python consta de tres parámetros que se corresponden con los valores x1, en este caso horas en casa, x2, en este caso horas fuera de casa, e y, en este caso si el día es libre o de trabajo. A continuación, se definen los siguientes endpoints para comunicarse con el cliente:

- `/train_data`: guarda los datos recibidos desde el cliente para su posterior uso
- `/fit`: realiza el entrenamiento con los datos recibidos del cliente con el endpoint anterior.
- `/predict`: se encarga de usar el modelo creado anteriormente sobre los datos recibidos para predecir el valor y, después devuelve el resultado obtenido.

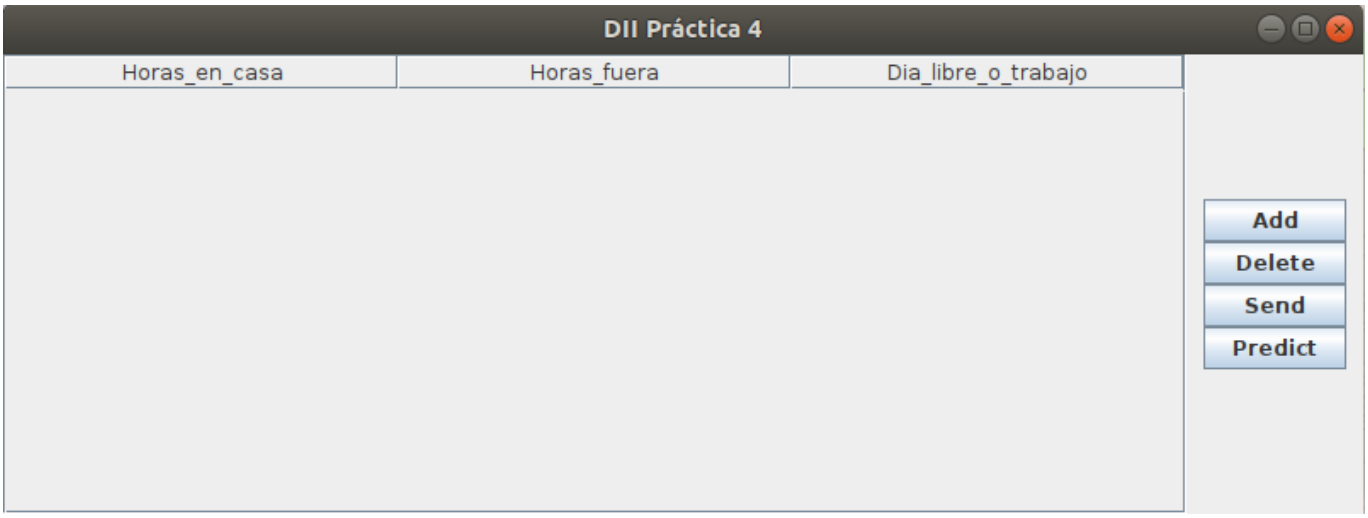
Por último, se ejecuta en 'localhost' en el puerto 8080.

Guía de uso

En este apartado se desarrollarán los pasos a seguir para hacer uso del programa.

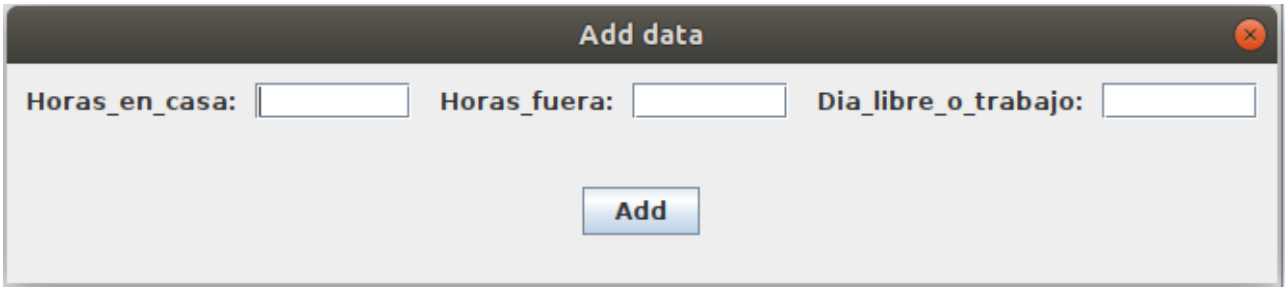
Se describirán a continuación los pasos a seguir:

1. Ejecutar el servidor python
2. Ejecutar el cliente java
3. Al ejecutar el cliente aparecerá la siguiente interfaz de usuario:

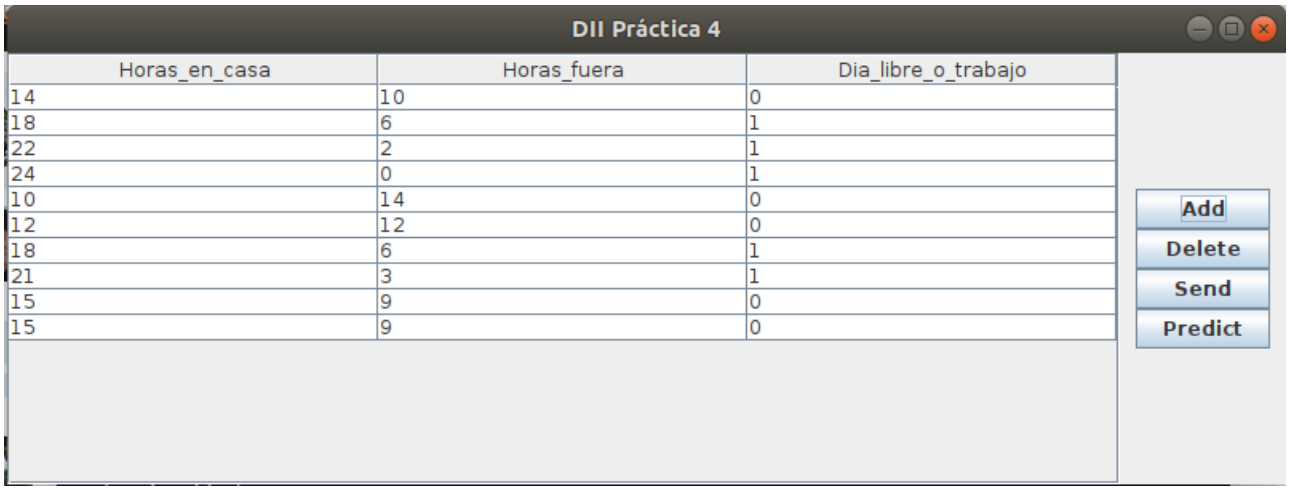


4. En esta interfaz se pueden elegir las siguientes opciones:

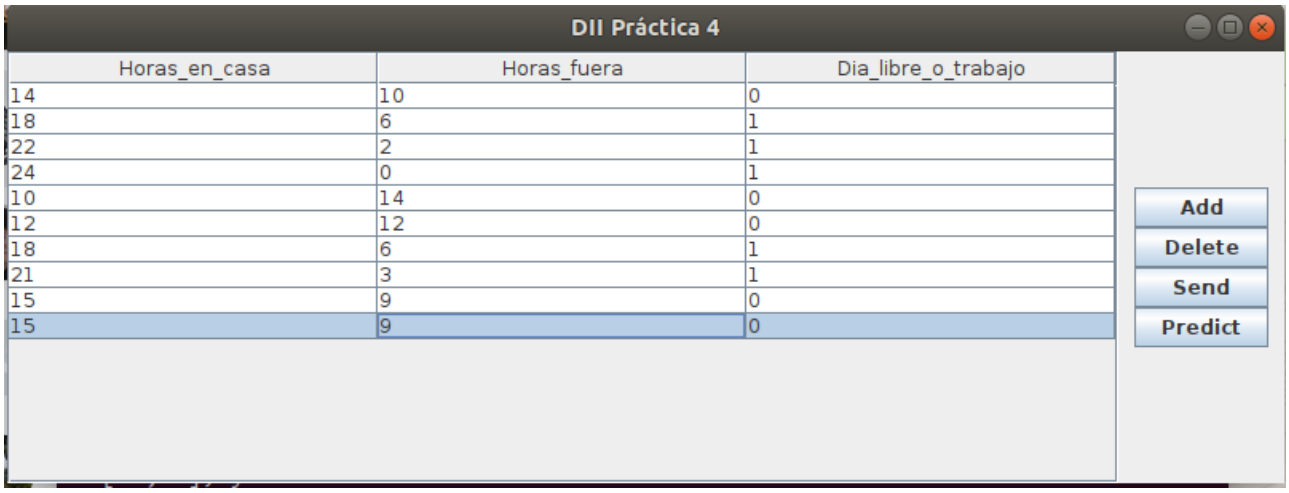
    - Add: permite añadir un conjunto de valores (horas en casa, horas fuera de casa y dia de libre/trabajo[0,1])



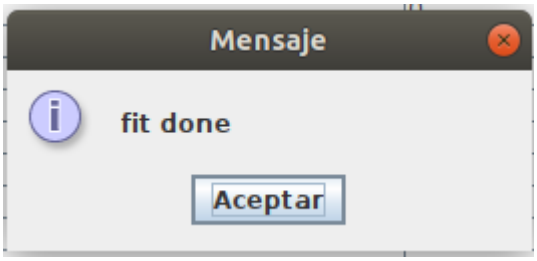
Si se añaden varios datos, la interfaz se queda de la siguiente forma:



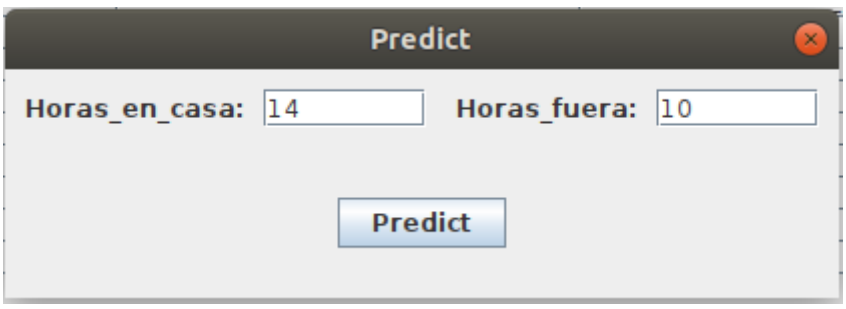
    - Delete: si se selecciona una linea de datos y se pulsa este botón se elimina dicha linea

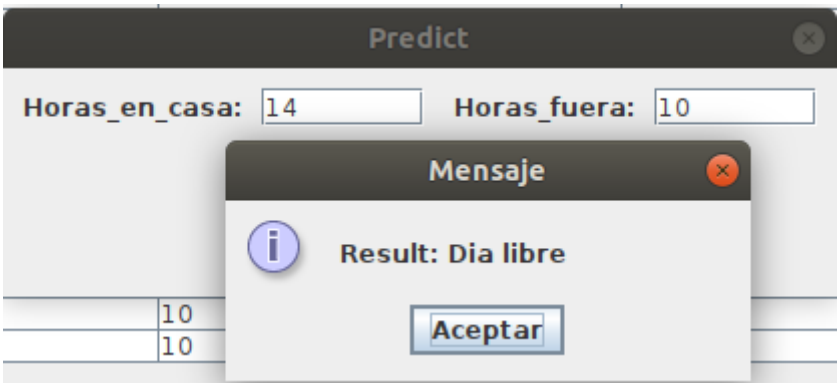


- Send: envía los datos guardados al servidor python donde se realiza el entrenamiento para la predicción. El cliente muestra el siguiente mensaje al enviar los datos



- Predict: pide al usuario los valores de horas y los envía al servidor donde haciendo uso del modelo generado con el Send de los datos introducidos. Como respuesta, el servidor envía el resultado al cliente quien lo interpreta como un día de trabajo o libre y se muestra por pantalla





5. Con esto elementos se puede hacer uso del programa para determinar si el día en cuestión ha sido un día laboral o libre.