

CRUD IL

Alumno: Óscar Talavera Sáez

Tutora PFC: Belén Gil Montilla

Curso: 2º DAM

Índice

1. Portada
2. Índice
3. Resumen del proyecto
4. Justificación y objetivos del proyecto
5. Desarrollo del proyecto
6. Conclusiones
7. Posibles actualizaciones
8. Bibliografía

Resumen del proyecto

Crudoil es una aplicación para obtener la información de las gasolineras de tu zona. Consta de una aplicación de backend(Api) que pasa la información a la aplicación que la filtra y muestra la información necesaria.

Justificación y objetivos del proyecto

En un mundo en el que el valor de los combustibles fósiles está al alza, para muchas familias es vital poder ahorrarse la mayor cantidad de dinero posible. Se pretende tener la información sobre el precio de las gasolineras, la dirección y el horario y si el usuario lo desea abrir google maps con las coordenadas de la gasolinera para facilitar el traslado del usuario hacia ella.

Desarrollo del proyecto

Análisis del mercado

En el mercado existen distintas aplicaciones parecidas a Crudoil. Las diferencias de esta aplicación es que está dirigida para pc y las aplicaciones semejantes y usa una Api propia que puede usarse de BackUp, ya que en caso de que la [Api estatal](#) dejase de funcionar la Api privada seguiría funcionando y otorgando la información necesaria a la Aplicación.

Metodologías utilizadas y Descripción de los componentes

La [Api](#) está programada en Java ya que es uno de los lenguajes por excelencia para el BackEnd. Está planteado con una clase ejecutora en el que se especifica la dirección y el puerto. También se ejecuta mediante hilos la clase en la que se pasa la información mediante una llamada Get. En la clase GestorHTTP se lee el json en un archivo porque la Api estatal está en una dirección https lo que dificulta la conexión y el intercambio de datos.

```

public class GestorHTTP implements HttpHandler {
    public void handle(HttpExchange httpExchange) {
        // TODO Auto-generated method stub
        String requestParamValue=null;
        if("GET".equals(httpExchange.getRequestMethod())) {
            requestParamValue = handleGetRequest(httpExchange);
            handleGETResponse(httpExchange, requestParamValue);
        }
        private String handleGetRequest(HttpExchange httpExchange) {
            return httpExchange.getRequestURI().toString();
        }
        private void handleGETResponse(HttpExchange httpExchange, String requestParamValue) {
            try {
                JSONObject jsonObject = (JSONObject) readJsonSimpleDemo("Api.json");

                OutputStream outputStream = httpExchange.getResponseBody();
                String htmlResponse = jsonObject.toString();
                httpExchange.sendResponseHeaders(200, htmlResponse.length());
                outputStream.write(htmlResponse.getBytes());
                outputStream.flush();
                outputStream.close();
            } catch (Exception e) {}
        }

        public static Object readJsonSimpleDemo(String filename) throws Exception {
            FileReader reader = new FileReader(filename);
            JSONParser jsonParser = new JSONParser();
            return jsonParser.parse(reader);
        }
    }
}

```

La App está programada en Java para poder aplicar lo aprendido durante las prácticas. Se ejecuta por consola y se pregunta por la preferencia del usuario para filtrar la información. [Recibe](#) y gestiona la información de la Api que son más de 11.600 gasolineras. Se [mapea](#) la información del Json con la librería [Gson](#) utilizando una clase poniendo los atributos que necesitamos coger. Puedes filtrar por provincia, pueblo o código postal.

```

static Scanner x = new Scanner(System.in);
static String urlApi= "http://127.0.0.1:7777/api";
static ArrayList<Gasolinera> gasolineras = new ArrayList<>();
static String mayuscula(String str) {
    char[] arr = str.toCharArray();
    arr[0] = Character.toUpperCase(arr[0]);
    return new String(arr);
}

public static void llamadaApi(String urlApi) throws IOException, InterruptedException{
    HttpClient client = HttpClient.newHttpClient();
    HttpRequest request = HttpRequest.newBuilder()
        .uri(URI.create(urlApi))
        .GET()
        .build();
    HttpResponse<String> respuesta=client.send(request,BodyHandlers.ofString());
    Gson gson = new Gson();
    Clase clas =gson.fromJson(respuesta.body(), Clase.class);
    gasolineras = clas.getGasol();
    System.out.println("ESTE ES RESULTADO " + clas.gasol.size());
    System.out.println("ESTE ES RESULTADO " + gasolineras.size());
    System.out.println("Que filtro quiere codigo postal, pueblo o provincia (Cualquier otro caracter)");
    String cp = x.nextLine().toLowerCase();
    if (cp.equals("codigo postal") ) {
        System.out.println("Codigo Postal en el que se desea buscar");
        String cod = x.nextLine();
        for(int i=0;i<gasolineras.size();i++){
            if( gasolineras.get(i).CP.equals(cod)) {
                System.out.println("Codigo Postal " +gasolineras.get(i).getCP());
                System.out.println("Pueblo " +gasolineras.get(i).getMunicipio());
                System.out.println("Marca " +gasolineras.get(i).getRotulo());
                System.out.println("Direccion " +gasolineras.get(i).getDireccion());
                System.out.println("Horario " +gasolineras.get(i).getHorario());
                System.out.println("Precio Diesel A " +gasolineras.get(i).getPrecioGasoleoA());
                System.out.println("Precio Diesel B " +gasolineras.get(i).getPrecioGasoleoB());
                System.out.println("Precio Diesel Premium " +gasolineras.get(i).getPrecioGasoleoPremium());
                System.out.println("Precio Gasolina 95 " +gasolineras.get(i).getPrecioGasolina95E5());
                System.out.println("Precio Gasolina 95 Premium " +gasolineras.get(i).getPrecioGasolina95E5Premium());
                System.out.println("Precio Gasolina 98 " +gasolineras.get(i).getPrecioGasolina98E5());
                System.out.println("¿Quieres ir a esa Gasolinera?");
                String ir = x.nextLine().toLowerCase();
                if(ir.equals("si")) {
                    String url = "https://www.google.es/maps/place/" + gasolineras.get(i).getLatitud().replace(",",".") + "," + gasolineras.get(i).getLongitud().rep.
                    Desktop enlace=Desktop.getDesktop();
                    try {
                        enlace.browse(new URI(url));
                    }
                }
            }
        }
    }
}

```

Resultados obtenidos

Cuando se ejecuta el servidor te muestra que se ha encendido y el puerto que usa

```
C:\Users\oscar\Desktop\crudoil exe>java -jar servercrudoil.jar
Servidor HTTP arranca en el puerto 7777
```

Cuando se ejecuta la aplicación te dice cuántas gasolineras hay y te pregunta cómo deseas filtrar. Después te consulta sobre la elección elegida y te muestra las opciones y te pregunta si quieres ir a la gasolinera para abrir el [link](#).

```
C:\Users\oscar\Desktop\crudoil exe>java -jar crudoil.jar
ESTE ES RESULTADO 11697
ESTE ES RESULTADO 11697
Que filtro quiere codigo postal, pueblo o provincia (Cualquier otro caracter)
codigo postal
Codigo Postal en el que se desea buscar
46470
Codigo Postal 46470
Pueblo Albal
Marca CEPSA
Direccion AVENIDA PADRE CARLOS FERRIS, 109
Horario L-D: 06:30-21:30
Precio Diesel A 1,909
Precio Diesel B
Precio Diesel Premium 1,975
Precio Gasolina 95 1,907
Precio Gasolina 95 Premium
Precio Gasolina 98 1,964
¿Quieres ir a esa Gasolinera?

Codigo Postal 46470
Pueblo Albal
Marca BIOMAR OIL SL
Direccion CALLE OCHO, 7
Horario L-D: 24H
Precio Diesel A 1,786
Precio Diesel B
Precio Diesel Premium
Precio Gasolina 95 1,836
Precio Gasolina 95 Premium
Precio Gasolina 98
¿Quieres ir a esa Gasolinera?
```

Qr y link de descarga



<https://drive.google.com/file/d/1YPGOsKcno8czhTBG0qvS2sRNA9j4b8Qz/view?usp=sharing>

Conclusiones

Crudoil es una aplicación en Java que te ayuda a elegir la gasolinera de tu zona e incluso te envía a google maps. Se ayuda de un servidor Api para recibir la información. Puedes buscar por tu pueblo o provincia lo que te ahorra tiempo.

Posibles actualizaciones

La principal futura actualización es crear una interfaz para poder facilitar una interacción al usuario.



Al tener una Api propia otra posible actualización es hacer una página web para que se pueda acceder desde cualquier plataforma.

Una actualización a la app actual sería poder ordenar por precio después de filtrar las gasolineras.

Bibliografía

Moya, R. (17 de mayo de 2017). *Gson (JSON) en Java, con ejemplos*

<https://jarroba.com/gson-json-java-ejemplos/>

Arlandy Rodríguez M. (17 de septiembre de 2012) Jugando con JSON en Java y la

librería Gson <https://www.adictosaltrabajo.com/2012/09/17/gson-java-json/>

Ureña Gómez F. (8 de octubre de 2014) Abrir un enlace desde nuestra aplicación en Java

<https://www.discoduroderoer.es/abrir-un-enlace-desde-nuestra-aplicacion-en-java/>

Sanz Requena R. (2021) DAM - Programación de servicios y procesos Tema 5 - Generación de servicios en red

https://www.floridaoberta.com/pluginfile.php/807920/mod_resource/content/1/Tema%205%20-%20Generacion%20de%20servicios%20en%20red.pdf

Sanz Requena R. (2021) DAM - Accés a dades Tema 4 - Servicis i recursos web

https://www.floridaoberta.com/pluginfile.php/807908/mod_resource/content/1/Tema%204%20-%20Servicis%20i%20recursos%20web.pdf

Ministerio de Industria, Turismo y Comercio (1 de julio de 2007) Orden

ITC/2308/2007, de 25 de julio, por la que se determina la forma de remisión de información al Ministerio de Industria, Turismo y Comercio sobre las actividades de suministro de productos petrolíferos.

https://www.boe.es/diario_boe/txt.php?id=BOE-A-2007-14592