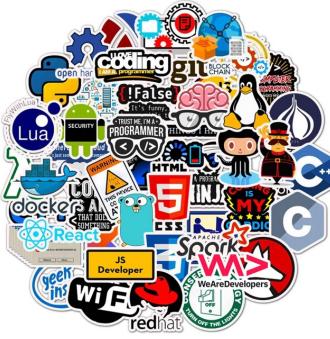




Quem se prepara, não para.



Linguagens de Programação

3º período

Prof. Dr. João Paulo Aramuni

Sumário

- Java
 - Serialização com Arquivos JSON

Sumário

- Java
 - **Serialização com Arquivos JSON**

- Conceito de Arquivos JSON
 - **JSON** é o acrônimo de JavaScript Object Notation, e trata-se de um subconjunto da linguagem JavaScript, porém não se limita a ela, apesar de usar a mesma sintaxe.
 - Esse formato foi pensado para implementação de intercâmbio de dados, assim como o XML.

- Conceito de Arquivos JSON
 - A principal vantagem do JSON sobre o XML está no fato de o JSON ser mais leve e muito mais simples de ser lido e entendido (SILVA, SILVA JÚNIOR, 2018).

```
<?xml version= "1.0"encoding= "UTF-8"?>
<id>1</id> <nome> João Maria</nome>
<endereco> Fortaleza </endereco>
```

XML

```
{"id":1,"nome":"João
Maria","endereco":"Fortaleza"}
```

JSON

- Conceito de Arquivos JSON

- É possível perceber, nesse exemplo simples, por que o JSON é mais simples que o XML (TONSIG, 2008).
- Além disso, o JSON tem as seguintes características marcantes:
 - Possui parsing mais fácil;
 - Consegue suportar classes e objetos;
 - É mais rápido na execução e no transporte de dados;
 - Cria arquivos menores;
 - Não é uma linguagem de marcação.
 - Não é um conjunto de esquemas de validação;
 - É atualmente utilizado por Google, Facebook, Amazon, Microsoft, etc.

- Conceito de Arquivos JSON
 - O JSON é constituído das seguintes estruturas:
 - 1) Uma coleção de pares nome/valor – em várias linguagens, isso é chamado de objetc, record, struct, dicionário, hash table, keyed list ou arrays associativas (os identificadores em JSON devem estar definidos entre aspas, podendo ser simples ou duplas);
 - 2) Uma lista ordenada de valores – na maioria das linguagens, isso é caracterizado como uma array, vetor, lista ou sequência.

- Conceito de Arquivos JSON

Quadro 1. Diferenças básicas entre JSON e XML

XML	JSON
É orientada a documento.	É orientada a objeto.
O dado tem maior segurança.	O dado tem menor segurança.
Arquivos menos legíveis.	Arquivos mais legíveis.
Exibição simplificada para o navegador Web descrever.	Não pode ser exibido no navegador sem análise.

- Exemplo JSON
- Vamos ver como seria a definição da sintaxe do formato JSON para representar um menu de um restaurante com 10 pratos principais:

```
{  
    "type": "Restaurante Menu",  
    "restaurante-info": {  
        "id": "121721",  
        "sponsored": "0",  
        "user-favourite": "",  
        "name": "Restaurante Brasil",  
        "address": "Rio de Janeiro, Copacabana, Q.10",  
        "rating": "0.00",  
        "minimum-order-price": "500.00",  
        "delivery-fee": "100.00",  
        "avg _ delivery _ time": "60",  
        "open-status": "Aberto",  
        "default-ad": "",  
        "value": null,  
        "total-category": "10",  
        "site-url": "www.restaurantebrasil.com.br",  
        "location-img": "",  
        "restaurant-url": "/restaurantebrasil/121721/menu",  
    }  
}
```

- Exemplo JSON
- Vamos ver como seria a definição da sintaxe do formato JSON para representar um menu de um restaurante com 10 pratos principais:

```
"payment-methods": [  
    {  
        "id": "10",  
        "type": "Cartão de débito"  
    },  
    {  
        "id": "30",  
        "type": "Cartão de crédito"  
    },  
    {  
        "id": "40",  
        "type": "Pagamento em dinheiro"  
    }],  
    "delivery-types": "0",  
    "tags": []
```

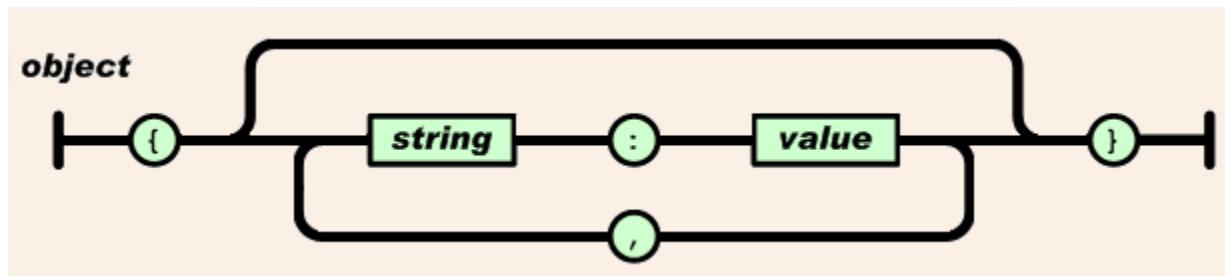
Java

- Exemplo JSON
- Vamos ver como seria a definição da sintaxe do formato JSON para representar um menu de um restaurante com 10 pratos principais:

```
"Menu-Pratos": [  
    {  
        "id": "1",  
        "name": "Feijoada"  
    },  
    {  
        "id": "2",  
        "name": "Macarrão com camarão internacional"  
    },  
    {  
        "id": "3",  
        "name": "Peixe nordestino assado"  
    },  
    {  
        "id": "4",  
        "name": "Picanha gaúcha à moda da casa"  
    },  
    {  
        "id": "5",  
        "name": "Baião de dois à mineira"  
    },  
    {  
        "id": "6",  
        "name": "Farofa de banana da terra"  
    },  
    {  
        "id": "7",  
        "name": "Carne de sol do Cerrado"  
    },  
    {  
        "id": "8",  
        "name": "Tutu à mineira"  
    },  
    {  
        "id": "9",  
        "name": "Galinha com pequi"  
    },  
    {  
        "id": "10",  
        "name": "Bife acebolado no arroz grego"  
    }]
```

- Estrutura JSON

- A estrutura básica JSON para representar um simples objeto é um conjunto desordenado de pares nome/valor.
- Um objeto começa com { (chave esquerda) e termina com } (chave direita).
- Cada nome é seguido por : (dois pontos) e os pares nome/valor são separados por , (vírgula). A figura abaixo mostra a estrutura de um objeto em JSON.



- Ferramentas
 - **JSONFormatter.io** é uma ferramenta on-line para formatar, analisar, visualizar, validar, editar e embelezar dados JSON em tempo real.
 - Seu editor JSON poderoso, rico em recursos e fácil de usar oferece uma experiência de edição nativa.
 - <https://www.jsonformatter.io/>

- Ferramentas

JSONFormatter.io

Upload (.json/.txt) Clear Live format

Download Copy Minify Prettify

```
1 {  
2   "type": "Restaurante Menu",  
3   "restaurante-info": {  
4     "id": "121721",  
5     "sponsored": "0",  
6     "user-favourite": "",  
7     "name": "Restaurante Brasil",  
8     "address": "Rio de Janeiro, Copacabana, Q.10",  
9     "rating": "0.00",  
10    "minimum-order-price": "500.00",  
11    "delivery-fee": "100.00",  
12    "avg_delivery_time": "60",  
13    "open-status": "Aberto",  
14    "default-ad": "",  
15    "value": null,  
16    "total-category": "10",  
17    "site-url": "www.restaurantebrazil.com.br",  
18    "location-img": "",  
19    "restaurant-url": "/restaurantebrazil/121721/menu",  
20    "payment-methods": [  
21      {  
22        "id": "10",  
23        "type": "Cartão de débito"  
24      },  
25      {  
26        "id": "30",  
27        "type": "Cartão de crédito"  
28      }]
```

1 {
2 "type": "Restaurante Menu",
3 "restaurante-info": {
4 "id": "121721",
5 "sponsored": "0",
6 "user-favourite": "",
7 "name": "Restaurante Brasil",
8 "address": "Rio de Janeiro, Copacabana, Q.10",
9 "rating": "0.00",
10 "minimum-order-price": "500.00",
11 "delivery-fee": "100.00",
12 "avg_delivery_time": "60",
13 "open-status": "Aberto",
14 "default-ad": "",
15 "value": null,
16 "total-category": "10",
17 "site-url": "www.restaurantebrazil.com.br",
18 "location-img": "",
19 "restaurant-url": "/restaurantebrazil/121721/menu",
20 "payment-methods": [
21 {
22 "id": "10",
23 "type": "Cartão de débito"
24 },
25 {
26 "id": "30",
27 "type": "Cartão de crédito"
28 }]

- JSON Minificado

- Minificar o JSON é um processo que visa reduzir o tamanho dos dados em um arquivo JSON, removendo todos os espaços em branco, caracteres de tabulação, quebras de linha e outros caracteres desnecessários.
- O objetivo principal da minificação é economizar espaço de armazenamento ou reduzir o tempo de transferência de dados pela rede.

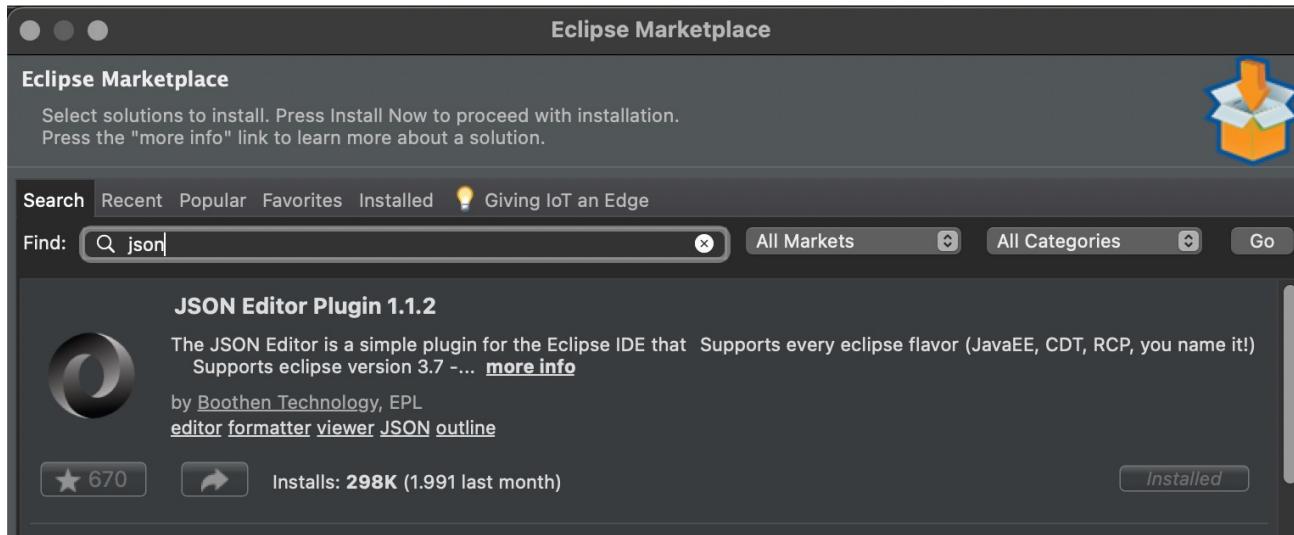
- JSON Minificado



```
Download Copy Minify Prettify

{
  "type": "Restaurante Menu",
  "restaurante-info": {
    "id": "121721",
    "sponsored": "0",
    "user-favourite": "",
    "name": "Restaurante Brasil",
    "address": "Rio de Janeiro, Copacabana, Q.10",
    "rating": "0.00",
    "minimum-order-price": "500.00",
    "delivery-fee": "100.00",
    "avg _ delivery _ time": "60",
    "open-status": "Aberto",
    "default-ad": "",
    "value": null,
    "total-category": "10",
    "site-url": "www.restaurantebrasil.com.br",
    "location-img": "",
    "restaurant-url": "/restaurantebrasil/121721/menu",
    "payment-methods": [
      {"id": "10", "type": "Cartão de débito"},
      {"id": "30", "type": "Cartão de crédito"},
      {"id": "40", "type": "Pagamento em dinheiro"}
    ],
    "delivery-types": "0",
    "tags": [],
    "Menu-Pratos": [
      {"id": "2", "name": "Macarrão com camarão internacional"},
      {"id": "3", "name": "Peixe nordestino assado"},
      {"id": "4", "name": "Picanha gaúcha à moda da casa"},
      {"id": "5", "name": "Baião de dois à mineira"},
      {"id": "6", "name": "Farofa de banana da terra"},
      {"id": "7", "name": "Carne de sol do Cerrado"},
      {"id": "8", "name": "Tutu à mineira"},
      {"id": "9", "name": "Galinha com pequi"},
      {"id": "10", "name": "Bife acebolado no arroz grego"}
    ]
  }
}
```

- Ferramentas
 - JSON Editor Plugin - para Eclipse
 - Editor simples para melhor visualizar os arquivos JSON no Eclipse.



- Ferramentas
 - Biblioteca **JSON**
 - Gson é uma biblioteca Java que pode ser usada para converter objetos Java em sua representação JSON. Esta biblioteca foi desenvolvida a partir de um projeto Google.
 - Também pode ser usada para converter uma string JSON em um objeto Java equivalente. O Gson pode trabalhar com objetos Java arbitrários, incluindo objetos pré-existentes dos quais você não possui o código-fonte.

Ferramentas

- Biblioteca **JSON**

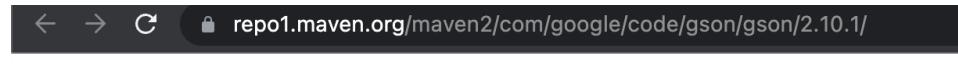
- <https://github.com/google/gson>
- <https://central.sonatype.com/artifact/com.google.code.gson/gson/2.10.1/overview?smo=true>

The screenshot shows the Sonatype Maven Central Repository interface. At the top, there's a search bar with 'Search' placeholder text and a dropdown menu set to '2.10.1'. Below the search bar, it says 'Used in 27002 components'. A yellow button labeled 'pkg:maven/com.google.code.gson/gson@2.10.1' has a copy icon next to it. The main navigation tabs are 'Overview' (which is active), 'Versions', 'Dependents', and 'Dependencies'. The 'Overview' section contains an 'Overview' box for the 'Gson JSON library' and a 'Snippets' box showing Maven dependency code. The 'Snippets' box includes a 'Copy to clipboard' button and a dropdown for 'Apache Maven'. The right sidebar features a 'Sonatype Safety Rating' section with a score of '3 out of 10' and a progress bar, along with a link to 'How did we get this score?'. It also shows 'OSS Index No vulnerabilities' with a 'View' link. The bottom right of the sidebar has a 'Metadata' section with details like '6 months ago', 'Licenses Apache-2.0', and file size '12.3 kB'.

- Biblioteca **JSON**

- 1º Passo – Download do Arquivo .JAR.

- <https://repo1.maven.org/maven2/com/google/code/gson/gson/2.10.1/>



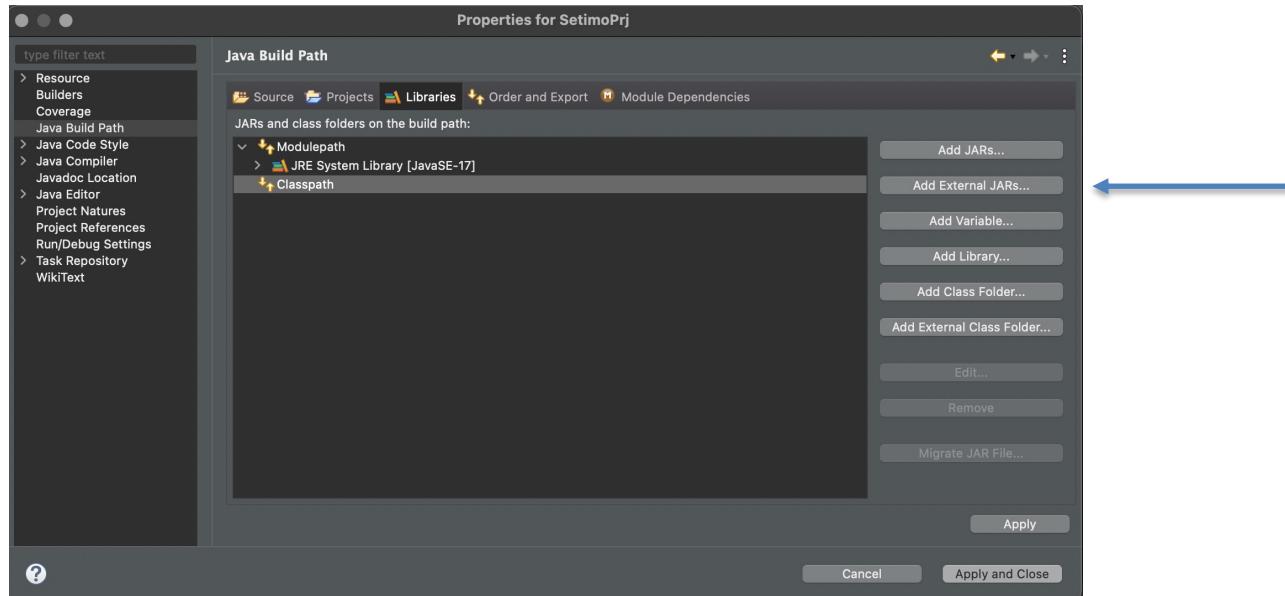
com/google/code/gson/gson/2.10.1

.. /			
gson-2.10.1-javadoc.jar	2023-01-06 15:44	706662	
gson-2.10.1-javadoc.jar.asc	2023-01-06 15:44	496	
gson-2.10.1-javadoc.jar.md5	2023-01-06 15:44	32	
gson-2.10.1-javadoc.jar.sha1	2023-01-06 15:44	40	
gson-2.10.1-sources.jar	2023-01-06 15:44	181840	
gson-2.10.1-sources.jar.asc	2023-01-06 15:44	496	
gson-2.10.1-sources.jar.md5	2023-01-06 15:44	32	
gson-2.10.1-sources.jar.sha1	2023-01-06 15:44	40	
gson-2.10.1.jar	2023-01-06 15:44	283367	
gson-2.10.1.jar.asc	2023-01-06 15:44	496	
gson-2.10.1.jar.md5	2023-01-06 15:44	32	
gson-2.10.1.jar.sha1	2023-01-06 15:44	40	
gson-2.10.1.pom	2023-01-06 15:44	9365	
gson-2.10.1.pom.asc	2023-01-06 15:44	496	
gson-2.10.1.pom.md5	2023-01-06 15:44	32	
gson-2.10.1.pom.sha1	2023-01-06 15:44	40	



Ferramentas

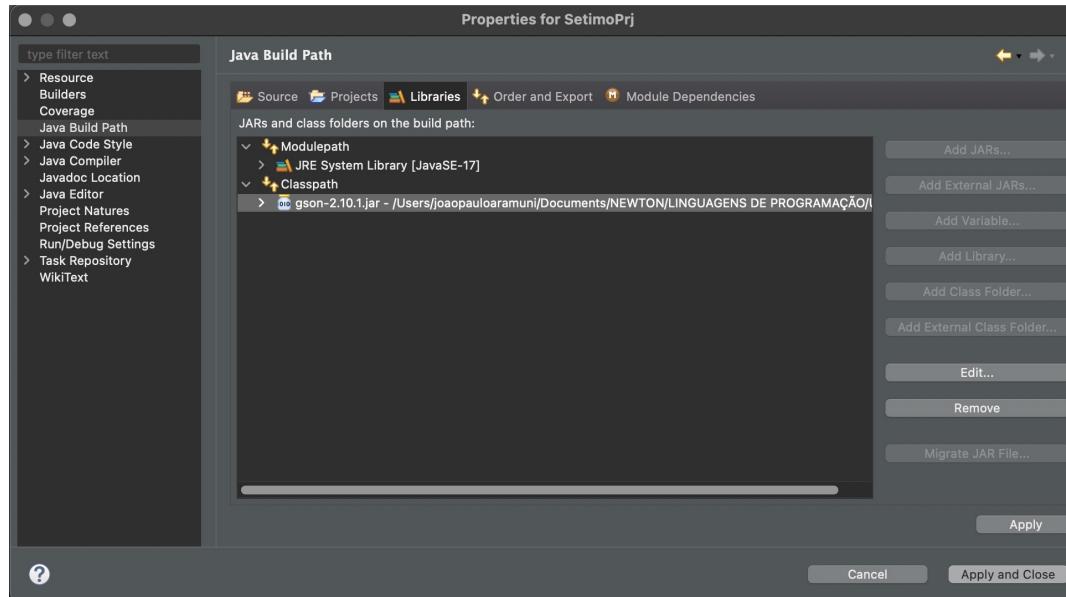
- Biblioteca **JSON**
 - 2º Passo – Criar um novo projeto em JAVA.
 - Clicar com o botão direito, Propriedades, escolher Java Build Path, clicar em Classpath e no botão Add External JARs.



Ferramentas

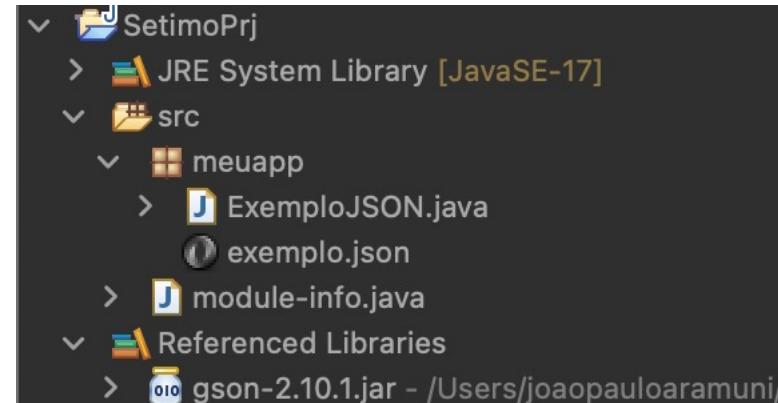
- Biblioteca **JSON**

- 2º Passo – Criar um novo projeto em JAVA.
 - Clicar com o botão direito, Propriedades, escolher Java Build Path, clicar em Classpath e no botão Add External JARs.



- Biblioteca **JSON**

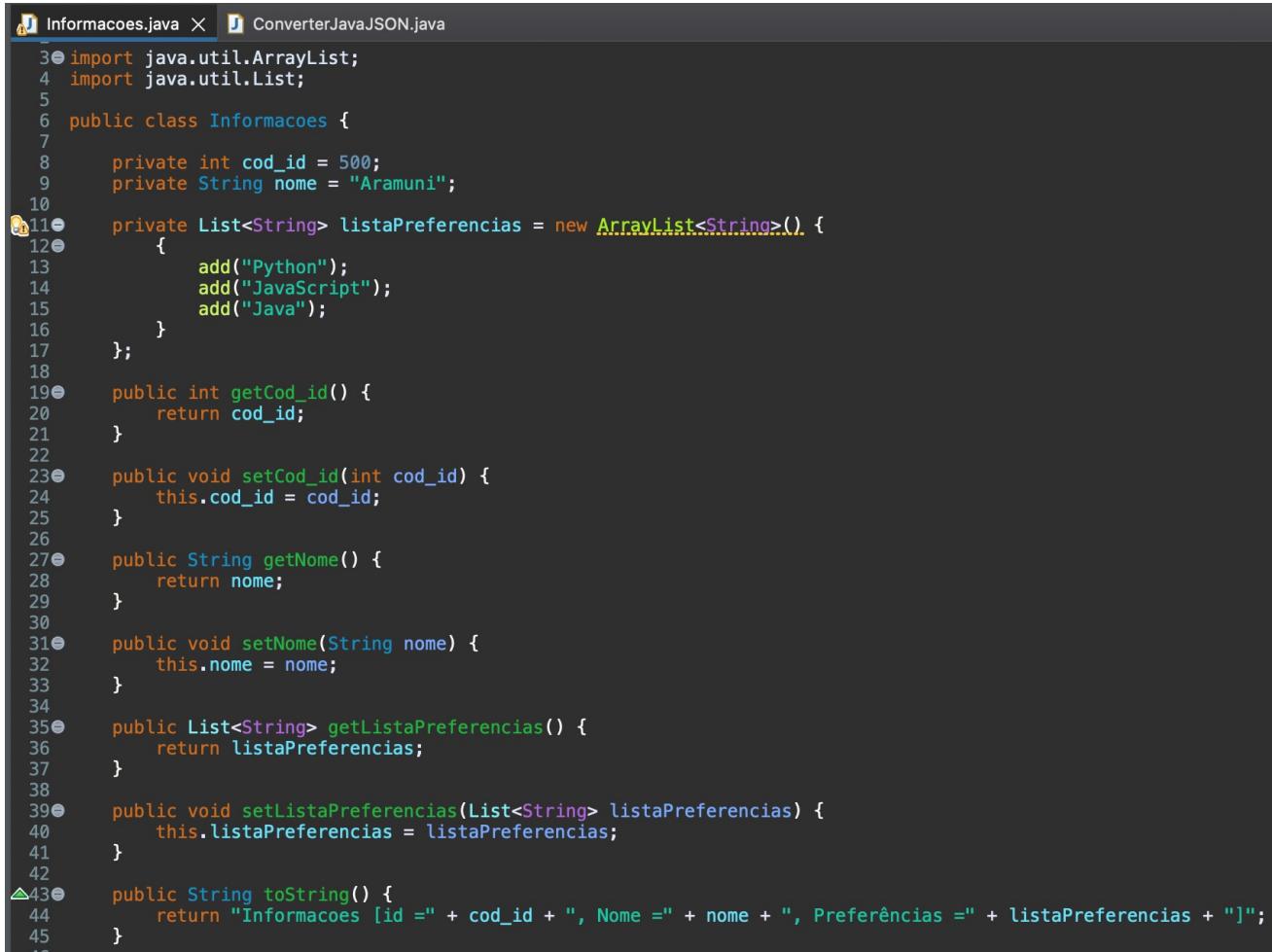
- 2º Passo – Criar um novo projeto em JAVA.
 - Clicar com o botão direito, Propriedades, escolher Java Build Path, clicar em Classpath e no botão **Add External JARs**.



Ferramentas

- Biblioteca **JSON**

3º Passo – Criar um nova classe que será responsável pelas informações do arquivo JSON.

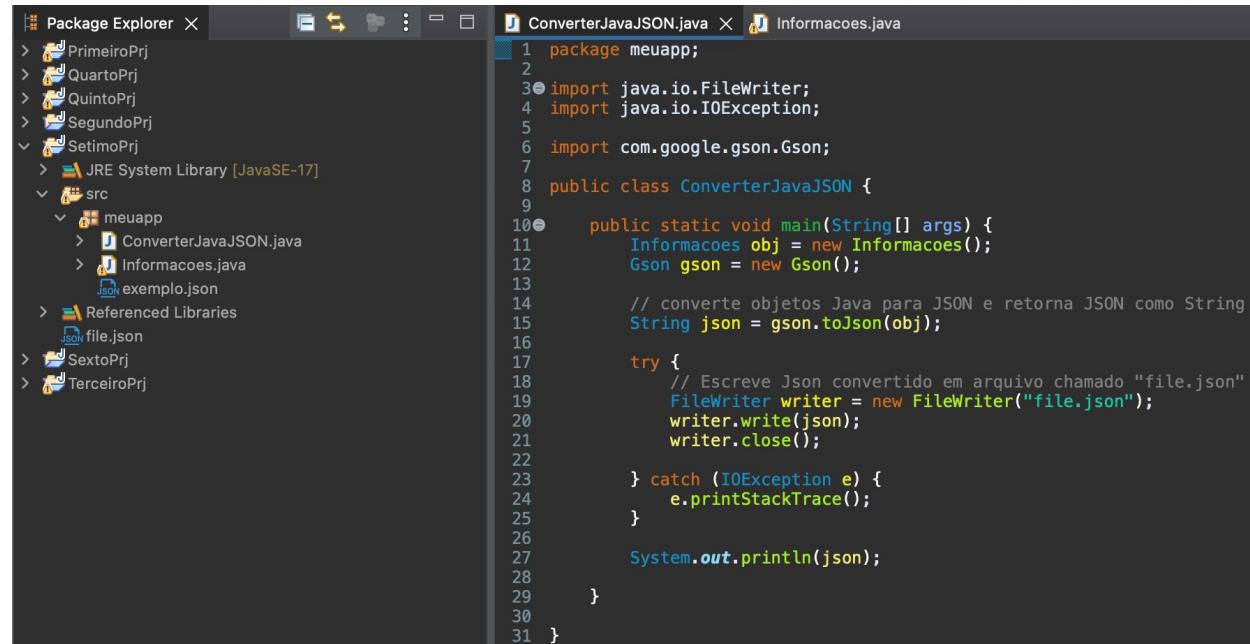


```
1 Informacoes.java X 2 ConverterJavaJSON.java
3 import java.util.ArrayList;
4 import java.util.List;
5
6 public class Informacoes {
7
8     private int cod_id = 500;
9     private String nome = "Aramuni";
10
11    private List<String> listaPreferencias = new ArrayList<String>();
12    {
13        add("Python");
14        add("JavaScript");
15        add("Java");
16    }
17 };
18
19    public int getCod_id() {
20        return cod_id;
21    }
22
23    public void setCod_id(int cod_id) {
24        this.cod_id = cod_id;
25    }
26
27    public String getNome() {
28        return nome;
29    }
30
31    public void setNome(String nome) {
32        this.nome = nome;
33    }
34
35    public List<String> getListaPreferencias() {
36        return listaPreferencias;
37    }
38
39    public void setListaPreferencias(List<String> listaPreferencias) {
40        this.listaPreferencias = listaPreferencias;
41    }
42
43    public String toString() {
44        return "Informacoes [id =" + cod_id + ", Nome =" + nome + ", Preferências =" + listaPreferencias + "]";
45    }
46
```

Ferramentas

- Biblioteca **JSON**

4º Passo – Criar um nova classe
que irá converter Java para
JSON, incluir o main.



The screenshot shows the Eclipse IDE interface. On the left, the Package Explorer view displays several Java projects: PrimeiroPrj, QuartoPrj, QuintoPrj, SegundoPrj, SetimoPrj (selected), JRE System Library [JavaSE-17], and SextoPrj, TerceiroPrj. Inside the SetimoPrj project, there is a src folder containing a meuapp package with files like ConverterJavaJSON.java, Informacoes.java, and exemplo.json, along with a Referenced Libraries folder containing file.json. On the right, the Editor view shows the code for ConverterJavaJSON.java:

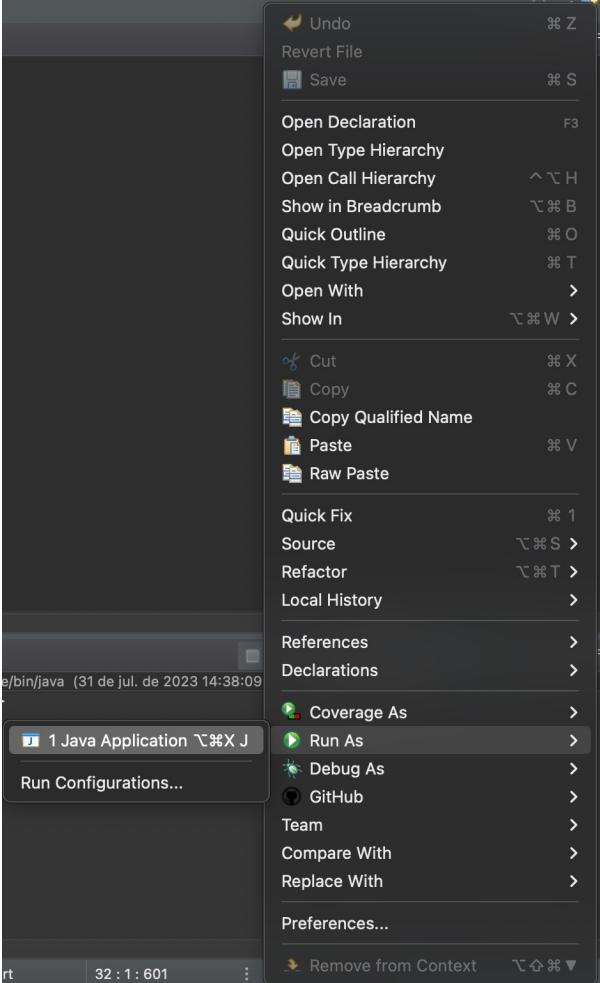
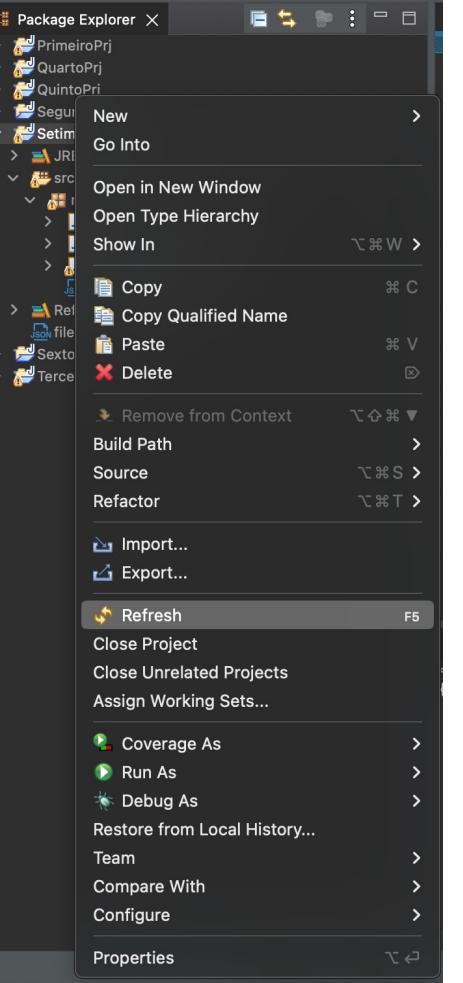
```
1 package meuapp;
2
3 import java.io.FileWriter;
4 import java.io.IOException;
5
6 import com.google.gson.Gson;
7
8 public class ConverterJavaJSON {
9
10    public static void main(String[] args) {
11        Informacoes obj = new Informacoes();
12        Gson gson = new Gson();
13
14        // converte objetos Java para JSON e retorna JSON como String
15        String json = gson.toJson(obj);
16
17        try {
18            // Escreve Json convertido em arquivo chamado "file.json"
19            FileWriter writer = new FileWriter("file.json");
20            writer.write(json);
21            writer.close();
22
23        } catch (IOException e) {
24            e.printStackTrace();
25        }
26
27        System.out.println(json);
28
29    }
30
31 }
```

Ferramentas

- Biblioteca **JSON**

4º Passo – Rode para ver o arquivo **file.json** ser criado.

Pode ser necessário clicar com o botão direito – **refresh** – no projeto.



Ferramentas

- Biblioteca **JSON**

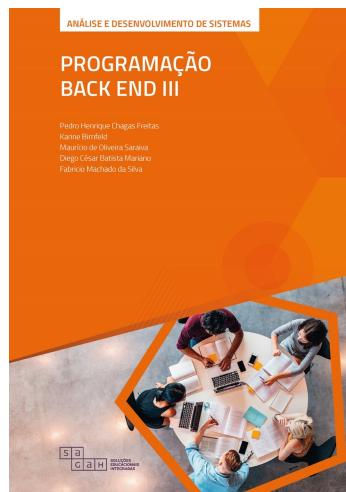
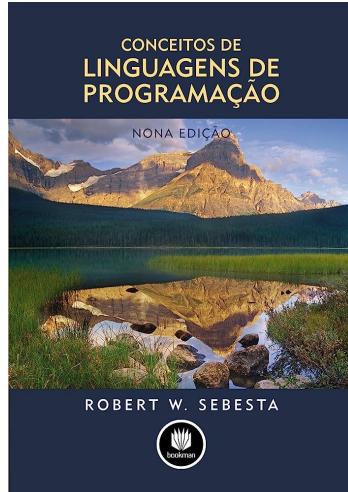
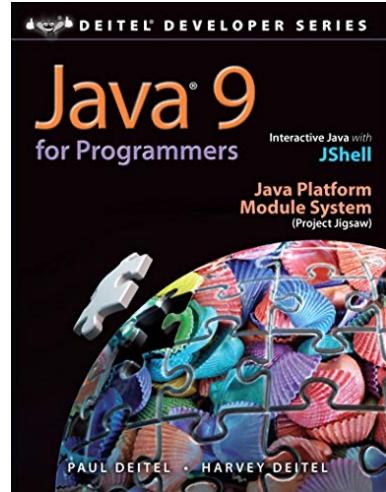
5º Passo – Criar um nova classe que irá converter JSON para Java, incluir o main.

The screenshot shows an IDE interface with three tabs at the top: 'ConverterJavaJSON.java', 'Informacoes.java', and 'ConverterJSONJava.java'. The 'ConverterJSONJava.java' tab is active, displaying the following Java code:

```
1 package meuapp;
2
3 import java.io.BufferedReader;
4 import java.io.FileReader;
5 import java.io.IOException;
6 import com.google.gson.Gson;
7
8 public class ConverterJSONJava {
9
10    public static void main(String[] args) {
11        Gson gson = new Gson();
12
13        try {
14
15            BufferedReader br = new BufferedReader(new FileReader("file.json"));
16
17            //Converte String JSON para objeto Java
18            Informacoes obj = gson.fromJson(br, Informacoes.class);
19
20            System.out.println(obj);
21
22        } catch (IOException e) {
23            e.printStackTrace();
24        }
25
26    }
27
28 }
```

At the bottom of the screen, there are several tabs: 'Console X', 'Problems', 'Javadoc', 'Declaration', 'Search', and 'Breakpoints'. Below these tabs, the status bar displays the path: '<terminated> ConverterJSONJava [Java Application] /Library/Java/JavaVirtualMachines/jdk-17.jdk/Contents/Home/bin/java'. The status bar also shows the output of the program: 'Informacoes [id =500, Nome =Aramuni, Preferências =[Python, JavaScript, Java]]'.

Referências





Obrigado!

joaoaramuni@newtonpaiva.br