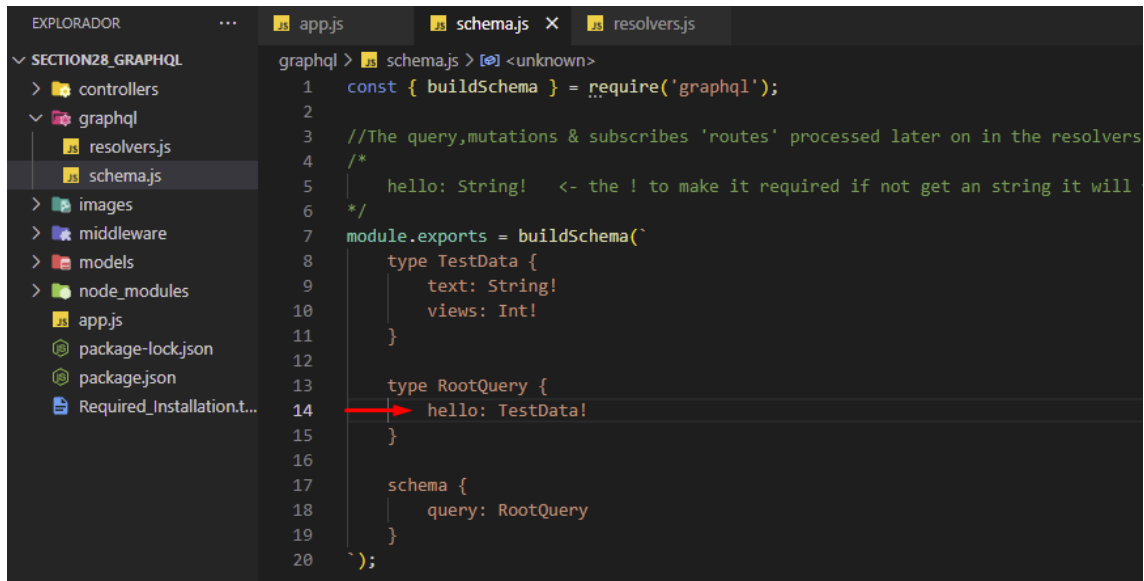


GraphQL System

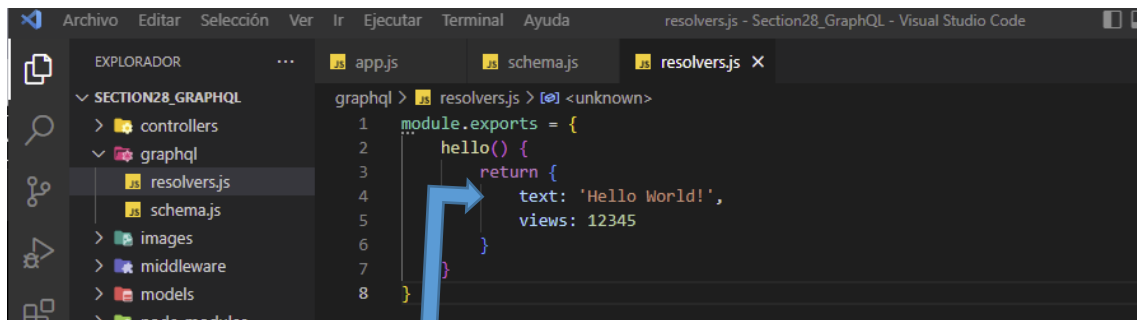
Schema:



The screenshot shows the Visual Studio Code editor with the 'EXPLORADOR' sidebar on the left displaying the project structure. The 'graphql' folder is expanded, showing 'resolvers.js' and 'schema.js'. The 'schema.js' file is open in the editor, showing the following code:

```
1 const { buildSchema } = require('graphql');
2
3 //The query,mutations & subscribes 'routes' processed later on in the resolvers
4 /*
5   hello: String! <- the ! to make it required if not get an string it will
6 */
7 module.exports = buildSchema(`
8   type TestData {
9     text: String!
10    views: Int!
11  }
12
13  type RootQuery {
14    hello: TestData!
15  }
16
17  schema {
18    query: RootQuery
19  }
20 `);
```

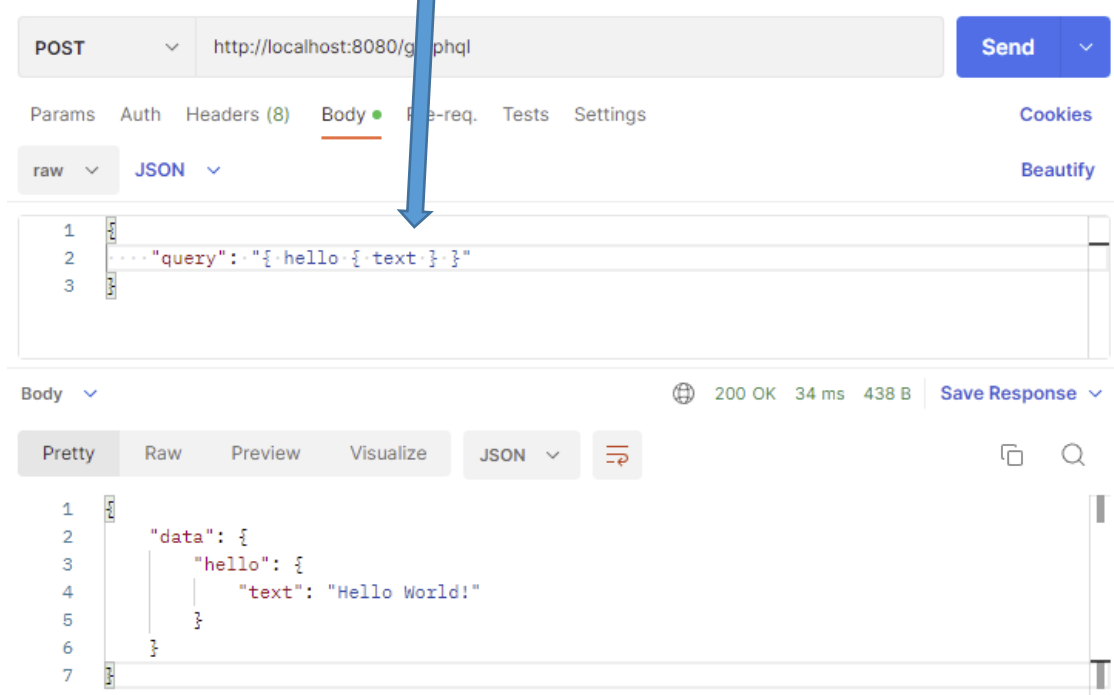
'Controller':



The screenshot shows the Visual Studio Code editor with the 'EXPLORADOR' sidebar on the left displaying the project structure. The 'graphql' folder is expanded, showing 'resolvers.js' and 'schema.js'. The 'resolvers.js' file is open in the editor, showing the following code:

```
1 module.exports = {
2   hello() {
3     return {
4       text: 'Hello World!',
5       views: 12345
6     }
7   }
8 }
```

Data obtained:



The screenshot shows a REST client interface with a POST request to 'http://localhost:8080/graphql'. The request body is a JSON object: `{ "query": "{ hello { text } }" }`. The response is a 200 OK status with a response time of 34 ms and a size of 438 B. The response body is a JSON object: `{ "data": { "hello": { "text": "Hello World!" } } }`.