

# Technical Instance <> .NET

.NET Practical Test from CodigoDelSur:

## API Creation and Consumption

- Objective: Develop a .NET application that interacts with an external API and exposes four of its endpoints.
- General Requirements:
- Language and Technologies:

Use C# as the programming language and .NET 7 or 8.

Use ASP.NET for API creation.

Implement Entity Framework Core for any database interactions or justify the use of another library or framework.

- External API:

Connect to an external API of your choice (e.g., OpenWeatherMap, The Movie Database) to retrieve data that will be used by your application.

Own Endpoints:

- Create the following four endpoints in your API:
  - 1.Registration: An endpoint for user registration.
  - 2.Authorization: An endpoint for user authentication. You must implement JWT tokens for session management or another mechanism, to justify your choice.
  - 3.Data Query: An endpoint that consumes data from the selected external API and presents it appropriately for the end-user, **MUST IMPLEMENT AT LEAST 2 FILTERS**.
  - 4.Personalization: An endpoint that allows users to save preferences or perform specific actions that directly impact data querying (endpoint 3), e.g., saving searched movies, and setting weather alerts.



**CodigoDelSur**

- Security:

Ensure endpoints 3 and 4 verify that the request comes from an existing user in the system.

- Tests:

At least 2 unit tests are required for any of the above endpoints. It's important to note that being unit tests, they do not need to cover the entire flow. The use of MSTests or a similar library is recommended.

- Documentation:

Document all endpoints using Swagger/OpenAPI to facilitate their use and testing.

Deliverable, The deliverable will consist of a link to a GitHub repository containing the following:

1. Provide the complete source code of the project, including all configuration files, models, controllers, and migrations that you consider necessary for a third party to run the application locally.
2. Implementation Instructions. A README detailing the steps to set up and run the application, including the necessary configuration to connect to the external API at a basic level.
3. API Documentation. A link to the API documentation generated with Swagger/OpenAPI, where the available endpoints, parameters, response formats, and HTTP status codes are described, which can be local.

- Evaluation Criteria:

- Functionality: The application meets all specified requirements and functions as expected.
- Code Quality: The code is clean, well-organized, handles exceptions, and follows .NET development best practices.
- Security: Security measures are properly implemented, especially regarding authentication and authorization.



**CodigoDelSur**

- Documentation: The documentation is clear, complete and facilitates the understanding and use of the API.
- Technology Proficiency: Demonstrates command of .NET technologies.

Feel free to ask any questions you think are necessary. You will have 48 hours from the time the project is delivered.

**Good luck!**



**CodigoDelSur**