# ARCHITECTURE PROPOSAL.

1. **Database**

SQL Server database called “Healthcare”.

A secured connection to a failover system is recommended for high availability and fault tolerance.

1. **Back-end**

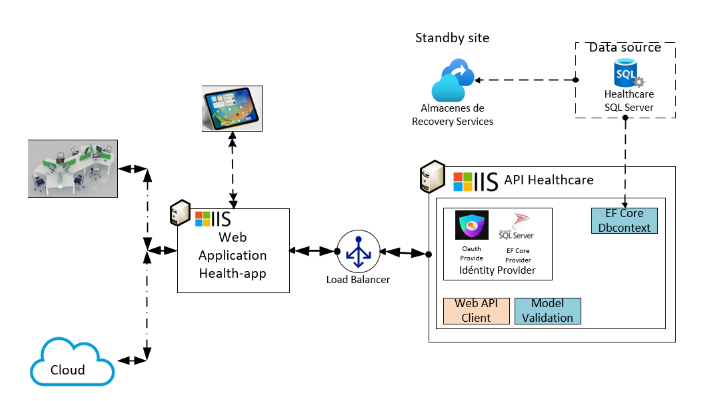
The back-end called “HealthAPI” is deployed in IIS on its own server and is protected with domain security policies. The Api service is implemented using .NET Framework 9 and is divided into the following layers:

**HealthcareAPI**: Controls client requests and operates as the presentation layer of the REST API. It includes a method to store traces and errors in a physical path called Log.

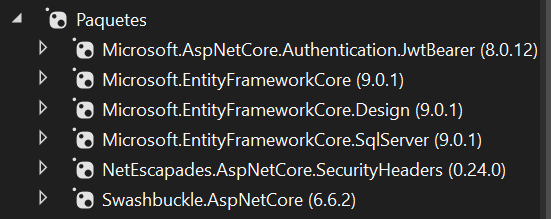
**Healthcare.BR**: It has business logic and validations.

**Healthcare.Data**: It has direct connection to the database.

**Healthcare.Common**: Cross-cutting layer for common utilities.



* 1. **Execution Of The Code**
* Open Visual Studio 2022
* Open the Healthcare.sln solution deliverd in the path “[Healthcare\_API](https://github.com/larcila/Healthcare_API/tree/main)/[DEV](https://github.com/larcila/Healthcare_API/tree/main/DEV)/Healtcare/” on Github.
* You must make sure that the packages are installed, if not, enter the Nuget package manager and install them.



* In appsetting.json configure the following:

ConnectionStrings: modifies the connection string with the database instance.

LogFilePath: Set the path where the error log will be stored.

1. **Font-end**

Web application develope**d in Angular. It consumes the back-end APIs through the load balancer. The load balancer** is recommended in case of high demand, this includes having another server for the API service.

The web application is deployed on another IIS server keeping separate back-end and front-end resources.

To conclude, you keep in mind that the separation of functions not only increases costs but also increases the complexity of configuration and maintenance. It is necessary to say that handling separate servers by functionality in addition to the load balancer ensures that the system can handle high traffic demands and gives the possibility to add more servers in case of growth.

* 1. **Execution Of The Code**

Before opening the project, make sure that it is installed:

|  |
| --- |
| Node.js y npm |

You must make sure you have the following version of angular installed

|  |
| --- |
| Angular CLI: 19.1.4  Node: 22.13.0  Package Manager: npm 11.0.0  OS: win32 x64  Angular: 19.1.3  ... animations, common, compiler, compiler-cli, core, forms  ... platform-browser, platform-browser-dynamic, router  Package Version  ---------------------------------------------------------  @angular-devkit/architect 0.1901.4  @angular-devkit/build-angular 19.1.4  @angular-devkit/core 19.1.4  @angular-devkit/schematics 19.1.4  @angular/cdk 19.1.1  @angular/cli 19.1.4  @angular/material 19.1.1  @schematics/angular 19.1.4  rxjs 7.8.1  typescript 5.7.3  zone.js 0.15.0 |

The Angular Material component installed is as follows:

|  |
| --- |
| @angular/material@19.1.1 |

Download the project from the Github repository “Healthcare\_API/DEV/HealthApp/”.

Open the project using Visual Studio Code

Install dependencies.

|  |
| --- |
| npm install |

With the above, the project would be ready for use.