An In-Depth Guide to Your Application's Technology Stack

This document provides a detailed explanation of the technologies used in your application, covering the back-end, database, front-end, and cloud infrastructure. Each section explores the core concepts and practical applications of the respective technologies.

Back-End Technologies

The back-end of your application is powered by a robust set of technologies that handle business logic, data access, and background processing. This section delves into the specifics of the O/RM Entity Framework 6, Hangfire for background jobs, and Zeus for electronic invoicing.

O/RM Entity Framework 6

Entity Framework 6 (EF6) is a mature and widely-used object-relational mapper (O/RM) for .NET [1]. An O/RM is a technology that creates a bridge between the object-oriented programming world of your application and the relational world of your database. It automates the translation of data between these two paradigms, significantly reducing the amount of boilerplate data-access code that developers need to write.

Core Concepts

At its core, EF6 allows developers to work with a database using strongly-typed .NET objects, known as entities. These entities are typically Plain Old CLR Objects (POCOs), meaning they are simple classes that are not dependent on any specific framework. This enables a clean separation of concerns and a more maintainable codebase.

EF6 provides several key features that simplify data access and management:

Feature	Description
Automatic Change Tracking	EF6 automatically keeps track of changes made to your entities, so you don't have to manually manage the state of your objects.
Identity Resolution & Unit of Work	EF6 ensures that each unique entity is represented by a single object instance within a given context, and it groups related database operations into a single transaction.
Querying with LINQ	You can use the familiar Language-Integrated Query (LINQ) syntax to write strongly-typed queries against your database, which are then translated into SQL by EF6.
Rich Mapping Capabilities	EF6 supports a wide range of mapping scenarios, including one-to-one, one-to-many, and many-to-many relationships, as well as inheritance hierarchies.

Practical Application

In your application, EF6 is used to interact with the database, performing CRUD (Create, Read, Update, Delete) operations on your data. It allows your C# code to work with objects like `Customer` and `Order` directly, without having to write explicit SQL statements. This not only speeds up development but also reduces the risk of SQL injection vulnerabilities.

Hangfire

Hangfire is an open-source library for .NET that provides a simple and reliable way to perform background processing in your application [2]. It allows you to offload long-running, CPU-intensive, or scheduled tasks to a separate process, ensuring that your main application remains responsive and available.

Core Concepts

Hangfire is designed to be easy to set up and use. It doesn't require a separate Windows Service or process, and it can be integrated directly into your existing ASP.NET application. It uses a persistent storage mechanism, such as SQL Server or Redis, to store information about background jobs, ensuring that they are not lost in case of an application restart or failure.

Here are the different types of background jobs that Hangfire supports:

Job Type	Description
Fire-and-Forget	Executed only once, almost immediately after being created.
Delayed	Executed only once, after a specified time interval.
Recurring	Executed many times on a specified CRON schedule.
Continuations	Executed when a parent job has finished.

Practical Application

In your application, Hangfire can be used for a variety of tasks, such as sending emails, generating reports, processing large data files, or performing scheduled maintenance. By using Hangfire, you can ensure that these tasks are executed reliably in the background, without impacting the performance of your user-facing application.

Zeus (NFe Generation)

Zeus is a specialized component for generating Brazilian electronic invoices (Nota Fiscal Eletrônica or NF-e) [3, 4]. The NF-e is a digital document that is used to formalize the circulation of goods and the provision of services in Brazil. It is a mandatory requirement for most businesses, and it is subject to strict regulations and technical specifications.

Core Concepts

The Zeus component simplifies the process of generating and managing NF-e documents by providing a high-level API that abstracts away the complexities of the underlying XML format and communication protocols. It handles the creation of the NF-e XML file, the digital signing of the document, and the transmission of the file to the SEFAZ (Secretaria da Fazenda), which is the Brazilian tax authority.

Practical Application

In your application, the Zeus component is used to generate NF-e documents for your sales transactions. It integrates with your existing business logic and data to create valid and compliant NF-e files, which are then sent to the SEFAZ for authorization. This ensures that your business is compliant with Brazilian tax regulations and that your invoices are legally valid.

Database Technologies

The database is the heart of your application, and it is built on a solid foundation of technologies that ensure data integrity, security, and performance. This section covers the use of Ledger with Blockchain, stored procedures, and SQL views.

Ledger with Blockchain

SQL Server's Ledger feature brings the power of blockchain technology to your relational database [5]. It provides tamper-evident capabilities that help you protect the integrity of your data and detect any unauthorized modifications. This is particularly important for applications that handle sensitive or financial data, where data integrity is paramount.

Core Concepts

Ledger works by creating a cryptographically-linked chain of data blocks, where each block contains a set of transactions. Each transaction is hashed, and the hashes are then combined to create a block hash. The block hashes are then linked together to form a blockchain, which is an immutable and verifiable record of all changes made to the database.

There are two types of ledger tables:

Table Type	Description
Updatable Ledger Tables	Allow you to update and delete rows, while preserving a full history of all changes.
Append-Only Ledger Tables	Only allow you to insert new rows, providing an even higher level of protection against tampering.

Practical Application

In your application, Ledger with Blockchain is used to ensure the integrity of your financial transactions and other sensitive data. It provides a verifiable audit trail of all changes made to the database, which can be used to detect and prevent fraud, as well as to comply with regulatory requirements.

Stored Procedures

Stored procedures are pre-compiled collections of SQL statements that are stored in the database [6]. They can be executed as a single unit, and they can accept input parameters and return output parameters. Stored procedures are a powerful tool for encapsulating business logic, improving performance, and enhancing security.

Core Concepts

Stored procedures offer several key benefits:

- **Reduced Network Traffic**: By executing a single stored procedure call instead of sending multiple SQL statements over the network, you can significantly reduce network latency and improve performance.
- **Stronger Security**: Stored procedures can be used to control access to data, allowing users to execute specific operations without having direct access to the underlying tables.
- **Code Reusability**: Stored procedures can be called from multiple applications and programming languages, promoting code reuse and reducing development time.
- **Improved Performance**: Stored procedures are compiled and optimized by the database engine, which can result in faster execution times compared to ad-hoc queries.

Practical Application

In your application, stored procedures are used to implement complex business logic, perform data validation, and enforce business rules. They provide a secure and efficient way to interact with the database, and they help to ensure the consistency and integrity of your data.

SQL Views

SQL views are virtual tables that are based on the result-set of a SQL query [7]. They provide a way to simplify complex queries, restrict access to data, and present data in a more user-friendly format. Views do not store data themselves, but rather they provide a window into the underlying data in the tables.

Core Concepts

Views offer several advantages:

- **Simplicity**: Views can be used to hide the complexity of the underlying database schema, making it easier for users to query and understand the data.
- **Security**: Views can be used to restrict access to sensitive data by only exposing a subset of the columns or rows in a table.
- **Consistency**: Views can be used to ensure that all users see the same data, even if the underlying tables are being updated.
- **Data Independence**: Views can be used to shield users from changes to the underlying database schema, allowing you to refactor your database without breaking existing applications.

Practical Application

In your application, SQL views are used to provide a simplified and secure view of the data to different users and roles. They are also used to create reports and dashboards, as well as to support ad-hoc queries from business users.

Front-End Technology

The front-end of your application is built with the Bulma CSS framework, which provides a modern and responsive design system for your user interface.

CSS Framework Bulma

Bulma is a free and open-source CSS framework that is based on Flexbox [8, 9]. It is a lightweight and modular framework that provides a set of pre-designed components and styles that you can use to build beautiful and responsive websites and applications.

Core Concepts

Bulma is designed to be simple and easy to use. It has a clean and intuitive class-based syntax, and it does not require any JavaScript. It is also highly customizable, allowing you to easily change the colors, fonts, and other design elements to match your brand.

Some of the key features of Bulma include:

- **Responsive Design**: Bulma is a mobile-first framework, which means that your application will look great on any device, from a small smartphone to a large desktop monitor.
- **Modular Architecture**: Bulma is divided into a set of independent modules, so you can import only the components that you need, keeping your CSS file size small.
- **Flexbox-based Layout**: Bulma's layout system is based on Flexbox, which is a modern and powerful CSS layout module that makes it easy to create complex and flexible layouts.

Practical Application

In your application, Bulma is used to style the user interface, including the layout, typography, colors, and components. It provides a consistent and professional look and feel to your application, and it ensures that your application is easy to use and navigate.

Cloud Technologies

Your application is hosted in a cloud environment that provides a scalable and reliable platform for your application. This section covers the use of IIS and the Windows Task Scheduler.

IIS (Internet Information Services)

IIS is Microsoft's web server, and it is used to host your application in the cloud [10]. It is a powerful and flexible web server that provides a wide range of features for hosting and managing web applications.

Core Concepts

IIS has a modular architecture that allows you to customize it to meet the specific needs of your application. It includes a set of built-in modules that provide features such as authentication, authorization, compression, and caching. You can also create your own custom modules to extend the functionality of IIS.

Practical Application

In your application, IIS is used to host your ASP.NET application and serve it to your users. It provides a secure and reliable environment for your application, and it can be scaled to handle a large number of users and requests.

Windows Task Scheduler

Windows Task Scheduler is a built-in Windows component that allows you to automate tasks and run them on a schedule [11]. It is a powerful tool for automating a wide range of tasks, from simple maintenance tasks to complex business processes.

Core Concepts

Task Scheduler allows you to create tasks that are triggered by a variety of events, such as a specific time, a user logon, or a system event. You can also configure tasks to run on a recurring schedule, such as daily, weekly, or monthly.

Practical Application

In your application, Windows Task Scheduler is used to automate a variety of tasks, such as running database backups, generating reports, and sending email notifications. It provides a reliable and efficient way to automate these tasks, and it helps to ensure that your application is running smoothly and efficiently.

References

[1] Overview of Entity Framework 6 [2] Hangfire – Background jobs and workers for .NET and .NET Core [3] NF-e - Zeus Automação [4] GitHub - ZeusAutomacao/DFe.NET [5] Ledger Overview - SQL Server [6] Stored Procedures (Database Engine) - SQL Server [7] Views - SQL Server [8] Official Bulma Documentation [9] Features | Bulma [10] Introduction to IIS Architectures [11] Task Scheduler for developers - Win32 apps