

**Curso:**

SC-702 Diseño y Desarrollo de Sistemas

Table of Contents

[Objectives. 3](#_Toc127303007)

[General Objective: 3](#_Toc127303008)

[Specific Objectives 3](#_Toc127303009)

[Reach 3](#_Toc127303010)

[Context of the selected company or entrepreneurship. 3](#_Toc127303011)

[Definitions: 3](#_Toc127303012)

[Technologies And Platforms 5](#_Toc127303013)

[Main Programing language: 5](#_Toc127303014)

[Back-end technologies: 5](#_Toc127303015)

[Frontend technologies: 5](#_Toc127303016)

[Hosting platform: 5](#_Toc127303017)

[User Requirements: 6](#_Toc127303018)

[Entity Relationship Diagram 7](#_Toc127303019)

[Gantt chart 8](#_Toc127303020)

[Repository 9](#_Toc127303021)

[Conclusions: 9](#_Toc127303022)

[Recommendations: 9](#_Toc127303023)

# Objectives.

## General Objective:

Create a tool to help people develop a plan to be debt free.

## Specific Objectives

* Provide a comparison between different pay-off strategies.
* Provide the dates of when each debt will get paid off.
* Provide a date of when the user will be death free.
* Provide feedback on how a one-time extra payment impacts the plan.
* Provide feedback on how a regular fix extra payment impacts the plan.

# Reach

To create a web app for the general public that works on Firefox-Based and Chromium-Base browsers currently supported by their manufacturers and was released within the last year.

For other browsers, it will be a best-effort scenario, but they will not be officially supported.

# Context of the selected company or entrepreneurship.

Given the increasing number of people that go on multiple debts, we have decided to help solve this problem by creating a tool that will enable them to implement strategies such as Debt Snowball, Debt Avalanche, and Debt Snowflake.

## Definitions:

**Debt Snowball:** is a strategy for paying down debts, popularized by personal finance author Dave Ramsey. In this method you would:

1. List all your debts.
2. Order them by smallest balance first (Strict Debt Snowball) or largest interest rate first (Debt Avalanche)
3. You pay the minimum on all loans except for the one towards the top, which you would put any extra that you have towards that one.
4. Once you pay off a loan, you will now put the extra amount towards the loan next on the list, and so on, until you pay all of it off.

**Strict Debt Snowball**: Here, you will pay off the loan with the smallest balance first. In this strategy, you will pay more interest compared to Debt Avalanche. The idea is to be motivated as you will cancel your deaths.

**Debt Avalanche**: In this strategy, you will first pay off the loan with the largest interest rate first. The main benefit is that you would pay the least amount in interest, making it the most “logical” choice compared to Strict Debt Snowball. But as Dave Ramsey points out, you did not get deep into debt because you were making logical decisions.

**Debt Snowflake**: This strategy involves making extra payments on your debts whenever you have extra money, such as a bonus or Aguinaldo (13th-month pay).

# Technologies And Platforms

We will create a An ASP.NET Core hosted Blazor WebAssembly app the fallowing way.

## Front-end:

* The front end will be created whit Blazor 6 LTSR.
* We will use the Code behind technique.
* We will use the third-party tool Auth0.
* We will use Repository pattern.
* Will have MVC architecture.
* We wil use this template: [Canvas | The Multi-Purpose HTML5 Template by SemiColonWeb | ThemeForest](https://themeforest.net/item/canvas-the-multipurpose-html5-template/9228123)

## Back-end:

* Will be a restful API.
* Will use a Generic Repository pattern.
* Will use a n-tier architecture. Whit the fallowing modification: the Presentation Layer, will basically be a Controller of a MVC architecture. The other layers Business Logic Layer(BLL), Data Access Layer (DAL), and Database Layer will work as customary
* We will use Entity framework

## Database:

* SQL server hosted on Azure

## Hosting platform:

* A basic WebApp hosted on Azure

# User Requirements:

1. **User Creation**: The tool should enable a user to create an account requesting only their Email and name and password.
2. **Loan Creation**: Ideally the user will provide all the required data, but the program should be able to calculate the remaining information using financial formulas data if it has at least 4 of the following:

* Remaining Balance
* Interest rate
* Fees
* Remaining Term
* Monthly Payment

1. **Loan Management**: The tool should allow users to add, remove, modify the information pertaining to a ezisting loan such as, remaining balance, interest rate.
2. **Debt Comparison**: The tool should provide a comparison between different debt repayment strategies, including Debt Snowball, Debt Avalanche, and Debt Snowflake.
3. **Stat Management**: The tool will gather practical user session information for statistical purposes, such as log on, log off, OS, and Browser.
4. **Debt Payoff Estimator**: The tool should be able to determine when a specific debt will get paid off and when they will be debt free.
5. **Debt-Free Estimator**: The tool should estimate when the user will be debt-free based on their chosen strategy and payment plan.
6. **Extra Payment Calculator**: The tool should allow users to determine the potential savings on interest and the time shortened to be debt free because of a one-time extra payment.
7. **Fixed Extra Payment Calculator**: The tool should allow users to determine the potential savings on interest and the time shortened to be debt free because of fixed extra monthly payments.

# Entity Relationship Diagram

CREATE TABLE `Users` (

`ID` INT NOT NULL ,

`FirstName` VARCHAR(10) NOT NULL ,

`LastName` VARCHAR(10) NOT NULL ,

`Email` VARCHAR(20) NOT NULL ,

`Password` VARCHAR(20) NOT NULL ,

`UserTypeID` INT NOT NULL ,

PRIMARY KEY (

`ID`

),

CONSTRAINT `uc\_Users\_Email` UNIQUE (

`Email`

)

);

CREATE TABLE `UserTypes` (

-- Posible types are:

-- App\_User

-- Admin\_User

`ID` INT NOT NULL ,

`Type` VARCHAR(20) NOT NULL ,

PRIMARY KEY (

`ID`

)

);

CREATE TABLE `SessionLog` (

`ID` INT NOT NULL ,

`UserID` INT NOT NULL ,

`LogonTimeStamp` DateTime NOT NULL ,

`LogoffTimeStam` DateTime NOT NULL ,

`OperatingSystem` VARCHAR(20) NOT NULL ,

`ClientSoftware` VARCHAR(20) NOT NULL ,

`RemoteIpAddress` VARCHAR(20) NOT NULL ,

PRIMARY KEY (

`ID`

)

);

CREATE TABLE `PaymentStrategyplans` (

`ID` INT NOT NULL ,

`UserID` INT NOT NULL ,

`StrategysTypeID` INT NOT NULL ,

`GlobalMonthlyPayment` NUMBER(10,3) NOT NULL ,

PRIMARY KEY (

`ID`

)

);

CREATE TABLE `Loans` (

`ID` INT NOT NULL ,

`LoanNickName` VARCHAR(20) NOT NULL ,

`PaymentStrategy` INT NOT NULL ,

`Principal` NUMBER(10,3) NOT NULL ,

`InterestRate` NUMBER(3,5) NOT NULL ,

`Fees` NUMBER(10,3) NOT NULL ,

`MonthlyPayment` NUMBER(3,5 NOT NULL ,

`RemainingTerm` INT NOT NULL ,

`Curency` INT NOT NULL ,

PRIMARY KEY (

`ID`

),

CONSTRAINT `uc\_Loans\_LoanNickName` UNIQUE (

`LoanNickName`

)

);

CREATE TABLE `Curencys` (

`ID` INT NOT NULL ,

`FormalName` VARCHAR(20) NOT NULL ,

`ShortName` VARCHAR(10) NOT NULL ,

`Sybal` VARCHAR(5) NOT NULL ,

PRIMARY KEY (

`ID`

)

);

CREATE TABLE `DebtSnowflake` (

-- Extra payments on the debts eg a bonus, aquinaldo, ect

`ID` INT NOT NULL ,

`Date` DATE NOT NULL ,

`Amount` NUMBER(10,3) NOT NULL ,

`PaymentStrategy` INT NOT NULL ,

PRIMARY KEY (

`ID`

)

);

CREATE TABLE `StrategyTypes` (

-- Payment Strategys are:

-- StrictDebtSnowball - Smallest balance First

-- DebtAvalanche - Highest interest rate

-- Custom - a user defined

`ID` INT NOT NULL ,

`Type` VARCHAR(20) NOT NULL ,

`HasCustomStrategy` BIT NOT NULL ,

PRIMARY KEY (

`ID`

)

);

CREATE TABLE `LoansCardinalOrder` (

`ID` INT NOT NULL ,

`PaymentStrategy` INT NOT NULL ,

`LoanID` INT NOT NULL ,

`CardinalOrder` INT NOT NULL ,

PRIMARY KEY (

`ID`

)

);

ALTER TABLE `Users` ADD CONSTRAINT `fk\_Users\_UserTypeID` FOREIGN KEY(`UserTypeID`)

REFERENCES `UserTypes` (`ID`);

ALTER TABLE `SessionLog` ADD CONSTRAINT `fk\_SessionLog\_UserID` FOREIGN KEY(`UserID`)

REFERENCES `Users` (`ID`);

ALTER TABLE `PaymentStrategyplans` ADD CONSTRAINT `fk\_PaymentStrategyplans\_UserID` FOREIGN KEY(`UserID`)

REFERENCES `Users` (`ID`);

ALTER TABLE `PaymentStrategyplans` ADD CONSTRAINT `fk\_PaymentStrategyplans\_StrategysTypeID` FOREIGN KEY(`StrategysTypeID`)

REFERENCES `StrategyTypes` (`ID`);

ALTER TABLE `Loans` ADD CONSTRAINT `fk\_Loans\_PaymentStrategy` FOREIGN KEY(`PaymentStrategy`)

REFERENCES `PaymentStrategyplans` (`ID`);

ALTER TABLE `Loans` ADD CONSTRAINT `fk\_Loans\_Curency` FOREIGN KEY(`Curency`)

REFERENCES `Curencys` (`ID`);

ALTER TABLE `DebtSnowflake` ADD CONSTRAINT `fk\_DebtSnowflake\_PaymentStrategy` FOREIGN KEY(`PaymentStrategy`)

REFERENCES `PaymentStrategyplans` (`ID`);

ALTER TABLE `LoansCardinalOrder` ADD CONSTRAINT `fk\_LoansCardinalOrder\_PaymentStrategy` FOREIGN KEY(`PaymentStrategy`)

REFERENCES `PaymentStrategyplans` (`ID`);

ALTER TABLE `LoansCardinalOrder` ADD CONSTRAINT `fk\_LoansCardinalOrder\_LoanID` FOREIGN KEY(`LoanID`)

REFERENCES `Loans` (`ID`);

[Debt snowball - QuickDBD (quickdatabasediagrams.com)](https://app.quickdatabasediagrams.com/#/d/UaATZX)

Graphical user interface, text, application, chat or text message

Description automatically generated

# Repository

# Conclusions:

We believe there is a real need for a tool such as this that provides users with an intuitive interface that allows them to manage their loans, compare different debt repayment strategies, and estimate when they will become debt-free.

Using technologies such as C# .Net 6, ASP.NET Core, Blazor, and Oracle will help us develop a robust and scalable application that can be easily deployed in the cloud.

# Recommendations:

Based on our feasibility analysis, we recommend:

1. Getting users verse in personal finances to test the application.
2. Make the web page responsive
3. To create a mobile version of the app.
4. To integrate the app with other financial tools, such as budgeting and savings apps, to provide users with a more comprehensive view of their financial health
5. To explore ways to monetize the app, such as offering premium features or partnering with financial institutions.