



Documentation Stage du 27 mai 2024 au 28 juin 2024 - 1 e année BTS SIO SLAM

MAF RODA AGROBOTIC

Maître de stage :

David Dumas

Etudiant :

Eliaz Talmitte

Lycée Polyvalent Claude Nougaro

Année scolaire 2023-2024

Remerciements

Je tiens à remercier toutes les personnes qui ont contribué au bon déroulement de mon stage. Tout d'abord, un grand merci à mes professeurs pour les connaissances qu'ils m'ont transmises. Leurs enseignements ont été essentiels pour me préparer et réussir ce stage.

Je remercie aussi mon maître de stage David Dumas pour sa disponibilité et ses conseils. Grâce à lui, j'ai pu m'intégrer facilement dans l'équipe et apprendre beaucoup.

Enfin, je souhaite remercier toute l'équipe de Maf Roda du service informatique pour leur accueil chaleureux et leur soutien. Leur gentillesse et leur esprit d'équipe ont rendu mon stage très agréable et enrichissant.

Sommaire

Remerciements.....	2
Sommaire	3
Introduction	4
Présentation de Maf Roda :	4
Contexte du stage.....	4
Outils utilisés :	5
Langages utilisés :	5
Frameworks et technologies utilisées :	6
Description des missions	6
Missions réalisées en supervision	6
Exercice en autonomie	7
Contrainte.....	26
Sources consultées.....	26
Conclusion	27

Introduction

Présentation de Maf Roda :

MAF Roda Agrobotic est une entreprise française spécialisée dans la conception, la fabrication, et la commercialisation de solutions technologiques avancées pour le tri, le calibrage, et le conditionnement de fruits et légumes. Fondée en 1905, MAF Roda est aujourd'hui un leader mondial dans son domaine, avec une forte présence internationale.

L'entreprise se distingue par son innovation constante et son engagement envers la qualité. Elle développe des machines et des systèmes automatisés capables d'effectuer des contrôles précis sur la qualité, la maturité, la taille, la couleur, et d'autres critères importants pour les produits agricoles. Ces solutions permettent aux producteurs de fruits et légumes de garantir des standards élevés, de réduire les pertes, et d'améliorer l'efficacité de leurs opérations.

MAF Roda Agrobotic est également reconnue pour son service client, avec une équipe dédiée à l'accompagnement de ses partenaires à travers le monde. Son siège social est situé en France à Montauban, mais l'entreprise dispose de filiales et de partenaires dans plusieurs pays, ce qui lui permet de répondre aux besoins spécifiques de chaque marché. La société comporte des services informatiques notamment en France et en Espagne.

Contexte du stage

J'ai choisi **MAF RODA** parce qu'elle dispose d'un service de développement de logiciel, domaine informatique qui m'intéresse le plus. L'objectif de ce stage était d'élargir mes connaissances théoriques et de les mettre en application. J'ai assisté à plusieurs étapes de codage de logiciels en c#.

Outils utilisés :

- **Visual Studio** : Environnement de développement intégré (IDE) pour le développement en C#.
- **DevExpress (WinForms)** : Framework pour la création d'applications Windows Forms, utilisé pour les interfaces graphiques (UI).
- **Microsoft Edge** : Navigateur web utilisé pour tester les APIs et consulter les tutoriels.
- **GitHub** : Plateforme de gestion de code source pour versionner et collaborer sur les projets.
- **Uno Platform** : Framework pour le développement d'applications multi-plateformes.
- **DbVisualizer** : Outil de gestion de bases de données, utilisé pour interagir avec des bases SQL.
- **IBM AS400** : Système de gestion de bases de données.
- **GLPI** : Logiciel de gestion de services informatiques, utilisé pour la gestion des tickets et des utilisateurs.
- **Swagger UI** : Une interface utilisateur interactive qui permet de visualiser et de tester les points d'entrée de l'API directement depuis le navigateur.

Langages utilisés :

- **C#** : Langage principal utilisé pour le développement d'applications, les scripts, et la manipulation de données.
- **SQL (SQLite)** : Langage utilisé pour les requêtes et les procédures stockées dans la gestion des bases de données.
- **JSON** : Format de données utilisé pour la sérialisation, la sauvegarde, et la lecture de données dans les fichiers.
- **XAML** : Langage de balisage utilisé avec Uno Platform pour définir les interfaces utilisateur.

Frameworks et technologies utilisées :

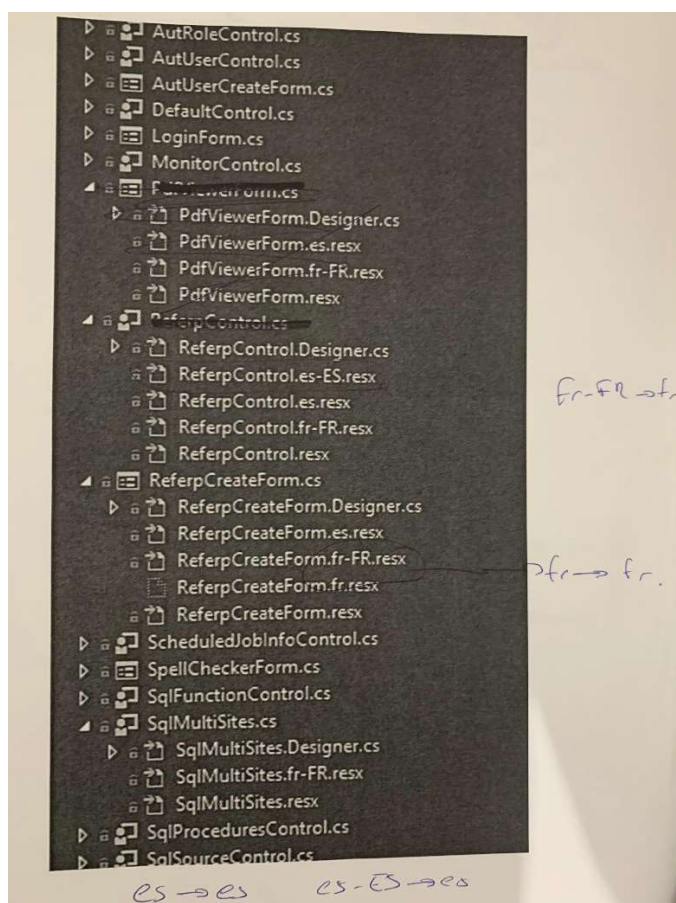
- **ASP.NET Web API** : Framework pour créer des APIs web.
- **.NET et .NET Core** : Frameworks utilisés pour le développement d'applications et d'APIs.

Description des missions

Mon maître de stage m'a confié divers exercices pratiques pour renforcer mes compétences tout en me permettant de travailler directement à ses côtés sur les applications qu'il développait. Sous sa supervision, j'ai contribué au développement de fonctionnalités, corrigé des erreurs de code, et appris les bonnes pratiques de développement logiciel en temps réel.

Missions réalisées en supervision

1. **Internationalisation des applications** : Adaptation des interfaces pour qu'elles soient disponibles en plusieurs langues (Français et Espagnol).

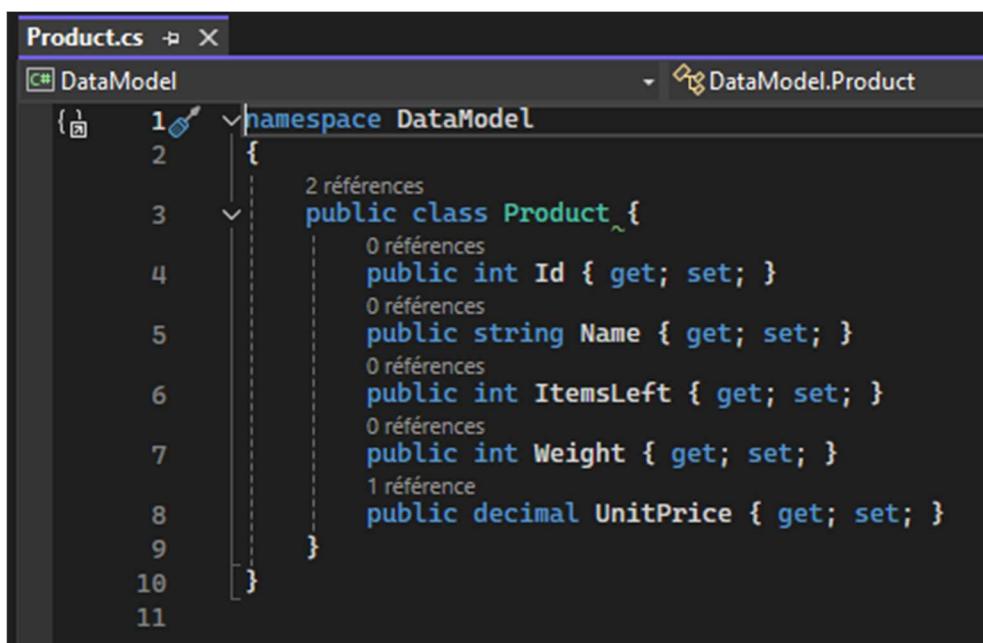


2. **Implémentation de règles SQL** : Création d'une règle SQL pour empêcher les doublons lors de l'ajout d'utilisateurs.
3. **Développement d'interactions utilisateur (UI)** : Amélioration du passage entre les écrans et du positionnement du curseur.
4. **Gestion des tickets dans GLPI** : Participation à la gestion des tickets dans GLPI.
5. **Ajout de fonctionnalités** : Développement d'une fonctionnalité permettant de copier les rôles d'un utilisateur à un autre.
6. **Procédures SQL** : Création et utilisation de procédures SQL pour automatiser certaines opérations.
7. **Publication d'une application** : Participation à la mise en production d'une application pour l'entreprise.
8. **Rédaction de documentation technique** : Rédaction de la documentation associée aux développements réalisés.

Exercice en autonomie

Ajout d'un attribut à une API Web Service : Sur une solution utilisant **Swagger** et un client en particulier winFormDesktopClient, j'ai ajouté un attribut weight à l'entité products dans une API Web Service.

Swagger est un outil qui permet de documenter et tester facilement des API. Il génère automatiquement une interface web où l'on peut voir toutes les routes de l'API et tester leur fonctionnement directement depuis le navigateur.



```
Product.cs  ↗ ✕
C# DataModel  DataModel.Product
1 namespace DataModel
2 {
3     2 références
4     public class Product {
5         0 références
6         public int Id { get; set; }
7         0 références
8         public string Name { get; set; }
9         0 références
10        public int ItemsLeft { get; set; }
11        0 références
12        public int Weight { get; set; }
13        1 référence
14        public decimal UnitPrice { get; set; }
15    }
16 }
```

Swagger :

The screenshot shows the Swagger UI interface for the endpoint `GET /api/Orders/{orderDate}`. Under the 'Schemas' section, the 'Product' schema is expanded, showing the following properties:

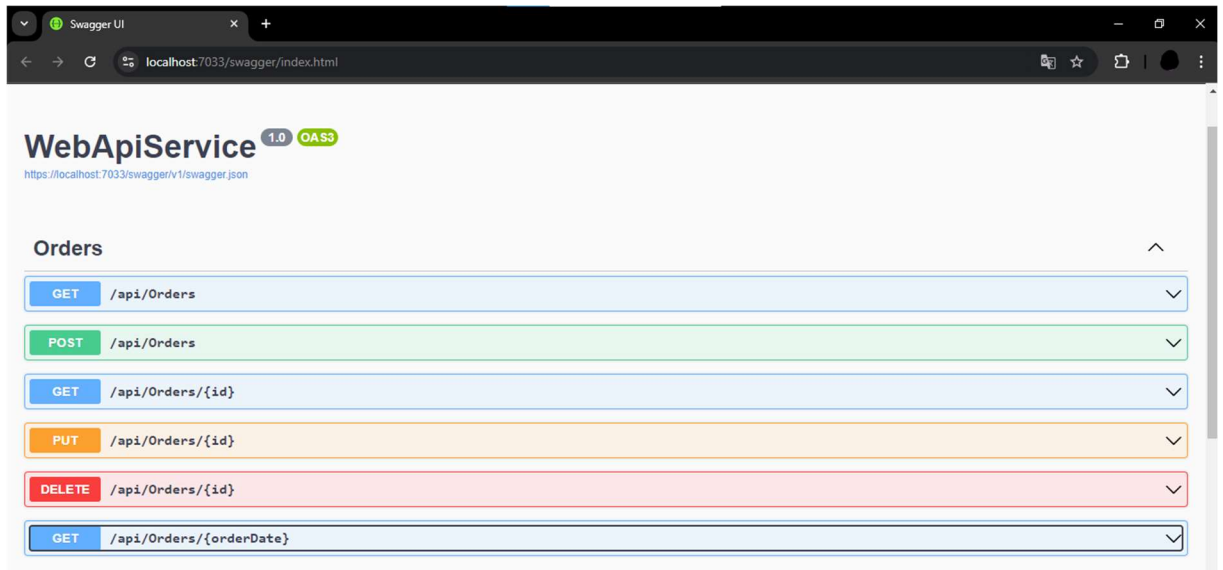
```
Product {
  id: integer($int32)
  name: string
  nullable: true
  itemsLeft: integer($int32)
  weight: integer($int32)
  unitPrice: number($double)
}
```

WinFormDesktopClient :

Id	State	Customer	Order Date	Items Count
1	Processed	Albert Menendez	01/10/2023	2
Items				
Drag a column header here to group by that column				
Id	Product Name	Product Unit Price	Quantity	Weight
1	Sir Rodney's Scones	10,00 €	90	20
2	Gnocchi di nonna Alice	38,00 €	15	76
2	Processed	Alex James	01/10/2023	4
3	Processed	John Doe	02/10/2023	6
4	Processed	Albert Menendez	03/10/2023	5

Création d'une méthode `getOrderByDate` : J'ai développé une méthode GET prenant en paramètre une date entrée par l'utilisateur et renvoyant la commande correspondante si elle est trouvée.

```
// GET: api/Orders/{OrderDate}
[HttpGet("{orderDate}")]
0 références
public async Task<ActionResult<IEnumerable<Order>>> GetOrdersByDate(string orderDate)
{
    return await _context.Orders.Where(order => order.OrderDate.ToString() == orderDate)
        .ToListAsync();
}
```



Conversion de minutes en heures : Sur une autre solution, j'ai programmé en C# un convertisseur de minutes en heures, incluant une boucle permettant de redemander à l'utilisateur s'il souhaite effectuer une nouvelle conversion.

```
Program.cs
convertMinute
1  int rep = 0;
2
3  while (rep == 0)
4  {
5      Console.WriteLine("Entrez des minutes : ");
6      int.TryParse(Console.ReadLine(), out var n);
7      Console.WriteLine($"{n} minutes = " + "{0:00}:{1:00}" + " heures", n / 60, n % 60);
8      Console.WriteLine("\nPour arrêter taper 1 pour continuer taper 0 : ");
9      rep = int.Parse(Console.ReadLine());
10 }
11 if (rep != 0)
12 {
13     Console.WriteLine("Programme terminer !!");
14 }
```

```
Entrez des minutes :
120

120 minutes = 02:00 heures

Pour arrêter taper 1 pour continuer taper 0 :
0
Entrez des minutes :
60

60 minutes = 01:00 heures

Pour arrêter taper 1 pour continuer taper 0 :
1
Programme terminer !!
```

Manipulation de listes et de labels avec DevExpress : Utilisation d'une liste et d'un label avec **DevExpress**, et développement d'une fonction pour retirer l'extension .txt des éléments présents dans la liste.

DevExpress est un ensemble d'outils et de bibliothèques qui facilitent la création d'interfaces graphiques pour des applications Windows. Il permet de développer des interfaces complexes avec des éléments interactifs, comme des tableaux, des boutons, ou des graphiques, tout en optimisant la performance et l'esthétique des applications

```
11 références
public class CityInfo
{
    int id;
    6 références
    public CityInfo(int id)
    {
        this.id = id;
    }
    [Display(Order = -1)]
    0 références
    public int ID { get { return id; } }
    8 références
    public string City { get; set; }
    6 références
    public string Country { get; set; }
    6 références
    public string Region { get; set; }
    1 référence
    static public List<CityInfo> Init()
    {
        return new List<CityInfo>() {
            new CityInfo(0) { City = "Barquisimeto.txt", Country = "Venezuela", Region = "Lara" },
            new CityInfo(1) { City = "Rio de Janeiro.txt", Country = "Brazil", Region = "RJ" },
            new CityInfo(2) { City = "Cunewalde.txt", Country = "Germany", Region = "" },
            new CityInfo(3) { City = "Madrid.txt", Country = "Spain", Region = "" },
            new CityInfo(4) { City = "Charleroi.txt", Country = "Belgium", Region = "" },
            new CityInfo(5) { City = "Sao Paulo.txt", Country = "Brazil", Region = "SP" }
        };
    }
}
```

```

public XtraForm1()
{
    InitializeComponent();
    var listEliaz = CityInfo.Init();
    gridControll.DataMember = "data";
    gridControll.DataSource = CreateDataSourceFromFile();
    lookUpEdit1.Properties.DataSource = listEliaz;
    lookUpEdit1.Properties.ValueMember = "City";
    lookUpEdit1.Properties.DisplayMember = "City";

    foreach (CityInfo city in listEliaz)
    {
        string filepath = city.City;
        city.City = Path.GetFileNameWithoutExtension(filepath);
    }
}
1 référence
private void lookUpEdit1_EditValueChanged(object sender, EventArgs e)
{
    label1.Text = lookUpEdit1.Text + ".txt";
}

```

City	Country	Region
Barquisimeto	Venezuela	Lara
Rio de Janeiro	Brazil	RJ
Cunewalde	Germany	
Madrid	Spain	
Charleroi	Belgium	
Sao Paulo	Brazil	SP

Cunewalde.txt

Vérification des doublons dans un fichier JSON : Sur un programme existant qui enregistre et met à jour des données dans un fichier **JSON**, j'ai ajouté une fonctionnalité pour tester si un enregistrement existait déjà avant d'ajouter de nouvelles données, afin d'éviter les doublons

Le format **JSON** (JavaScript Object Notation) est un format léger de stockage et de transmission de données, souvent utilisé dans les applications web pour échanger des informations entre un serveur et un client. Il est facile à lire et à écrire pour les humains, et simple à analyser pour les machines.

```

var test = records.Find(r => r.Key == newRecord.Key);

if (test != null)
{
    Console.WriteLine("Enregistrement déjà existant pour l'insert.");
}
else
{
    records.Add(newRecord);
    Console.WriteLine("Enregistrement ajouté avec succès.");
}

string updatedJsonData = JsonConvert.SerializeObject(records, Formatting.Indented);
File.WriteAllText(jsonPath, updatedJsonData);

```

```

Enregistrement mis à jour avec succès.
Enregistrement déjà existant pour l'insert.

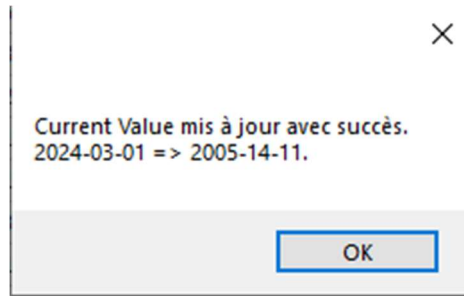
```

Intégration et manipulation de données JSON avec DevExpress : Dans le même programme, j'ai implémenté un GridControl (tableau) pour afficher les données du fichier data.json. J'ai également ajouté un TextEdit permettant de modifier les enregistrements en rentrant une nouvelle valeur, ainsi qu'un SimpleButton pour valider les modifications. Lorsque l'utilisateur clique sur "OK", les nouvelles données sont automatiquement mises à jour dans le fichier data.json.

Le **GridControl** est un composant graphique utilisé pour afficher et manipuler des données sous forme de tableau dans une interface graphique. Il permet d'organiser les données de manière visuelle et interactive, facilitant ainsi leur lecture et leur gestion.

	Key	CurrentValue	OldValues
	BBBBB-3333-29	2002-05-06	2002-05-06
	BBBBB-3333-30	2008-05-01	2007-05-01
✎	BBBBB-3333-28	2024-03-01	2023-03-01
	BBBBB-3333-27	2001-12-11	2001-12-12
	BBBBB-3333-24	2019-01-03	2018-01-03
	BBBBB-3333-22	2005-21-06	2004-21-06
	BBBBB-3333-26	2019-08-03	2018-08-02
	BBBBB-3333-23	2010-11-20	2009-11-11

2023-03-01	OK
------------	----



Key	CurrentValue	OldValues
BBBBB-3333-29	2002-05-06	2002-05-06
BBBBB-3333-30	2008-05-01	2007-05-01
BBBBB-3333-28	2005-14-11	2023-03-01
BBBBB-3333-27	2001-12-11	2001-12-12
BBBBB-3333-24	2019-01-03	2018-01-03
BBBBB-3333-22	2005-21-06	2004-21-06
BBBBB-3333-26	2019-08-03	2018-08-02
BBBBB-3333-23	2010-11-20	2009-11-11

2005-14-11 OK

Voici a quoi ressemble le fichier data.json :

```
data.json  [X]
Schéma : <Aucun schéma sélectionné>
1  [
2  {
3    "Key": "BBBBB-3333-29",
4    "CurrentValue": "2002-05-06",
5    "OldValues": "2002-05-06"
6  },
7  {
8    "Key": "BBBBB-3333-30",
9    "CurrentValue": "2008-05-01",
10   "OldValues": "2007-05-01"
11  },
12  {
13   "Key": "BBBBB-3333-28",
14   "CurrentValue": "2005-14-11",
15   "OldValues": "2023-03-01"
16  },
17  {
18   "Key": "BBBBB-3333-27",
19   "CurrentValue": "2001-12-11",
20   "OldValues": "2001-12-12"
21  },
22  {
23   "Key": "BBBBB-3333-24",
24   "CurrentValue": "2019-01-03",
25   "OldValues": "2018-01-03",
```

Ajout d'une colonne Qty et calcul automatique : J'ai ajouté une colonne Qty (quantité) au GridControl et implémenté une fonctionnalité permettant de calculer automatiquement les valeurs restantes en fonction de la quantité saisie par l'utilisateur. La colonne "Calcul" soustrait les valeurs de Qty les unes après les autres à la valeur initiale. Toutes les modifications effectuées sont également enregistrées et actualisées dans le fichier data.json en temps réel.

```
int search = int.Parse(textEdit2.Text);
var dataListArray = records.ToArray();

dataListArray[0].Calcul = search - dataListArray[0].Qty;

for (int i = 1; i < dataListArray.Length; i++)
{
    int Stock_Qty = dataListArray[i - 1].Calcul;
    dataListArray[i].Calcul = Stock_Qty - dataListArray[i].Qty;
}
```

	Qty	Calcul
▶	10	990
	25	965
	15	950
	40	910
	50	860
	60	800
	70	730
	100	630

Voici à quoi ressemble maintenant le fichier data.json

```
data.json  ↳ ×
Schéma : <Aucun schéma sélectionné>
1  [
2  {
3    "Key": "BBBBB-3333-29",
4    "CurrentValue": "2002-05-06",
5    "OldValues": "2002-05-06",
6    "Qty": 10,
7    "Calcul": 990
8  },
9  {
10   "Key": "BBBBB-3333-30",
11   "CurrentValue": "2008-05-01",
12   "OldValues": "2007-05-01",
13   "Qty": 25,
14   "Calcul": 965
15  },
16  {
17   "Key": "BBBBB-3333-28",
18   "CurrentValue": "2024-03-01",
19   "OldValues": "2023-03-01",
20   "Qty": 15,
21   "Calcul": 950
22  },
23  {
24   "Key": "BBBBB-3333-27",
25   "CurrentValue": "2001-12-11",
26   "OldValues": "2000-12-11",
27   "Qty": 40,
28   "Calcul": 910
29  },
30 ]
```

API avec .NET Framework : J'ai suivi un tutoriel pour créer une API Web avec le .NET Framework.

Une **API** (Application Programming Interface) est un ensemble de règles qui permet à des logiciels de communiquer entre eux. Elle définit comment les demandes doivent être faites et comment les réponses doivent être fournies.

```
localhost:44330/api/values/
localhost:44330/api/values/
This XML file does not appear to have any style information associated with it. The document tree is shown below.
<ArrayOfstring xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
  <string>value1</string>
  <string>value2</string>
</ArrayOfstring>
```

```
localhost:44330/api/values/5
localhost:44330/api/values/5
This XML file does not appear to have any style information associated with it. The document tree is shown below.
<string xmlns="http://schemas.microsoft.com/2003/10/Serialization/">value</string>
```

```

// GET api/values
0 références
public IEnumerable<string> Get()
{
    return new string[] { "value1", "value2" };
}

// GET api/values/5
0 références
public string Get(int id)
{
    return "value";
}

```

API auto-hébergée avec OWIN : Toujours guidé par un tutoriel, j'ai mis en place une API ASP.NET Web API utilisant OWIN pour l'auto-hébergement, sous forme d'une application console. Owin signifie une abstraction entre les serveurs Web .NET et les applications Web.

```

StatusCode: 200, ReasonPhrase: 'OK', Version: 1.1, Content: System.Net.Http.StreamContent, Headers:
{
    Date: Sun, 01 Sep 2024 14:08:57 GMT
    Server: Microsoft-HTTPAPI/2.0
    Content-Length: 19
    Content-Type: application/json; charset=utf-8
}
["value1","value2"]

```

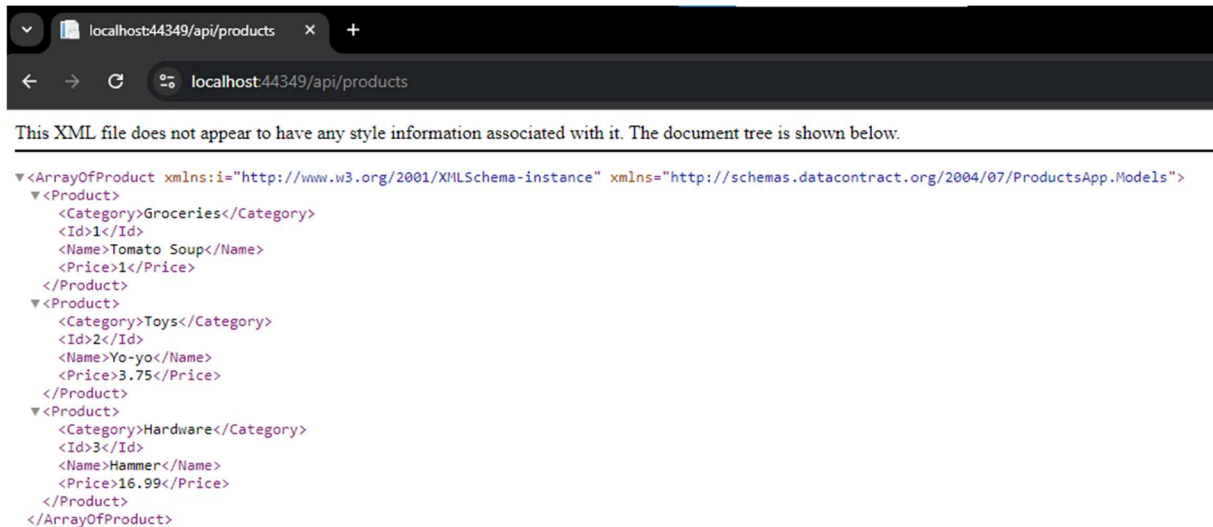
This XML file does not appear to have any style information associated with it. The document tree is shown below.

```

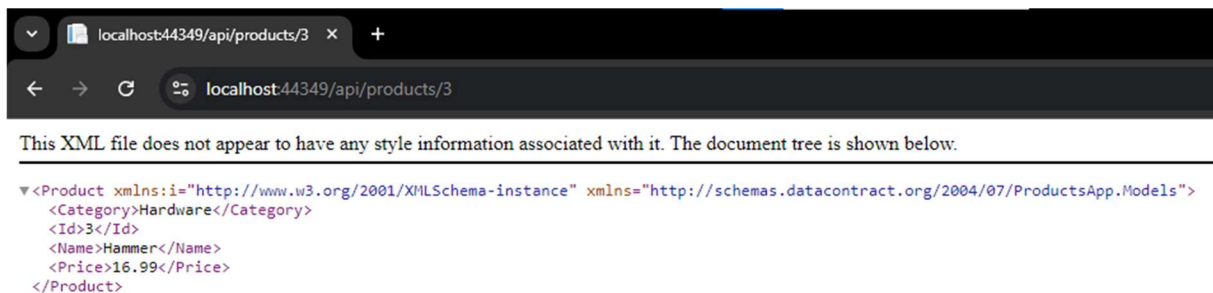
<ArrayOfstring xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
  <string>value1</string>
  <string>value2</string>
</ArrayOfstring>

```

API ASP.NET Web API 2 : Une autre API a été développée en suivant un tutoriel pour utiliser ASP.NET Web API 2 qui renvoie une liste de produits.



```
<ArrayOfProduct xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://schemas.datacontract.org/2004/07/ProductsApp.Models">
  <Product>
    <Category>Groceries</Category>
    <Id>1</Id>
    <Name>Tomato Soup</Name>
    <Price>1</Price>
  </Product>
  <Product>
    <Category>Toys</Category>
    <Id>2</Id>
    <Name>Yo-yo</Name>
    <Price>3.75</Price>
  </Product>
  <Product>
    <Category>Hardware</Category>
    <Id>3</Id>
    <Name>Hammer</Name>
    <Price>16.99</Price>
  </Product>
</ArrayOfProduct>
```



```
<Product xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://schemas.datacontract.org/2004/07/ProductsApp.Models">
  <Category>Hardware</Category>
  <Id>3</Id>
  <Name>Hammer</Name>
  <Price>16.99</Price>
</Product>
```

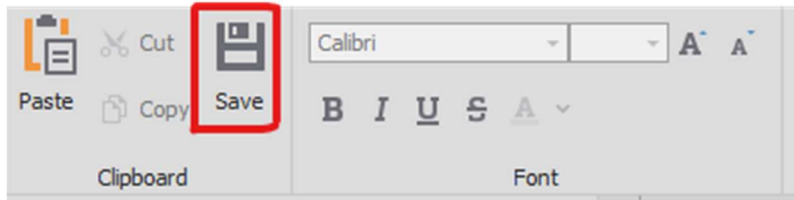
Le but était de configurer et de tester toutes ces API pour qu'elles fonctionnent correctement sur un PC distant en l'occurrence celui de mon maître de stage.

Création d'une démo avec DevExpress WinForms :

- **Création d'un bouton de sauvegarde et de chargement** : Développement d'un bouton permettant de sauvegarder et de charger automatiquement le dernier fichier enregistré lors du débogage d'une application utilisant **DevExpress WinForms**.

WinForms, ou **Windows Forms**, est une bibliothèque de Microsoft qui permet de créer des interfaces graphiques pour des applications Windows. Cela permet de concevoir des fenêtres et des éléments interactifs, comme des boutons et des menus.

```
    else
    {
        diagramControll1.LoadDocument(diagramName);
    }
}
1 référence
private void barButtonItemSave_ItemClick(object sender, DevExpress.XtraBars.ItemClickEventArgs e)
{
    diagramControll1.SaveDocument(diagramName);
}
```



- **Lecture et traitement d'un fichier .txt** : Lecture du contenu d'un fichier .txt, qui contenait des lettres et des chiffres. Chaque ligne du fichier a été représentée par un rectangle dans une interface graphique.

```
using (StreamReader sr = new StreamReader("ZONE EN COURS.txt"))
{
    string line;
    var y = 10;

    DiagramShape diagramItemXXXX = new DiagramShape();
    while ((line = sr.ReadLine()) != null)
    {
        DiagramShape diagramItem1 = new DiagramShape(BasicShapes.Rectangle, 450, y, 300, 100);
        diagramItem1.Content = line;
        diagramControll1.Appearance.Shape.Font = new Font("Times New Roman", 40);
    }
}
```

- **Mise en forme conditionnelle des rectangles** : Application de règles de mise en forme où les rectangles représentant des éléments commençant par la lettre "m" étaient colorés en bleu, tandis que les autres étaient en jaune.

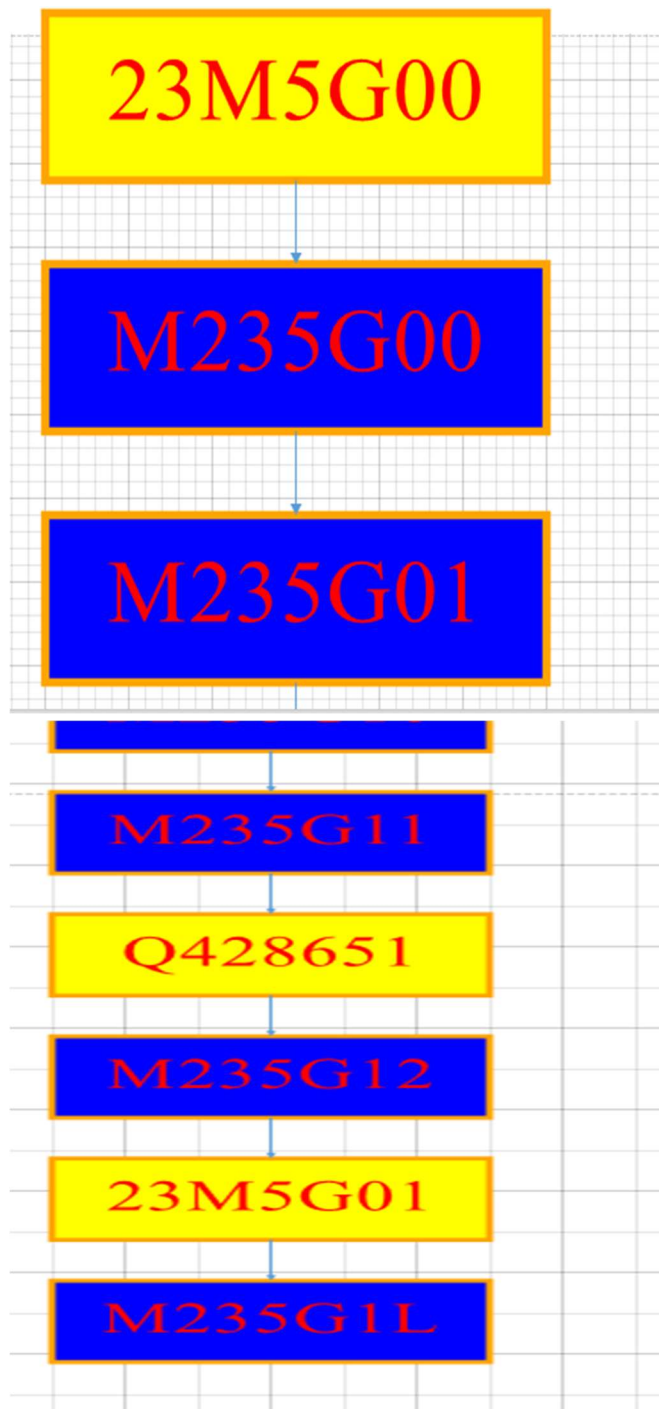
```
if (line.StartsWith("M"))
{
    diagramItem1.Appearance.BackColor = Color.Blue;
}
else
{
    diagramItem1.Appearance.BackColor = Color.Yellow;
}
```

- **Placement de flèches entre les rectangles** : Ajout d'une fonctionnalité pour dessiner des flèches reliant les rectangles dans l'ordre de leur disposition, représentant visuellement les relations entre les éléments.

```
diagramControl1.Items.Add(diagramItem1);
y += 150;
if (diagramItemXXXX.Content != null)
{
    DiagramConnector connector1 = new DiagramConnector(diagramItemXXXX, diagramItem1);
    connector1.EndArrow = ArrowDescriptions.Filled90;
    diagramControl1.Items.Add(connector1);
}

diagramItemXXXX = diagramItem1;
```

Ce qui donne :



Gestion des utilisateurs dans GLPI :

- **Filtrage des utilisateurs via Active Directory** : Modification d'un script en C# pour récupérer uniquement les noms des personnes travaillant dans le service informatique en Espagne à partir de l'**Active Directory**.

Active Directory est un service de gestion des identités et des ressources réseau développé par Microsoft. Il permet d'organiser et de gérer les utilisateurs, les groupes, les ordinateurs et d'autres objets dans un réseau d'entreprise.

```
static void Main(string[] args)
{
    //
    // string ip_address = ; //MAF FR
    string ip_address = ; //MAF ES,
    string filter = "OU=MIS,OU=RIB,DC=mafroda,DC=es";
    using (PrincipalContext context = new PrincipalContext(ContextType.Domain, ip_address, filter))
    {
        UserPrincipal userPrincipal = new UserPrincipal(context);
        using (PrincipalSearcher searcher = new PrincipalSearcher(userPrincipal))
        {
            foreach (Principal result in searcher.FindAll())
            {
                if (result is UserPrincipal user)
                {
                    if (!user.SamAccountName.StartsWith("vpn") & (user.DisplayName != null) )
                    {
                        if (Convert.ToBoolean(user.Enabled))
                        {
                            Console.WriteLine("username: " + user.SamAccountName);
                            Console.WriteLine("Name: " + user.Name);
                            Console.WriteLine("Display Name: " + user.DisplayName);
                            Console.WriteLine("User Principal Name: " + user.UserPrincipalName);
                            Console.WriteLine("Distinguished Name: " + user.DistinguishedName);
                            Console.WriteLine("Email: " + user.EmailAddress);
                            Console.WriteLine("Description: " + user.Description);
                            Console.WriteLine("Employee ID: " + user.EmployeeId);
                            Console.WriteLine();
                        }
                    }
                }
            }
        }
    }
}

string filter = "OU=MIS,OU=RIB,DC=mafroda,DC=es";
using (PrincipalContext context = new PrincipalContext(ContextType.Domain, ip_address, filter))
```

- **Intégration dans GLPI** : Ajout de ces utilisateurs au groupe « Informatique Espagnol » dans **GLPI**, avec attribution des rôles de technicien et d'administrateur.

GLPI (Gestionnaire Libre de Parc Informatique) est un logiciel libre pour la gestion des services informatiques. Il permet de suivre les incidents, gérer les demandes de support, les équipements et les actifs, ainsi que de gérer les utilisateurs en les ajoutant, les organisant en groupes, et en attribuant des rôles et des permissions.



MAF RIB = Espagne

Sélectionnez l'entité souhaitée

Astuce : vous pouvez appeler cette fenêtre modale avec la combinaison de touches **Ctrl + Alt + E**

Cliquez sur l'icone  pour charger les éléments de l'entité sélectionnée ainsi que de ses sous-entités.






Chercher une entité


  

MAF-GROUPE 

- MAF-FRANCE
- MAF-RIB
- MAF-US

Administration

-  Utilisateurs
-  Groupes
-  Entités
-  Règles
-  Dictionnaires
-  Profils
-  File d'attente des notifications
-  Journaux
-  Inventaire

Actions   

NOM COMPLET	ENTITÉ	COMMENTAIRES
<input type="checkbox"/> IT SERVICE INFORMATIQUE	MAF-GROUPE ▶ MAF-RIB	Contains all members of the IT department (RIB).
<input type="checkbox"/> IT SERVICE INFORMATIQUE > DEVELOPPEMENTS / APPLICATIONS	MAF-GROUPE ▶ MAF-RIB	Contains internal/external application developments (RIB)
<input type="checkbox"/> IT SERVICE INFORMATIQUE > DEVELOPPEURS AS400 / MACPAC	MAF-GROUPE ▶ MAF-RIB	Contains MACPAC/AS400 Developers (RIB)
<input type="checkbox"/> IT SERVICE INFORMATIQUE > INFRASTRUCTURES / SYSTEME	MAF-GROUPE ▶ MAF-RIB	Contains technicians for Infrastructure and system (RIB)

20 lignes / page De 1 à 4 sur 4 lignes

Groupes - IT SERVICE INFORMATIQUE
MAF-GROUPE > MAF-RIB
Sous-entités
Actions
1/4 >>

Groupes

Sous-groupes

Éléments utilisés

Éléments gérés

Liaison annuaire LDAP

Utilisateurs

Notifications

Tickets

Problèmes

Changements

Notes

Historique

Tous

Ajouter un utilisateur

----- i Responsable Non Délégué Non Ajouter

Utilisateurs

Critère ----- Sous-groupes Non

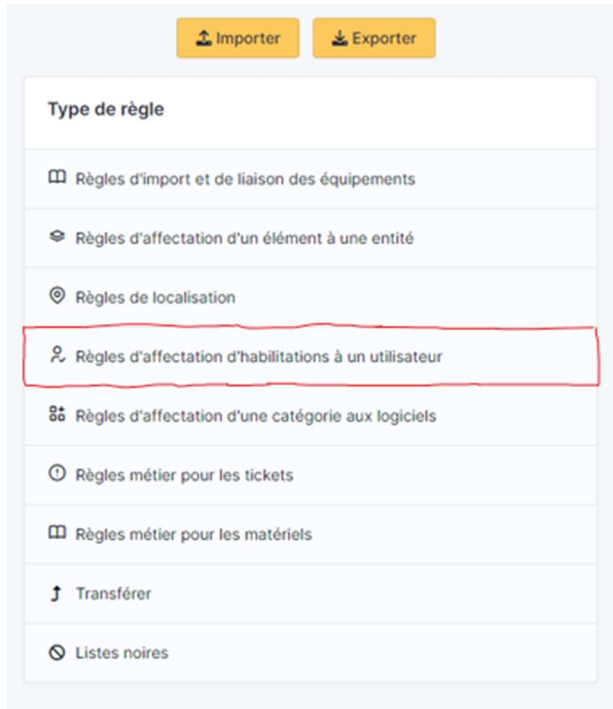
Utilisateurs (D=Dynamique)

Affichage (nombre d'éléments) 20 De 1 à 8 sur 8

Utilisateur	Dynamique	Responsable	Délégué	Actif
████████████████████				✔
████████████████████				✔
████████████████████				✔
████████████████████				✔
████████████████████				✔
████████████████████				✔
████████████████████				✔
████████████████████				✔
████████████████████				✔
████████████████████				✔

Actions

- **Création et configuration des règles** : Mise en place de règles dans GLPI pour attribuer automatiquement les rôles de technicien et d'administrateur selon des critères définis. Ajout d'une règle similaire pour les utilisateurs français du service informatique.



Voici les règles que j'ai créé :

Nom	Description	Critères	Actions	Actif
<input type="checkbox"/> Tech_ES	Technicien	(LDAP)DistinguishedName > contient > OU=MIS,OU=RIB,DC=mafroda,DC=es Identifiant > n'est pas [redacted]	Profils > Assigner > Technicien Récursif > Assigner > Oui	●
<input type="checkbox"/> Admin_ES	administrateur	(AD)User ID > est [redacted]	Profils > Assigner > Admin Récursif > Assigner > Oui	●
<input type="checkbox"/> Tech_FR	Technicien	(LDAP)DistinguishedName > contient > OU=INFO,OU=MAF-RODA,DC=reseauglobal,DC=maf-roda,DC=com Identifiant > n'est pas [redacted] Identifiant > n'est pas [redacted] Identifiant > n'est pas [redacted] Identifiant > n'est pas [redacted] Identifiant > n'est pas [redacted] Identifiant > n'est pas [redacted] Identifiant > n'est pas [redacted] Identifiant > n'est pas [redacted] Identifiant > n'est pas [redacted] Identifiant > n'est pas [redacted] Identifiant > n'est pas [redacted] Identifiant > n'est pas [redacted] Identifiant > n'est pas [redacted]	Profils > Assigner > Technicien Récursif > Assigner > Oui	●
<input type="checkbox"/> Admin_FR	administrateur	(LDAP)DistinguishedName > contient > OU=BEMECA,OU=MAF-RODA,DC=reseauglobal,DC=maf-roda,DC=com (LDAP)DistinguishedName > contient [redacted]	Profils > Assigner > Admin Récursif > Assigner > Oui	●
<input type="checkbox"/> Tech_FR [redacted]	Technicien	(LDAP)DistinguishedName > contient > OU=CRTOFBE,OU=BEMECA,OU=MAF-RODA,DC=reseauglobal,DC=maf-roda,DC=com (LDAP)DistinguishedName > contient [redacted]	Profils > Assigner > Technicien Récursif > Assigner > Oui	●

Création de la règle Tech_ES :

« < [] Règle - Tech_ES [] Actions v 16/60 > »

Règle

Nom: Tech_ES Description: Technicien

Opérateur logique: et Actif: Oui

Commentaires:

Dernière mise à jour le 20-06-2024 15:38

Tester

Supprimer définitivement Sauvegarder

« < [] Règle - Tech_ES [] Actions v »

Règle

Ajouter un nouveau critère

Actions

Historique

Tous

Critères	Condition	Motif
<input type="checkbox"/> (LDAP)DistinguishedName	contient	OU=MIS,OU=RIB,DC=mafroda,DC=es
<input type="checkbox"/> Identifiant	n'est pas	
Critère	Condition	Motif

Actions

« < [] Règle - Tech_ES [] Actions v 1 »

Règle

Il est possible d'affecter le résultat d'une expression régulière en utilisant la chaîne #0

Ajouter une nouvelle action

Historique

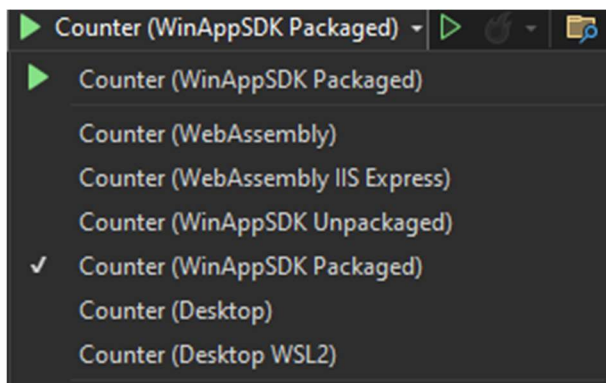
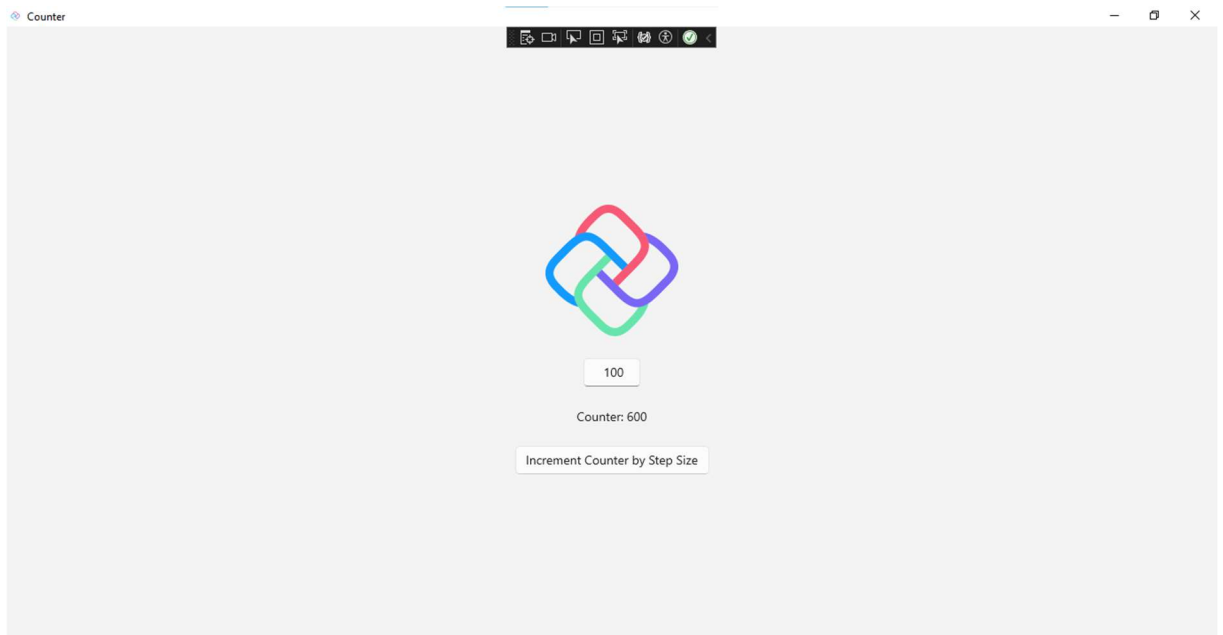
Tous

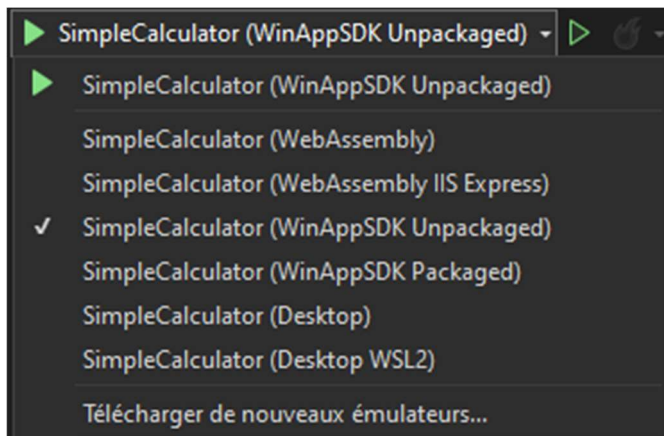
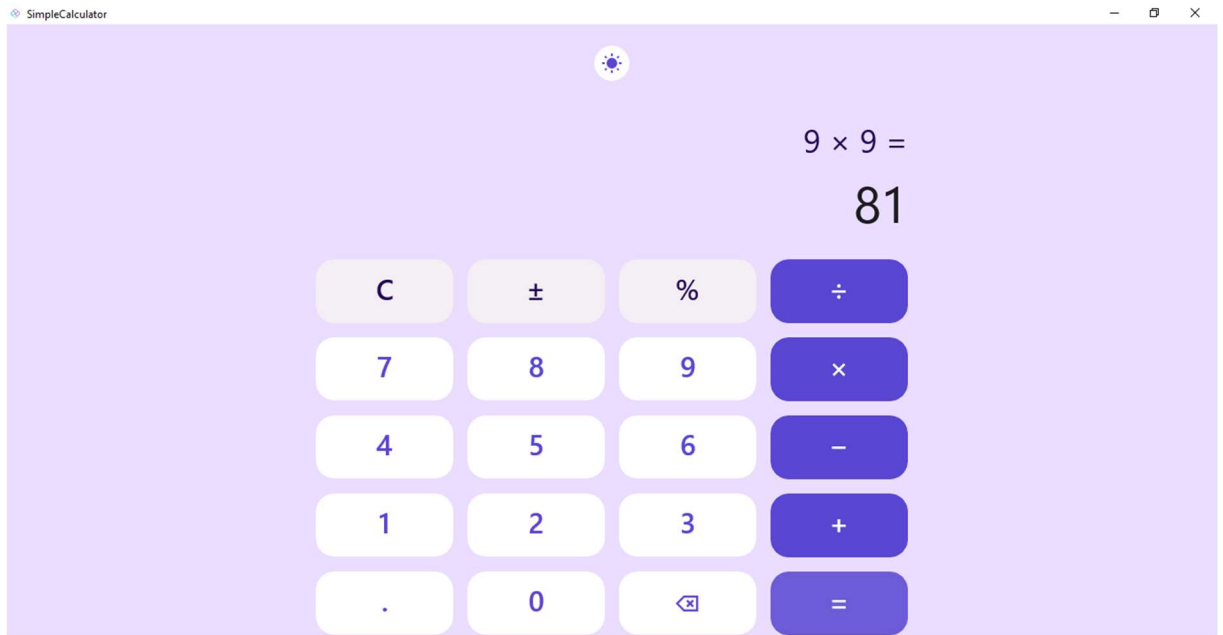
Actions	Champs	Type d'action	Valeur
<input type="checkbox"/> Profils	Assigner		Technician
Champs	Type d'action	Valeur	

Actions

Projets de démonstration avec Uno Platform :

- **Réalisation d'applications multi-plateformes** : Réalisation de deux projets de démonstration utilisant **Uno Platform**, un outil permettant de créer des applications pour diverses plateformes (iOS, Android, Linux, Windows, navigateur) à partir du même code source.





Réalisation de deux projets de démonstration utilisant **Uno Platform** pour évaluer la faisabilité et les capacités de l'outil. Mon maître de stage avait une problématique spécifique liée au développement d'une application pour un employé chargé et déchargeant des camions, qui devait être compatible avec une tablette. Il m'a donc demandé de faire des tests.

Contrainte

J'ai rencontré des difficultés dès le début de mon stage, car je manquais de connaissances. Je ne connaissais rien, ce qui m'a rendu complètement perdu au départ. Ensuite, j'ai dû me familiariser avec une API web sur GitHub et l'adapter pour qu'elle soit compatible avec des données de l'AS400. Malheureusement, c'était trop complexe pour mon niveau.

Au-delà de cet exercice, j'ai dû plus largement me familiariser avec les outils, les langages et les technologies utilisés dans l'entreprise. J'ai également rencontré des difficultés en supervision avec mon maître de stage surtout au début, car je ne connaissais rien dans ce domaine non plus. Cependant, j'ai rapidement pris la main et cela s'est mieux passé. Quant aux exercices que l'on me donnait, ils restaient assez compliqués tout au long du stage.

Sources consultées

1. **Stack Overflow** - Plateforme communautaire pour poser des questions et trouver des solutions à des problèmes de programmation :
<https://stackoverflow.com>.
2. **Microsoft Learn** - Documentation officielle de Microsoft, avec des tutoriels et des démos sur ASP.NET Web API, .NET, et d'autres technologies :
 - Documentation générale : <https://learn.microsoft.com>.
 - Exemple "Owin Self Host Sample" : <https://learn.microsoft.com/en-us/aspnet/web-api/overview/hosting-aspnet-web-api/use-owin-to-self-host-web-api>.
 - Exemple "Product APP" : <https://learn.microsoft.com/en-us/aspnet/web-api/overview/getting-started-with-aspnet-web-api/tutorial-your-first-web-api>.
3. **DevExpress Documentation** - Documentation officielle de DevExpress, utilisée pour se renseigner sur les méthodes et suivre certains tutoriels pour WinForms :
 - Documentation générale : <https://docs.devexpress.com>.
 - Exemples : <https://github.com/DevExpress-Examples/wpf-winforms-maui-shared-codebase>.
 - Démos : <https://docs.devexpress.com/WindowsForms/116015/controls-and-libraries/editors-and-simple-controls/lookup-editors/standard-binding-to-simple-data-types>.

4. **Uno Platform** Documentation - Documentation et tutoriels pour Uno Platform, utilisée pour développer des applications multi-plateformes :

- Documentation générale : <https://platform.uno/docs>.
- Exemple "Counter" : <https://platform.uno/docs/articles/getting-started/counterapp/get-started-counter-xaml-mvux.html?tabs=vs#wrap-up>.
- Exemple "Simple Calc" : <https://platform.uno/docs/articles/external/workshops/simple-calc/README.html>.

Conclusion

En conclusion, j'ai pu m'intégrer dans le milieu professionnel et observer le fonctionnement d'un service informatique en entreprise. Ce stage m'a permis de m'améliorer, même si j'ai rencontré des difficultés régulièrement. J'ai beaucoup appris et élargi mes connaissances. Cette expérience a été très bénéfique, car mon maître de stage a su trouver un bon équilibre entre autonomie et accompagnement.

Cet apprentissage m'a préparé à ma future vie active, en m'aidant à gérer les défis quotidiens d'un environnement de travail et à mieux comprendre les attentes du milieu professionnel, ce qui me sera très utile dans ma carrière. Je remercie encore MAF Roda de m'avoir accueilli et de m'avoir intégré dans leur équipe. Ce stage était idéal, car il représentait la suite logique de mon BTS SIO SLAM, étant donné que le service développait des logiciels pour l'entreprise.