

# enia @ Script d'extinction d'un ordinateur

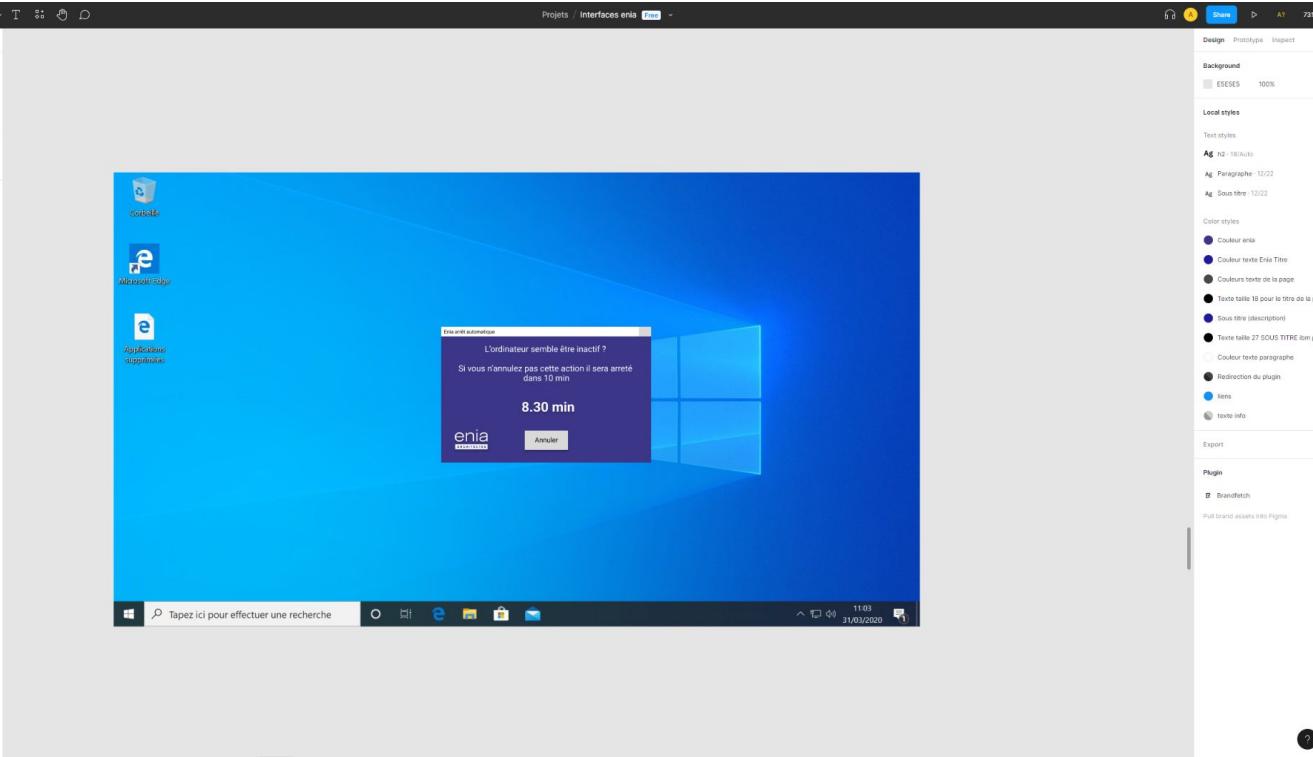
Eteindre un ordinateur à une heure précise pour réduire la consommation électrique



# Plan

1. [Création de l'affichage sur Figma](#)
2. [Création de l'affichage sur Visual studio](#)
3. [Explication du script](#)
4. [Planificateur de tâche](#)

# Création de l'affichage sur Figma



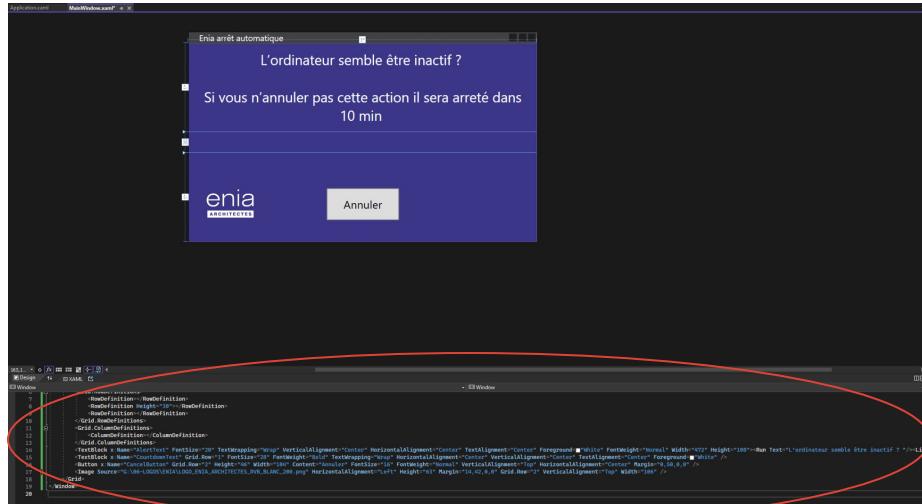
- Le but est d'avoir un affichage interactif et visible pour l'utilisateur.
- Celui-ci comportera un timer et un bouton annuler.
- La charte est celle d'Enia avec une couleur adapté à l'agence

# Création de l'affichage sur Visual Studio



Il faut installer Visual Studio Community. Il faut aussi installer la bibliothèque .NET framework.

# Création de l'affichage sur Visual Studio



Une fois qu'on a créé le visuel sur Visual Studio, on peut récupérer le code XAML qu'on retrouve juste en bas.

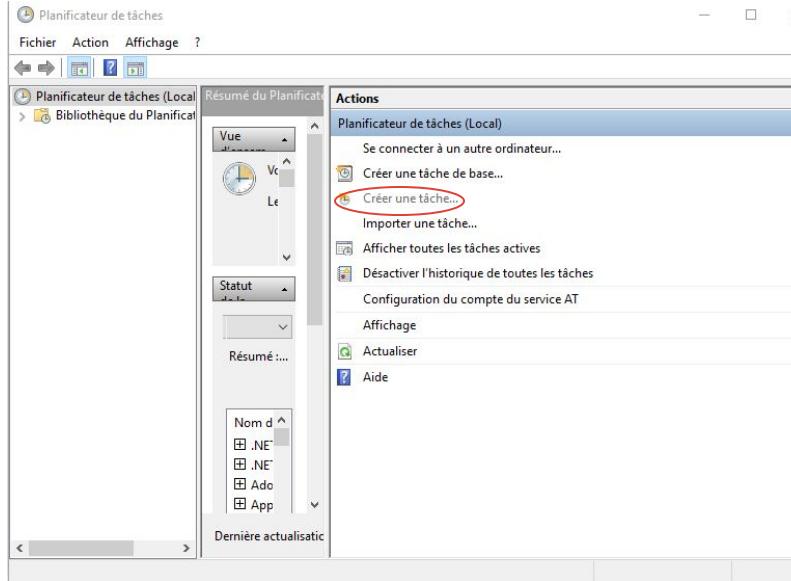
On pourra l'intégrer à notre script PowerShell.

# Explication du script

```
scriptpoweroff.ps1 x
1 Param(
2     [Parameter()]
3     [int]$TimeOut = 600 # Default to 30 second countdown
4 )
5
6 # Load WPF assembly
7 Add-type -AssemblyName PresentationFramework
8
9 # Define WPF window
10 $window = [Windows.Markup.XamlReader]::Load(
11     [Xml.XmlNodeReader]::new([xml]$xaml)
12     <window xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
13         xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
14         xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
15         Title="enia arrêt automatique" Height="300" Width="500" MaxHeight="500" MinHeight="300" ShowInTaskbar="False" Topmost="True" WindowStartupLocation="CenterScreen" WindowStyle="Toc
16     <Grid Background="#FF30306B">
17         <RowDefinition>
18             <RowDefinition Height="30"></RowDefinition>
19             <RowDefinition></RowDefinition>
20             <RowDefinition></RowDefinition>
21         </Grid>
22         <Grid.ColumnDefinitions>
23             <ColumnDefinition>
24             <ColumnDefinition>
25                 <TextBlock x:Name="CountdownText" Grid.Row="1" FontSize="28" FontWeight="Bold" TextWrapping="Wrap" VerticalAlignment="Center" HorizontalAlignment="Center" TextAlignment="Center" Foreground="White" FontWeight="Normal" Width="472" Height="108"><Run
26                     <textBlock></textBlock></Run>
27                     <TextBlock x:Name="CancelButton" Grid.Row="2" Height="46" Width="104" Content="Annuler" FontSize="16" FontWeight="Normal" VerticalAlignment="Top" HorizontalAlignment="Center" Margin="0,50,0,0" />
28                     <Image Source="\\172.18.10.1\enia\000 - GENERALITES\06-LOGOS\ENIA\LOGO_ENIA_ARCHITECTES_RVB_BLANC_200.png" HorizontalAlignment="Left" Height="63" Margin="14,42,0,0" Grid.Row="2" VerticalAlignment="Top" Width="106
29                 </TextBlock>
30             </Grid>
31         </Grid>
32     </Window>
33 )
34
35 # Get controls to manipulate
36 $CountdownText = $window.FindName('CountdownText')
37 $CancelButton = $window.FindName('CancelButton')
38
39 # Add event handler to cancel button
40 $CancelButton.Add_Click({$window.Close()})
41
42 $window.Add_SourceInitialized({
43     Script:seconds = $TimeOut
44
45     $CountdownText.Text = "$($script:seconds)s"
46
47     $Script:timer = New-Object System.Windows.Threading.DispatcherTimer
48     $Script:timer.Interval = ([TimeSpan]'0:0:1.0') # Fire every second
49
50     $Script:timer.Add_Tick({
51         $CountdownText.Text = "$($script:seconds)s"
52
53         if($script:seconds -le 0) {
54             Stop-Computer -Force
55         }
56     })
57
58     $Script:timer.Start()
59 })
60
61 # Show the window
62 $window.ShowDialog() | Out-Null
```

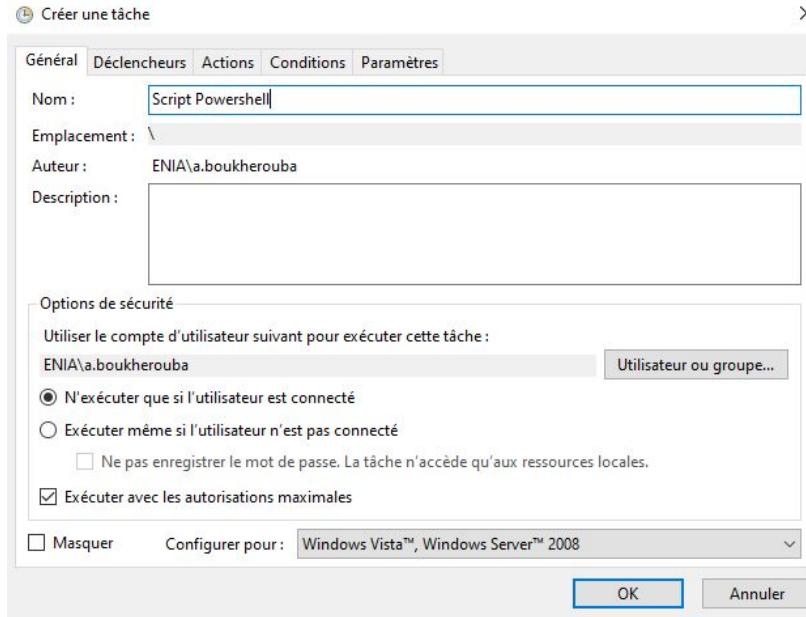
- Le script ici présent va stocker dans une variable le timer qui sera de 10 min donc 600 sec.
- On va charger l'affichage grâce au code XML qu'on a réalisé sur Visual Studio.
- On va définir les actions aux boutons sur l'affichage pour le bouton annuler, le changement de seconde et le bouton croix rouge.
- On va définir l'intervalle qui est en seconde puis dès que le timer sera écoulé, nous allons éteindre l'ordinateur.

# Planificateur de tâche



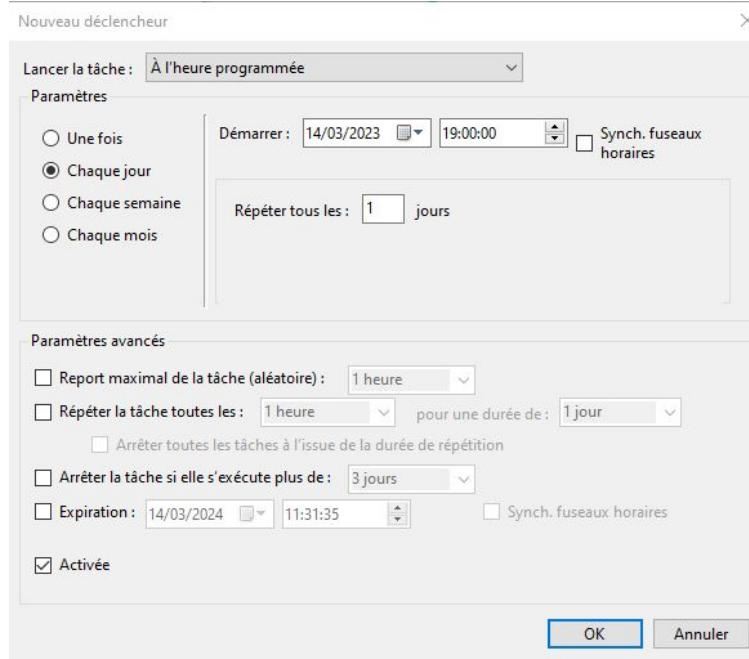
- On crée une tâche dans le planificateur de tâche.

# Planificateur de tâche



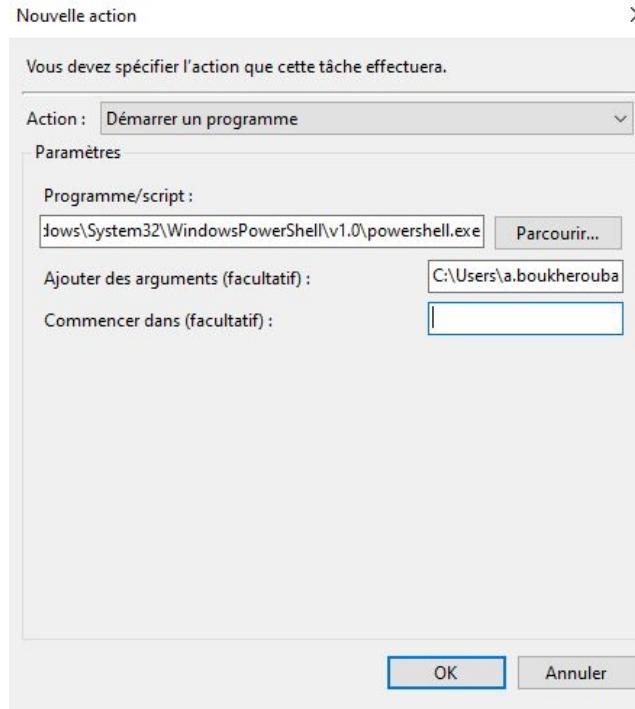
On donne un nom à notre tâche puis on exécute si l'utilisateur est connecté (impossible quand l'utilisateur n'est pas connecté) et on exécute avec les autorisations maximales.

# Planificateur de tâche



- On crée une tâche avec un déclencheur qui sera une heure programmée à 19h pile.

# Planificateur de tâche



On crée une action en utilisant PowerShell et en ajoutant un argument qui sera le script.