

Curso Superior de Tecnologia em Gestão da Produção Industrial

API

GESTEC – Soluções Empresariais

SPRINT 3

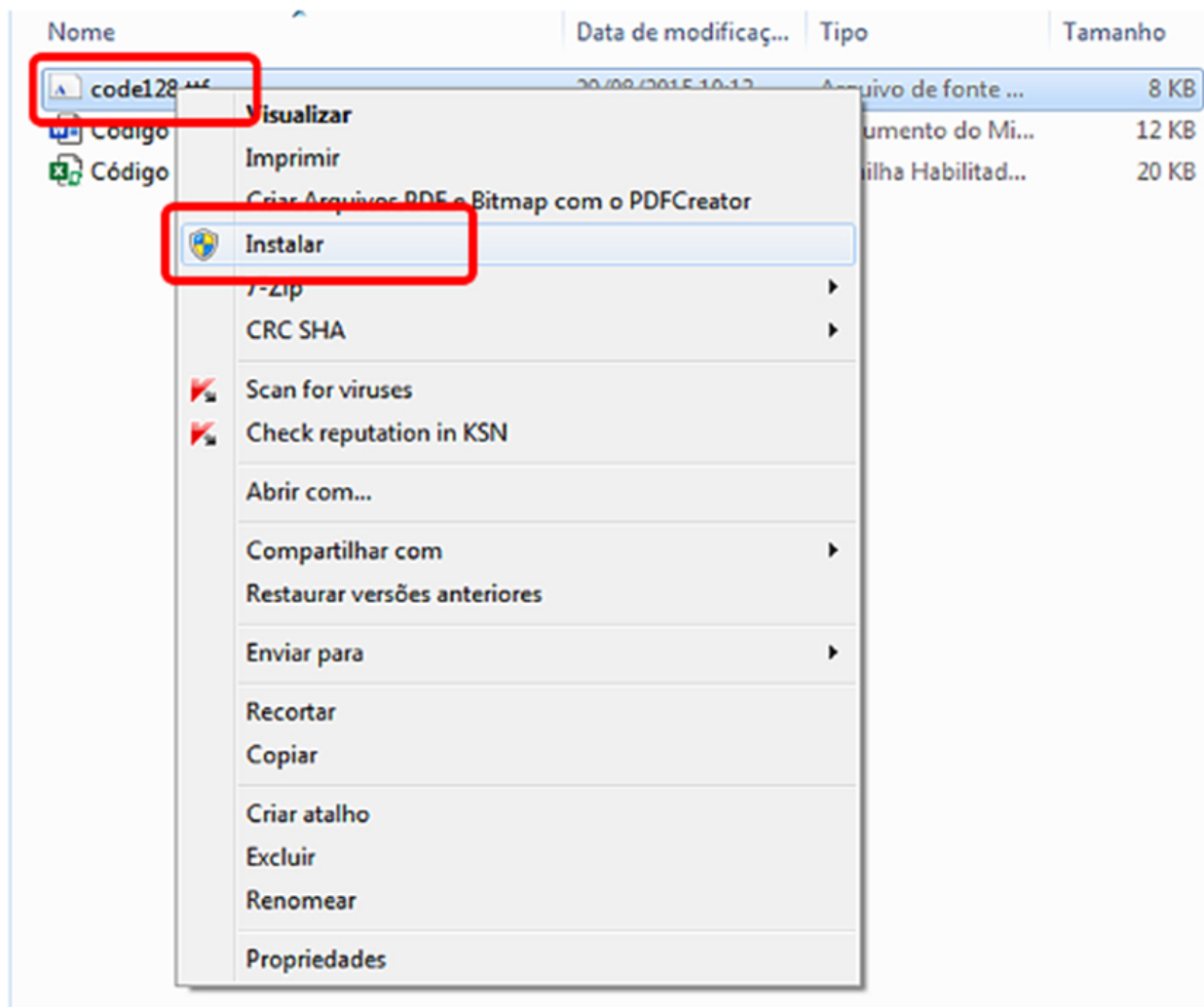
Para implementação da Sprint 3, segue o passo-a-passo:

Passo 1:

Fazer o download da fonte code128 do site: <https://dicasdoexcel.com.br/codigo-debarras-no-excel/>

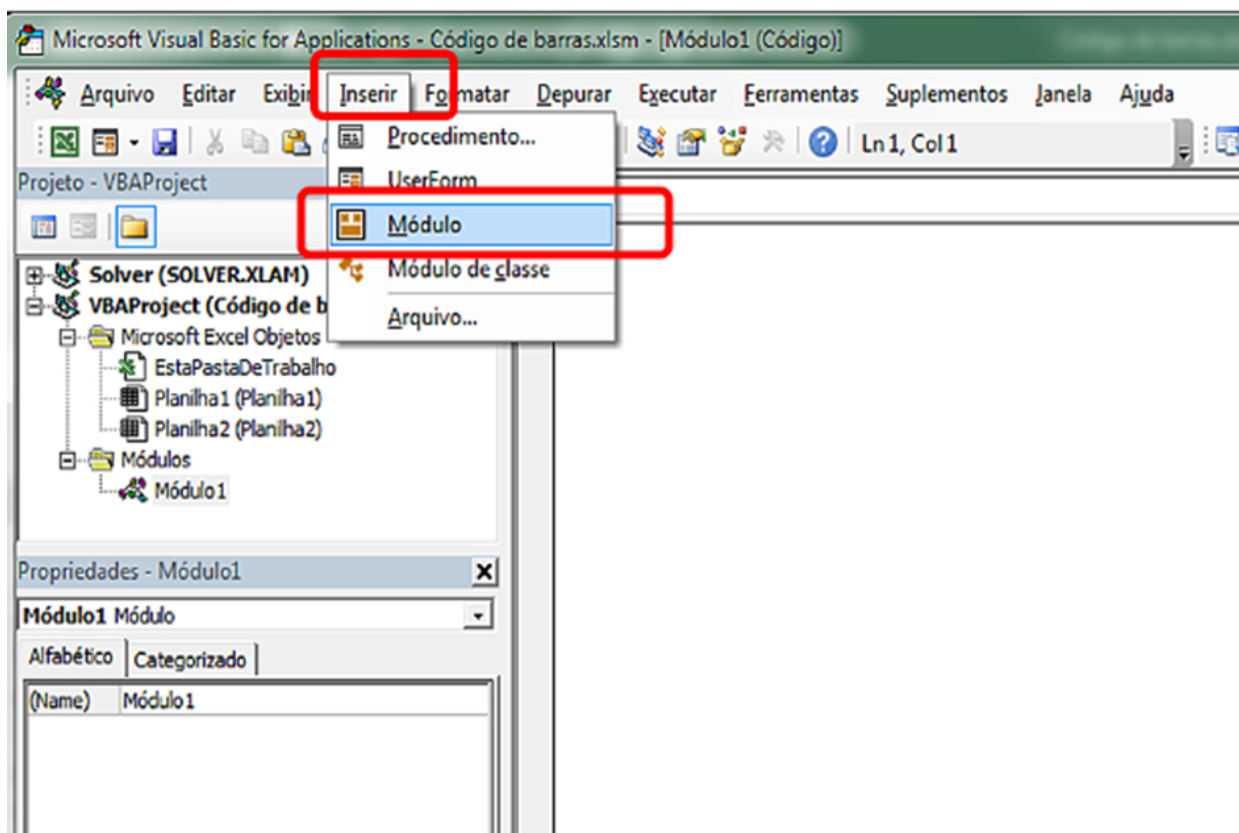
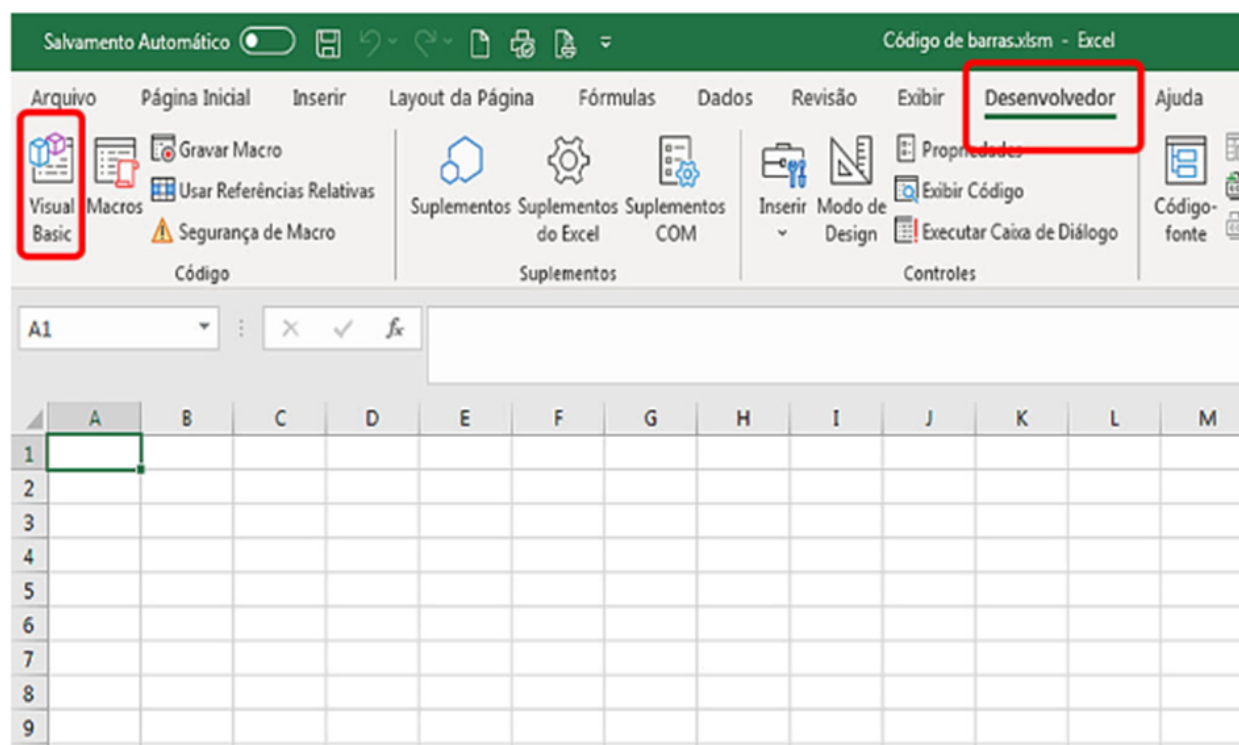
Passo 2:

Instalar a fonte no computador



Passo 3:

Inserir esse código fonte no VBA do Excel. Para isso, será necessário habilitar a guia Desenvolvedor/ Visual Basic e inserir módulo novo.



Deve-se então inserir o código:

```
Public Function code128$(chaine$)

Dim i%, checksum%, mini%, dummy%, tableB As Boolean

code128$ = ""

If Len(chaine$) > 0 Then

For i% = 1 To Len(chaine$)

Select Case Asc(Mid$(chaine$, i%, 1))

Case 32 To 126, 203

Case Else

i% = 0

Exit For

End Select

Next

code128$ = ""

tableB = True

If i% > 0 Then

i% = 1

Do While i% <= Len(chaine$)

If tableB Then

mini% = If(i% = 1 Or i% + 3 = Len(chaine$), 4, 6)

GoSub testnum

If mini% < 0 Then

If i% = 1 Then

code128$ = Chr$(205)

Else

code128$ = code128$ & Chr$(199)

End If

End If
```

```
tableB = False

Else

If i% = 1 Then code128$ = Chr$(204)

End If

End If

If Not tableB Then

mini% = 2

GoSub testnum

If mini% < 0 Then

dummy% = Val(Mid$(chaine$, i%, 2))

dummy% = IIf(dummy% < 95, dummy% + 32, dummy% + 100)

code128$ = code128$ & Chr$(dummy%)

i% = i% + 2

Else

code128$ = code128$ & Chr$(200)

tableB = True

End If

End If

If tableB Then

code128$ = code128$ & Mid$(chaine$, i%, 1)

i% = i% + 1

End If

Loop For i% = 1 To Len(code128$)

dummy% = Asc(Mid$(code128$, i%, 1))

dummy% = IIf(dummy% < 127, dummy% - 32, dummy% - 100)

If i% = 1 Then checksum& = dummy%

checksum& = (checksum& + (i% - 1) * dummy%) Mod 103

Next
```

```
checksum& = If(checksum& < 95, checksum& + 32, checksum& + 100)

code128$ = code128$ & Chr$(checksum&) & Chr$(206)

End If

End If

Exit Function

testnum:

mini% = mini% - 1

If i% + mini% <= Len(chaine$) Then

Do While mini% >= 0

If Asc(Mid$(chaine$, i% + mini%, 1)) < 48 Or Asc(Mid$(chaine$, i% + mini%, 1)) > 57 Then Exit Do

mini% = mini% - 1

Loop



End If

Return

End Function
```

Passo 4:

Após, pode-se criar os códigos de barras para os componentes da planilha de estoque ao colocar na célula a função code128:

| | A | B | C |
|---|--------------|----|--|
| 1 | Componente 1 | 18 |  |
| 2 | Componente 2 | 15 |  |

Passo 4:

Em seguida deve-se criar o arquivo do sistema de gerenciamento de estoque na ferramenta de desenvolvimento para programadores de Python, o PyCharm Community Edition que é a versão gratuita que pode ser baixada via web. Será necessário baixar também algumas bibliotecas no Python.

```
import cv2
```

```
import zbar
```

```
import smtplib
```

```
import openpyxl
```

```
# Importa as bibliotecas de código de barras, e-mail e planilhas Excel
```

```
def ler_codigo_de_barra(imagem):
```

```
# Captura o código de barras da imagem
```

```
codificador = zbar.ImageScanner()
```

```
codificador.scan(imagem)
```

```
# Obtém o conteúdo do código de barras for barras in codificador.results:
```

```
return barras.data
```

```
def subtrair_valor_celula(planilha, coluna, linha):
```

```
# Obtém o valor da célula
```

```
valor = planilha.cell(linha, coluna).value
```

```
# Subtrai 1 do valor
```

```
novo_valor = valor - 1
```

```
# Altera o valor da célula
```

```
planilha.cell(linha, coluna).value = novo_valor
```

```
def enviar_email(conteudo):
```

```
# Configura as informações do e-mail
```

```
remetente = "gmarianamartins@gmail.com"
```

```
destinatario = "Gilberto.martins@ccm-ita.org.br"
```

```
assunto = "Saída de estoque"
```

Cria uma conexão com o servidor de e-mail

```
servidor = smtplib.SMTP("smtp.gmail.com", 587)
```

```
servidor.starttls() servidor.login("gmarianamartins@gmail.com", "senha1234")
```

Envia o e-mail corpo = f"Um componente {conteudo} saiu do estoque. O estoque está com {novo_valor} unidades restantes."

```
servidor.sendmail(remetente, destinatario, f"Subject: {assunto}\n\n{corpo}")
```

Fecha a conexão com o servidor de e-mail servidor.quit()

Captura uma imagem do código de barras imagem = cv2.imread("codigo_de_barra.jpg")

Lê o conteúdo do código de barras codigo_de_barra = ler_codigo_de_barra(imagem)

Abre a planilha Excel planilha = openpyxl.load_workbook("estoque.xlsx")

Obtém a coluna e a linha da célula a ser alterada

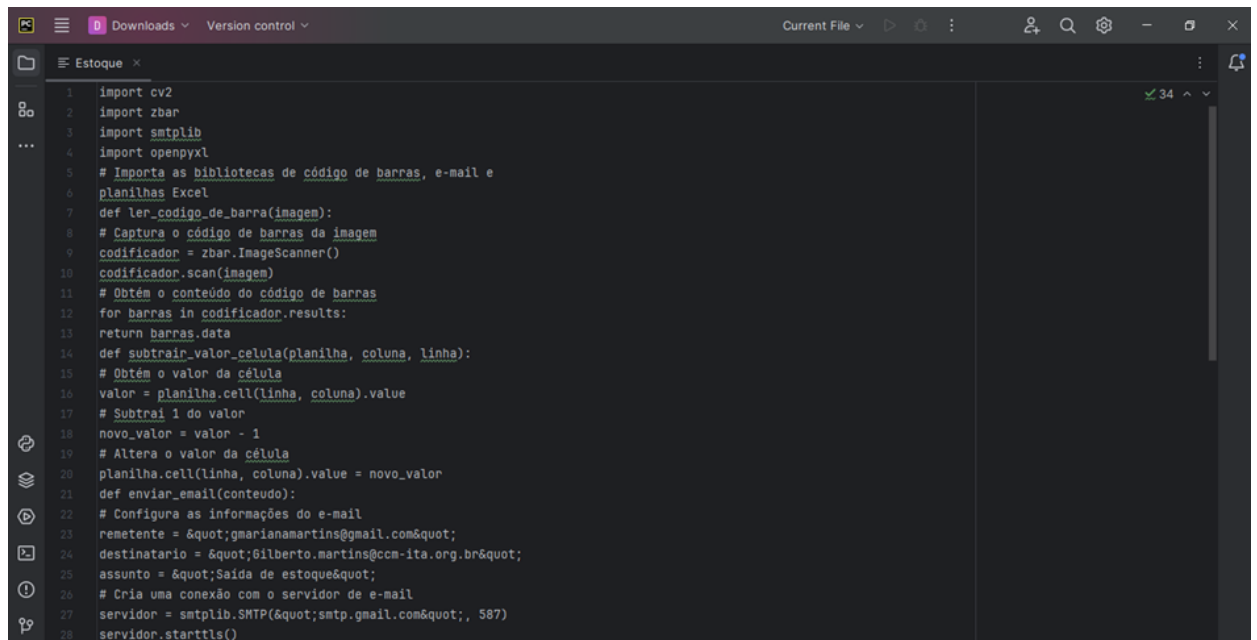
```
coluna = 1
```

```
linha = 2
```

Subtrai o valor da célula subtrair_valor_celula(planilha, coluna, linha)

Salva a planilha Excel planilha.save("estoque.xlsx")

Envia o e-mail enviar_email(codigo_de_barra) Ficará assim no PyCharm:



```
1 import cv2
2 import zbar
3 import smtplib
4 import openpyxl
5 # Importa as bibliotecas de código de barras, e-mail e
6 planilhas Excel
7 def ler_codigo_de_barra(imagem):
8     # Captura o código de barras da imagem
9     codificador = zbar.ImageScanner()
10    codificador.scan(imagem)
11    # Obtém o conteúdo do código de barras
12    for barras in codificador.results:
13        return barras.data
14    def subtrair_valor_celula(planilha, coluna, linha):
15        # Obtém o valor da célula
16        valor = planilha.cell(linha, coluna).value
17        # Subtrai 1 do valor
18        novo_valor = valor - 1
19        # Altera o valor da célula
20        planilha.cell(linha, coluna).value = novo_valor
21    def enviar_email(conteudo):
22        # Configura as informações do e-mail
23        remetente = "gmarianamartins@gmail.com"
24        destinatario = "Gilberto.martins@ccm-ita.org.br"
25        assunto = "Saída de estoque"
26        # Cria uma conexão com o servidor de e-mail
27        servidor = smtplib.SMTP("smtp.gmail.com", 587)
28        servidor.starttls()
```

```
1 import cv2
2 import zbar
3 import smtplib
4 import openpyxl
5 # Importa as bibliotecas de código de barras, e-mail e
6 planilhas Excel
7 def ler_codigo_de_barra(imagem):
8     # Captura o código de barras da imagem
9     codificador = zbar.ImageScanner()
10    codificador.scan(imagem)
11    # Obtém o conteúdo do código de barras
12    for barras in codificador.results:
13        return barras.data
14    def subtrair_valor_celula(planilha, coluna, linha):
15        # Obtém o valor da célula
16        valor = planilha.cell(linha, coluna).value
17        # Subtrai 1 do valor
18        novo_valor = valor - 1
19        # Altera o valor da célula
20        planilha.cell(linha, coluna).value = novo_valor
21    def enviar_email(conteudo):
22        # Configura as informações do e-mail
23        remetente = "marianamartins@gmail.com"
24        destinatario = "Gilberto.martins@ccm-ita.org.br"
25        assunto = "Saída de estoque"
26        # Cria uma conexão com o servidor de e-mail
27        servidor = smtplib.SMTP("smtp.gmail.com", 587)
28        servidor.starttls()
```

→ A planilha de estoque do excel deverá ficar salva na mesma pasta do arquivo de sistema de gerenciamento de estoque do PyCharm.

Funcionamento do Programa:

Este programa funciona da seguinte forma:

→ A função `ler_codigo_de_barra()` captura o código de barras da imagem e retorna o seu conteúdo.

→ A função `subtrair_valor_celula()` obtém o valor da célula, subtrai 1 do valor e altera o valor da célula.

→ A função `enviar_email()` configura as informações do e-mail e envia o e-mail com o conteúdo do código de barras.