

**Observatório Universal de Projetos
Códigos/Ferramentas**

Histórico de Alterações

Autor	Data	Descrição
Álvaro Pinheiro	20/05/2018	Criação do doc de códigos para captura de dados

Finalidade: Varrer diretório para capturar arquivos com a extensão especificada

```
public String[] LerPasta(String pPasta, String sExtensao)
{
    String sourceDirectory = pPasta;
    String[] txtFiles = null;
    try
    {
        txtFiles = System.IO.Directory.GetFiles(sourceDirectory, sExtensao,
SearchOption.AllDirectories);
    }
    catch (Exception ex)
    {
        Geral.Alerta(this, ex.Message, "Alerta");
    }
    return (txtFiles);
}
```

Finalidade: Descompactar arquivos zipados para obter os dados do mesmo

```
public void DescompactarZIP(String sZip)
{
    try
    {
        String sourceDirectory = Server.MapPath(".") + sZip;
        String[] txtFiles = LerPasta(sourceDirectory);
        foreach (String currentFile in txtFiles)
        {
            String fileName = currentFile.Substring(sourceDirectory.Length);
            String sourceDirectoryBkp = Server.MapPath(".") + sZip;
            String fileNameBkp = "Bkp_" +
DateTime.Now.ToString("yyyy_MM_dd_hh_mm_") +
currentFile.Substring(sourceDirectory.Length);
            System.IO.File.Copy(sourceDirectory + fileName, sourceDirectoryBkp +
fileNameBkp);
            ZipFile zip = new ZipFile(sourceDirectory + fileName);
            zip.ExtractAll(sourceDirectory,
ExtractExistingFileAction.OverwriteSilently);
            zip.Dispose();
            System.IO.File.Delete(sourceDirectory + fileName);
        }
    }
    catch (Exception ex)
    {
        String sEx = ex.Message;
    }
}
```

Finalidade: Ler conteúdo de arquivo para fazer pesquisa

```
public static byte[] LerArquivo(string pCaminhoArquivo)
{
    byte[] buffer;
    System.IO.FileStream fs = new System.IO.FileStream(pCaminhoArquivo,
System.IO.FileMode.Open, System.IO.FileAccess.Read);
    try
    {
        int length = (int)fs.Length;
        buffer = new byte[length];
        int count;
```

Observatório Universal de Projetos Códigos/Ferramentas

```
        int sum = 0;
        while ((count = fs.Read(buffer, sum, length - sum)) > 0)
            sum += count;
    }
    finally
    {
        fs.Close();
    }
    return buffer;
}
```

Finalidade: Pesquisa conteúdo de arquivo texto na busca de palavras chaves

```
public string PesquisarTXT(String pTexto, String pChave, string pFinal)
{
    String sRetorno = "";
    try
    {
        int nPosicaoInicial = pTexto.IndexOf(pChave);
        if (nPosicaoInicial != 0)
        {
            String sOcorrencia = pTexto.Substring(nPosicaoInicial);
            sOcorrencia = sOcorrencia.Replace(pChave, "");
            int nPosicaoFinal = sOcorrencia.IndexOf(pFinal);
            sOcorrencia = sOcorrencia.Substring(0, nPosicaoFinal);
            sRetorno = sOcorrencia.Trim();
        }
    }
    catch (Exception ex)
    {
        Geral.Alerta(this, ex.Message, "Alerta");
    }
    return (sRetorno.Trim().Replace("<BR>", "").Replace("<br>", ""));
}
```

Finalidade: Converte PDF em Texto

```
public string ExtrairTextoPDF(String caminho)
{
    String sRetorno = "";
    try
    {
        using (iTextSharp.text.pdf.PdfReader leitor = new
iTextSharp.text.pdf.PdfReader(caminho))
        {
            System.Text.StringBuilder texto = new System.Text.StringBuilder();
            for (int i = 1; i <= leitor.NumberOfPages; i++)
            {
                texto.Append(iTextSharp.text.pdf.parser.PdfTextExtractor.GetTextFromPage(leitor,
i));
            }
            sRetorno = texto.ToString();
        }
    }
    catch (Exception ex)
    {
        Geral.Alerta(this, ex.Message, "Alerta");
    }
    return (sRetorno.Trim());
}
```

Finalidade: Insere linhas de uma planilha numa base de dados

```
public void ExcelImportacao(String sTipo, String sArquivo, String sPasta)
{
}
```

Observatório Universal de Projetos Códigos/Ferramentas

```
String sNomeArquivo = "";
String sArquivoCaminho = "";
String sExtensao = "";
String sCmd = "";
long nTamanho = 0;
sNomeArquivo = System.IO.Path.GetFileName(sArquivoCaminho);
if (sNomeArquivo == "")
{
    Geral.Alerta(this, "Informe a planilha para importação!", "Aviso");
}
else
{
    sExtensao = System.IO.Path.GetExtension(sArquivoCaminho).ToLower();
    if (sExtensao != ".xlsx")
    {
        Geral.Alerta(this, "Não é uma planilha excel!", "Aviso");
    }
    else
    {
        sExtensao = "application/vnd.ms-excel";
        nTamanho = ImportarEsgotados.PostedFile.InputStream.Length;
        System.IO.Stream oStream = ImportarEsgotados.PostedFile.InputStream;
        System.IO.BinaryReader oBinario = new
System.IO.BinaryReader(oStream);
        Byte[] bytes = oBinario.ReadBytes((Int32)oStream.Length);

        ImportarEsgotados.PostedFile.SaveAs(Session["CaminhoFisico"].ToString() +
sNomeArquivo);
        String sConexao = @"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" +
Session["CaminhoFisico"].ToString() + sNomeArquivo + "; Extended Properties=Excel 8.0";
        System.Data.OleDb.OleDbConnection oconn = new
System.Data.OleDb.OleDbConnection(sConexao);
        System.Data.OleDb.OleDbCommand ocmd = new
System.Data.OleDb.OleDbCommand("SELECT * FROM [" + sPasta + "$]", oconn);
        oconn.Open();
        System.Data.OleDb.OleDbDataReader odr = ocmd.ExecuteReader();
        if (odr.FieldCount == 0)
        {
            Geral.Alerta(this, "Não foi possível abrir a planilha!",
"Aviso");
        }
        else
        {
            while (odr.Read())
            {
                ArrayList al = new ArrayList();
                for (int i = 0; i <= 24; i++)
                {
                    object val = odr[i];
                    if (val == DBNull.Value)
                    {
                        al.Add("");
                    }
                    else
                    {
                        al.Add(val.ToString().Replace("\n",
"<br>").Replace(" ", "").Replace("00:00:00",""));
                    }
                }
                StringBuilder sb = new StringBuilder();
                sb.Append(sCmd);
                for (int i = 0; i <= 24; i++)
                {
```

Observatório Universal de Projetos Códigos/Ferramentas

```
sb.Append("'" + al[i].ToString().Replace("'", "") +
"',");
    }
    System.Data.SqlClient.SqlConnection conn = new
System.Data.SqlClient.SqlConnection(ConfigurationManager.AppSettings["BancoConexao"].To
String());
    System.Data.SqlClient.SqlCommand cmd = new
System.Data.SqlClient.SqlCommand();
    cmd.Connection = conn;
    cmd.CommandText = sb.ToString();
    cmd.CommandType = CommandType.Text;
    conn.Open();
    cmd.ExecuteNonQuery();
    conn.Close();
    }
    oconn.Close();
}
}
}
```