

## EJERCICIO

Al analizar el código, el mismo se encarga de obtener un CV el cual a continuación es parseado. Retornando únicamente los datos necesarios.

```
static List<List<string>> read_csv_stream(Stream dataFile, string filename,
bool ingnorar_primerlinea)
{
    List<List<string>> datos = new List<List<string>>();

    StreamReader sr = new StreamReader(dataFile);

    string sline = "";
    string concat = "";
    string celda = "";
    string[] _values = null;
    int cont = 0;
    while (!sr.EndOfStream)
    {
        cont ++;
        sline = sr.ReadLine();

        bool ban = false;
        int cont2 = 0;

        foreach (int item in sline)
        {
            if (item == 34) /* " */ { ban = true; cont2++; }

            if (ban & cont2 < 2) { celda += item; }

            if (cont2 == 2) { concat += celda.Replace(",", ""); celda = "";

            if (cont2 == 0) { concat += "$$" + item; }

            if (cont2 == 2) { concat += item; ban = false; cont2 = 0; }
        }

        sline = concat.Replace("\"", "");

        if (ingnorar_primerlinea && cont == 1) continue;

        _values = sline.Split(',');
        List<string> row = new List<string>();
```

```
foreach (var str in _values)
{
    var val = str.Replace("&coma;", ",");
    val = val.Replace("&vacio", "");
    row.Add(val);
    val = null;
}

row.Add(cont.ToString());
datos.Add(row);
row = null;
concat = "";
celda = "";

return datos;
}
sr.Close();
}
```