



# **Práctica**

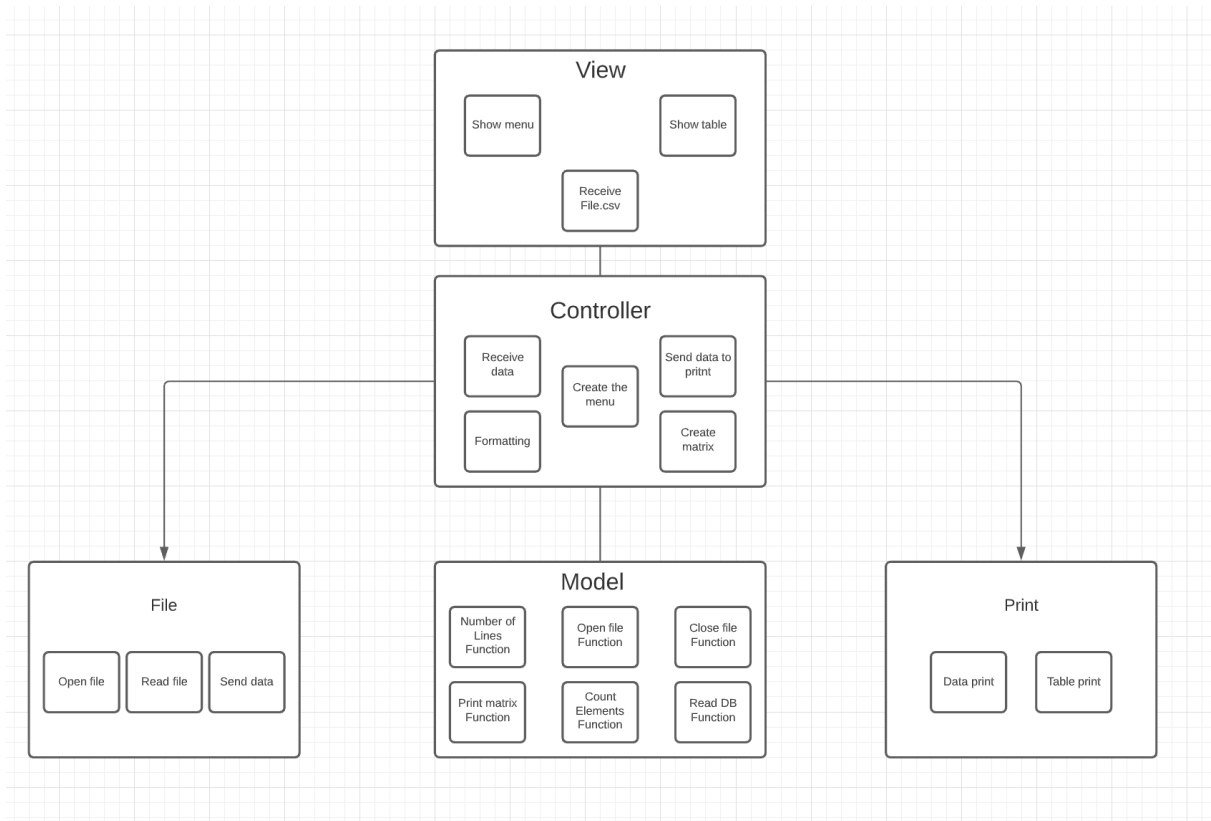
## **“Data Shell.”**

Integrantes:  
Samantha Morris  
Mauro Bernal

Taller de Desarrollo de Aplicaciones

19/03/2021

# MVC



## View

**Main menu:**

```
Hi! :)
Name of File: 
```

**Second menu:**

```
What do you wanna do?:  
[1] Print data.  
[2] Exit.  
Answer: █
```

# Pseudocode

## Model

```
#include "DataShell.h"  
  
//Function that imprimes a Matrix  
Funcion imprimeMatrix (float **Matrix, int *numLines, int *numElements)  
{  
    repite(int i = 0; i < *numLines; i++)  
    {  
        repite(int j = 0; j < *numElements; j++)  
        {  
            imprime("%.6f ", *((*(Matrix+i))+j));  
        }  
        fin repite  
        imprime("\n");  
    }  
    fin repite  
}  
  
fin funcion  
  
//Function that reads DB and stores it in a Matrix  
Funcion readDB (float **Matrix, char *fileName, int numElements)  
{  
    char buff[255];  
    char * value;  
    float *ptr = NULL;  
    int cont = 0, n =0;  
  
    FILE *file = openMyfile(fileName);  
  
    mientras(!feof(file))  
    {  
        lee(file, "%s", buff);  
  
        //numElements = countElements(buff);  
        float *ptr = malloc(sizeof(float) * numElements);  
        value = strtok(buff, ",");  
  
        mientras(value != NULL)
```

```

    {
        ptr[cont] = atof(value); //Stores it in ptr GUARDO LOS ELEMENTOS EN EL POINTER
        //imprime("%.6f ", ptr[cont]);
        cont ++;
        value = strtok(NULL, ",");
    }
    //imprime("\n");
    cont = 0;
    Matrix[n] = ptr; //MATRIX AHORA APUNTA A PTR
    n ++;

}
fin mientras

closeMyfile(file);

//imprime("\n%.6f %.6f\n", *((*(Matrix+49))+0), *((*(Matrix+49))+1) );

return;
}

```

Fin funcion

//Function that tells how many elements does a csv line has  
 Functon countElements(char \*fileName)

```

{
    int numElements = 0;
    char buff[255];

    FILE *file = openMyfile(fileName);
    lee(file, "%s", buff);
    char* value = strtok(buff, ",");

    mientras(value != NULL)
    {
        numElements ++;
        value = strtok(NULL, ",");
    }
    fin mientras
    closeMyfile(file);
    return numElements;
}

```

//Function that opens a file  
 FILE\* openMyfile (char \*fileName)

```

{
    FILE *file = fopen(fileName, "r");

    if(file == NULL)
    {
        imprime("Couln't open file\n");
        exit(0);
    }
    return file;
}

```

//Function that closes a file  
 Funcion closeMyfile(FILE \*file)

```

{
    fclose(file);
}

```

```

}
fin funcion

//Function that return number of lines
funcion NumberLines(char *fileName)
{
    int numLines = 0;
    char buff[255];

    FILE *file = openMyfile(fileName);

    mientras(!feof(file))
    {
        lee(file, "%s", buff);
        numLines ++;
    }

    fin mientras

    closeMyfile(file);

    return numLines;
}

fin funcion

```

## View

```

#include "DataShell.h"

funcion menuOne(char **fileName)
{
    system("clear");
    imprime("Name of File: ");
    scanf("%m[^\\n]", fileName); //Reads until it finds a \\n
    imprime("Reading from: %s...\\n", *fileName);
}
fin funcion

funcion menuTwo(void)
{
    int option;

    system("clear");
    imprime("\\nWhat do you wanna do?:\\n[1] imprime data. \\n[2] Exit.\\nAnswer: ");
    lee("%d", &option);
    imprime("\\n");
    system("clear");
    return(option);
}
fin funcion

```

## Controller

```
#include "DataShell.h"
```

```
Funcion main (void)
```

```
{  
    char* fileName=NULL;  
    int numLines = 0, numElements;  
    int option;  
  
    menuOne(&fileName);  
  
    numLines = NumberLines(fileName);  
    numElements = countElements(fileName);  
  
    float **Matrix = malloc(sizeof(float) * numLines * numElements);  
  
    readDB(Matrix, fileName, numElements);  
  
    haz{  
        option = menuTwo();  
        if(option == 1)  
        {  
            imprimeMatrix(Matrix, &numLines, &numElements);  
            imprime("\nPress [enter] to go back to the menu.\n");  
            __fpurge(stdin);  
            getchar();  
        }  
    }mientras(option != 2);  
  
    fin haz mientras  
  
    free(Matrix);  
    return 0;  
}
```

```
fin funcion
```