Inputs.

Programación I – Laboratorio I.
Tecnicatura Superior en Programación.
UTN-FRA

Autores: *Lic. Mauricio Dávila*

Revisores: *Ing. Ernesto Gigliotti*

Versión : 1

Índice de contenido

1 Biblioteca Inputs

Representa a una familia de funciones que analizan una entrada de datos con formato y cargan el resultado en el argumento que se pasa por referencia a dichas funciones. Todas ellas leen caracteres, los interpretan según un formato, validan, y almacenan los resultados en el argumento input.

1.1 Función getInt

```
/**
 * \brief Solicita un número al usuario y lo valida
 * \param input Se carga el numero ingresado
 * \param message Es el mensaje a ser mostrado
 * \param eMessage Es el mensaje a ser mostrado en caso de error
 * \param lowLimit Limite inferior a validar
 * \param hiLimit Limite superior a validar
 * \return Si obtuvo el numero [0] si no [-1]
 *
 */
int getInt(int* input,char message[],char eMessage[], int lowLimit, int hiLimit)
{
      //.........
}
```

Ejemplo:

```
int edad;
int r; // Respuesta
r = getInt(&edad,"¿Cual es tu edad?","Rango valido [0 - 100]",1,100);
if(r == 0)
      printf("La edad es: %d",edad);
```

1.2 Función getFloat

```
/**
 * \brief Solicita un número al usuario y lo valida
 * \param input Se carga el numero ingresado
 * \param message Es el mensaje a ser mostrado
 * \param eMessage Es el mensaje a ser mostrado en caso de error
 * \param lowLimit Limite inferior a validar
 * \param hiLimit Limite superior a validar
 * \return Si obtuvo el numero [0] si no [-1]
 *
 */
int getFloat(float* input,char message[],char eMessage[], float lowLimit, float hiLimit)
{
      //.........
}
```

Ejemplo:

```
float precio;
int r; // Respuesta
r = getFloat(&precio,"¿Cual es el precio?","Rango valido [0 - 10000]",1,10000);
if(r == 0)
      printf("El precio es: %f",precio);
```

1.3 Función getChar

```
/**
 * \brief Solicita un caracter al usuario y lo valida
 * \param input Se carga el caracter ingresado
 * \param message Es el mensaje a ser mostrado
 * \param eMessage Es el mensaje a ser mostrado en caso de error
 * \param lowLimit Limite inferior a validar
 * \param hiLimit Limite superior a validar
 * \return Si obtuvo el caracter [0] si no [-1]
 *
 */
int getChar(char* input,char message[],char eMessage[], char lowLimit, char hiLimit)
{
      //.........
}
```

Ejemplo:

```
char continuar;
int r; // Respuesta
r = getChar(&continuar,"Ingrese Opcion [A][B][C]","Solo [A][B][C]",'A','C');
if(r == 0)
      printf("Continuar: %c",continuar);
```

1.4 Función getString

```
/**
 * \brief Solicita una cadena de caracteres al usuario y la valida
 * \param input Se carga el string ingresado
 * \param message Es el mensaje a ser mostrado
 * \param eMessage Es el mensaje a ser mostrado en caso de error
 * \param lowLimit Longitud mínima de la cadena
 * \param hiLimit Longitud máxima de la cadena
 * \return Si obtuvo la cadena [0] si no [-1]
 *
 */
int getString(char* input,char message[],char eMessage[], int lowLimit, int hiLimit)
{
      //.........
}
```

Ejemplo:

```
char nombre[51];
int r; // Respuesta
r = getString(nombre,"Nombre: ","El largo debe ser entre 2 y 50", 2, 50);
if(r == 0)
      printf("Nombre: %s",nombre);
```