



Recuperación 1

Programación de Redes

Profesor: Gabriel Barron Rodríguez

Alumno: Alan Francisco Emmanuel Aguilar Fuentes

Laboratorios Python 2

## Contenido

Laboratorios.....	1
2.3.1.18.....	1
2.4.1.6.....	1
2.8.1.4.....	2
3.2.1.14.....	2
3.2.1.15.....	3
3.2.1.16.....	3
3.4.1.12.....	4
3.4.1.13.....	4
3.4.1.14.....	5
3.4.1.15.....	5
4.3.1.15.....	6
4.3.1.16.....	6
4.3.1.17.....	7
4.4.1.8.....	7
4.5.1.22.....	8
4.6.1.13.....	8

# Laboratorios

## 2.3.1.18

The screenshot shows a web browser with two tabs. The left tab is titled 'Plataforma Educativa Institución' and displays the website of the Universidad Tecnológica del Norte de Guanajuato. The right tab is titled 'Edube Interactive : 2.3.1.18 Tu p...' and shows a code editor with a Python script. The script defines a function `split(string)` that splits a string into a list of words. The console output shows the result of calling `split('Ser o no ser, esa es la pregunta')`, which is `['Ser', 'o', 'no', 'ser,', 'esa', 'es', 'la', 'pregunta']`.

**Objetivos**

- Mejorar las habilidades del alumno al trabajar con cadenas.
- Utilizar los métodos incorporados de Python para las cadenas.

**Escenario**

Ya sabes como funciona el método `split()`. Ahora queremos que lo pruebes.

Tu tarea es **escribir tu propia función, que se comporte casi como el método original** `split()`, por ejemplo:

- Debe aceptar únicamente un argumento: una cadena.
- Debe devolver una lista de palabras creadas a partir de la cadena, dividida en los lugares donde la cadena contiene espacios en blanco.
- Si la cadena está vacía, la función debería devolver una lista vacía.
- Su nombre debe ser: `mysplit()`.

## 2.4.1.6

The screenshot shows a web browser with two tabs. The left tab is titled 'Plataforma Educativa Institución' and displays the website of the Universidad Tecnológica del Norte de Guanajuato. The right tab is titled 'Edube Interactive : 2.4.1.6 LABO...' and shows a code editor with a Python script. The script defines a function `display_siete_segmentos(numero)` that takes a number and returns a list of strings representing the seven segments of a digital display. The console output shows the result of calling `display_siete_segmentos(67)`, which is `['###', '###', '###', '###', '###', '###', '###']`.

**Objetivos**

- Mejorar las habilidades del alumno al trabajar con cadenas.
- Usar cadenas para representar datos que no son texto.

**Escenario**

Seguramente has visto un *display de siete segmentos*.

Es un dispositivo (a veces electrónico, a veces mecánico) diseñado para presentar un dígito decimal utilizando un subconjunto de siete segmentos. Si aún no sabes lo qué es, consulta la siguiente liga en Wikipedia: [https://en.wikipedia.org/wiki/Seven-segment\\_display](https://en.wikipedia.org/wiki/Seven-segment_display).

Tu tarea es **escribir un programa que puede simular el funcionamiento de un display de siete segmentos**, aunque vas a usar LEDs individuales en lugar de

#### 2.8.1.4

Universidad Tecnológica del Norte de Guanajuato  
"Educación y progreso para la vida"

TSU - Ingenierías/Licenciaturas - Manuales/Entrenamiento

**Bienvenido a la Plataforma de Aprendizaje**

¡Inscríbete y comienza a diseñar tu futuro!

ALAN FRANCISCO EMMANUEL AGUILAR

Área personal

Perfil

Calificaciones

### Objetivos

- Mejorar las habilidades del alumno al definir funciones.
- Utilizar excepciones para proporcionar un entorno de entrada más seguro.

### Escenario

Tu tarea es escribir una **función capaz de ingresar valores enteros y verificar si están dentro de un rango especificado.**

La función deberá:

- Aceptar tres argumentos: una entrada, un límite inferior aceptable y un límite superior aceptable.

Python 3.8.10

```
def obtener_valor_en_rango(minimo, maximo):
    """
    Función que obtiene un valor de entrada y verifica si está dentro de un rango permitido.
    """
    try:
        entrada = int(input(f'Ingresar un número entre (minimo) y (maximo): '))
        if minimo <= entrada <= maximo:
            return entrada
        else:
            print(f'Error: el valor no está dentro del rango permitido ((minimo)..(maximo))')
    except ValueError:
        print('Error: entrada incorrecta')

if __name__ == '__main__':
    # Obtener valor en rango (-10, 10)
    valor = obtener_valor_en_rango(-10, 10)
    print(f'El número es: {valor}')
```

Console >...

```
Ingresar un número entre -10 y 10: -2
El número es: -2
Ingresar un número entre -10 y 10: 0.5
Error: entrada incorrecta
Ingresar un número entre -10 y 10: 04
El número es: 4
```

## 3.2.1.14

The screenshot shows a web browser with two tabs. The active tab is 'Edube Interactivo :: 3.2.1.14 Pila', displaying a course page for '4 Pila Contadora'. The page header includes the logo of Universidad Tecnológica del Norte de Guanajuato and navigation links for 'TSU', 'Ingenierías/Licenciaturas', and 'Manuales/Entrenamiento'. The main content area features a circular profile picture of a man and a woman, the name 'ALAN FRANCISCO EMMANUEL AGUILAR', and a blue button labeled 'Bienvenido a la Pila'. Below this, there are links for 'Área personal', 'Perfil', and 'Calificaciones'. The 'Objetivos' section lists two goals: 'Mejorar las habilidades del estudiante para definir clases.' and 'Emplear clases existentes para crear nuevas clases equipadas con nuevas funcionalidades.' The 'Escenario' section explains that the goal is to extend the capabilities of the 'Stack' class by defining a new class that inherits from it and adds new functionalities. The task is to extend the behavior of the 'Stack' class. The second tab is a code editor showing Python code for a 'Pila' (Stack) class. The code includes a 'pop' method that removes the top element and a 'get\_counter' method that returns the current count. The code is as follows:

```

20
21
22
23
def pop(self):
    val = self._stk[-1]
    del self._stk[-1]
    return val

def __init__(self):
    super().__init__()
    self.__counter = 0

def get_counter(self):
    return self.__counter

def pop(self):
    val = super().pop()
    self.__counter -= 1

```

The console output shows the number '100'.

### 3.2.1.15

The screenshot shows a web browser on the left and a code editor on the right. The browser displays the 'Plataforma Educativa Institución' page for 'Universidad Tecnológica del Norte de Guanajuato'. The page includes a header with the university logo and navigation links, a user profile section for 'ALAN FRANCISCO EMMANUEL AGUILAR', and a main content area with 'Objetivos' and 'Escenario' sections. The 'Escenario' section describes a queue (FIFO) and provides a task description. The code editor on the right shows a Python implementation of a queue using a list. The code includes a `QueueError` exception, a `Queue` class with `init`, `put`, and `get` methods, and a `main` function that demonstrates the queue's behavior. The console output shows the results of the queue operations: '1', 'perro', 'False', and 'Error de Cola: La cola está vacía'.

Plataforma Educativa Institución: x +

https://elearning.ut... TSU - Ingenierías/Licenciaturas - Manuales/Entrenamie

Universidad Tecnológica del Norte de Guanajuato  
"Educación y progreso para la vida"

Bienvenido a la Plataforma Educativa

ALAN FRANCISCO EMMANUEL AGUILAR

Área personal  
Perfil  
Calificaciones

**Objetivos**

- Mejorar las habilidades del estudiante para definir clases desde cero.
- Implementar estructuras de datos estándar como clases.

**Escenario**

Como ya sabes, una *pila* es una estructura de datos que realiza el modelo LIFO (último en entrar, primero en salir). Es fácil y ya te has acostumbrado a ello perfectamente.

Problemas algo nuevo ahora. Una *cola* (*queue*) es un modelo de datos caracterizado por el término **FIFO: primero en entrar, primero en salir**. Nota: una cola (fila) regular que conozcas de las tiendas u oficinas de correos funciona exactamente de la misma manera: un cliente que llegó primero también es el primero en ser atendido.

5 Colas alias FIFO >

```
class QueueError(IndexError):  
    pass  
  
class Queue:  
    def __init__(self):  
        self.items = []  
    def put(self, elem):  
        self.items.insert(0, elem)  
    def get(self):  
        if not self.items:  
            raise QueueError("La cola está vacía")  
        return self.items.pop()  
    def __str__(self):  
        return str(self.items)  
  
if __name__ == "__main__":  
    que = Queue()  
    que.put(1)  
    que.put("perro")  
    que.put(False)  
    try:  
        for i in range(4):  
            print(que.get())  
    except QueueError as e:  
        print(e)
```

Console >...

```
1  
perro  
False  
Error de Cola: La cola está vacía
```

### 3.2.1.16

The screenshot shows a web browser on the left and a code editor on the right. The browser displays the 'Plataforma Educativa Institución' page for 'Universidad Tecnológica del Norte de Guanajuato'. The page includes a header with the university logo and navigation links, a user profile section for 'ALAN FRANCISCO EMMANUEL AGUILAR', and a main content area with 'Objetivos' and 'Escenario' sections. The 'Escenario' section describes a queue (FIFO) and provides a task description. The code editor on the right shows a Python implementation of a queue using a list. The code includes a `QueueError` exception, a `Queue` class with `init`, `put`, and `get` methods, and a `main` function that demonstrates the queue's behavior. The console output shows the results of the queue operations: '1', 'perro', 'False', and 'Cola vacía'.

Plataforma Educativa Institución: x +

https://elearning.ut... TSU - Ingenierías/Licenciaturas - Manuales/Entrenamie

Universidad Tecnológica del Norte de Guanajuato  
"Educación y progreso para la vida"

Bienvenido a la Plataforma Educativa

ALAN FRANCISCO EMMANUEL AGUILAR

Área personal  
Perfil  
Calificaciones

**Objetivos**

- Mejorar las habilidades del estudiante para definir subclases.
- Agregar nueva funcionalidad a una clase existente.

**Escenario**

Tu tarea es extender ligeramente las capacidades de la clase `Queue`. Queremos que tenga un método sin parámetros que devuelva `True` si la cola está vacía y `False` de lo contrario.

Completa el código que te proporcionamos en el editor. Ejecútalo para comprobar si genera un resultado similar al nuestro.

A continuación, puedes copiar el código que usamos en el laboratorio anterior:

Realizar

2 Colas alias FIFO: parte 2 >

```
class QueueError(IndexError):  
    pass  
  
class Queue:  
    def __init__(self):  
        self.items = []  
    def put(self, item):  
        self.items.append(item)  
    def get(self):  
        if not self.is_empty():  
            return self.items.pop(0)  
        else:  
            raise QueueError("Queue is empty")  
    def is_empty(self):  
        return len(self.items) == 0  
  
class SuperQueue(Queue):  
    def __init__(self):  
        super().__init__()  
  
if __name__ == "__main__":  
    que = SuperQueue()  
    que.put(1)  
    que.put("perro")  
    que.put(False)  
    for i in range(4):  
        if not que.is_empty():  
            print(que.get())  
        else:  
            print("Cola vacía")
```

Console >...

```
Warning: The pty module has been disabled for security reasons.  
1  
perro  
False  
Cola vacía
```

```
1  
perro  
False  
Cola vacía
```

### 3.4.1.12

Plataforma Educativa Institución: x +

https://elearning.ut... TSU - Ingenierías/Licenciaturas - Manuales/Entrenamie

Universidad Tecnológica del Norte de Guanajuato  
"Educación y progreso para la vida"

Bienvenido a la Plataforma Educativa

ALAN FRANCISCO EMMANUEL AGUILAR

Área personal

Perfil

Calificaciones

**Objetivos**

- Mejorar las habilidades del estudiante para definir clases desde cero.
- Definir y usar variables de instancia.
- Definir y usar métodos.

**Escenario**

Necesitamos una clase capaz de contar segundos. ¿Fácil? No es tan fácil como podrías pensar, ya que tendremos algunos requisitos específicos.

Léelos con atención, ya que la clase sobre la que escribes se utilizará para lanzar cohetes

2 La clase Timer

```
self.__hours += 1
if self.__hours == 24:
    self.__hours = 0

prev_second(self):
self.__seconds -= 1
if self.__seconds == -1:
    self.__seconds = 59
    self.__minutes -= 1
if self.__minutes == -1:
    self.__minutes = 59
    self.__hours -= 1
if self.__hours == -1:
    self.__hours = 23

def __format_time(self, value):
    return f"{0(value)}" if value < 10 else str(value)
```

Console >...

```
23:59:59
00:00:00
23:59:59
```

### 3.4.1.13

Plataforma Educativa Institución: x +

https://elearning.ut... TSU - Ingenierías/Licenciaturas - Manuales/Entrenamie

Universidad Tecnológica del Norte de Guanajuato  
"Educación y progreso para la vida"

Bienvenido a la Plataforma Educativa

ALAN FRANCISCO EMMANUEL AGUILAR

Área personal

Perfil

Calificaciones

**Objetivos**

- Mejorar las habilidades del estudiante para definir clases desde cero.
- Definir y usar variables de instancia.
- Definir y usar métodos.

**Escenario**

Tu tarea es implementar una clase llamada Weekday. Si, tus ojos no te engañan, este

Días de la semana

```
self.__current_day = self.__current_day.__sub__(new_weekday)

subtract_days(self, n):
index = self.DAYS_OF_WEEK.index(self.__current_day)
new_index = (index - n) % len(self.DAYS_OF_WEEK)
self.__current_day = self.DAYS_OF_WEEK[new_index]

if __name__ == "__main__":
    weekday = Weekday('Lun')
    print(weekday)
    weekday.add_days(15)
    print(weekday)
    weekday.subtract_days(23)
    print(weekday)
    weekday = Weekday('Lun')
    print(weekday)
except WeekDayError:
    print("¡Lo siento, no puedo atender tu solicitud.")
```

Console >...

```
Lun
Mar
Dom
```

### 3.4.1.14

The screenshot shows a web browser with two tabs. The left tab is 'Plataforma Educativa Institución...' with the URL 'https://elearning.ut...'. It displays the header of the 'Universidad Tecnológica del Norte de Guanajuato' and a sidebar with a user profile for 'ALAN FRANCISCO EMMANUEL AGUILAR'. The main content area is titled 'Objetivos' and 'Escenario'. The 'Objetivos' section lists: 'Mejorar las habilidades del estudiante para definir clases desde cero.', 'Definir y usar variables de instancia.', and 'Definir y usar métodos.'. The 'Escenario' section says: 'Visitemos un lugar muy especial: un plano con el sistema de coordenadas cartesianas'. The right tab is 'Edube Interactive : 3.4.1.14 Puntos en un plano'. It shows a Jupyter Notebook interface with a code editor and a console. The code defines a 'Point' class with attributes 'x' and 'y', and methods 'getx()', 'gety()', 'distance\_from\_xy()', and 'distance\_from\_point()'. The console shows the output of the code: '1.4142135623730951'.

Plataforma Educativa Institución: x + -

https://elearning.ut... TSU - Ingenierías/Licenciaturas - Manuales/Entrenamie

Universidad Tecnológica del Norte de Guanajuato  
"Educación y progreso para la vida"

ALAN FRANCISCO EMMANUEL AGUILAR

Objetivos

- Mejorar las habilidades del estudiante para definir clases desde cero.
- Definir y usar variables de instancia.
- Definir y usar métodos.

Escenario

Visitemos un lugar muy especial: un plano con el sistema de coordenadas cartesianas

Edube Interactive : 3.4.1.14 Puntos en un plano

Sandbox

```
math
class Point:
    def __init__(self, x=0.0, y=0.0):
        self._x = x
        self._y = y
    def getx(self):
        return self._x
    def gety(self):
        return self._y
    def distance_from_xy(self, x, y):
        return math.hypot(self._x - x, self._y - y)
    def distance_from_point(self, point):
        return math.hypot(self._x - point.getx(), self._y - point.gety())
if __name__ == '__main__':
    point1 = Point(0, 0)
    point2 = Point(1, 1)
    print(point1.distance_from_point(point2))
    print(point2.distance_from_xy(2, 0))
```

Console >\_

1.4142135623730951  
1.4142135623730951

### 3.4.1.15

The screenshot shows a web browser with two tabs. The left tab is 'Plataforma Educativa Institución...' with the URL 'https://elearning.ut...'. It displays the header of the 'Universidad Tecnológica del Norte de Guanajuato' and a sidebar with a user profile for 'ALAN FRANCISCO EMMANUEL AGUILAR'. The main content area is titled 'Objetivos' and 'Escenario'. The 'Objetivos' section lists: 'Mejorar las habilidades del estudiante para definir clases desde cero.' and 'Emplear composición.'. The 'Escenario' section says: 'Ahora vamos a colocar la clase Point (ver Lab 3.4.1.14) dentro de otra clase. Además, vamos a poner tres puntos en una clase, lo que nos permitirá definir un triángulo. ¿Cómo podemos hacerlo?'. The right tab is 'Edube Interactive : 3.4.1.15 Triángulo'. It shows a Jupyter Notebook interface with a code editor and a console. The code defines a 'Triangle' class that inherits from 'Point' and has attributes 'vertices' and a method 'perimeter()'. The console shows the output of the code: '3.414213562373095'.

Plataforma Educativa Institución: x + -

https://elearning.ut... TSU - Ingenierías/Licenciaturas - Manuales/Entrenamie

Universidad Tecnológica del Norte de Guanajuato  
"Educación y progreso para la vida"

ALAN FRANCISCO EMMANUEL AGUILAR

Objetivos

- Mejorar las habilidades del estudiante para definir clases desde cero.
- Emplear composición.

Escenario

Ahora vamos a colocar la clase Point (ver Lab 3.4.1.14) dentro de otra clase. Además, vamos a poner tres puntos en una clase, lo que nos permitirá definir un triángulo. ¿Cómo podemos hacerlo?

Edube Interactive : 3.4.1.15 Triángulo

Sandbox

```
math
class Point:
    def __init__(self, x=0.0, y=0.0):
        self._x = x
        self._y = y
    def getx(self):
        return self._x
    def gety(self):
        return self._y
    def distance_from_xy(self, x, y):
        return math.hypot(self._x - x, self._y - y)
    def distance_from_point(self, point):
        return math.hypot(self._x - point.getx(), self._y - point.gety())
class Triangle:
    def __init__(self, vertice1, vertice2, vertice3):
        self._vertices = [vertice1, vertice2, vertice3]
    def perimeter(self):
        side1 = self._vertices[0].distance_from_point(self._vertices[1])
        side2 = self._vertices[1].distance_from_point(self._vertices[2])
        side3 = self._vertices[2].distance_from_point(self._vertices[0])
        return side1 + side2 + side3
```

Console >\_

3.414213562373095

### 4.3.1.15

Diagrama de frecuencia de caracteres

```
def car_letras(contenido):  
    contenido = contenido.lower()  
    recuentos_letras = {}  
    for caracter in contenido:  
        if caracter.isalpha() and caracter.isascii():  
            recuentos_letras[caracter] = recuentos_letras.get(caracter, 0) + 1  
    for letra in sorted(recuentos_letras):  
        print(f"{letra} -> {recuentos_letras[letra]}")  
  
except Exception as e:  
    print(f"Se produjo un error: {e}")  
  
if __name__ == "__main__":  
    Ingrese el contenido del archivo: hahajaajajaa  
a -> 6  
h -> 3  
j -> 3  
s -> 1
```

### 4.3.1.16

Diagrama de frecuencia de caracteres ordenado

```
def car_letras(entrada):  
    contenido = entrada.lower()  
    recuentos_letras = {}  
    for caracter in contenido:  
        if caracter.isalpha() and caracter.isascii():  
            recuentos_letras[caracter] = recuentos_letras.get(caracter, 0) + 1  
    recuentos_ordenados = sorted(recuentos_letras.items(), key=lambda x: x[1], reverse=True)  
    for letra, frecuencia in recuentos_ordenados:  
        print(f"{letra} -> {frecuencia}")  
  
except Exception as e:  
    print(f"Se produjo un error: {e}")  
  
if __name__ == "__main__":  
    cadena_entrada = input("Ingrese la cadena de entrada: ")  
    car_letras(cadena_entrada)
```



### 4.3.1.17

The screenshot shows a web browser window with the URL <https://elearning.ut...>. The page displays the logo of the Universidad Tecnológica del Norte de Guanajuato and a user profile for ALAN FRANCISCO EMMANUEL AGUILAR. Below the profile, there is a table with student data:

Nombre	Puntaje
Anna Boleyn	15.5
John Smith	7.0

Below the table, there is a note:

Nota:

- Tu programa debe estar completamente protegido contra todas las fallas posibles: la inexistencia del archivo, el vacío del archivo o cualquier falla en los datos de entrada; encontrar cualquier error de datos debería causar la terminación

On the right, a code editor shows a Python script for processing student data:

```
def process_line(line_number, line):
    self.process_line(line_number, line)
    self.print_report()

except FileNotFoundError:
    print(f"The file '{self.file_name}' was not found.")
except StudentsDataException as e:
    print(f"Error: {e}")
    process_line(self, line_number, line):
    try:
        parts = line.strip().split()
        if len(parts) != 3:
            raise WrongLine(line_number, line)
        first_name, last_name, points_str = parts
        points = float(points_str)
        student_key = (first_name, last_name)
        self.students_data[student_key] = self.students_data.get(student_key, 0) + points
    except ValueError:
```

```
PS C:\Users\alanf\OneDrive\Escritorio\Unidad2_R1_Prog_Redes\Modulo4> python estudiantes.py
John Smith 5
Anna Boleyn 4.5
John Smith 2
Anna Boleyn 11
Andrew Cox 1.5
```

### 4.4.1.8

The screenshot shows a web browser window with the URL <https://elearning.ut...>. The page displays the logo of the Universidad Tecnológica del Norte de Guanajuato and a user profile for ALAN FRANCISCO EMMANUEL AGUILAR. Below the profile, there is a table with student data:

Nombre	Puntaje
Anna Boleyn	15.5
John Smith	7.0

Below the table, there is a note:

Nota:

- Tu programa debe estar completamente protegido contra todas las fallas posibles: la inexistencia del archivo, el vacío del archivo o cualquier falla en los datos de entrada; encontrar cualquier error de datos debería causar la terminación

On the right, a code editor shows a Python script for finding a directory:

```
def find(path, target_dir):
    current_dir = os.path.abspath(path)
    dir_contents = os.listdir(current_dir)
    except FileNotFoundError:
        print(f"El directorio {current_dir} no existe.")
        return
    target_dir = os.path.join(current_dir, target_dir)
    print(target_dir)
    for item in dir_contents:
        item_path = os.path.join(current_dir, item)
        if os.path.isdir(item_path):
            find(item_path, target_dir)
    path = "."
    target_directory = "python"
    find(path, target_directory)
```

#### 4.5.1.22

The screenshot shows a web browser with two tabs. The left tab is 'Plataforma Educativa Institución' with the URL 'https://elearning.ut...'. It displays the header of the 'Universidad Tecnológica del Norte de Guanajuato' and a sidebar for user 'ALAN FRANCISCO EMMANUEL AGUILAR FUENTES' with links to 'Área personal', 'Perfil', and 'Calificaciones'. The main content area is titled 'Objetivos' and 'Escenario', discussing the use of the 'strftime' method. The right tab is 'Edube Interactive : 4.5.1.22 LAB...', showing a Jupyter Notebook titled 'Los módulos datetime y time'. The code in the notebook defines a 'datetime' object for November 4, 2020, and uses 'strftime' to format it in various ways. The console output shows the formatted date and time, the day of the week, and the day of the year.

```
datetime import datetime
dt = datetime.datetime(2020, 11, 4, 14, 53, 00)
ed_date_time = dt.strftime('%Y/%m/%d %H:%M:%S')
formatted_date_time = dt.strftime('%Y/%m/%d %I:%M:%S %p')
formatted_date_time_long = dt.strftime('%A, %Y %B %d')
ed_date_short = dt.strftime('%a, %Y %b %d')
formatted_date_short = dt.strftime('%A, %Y %B %d')
ed_date_long = dt.strftime('%A, %Y %B %d')
formatted_date_long = dt.strftime('%w')
number = dt.strftime('%w')
Dia de la semana = dt.strftime('%j')
day_of_year = dt.strftime('%j')
14 print("Día del año:", day_of_year)
15 print("Día de la semana:", weekday_number)
16 week_number = dt.strftime('%W')
17 print("Número de semana en el año:", week_number)
```

Console Output:

```
2020/11/04 14:53:00
20/November/04 02:53:00 PM
Wed, 2020 Nov 04
Wednesday, 2020 November 04
Día de la semana: 3
Día del año: 309
Número de semana en el año: 44
```

#### 4.6.1.13

The screenshot shows a web browser with two tabs. The left tab is 'Plataforma Educativa Institución' with the URL 'https://elearni...'. It displays the header of the 'Universidad Tecnológica del Norte de Guanajuato' and a sidebar for user 'ALAN FRANCISCO EMMANUEL AGUILAR FUENTES' with links to 'Área personal', 'Perfil', 'Calificaciones', and 'Mensajes'. The main content area is titled 'Objetivos' and 'Escenario', discussing the use of the 'Calendar' class. The right tab is 'Edube Interactive : 4.6.1.13 LAB...', showing a Jupyter Notebook titled 'El módulo calendar'. The code in the notebook defines a 'MyCalendar' class and uses it to calculate the number of times a specific day of the week appears in a given year. The console output shows the result for the year 2019.

```
from calendar import Calendar
def count_weekday_in_year(year, weekday):
    count = 0
    for day in range(1, 12):
        if any([day[weekday] != 0 for day in week]):
            count += 1
    return count

cal = MyCalendar()
result = cal.count_weekday_in_year(2019, 0)
print(f"En 2019, el lunes (día 0) aparece {result} veces en el año.")
```

Console Output:

```
En 2019, el lunes (día 0) aparece 12 veces en el año.
```