1) Functional Requirements

|  |  |
| --- | --- |
| **Name:** | R. #1 Generate prime numbers. |
| **Description:** | The program must be able to generate (n) prime numbers. It must have three algorithms that can perform this task. |
| Input: | Amount (n) of prime numbers |
|  |  |
| Output: | A table bidimensional with de first (n) prime numbers |
|  |  |

|  |  |
| --- | --- |
| **Name:** | R. #2 Get input |
| **Description:** | The program must be able to receive the input from the user via a GUI.  This input must be an integer (n) >0 and will be used to generate the prime numbers and create a matrix containing them in order to display them on the screen. |
| Input: | An input (n) that represents the maximum amount of prime numbers that must be generated. |
|  |  |
| Output: | <None> |
|  |  |

|  |  |
| --- | --- |
| **Name:** | R. #3 Generate Matrix |
| **Description:** | The program must generate a matrix containing all the integers from 0 to (n) where (n) is an input given by the user. |
| Input: | Input (n) |
|  |  |
| Output: | A matrix containing all the numbers from 0 to (n) |
|  |  |

|  |  |
| --- | --- |
| **Name:** | R. #4 Differeance the primes numbers |
| **Description:** | as the algorithm finds that the number is or is not a prime, that is, that allows to show in real time the process performed by the algorithm to find these prime numbers. |
| Input: | <None> |
|  |  |
| Output: | Green: Prime numbers |
|  | Red: Numbers not prime |

2) Class Diagram

Imagen que contiene captura de pantalla

Descripción generada automáticamente

3) Test cases design

Stage configuration

|  |  |  |
| --- | --- | --- |
| **Name** | **Class** | **Stage** |
| setupScenario1 | GeneratorTest | An object of the Generator class with n=9 |
| setupScenario2 | GeneratorTest | An object of the Generator class with n=18 |
| setupScenario3 | GeneratorTest | An object of the Generator class with n=74 |
| setupScenario4 | GeneratorTest | An object of the Generator class with n=45 |

Test Case Design

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Objective:** verify that the testIsPrime method works correctly by calculating if 5 is a prime number | | | | |
| **Class** | **Method** | **Stage** | **Input** | **Outcome** |
| GeneratorTest | testIsPrime | setupScenario1 | n = 5 | 5 is a prime number |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Objective:** verify that the testIsPrime2 method works correctly by calculating if 100 is a prime number | | | | |
| **Class** | **Method** | **Stage** | **Input** | **Outcome** |
| GeneratorTest | testIsPrime2 | setupScenario1 | n = 100 | 100 is not a prime number |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Objective:** verify that the testIsPrime3 method works correctly by calculating if 1 is a prime number | | | | |
| **Class** | **Method** | **Stage** | **Input** | **Outcome** |
| GeneratorTest | testIsPrime3 | setupScenario1 | n = 1 | 1 is not a prime number |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Objective:** Verify that the number of primes found by testAmmountOfPrimes is as expected | | | | |
| **Class** | **Method** | **Stage** | **Input** | **Outcome** |
| GeneratorTest | testAmmountOfPrimes | setupScenario1 | n = 9 | There are 4 prime numbers |
| **Test Objective:** Verify that the number of primes found by testAmmountOfPrimes2 is as expected | | | | |
| **Class** | **Method** | **Stage** | **Input** | **Outcome** |
| GeneratorTest | testAmmountOfPrimes2 | setupScenario2 | n = 18 | There are 7 prime numbers |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Objective:** Verify if the matrix distribution is the expected | | | | |
| **Class** | **Method** | **Stage** | **Input** | **Outcome** |
| GeneratorTest | matrixDistributionTest | setupScenario1 | n = 9 | The matrix will have 3 rows and 3 columns |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Objective:** Verify if the matrix distribution is the expected | | | | |
| **Class** | **Method** | **Stage** | **Input** | **Outcome** |
| GeneratorTest | matrixDistributionTest2 | setupScenario2 | n = 18 | The matrix will have 5 rows and 4 columns |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Objective:** Verify if the matrix distribution is the expected | | | | |
| **Class** | **Method** | **Stage** | **Input** | **Outcome** |
| GeneratorTest | matrixDistributionTest3 | setupScenario3 | n = 45 | The matrix will have 8 rows and 6 columns |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Objective:** Verify if the matrix distribution is the expected | | | | |
| **Class** | **Method** | **Stage** | **Input** | **Outcome** |
| GeneratorTest | matrixDistributionTest4 | setupScenario4 | n = 74 | The matrix will have 10 rows and 8 columns |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Objective:** Verify that the betwiseSieve method finds the expected prime numbers | | | | |
| **Class** | **Method** | **Stage** | **Input** | **Outcome** |
| GeneratorTest | betwiseSieveTest | setupScenario2 | n = 18 | Prime numbers are: 2,3,5,7,11,13,17 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Objective:** Verify that the betwiseSieve2 method finds the expected prime numbers | | | | |
| **Class** | **Method** | **Stage** | **Input** | **Outcome** |
| GeneratorTest | betwiseSieveTest2 | setupScenario4 | n = 45 | Prime numbers are: 2,3,5,7,11,13,17,19,23,29,31,37,41,43 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Objective:** Verify that the betwiseSieve3 method finds the expected prime numbers | | | | |
| **Class** | **Method** | **Stage** | **Input** | **Outcome** |
| GeneratorTest | betwiseSieveTest3 | setupScenario1 | n = 9 | Prime numbers are: 2,3,5,7 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Objective:** Verify that the sieveOfSundaramTest method finds the expected prime numbers | | | | |
| **Class** | **Method** | **Stage** | **Input** | **Outcome** |
| GeneratorTest | sieveOfSundaramTest | setupScenario2 | n = 18 | Prime number are: 2,3,5,7,11,13,17 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Objective:** Verify that the sieveOfSundaramTest2 method finds the expected prime numbers | | | | |
| **Class** | **Method** | **Stage** | **Input** | **Outcome** |
| GeneratorTest | sieveOfSundaramTest2 | setupScenario4 | n = 45 | Prime number are: 2,3,5,7,11,13,17,19,23,29,31,37,41,43 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Objective:** Verify that the sieveOfSundaramTest3 method finds the expected prime numbers | | | | |
| **Class** | **Method** | **Stage** | **Input** | **Outcome** |
| GeneratorTest | sieveOfSundaramTest3 | setupScenario1 | n = 9 | Prime number are: 2,3,5,7 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Objective:** Verify that the sieveOfEratosthenesTest method finds the expected prime numbers | | | | |
| **Class** | **Method** | **Stage** | **Input** | **Outcome** |
| GeneratorTest | sieveOfEratosthenesTest | setupScenario2 | n = 18 | Prime number are: 2,3,5,7,11,13,17 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Objective:** Verify that the sieveOfEratosthenesTest2 method finds the expected prime numbers | | | | |
| **Class** | **Method** | **Stage** | **Input** | **Outcome** |
| GeneratorTest | sieveOfEratosthenesTest2 | setupScenario4 | n = 45 | Prime number are: 2,3,5,7,11,13,17,19,23,29,31,37,41,43 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Objective:** Verify that the sieveOfEratosthenesTest3 method finds the expected prime numbers | | | | |
| **Class** | **Method** | **Stage** | **Input** | **Outcome** |
| GeneratorTest | sieveOfEratosthenesTest3 | setupScenario1 | n = 9 | Prime number are: 2,3,5,7 |