**Operational Specification Template**

|  |  |  |  |
| --- | --- | --- | --- |
| **Student** | Cristian Berrio & David Acevedo | **Date** | 28/09/2017 |
| **Program** | Calcular rangos relativos | **Program #** | 4 |
| **Instructor** | Juan Carlos Marín | **Language** | JavaScript |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Scenario Number** | 1 | **User Objective** | Ingresar archivo de texto | |
| **Scenario Objective** | | Ingresar un archivo de texto por medio del botón cargar archivo | | |
| **Source** | **Step** | **Action** | | **Comments** |
| Sistema | 1 | Desplegar la vista | | La vista principal es ~/programa4 |
| Usuario | 2 | Presionar el botón de cargar archivo | |  |
| Sistema | 3 | Desplegar la vista de directorios de archivo locales | |  |
| Usuario | 4 | Seleccionar un archivo de texto | |  |
| Usuario | 5 | Presionar el botón aceptar para cargar el archivo | |  |
| Sistema | 6 | Reconocer el archivo de texto | |  |
| Sistema | 7 | Desplegar el contenido del archivo en un área de la vista | |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Student** | Cristian Berrio & David Acevedo | **Date** | 28/09/2017 |
| **Program** | Calcular rangos relativos | **Program #** | 4 |
| **Instructor** | Juan Carlos Marín | **Language** | JavaScript |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Scenario Number** | 2 | **User Objective** | Calcular rangos exponenciales | |
| **Scenario Objective** | | Calcular los rangos exponenciales de los conjuntos de datos ingresados | | |
| **Source** | **Step** | **Action** | | **Comments** |
| Sistema | 1 | Habilitar el botón de calcular LOC | |  |
| Usuario | 2 | Presionar el botón de calcular LOC | |  |
| Sistema | 3 | Desplegar el resultado de las líneas de código por métodos. | |  |
| Sistema | 4 | Deshabilitar el botón calcular LOC | |  |
| Sistema | 5 | Habilitar el botón Calcular Logaritmos | |  |
| Usuario | 6 | Presionar el botón Calcular Logaritmos | |  |
| Sistema | 7 | Desplegar el resultado de los logaritmos naturales del LOC encontrados anteriormente. | |  |
| Sistema | 8 | Deshabilitar el botón Calcular Logaritmos | |  |
| Sistema | 9 | Habilitar el botón de Calcular promedio de logaritmos | |  |
| Usuario | 10 | Presionar el botón Calcular promedio de logaritmos | |  |
| Sistema | 11 | Desplegar el promedio de los logaritmos | |  |
| Sistema | |2 | Deshabilitar el botón Calcular promedio de logaritmos | |  |
| Sistema | 13 | Habilitar el botón calcular varianza | |  |
| Usuario | 14 | Presionar el botón calcular varianza | |  |
| Sistema | 15 | Desplegar la varianza | |  |
| Sistema | 16 | Deshabilitar el botón calcular varianza | |  |
| Sistema | 17 | Habilitar el botón calcular desviación estándar | |  |
| Usuario | 18 | Presionar el botón calcular desviación estándar | |  |
| Sistema | 19 | Desplegar la desviación estándar | |  |
| Sistema | 20 | Deshabilitar el botón calcular desviación estándar | |  |
| Sistema | 21 | Habilitar el botón calcular rangos logarítmicos | |  |
| Usuario | 22 | Presionar el botón calcular rangos logarítmicos | |  |
| Sistema | 23 | Desplegar los rangos logarítmicos | |  |
| Sistema | 24 | Deshabilitar el botón calcular rangos logarítmicos | |  |
| Sistema | 25 | Habilitar el botón calcular rangos exponenciales | |  |
| Usuario | 26 | Presionar el botón calcular rangos exponenciales | |  |
| Sistema | 27 | Desplegar los rangos exponenciales | |  |
| Sistema | 28 | Deshabilitar el botón calcular rangos exponenciales | |  |

**Operational Specification Template Instructions**

|  |  |
| --- | --- |
| **Purpose** | * To hold descriptions of the likely operational scenarios followed during program use * To ensure that all significant usage issues are considered during program design * To specify test scenarios |
| **General** | * Use this template for complete programs, subsystems, or systems. * Group multiple small scenarios on a single template, as long as they are clearly distinguished and have related objectives. * List the major scenarios and reference other exception, error, or special cases under comments. * Use this template to document the operational specifications during planning, design, test development, implementation, and test. * After implementation and testing, update the template to reflect the actual implemented product. |
| **Header** | * Enter your name and the date. * Enter the program name and number. * Enter the instructor’s name and the programming language you are using. |
| **Scenario Number** | Where several scenarios are involved, reference numbers are needed. |
| **User Objective** | List the users’ likely purpose for the scenario, for example, to log onto the system or to handle an error condition. |
| **Scenario Objective** | List the designer’s purpose for the scenario, for example, to define common user errors or to detail a test scenario. |
| **Source** | * Enter the source of the scenario action. * Example sources could be user, program, and system. |
| **Step** | Provide sequence numbers for the scenario steps. These facilitate reviews and inspections. |
| **Action** | Describe the action taken, such as   * Enter incorrect mode selection. * Provide error message. |
| **Comments** | List significant information relating to the action, such as   * User enters an incorrect value. * An error is possible with this action. |

**Functional Specification Template**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Student** | | | Cristian Berrio & David Acevedo | | **Date** | 28/09/2017 |
| **Program** | | | Calcular rangos relativos | | **Program #** | 4 |
| **Instructor** | | | Juan Carlos Marín | | **Language** | JavaScript |
|  | | | | | | |
| **Class Name** | | programa4.js | | | | |
|  | | | | | | |
| **Attributes** | | | | | | |
|  | **Declaration** | | | **Description** | | |
|  | text | | | Variable que contiene el texto que se encuentra dentro del archivo ingresado por el usuario | | |
|  | division | | | Variable que contiene el resultado de la división de las líneas de código por métodos. | | |
|  | logaritmos | | | Variable que contiene el resultado de los logaritmos naturales de los resultados de las divisiones antes halladas. | | |
|  | promedio | | | Variable que contiene el promedio de los logaritmos | | |
|  | varianza | | | Variable que contiene la varianza de los logaritmos | | |
|  | desviación | | | Variable que contiene la desviación de los logaritmos | | |
|  | logVeryShort | | | Variable que contiene el logaritmo del rango VS | | |
|  | logShort | | | Variable que contiene el logaritmo del rango S | | |
|  | logMedium | | | Variable que contiene el logaritmo del rango M | | |
|  | logLong | | | Variable que contiene el logaritmo del rango L | | |
|  | logVeryLong | | | Variable que contiene el logaritmo del rango VL | | |
|  | veryShort | | | Variable que contiene el rango VS | | |
|  | short | | | Variable que contiene el rango S | | |
|  | medium | | | Variable que contiene el rango M | | |
|  | long | | | Variable que contiene el rango L | | |
|  | veryLong | | | Variable que contiene el rango VL | | |
|  | | | | | | |
| **Items** | | | | | | |
|  | **Declaration** | | | **Description** | | |
|  | metodo\_leerArchivo | | | Este método interpreta el evento del botón designado para cargar el archivo y obtiene el archivo seleccionado por el usuario y lo convierte en una variable. | | |
|  | metodo\_dividirLOCMetodos | | | Este método divide la fila de LOC de las clases entre el número de métodos que hay, lo guarda en una vector y lo retorna. | | |
|  | metodo\_calcularLog | | | Este método calcula el logaritmo natural de los resultados de las divisiones hechas que se hallaron anteriormente. | | |
|  | metodo\_calcularProm | | | Este método calcula el promedio de los logaritmos hallados, para poder calcular la varianza. | | |
|  | metodo\_calcularVarianza | | | Este método calcula la varianza de los logaritmos naturales, para ello necesita el promedio hallado y los logaritmos hallados anteriormente. | | |
|  | metodo\_calcularDesviacion | | | Este método calcula la desviación estándar teniendo en cuenta la varianza hallada anteriormente. | | |
|  | metodo\_calcularRangos | | | Este método calcula los rangos logarítmicos, mediante la desviación y el promedio que se halló. | | |
|  | metodo\_calcularExpRangos | | | Este método calcula la función euler de los rangos que se hallaron anteriormente, esto para hallar los rangos vs, s, m, l, vl. | | |

**Functional Specification Template Instructions**

|  |  |
| --- | --- |
| **Purpose** | * To hold a part’s functional specifications * To describe classes, program modules, or entire programs |
| **General** | * Use this template for complete programs, subsystems, or systems. * Use this template to document the functional specifications during planning, design, test development, implementation, and test. * After implementation and testing, update the template to reflect the actual implemented product. |
| **Header** | * Enter your name and the date. * Enter the program name and number. * Enter the instructor’s name and the programming language you are using. |
| **Class Name** | * Enter the part or class name and the classes from which it directly inherits. * List the class names starting with the most immediate. * Where practical, list the full inheritance hierarchy. |
| **Attributes** | * Provide the declaration and description for each global or externally visible variable or parameter with any constraints. * List pertinent relationships of this part with other parts together with the multiplicity and constraints. |
| **Items** | * Provide the declaration and description for each item. * Precisely describe the conditions that govern each item’s return values. * Describe any initialization or other key item responsibilities. |
| **Example Items** | An item could be a class method, procedure, function, or database query, for example. |

**State Specification Template**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Student | | Cristian Berrio & David Acevedo | | | Date | | 28/09/2017 |
| Program | | Calcular rangos relativos | | | Program # | | 4 |
| Instructor | | Juan Carlos Marín | | | Language | | JavaScript |
|  | | |  | | | | |
| **State Name** | | | | **Description** | | | |
|  | | | |  | | | |
| texto\_sin\_enter | | | | Al texto que se ingreso, se reemplaza por uno que no tenga espacios de enter. | | | |
| texto\_filas | | | | El texto se divide en filas mediante un split. | | | |
| texto\_numeros | | | | cada fila del texto se divide en números por posición de cada fila. | | | |
|  | | | |  | | | |
|  | | | |  | | | |
|  | | | |  | | | |
|  | | | |  | | | |
|  | | | |  | | | |
|  | | | |  | | | |
|  | | | |  | | | |
|  | | | |  | | | |
|  | | | |  | | | |
|  | | | |  | | | |
|  | | | |  | | | |
| **Function/Parameter** | | | | **Description** | | | |
|  | | | |  | | | |
| this.set() | | | | Establece la variable que se nombró con el contenido que se está enviando. | | | |
| this.get() | | | | Obtiene el contenido de la variable que se pidió. | | | |
| Math.log() | | | | Esta función retorna el logaritmo natural del número deseado | | | |
| Math.pow() | | | | Obtiene la potencia del número que se mande. | | | |
| desvi = Math.sqrt(vari) | | | | Fórmula aplicada de la varianza | | | |
| Math.round() | | | | Permite redondear el numéro que se coloque | | | |
| Math.exp() | | | | Permite encontrar la función euler del número colocado | | | |
|  | | | |  | | | |
|  | | | |  | | | |
| **States/Next States** | | | | **Transition Condition** | | **Action** | |
| Habilitar botones | | | | Habilitar botón de cada paso para hallar los rangos | | button.disabled o button.hidden | |
| texto\_sin\_enter | | | | Se seleccionó el texto correctamente | | texto.replace() | |
| texto\_filas | | | | El texto seleccionado se proceso y ya no tiene enter | | texto.split() | |
| texto\_numeros | | | | El texto se encuentra dividido por filas | | texto.split() | |
|  | | | |  | |  | |
|  | | | |  | |  | |
|  | | | |  | |  | |
|  | | | |  | |  | |
|  | | | |  | |  | |
|  | | | |  | |  | |
|  | | | |  | |  | |
|  | | | |  | |  | |
|  | | | |  | |  | |

|  |  |  |
| --- | --- | --- |
| **State Specification Template Instructions** | | |
| **Purpose** | * To hold the state and state transition specifications for a system, class, or program * To support state-machine analysis during design, design reviews, and design inspections | |
| **General** | * This form shows each system, program, or routine state, the attributes of that state, and the transition conditions among the states. * Use this template to document the state specifications during planning, design, test development, implementation, and test. * After implementation and testing, update the template to reflect the actual implemented product. | |
| **Header** | * Enter your name and the date. * Enter the program name and number. * Enter the instructor’s name and the programming language you are using. | |
| **State Name** | * Name all of the program’s states. * Also enter each state name in the header space at the top of each “States/Next States” section of the template. | |
| **State Name Description** | * Describe each state and any parameter values that characterize it. * For example, if a state is described by SetSize=10 and SetPosition=3, list SetSize=10 and SetPosition=3. | |
| **Function/Parameter** | * List the principal functions and parameters. * Include all key variables or methods used to define state transitions or actions. | |
| **Function/Parameter Description** | * For each function, provide its declaration, parameters, and returns. * For each parameter, define its type and significant values. | |
| **Next State** | * For each state, list the names of all possible next states. * Include the state itself. | |
| **Transition Condition** | List the conditions for transition to each next state.   * Use a mathematical or otherwise precise notation. * If the transition is impossible, list "impossible," with a note saying why. | |
| **Action** | List the actions taken with each state transition. | |

**Logic Specification Template**

|  |  |  |  |
| --- | --- | --- | --- |
| Student | Cristian Berrio & David Acevedo | Date | 28/09/2017 |
| Program | Calcular rangos relativos | Program # | 4 |
| Instructor | Juan Carlos Marín | Language | JavaScript |

|  |  |
| --- | --- |
| **Design** | * Especificación funcional |
| **References** | * Especificación operacional |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Parameters** | **cadena** text, **cadena** textFilas, **entero** division**, entero** logaritmos**, entero** promedio**, entero** varianza**, entero** desviacion**, entero** logVeryShort**, entero** logShort**, entero** logMedium**, entero** logLong**, entero** logVeryLong**, entero** veryShort**, entero** short, **entero** medium, **entero** long, **entero** veryLong |
|  |  |
|  |  |
|  |  |
|  |  |

metodo\_leerArchivo{

|  |
| --- |
| recogerEventoDelboton() |
| text = textoDelArchivo |
| } |
|  |
| metodo\_devidirLOCMetodos { |
| **cadena** textoInterpretado |
| texto = interpretarTexto(text) |
| **para** i **hasta** finDeDatos |
| div = texto[1]/texto[2] |
| **finPara** |
| } |
|  |
| metodo\_calcularLog {  **para** i **hasta** finDeDatos |
| log = Math.log(div) |
| **finPara** |
| } |
|  |
| metodo\_calcularProm { |
| **para** i **hasta** finDeDatos |
| suma = suma + log[i] |
| **finPara** |
| prom = suma / size |
| } |
|  |
| metodo\_calcularVarianza { |
| denom = log.length - 1 |
| **para** i **hasta** finDeDatos |
| num = log[i] - prom |
| num = Math.pow(num,2) |
| sum = suma + num |
| **finPara** |
| vari = suma /denom |
| metodo\_calcularDesviacion { |
| desvi = Math.sqrt(vari) |
| } |
|  |
| metodo\_calcularRangos { |
| dif = 2\*desvi |
| logvs = prom -dif |
| logs = prom - desvi |
| logm = prom |
| logl = prom + desvi |
| logvl = prom + dif |
| } |
|  |
| metodo\_calcularExpRangos { |
| vs = Math.exp(logvs) |
| s = Math.exp(logs) |
| m = Math..exp(logm) |
| l = Math.exp(logl) |
| vl = Math.exp(logvl) |
| } |
|  |

**Logic Specification Template Instructions**

|  |  |
| --- | --- |
| **Purpose** | * To contain the pseudocode for a program, component, or system * To enable precise and complete program implementation * To facilitate thorough design and implementation reviews and inspections |
| **General** | * Use this template to document the program’s detailed logic. * After implementation and testing, update the template to reflect the actual implemented product. * During detailed design, write the pseudocode needed to describe all of the program’s logic. * Use plain language and avoid using programming instructions wherever practical. |
| **Header** | * Enter your name and the date. * Enter the program name and number. * Enter the instructor’s name and the programming language you are using. |
| **Design References** | List the references used to produce the program’s logical design.   * the Operational, Functional, and State templates * the program’s requirements * any other pertinent source |
| **Parameters** | * Where needed, define any parameters or abbreviations used. * Avoid duplicating definitions on other templates and reference these other definitions where they are needed. |