**COP16:**

**Short description:** The Major needsan app that lets the knowledge and the navigation of the places with biological diversity and the communities that take care of those places.

**Client:** The major

**User:** Ministerio de Ambiente y Desarrollo Sostenible

-Members of the government that are going to participate in the COP16 meeting.

**Inputs and outputs:**

|  |  |
| --- | --- |
| **Inputs** | **Outputs** |
| Community   * Community’s name. * Community’s type: afrocolombian, native, raizal. * Community representative’s name and phone. * Habitants. * Biggest problems   **Product**   * Product’s name * Percentage of natural products employed in their elaboration. * Type: food or handcrafts. * Handmade: yes/no   Place   * Place’s name * Departments name: Chocó, Valle, Cauca, Nariño). * Quantity of square kilometers. * Type: protected area, national park, private area. * Inauguration date. * Place’s picture. * Community that takes care of it. * Money available for its sustainability. * Species (50 max)   -Name  -Fauna or flora  -Picture  -Quantity at the place.   * Test cases. | * Place information (picture,attributes) * Department’s community information. * Communities with problems about hospitals or schools. * Name with the highest quantity of species. * 3 biggest places in kmˆ2 |

**Requeriments:**

|  |  |
| --- | --- |
| **Contexto del problema** | El alcalde necesita un software para el evento COP16, en donde se pueda ingresar o consultar una comunidad (nombre, tipo, nombre y celular del representante, cantidad de habitantes, mayores problemas e inventario de productos), ingresar o consultar un lugar (nombre, departamento, cantidad de kilómetros cuadrados, tipo, fecha de inauguración, foto, comunidad, recursos económicos, especies e información de las especies: nombre, flora/fauna, foto, y cantidad de ejemplares), ingresar un producto(nombre, porcentaje de productos naturales empleados, tipo y si es hecho a mano).  El software debe tener dos menús uno administrativo y uno de consultas. En el administrativo se debe tener opciones para ingresar una comunidad, ingresar un lugar, ingresar/eliminar un producto, ingresar la especie a un lugar y crear los casos de prueba.  En el menú de consultas debe desplegarle al usuario determinadas opciones como: poder buscar la información de un lugar, la información de las comunidades de un departamento, la información de las comunidades con problemas con los hospitales/escuelas, el nombre del lugar con mayor cantidad de especies y lugares con mayor número de kilómetros cuadrados. |
| **Requerimientos funcionales** | RF1 – Registrar comunidad.  RF2 – Registrar lugar.  RF3 – Registrar un producto.  RF4 - Registrar una especie.  RF5 – Modificar los datos de una especie.  RF6 – Consultar las comunidades en los departamentos.  RF7 – Consultar una comunidad que tenga problemas de hospitales/escuelas.  RF8 – Consultar el lugar con el mayor número de especies.  RF9 – Consultar los 3 lugares más grandes por kilómetro cuadro.  RF10 – Consultar un lugar  RF11 – Eliminar un producto. |
| **Requerimientos no funcionales** | RNF 1 – Garantizar un sistema intuitivo y fácil de usar, de mano de los diseñadores de diseño UX.  RNF 2 – Garantizar la fácil actualización y mantenimiento a futuro.  RNF 3 – Garantizar un buen desempeño del software mediante buenas prácticas de desarrollo. |

**RF1- Registrar comunidad.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identificador y nombre | *[RF1- Registrar comunidad.]* | | | |
| Resumen | Para el registro e ingreso de una comunidad, se debe tener en cuenta su nombre, a qué tipo pertenece (Afrocolombiana, Indígena o Raizal), cantidad de habitantes, el nombre y el celular de su representante, cuál es su mayor problema (falta de hospital, falta de escuela, falta de agua potable o falta de alimentación básica) y el inventario de los productos que la comunidad ofrece (máximo 20) que se asociarán a ella una vez se ingresen. Una vez se registra la comunidad, se le mostrará un mensaje al usuario que le informará que le registro fue exitoso y en caso de haber algún error en el diligenciamiento, se le informará mediante un mensaje alerta y se le redirigirá a llenar nuevamente los campos. | | | |
| Entradas | **Nombre entrada** | **Tipo de dato** | | **Condición valores válidos** |
| nameComunnity | String | | No agregar números ni signos de puntuación. |
| type | TypeCommunity | | Se desplegará una lista donde el usuario elegirá de la lista el tipo de comunidad a la que pertenece la que estamos ingresando. |
| nameRepresentative | String | | No agregar números ni signos de puntuación. |
| phoneRepresentative | String | | Los caracteres ingresados no deben ser números decimales, deben ser números enteros. |
| quantityHabitants | int | | Números enteros mayores o iguales de cero. Al introducir los datos se debe hacer sin puntos. |
| problemsCommunity | Problem | | Se desplegará una lista donde el usuario elegirá el mayor problema que tiene la comunidad que está ingresando (falta de hospital, falta de escuela, falta de agua potable o falta de alimentación básica). |
| Resultado o Postcondición | Una vez el usuario termina de llenar los campos, se registra la comunidad, y se le mostrará un mensaje al usuario que le informará que el registro fue exitoso y en caso de haber algún error en el diligenciamiento, se le informará mediante un mensaje alerta y se le redirigirá a llenar nuevamente los campos. | | | |
| Salidas | **Nombre salida** | | **Tipo de dato** | **Formato** |
| successMessaje | | String | “El registro de la comunidad” + nameCommunity + “ha sido exitoso.” |
| errorMessaje2 | | String | “Hubo un error a la hora de registrar la comunidad, por favor inténtelo de nuevo” |

**RF2 – Registrar un lugar.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identificador y nombre | *[RF2 – Registrar un lugar.]* | | | |
| Resumen | Para el registro e ingreso de un lugar, se debe tener en cuenta su nombre, el nombre del departamento al que pertenece, la cantidad de kilómetros cuadrados que cubre el terreno, el tipo de lugar al que pertenece (área protegida, parque nacional o área privada), la fecha de inauguración, la foto del lugar, los recursos económicos necesarios para un buen cuidado del hábitat y el nombre de la comunidad que la cuida Una vez se registra el lugar se le mostrará un mensaje al usuario que le informará que le registro fue exitoso y en caso de haber algún error en el diligenciamiento, se le informará mediante un mensaje alerta y se le redirigirá a llenar nuevamente los campos. | | | |
| Entradas | **Nombre entrada** | | **Tipo de dato** | **Condición valores válidos** |
| namePlace | | String | No agregar números ni signos de puntuación. |
| nameDepartment | | Department | Se desplegará una lista donde el usuario elegirá el departamento en el que está ubicado el lugar que está registrando. (Chocó, Valle, Nariño o Cauca) |
| quantitySquareKilometers | | double | Números mayores o iguales de cero. |
| typePlace | | TypePlace | Se desplegará una lista donde el usuario elegirá el tipo de lugar a que corresponde el lugar que está registrando (área protegida, parque nacional o área privada). |
| inagurationDate | | Date | No agregar letras ni signos de puntuación. |
| picture | | String | No tiene ningún formato en específico. |
| economicResources | | double | Números mayores o iguales de cero. Al introducir los datos se debe hacer sin puntos. |
| communityTakingCare | | int | Se desplegará una lista enumerada que le mostrará al usuario las comunidades ingresadas en el sistema, donde el usuario elegirá a la que pertenece el lugar que está registrando. |
| Resultado o Postcondición | Una vez el usuario termina de llenar los campos, se registra el lugar, y se le mostrará un mensaje al usuario que le informará que el registro fue exitoso y en caso de haber algún error en el diligenciamiento, se le informará mediante un mensaje alerta y se le redirigirá a llenar nuevamente los campos. | | | |
| Salidas | **Nombre salida** | **Tipo de dato** | | **Formato** |
| successMessaje | String | | “El lugar” + namePlace + “fue registrado exitosamente” |
| errorMessaje2 | String | | “Hubo un error a la hora de registrar el lugar, por favor inténtalo de nuevo” |

**RF3 – Registrar un producto.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identificador y nombre | *[RF3 – Registrar un producto.]* | | | |
| Resumen | Para el registro e ingreso de un producto, se debe tener en cuenta su nombre, el porcentaje de productos naturales empleados en su fabricación, el tipo de producto (alimento o artesanía) y si es hecho a mano o no. Una vez se registra el producto, se le mostrará un mensaje al usuario que le informará que le registro fue exitoso y en caso de haber algún error en el diligenciamiento, se le informará mediante un mensaje alerta y se le redirigirá a llenar nuevamente los campos. Es importante resaltar que cada comunidad tiene un máximo de 20 productos. | | | |
| Entradas | **Nombre entrada** | **Tipo de dato** | | **Condición valores válidos** |
| nameProduct | String | | No agregar números ni signos de puntuación. |
| porcentageNaturalProducts | double | | Números mayores o iguales de cero. Al introducir los datos se debe hacer sin puntos. |
| typeProduct | TypeProduct | | Se desplegará una lista donde el usuario elegirá el tipo de producto a que corresponde el que está registrando (FOOD o CRAFT). |
| handmade | Handmade | | Se desplegará una lista donde el usuario elegirá el si el producto que está ingresando es hecho a mano o no. (YES or NO) |
| nameCommunity | int | | Se desplegará una lista enumerada que le mostrará al usuario las comunidades ingresadas en el sistema, donde el usuario elegirá a la que pertenece el producto que está registrando. |
| Resultado o Postcondición | Una vez el usuario termina de llenar los campos, se registra el producto, se asocia a su comunidad correspondiente y se le mostrará un mensaje al usuario que le informará que el registro fue exitoso y en caso de haber algún error en el diligenciamiento, se le informará mediante un mensaje alerta y se le redirigirá a llenar nuevamente los campos. | | | |
| Salidas | **Nombre salida** | | **Tipo de dato** | **Formato** |
| successMessaje | | String | “El registro del producto” + nameProduct + “se hizo de manera exitosa” |
| errorMessaje | | String | No tiene ningún formato en específico. |

**RF4 – Registrar una especie.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identificador y nombre | *[RF5 – Ingresar una especie.]* | | | |
| Resumen | Para el registro e ingreso de una especie, se debe tener en cuenta su nombre, el tipo (flora o fauna), una foto, el lugar al que pertenece y la cantidad que habita en dicho lugar. Una vez se registra la especie, se le mostrará un mensaje al usuario que le informará que le registro fue exitoso y en caso de haber algún error en el diligenciamiento, se le informará mediante un mensaje alerta y se le redirigirá a llenar nuevamente los campos. Es importante resaltar que son máximo 50 especies por lugar. | | | |
| Entradas | **Nombre entrada** | **Tipo de dato** | | **Condición valores válidos** |
| nameSpecie | String | | No agregar números ni signos de puntuación. |
| typeSpecie | TypeSpecie | | Se desplegará una lista enumerada donde el usuario elegirá el tipo de especie a que corresponde la que está registrando (flora o fauna). |
| pictureSpecie | String | | URL |
| quantitySpecie | int | | Números enteros mayores o iguales de cero. Al introducir los datos se debe hacer sin puntos. |
| namePlace | int | | Se desplegará una lista enumerada que le mostrará al usuario los lugares ingresados en el sistema, donde el usuario elegirá a la que pertenece la especie que está registrando. |
| Resultado o Postcondición | Una vez el usuario selecciona su opción, el sistema busca el producto que coincida con la opción que eligió y procede a eliminarlo. Una vez esto, se le mostrará un mensaje al usuario que le informará que la eliminación fue exitosa y en caso de haber algún error en el diligenciamiento, se le informará mediante un mensaje alerta y se le redirigirá a elegir una opción nuevamente. | | | |
| Salidas | **Nombre salida** | | **Tipo de dato** | **Formato** |
| successMessaje | | String | “La especie ” + nameSpecie + “ha sido registrada exitosamente.” |
| errorMessaje | | String | “Hubo un error a la hora de registrar la especie, por favor vuélvelo e intentar.” |

**RF5 – Modificar los datos de una especie.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identificador y nombre | *[RF6 – Modificar los datos de una especie.]* | | | |
| Resumen | Para la modificación de los datos de una especie, se debe tener en cuenta el dato que desea modificar el usuario, que lo elegirá a partir de un menú de opciones, donde posteriormente ingresará el nuevo dato. Una vez se confirma el cambio del dato de la especie, se le mostrará un mensaje al usuario que le informará que la actualización de datos fue exitosa y en caso de haber algún error en el diligenciamiento, se le informará mediante un mensaje alerta y se le redirigirá a diligenciar nuevamente el campo (en caso de ingresar un valor inválido o el mismo valor que tiene en ese momento el atributo). | | | |
| daEntradas | **Nombre entrada** | **Tipo de dato** | | **Condición valores válidos** |
| namePlace | int | | Se desplegará una lista enumerada en donde se mostrará al usuario los lugares ingresados en el sistema, donde el usuario elegirá al que pertenece la especie que desea modificar. |
| nameSpecie | int | | Se desplegará una lista enumerada que mostrará los nombres de las especies ingresadas en el sistema, lo que permitirá al usuario elegir la que desea modificar. |
| specieData | int | | Se desplegará una lista enumerada la cual permita escoger cual de los datos se desea modificar. (nombre, tipo, imagen, cantidad o lugar al que pertenece) |
| newData | String / int (según corresponda) | | Si newData: String->  No tiene ningún formato en específico.  Si newData: int ->  Números enteros mayores o iguales de cero. Al introducir los datos se debe hacer sin puntos. |
| Resultado o Postcondición | Una vez el usuario selecciona su opción, el sistema busca el dato que coincida con la opción que eligió, procede a solicitar por el valor que desea actualizarlo y lo reemplaza. Una vez esto, se le mostrará un mensaje al usuario que le informará que la actualización fue exitosa y en caso de haber algún error en el diligenciamiento, se le informará mediante un mensaje alerta y se le redirigirá a diligenciar nuevamente el campo. | | | |
| Salidas | **Nombre salida** | | **Tipo de dato** | **Formato** |
| successMessaje | | String | “Los datos de la especie ” + nameSpecie + “han sido actualizados exitosamente” |
| errorMessaje | | String | “Hubo un error a la hora de actualizar los datos, por favor vuélvalo a intentar.” |

**RF6 – Consultar las comunidades en los departamentos.**

|  |  |  |  |
| --- | --- | --- | --- |
| Identificador y nombre | *[RF7 – Consultar las comunidades en los departamentos]* | | |
| Resumen | *(debe incluir las entradas, actividades y condiciones necesarias para transformar las entradas en salidas, las salidas y la postcondición.)*  El sistema debe mostrar una lista desplegable con los departamentos (Chocó, Valle, Cauca y Nariño), en donde después de escoger el departamento, se van a mostrar las comunidades almacenadas en el sistema en el respectivo lugar. En caso de que no se encuentre ninguna, se le informará al usuario. | | |
| Entradas | **Nombre entrada** | **Tipo de dato** | **Condición valores válidos** |
| department | String | Se ingresa a partir de una lista desplegable, por ende, debe ser un String perteneciente a esta: “Chocó”, “Cauca”, “Valle” ó “Nariño” |
| Resultado o Postcondición | Después de escoger el dato en la lista desplegable, se mostrará la(s) comunidad(es) que se encuentran en el departamento escogido. En caso de que no se encuentre ninguna, se le informará al usuario la ausencia de comunidades en dicho departamento. | | |
| Salidas | **Nombre salida** | **Tipo de dato** | **Formato** |
| mesage1 | String | “En el departamento escogido no se ha registrado ninguna comunidad” |

**RF7 – Consultar una comunidad que tenga problemas de hospitales/escuelas.**

|  |  |  |  |
| --- | --- | --- | --- |
| Identificador y nombre | *[RF8 – Consultar una comunidad que tenga problemas de hospitales/escuelas.]* | | |
| Resumen | *(debe incluir las entradas, actividades y condiciones necesarias para transformar las entradas en salidas, las salidas y la postcondición.)*  El sistema debe mostrar las comunidades las cuales tengan problemas de hospitales o escuelas cuando el usuario desee consultar al software sobre dicho aspecto. Para el sistema usará un filtro el cual tomé las comunidades que cumplan con la condición del tipo de problema y se los muestra al usuario. | | |
| Entradas | **Nombre entrada** | **Tipo de dato** | **Condición valores válidos** |
| N/A | N/A | N/A |
| Resultado o Postcondición | Después de que el usuario le pide al software consultar una comunidad, después de comparar los problemas de las comunidades, se traerá las que tienen problemas con hospitales o escuelas y se le mostrará cuales al usuario. En caso de que no hayan comunidades con tales características se mostrará un mensaje el cual le diga al usuario que: “No se han registrado comunidades con problemas relacionados a hospitales o escuelas” | | |
| Salidas | **Nombre salida** | **Tipo de dato** | **Formato** |
| message1 | String | “En el departamento escogido no se ha registrado ninguna comunidad” |

**RF8 – Consultar el lugar con el mayor número de especies.**

|  |  |  |  |
| --- | --- | --- | --- |
| Identificador y nombre | *[RF8 – Consultar el lugar con el mayor número de especies.]* | | |
| Resumen | *(debe incluir las entradas, actividades y condiciones necesarias para transformar las entradas en salidas, las salidas y la postcondición.)*  El sistema debe permitir consultar el lugar con el mayor número de especies, teniendo en cuenta el número de especies presentes en cada lugar, se va a comparar para sacar el lugar con el mayor número, para proceder a mostrarle al usuario la información recolectada. En caso de que haya dos lugares con la misma cantidad de especies se mostrarán ambos. | | |
| Entradas | **Nombre entrada** | **Tipo de dato** | **Condición valores válidos** |
| N/A | N/A | N/A |
| Resultado o Postcondición | En el momento en el que se necesite consultar el lugar con la mayor cantidad de especies, el sistema comparará los lugares registrados para obtener esta información. En caso de que dos lugares o más tengan el mayor número, se mostrarán al usuario. | | |
| Salidas | **Nombre salida** | **Tipo de dato** | **Formato** |
| message1 | String | “El lugar con el mayor número de especies es: ” |

**RF9 – Calcular/Consultar los 3 lugares más grandes por kilómetro cuadro.**

|  |  |  |  |
| --- | --- | --- | --- |
| Identificador y nombre | *[*RF9 – Calcular/Consultar los 3 lugares más grandes por kilómetro cuadro.*]* | | |
| Resumen | *(debe incluir las entradas, actividades y condiciones necesarias para transformar las entradas en salidas, las salidas y la postcondición.)*  El sistema debe mostrar los 3 lugares más grandes por kilómetro cuadrado, por tanto, cuando el usuario pida la información a través de un botón, se compararán los datos de los tamaños de los lugares, dando como salida los nombres de estos. | | |
| Entradas | **Nombre entrada** | **Tipo de dato** | **Condición valores válidos** |
| N/A | N/A | N/A |
| Resultado o Postcondición | En el momento en el que el usuario le pida los 3 lugares más grandes por kilometro cuadrado, el sistema comparará el tamaño de los lugares, brindando los 3 con mayor número de kilómetros. | | |
| Salidas | **Nombre salida** | **Tipo de dato** | **Formato** |
| message1 | String | “Los 3 lugares con mayor cantidad de kilómetros cuadrados son:  1...  2…  3…” |

**RF10 – Consultar un lugar.**

|  |  |  |  |
| --- | --- | --- | --- |
| Identificador y nombre | *[RF10 – Consultar un lugar.]* | | |
| Resumen | *(debe incluir las entradas, actividades y condiciones necesarias para transformar las entradas en salidas, las salidas y la postcondición.)*  El sistema debe mostrar una lista desplegable en donde muestre al usuario los lugares registrados hasta el momento en orden alfabético, en donde se podrá escoger cual de estos desea consultar. Después de haber hecho la elección, se mostrará el lugar con todos sus atributos y foto. | | |
| Entradas | **Nombre entrada** | **Tipo de dato** | **Condición valores válidos** |
| namePlace | int | Se desplegará una lista enumerada que le mostrará al usuario los lugares en el sistema, permitiéndole escoger cual desea consultar. |
| Resultado o Postcondición | Después de escoger el dato en la lista desplegable, se mostrará toda la información del lugar: atributos y fotos. En caso de que no haya ningún lugar registrado de momento se mostrará al usuario un mensaje que le informe sobre la falta de lugares. | | |
| Salidas | **Nombre salida** | **Tipo de dato** | **Formato** |
| placeName | String | No debe tener números ni caracteres especiales. |
| department | String | No debe tener números ni caracteres especiales. |
| quantitySquareKilometers | Double | Valor > 0 |
| typePlace | String | “Area protegida” ó “Parque Nacional” ó “área privada” |
| inagurationDate | Date | Fecha menor y diferente a la actual |
| picture | String | URL |
| communityTakingCare | String | No debe tener números ni caracteres especiales. |
| economicResources | Double | Valor > 0 |
|  | Species | String | -“Nombre”  -“Flora/fauna”,  -“Foto”  - “Cantidad ejemplares” |

**RF11 – Eliminar un producto.**

|  |  |  |  |
| --- | --- | --- | --- |
| Identificador y nombre | *[RF11 – Eliminar un producto.]* | | |
| Resumen | *(debe incluir las entradas, actividades y condiciones necesarias para transformar las entradas en salidas, las salidas y la postcondición.)*  A través de una lista desplegable se mostrarán las comunidades que se han registrado en orden alfabético, después de escoger una de las comunidades, se mostraran en otra lista desplegable los productos que maneja la comunidad escogida y haciendo uso de un botón que diga “Eliminar” se borrará los datos sobre dicho producto. | | |
| Entradas | **Nombre entrada** | **Tipo de dato** | **Condición valores válidos** |
| namePlace | int | Se desplegará una lista enumerada que le mostrará al usuario los lugares ingresados en el sistema, lo que permitirá al usuario elegir el lugar donde pertenece el producto que va a eliminar. |
| nameCommunity | int | Se desplegará una lista enumerada que le mostrará al usuario las comunidades ingresadas en el sistema, lo que permitirá al usuario elegir la comunidad donde pertenece el producto que va a eliminar. |
| nameProduct | int | Se desplegará una lista enumerada que le mostrará al usuario los productos ingresados en el sistema, lo que permitirá al usuario elegir el producto que va a eliminar. |
| Resultado o Postcondición | Se debe escoger una de las comunidades registradas presentes en la lista desplegable para luego proceder a tener una lista desplegable con los productos. Después de presionar el botón de eliminar producto se mostrará un mensaje en donde diga que fue hecho con éxito o si por el contrario hubo un error y se tiene que intentar de nuevo. | | |
| Salidas | **Nombre salida** | **Tipo de dato** | **Formato** |
| successMessage | String | “El producto” + nameProduct + “se borró de una manera exitosa” |
| errorMessage | String | “El producto no se pudo eliminar, por favor vuelve a intentar de nuevo.” |

|  |  |  |
| --- | --- | --- |
| **Requerimiento Funcional** | **Nombre de la Clase** | **Nombre del método** |
| RF1  Registrar comunidad | Main | addComunity(…) |
| COP16 | addComunity(…) |
| RF2  Registrar un lugar | Main | addPlace(…) |
| COP16 | addPlace(…) |
| COP16 | searchPlaceUnique(…) |
| RF3  Registrar un producto | Main | addProduct(…) |
| COP16 | addProduct(…) |
| COP16 | availableProduct(…) |
| RF4  Registrar una especie | Main | addSpecie(…) |
| COP16 | addSpecie(…) |
| COP16 | availableSpecie(…) |
| COP16 | searchSpecieUnique(…) |
| RF5  Modificar los datos de una especie | Main | changeSpecieInfo(…) |
| COP16 | changeSpecieInfo(…) |
| RF6  Consultar comunidades en los departamentos | Main | searchCommunityByDepartments(…) |
| COP16 | searchCommunityByDepartments |
| RF7  Consultar una comunidad que tenga problemas de hospitales/escuelas. | Main | communityProblemsHospitalSchool(…) |
| COP16 | communityProblemsHospitalSchool(…) |
| RF8  Consultar el lugar con el mayor número de especies. | Main | placeMostSpecies(…) |
| COP16 | placeMostSpecies(…) |
| RF9  Consultar los 3 lugares más grandes por kilómetros cuadrados | Main | biggestThreePlaces(…) |
| COP16 | biggestThreePlaces(…) |
| RF10  Consultar un lugar | Main | searchPlace(…) |
| COP16 | searchPlace(…) |
| RF11  Eliminar un producto | Main | deleteProduct(…) |
| COP16 | deleteProduct(…) |