**ETD**

**ESPECIFICACIÓN TÉCNICA DE DISEÑO**

Proyecto: Sistema de Gestión de la Configuración de Software

Versión 1.0

**CONTROL DE VERSIONES**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Versión** | **Realizado por** | **Revisado por** | **Aprobado por** | **Fecha** | **Motivo** |
| 1.0 | Gonzales Cave, Angel Gabriel  Layme Valeriano, Diego Rolando  Mamani Calisaya, Yonathan William  Moreno Mulluni, Luis Angel | Gonzales Cave, Angel Gabriel  Layme Valeriano, Diego Rolando  Mamani Calisaya, Yonathan William  Moreno Mulluni, Luis Angel |  | 14/10/2019 | Versión original |

**ESPECIFICACIÓN TÉCNICA DE DISEÑO**

1. **IDENTIFICACIÓN**

|  |  |
| --- | --- |
| Cliente | Ing. Ricardo Eduardo Valcárcel Alvarado |
| Proyecto | Sistema de Gestión de la Configuración del Software |
| Diseñador | Angel Gonzales Cave – Luis Angel Moreno Mulluni |
| Referencia del Caso de Uso | Caso de uso : Gestionar Metodologías |
| Descripción Breve | Para que el usuario pueda gestionar las metodologías de desarrollo de software. |
| Complejidad | Baja |
| Horas | 3 horas de Implementación  1 horas de pruebas |

1. **TABLAS DE DATOS**

|  |
| --- |
| Se creará la base de datos dbscm con los siguientes campos:  CREATE TABLE IF NOT EXISTS metodologia  (  Id INT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,  Nombre VARCHAR (50) NULL,  )  GO |

|  |
| --- |
| CREATE TABLE IF NOT EXISTS fase  (  Id INT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,  Nombre VARCHAR (200) NOT NULL,  Metodologiaid INT UNSIGNED,  **FOREIGN** **KEY** (metodologiaid) REFERENCES metodologia(Id)  ) |

1. **DETALLE DE DISEÑO**

|  |
| --- |
| Al momento de ingresar al siguiente menú Metodologías luego en la opción de “Agregar” se mostrará la pantalla siguiente:  La ventana tendrá como nombre **Nueva Metodología** y tendrá como título “**Nueva Metodología”.**  Se encontrará una caja de texto input para ingresar el nombre de la metodología.  En la parte inferior de la caja de texto se encontrará el botón “Crear Metodología”, el cual se encargará de registrar los datos ingresados en la caja de texto.  Al presionar el botón Crear Metodología los datos registrados serán enviados a través de form action=”” al controlador Metodologia.Controller el cual determinará la acción realizada , que en este caso es ActAgregar() , luego se obtendrán los parámetros enviados desde la vista Metodologia.Agregar.  Luego se instanciara un objeto de la clase Metodologia para el registro en la Base de datos.  Luego de registrar el Controller redireccionará a la vista Metodologia.listar se mostrará el siguiente formulario “Listado de Metodologías” se mostrará una tabla con las siguientes columnas Numero correlativo, Nombre de la Metodologia, Acciones.    Al presionar el botón “EDITAR” la vista redireccionará a la vista Metodologia.Editar en donde se mostrará el campo Nombre de la metodología completado con la metodología, podemos editar el texto del campo y luego presionamos el botón “GUARDAR CAMBIOS”. La etiqueta form action enviará los parámetros al controlador Metodologia.Controller en la función ActEditar() , el cual recibirá el ID de la metodología y el nombre de la metodología, ejecutara la función EditarMetodologia().  Luego el controlador Metodologia.Controller redireccionará a la vista Metodologia.Listar y se podrá visualizar los cambios introducidos.    El botón “ELIMINAR” enviará el ID de la metodología si la metodología tiene asignadas alguna fase no se podrá eliminar por la relación de tablas que existe entre metodología y fase.  Al presionar el botón “VER” de la vista Metodologia.Listar se visualizara la vista Metodologia.Ver donde se podrá agregar las fases a la metodología.    El botón “+Fase” se encarga de registrar las fases. Si presionamos el botón se nos mostrará un modal con un campo de texto el cual se deberá de ingresar.  Al presionar el botón “CREAR FASE” la vista redireccionará al controlador FaseController el cual ejecutará la función ActAgregar(Request $request) donde se enviaron los parámetros de la vista la función instanciará una nuevo objeto de la Clase Fase(), y asignará a los parámetros el valor correspondiente y si el resultado es mayor a 0 la función redireccionará a la vista Metodologia.Ver.    Luego con el botón “Editar” el controlador FaseController ejecutará la función FrmEditar($FaseId) la cual obtendrá los datos de la metodología ejecutando la función ObtenerPorId($FaseId) para recuperar los datos de la faseid seleccionada y redireccionando a la Vista Fase.Editar.    Una vez redireccione a la vista Fase.Editar se mostrará el siguiente formulario llamado Fase “Nombre de la Fase”.    Al presionar el botón Guardar Cambios la vista Fase.Editar redirecciona los parametros al controlador FaseController donde ejecutará la función ActEditar(Request $request) y ejecutará la función ObtenerPorId() para recuperar el id de la fase y asignará el objeto Fase al parámetro nombre si el resultado de la funcion Editar es mayor a 0 se redireccionará a la vista Metodologia.Ver. |

1. **IDENTIFICACIÓN**

|  |  |
| --- | --- |
| Cliente | Ing. Ricardo Eduardo Valcárcel Alvarado |
| Proyecto | Sistema de Gestión de la Configuración del Software |
| Diseñador | Angel Gonzales Cave – Luis Angel Moreno Mulluni |
| Referencia del Caso de Uso | Caso de uso : Gestionar Planilla elementos de la configuración |
| Descripción Breve | El jefe de proyectos podrá asignar elementos de la configuración para una fase de una metodología seleccionada. |
| Complejidad | Media |
| Horas | 6 horas de Implementación  1 hora de prueba |

1. **TABLAS DE DATOS**

|  |
| --- |
| CREATE TABLE IF NOT EXISTS elemento\_configuracion  (  Id INT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,  Codigo VARCHAR (20) NOT NULL,  Nombre VARCHAR (200) NOT NULL,    ) |

|  |
| --- |
| CREATE TABLE IF NOT EXISTS plantilla\_elemento\_configuracion  (  Id INT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,  ElementoConfiguracionId INT NOT NULL,  Faseid INT UNISIGNED NOT NULL,  **FOREIGN** **KEY** (Faseid) REFERENCES fase(id),  **FOREIGN** **KEY** (ECSid) REFERENCES elemento\_configuracion(id)  ) |

1. **DETALLE DEL DISEÑO**

|  |
| --- |
| Ingresamos al menú principal nos aparecerá en el panel de administración la opción “ECS” y el submenú desplegable presionamos la opción “Agregar”. Nos aparecerá el formulario **Nuevo Elemento de Configuración.**    En el formulario podremos ingresar los siguientes campos **Código** el cual será el identificador del Elemento de Configuración y el **Nombre**.  Al presionar el botón **Crear ECS** la vista ECS.agregar envía los datos de los campos llenados al controlador ElementoConfiguracionController el cual a través de la función ActAgregarElementoConfiguracion(request $request) capturará los datos ingresados en la vista ECS.agregar,  Instanciará un nuevo objeto de la clase ElementoConfiguracion() , si los datos son registrados correctamente la función retornará a la vista ElementoConfiguracion.listar.  Al ingresar en el menú principal nos dirigimos en la opción “ECS” y presionamos la opción “Listar”, la opción nos redigirá a la vista ElementoConfiguracion.listar. Nos aparecerá un formulario llamado **“Listado de Elementos de Configuración”.**  En el formulario se listarán los elementos de la configuración registrados.    Para editar el nombre del Elemento de la Configuración es necesario presionar el botón “Editar”, el botón editar nos capturará el identificador único del Elemento de Configuración seleccionado nos redigira al controlador ElementoConfiguracionController y ejecutará la función FrmEditarElementoConfiguracion() capturando el Id del EC a través de la función ObtenerPorId() y nos enviará a la vista ElementoConfiguracion.Editar.  Estando en la vista ElementoConfiguracion.Editar nos cargará los datos del elemento de configuración seleccionado una vez realizado los cambios presionamos en el botón “**Guardar Cambios”** y nos redigira al controlador ElementoConfiguracionController y ejecutará la función ActEditarElementoConfiguracion() el cual obtendrá los nuevos cambios , sin antes ejecutar la función **ObtenerPorId() ,** si la función Editar(ObjECS) devuelve “1” el controlador nos enviará a la vista ElementoConfiguracion.Listar.  Para eliminar algún elemento de configuración es necesario presionar el botón “Eliminar”, el botón eliminar nos capturará el identificador del Elemento de Configuracion seleccionado nos redigirá al controlador ElementoConfiguracionController y ejecutará la función ActEliminar(Request $request) en donde ejecutará la función Eliminar($ObjPECS) y redireccionará a la vista Metodologia.Ver.  Luego para la asignación de los elementos de configuración a una fase en específico redireccionamos al formulario siguiente vista Metodologia.Ver :    En el modal “Nuevo Elemento de Configuración” se cargará las fases de la metodología dentro de la vista metodología.ver se ejecutara el controlador ElementoConfiguracionController, ejecutando la función ListarElementoConfiguracion() y retornará a la vista metodología.ver el listado de elemento de Configuración registrados.  En el segundo campo seleccionable “Combobox Fase” se listarán las fases de la metodología seleccionada a través del controlador MetodologiaController ejecutando la función VerMetodologia($MetodologiaId) el cual devolverá el listado de Fases de la metodología.  Una vez seleccionados los dos campos del modal    El usuario al presionar el botón CREAR ELEMENTO el botón invocará al controlador PlantillaElementoConfiguracionController el cual ejecutará la función ActAgregarPlantillaElementoConfiguracion(Request $request) en donde se instanciará un nuevo objeto de la Clase PlantillaElementoConfiguracion. Si la función devuelve “1” el registro se realizó correctamente.  En caso se requiera remover algún elemento de configuración de una fase predeterminada nos dirigimos a la vista Metodologia.Ver en donde seleccionaremos el botón siguiente :    Si presionamos el botón Remover como la imagen resalta la vista Metodologia.ver redireccionará al controlador PlantillaElementoConfiguracionControlller en donde ejecutará la función ActEliminar(Request $request) donde instanciará un objeto de la clase PlantillaElementoConfiguracion para obtener su id ejecuta la funcion ObtenerPorId() y redireccionará a la vista Metodologia.Ver. |

1. **IDENTIFICACIÓN**

|  |  |
| --- | --- |
| Cliente | Ing. Ricardo Eduardo Valcárcel Alvarado |
| Proyecto | Sistema de Gestión de la Configuración del Software |
| Diseñador | Angel Gonzales Cave – Luis Angel Moreno Mulluni |
| Referencia del Caso de Uso | Caso de uso : Gestionar Proyecto |
| Descripción Breve | El jefe de proyectos podrá registrar los proyectos con la metodología y fase con sus respectivos ECS. |
| Complejidad | Media |
| Horas | 18 horas de Implementación  1 hora de prueba |

1. **TABLA DE DATOS**

|  |
| --- |
| CREATE TABLE IF NOT EXISTS proyecto  (  Id INT (10) AUTO\_INCREMENT PRIMARY KEY,  Codigo VARCHAR(191) NOT NULL,  Nombre VARCHAR (191) NOT NULL,  FechaInicio DATE NOT NULL,  FechaTermino DATE NOT NULL,  UsuarioJefeId INT(11) NOT NULL,  MetodologiaId INT(10) NOT NULL,  Descripcion VARCHAR (191) NOT NULL,  Estado VARCHAR (191) NOT NULL,  **FOREIGN** **KEY** (MetodologiaId) REFERENCES metodologia(Id)  **FOREIGN** **KEY** (UsuarioJefeId) REFERENCES usuario(Id)  ) |

1. **DETALLE TÉCNICO**

|  |
| --- |
| Para la gestión de proyecto nos dirigimos a la vista Proyecto.Listar en la cual el controlador ProyectoController redireccionará a la función FrmAgregar() el cual retornará un arrayList ListadoMetodologia.  En el campo Metodología se listarán todas las metodologías registradas una vez seleccionada el campo se listarán las fases y los elementos que la componen.  Para la gestión de proyecto es necesario contar con la plantilla de elementos de la configuración listando a través del controlador MetodologiaController ejecutando la función VerMetodologia() retornando un Array ListadoPlantillaECS.    Ingresados los campos solicitados, la vista Proyecto.agregar redireccionará al controlador ProyectoController y ejecutará la función ActAgregar(Request $request) en donde se instanciará un nuevo objeto de la Clase Proyecto() y se asignará a cada objeto de los parametros enviados desde la vista Proyecto.Agregar además si el resultado es mayor a 0 , es decir se ejecuto correctamente el registro se instancia un nuevo objeto de la clase Cronograma() y a cada objeto $Cronograma se le asignará los parámetros enviados de la vista Proyecto.agregar y además de registrar es decir el resultado sea mayor a 0 , se instanciara un nuevo objeto de la clase CronogramaFase() en donde se asignará al objeto $ObjCronogramaFase los parámetros desde enviados desde la vista Proyecto.Agregar, luego se instanciará un nuevo objeto $ElementoConfiguracion de la clase CronogramaElementoConfiguracion() si el resultado es mayor de 0 se redireccionará ala vista Proyecto.Listar. |

1. **IDENTIFICACIÓN**

|  |  |
| --- | --- |
| Cliente | Ing. Ricardo Eduardo Valcárcel Alvarado |
| Proyecto | Sistema de Gestión de la Configuración del Software |
| Diseñador | Angel Gonzales Cave – Luis Angel Moreno Mulluni |
| Referencia del Caso de Uso | Caso de uso : Gestionar Miembros |
| Descripción Breve | El jefe de proyectos podrá registrar aquellas responsabilidades que consideren que un proyecto tenga y además de los nombres de los responsables que se encargarán de realizar las tareas de los entregables. |
| Complejidad | Media |
| Horas | 8 horas de Implementación  1 hora de prueba |

1. **TABLA DE DATOS**
2. **DETALLE TÉCNICO**

|  |  |
| --- | --- |
| Cliente | Ing. Ricardo Eduardo Valcárcel Alvarado |
| Proyecto | Sistema de Gestión de la Configuración del Software |
| Diseñador | Diego Layme Valeriano |
| Referencia del Caso de Uso | Caso de uso : Gestionar Cronograma |
| Descripción Breve | El jefe de proyecto podrá administrar un proyecto iniciando con la asignación de tareas y los responsables de la realización de las tareas. |
| Complejidad | Media |
| Horas | 24 horas de Implementación  1 hora de prueba |

|  |
| --- |
| CREATE TABLE IF NOT EXISTS cronograma  (  Id INT(10) UNSIGNED AUTO\_INCREMENT PRIMARY KEY,  Proyectoid INT(10) NOT NULL,  FechaInicio DATE NOT NULL,  FechaTermino DATE NOT NULL,  **FOREIGN** **KEY** (ProyectoId) REFERENCES proyecto(Proyectoid)  ) |

|  |
| --- |
| CREATE TABLE IF NOT EXISTS cronograma\_elemento\_configuracion  (  Id INT(10) AUTO\_INCREMENT PRIMARY KEY,  Nombre VARCHAR(191) NOT NULL,  Codigo VARCHAR(191) NOT NULL,  CronogramaFaseId INT(11) NOT NULL,  **FOREIGN** **KEY** (CronogramaFaseId) REFERENCES cronograma\_elemento\_configuracion(CronogramaFaseId)  ) |

|  |
| --- |
| CREATE TABLE IF NOT EXISTS cronograma\_fase  (  Id INT(10) AUTO\_INCREMENT PRIMARY KEY,  CronogramaId INT(11) NOT NULL,  Nombre VARCHAR(191) NOT NULL,  **FOREIGN** **KEY** (CronogramaId ) REFERENCES cronograma(CronogramaId)  ) |