

Identificación del problema y análisis de requerimientos

Cliente	GreenSQA
Usuario	-Administrador -Trabajadores
Requerimientos funcionales	<i>RF1-Gestion de lo proyectos</i> <i>RF2-Gestión de las etapas del proyecto</i> <i>RF3- Culminación de una etapa de proyecto</i> <i>RF4-Registrar cápsulas de conocimiento</i> <i>RF5- Aprobación de las capsulas</i> <i>RF6- Publicación de las cápsulas a la organización</i> <i>RF7- Consultar capsulas</i>
Contexto del problema	<p>GreenSQA es una empresa de tecnología que trabaja en proyectos de aseguramiento de calidad de software para organizaciones que buscan productos de alta calidad. Nuestro cliente ha descrito el problema de la fuga de conocimiento en empresas de TI y ha establecido el objetivo de retener el conocimiento de los empleados antes de que roten a otros empleadores. Para ello, piden que se cree un proceso de captura de conocimiento mediante cápsulas generadas por los colaboradores. Cada etapa del proyecto se divide en seis etapas y los colaboradores generan cápsulas de conocimiento en cada etapa. Las cápsulas aprobadas se publicarán en la Intranet de la organización y se podrán buscar mediante una cadena de búsqueda o "hashtag".</p>
Requerimientos no funcionales	<ul style="list-style-type: none"> - Seguridad: El código debe garantizar la seguridad de la información confidencial, previniendo cualquier tipo de acceso no autorizado y ataques malintencionados a los datos almacenados o transmitidos. - Usabilidad: El proceso de captura de conocimiento debe ser sencillo de utilizar y estar disponible para todos los usuarios de la organización, sin importar su nivel de habilidad técnica. - Escalabilidad: El código debe ser capaz de manejar grandes volúmenes de cápsulas de conocimiento y ser escalable para soportar un aumento en la cantidad de usuarios y datos en el futuro. - Rendimiento: El proceso de captura de conocimiento debe ser rápido y eficiente, y el código debe tener la capacidad de procesar grandes cantidades de datos sin afectar el rendimiento. - Mantenibilidad: El código debe estar diseñado de manera que sea fácil de mantener y actualizar, adaptándose a los cambios de requerimientos y permitiendo agregar nuevas funcionalidades en el futuro.

Identificador y nombre	<i>[RF1-Gestion de lo proyectos]</i>		
Resumen	Al recibir un proyecto aprobado de un cliente, el software debe almacenar toda la información necesaria del mismo. La capacidad de gestionar estos datos es vital para fomentar una comunicación eficaz entre los gerentes de ambas partes y para supervisar el progreso del trabajo. Después de ingresar los datos se muestra en pantalla un mensaje de confirmación o de error dependiendo si las entradas son correctas. Para el período piloto, el software deberá tener la capacidad de almacenar un máximo de diez proyectos.		
Entradas	Nombre entrada	Tipo de dato	Condición valores válidos
	name_project	String	No se permite ingresar números ni caracteres especiales. Solo letras (Max 25)
	name_client	String	No se permite ingresar números ni caracteres especiales. Solo letras (Max 25)
	startDate	date	Se debe ingresar una fecha valida de calendario (Solo

			números) de la siguiente manera: [dd/mm/yyyy]
	endDate	date	Se debe ingresar una fecha valida de calendario (Solo números) de la siguiente manera: [dd/mm/yyyy]
	budget_project	String	No se permite ingresar letras ni caracteres especiales. Solo números (Sin puntos)
	name_manager	String	No se permite ingresar números ni caracteres especiales. Solo letras (Max 25)
Resultado o Postcondición	Se almacenan los datos del proyecto en la base de datos. Se muestra en pantalla una ventana emergente mostrando el registro con éxito o si hay un error por corregir.		
Salidas	Nombre salida	Tipo de dato	Formato
	log_Result	String	Información confirmando el registro con éxito del proyecto, en caso de haber algún error; informa corregir.

Identificador y nombre	<i>[RF2-Gestión de las etapas del proyecto]</i>
Resumen	Cuando se crea un proyecto, automáticamente el programa debe crear sus 6 etapas (inicio, análisis, diseño, ejecución, cierre y seguimiento y control del proyecto) pero solo la etapa de inicio debe quedar activa. Cada etapa tendrá una fecha de inicio y fin (planeada) y una fecha de inicio y fin (real). También el programa guardará la aprobación del cumplimiento de la etapa y para

	lograr asignar las fechas planeadas se deberá solicitar al usuario la cantidad de meses que se lleva cada etapa.		
Entradas	Nombre entrada	Tipo de dato	Condición valores válidos
Entradas	months	int	Se debe ingresar la cantidad de meses en las que se desarrolla cada etapa, solo se permite ingresar números enteros.
Entradas	start_StagePlan	date	Se debe ingresar una fecha valida de calendario esta entrada es para ingresar la fecha planeada en la que se inicia la etapa (Solo números) de la siguiente manera: [dd/mm/yyyy]
	end_StagePlan	date	Se debe ingresar una fecha valida de calendario, esta entrada es para ingresar la fecha planeada en la que se finaliza la etapa (Solo números) de la siguiente manera: [dd/mm/yyyy]
	start_StageReal	date	Se debe ingresar una fecha valida de calendario esta entrada es para ingresar la fecha real en la que se inicia la etapa (Solo números) de la siguiente manera: [dd/mm/yyyy]
Resultado o Postcondición	Se almacenan los datos de la etapa. Se muestra en pantalla una ventana emergente mostrando el registro con éxito o si hay un error por corregir. Y si todo es correcto se muestra en pantalla que la Etapa esta activa.		

	Nombre salida	Tipo de dato	Formato
Salidas	result_Stage	String	Información confirmando el registro con éxito de la etapa, en caso de haber algún error; informa corregir.
Salidas	condition	String	El programa procede a mostrar que la etapa se encuentra Activa

Identificador y nombre	<i>[RF3- Culminación de una etapa de proyecto]</i>		
Resumen	Cuando el usuario aprueba una etapa, el programa debe pedir la fecha real de finalización, después la condición de la etapa pasa a inactiva para activar la siguiente. Posteriormente muestra un mensaje en pantalla que confirma al usuario que la siguiente etapa ya se encuentra disponible.		
Entradas	Nombre entrada	Tipo de dato	Condición valores válidos
Entradas	status_stage	String	Se debe ingresar el estado de la etapa, entre (Por definir, Aprobada, No aprobada) solo letras, no se aceptan números ni caracteres especiales
Entradas	end_StageReal (Solo si el estado de la etapa es Aprobado)	date	Se debe ingresar una fecha valida de calendario, esta entrada es para ingresar la fecha real en la que se finaliza la etapa (Solo

			números) de la siguiente manera: [dd/mm/yyyy]
Resultado o Postcondición	Se guarda el nuevo estado de la etapa con su fecha de finalización y se muestra en pantalla que ya esta disponible la nueva etapa.		
Salidas	Nombre salida	Tipo de dato	Formato
	new_Status	String	Información confirmando el nuevo estado de la etapa, también se informa que la nueva etapa ya está habilitada.

Identificador y nombre	[RF4-Registrar cápsulas de conocimiento]		
Resumen	En cada una de las etapas del proyecto, se deben generar cápsulas de conocimiento. El programa debe recibir de estas capsulas un identificador único, una descripción de la situación que desea registrar, un tipo de cápsula, entre (técnico, gestión, dominio y experiencias), el nombre y cargo del colaborador. Además, el aprendizaje o lección aprendida con dicha situación. Estos junto con la descripción deben iniciar y terminar con “#” en las palabras claves.		
Entradas	Nombre entrada	Tipo de dato	Condición valores válidos
Entradas	id	String	Debe ser un número con letras, no se permite espacios vacíos, caracteres especiales ni puntos. (Max 12 caracteres)
Entradas	description	String	Se debe ingresar que hace la capsula. No se permite ingresar caracteres especiales, debe tener por lo

			<p>menos una palabra clave que inicie con “#”</p> <p>(Max 300 caracteres)</p>
Entradas	capsule_Type	String	<p>Se debe ingresar el tipo de la capsula, solo se permite ingresar las siguientes palabras:</p> <p>(Técnico, gestión, dominio y experiencias)</p>
Entradas	name_Employee	String	<p>Se debe ingresar el nombre del colaborador, no se permite ingresar números, puntos ni caracteres especiales</p>
Entradas	position_Employee	String	<p>Se debe ingresar el cargo del colaborador, no se permite ingresar números, puntos ni caracteres especiales</p>
Entradas	knowledge	String	<p>Se debe ingresar el conocimiento adquirido. No se permite ingresar caracteres especiales, debe tener por lo menos una palabra clave que inicie con “#”</p>
Resultado o Postcondición	<p>Se guarda esta capsula de conocimiento en la etapa creada, además se muestra en pantalla un mensaje que verifica que la información de la capsula fue guardada correctamente y el estado de la capsula queda “por definir” de manera predeterminada.</p>		
Salidas	Nombre salida	Tipo de dato	Formato
	capsule_Created	String	<p>Información confirmando la nueva capsula registrada con éxito.</p>

Identificador y nombre	<i>[RF5- Aprobación de las capsulas]</i>		
Resumen	Cuando el usuario aprueba una capsula, el programa procede a pedir la fecha en la que fue aprobada.		
Entradas	Nombre entrada	Tipo de dato	Condición valores válidos
Entradas	status_Capsule	String	Se debe ingresar el estado de la capsula, entre (Por definir, Aprobada, No aprobada) solo letras, no se aceptan números ni caracteres especiales
Entradas	date_approved (Solo si el estado de la etapa es Aprobado)	date	Se debe ingresar una fecha valida de calendario, esta entrada es para ingresar la fecha real en la que se aprobó la capsula(Solo números) de la siguiente manera: [dd/mm/yyyy]
Resultado o Postcondición	Se guarda el nuevo estado de la capsula con su fecha de aprobación y se muestra en pantalla que se guardó con éxito.		
Salidas	Nombre salida	Tipo de dato	Formato
	true_message	String	Información confirmando el nuevo estado de la capsula.

Identificador y nombre	<i>[RF6- Publicación de las cápsulas a la organización]</i>		
Resumen	Las capsulas que son de interés organizacional y son aprobadas, se generan en formato HTML para ser publicadas en la Intranet de la Organización, por lo cual se almacenará la url.		
Resultado o Postcondición	Si la capsula es de interés organizacional y ya fue aprobada el programa genera la url del formato HTML		
Salidas	Nombre salida	Tipo de dato	Formato
	htl_message	String	Información confirmando que se ha generado el html

			correctamente y se muestra su url.
--	--	--	------------------------------------

Identificador y nombre	<i>[RF7- Consultar capsulas]</i>		
Resumen	El programa debe permitir consultar a las personas de la empresa las capsulas de conocimiento creadas mediante palabras claves que están con “#”		
Entradas	Nombre entrada	Tipo de dato	Condición valores válidos
Entradas	search_Word	String	Se debe ingresar las palabras claves para buscar la capsula de conocimiento, estas palabras deben estar con “#” no se permiten números ni caracteres especiales.
Resultado o Postcondición	Se muestra en pantalla la capsula de conocimiento encontrada, en caso de no encontrarla muestra una venta emergente que informa que no existe una capsula que tenga esa palabra.		
Salidas	Nombre salida	Tipo de dato	Formato
	search_Result	String	Información mostrando el resultado de la búsqueda encontrada mediante las palabras claves.
Salidas	Error_Result	String	Información mostrando que el resultado de la búsqueda no fue exitoso