



Universidad Luterana Salvadoreña

**Técnico en Desarrollo de Aplicaciones
informáticas**

Tema:

Entrega final del Proyecto

Asignatura:

Evaluación de la Calidad del Software

Docente:

Inga. Lesbia Mancía

Estudiantes:

Rene Alexis Castro Hernández

Karla Daniela Rivera Rivera

Kevin Francisco López Reyes

Flor Abigail Climaco Rivera

INDICE

INTRODUCCION DEL PROYECTO	3
OBJETIVOS GENERAL Y ESPECIFICOS DEL PROYECTO FINAL.....	4
TEMA DEL PROYECTO	5
DESCRIPCION DEL PROYECTO.....	5
JUSTIFICACION DEL PROYECTO	5
APLICACIÓN DE LOS NORMAS ISO	6
APLICACIÓN DE LAS TECNICAS DE EVALUACIÓN DE CALIDAD DE SOFTWARE	6
PLAN DE EVALUACION DE CALIDAD DE SOFTWARE	22
DOCUMENTACION DE LAS PRUEBAS REALIZADAS AL SOFTWARE	32
ANEXOS	39

INTRODUCCION DEL PROYECTO

En el presente trabajo explicaremos sobre el “CrudTodolist” consiste en colocar tareas y editarla para próximas entregas, como grupo se realizó una investigación minuciosamente sobre el software, se realizó el estudio de las diferentes ISO en donde cada una de ellas parecen ser iguales, pero cada una con diferentes funciones para poder realizar el estudio a nuestra aplicación. Cada ISO juega un papel muy importante desde la identificación de cada una de las características y las su sub características que las compone y el significado que lleva cada uno, para darle cada uso en el estudio del Software que realizamos, en si identificar cada paso minuciosamente para poder identificar lo que estamos estudiantes. Posteriormente nos dedicamos a recopilar los requerimientos de la aplicación para estudiar los errores que tenía y las funcionalidades que posee, además se realizó un diagrama de clases en donde minuciosamente se hizo el estudio de la aplicación. También se hizo un estudio de checklist, se estudió el código y se analizó el diagrama de clases.

Se realizó el estudio para identificar los diferentes caminos por medio de caja blanca en donde se estudia el código y luego se realiza un diagrama para ordenar los caminos, como grupo también hicimos el estudio de un plan de la evaluación de la calidad del software. Así mismo se realizó un informe sobre los casos de prueba para esta aplicación.

OBJETIVOS GENERAL Y ESPECIFICOS DEL PROYECTO FINAL

GENERAL

Analizar los diferentes métodos que utilizamos para poder realizar el estudio de software del Crudtodolist el cual fue muy importante tomar en cuenta los diferentes pasos que podríamos realizar para que se nos hiciera más útil el analizar este software.

ESPECIFICOS

- Identificar cada una de las metodologías utilizadas para realizar estudio del software del Crudtodolist.

- Interpretar los diferentes pasos que se utilizaron para recopilar la información para la realización de dicho estudio.

- Indicar cada una de las diferentes variantes que se realizó en la investigación.

TEMA DEL PROYECTO

Evaluación de aplicación "Todolist"

DESCRIPCION DEL PROYECTO

En este trabajo se han realizado diferentes pruebas para la ejecución de la app todolist siguiendo el reglamento de las normas ISO, así como también se han realizado las técnicas de abstracción y pruebas de caja blanca todo esto con el fin de detectar y resolver las fallas técnicas que se puedan encontrar.

JUSTIFICACION DEL PROYECTO

La aplicación "TODOLIST" se a realizado para tener un mayor resguardo de todas nuestras notas y es por eso que desde el inicio se ha venido revisando meticulosamente cada paso realizado para su elaboración, desde las herramientas a utilizar, lenguaje de programación.

Todo esto con el fin de tener una app fácil de utilizar y sobre todo segura para el usuario. Y es por eso que se le han realizado las diferentes pruebas para que sea segura y fácil de usar.

APLICACIÓN DE LOS NORMAS ISO

Norma ISO/IEC 9126-1

Norma ISO/IEC 9126-2

Norma ISO/IEC 9126-3

Norma ISO/IEC 9126-4

APLICACIÓN DE LAS TECNICAS DE EVALUACIÓN DE CALIDAD DE SOFTWARE

Técnica de abstracción sucesiva:

Técnica de abstracción sucesiva (ToDoList)				
Requisito	Código	Abstracción Sucesiva	Cumpl e	
			SI	NO
1.1 El sistema deberá verificar que el correo electrónico contenga @ y el dominio	<pre> this.loginS.registerWithEmailAnd Passwor(email, pass).then((user) => console.log(user); if (!user) alert('Error al crear el usuario'); } </pre>	<p>Se captura email y password del usuario.</p> <p>Se imprime en consola los datos que se capturen de user.</p> <p>Sí el usuario no es correcto entonces mensaje de alerta "Error al crear el usuario"</p>	X	

<p>1.1.1 El sistema deberá enviar un mensaje que el correo contenga @ y el dominio</p>		<p>En este caso el programa no cumple con la parte de mandar un mensaje del contenido que se le requiere</p>		<p>X</p>
<p>1.2 El sistema deberá verificar que contraseña y repetir contraseña sean iguales</p>	<pre>const { email, pass, pass1 } = this.usuarios; if (pass != pass1) { alert('Las contraseñas no coinciden'); return; }</pre>	<p>Se captura email, y ambas contraseñas del usuario. Si pass diferente de pass1 entonces mensaje de alerta "Las contraseñas no coinciden"</p>	<p>X</p>	
<p>1.2.1 El sistema deberá enviar un mensaje que las contraseñas coincidan</p>	<pre>alert('Las contraseñas no coinciden');</pre>	<p>Se alerta con el mensaje "Las contraseñas no coinciden"</p>	<p>X</p>	
<p>1.3 El sistema deberá corroborar que el correo no esté registrado previamente</p>	<p>No aplica</p>			

<p>1.3.1 El sistema deberá enviar un mensaje que ya está registrado el correo</p>	<p>No aplica</p>			
<p>2.1 El sistema deberá verificar que el correo electrónico contenga @ y el dominio</p>	<pre>const { email, pass } = this.usuarios; this.loginS.loginEmailAndPasswo rd(email, pass).then((user) => { if (!user) { alert('Error al logearse'); return; } this.router.navigate(['/home']); this.usuarios.email = ""; this.usuarios.pass = ""; });</pre>	<p>Se captura email y contraseñas del usuario. Si el usuario diferente, entonces mensaje de alerta "Error al logearse". Luego retorna a la pagina /home donde el usuario registra email y pass</p>	<p>X</p>	

<p>2.2 El sistema deberá verificar que contraseña y correo coincidan con el registro dado</p>	<pre>const { email, pass } = this.usuarios; this.loginS.loginEmailAndPassword(email, pass).then((user) => { if (!user) { alert('Error al logearse'); return; } })</pre>	<p>Se captura email y contraseñas del usuario. Si el usuario diferente, entonces mensaje de alerta "Error al logearse".</p>	<p>X</p>	
<p>3.1 El sistema deberá verificar que todos los datos descritos sean llenados completamente</p>	<p>No aplica</p>			<p>X</p>
<p>4.1 El sistema deberá verificar que, a la hora de actualizar los datos descritos, no dejar ningún campo vacío.</p>	<p>No aplica</p>			<p>X</p>

5.1 El sistema elimina la tarea solo al presionar el botón eliminar.	<pre>eliminarTarea() this.db.eliminarTarea(this.id); this.router.navigate(['/home']); }</pre>	{ En este caso el programa si cumple con lo requerido	X	
6. El usuario puede salirse de la aplicación.	<pre>logout() this.loginService.logout(); }</pre>	{ En este caso el programa si cumple con lo requerido	X	

CHECKLIST PARA REQUISITOS

REQUERIMIENTOS	PREGUNTAS CHECKLIST	SI	NO
----------------	---------------------	----	----

1. El usuario podrá registrarse al ingresar a la aplicación:	¿Existen contradicciones en la especificación de los requisitos?		X
a) Correo electrónico b) Contraseña c) Repetir Contraseña	¿Resulta comprensible la especificación?	X	
1.1 El sistema deberá verificar que el correo electrónico contenga @ y el dominio	¿Está especificado el rendimiento?	X	
1.1.1 El sistema deberá enviar un mensaje que el correo contenga @ y el dominio	¿Puede ser eliminado algún requisito? ¿Pueden juntarse dos requisitos?	X	X
1.2 El sistema deberá verificar que contraseña y repetir contraseña sean iguales	¿Son redundantes o contradictorios?	X	X
1.2.1 El sistema deberá enviar un mensaje que las contraseñas coincidan	¿Se han especificado todos los recursos hardware necesarios?		
1.3 El sistema deberá corroborar que el correo no este registrado previamente	¿Se han especificado las interfaces externas necesarias?	X	
1.3.1 El sistema deberá enviar un mensaje que ya está registrado el correo	¿Se han definido los criterios de aceptación para cada una de las funciones especificadas?	X	
2. El usuario podrá logearse:	¿Existen contradicciones en la especificación de los requisitos?	X	
a) Correo electrónico b) Contraseña			
2.1 El sistema deberá verificar que el correo electrónico contenga @ y el dominio	¿Resulta comprensible la especificación?	X	

<p>2.2 El sistema deberá verificar que contraseña y correo coincidan con el registro dado</p>	<p>¿Está especificado el rendimiento?</p> <p>¿Puede ser eliminado algún requisito?</p> <p>¿Pueden juntarse dos requisitos?</p> <p>¿Son redundantes o contradictorios?</p> <p>¿Se han especificado todos los recursos hardware necesarios?</p> <p>¿Se han especificado las interfaces externas necesarias?</p> <p>¿Se han definido los criterios de aceptación para cada una de las funciones especificadas?</p>	<p>X</p> <p>X</p> <p>X</p> <p>X</p> <p>X</p> <p>X</p>	<p>X</p> <p>X</p>
<p>3. El usuario puede crear una nota:</p> <p>a) Tarea b) Descripción de tarea c) Estado (creada, en proceso, finalizada) d) Fecha de creación de tarea</p> <p>3.1 El sistema deberá verificar que todos los datos descritos sean llenados completamente</p>	<p>¿Existen contradicciones en la especificación de los requisitos?</p> <p>¿Resulta comprensible la especificación?</p> <p>¿Está especificado el rendimiento?</p> <p>¿Puede ser eliminado algún requisito?</p> <p>¿Pueden juntarse dos requisitos?</p>	<p>X</p> <p>X</p> <p>X</p> <p>X</p> <p>X</p>	<p>X</p>

	<p>¿Son redundantes o contradictorios?</p> <p>¿Se han especificado todos los recursos hardware necesarios?</p> <p>¿Se han especificado las interfaces externas necesarias?</p> <p>¿Se han definido los criterios de aceptación para cada una de las funciones especificadas?</p>	<p>X</p> <p>X</p> <p>X</p> <p>X</p>	<p>X</p> <p>X</p> <p>X</p> <p>X</p>
<p>El usuario puede actualizar una nota:</p> <p>a) Tarea b) Descripción de tarea c) Estado (creada, en proceso, finalizada) d) Fecha de creación de tarea</p> <p>4.1 El sistema deberá verificar que, a la hora de actualizar los datos descritos, no dejar ningún campo vacío.</p>	<p>¿Existen contradicciones en la especificación de los requisitos?</p> <p>¿Resulta comprensible la especificación?</p> <p>¿Está especificado el rendimiento?</p> <p>¿Puede ser eliminado algún requisito?</p> <p>¿Pueden juntarse dos requisitos?</p> <p>¿Son redundantes o contradictorios?</p> <p>¿Se han especificado todos los recursos hardware necesarios?</p>	<p>X</p> <p>X</p> <p>X</p> <p>X</p> <p>X</p> <p>X</p> <p>X</p>	<p>X</p> <p>X</p> <p>X</p> <p>X</p> <p>X</p> <p>X</p> <p>X</p>

	¿Se han especificado las interfaces externas necesarias?	X	
	¿Se han definido los criterios de aceptación para cada una de las funciones especificadas?	X	
5. El usuario puede eliminar una nota: a) Tarea b) Descripción de tarea c) Estado (creada, en proceso, finalizada) d) Fecha de creación de tarea 5.1 El sistema elimina la tarea solo al presionar el botón eliminar.	¿Existen contradicciones en la especificación de los requisitos? ¿Resulta comprensible la especificación? ¿Está especificado el rendimiento? ¿Puede ser eliminado algún requisito? ¿Pueden juntarse dos requisitos?	X X X X	X X X
6. El usuario puede salirse de la aplicación.	¿Son redundantes o contradictorios? ¿Se han especificado todos los recursos hardware necesarios? ¿Se han especificado las interfaces externas necesarias? ¿Se han definido los criterios de aceptación para cada una de las funciones especificadas?	X X X X	X

CHECKLIST PARA DISEÑO

REQUERIMIENTOS	PREGUNTAS CHECKLIST	SI	NO
<p>1. El usuario podrá registrarse al ingresar a la aplicación:</p> <p>a) Correo electrónico b) Contraseña c) Repetir Contraseña</p> <p>1.1 El sistema deberá verificar que el correo electrónico contenga @ y el dominio</p> <p>1.1.1 El sistema deberá enviar un mensaje que el correo contenga @ y el dominio</p> <p>1.2 El sistema deberá verificar que contraseña y repetir contraseña sean iguales</p> <p>1.2.1 El sistema deberá enviar un mensaje que las contraseñas coincidan</p> <p>1.3 El sistema deberá corroborar que el correo no este registrado previamente</p> <p>1.3.1 El sistema deberá enviar un mensaje que ya está registrado el correo</p>	<p>¿Cubre el diseño todos los requisitos funcionales?</p> <p>¿Resulta ambigua la documentación del diseño?</p> <p>¿Se ha aplicado la notación de diseño correctamente?</p> <p>¿Se han definido correctamente las interfaces entre elementos del diseño?</p> <p>¿Es el diseño suficientemente detallado como para que sea posible implementarlo en el lenguaje de programación elegido?</p>	<p>X</p> <p>X</p> <p>X</p> <p>X</p> <p>X</p>	
<p>2. El usuario podrá logearse:</p> <p>a) Correo electrónico b) Contraseña</p> <p>2.1 El sistema deberá</p>	<p>¿Cubre el diseño todos los requisitos funcionales?</p> <p>¿Resulta ambigua la</p>	<p>X</p> <p>X</p>	

<p>verificar que el correo electrónico contenga @ y el dominio</p> <p>2.2 El sistema deberá verificar que contraseña y correo coincidan con el registro dado</p>	<p>documentación del diseño?</p> <p>¿Se ha aplicado la notación de diseño correctamente?</p> <p>¿Se han definido correctamente las interfaces entre elementos del diseño?</p> <p>¿Es el diseño suficientemente detallado como para que sea posible implementarlo en el lenguaje de programación elegido?</p>	<p>X</p> <p>X</p> <p>X</p>	
<p>3. El usuario puede crear una nota:</p> <p>a) Tarea b) Descripción de tarea c) Estado (creada, en proceso, finalizada) d) Fecha de creación de tarea</p> <p>3.1 El sistema deberá verificar que todos los datos descritos sean llenados completamente</p>	<p>¿Cubre el diseño todos los requisitos funcionales?</p> <p>¿Resulta ambigua la documentación del diseño?</p> <p>¿Se ha aplicado la notación de diseño correctamente?</p> <p>¿Se han definido correctamente las interfaces entre elementos del diseño?</p> <p>¿Es el diseño suficientemente detallado como para que sea posible implementarlo en el lenguaje de programación elegido?</p>	<p>X</p> <p>X</p> <p>X</p> <p>X</p>	<p>X</p> <p>X</p>
<p>4. El usuario puede actualizar una nota:</p> <p>a) Tarea b) Descripción de tarea</p>	<p>¿Cubre el diseño todos los requisitos funcionales?</p>	<p>X</p>	

<p>c) Estado (creada, en proceso, finalizada) d) Fecha de creación de tarea</p> <p>4.1 El sistema deberá verificar que, a la hora de actualizar los datos descritos, no dejar ningún campo vacío.</p>	<p>¿Resulta ambigua la documentación del diseño?</p> <p>¿Se ha aplicado la notación de diseño correctamente?</p> <p>¿Se han definido correctamente las interfaces entre elementos del diseño?</p> <p>¿Es el diseño suficientemente detallado como para que sea posible implementarlo en el lenguaje de programación elegido?</p>	<p>X</p> <p>X</p> <p>X</p> <p>X</p>	<p>X</p>
<p>5. El usuario puede eliminar una nota:</p> <p>a) Tarea b) Descripción de tarea c) Estado (creada, en proceso, finalizada) d) Fecha de creación de tarea</p> <p>5.1 El sistema elimina la tarea solo al presionar el botón eliminar.</p> <p>6. El usuario puede salirse de la aplicación.</p>	<p>¿Cubre el diseño todos los requisitos funcionales?</p> <p>¿Resulta ambigua la documentación del diseño?</p> <p>¿Se ha aplicado la notación de diseño correctamente?</p> <p>¿Se han definido correctamente las interfaces entre elementos del diseño?</p> <p>¿Es el diseño suficientemente detallado como para que sea posible implementarlo en el lenguaje de programación elegido?</p>	<p>X</p> <p>X</p> <p>X</p> <p>X</p>	<p>X</p>

CHEKLIST PARA CODIGO

PREGUNTAS CHEKLIST	SI	NO
Lógica del programa: ¿Es correcta la lógica del programa?	X	
¿Está completa la lógica del programa?, es decir, ¿está todo correctamente especificado sin faltar ninguna función		X
Interfaces Internas: ¿Es igual el número de parámetros recibidos por el módulo a probar al número de argumentos enviados?, además, ¿el orden es correcto? ¿Los atributos (por ejemplo, tipo y tamaño) de cada parámetro recibido por el módulo a probar coinciden con los atributos del argumento correspondiente? ¿Coinciden las unidades en las que se expresan parámetros y argumentos? Por ejemplo, argumentos en grados y parámetros en radianes. ¿Altera el módulo un parámetro de sólo lectura? ¿Son consistentes las definiciones de variables globales entre los módulos?	X X	 X
Interfaces Externas: ¿Se declaran los ficheros con todos sus atributos de forma correcta? ¿Se abren todos los ficheros antes de usarlos?		X X

<p>¿Coincide el formato del fichero con el formato especificado en la lectura? ¿Se manejan correctamente las condiciones de fin de fichero? ¿Se los libera de memoria?</p> <p>¿Se manejan correctamente los errores de entrada/salida?</p>	<p>X</p>	<p>X</p>
<p>Datos.</p> <p>Utilizar variables no inicializadas</p> <p>Salirse del límite de las matrices y vectores</p> <p>Superar el límite de tamaño de una cadena</p> <p>Comprobar que no hay dos variables con el mismo nombre</p> <p>Comprobar que todas las variables estén declaradas</p> <p>Comprobar que las longitudes y tipos de las variables sean correctos.</p> <p>Cálculo.</p> <p>Comprobar que no se producen overflow o underflow (valores fuera de rango, por encima o por debajo) en los cálculos o divisiones por cero.</p> <p>Comparación.</p> <p>Comprobar que no existen comparaciones entre variables con diferentes tipos de datos o si las variables tienen diferente</p>	<p>X</p> <p>X</p> <p>X</p> <p>X</p> <p>X</p> <p>X</p> <p>X</p> <p>X</p> <p>X</p>	<p>X</p> <p>X</p> <p>X</p> <p>X</p> <p>X</p>

longitud.		
-----------	--	--

Lectura Activa de Diseño.

Defectos	Correcciones
<p>El método registrarse era igual que la clase.</p> <p>Las iniciales de las propiedades del método Registrar estaban en minúscula.</p> <p>En el diagrama de clase habían caracteres especiales por ejemplo la Ñ.</p> <p>En las propiedades del método Registrarse no estaba la opción de Repetir contraseña</p> <p>Habían espacios en blanco en algunas propiedades del diagrama</p>	<p>El método se cambió a Registrar.</p> <p>Las iniciales de las propiedades del método fueron cambiadas a mayúsculas.</p> <p>En las propiedades del método Registrarse fue cambiada contraseña a contraseña.</p> <p>En las propiedades del método Registrarse fue agregada la opción de Repetir contraseña</p> <p>La solución fue agregar el guión bajo.</p>

Diagrama de clase:

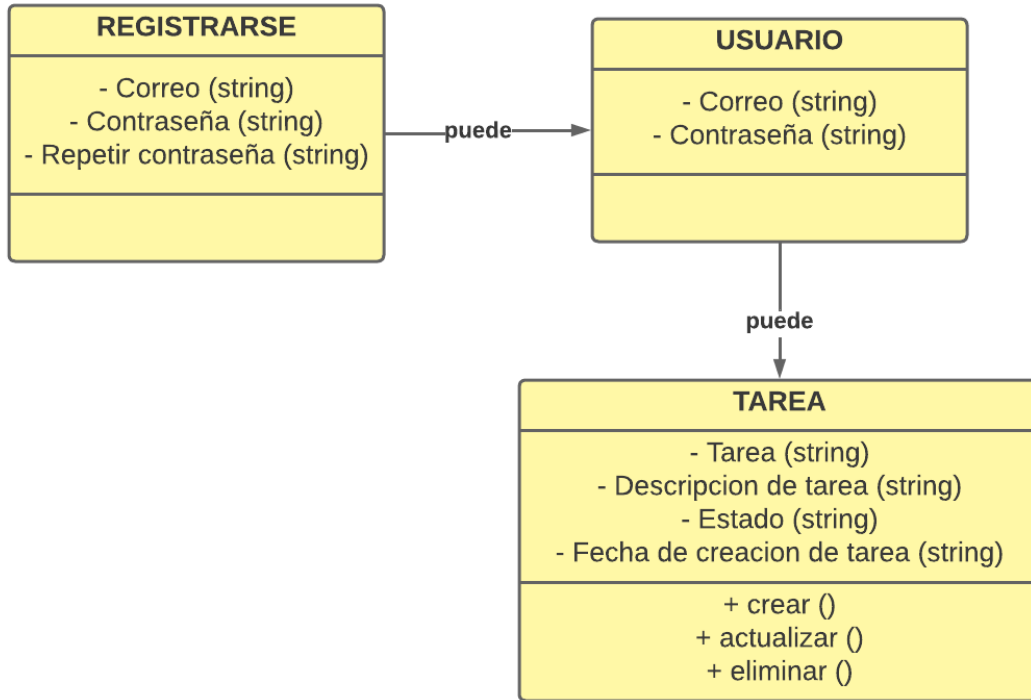
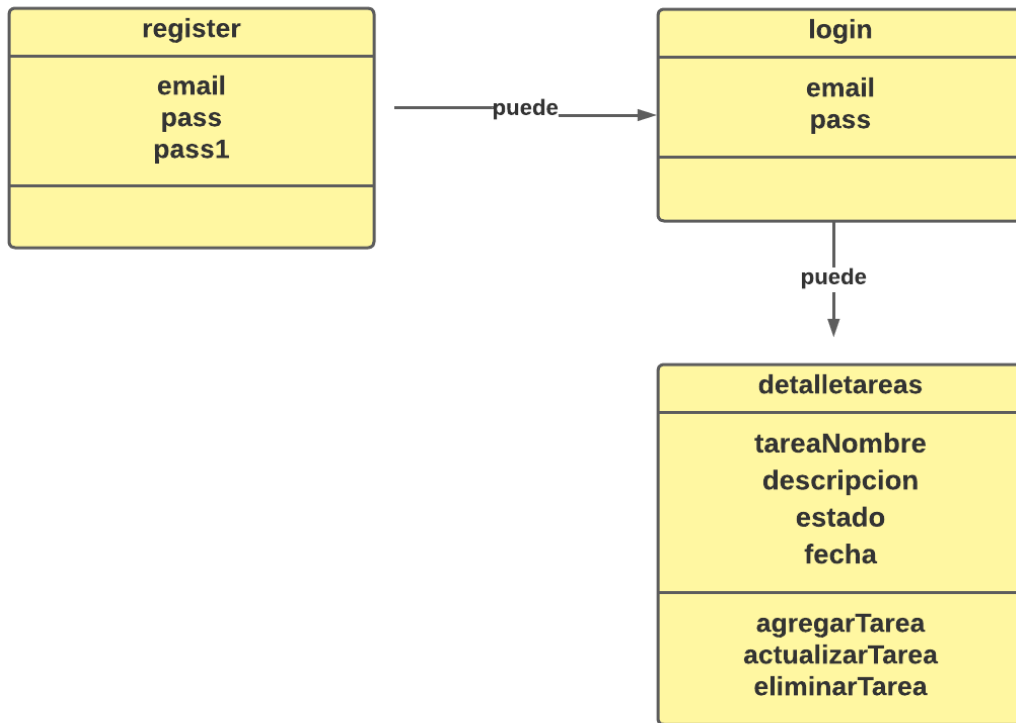


Diagrama de clase base código



PLAN DE EVALUACION DE CALIDAD DE SOFTWARE

Revisiones y auditorías

Objetivo

Definición de las revisiones y auditorías técnicas y de gestión que se realizarán.
Especificación de cómo serán llevadas a cabo dichas revisiones y auditorías.

Requerimientos mínimos

Revisión de requerimientos:

Esta revisión se realiza para asegurar que se cumplió con los requerimientos especificados por el Cliente.

Revisión de diseño preliminar

Esta revisión se realiza para asegurar que el diseño de la aplicación tenga una vista previa y la consistencia, suficiencia técnica del diseño preliminar del software.

Revisión de diseño crítico

Esta revisión se realiza para asegurar la consistencia del diseño detallado con la especificación de requerimientos.

Revisión del Plan de Verificación & Validación

Esta revisión se realiza para asegurar la consistencia y completitud de los métodos especificados en el Plan de V & V.

Auditoría funcional

Esta auditoría se realiza previa a la liberación del software, para verificar que todos los requerimientos especificados en el documento de requerimientos fueron cumplidos.

Auditoría física

Esta revisión se realiza para verificar que el software y la documentación son consistentes y están aptos para la liberación.

Auditorías internas al proceso

Estas auditorías son para verificar la consistencia: del código versus el documento de diseño, especificaciones de interfase, implementaciones de diseño versus requerimientos funcionales, requerimientos funcionales versus descripciones de testeo.

Revisiones de gestión

Estas revisiones se realizan periódicamente para asegurar la ejecución de todas las actividades identificadas en este Plan. Deben realizarse por una persona ajena al grupo de trabajo (en caso de que sea posible). Se debe de hacer una revisión por

otra persona que no pertenezca al grupo en este caso es la docente que nos realiza las revisiones.

Revisión del Plan de gestión de configuración

Esta revisión se realiza para asegurar la consistencia y completitud de los métodos especificados en el Plan de gestión de configuración.

Revisión Post Mortem

Esta revisión se realiza al concluir el proyecto para especificar las actividades de desarrollo implementadas durante el proyecto y para proveer recomendaciones.

Agenda

[En esta sección se deberá especificar la agenda para las revisiones y auditorías detalladas anteriormente.] se realizará el orden en que se ha ido trabajando y realizando las entregas de nuestro proyecto con las fechas específicas de acuerdo según la planificación de nuestro docente y se dividía en primera etapa, segunda etapa y entrega final

Otras revisiones

Revisión de documentación de usuario

Se revisa la completitud, claridad, correctitud y aplicación de uso.

Documentación

Propósito

Identificación de la documentación relativa a desarrollo, Verificación & Validación, uso y mantenimiento del software.

Establecer como los documentos van a ser revisados para chequear consistencia: se confirman criterio e identificación de las revisiones.

Documentación mínima requerida

La documentación mínima es la requerida para asegurar que la implementación logrará satisfacer los requerimientos.

Especificación de requerimientos del software

El documento de especificación de requerimientos deberá describir, de forma clara y precisa, cada uno de los requerimientos esenciales del software además de las interfaces externas.

El cliente deberá obtener como resultado del proyecto una especificación adecuada a sus necesidades en el área de alcance del proyecto, de acuerdo al compromiso inicial del trabajo y a los cambios que este haya sufrido a lo largo del proyecto, que cubra aquellos aspectos que se haya acordado detallar con el cliente.

La especificación debe:

- Ser completa :
 - a. Externa, respecto al alcance acordado.
 - b. Internamente, no deben existir elementos sin especificar.
 - Ser consistente, no pueden haber elementos contradictorios.
 - Ser no ambigua, todo término referido al área de aplicación debe estar definido en un glosario.
 - Ser verificable, debe ser posible verificar siguiendo un método definido, si el producto final cumple o no con cada requerimiento.
 - Estar acompañada de un detalle de los procedimientos adecuados para verificar si el producto cumple o no con los requerimientos.
 - Incluir requerimientos de calidad del producto a construir.

Los requerimientos de calidad del producto a construir son considerados dentro de atributos específicos del software que tienen incidencia sobre la calidad en el uso ISO 9126-4 y se detallan a continuación:

Funcionalidad

- a. adecuación a las necesidades
- b. precisión de los resultados
- c. seguridad de los datos

Confiabilidad

- a. madurez
- b. tolerancia a faltas
- c. recuperabilidad

Usabilidad

- a. comprensible
- b. aprendible
- c. operable

Eficiencia

- a. comportamiento respecto al tiempo
- b. utilización de recursos

Mantenibilidad

- a. analizable
- b. modificable
- c. estable, no se producen efectos inesperados luego de modificaciones
- d. verificable

Portabilidad

- a. adaptable (Ver si aplica)
- b. instalable

Cada uno de estos atributos debe cumplir con las normas y regulaciones aplicables a cada uno.

Descripción del diseño del software

El documento de diseño especifica como el software será construido para satisfacer los requerimientos.

Este documento deberá ser elaborado primero como Preliminar y luego será gradualmente extendido hasta llegar a obtener el Detallado. Deberá describir los componentes y subcomponentes del diseño del software, incluyendo interfaces internas.

El cliente deberá obtener como resultado del proyecto el diseño de un producto de software que cubra aquellos aspectos que se haya acordado con el cliente incorporar al diseño, en función de la importancia que estos presenten y de sus conexiones lógicas.

El diseño debe:

- Corresponder a los requerimientos a incorporar:
 - a. Todo elemento del diseño contribuye a los requerimiento

- b. La implementación de todo requerimiento a incorporar debe estar contemplada en por lo menos un elemento del diseño.
- Ser consistente con la calidad del producto

Plan de Verificación & Validación

El Plan de V & V deberá identificar y describir los métodos a ser utilizados en:

- La verificación de que:
 - a. los requerimientos descritos en el documento de requerimientos han sido aprobados por una autoridad apropiada. En este caso sería que cumplan con el acuerdo logrado entre el cliente y el equipo.
 - b. los requerimientos descritos en el documento de requerimientos son implementados en el diseño expresado en el documento de diseño.
 - c. el diseño expresado en el documento esta implementado en código.
- Validar que el código, cuando es ejecutado, se adecua a los requerimientos expresados en el documento de requerimientos.

Reportes de Verificación & Validación

Este documento especifica el resultado de la ejecución de los procesos descritos en el Plan de V & V.

Documentación de usuario

El documentación de usuario específica y describe los datos y entradas de control requeridos, así como la secuencia de entrada, limitaciones de programa y otros elementos necesarios para la ejecución exitosa del software.

Los errores identificados y las acciones correctivas descritas.

Como resultado del proyecto el cliente obtendrá una documentación para el usuario de acuerdo a los requerimientos específicos del proyecto.

Plan de Gestión de configuración

El Plan de gestión de configuración contiene métodos para identificar componentes de software, control e implementación de cambios, y registro y reporte del estado de los cambios implementados.

Otros documentos

N/A

Gestión de software

Organización

Las líneas de trabajo dentro de la organización son verificación y gestión de proyecto en este caso se utilizara el metodología scrum.

Ciclo de vida de un software cubierto por el plan

Las etapas más importantes dentro del ciclo de vida del software son la etapa del relevamiento de requerimientos y el principio de la etapa de diseño dado una buena especificación de requerimientos, plan de verificación, alcances de sistema y descripción de la arquitectura. Todo esto aplicado a nuestra aplicación todolist.

Actividades de calidad a realizarse

Revisar nuestro producto

Realizar revisión técnica formal

Revisar el ajuste al proceso

Asegurar que las actividades son documentadas

Revisar cada producto

Se deberá identificar, documentar y seguir la pista a las desviaciones encontradas y verificar las fallas en nuestra aplicación.

Al encontrar fallas realizaremos una documentación y seguir a detalle para darle solución al problema.

Revisar el ajuste al proceso

se realizara una recopilación de datos para resolver fallas y así entregar un buen producto, todo esto se realizara al obtener los datos obtenidos de el plan del proyecto, plan de la iteración, plan de verificación.

La revisión del producto se realizara utilizando una checklist.

Realizar revisión técnica formal (RTF)

La revisión técnica se realiza para descubrir errores en la funciones, este es un proceso riguroso, su objetivo es llegar a detectar fallas en el producto todo esto involucra al equipo de trabajo.

Relaciones entre las actividades de SQA y la planificación

Actividad	Semana cuando se realiza
Planificar la Calidad	Semana 1
Evaluar y Ajustar el Plan de SQA	Semanas 2 y 3
Revisión Técnica Formal (RTF)	Semanas 4, 5 6, 7, 8, 9,10,11 y 12
Revisar las entregas	Todas las semanas
Revisar el Ajuste al Proceso	Semanas 3 a 12 (inclusive)
Evaluar la calidad de los productos	Semanas 3 a 14 (inclusive)
Realizar el informe final de SQA	Semana 14
Describir la Versión	Semanas 5, 7, 9, 11 y 13
Escribir las notas de la versión	Semanas 6, 8, 10, 12 y 14

Estándares, prácticas, convenciones y métricas Estándar de documentación

Como estándares de documentación se definirán dos documentos:

- Estándar de documentación técnica y
- Estándar de documentación de usuario.

La documentación técnica del producto debe:

- Ser adecuada para que un grupo independiente del de desarrollo pueda encarar el mantenimiento del producto.
- Incluir fuentes, Modelos de Casos de Uso, Objetos

Para la escritura de documentos se han definido plantillas para ser utilizadas en la elaboración de entregables.

En estas plantillas se definen:

- encabezado y pie de página.
- fuente y tamaño de fuente para estilo normal
- fuente y tamaño de fuente para los títulos a utilizar
- datos mínimos que se deben incluir: fecha, versión y responsables.

En nuestro proyecto en los estándares de documentos fueron:

- Tipo de letra Arial.
- Tamaño de letra 12.
- Letra interlineada.
- Letra Justificada.
- Utilización de las normas APA.

Para la creación de la documentación del manual de usuario se debe estar seguro que la aplicación funciona y cumpla con lo que debe realizar, para esto se deben hacer pruebas y dar a conocer las diferentes funciones de la aplicación todolist.

El manual debe contener al menos las siguientes funciones:

1. Registrarse.
2. Iniciar sesión.
3. Podemos agregar notas o ver las que ya están.
4. Especificamos nombre de la nota , descripción, estado(creada, en proceso, finalizada) y fecha que fue creada.
5. Podemos actualizar nota.
6. Eliminar nota.

Estándar de verificación y prácticas

Se utilizan las prácticas definidas en el Plan de Verificación y Validación.

Como estándar se utiliza el documento de:

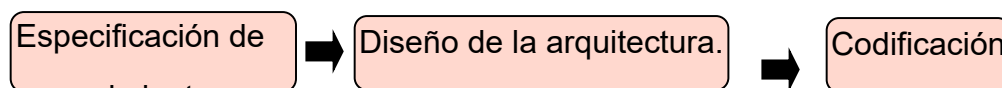
Std 1012-1986 IEEE Standard for Software Verification and Validation Plans.

Otros Estándares

- Norma ISO 9126-2
- Norma ISO 9126-3
- Norma ISO 9126-4

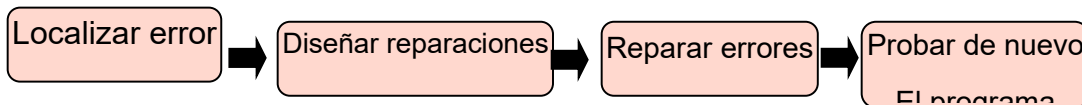
Verificación

Para realizar la verificación y validación estática y dinámica debe llevar las Inspecciones de software:



Se realizaran pruebas dando a los usuarios la aplicación para que ellos vean que posibles errores tiene la aplicación o para que los desarrolladores vean lo que da error y cambiarlo o que se les dificulta a los usuarios para también modificarlo y hacerlo mas fácil el uso

Para el proceso de depuración debe de llevar lo siguiente:



Reporte de problemas y acciones correctivas

En esta se colora los siguientes valores que se deben de tomar en cuenta para la implementación del desarrollo:

Número	Error	Descripción	Acción Correctiva

Herramientas, técnicas y metodologías

Las técnicas que se usaron en el proyecto fueron:

- Técnica de abstracción sucesiva

- Lectura Activa de Diseño y Diagrama de Clase (Código).
- Checklis, Requerimientos, Diseño, Código.
- Caja blanca.

Gestión de riesgos

Ver Documento de Riesgos.

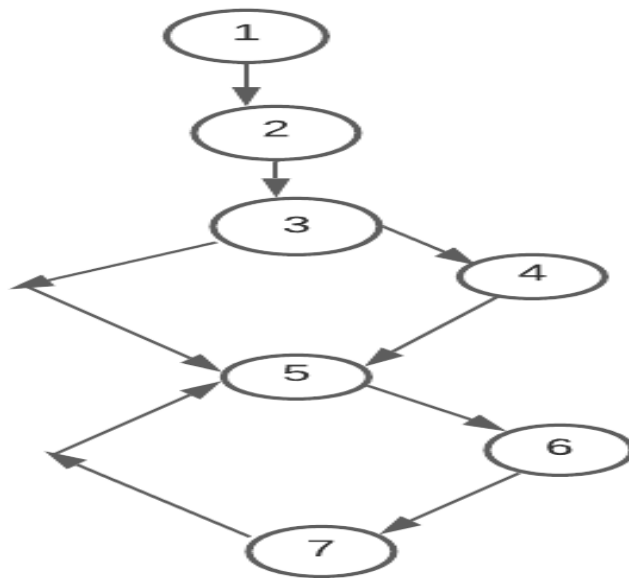
DOCUMENTACION DE LAS PRUEBAS REALIZADAS AL SOFTWARE

- Funciones:

```
export class RegisterPage implements OnInit {
  usuarios = {
    email: '',
    pass: '',
    pass1: '',
  };
  constructor(private loginS: LoginService, private router: Router) {}

  ngOnInit() {}
  registrar() {
    //destructuring
    const { email, pass, pass1 } = this.usuarios;
    if (pass !== pass1) {
      alert('Las contraseñas no coinciden');
      return;
    }
    this.loginS.registerWithEmailAndPasswor(email, pass).then((user) => {
      console.log(user);
      if (!user) {
        alert('Error al crear el usuario');
      }
    });
    this.router.navigate(['/login']);
    this.usuarios.email = '';
    this.usuarios.pass = '';
    this.usuarios.pass1 = '';
  }
}
```


Gráfico de flujo:



- $V(G)$

$V(G) = 3$

- Identificar caminos:

1. 1-2-3-4-5-6-7
2. 1-2-3-4-5-6
3. 1-2-3-5-7

- Casos de prueba:

Numero	Caminos	Registro
1	1-2-3-4-5-6-7	Se representa por medio de un Email, también por medio de los caminos se identifica que otras opciones tendrá dicha aplicación.
2	1-2-3-4-5-6	Luego se le coloca una contraseña con varios dígitos para que se ha segura, y que también se ha fácil de recordar para el usuario así mismo orienta a las personas a proteger sus cuentas.
3	1-2-3-5-7	Se colocará de nuevo la contraseña para verificar si es correcta

Casos de prueba Caja Blanca (Registro)

Primer camino

Nombre	Camino 1-2-3-4-5-6-7
Objetivo	En esta parte muestra los diferentes caminos para poder obtener lo que se está buscando
Entorno	Para que el usuario comprenda mejor la aplicación y sobre todo que pueda realizarla de manera eficaz
Interfaz	Presenta una serie de procedimiento para poder crear un registro
Componente	Se colocará un email, y una contraseña en donde también se repetirá la misma contraseña
Acciones	Se registran las personas que necesitaran acceder a la aplicación
Valores de entrada	El email y la contraseña son los valores de entrada.
Resultados obtenidos	Creación de un usuario satisfactoriamente
Resultado Esperados	Un usuario creado
Desvíos	Posibles fallas a la hora de crearlo
Observaciones	Al no colocar bien el correo electrónico o la contraseña puede dar errores

Segundo Camino

Nombre	1-2-3-4-5-6
Objetivo	Identificar al usuario por medio de un email personal
Entorno	Para que la persona comprenda más el uso de la aplicación
Interfaz	Escribir su email personal y contraseña
Componente	Consiste en el email y la contraseña
Acciones	Se pueden registrar de manera eficaz
Valores de entrada	Email y contraseña
Resultados obtenidos	Que la creación no de errores

Resultado Esperados	El funcionamiento perfecto de la aplicación
Desvíos	Posibles fallas a la hora de crearlo
Observaciones	No se observan ninguna

Tercer Camino

Nombre	1-2-3-5-7
Objetivo	Colocar una contraseña segura
Entorno	Que la contraseña se ha fácil de recordar para el usuario
Interfaz	Contraseña con muchos dígitos para que se ha confiable
Componente	Se realiza para que el usuario tenga segura su cuenta
Acciones	Contraseña fácil y segura
Valores de entrada	Es el tercer paso para crear un usuario
Resultados obtenidos	Creación de una contraseña segura
Resultado Esperados	Una contraseña eficaz
Desvíos	Posibles fallas a la hora de crear la contraseña
Observaciones	No se observan ninguna

-

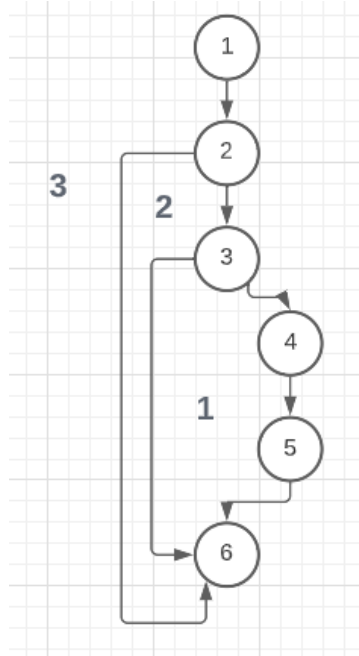
Funciones:

```

ngOnInit() {}
registrar() {
  //destructuring
  const { email, pass, pass1 } = this.usuarios; 1
  if (pass !== pass1) { 2
    alert('Las contraseñas no coinciden');
    return;
  }
  this.loginS.registerWithEmailAndPasswor(email, pass).then((user) => { 3
    console.log(user); 4
    if (!user) { 5
      alert('Error al crear el usuario');
    }
  });
  this.router.navigate(['/login']); 6
  this.usuarios.email = '';
  this.usuarios.pass = '';
  this.usuarios.pass1 = '';
}
}

```

- Gráfico de flujo:



- $V(G)$

$V(G) = 3$

- Identificar caminos:

- 1) 1-2-6
- 2) 1-2-3-4-5-6
- 3) 1-2-3-6

- Casos de prueba:

Número	Camino	email	pass	pass1	Resultado
1	1 2 6	reyes_1090@gmail.com	123	12345	Las contraseñas no coinciden email = " pass = " pass1 = "

2	1 2 3 4 5 6	Lopez_05@gmail.com	5555	5555	Si los datos del usuario coinciden en todo el proceso, sea en logearse o registrarse pasarían al módulo de tareas
3	1 2 3 6	uls2022@gmail.com	2022	2022	Si los datos del usuario coinciden con el registro email = ""; pass = ""; pass1 = ""; lo re direcciona a Login

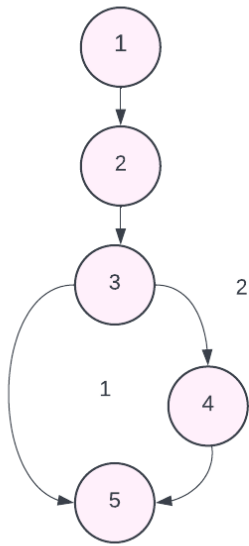
Funciones:

```

27   private router: Router
28   ) {}
29
30   ngOnInit() {
31     this.id = JSON.parse(this.rutaActiva.snapshot.params.id); 1
32     this.userId = JSON.parse(this.rutaActiva.snapshot.params.userId); 2
33     if (this.id) { 3
34       this.loadTareas(); 4
35     } 5
36   }
37   async loadTareas() {
38     const loading = await this.loadingCtrl.create({
39     message: 'Cargando...',

```

Gráfico de flujo:



$V(G)$

$V(G) = 2$

Identificar caminos:

1-2-3-4-5

Casos de prueba:

Número	Camino	nombre	Descripción	estado	fecha
1	1-2-3-4-5	prueba	Prueba número seis	En proceso	20112022

ANEXOS

Autoevaluación		
Nombre del Estudiante: René Alexis Castro Hernandez		
N°	Aspecto a Evaluar	Rúbrica (Nota)
1	Colaboré con las tareas que me fueron asignadas	9
2	Participé en forma activa en el trabajo de equipo	9
3	Mantuve comunicación con el equipo	7
4	Cumplí a tiempo con las actividades designadas	9
5	Aporté ideas de calidad	7
Total		41
Autoevaluación		
Nombre del Estudiante: Karla Daniela Rivera Rivera		
N°	Aspecto a Evaluar	Rúbrica (Nota)
1	Colaboré con las tareas que me fueron asignadas	9
2	Participé en forma activa en el trabajo de equipo	9
3	Mantuve comunicación con el equipo	7
4	Cumplí a tiempo con las actividades designadas	8
5	Aporté ideas de calidad	8
Total		41

Autoevaluación		
Nombre del Estudiante: Flor Abigail Climaco Rivera		
N°	Aspecto a Evaluar	Rúbrica (Nota)
1	Colaboré con las tareas que me fueron asignadas	8
2	Participé en forma activa en el trabajo de equipo Autoevaluación	8
3	Mantuve comunicación con el equipo	7
Nombre del Estudiante: Kevin Francisco López Reyes		
4	Cumplí a tiempo con las actividades designadas	9
N°	Aspecto a Evaluar	Rúbrica (Nota)
5	Aporté ideas de calidad	9
1	Colaboré con las tareas que me fueron asignadas	9
Total		41
2	Participé en forma activa en el trabajo de equipo	9
3	Mantuve comunicación con el equipo	7
4	Cumplí a tiempo con las actividades designadas	8
5	Aporté ideas de calidad	8
Total		41

Coevaluaciones

Coevaluación		
Nombre del Evaluador: Karla Daniela Rivera Rivera		
Nombre del Evaluado: René Alexis Castro Hernandez		
N°	Aspecto a Evaluar	Rúbrica (Nota)
1	Demostró compromiso y responsabilidad con el grupo	8
2	Demostró respeto y tolerancia hacia las opiniones de los demás	9
3	Aportó al desarrollo del proyecto	9
4	Propicia un clima de trabajo agradable	8
5	Antes de entregar la tarea, fue revisado por el evaluado	9
Total		43

Coevaluación		
Nombre del Evaluador: René Alexis Castro Hernandez		
Nombre del Evaluado: Karla Daniela Rivera Rivera		
N°	Aspecto a Evaluar	Rúbrica (Nota)
1	Demostró compromiso y responsabilidad con el grupo	9
2	Demostró respeto y tolerancia hacia las opiniones de los demás	8

	demás	
3	Aportó al desarrollo del proyecto	8
4	Propicia un clima de trabajo agradable	9
5	Antes de entregar la tarea, fue revisado por el evaluado	9
Total		43

Coevaluación		
Nombre del Evaluador: Kevin Francisco López Reyes		
Nombre del Evaluado: Flor Abigail Climaco Rivera		
N°	Aspecto a Evaluar	Rúbrica (Nota)
1	Demostró compromiso y responsabilidad con el grupo	8
2	Demostró respeto y tolerancia hacia las opiniones de los demás	8
3	Aportó al desarrollo del proyecto	9
4	Propicia un clima de trabajo agradable	9
5	Antes de entregar la tarea, fue revisado por el evaluado	9
Total		43

Coevaluación		
Nombre del Evaluador: Flor Abigail Climaco Rivera		
Nombre del Evaluado: Kevin Francisco López Reyes		
N°	Aspecto a Evaluar	Rúbrica (Nota)
1	Demostró compromiso y responsabilidad con el grupo	8
2	Demostró respeto y tolerancia hacia las opiniones de los demás	8
3	Aportó al desarrollo del proyecto	9
4	Propicia un clima de trabajo agradable	9
5	Antes de entregar la tarea, fue revisado por el evaluado	9
Total		43