

Two thick, horizontal, slightly wavy bars. The top bar is teal and the bottom bar is yellow, both spanning the width of the page.

Seiton

Autor: Samuel Chamizo González

Tutor: Javier Martín Rivero

Fecha de entrega: 10/06/2022

Convocatoria: 2021 202

Índice

2. INTRODUCCIÓN.....	1
3. ABSTRACT	1
4. ORIGEN	2
5. DISEÑO DEL PROTOTIPO	3
Inicio de sesión	4
Registro.....	4
Panel principal.....	4
Panel de creación de actividad.....	4
Panel de calendario.....	4
Panel de informes	4
6. OBJETIVOS PROPUESTOS	4
7. ANÁLISIS	5
Requisitos funcionales	5
Requisitos no funcionales.....	5
Diagramas de Casos de Uso.....	6
Inicio de sesión	6
Registro	7
Crear actividad	8
Borrar actividad.....	9
Ver actividad	10
Responder notificaciones.....	11
Actualizar calendario	12
Actializar mes calendario	13
Cambiar imagen	14
Actualizar informe	15

Enviar correo	16
Diagrama de clases.....	16
Diagramas de Entidad-Relación	17
8. DISEÑO DEL DESARROLLO.....	17
Inicio de sesión	17
Registro.....	18
Principal	18
Panel de crear actividad.....	19
Panel de calendario.....	19
Panel del gráfico circular	20
Panel de contacto	20
Clase “logica”	21
Clase “conexion”	21
Hilo “HiloEnvíaMensaje”	22
Hilo “NotificacionTareas”	22
9. DESPLIEGUE Y PRUEBAS.....	22
10. METODOLOGÍAS UTILIZADAS.....	25
Primeras ideas	25
Forma de trabajo.....	25
Sesión 1 (10/03/2022).....	25
Sesión 2 (20/03/2022).....	26
Sesión 3 (1/04/2022)	26
Sesión 4 (2/05/2022)	27
Sesión 5 (9/05/2022)	28
Sesión 6 (16/05/2022).....	28
Sesión 7 (23/05/2022).....	29

Sesión 8 (30/05/2022).....	29
Sesión 9 (3/06/2022)	29
11. ESTIMACIÓN DE RECURSOS Y PLANIFICACIÓN.....	31
12. TECNOLOGÍAS Y HERRAMIENTAS UTILIZADAS.....	31
Hostings	31
Desarrollo.....	31
Edición de imágenes	32
13. VIAS FUTURAS	32
Inicio de sesión	32
Principal	32
Crear actividad	32
Calendario.....	32
Informes	32
Panel Contacto.....	33
General	33
14. CONCLUSIONES	33
Lecciones aprendidas	33
Dificultades/errores	33
Opinión personal	34
15. BIBLIOGRAFÍA/WEBGRAFÍA	35
16. ANEXOS	1
Diagrama de Gantt - Estimación.....	1
Diagrama de Gantt – Conclusión.....	3
Diagrama de clases.....	6

1. INTRODUCCIÓN

A lo largo de este documento, se va a mostrar el aplicativo de desarrollo con nombre **Seiton**.

Seiton es un programa creado en el que el usuario podrá **introducir actividades**. El programa procesará la información introducida por el usuario y plasmará el contenido en un calendario.

El aplicativo se ha desarrollado intentando dejar muy claras las funcionalidades. Se ha implementado una serie de iconos que muestren el estado de la actividad, si es una actividad propia, compartida o está en espera de que la otra persona acepte.

Por otra parte, ¿cuántas veces has pensado en qué dedicas tantas horas de tu vida?

Ya no tienes que preocuparte por eso, Seiton te crea un informe con las horas que dedicas a categorías como trabajo, ocio, entrenamiento, familia. De esta forma podrás tener un control y decidir si cambiar tu día a día o estás dedicando el tiempo a lo que quieres.

2. ABSTRACT

Throughout this document, the development application named Seiton will be shown.

Seiton is a program created in which the user can enter activities. The program will process the information entered by the user and will translate the content into a calendar.

The application has been developed trying to make the functionalities very clear. A series of icons have been implemented to show the status of the activity, whether it is an own activity, shared or waiting for the other person to accept.

On the other hand, how many times have you thought about what you spend so many hours of your life on?

You no longer have to worry about that, Seiton creates a report with the hours you dedicate to categories such as work, leisure, training, family. This way you can have a control and decide whether to change your day to day or you are dedicating your time to what you want.

3. ORIGEN

El proyecto nace del valor de uno de los bienes más preciados de la vida, **el tiempo**. El tiempo es algo que no podemos comprar ni recuperar por eso es algo **único**.



Imagen 1 - Logo Seiton (Samuel, Propio, 2022)

Este programa no da una puntada sin hilo y tanto es así que debemos conocer la raíz del significado de su nombre, **Seiton**.

El nombre proviene de una técnica desarrollada en Japón llamada las 5S (Cinco eses) la cual hace referencia a 5 acciones implementadas.



Imagen 2 – Estructura (Anónimo, 2022)

Teniendo en cuenta el proyecto, **Seiton** es la acción que más refleja lo que se quiere conseguir por lo tanto se le ha atribuido ese nombre.

Pero, ¿para quién está enfocado? Pues esta es una de las muchas ventajas, que es **eficiente** para todo tipo de personas. Lo pueden utilizar tanto jefes de taller en empresas para **gestionar** e **indicar las tareas** de un empleado, como un niño/a de primaria para organizar sus ejercicios y tiempos de **estudio** de cara a los exámenes.

4. DISEÑO DEL PROTOTIPO

Con la idea más o menos clara, se empezó a trabajar en un prototipo gráfico para ir guiando el proyecto. Aunque hay nuevas funcionalidades y otras que se han quitado porque no aportaban valor, este era el prototipo inicial de Seiton.

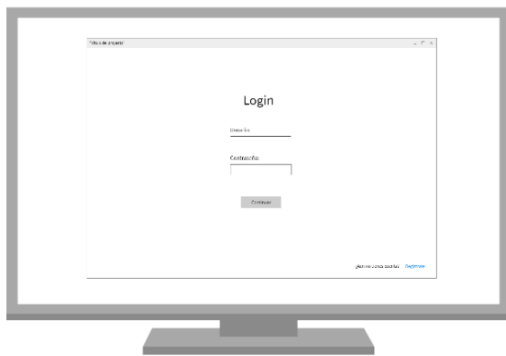


Imagen 3 – Prototipo (Samuel, Mockflow, 2022)

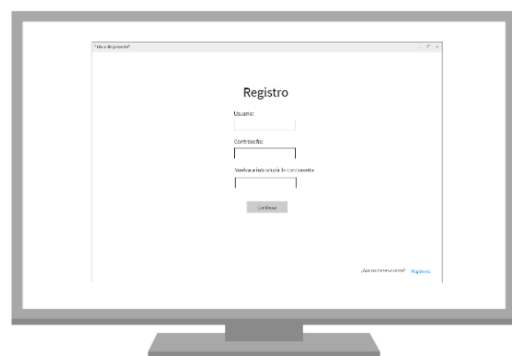


Imagen 4 - Prototipo (Samuel, Mockflow, 2022)



Imagen 5 – Prototipo (Samuel, Mockflow, 2022)



Imagen 6 – Prototipo (Samuel, Mockflow, 2022)



Imagen 7 – Prototipo (Samuel, Mockflow, 2022)

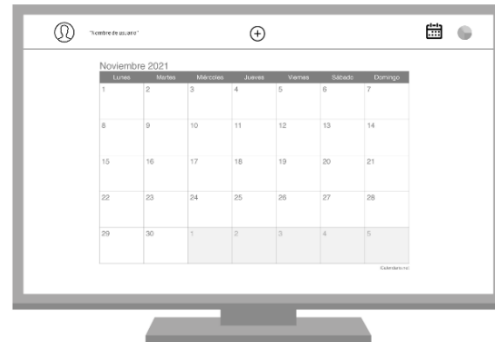


Imagen 8 – Prototipo (Samuel, Mockflow, 2022)

Inicio de sesión

Contaremos con un primer frame donde el usuario podrá iniciar sesión con sus datos.

Registro

Nos permitirá crear un usuario nuevo en la base de datos.

Panel principal

Mostrará una frase de motivación diaria y tendremos un menú con información del usuario.

Panel de creación de actividad

Panel donde el usuario introducirá información de la actividad para después subirla a la base de datos

Panel de calendario

Mostrará las actividades del usuario en su correspondiente día y con su correspondiente información

Panel de informes

Aquí podremos ver un informe del tiempo que el usuario le ha dedicado a cada categoría.

5. OBJETIVOS PROPUESTOS

Este proyecto pretende optimizar nuestro tiempo, organizando nuestras tareas de manera muy visual para tener siempre claro a qué lo estamos dedicando.

Partiendo de esa base hemos ido desarrollando funcionalidades como compartir actividades. Esta funcionalidad hace que cumplamos otro de nuestros objetivos, conseguir más cercanía social y que las personas puedan ayudarse entre sí o realizar actividades juntas.

Seiton también tiene otro objetivo a largo plazo y es crear una comunidad de personas que den feedback al equipo de desarrollo, haciéndoles ver qué funcionalidades tienen más potencial o enviando sugerencias de otras nuevas.

En resumen, crecer junto a nuestros usuarios es una de las mayores motivaciones que mueve este proyecto.

6. ANÁLISIS

Requisitos funcionales

- El usuario puede iniciar sesión con usuario y contraseña
- El usuario puede registrarse
- El usuario puede crear una tarea
- El usuario puede borrar una tarea
- El usuario puede ver una tarea
- El usuario puede responder a sus notificaciones
- El usuario puede actualizar el calendario
- El usuario puede cambiar el mes del calendario
- El usuario puede cambiar la imagen de perfil
- El usuario podrá actualizar el informe
- El usuario puede enviar un correo a soporte

Requisitos no funcionales

- La contraseña debe tener al menos 1 mayúscula
- La contraseña debe tener al menos 1 minúscula
- La contraseña debe tener al menos 1 símbolo
- La contraseña debe tener al menos 1 número
- La contraseña debe tener entre 8 y 16 caracteres
- La imagen de perfil debe pesar menos de 1 Mb
- La fecha de creación de la actividad no puede ser inferior al día actual

- Los días tienen un límite de 3 actividades

Diagramas de Casos de Uso

Inicio de sesión

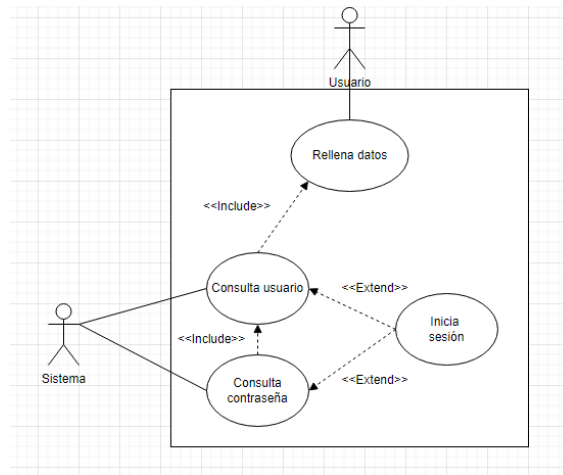


Imagen 9 - Caso de uso (Samuel, Draw.io, 2022)

Caso de uso 1	- Iniciar sesión
Descripción	- La persona podrá iniciar sesión
Actores	- Usuario - Sistema
Precondiciones	- La persona debe tener una cuenta registrada
Postcondiciones	- La persona habrá iniciado sesión
Escenario principal	1. Introduce usuario 2. Introduce contraseña 3. Pulsa iniciar sesión
Excepciones	3.1 Si el usuario no existe vuelve al punto 1 3.2 Si la contraseña no es correcta vuelve al punto 1
Comentarios	- Ningun comentario

Registro

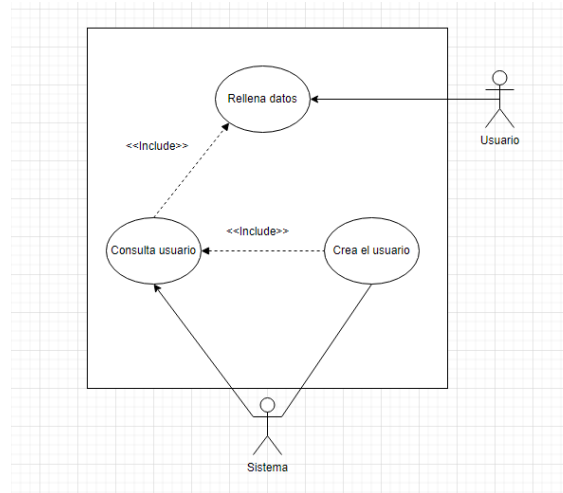


Imagen 10 - Caso de uso (Samuel, Draw.io, 2022)

Caso de uso 2	- Registro
Descripción	- La persona podrá registrarse
Actores	- Usuario - Sistema
Precondiciones	- La persona debe entrar en el panel de registro
Postcondiciones	- La persona se habrá registrado
Escenario principal	<ol style="list-style-type: none"> 1. Introduce Nombre 2. Introduce primer apellido 3. Introduce segundo apellido 4. Introduce contraseña 5. Introduce repetir contraseña 6. Introduce correo 7. Pulsa registrarse
Excepciones	<ol style="list-style-type: none"> 7.1 Si no se rellena todos los datos, vuelve al punto 1 7.2 Si el nombre de usuario existe, vuelve al punto 1 7.3 Si la contraseña no contiene mayúscula, vuelve al punto 4 7.4 Si la contraseña no contiene minúscula, vuelve al punto 4

	<p>7.5 Si la contraseña no contiene símbolo, vuelve al punto 4</p> <p>7.6 Si la contraseña no contiene número, vuelve al punto 4</p> <p>7.7 Si la contraseña no contiene entre 8 y 16 caracteres, vuelve al punto 4</p> <p>7.8 Si la contraseña repetida no coincide con la anterior, vuelve al punto 5</p>
Comentarios	- Ningun comentario

Crear actividad

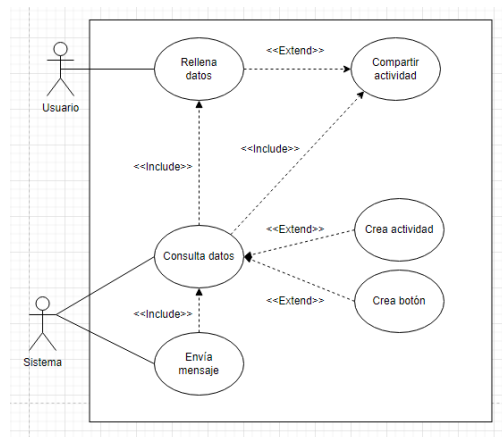


Imagen 11 - Caso de uso (Samuel, Draw.io, 2022)

Caso de uso 3	- Crear actividad
Descripción	- La persona podrá crear una actividad
Actores	- Usuario - Sistema
Precondiciones	- Debe existir otro usuario para compartir una actividad
Postcondiciones	- La persona habrá creado una actividad
Escenario principal	<ol style="list-style-type: none"> 1. Rellena nombre 2. Selecciona una fecha 3. Selecciona una hora 4. Selecciona con quién quiere compartir

	5. Selecciona la prioridad 6. Selecciona la categoría 7. Rellena el comentario 8. Pulsa añadir
Excepciones	8.1 Si el nombre no existe, vuelve al punto1. 8.2 Si la fecha es anterior a la actual, vuelve al punto 2. 8.3 Si la actividad es compartida, se consulta la disponibilidad del usuario para poder crear la actividad.
Comentarios	- Ningun comentario

Borrar actividad

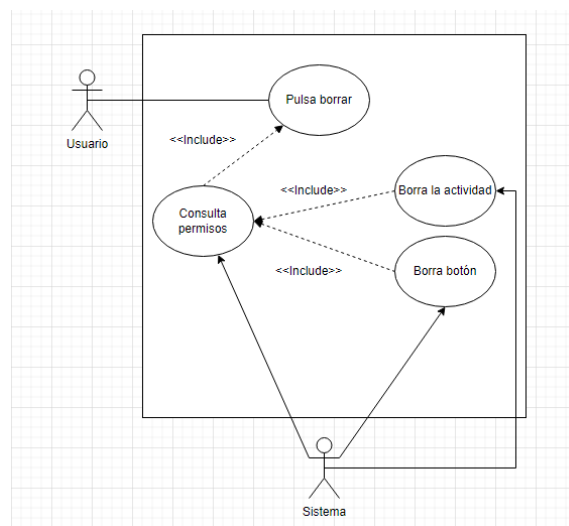


Imagen 12 - Caso de uso (Samuel, Draw.io, 2022)

Caso de uso 4	- Borrar actividad
Descripción	- La persona podrá borrar una actividad
Actores	- Usuario - Sistema
Precondiciones	- La persona debe tener al menos una actividad
Postcondiciones	- La persona habrá borrado la actividad
Escenario principal	1. Pulsa el botón de borrar actividad

	2. El sistema consulta los permisos 3. El sistema borra el boton 4. El sistema borra la actividad
Excepciones	2.1 Si el usuario no tiene permisos, vuelve al punto 1
Comentarios	- Ningun comentario

Ver actividad

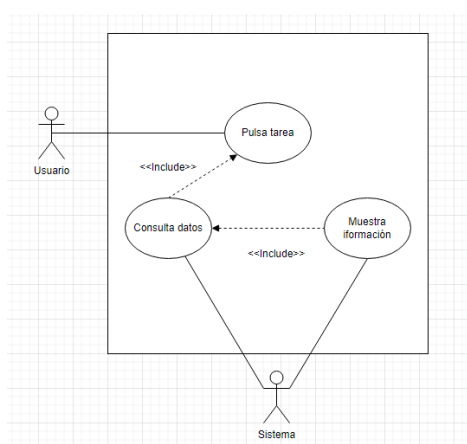


Imagen 13 - Caso de uso (Samuel, Draw.io, 2022)

Caso de uso 5	- Ver actividad
Descripción	- La persona podrá ver una actividad
Actores	- Usuario - Sistema
Precondiciones	- La persona debe tener al menos una actividad
Postcondiciones	- La persona habrá visto la actividad
Escenario principal	1. Pulsa el botón de la actividad 2. El sistema consulta los datos 3. El sistema muestra la información
Excepciones	
Comentarios	- Ningun comentario

Responder notificaciones

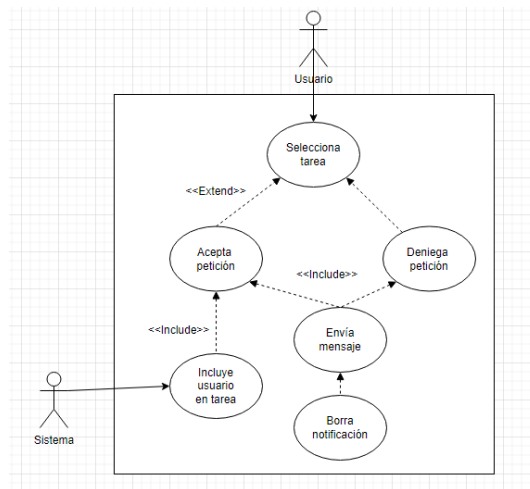


Imagen 14 - Caso de uso (Samuel, Draw.io, 2022)

Caso de uso 6	- Responder notificaciones
Descripción	- La persona podrá responder a sus notificaciones
Actores	- Usuario - Sistema
Precondiciones	- La persona debe tener al menos una notificación
Postcondiciones	- La persona ha respondido su notificación
Escenario principal	<ol style="list-style-type: none"> 1. El usuario selecciona la tarea 2. El usuario responde a la tarea con si 3. El sistema incluye al usuario en la tarea 4. El usuario responde a la tarea con no 5. Envía el mensaje a la persona que compartió la actividad 6. El sistema borra la notificación
Excepciones	
Comentarios	- Ningun comentario

Actualizar calendario

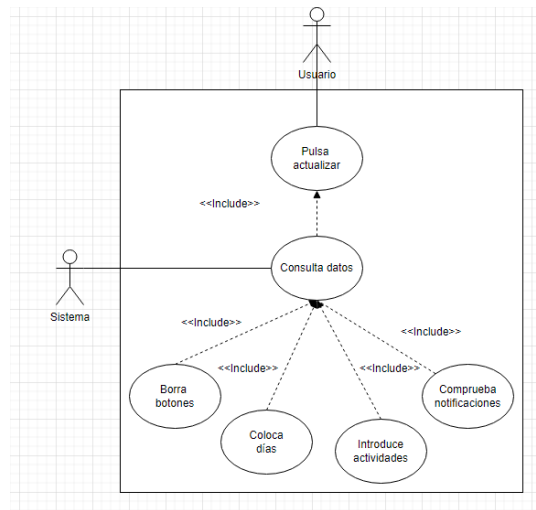


Imagen 15 - Caso de uso (Samuel, Draw.io, 2022)

Caso de uso 7	- Actualizar calendario
Descripción	- La persona podrá actualizar el calendario
Actores	- Usuario - Sistema
Precondiciones	
Postcondiciones	- La persona ha actualizado el calendario
Escenario principal	<ol style="list-style-type: none"> 1. El usuario pulsa el boton de actualizar 2. El sistema consulta los datos 3. El sistema borra las actividades actuales 4. El sistema coloca los días en los paneles 5. El sistema introduce las tareas actualizadas 6. El sistema actualiza las notificaciones
Excepciones	
Comentarios	- Ningun comentario

Actializar mes calendario

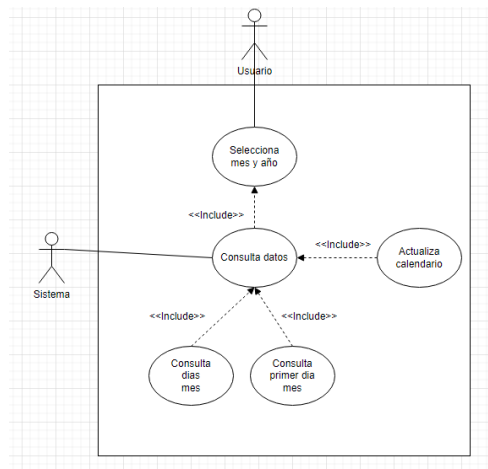


Imagen 16 - Caso de uso (Samuel, Draw.io, 2022)

Caso de uso 8	- Actualizar mes del calendario
Descripción	- La persona podrá actualizar el mes del calendario
Actores	- Usuario - Sistema
Precondiciones	
Postcondiciones	- La persona ha actualizado el mes del calendario
Escenario principal	<ol style="list-style-type: none"> 1. El selecciona el mes 2. El usuario selecciona el año 3. El usario pulsa el botón para buscar 4. El sistema consulta los datos con la fecha introducida por el usuario 5. El sistema borra las actividades actuales 6. El sistema coloca los días en los paneles 7. El sistema introduce las tareas actualizadas 8. El sistema actualiza las notificaciones
Excepciones	<ol style="list-style-type: none"> 3.1 Si el usuario no ha seleccionado el mes, vuelve al punto 1 3.2 Si el usuario no ha seleccionado el año, vuelve al punto 2

Comentarios

- Ningun comentario

Cambiar imagen

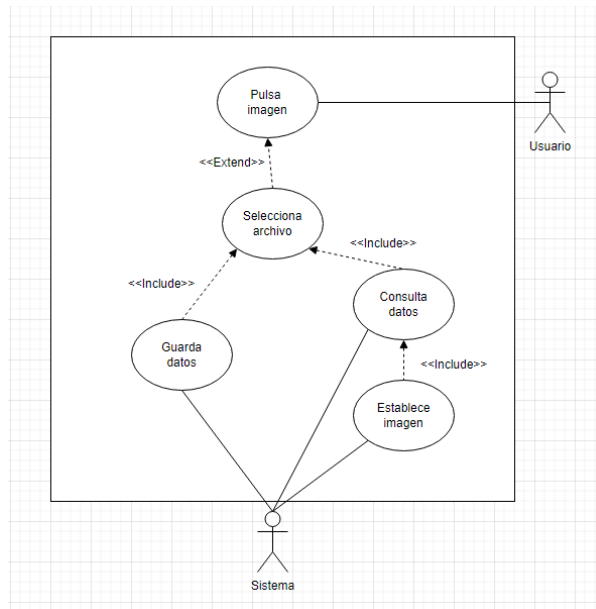


Imagen 17 - Caso de uso (Samuel, Draw.io, 2022)

Caso de uso 9	- Cambiar imagen de perfil
Descripción	- La persona podrá cambiar su imagen de perfil
Actores	- Usuario - Sistema
Precondiciones	- Tener la imagen que se quiere colocar
Postcondiciones	- La persona ha actualizado su foto de perfil
Escenario principal	<ol style="list-style-type: none"> 1. El usuario pulsa la imagen 2. El usuario selecciona la imagen de su dispositivo 3. El sistema guarda los datos en la bbdd 4. El sistema consulta los datos de la imagen 5. El sistema establece la imagen de perfil
Excepciones	3.1 La imagen no puede ser mayor a 1 mb
Comentarios	- Ningun comentario

Actualizar informe

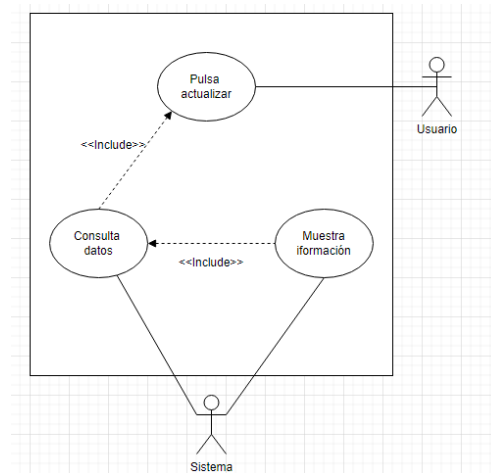


Imagen 18 - Caso de uso (Samuel, Draw.io, 2022)

Caso de uso 10	- Actualizar informe
Descripción	- La persona podrá actualizar su informe
Actores	- Usuario - Sistema
Precondiciones	- Que haya algún cambio desde la última vez que se actualizó
Postcondiciones	- La persona ha actualizado su informe
Escenario principal	<ol style="list-style-type: none"> 1. El usuario pulsa el botón actualizar 2. El sistema consulta los datos 3. El sistema muestra los datos
Excepciones	
Comentarios	- Ningun comentario

Enviar correo

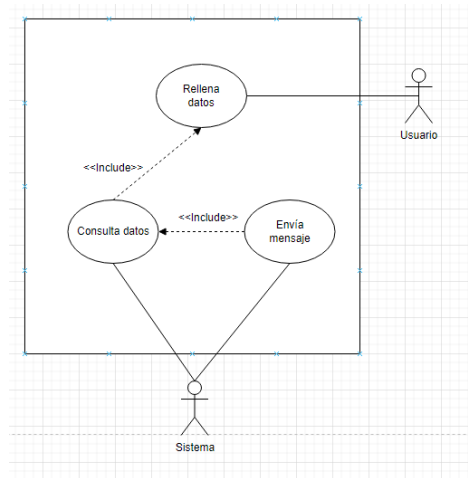


Imagen 19 - Caso de uso (Samuel, Draw.io, 2022)

Caso de uso 10	- Enviar mensaje
Descripción	- La persona podrá enviar un mensaje a la empresa
Actores	- Usuario - Sistema
Precondiciones	-
Postcondiciones	- La persona ha enviado un correo
Escenario principal	1. El usuario rellena los datos. 2. El sistema consulta los datos. 3. El sistema envía el mensaje.
Excepciones	2.1 Si los datos no son correctos, vuelve al punto 1
Comentarios	- Ningun comentario

Diagrama de clases

Debido a la extensión del diagrama de clases. Es conveniente visualizarlo en el anexo de [Diagrama de clases](#)

Diagramas de Entidad-Relación

Respecto a la base de datos del programa, se decidió hacer el menor número de consultas posibles así que se decidió traer la información de la base de datos al programa para después trabajar desde la información local.

Es por esto que las tablas no tienen relación, ya que no se necesita hacer consultas específicas.

Mf7qO3TN9q Frases	Mf7qO3TN9q Actividades	Mf7qO3TN9q Usuario
id : int(31)	id : int(11)	id : int(100)
Frase : varchar(500)	Nombre : varchar(30)	Nombre : varchar(20)
Autor : varchar(30)	Fecha_creacion : date	Primer_apellido : varchar(20)
	Fecha_realizacion : date	Segundo_apellido : varchar(20)
	Hora_realizacion : time(6)	Nombre_usuario : varchar(20)
	Prioridad : varchar(15)	Contraseña : varchar(20)
	Categoria : varchar(15)	Correo : text
	Comentario : varchar(250)	Imagen : longblob
	Usuario : varchar(20)	
	Compartido_estado : varchar(30)	
	Compartido_con : varchar(30)	
	Compartido_por : varchar(30)	

Imagen 20 - Entidad-Relación (Samuel, Phpmyadmin, 2022)

7. DISEÑO DEL DESARROLLO

En esta sección explicaremos a fondo las funcionalidades del aplicativo junto a sus clases y archivos utilizados.

Inicio de sesión

- El usuario debe introducir su usuario y contraseña.
- Los campos no pueden estar vacíos.
- Para el inicio de sesión el usuario debe estar creado en la base de datos.
- El texto del campo contraseña, debe coincidir con la contraseña del usuario guardada en la base de datos.
- Las mayúsculas **NO** influyen para el campo usuario. Es decir, el programa entiende lo mismo por "UsuArio" que por "usuario".
- En el campo contraseña, **SI** se diferencian mayúsculas y minúsculas
- En el campo contraseña tendremos un botón para mostrarla.

- Si los datos son correctos se pasa a la sección principal del programa

Registro

- El usuario debe introducir todos los datos, ningún campo puede estar vacío.
- El usuario podrá volver a la página de login en cualquier momento.
- El usuario no podrá pegar del portapapeles nada en los campos de contraseña
- Una vez pulsado el botón, el programa se asegurará de que no exista ninguna cuenta creada con el mismo nombre de usuario.
- Se evaluará si las contraseñas introducidas coinciden.
- Se evaluará si la contraseña cumple los requisitos(Mayúscula, minúscula, símbolo, numero y longitud).
- Si existe la cuenta, mostrará un mensaje informando de que el usuario ya está registrado
- Si no existe ninguna cuenta con ese nombre de usuario, se guardarán los datos introducidos por el usuario en la base de datos. El programa habrá creado el usuario.
- Cuando se crea el usuraio se setea los valores de los campos

Principal

- Hay un panel central donde se irá cambiando de secciones del programa (CardLayout).
- Barra de acción (Encontraremos información sobre el usuario y botones con interacciones)
- El usuario podrá actualizar su foto de perfil. El programa guardará la información en la base de datos y después escalará la imágen en el recuadro.
- Al hacer clic en la imagen, se abre el seleccionador de archivos para seleccionar la foto.
- El logotipo ubicado en el centro del menú sirve como botón para volver a la página principal.
- El boton con símbolo +, lleva al panel para crear una actividad.
- El boton con símbolo del calendario, lleva al panel del calendario.
- El boton con símbolo del gráfico, lleva al panel para ver los informes.

- El boton con símbolo del teléfono, lleva al panel del formulario de contacto.
- Cuando el usuario pase el ratón por encima de zonas de interacción, estos objetos modificarán ligeramente su estado.
- El primer panel contendrá la “frase del día”. Esté panel mostrará la frase pedida previamente a la base de datos e irá cambiando diariamente.

Panel de crear actividad

- Este panel permitirá crear una actividad.
- El usuario debe introducir todos los datos. Ningún campo puede estar vacío.
- Si existe la actividad, mostrará un mensaje informando de que esa actividad ya está registrada.
- Se podrá compartir una actividad con otro usuario
- Si se comparte, se verificará que el usuario de destino tenga hueco en su calendario para realizarla.
- Si no tiene hueco el programa lanza un aviso y no sería posible compartir ese día a esa hora.
- Cuando pulse el botón, el programa recogerá los datos y lo registrará en la base de datos.
- Si alguno de los datos son erróneos, se muestra un aviso por pantalla
- Si los datos son válidos, se manda automáticamente un coreo al destinatario comunicandole que se ha compartido una actividad con él

Panel de calendario

- Este panel mostrará las del usuario, tanto suyas como compartidas con él.
- El calendario será dinámico, automáticamente el programa averiguará los días que tiene el mes seleccionado y el día en el que empieza.
- En la parte inferior derecha se encuentra dos desplegables para seleccionar el mes y el año que queramos ver.
- Al pulsar la lupa, el programa volverá a consultar los días para colocarlos en los paneles a demás de los registros de la base de datos para las actividades.
- Cada panel podrá guardar un máximo de 3 actividades
- Cada botón es una actividad
- Si la actividad es comparida, se introducirá un icono con 2 personas.

- Si la actividad está a la espera de que se responda, se introducirá un icono con un reloj.
- El botón actualizar vuelve a hacer una consulta a la base de datos y reintroduce las actividades teniendo en cuenta el día actual.
- Al pulsar un botón se muestra un panel con la información detallada de la actividad.
- El panel contiene un botón para borrar dicha actividad de la base de datos
- Solo el usuario que haya creado la actividad, podrá borrarla. La persona destinataria no tiene permisos.
- En la parte inferior izquierda, encontraremos un botón con el icono de las dos personas. Al pulsar ese botón, se abre una ventana donde parecerán las notificaciones.
- El botón tendrá el número de notificaciones que tiene el usuario
- Dentro de las tareas compartidas, encontraremos una lista que se consultará a la base de datos con las tareas y el usuario que la ha compartido
- El usuario seleccionará una tarea y se activarán los botones de confirmación.
- En caso afirmativo, se creará la tarea en el calendario del usuario y se comunicará a la persona que compartió la actividad.
- En caso negativo, solo se notificará al usuario que compartió la tarea.
- Una vez se responda, la tarea desaparece de la lista de notificaciones.
- Para volver al calendario se pulsará la flecha hacia atrás, situada en la parte superior izquierda de la ventana.
- Las notificaciones se actualizan cada 10 segundos aunque se puede actualizar manualmente con el botón “Actualizar”

Panel del gráfico circular

- Este panel muestra la cantidad de horas que el usuario ha dedicado a cada categoría en forma de gráfica circular.
- El botón actualizar, actualiza los registros y devuelve un nuevo informe actualizado.
- Muestra también un informe con el recuento de horas.

Panel de contacto

- Panel para contactar con la empresa

- La información referente al usuario (nombre, apellidos y correo) se autorellenará, aunque el usuario puede cambiar los datos.
- El usuario tendrá que elegir el asunto del mensaje ya que irá destinado a un correo u a otro.
- Si el usuario deja en blanco el panel comentario, no se enviará el mensaje.
- Si la información es correcta se enviará el mensaje al departamento correspondiente

Clase “logica”

La clase “logica” se ha creado para ahorrar trabajo al desarrollador ya que hay código que repite muchas veces. La solución es crear un método con dichos códigos y hacerlos públicos para poder acceder a ellos.

- Función para consultar una imagen del host y setearla en un label
- Función para consultar una imagen del host y setearla en un boton
- Función para consultar una imagen de la base de datos y setearla en un label
- Función para escalar la imagen respetando la relación de aspecto
- Función que devuelve el nombre del mes pasado por parámetro
- Función que devuelve el nombre del mes actual
- Función que devuelve el número total de días del mes pasado por parámetro
- Función que devuelve el primer día del mes y año pasado por parámetro
- Función para enviar mensajes
- Función que hace de biblioteca, se le pasa una opción y te devuelve el mensaje que corresponde.

Clase “conexion”

La clase “conexion” se utiliza para realizar operaciones relacionadas con la base de datos, ya sea consultas, actualización o eliminación de registros, etc.

- Función para conectar con la bbdd.
- Función para desconectar de la bbdd.
- Consulta para sacar 1 solo valor.
- Consulta para sacar varios valores.
- Consulta para sacar el valor binario de la imagen.
- Función para actualizar o eliminar.

Hilo “HiloEnvíaMensaje”

Esta clase se utilizará como optimización de tiempo para el programa. A la hora de enviar un mensaje, el tiempo de espera del proceso ronda los 15 segundos.

Esto hace que se relentice mucho el proceso de compartir tareas y enviar el formulario de contacto. Debido a esto, se crea un hilo en segundo plano que lo envía mientras el usuario puede seguir realizando otras gestiones.

- Se le pasa la información por parámetros para configurar el envío
- Se conecta con el servidor (correo) con el usuario y contraseña pasados
- Se envía el mensaje pasado por parámetros a la persona destinataria

Hilo “NotificacionTareas”

Al igual que la anterior, esta clase se utiliza para que el rendimiento del programa sea más óptimo. Para ello deja en segundo plano los procesos para actualizar la notificación de tareas cada 10 segundos.

8. DESPLIEGUE Y PRUEBAS

En esta sección se llevará a cabo las pruebas de caja negra de los requisitos funcionales del aplicativo.

Funcionalidad	Especificación de pruebas
Iniciar sesión	<p>Objetivo: Iniciar sesión</p> <p>Pruebas:</p> <ul style="list-style-type: none"> • Introducir mal el usuario • Introducir mal la contraseña • Pegar contraseña del portapapeles <p>Conclusión: Funcionalidad correcta</p>
Registro	<p>Objetivo: Registrarse</p> <p>Pruebas:</p> <ul style="list-style-type: none"> • No introducir todos los datos • Introducir contraseña sin mayúscula

	<ul style="list-style-type: none"> • Introducir contraseña sin minúscula • Introducir contraseña sin símbolo • Introducir contraseña sin número • Introducir contraseña menos de 8 caracteres • Introducir contraseña mayor de 16 caracteres • Introducir contraseña repetida mal <p>Conclusión: Funcionalidad correcta</p>
Crear actividad	<p>Objetivo: Crear actividad</p> <p>Pruebas:</p> <ul style="list-style-type: none"> • No rellenar el nombre • Rellenar el nombre con un nombre existente • Poner fecha anterior a la actual • Compartir actividad con alguien que no tenga disponibilidad <p>Conclusión: Funcionalidad correcta</p>
Borrar actividad	<p>Objetivo: Borrar actividad</p> <p>Pruebas:</p> <ul style="list-style-type: none"> • Borrar actividad sin tener permisos <p>Conclusión: Funcionalidad correcta</p>
Ver actividad	<p>Objetivo: Ver actividad</p> <p>Pruebas:</p> <ul style="list-style-type: none"> • Pulsar sobre la actividad <p>Conclusión: Funcionalidad correcta</p>
Responder notificaciones	<p>Objetivo: Ver actividad</p> <p>Pruebas:</p> <ul style="list-style-type: none"> • Seleccionar 1 actividad y aceptar • Seleccionar 1 actividad y denegar • Seleccionar varias actividades y responder

	Conclusión: Funcionalidad incorrecta parcialmente
Actualizar calendario	Objetivo: Actualizar el calendario Pruebas: <ul style="list-style-type: none">• Pulsar el botón teniendo tareas propias• Pulsar el botón teniendo tareas en espera• Pulsar el botón teniendo tareas compartidas• Pulsar botón en otro mes diferente Conclusión: Funcionalidad correcta
Cambiar mes	Objetivo: Actualizar el calendario con el mes que introduzca el usuario. Pruebas: <ul style="list-style-type: none">• Poner diferentes días• Poner diferentes meses Conclusión: Funcionalidad correcta
Cambiar imagen	Objetivo: Actualizar la foto de perfil Pruebas: <ul style="list-style-type: none">• Seleccionar un archivo que no sea imagen• Seleccionar un archivo mayor a 1mb Conclusión: Funcionalidad incorrecta parcialmente
Actualizar informe	Objetivo: Actualizar el informe de horas Pruebas: <ul style="list-style-type: none">• Añadir actividades en diferentes categorías y pulsar botón Conclusión: Funcionalidad correcta
Enviar mensaje	Objetivo: Enviar mensaje a soporte Pruebas: <ul style="list-style-type: none">• No rellenar los datos

	<ul style="list-style-type: none">• Rellenar mal el correo <p>Conclusión: Funcionalidad correcta. Si no tiene su correo bien puesto, simplemente no le llegará los mensajes.</p>
--	---

9. METODOLOGÍAS UTILIZADAS

Primeras ideas

Respecto a la captación de ideas, se utilizó la metodología de **Brain storming** para empezar a soltar ideas. Se hizo una lista con posibles ideas, en principio tanto ideas sencillas como ideas complejas.

Después se fue clasificando para así tener un control de qué funcionalidades si se podían implementar y cuáles no.

Forma de trabajo

Analizando el proyecto, se decidió que la metodología más adecuada para llevar a cabo era Scrum.

La razón de esta decisión se debe a que el tiempo de desarrollo era limitado y el equipo se quería asegurar de que, llegado el día, se tuviera un aplicativo funcional.

De esta forma, completando sprints, nos asegurábamos la entrega ya que si se disponía de más tiempo se seguían haciendo nuevos sprints y de lo contrario se dejaría con el último sprint estable. Aquí podemos ver el resultado de la organización

Sesión 1 (10/03/2022)

Desarrollo del aplicativo

En el primer Sprint el equipo ha trabajado en los siguientes puntos.

- Base de datos
- Hosting para almacenamiento de imágenes del programa.
- Creación de la tabla “usuarios” en la base de datos.
- Creación del logotipo del aplicativo.
- Almacenamiento de imágenes relativas al proyecto en el hosting

- Módulo de inicio de sesión
 - Consulta a base de datos.
 - Mostrar y ocultar contraseña.
- Módulo de registro
 - Contraseña de 8-16 caracteres, minúscula, mayúscula y símbolo.
 - Las contraseñas deben coincidir

Conclusión

La progresión del trabajo es la esperada. El equipo está satisfecho con la evolución del proyecto y nos surge una nueva funcionalidad en el módulo del registro de usuarios:

- El usuario no debe pegar desde el portapapeles en los campos de contraseña.

El equipo ve oportuno incluir la funcionalidad porque la finalidad de repetir la contraseña es que el usuario se asegure de que ha introducido las 2 correctamente.

Sesión 2 (20/03/2022)

Desarrollo del aplicativo

En el segundo Sprint el equipo ha trabajado en los siguientes puntos.

- Función para que el usuario no pueda pegar desde el portapapeles en los campos de contraseña.
- Función para que no pueda registrarse un usuario que ya tenga cuenta.
- Frase del día. Alojada en la base de datos.
- Mostrar nombre, apellidos y nombre del usuario que haya iniciado sesión.
- Cambio de paneles mediante CardLayout.
- Animación de 'botones'.

Conclusión

El equipo llega a la conclusión de que se está desarrollando el proyecto al ritmo acordado.

Sesión 3 (1/04/2022)

Desarrollo del aplicativo

En el tercer Sprint el equipo ha trabajado en los siguientes puntos del módulo de “Crear actividad” y calendario.

- Implementar dependencia de componente reloj.
- Evaluar que todos los campos están rellenos correctamente.
- Crear actividad (Subir a la base de datos)
- Cuando el usuario pase el ratón por encima de zonas interacción, estos objetos modificarán ligeramente su estado.
- Creación de paneles, labels y organización de los layouts,
- Función que introduce los números dentro del calendario
- Función que introduce las tareas en el calendario.
- Optimización de código. La clase conexión tenía 295 líneas y se ha quedado en 96.

Conclusión

El equipo ha tenido que reunirse debido a la situación, el aplicativo no va a poder incorporar todas las funcionalidades previstas.

Se ha llegado a la conclusión de que, en vez de hacer un calendario dinámico, se desarrolle de manera estática para el mes de junio de 2021.

Sesión 4 (2/05/2022)

Desarrollo del aplicativo

En el cuarto Sprint el equipo ha trabajado en los siguientes puntos del módulo de “Crea actividad”, “Calendario” y “Gráfico”.

- Recuperar y guardar actividades de la base de datos en el programa.
- Validar el ingreso de datos en el cajón del tiempo estimado.
- Introducir botones dinámicos dentro un día.
- Crear gráfico circular.
- Hacer cálculo de los datos respecto de las actividades por categoría.
- Sincronización de tareas con la base de datos.

Cambios del proyecto

- Las actividades duran 2 horas.

- La prioridad la muestra con colores, pero solo ordena por fecha de entrega.
- El tiempo estimado se utilizará para la siguiente versión (Seiton 2.0)
- La fecha de entrega se utilizará para la siguiente versión (Seiton 2.0)
- La hora límite se utilizará para la siguiente versión (Seiton 2.0)

Conclusión

El equipo ve conveniente dejar ciertas funcionalidades para actualizaciones futuras porque no hay tiempo suficiente para desarrollarlas

Sesión 5 (9/05/2022)

Desarrollo del aplicativo

En el quinto Sprint el equipo ha trabajado en todos los módulos ya que, por seguridad, se ha revisado y asegurado del funcionamiento de todas las funcionalidades del aplicativo.

- Botón para actualizar el calendario
- Botón para actualizar el gráfico
- Mostrar ventana de información al pulsar una actividad.
- Adaptar los días para el mes de diciembre
- Optimización de código
- Revisión del proyecto completo

Conclusión

Llegados a este punto, el equipo considera que se debe hacer una versión estable por lo que se revisa el código y la funcionalidad del proyecto.

Sesión 6 (16/05/2022)

Situación

En el sexto Sprint, el equipo se ha vuelto a reunir para analizar el tiempo que queda hasta la entrega y decidir de forma objetiva qué funcionalidades pueden incluirse dentro del plazo.

El equipo se da cuenta que la lógica de introducción de tareas en el calendario va a costar demasiado tiempo y, aunque se puede terminar, quedaría un proyecto pobre en cuanto a funcionalidades.

Al final se decide que las tareas no las va a organizar el propio programa sino que es la persona la que elige el día y la hora para realizarla.

Con esto el equipo puede dedicar el tiempo a las nuevas funcionalidades.

Sesión 7 (23/05/2022)

Desarrollo del aplicativo

En el séptimo Sprint, el equipo ha trabajado en el código general debido a la situación descrita en la sesión número 6.

- Eliminación de código innecesario
- Reorganización de código válido

Sesión 8 (30/05/2022)

Desarrollo del aplicativo

En el octavo Sprint, el equipo ha trabajado sobre el panel del calendario.

- Función que devuelve el primer día del mes pasado por parámetro
- Función que devuelve el nombre del mes del número pasado por parámetro
- Función que pone los días en los paneles correspondientes
- Diseño de frame
- Insertar actividad en un día concreto
- Insertar actividad a una hora concreta
- Función que sube una imagen a la base de datos en formato binario
- Función que escala la imagen respetando la relación de aspecto
- Función que consulta los datos de la imagen y la coloca en un label

Sesión 9 (3/06/2022)

Desarrollo del aplicativo

En el noveno Sprint, el equipo ha trabajado sobre varios paneles para la implementación de las tareas compartidas.

- Creación de nuevos campos en el registro del usuario (correo)
- Creación de nuevos campos en el panel de creación de la actividad
 - Función para mostrar un desplegable con los nombre de usuarios
- Modificación de funciones relacionadas con la base de datos
- Función para comprobar que se puede compartir la actividad.
- Función que envía un mensaje al usuario con el que se comparte la actividad
- Boton que abra una pestaña con todas las notificaciones de tareas compartidas
- Creación de mensajes personalizados
- Función en la parte del destinatario para que acepte o rechace la actividad
- Función que añada iconos a las actividades dependiendo de su estado
- Función que verifique que se pueda borrar actividad
- Creación de la nueva funcionalidad Borrar actividad
- Diseño del panel

Conclusión

En el Sprint numero 9, el equipo ha desarrollado una de las funcionalidades más complejas y quiere ir revisando funcionalidades y encaminar la recta final del proyecto.

Sesión 10 (10/06/2022)

Desarrollo del aplicativo

En el duodécimo Sprint, el equipo ha trabajado sobre varias clases.

- Creación de un nuevo panel para la sección de contactos
- Diseño del panel
- Autorelleno de datos
- Creación de mensaje personalizado
- Optimización de código en la clase Conexión, PanelCalendario y TareasPendienes
- Revisión de código innecesario.
- Revisar funcionalidades.
- Hacer pruebas.
- Revisión del diseño de los paneles y frames
- Preparación de usuarios de cara a la presentación

Conclusión

El equipo ve correcto los cambios aplicados durante el desarrollo del proyecto debido a los contratiempos.

10. ESTIMACIÓN DE RECURSOS Y PLANIFICACIÓN

Una vez visto los ciclos por los que ha pasado el proyecto, se muestra el diagrama de Gantt inicial en comparación con el resultado final para analizarlos y sacar conclusiones de mejoras. Mire el anexo [Diagrama de Gantt](#)

11. TECNOLOGÍAS Y HERRAMIENTAS UTILIZADAS

Hostings

Respecto a las herramientas externas, teniendo en cuenta que no había presupuesto, se ha optado por buscar **servidores online gratuitos**. Concretamente se han elegido estos porque cumplen con los requisitos de **capacidad** y **procesamiento** para este proyecto.

Base de datos: Remotemysql.com

Almacenamiento de imágenes: 000webhost.com

Desarrollo

En cuanto al desarrollo, he utilizado el lenguaje que considero que mejor domino, mejor me siento programando y con el que he pensado que podía hacer un buen proyecto.

Como herramientas he utilizao **Netbeans** para el desarrollo y **PHPmyadmin** para guardar los datos porque son las herramientas que mejor domino.

Como lenguajes de programación, he utilizado **Java Swing** porque quería hacer un proyecto con elementos visuales y sus librerías me permitían hacerlo. Por otra parte he utilizado **MySQL** por su facilidad de uso, su **disponibilidad** en los proveedores de hosting y por su eficiencia

Edición de imágenes

Para la edición de imágenes he utilizado **Gimp**, he trabajado en otras ocasiones con este programa y tiene mucho potencial, a demás de ser gratuito.

12. VIAS FUTURAS

Inicio de sesión

- Barra de carga al iniciar sesión
- Inicio de sesión con faceID
- Confirmación móvil para el inicio de sesión

Principal

- Poder ajustar la imagen de perfil aumentando o disminuyendo la imagen.

Crear actividad

- Lista de horas completas de un día para insertar la actividad.
- Poder crear categorías
- Poder añadir a más personas en una tarea
- Dar permisos para borrar actividades a los destinatarios

Calendario

- Comprobar que la actividad haya terminado dependiendo del día y la hora.
- Pintar de otro color los paneles que no se estén utilizando.
- Poder configurar la actividad para que sea tipo Kanban (por hacer, en proceso, hecho)
- Eliminar personas de una tarea
- Hacer que las actividades se puedan arrastrar
- Al pulsar un día, que despliegue una ventana con todas las horas del día y sus respectivas actividades.

Informes

- Estadísticas semanales.
- Estadísticas mensuales

- Estadísticas anuales.
- Descargar report en PDF
- Crear avisos cuando se supere un límite de horas establecido por el usuario

Panel Contacto

- Crear un chat para una comunicación más cercana y rápida

General

- Desarrollar la parte front con Angular o cualquier otro lenguaje que permita darle una mejor vista. Se realizaría una integración Java + Angular.
- Diseño del correo de notificación más profesional
- Los datos se enviarán cifrados a la base de datos.
- Crear animaciones en los objetos

13. CONCLUSIONES

Lecciones aprendidas

- Ser más objetivo y previsible. No se puede hacer un proyecto de X meses en la mitad de tiempo teniendo en cuenta, además, las tareas ajenas al proyecto.
- No dejar de lado o quitarles importancia a tareas simples, aunque quede mucho tiempo para la entrega.
- Crear un diagrama de clases lo más completo posible antes de empezar a desarrollar. Esto ayuda a tener más clara la idea y el camino que se debe seguir.
- Tener un informe diario con las cosas que se han trabajado. El siguiente día que se trabaje, tienes toda la documentación para saber qué se hizo en sesiones pasadas y por dónde seguir.
- Las tareas “sencillas”, no siempre son sencillas. Puede dificultarse y ocupar varias horas de trabajo.

Dificultades/errores

- En principio el proyecto se enfocaba, sobre todo, en la inteligencia artificial que optimizaba el tiempo de las personas ordenando sus tareas. Después de haber empezado y haber visto la complejidad, el proyecto se va a centrar más en enriquecer el aplicativo con más herramientas.

- He llegado a la conclusión de que es mejor empezar por crear tareas fijas y a medida que se vaya desarrollando el aplicativo, seguir con las tareas dinámicas. Como su nombre dice, las tareas fijas son estáticas, es decir, van a estar en una posición concreta siempre. Sin embargo, las tareas dinámicas me estaban dando muchos comederos de cabeza.
- En el layout del calendario, había un botón que actualizaba las actividades. El problema era que tardaba mucho en actualizar y pensaba que era por la base de datos. Haciendo pruebas me he dado cuenta que el proceso lo hacía, el problema era que la interfaz se actualizaba cada x segundos. La solución ha sido hacer un update de la interfaz justo cuando hago el proceso de actualización.
- El .equals se queda procesando ciertos procesos y deja el programa "pausado". He tenido que utilizar la comparación "!=" para solucionarlo.
- A la hora de enviar un correo tardaba sobre 30 o 40 seg en enviar y para evitar que el usuario pierda tanto tiempo, cree un hilo para que se enviara en segundo plano.
- Cuando el usuario seleccionaba el componente jlist devolvía null porque el foco estaba en el componente completo y no en la opción correspondiente. He capturado las excepciones para controlar el problema.

Opinión personal

Respecto a mi opinión personal del trabajo, considero que para el tiempo que he tenido, junto a mis conocimientos de estudiante y los contratiempos, considero que he hecho un trabajo apto.

Personalmente me hubiera gustado poder dedicarle mucho más tiempo porque de verdad creo en la funcionalidad y capacidad del programa. Es por ello que seguiré desarrollándolo e implementando todas las funcionalidades señaladas en vías futuras a demás, de nuevas que vayan surgiendo.

14. BIBLIOGRAFÍA/WEBGRAFÍA

- (W3schools, 2022)
- (Stackverflow, 2022)
- (Delftstack, 2022)
- (Oracle, 2022)
- (Lawebdelprogramador, 2022)

15. ANEXOS

Diagrama de Gantt - Estimación



Imagen 21 - Diagrama de Gantt (Samuel, Canva, 2022)

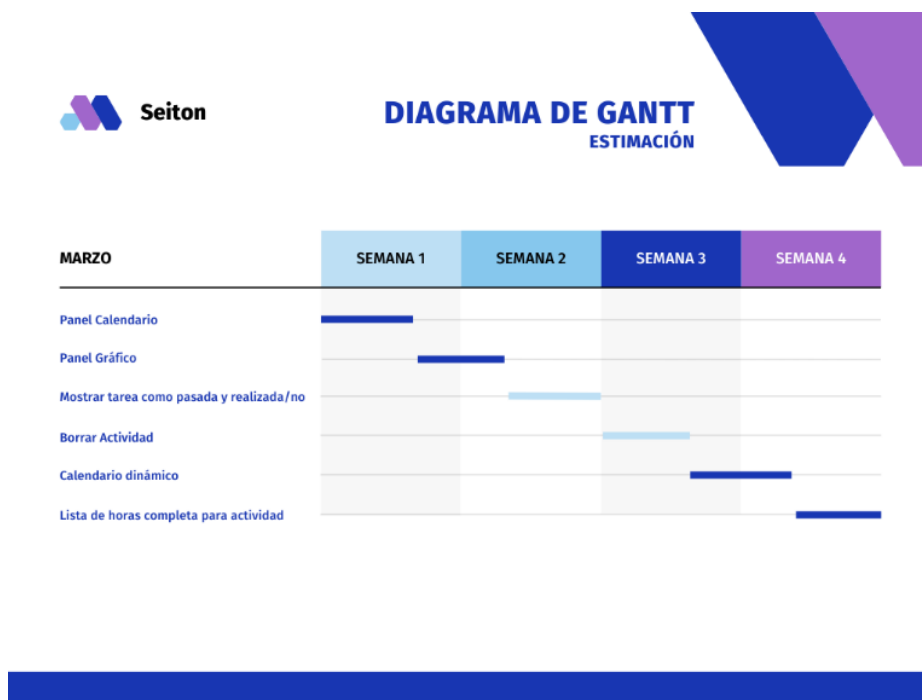


Imagen 22 - Diagrama de Gantt (Samuel, Canva, 2022)



Imagen 23 - Diagrama de Gantt (Samuel, Canva, 2022)

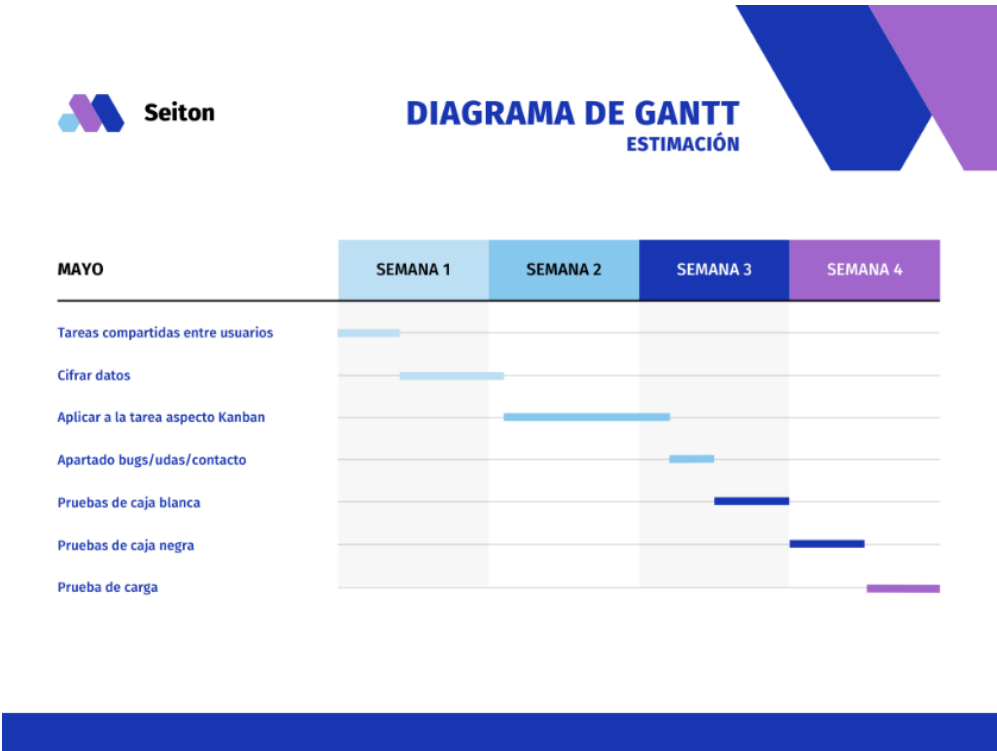


Imagen 24 - Diagrama de Gantt (Samuel, Canva, 2022)

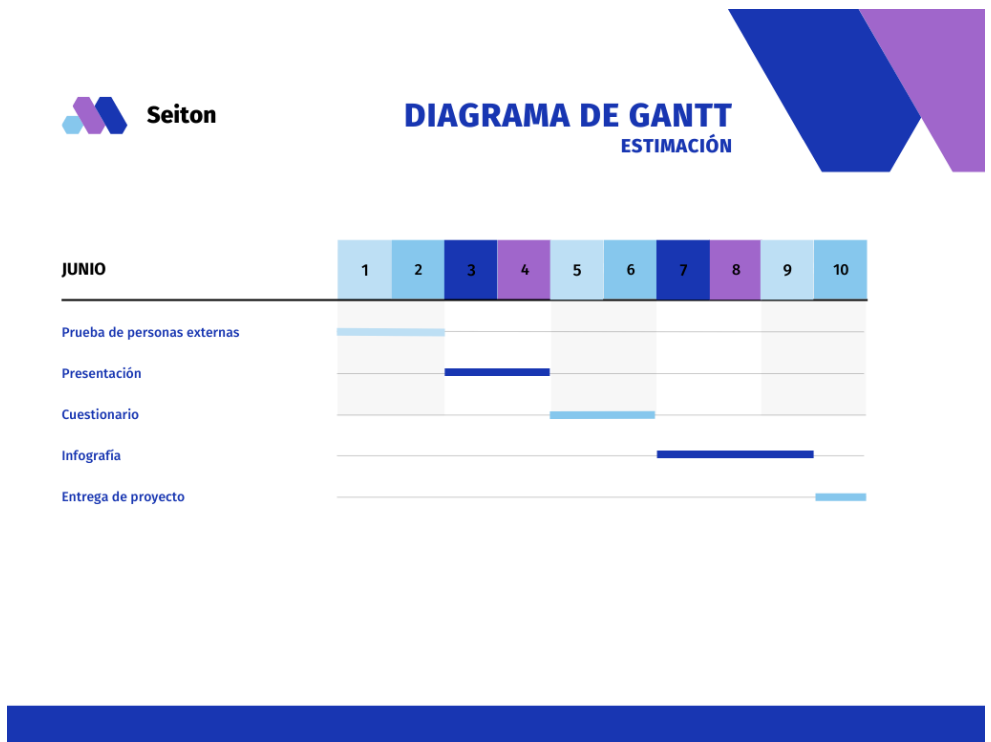


Imagen 25 – Diagrama de Gantt (Samuel, Canva, 2022)

Diagrama de Gantt – Conclusión

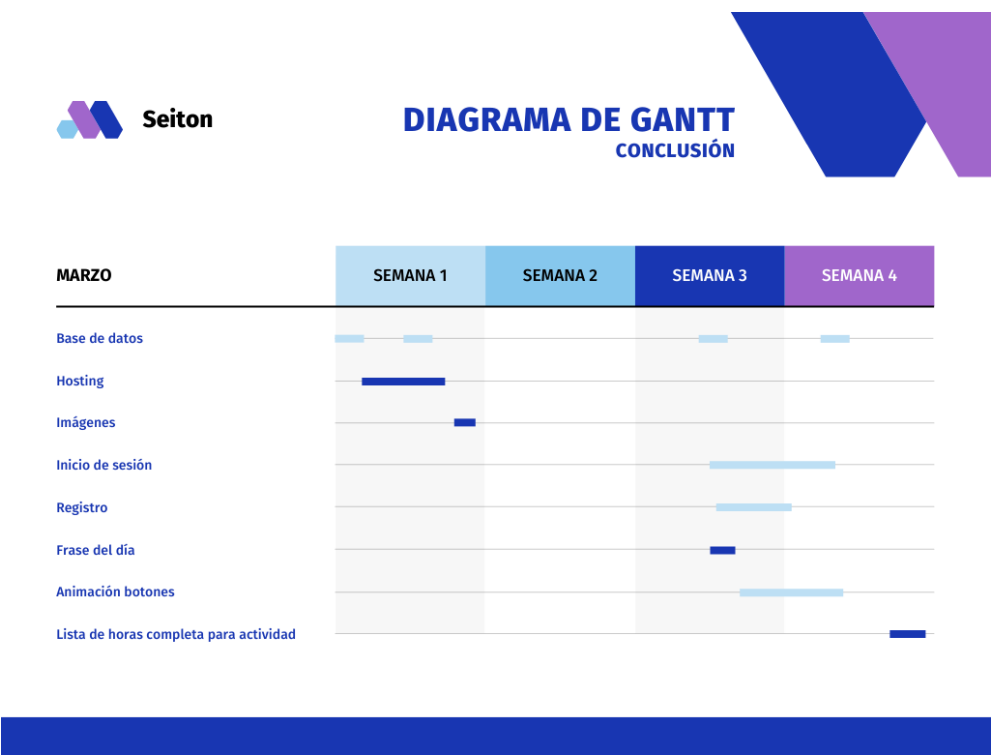


Imagen 26 - Diagrama de Gantt (Samuel, Canva, 2022)

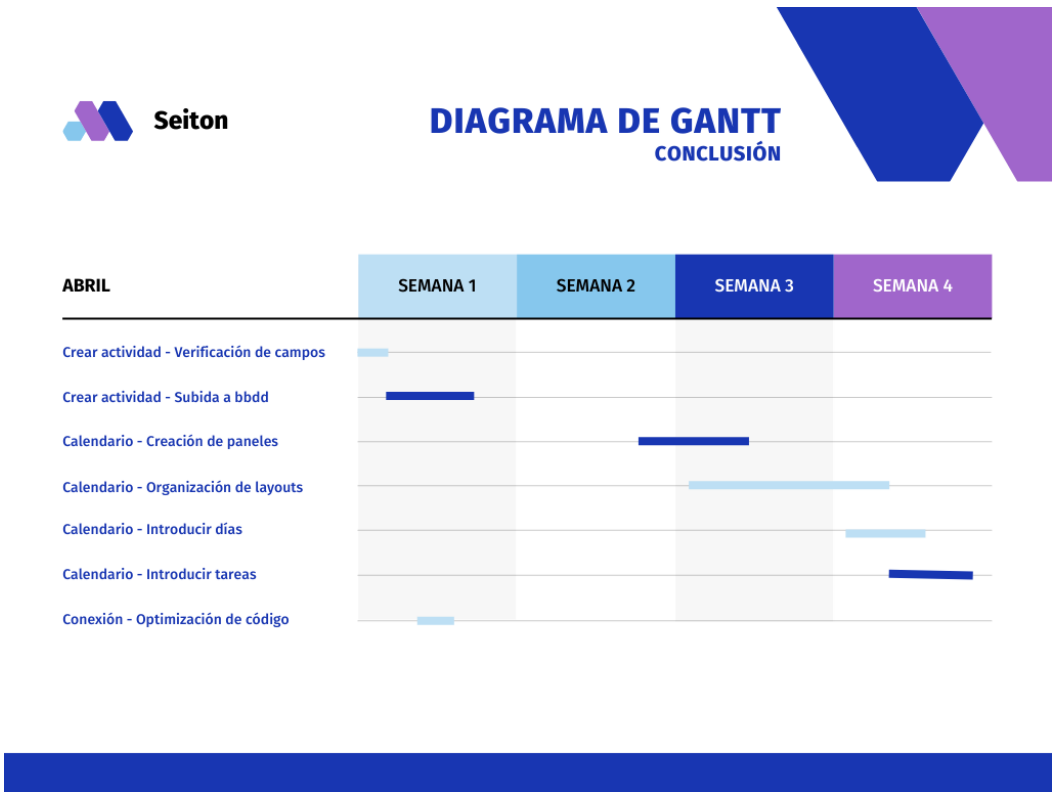


Imagen 27 - Diagrama de Gantt (Samuel, Canva, 2022)

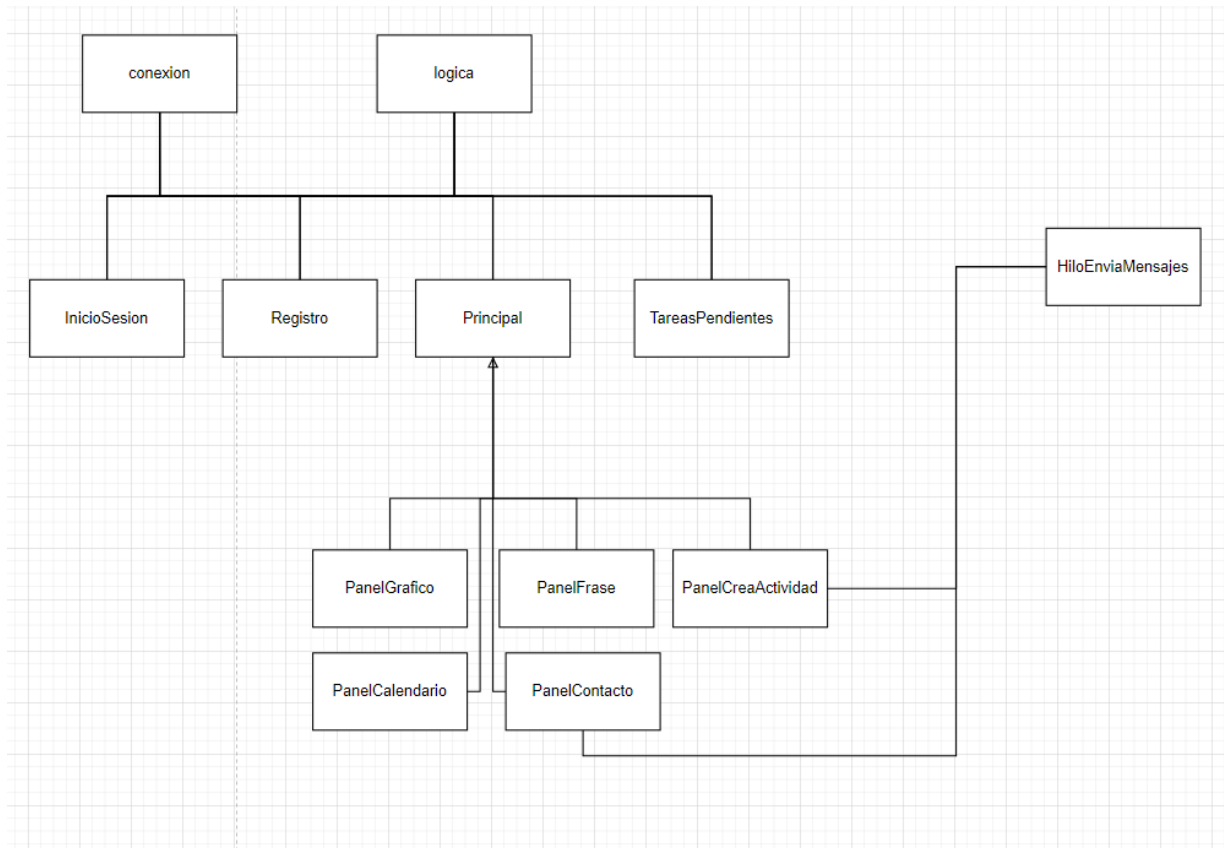


Imagen 28 - Diagrama de Gantt (Samuel, Canva, 2022)



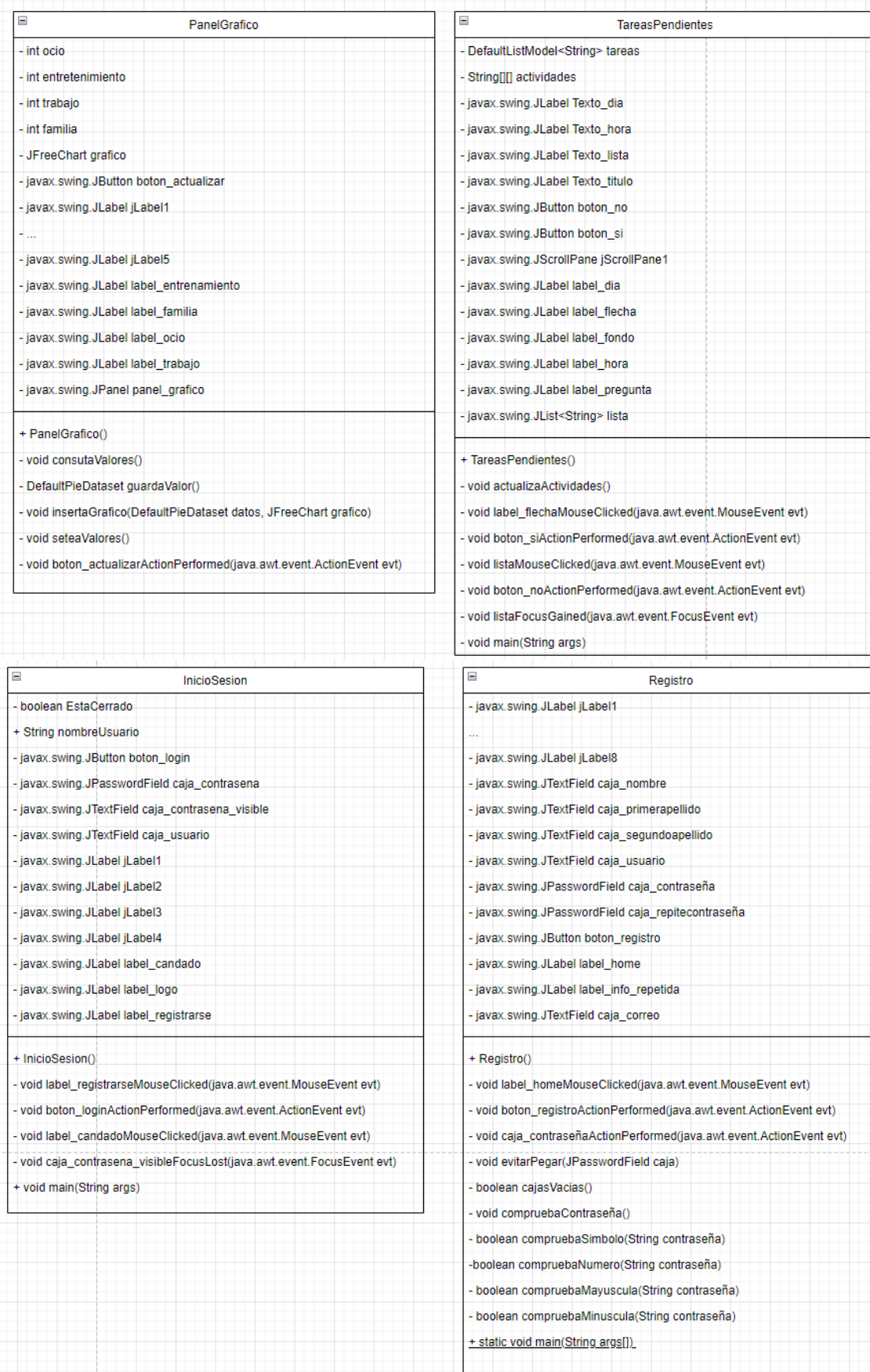
Imagen 29 - Diagrama de Gantt (Samuel, Canva, 2022)

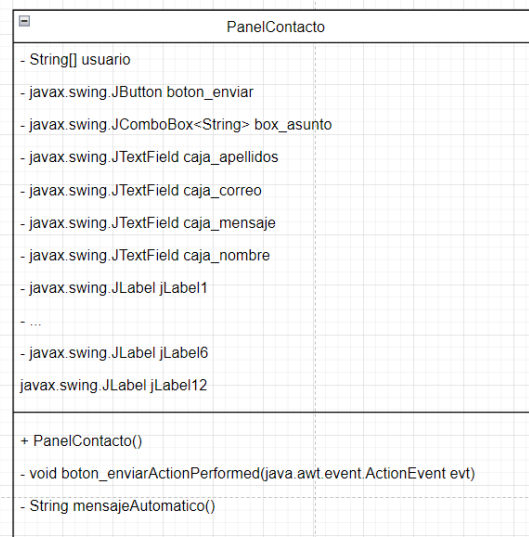
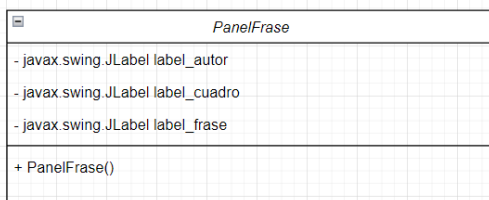
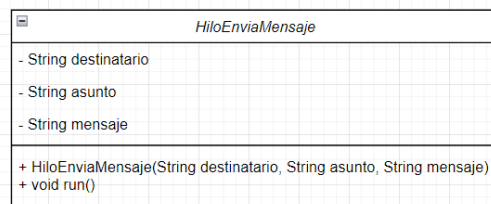
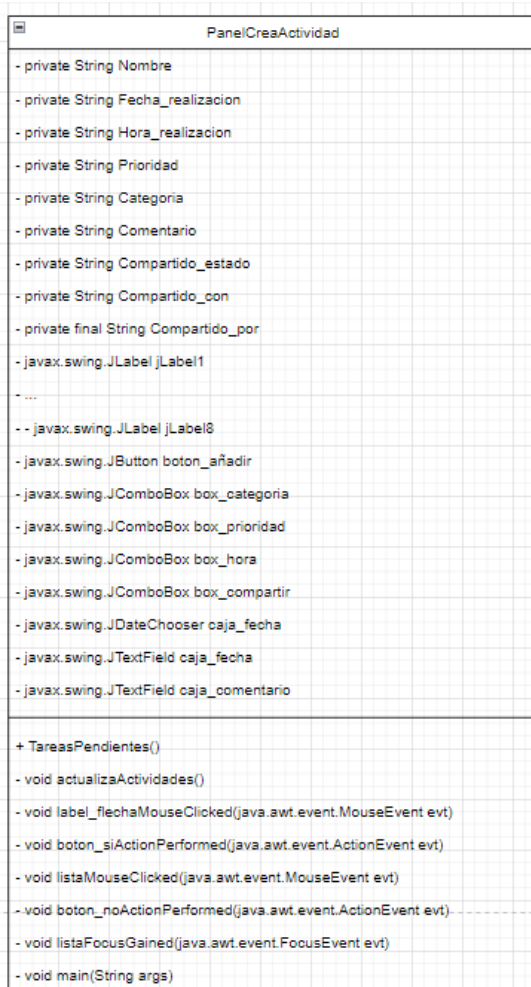
Diagrama de clases



Conexion
- Connection con - String driver - String user - String pass - String url - Statement st
+ void conectar() + void desconectar() + String consultaValor(String valorBuscado, String consulta) + String[][] consultaTareasEsperando(String usuario) + String[][] consultaActividadesCompartidas(int ano, int mes, String usuario) + String[][] consultaActividadesMes(int ano, int mes, String usuario) + String[] consultaValoresUsuario(String usuario) + String[] consultaUsuarios() + void update_delete(String consulta) + void subelImagen(String ruta) + Image consultalImagen()

logica
+ void ponelImagenBoton(JButton boton, String ruta, String opcion) + Dimension escalalImagen(Dimension dim_original, Dimension dim_final) + void ponelImagenLabel(JLabel cuadro, String ruta) + void ponelImagenLabelBBDD(JLabel cuadro, Image imagen) + String devuelveMes(int mes) + String devuelveMesAcutal() + int numeroDiasMes(int mes) + int primerDiaMes(int mes, int ano) + String devuelveHoy() + void enviaGmail(String destinatario, String asunto, String cuerpo) + String mensajes(String destinatario, String tarea, String opcion)





Principal
- String usuario
- Conexion conexion
- PanelFrase panel_frase
- PanelCreaActividad panel_creaActividad
- PanelCalendario panel_calendario
- PanelGrafico panel_grafico
- PanelContacto panel_contacto
- CardLayout card
- javax.swing.JSeparator jSeparator1
- javax.swing.JLabel label_actividad
- javax.swing.JLabel label_calendario
- javax.swing.JLabel label_contacto
- javax.swing.JLabel label_foto
- javax.swing.JLabel label_grafica
- javax.swing.JLabel label_logo
- javax.swing.JLabel label_nombre
- javax.swing.JLabel label_usuario
- javax.swing.JPanel panel/cambia
- javax.swing.JPanel panel/cambia_actividad
+ Principal()
- void label_actividadMouseClicked(java.awt.event.MouseEvent evt)
- void label_calendarioMouseClicked(java.awt.event.MouseEvent evt)
- void label_graficaMouseClicked(java.awt.event.MouseEvent evt)
- void label_logoMouseClicked(java.awt.event.MouseEvent evt)
- void label_logoMouseExited(java.awt.event.MouseEvent evt)
- void label_actividadMouseExited(java.awt.event.MouseEvent evt)
- void label_calendarioMouseExited(java.awt.event.MouseEvent evt)
- void label_graficaMouseExited(java.awt.event.MouseEvent evt)
- void label_logoMouseEntered(java.awt.event.MouseEvent evt)
- void label_actividadMouseEntered(java.awt.event.MouseEvent evt)
- void label_calendarioMouseEntered(java.awt.event.MouseEvent evt)
- void label_graficaMouseEntered(java.awt.event.MouseEvent evt)
- void label_contactoMouseClicked(java.awt.event.MouseEvent evt)
- void label_contactoMouseEntered(java.awt.event.MouseEvent evt)
- void label_contactoMouseExited(java.awt.event.MouseEvent evt)
- void label_fotoMouseClicked(java.awt.event.MouseEvent evt)
+ void main(String args)

PanelCalendario
- int total_actividades_mes
- int total_act_compartidas_mes
- HashMap paneles
- HashMap botones
- String[] actividades_propias
- String[] actividades_compartidas
- String[] total_actividades
- int mes_actual
- int ano_actual
- boolean hayNotificacion
- TareasPendientes tp_Frame
- javax.swing.JButton boton_actualizar
- javax.swing.JButton boton_borrar
- javax.swing.JComboBox<String> box_ano
- javax.swing.JComboBox<String> box_mes
- javax.swing.JTextField caja_categoria
- javax.swing.JTextField caja_compartido_con
- javax.swing.JTextField caja_compartido_por
- javax.swing.JTextField caja_fecha
- javax.swing.JTextField caja_hora
- javax.swing.JTextField caja_nombre
- javax.swing.JTextField caja_prioridad
- javax.swing.JLabel jLabel1
...
- javax.swing.JLabel jLabel10
- javax.swing.JLabel label1
...
- javax.swing.JLabel label42
- javax.swing.JLabel label_c1
...
javax.swing.JLabel label_c42
- javax.swing.JLabel label_calendario
- javax.swing.JLabel label_cirulo
- javax.swing.JLabel label_comentario
- javax.swing.JLabel label_domingo
- javax.swing.JLabel label_flecha
- javax.swing.JLabel label_jueves
- javax.swing.JLabel label_lunes

Esta clase sigue en la página siguiente

```
- javax.swing.JLabel label_lupa
- javax.swing.JLabel label_martes
- javax.swing.JLabel label_mes
- javax.swing.JLabel label_miercoles
- javax.swing.JLabel label_num_notificaciones
- javax.swing.JLabel label_sabado
- javax.swing.JLabel label_tareas_pendientes
- javax.swing.JLabel label_viernes
- javax.swing.JPanel panel1
...
- javax.swing.JPanel panel42
- javax.swing.JPanel panel_actividad
- javax.swing.JPanel panel_completo

+ PanelCalendario()
- void introduceActividades(int ano, int mes)
- void introduceActividades(int ano, int mes)
- void introduceBoton(int actividad, JPanel panel)
- void poneIconoBotones(int actividad, JButton boton)
- void botones_actionPerformed(ActionEvent e)
- void rellenaHashmaps()
- void rellenaActividad(int actividad)
- void borraBotones()
- JPanel devuelvePanel(String name)
- void poneNombresPaneles()
- void colocaDias(int primer_dia, int num_dias_mes)
- void actualizaCalendario()
- int primerDiaMesActual()
- void label_flechaMouseClicked(java.awt.event.MouseEvent evt)
- void label_lupaMouseClicked(java.awt.event.MouseEvent evt)
- void label_lupaMouseEntered(java.awt.event.MouseEvent evt)
- void label_lupaMouseExited(java.awt.event.MouseEvent evt)
- void boton_borrarActionPerformed(java.awt.event.ActionEvent evt)
- void boton_actualizarActionPerformed(java.awt.event.ActionEvent evt)
- void label_tareas_pendientesMouseClicked(java.awt.event.MouseEvent evt)
```