Imagen que contiene objeto, reloj, señal

Descripción generada automáticamenteLogotipo

Descripción generada automáticamenteUniversidad Autónoma del Estado de México

Unidad Académica Profesional Tianguistenco

Proyecto: **MOCHILA VIRTUAL BACKAPP**

Empresa: **rk**

Equipo: 1

Alumnos:

Kevin Jesús Escutia Ceja

Rubi Esmeralda Rosales Chavero

Carrera: Ingeniería en software

**CONTENIDO**

[Introducción 3](#_Toc72241960)

[Fundamentos 4](#_Toc72241961)

[Antecedentes 4](#_Toc72241962)

[Vista general del proyecto](#_Toc72241963)

[Planteamiento del problema 6](#_Toc72241964)

[Justificación 7](#_Toc72241965)

[Objetivos 8](#_Toc72241966)

[Gestión de procesos](#_Toc72241967)

[Metodología 10](#_Toc72241968)

[Cronograma de actividades 13](#_Toc72241969)

[Análisis de requerimientos](#_Toc72241970)

[Definición de requerimientos 16](#_Toc72241971)

[Control y seguimiento de los requerimientos 17](#_Toc72241972)

[Especificaciones del sistema](#_Toc72241973)

[Diagramas 23](#_Toc72241974)

[Base de datos 28](#_Toc72241975)

[Implementación y pruebas](#_Toc72241976)

[Interfaces de la aplicación 32](#_Toc72241977)

[Pruebas 38](#_Toc72241980)

[Resultados obtenidos 39](#_Toc72241981)

[Conclusiones 40](#_Toc72241982)

## **INTRODUCCIÓN**

Este documento tiene como principal objetivo documentar el diseño y desarrollo de backapp, la cual es una aplicación de escritorio que funciona como una mochila virtual al permitirle a sus usuarios guardar notas en libretas virtuales de todas las materias que registren en el horario ahí mismo en la aplicación.



Este programa es desarrollado por la empresa rk, fundada por Kevin Jesús Escutia Ceja y Rubi Esmeralda Rosales Chavero en el año 2021. Nuestra misión es brindar confianza y mejorar la calidad de vida de nuestros clientes, diseñando productos útiles y prácticos.

Imagen que contiene objeto, reloj, señal

Descripción generada automáticamente

## **FUNDAMENTOS**

Con la aparición de una nueva enfermedad que rápidamente se convirtió en pandemia surgieron muchos cambios para los cuales no estábamos preparados. Uno de ello es el de la educación a distancia que, si bien tiene muchos beneficios en cuestiones de salvaguardar la salud, también tiene algunos contras cuando se trata del aprendizaje de los alumnos.

La idea de este programa fue concebida al notar dificultades para realizar apuntes de clase en la modalidad a distancia de manera sencilla y práctica; principalmente para los alumnos que nunca habían experimentado con esta modalidad de aprendizaje. Ya existen muchas herramientas para tomar notas y existen también editores de texto, no obstante, su propósito no es el de tomar apuntes de clases por lo que no están optimizados para tal fin, a diferencia de backapp.

## **ANTECEDENTES**

La educación a distancia es una modalidad de estudio o proceso de formación mediada por diversas tecnologías, con la finalidad de promover el aprendizaje sin limitaciones de ubicación, ocupación o edad de los estudiantes. Es un estudio autodirigido por el estudiante, quien debe planificar y organizar su tiempo, material didáctico y guía tutorial para responder a las exigencias del curso, carrera o especialización que sigue.

Esta modalidad de estudio se caracteriza, fundamentalmente, por la separación física de los profesores; el uso la tecnología de información y comunicación (material impreso, material digitalizado, material audiovisual, recursos informáticos, etc.), para la instrucción y la interacción entre profesores, estudiantes, y estudiantes entre sí.

# **VISTA GENERAL DEL PROYECTO**

## **PLANTEAMIENTO DEL PROBLEMA**

Durante el confinamiento por la pandemia hemos descubierto problemas que antes estaban bien ocultos dados el contexto en que antes nos encontrábamos, y uno de esos cambios fueron las clases en línea que nos demostró que no estábamos preparados para cambiar radicalmente de hábitos escolares.

Uno de esos hábitos es el de tomar notas, no es lo mismo tener al profesor al frente de la clase y tu tener una libreta, o carpeta con hojas y tomar apuntes de la clase, ocasionalmente tomar fotos para adquirir el conocimiento y fortalecerlo más tarde.

En las clases a distancia la dinámica cambia a estar viendo casi de forma exclusiva el monitor del profesor con información siendo presentada por el profesor, lo cual sumado al poco tiempo que una persona puede mantener su concentración crea un problema al momento de tomar apuntes.

**Delimitación del problema**

Si analizamos con detenimiento la problemática vemos que no es un problema tan simple de resolver porque hay varias cosas involucradas, en primera instancia el confinamiento por la pandemia, ante esta situación prácticamente no hay nada por hacer.

También puede verse como un problema a nivel sociológico y sociocultural, ya que la humanidad esta habituada desde siempre a vivir en sociedad. El repentino cambio de ambiente deriva en múltiples conflictos con la interacción tanto con otras personas, como con el mundo e inclusive con nosotros mismos; no nos es sencillo cambiar de hábitos, y el tomar apuntes en clases de cierta forma es uno de ellos.

Entonces, desde la perspectiva de la ingeniería en software, la solución propuesta es el desarrollo de una aplicación en la que se pueda principalmente tomar notas de manera sencilla, y contar con notificaciones de las sesiones, con las herramientas más básicas para que con el uso de comandos y unos pocos clics los alumnos puedan guardar detalles importantes al tomar sus clases en línea.

## **JUSTIFICACIÓN**

Es más difícil trabajar en word por la inmensa cantidad de herramientas que pose pues no está pensada para tomar notas de clase, si no para trabajos escritos más elaborados, y el bloc de notas es demasiado sencillo como para tomarlo en cuenta por esto mismo.

Así que decidimos crear una aplicación que una lo mejor de ambos mundos para crear una serie de libretas digitales que nos permitan resolver esta problemática. El proyecto que hemos ideado es una mochila virtual.

Esta aplicación permite hacer varias cosas la principal es crear una libreta en donde podrás tomar notas de tus clases virtuales de forma simple, también podrás agregar a las notas las capturas de pantalla de un problema o texto que el profe comparta en la videollamada, estas notas estarán guardadas en la computadora.

Otras funciones son: un horario en el cual la persona puede poner las horas clase y la aplicación se encargará de enviar una notificación al escritorio avisando que la clase empezar en algunos minutos para que se prepare, y finalmente, podemos compartir nuestras notas con amigos y compañeros exportando e importando el archivo generado.

## **OBJETIVOS**

**Principal**

Desarrollo de una aplicación de escritorio que permita ingresar un horario a la aplicación y esta debe mostrar notificaciones de las clases minutos antes de que inicien. Además, debe permitir crear notas y guardarlas en diversos apartados (libretas).

**Específicos**

* Clasificar y guardar las notas en hojas; y las hojas en libretas.
* Permitir la importación y exportación de libretas (carga y descarga de archivos).
* Almacenar horario de clases para enviar notificaciones de las clases minutos antes de que comiencen.

# **GESTIÓN DE PROCESOS**

## **METODOLOGÍA**

**Scrum**

En scrum un proyecto se ejecuta en ciclos temporales cortos y de duración fija ([iteraciones](https://proyectosagiles.org/desarrollo-iterativo-incremental) que normalmente son de 2 semanas, aunque en algunos equipos son de 3 y hasta 4 semanas, límite máximo de feedback de producto real y reflexión).

Cada iteración tiene que proporcionar un resultado completo, un incremento de producto final que sea susceptible de ser entregado con el mínimo esfuerzo al cliente cuando lo solicite.



El proceso parte de la [lista de objetivos/requisitos priorizada](https://proyectosagiles.org/lista-requisitos-priorizada-product-backlog) del producto, que actúa como plan del proyecto. En esta lista el [cliente (product owner)](https://proyectosagiles.org/cliente-product-owner) prioriza los objetivos balanceando el valor que le aportan respecto a su coste (que el [equipo](https://proyectosagiles.org/equipo-team/) estima considerando la [definición de hecho](https://proyectosagiles.org/definicion-de-hecho-definition-of-done/)) y quedan repartidos en iteraciones y entregas.

Las actividades que se llevan a cabo en scrum son las siguientes (*los tiempos indicados son para iteraciones de 2 semanas*):

[**Planificación de la iteración**](https://proyectosagiles.org/planificacion-iteracion-sprint-planning)

El primer día de la iteración se realiza la reunión de planificación de la iteración. Tiene dos partes:

1. Selección de requisitos (2 horas). El cliente presenta al equipo la lista de requisitos priorizada del producto o proyecto. El equipo pregunta al cliente las dudas que surgen y selecciona los requisitos más prioritarios que prevé que podrá completar en la iteración, de manera que puedan ser entregados si el cliente lo solicita.
2. Planificación de la iteración (2 horas). El equipo elabora la [lista de tareas de la iteración](https://proyectosagiles.org/lista-tareas-iteracion-sprint-backlog) necesarias para desarrollar los requisitos seleccionados. La estimación de esfuerzo se hace de manera conjunta y los miembros del equipo se autoasignan las tareas, se [autoorganizan](https://proyectosagiles.org/2018/10/09/auto-organizacion-fundamentos-y-relacion-con-la-motivacion-intrinseca/) para trabajar incluso en parejas (o grupos mayores) con el fin de compartir conocimiento (creando un equipo más resiliente) o para resolver juntos objetivos especialmente complejos.

[**Ejecución de la iteración**](https://proyectosagiles.org/ejecucion-iteracion-sprint)

Cada día el equipo realiza una [reunión de sincronización](https://proyectosagiles.org/reunion-diaria-de-sincronizacion-scrum-daily-meeting) (15 minutos), normalmente delante de un [tablero físico o pizarra (scrum taskboard)](https://proyectosagiles.org/2010/09/26/ejemplo-tablero-pizarra-tareas-scrum-taskboard/). El equipo inspecciona el trabajo que el resto está realizando (dependencias entre tareas, progreso hacia el objetivo de la iteración, obstáculos que pueden impedir este objetivo) para poder hacer las adaptaciones necesarias que permitan cumplir con la previsión de objetivos a mostrar al final de la iteración.

En la reunión cada miembro del equipo responde a tres preguntas: ¿qué he hecho desde la última reunión de sincronización para ayudar al equipo a cumplir su objetivo? ¿qué voy a hacer a partir de este momento para ayudar al equipo a cumplir su objetivo? ¿qué impedimentos tengo o voy a tener que nos impidan conseguir nuestro objetivo?

Durante la iteración el [facilitador (scrum master)](https://proyectosagiles.org/facilitador-scrum-master) se encarga de que el equipo pueda mantener el foco para cumplir con sus objetivos.

* Elimina los obstáculos que el equipo no puede resolver por sí mismo.
* Protege al equipo de interrupciones externas que puedan afectar el objetivo de la iteración o su productividad.

Durante la iteración, el cliente junto con el equipo [refinan la lista de requisitos (para prepararlos para las siguientes iteraciones) y, si es necesario, cambian o replanifican los objetivos del proyecto (10%-15% del tiempo de la iteración)](https://proyectosagiles.org/replanificacion-proyecto) con el objetivo de [maximizar la utilidad de lo que se desarrolla](https://proyectosagiles.org/beneficios-de-scrum#flexibilidad-adaptacion) y el [retorno de inversión](https://proyectosagiles.org/beneficios-de-scrum#gestion-roi).

Inspección y adaptación

**Inspección y adaptación**

El último día de la iteración se realiza la reunión de revisión de la iteración. Tiene dos partes:

1. [Revisión (demostración)](https://proyectosagiles.org/demostracion-requisitos-sprint-review) (1,5 horas). El equipo presenta al cliente los requisitos completados en la iteración, en forma de incremento de producto preparado para ser entregado con el mínimo esfuerzo. En función de los resultados mostrados y de los cambios que haya habido en el contexto del proyecto, el cliente realiza las adaptaciones necesarias de manera objetiva, ya desde la primera iteración, replanificando el proyecto.
2. [Retrospectiva](https://proyectosagiles.org/retrospectiva-sprint-retrospective) (1,5 horas). El equipo analiza cómo ha sido su manera de trabajar y cuáles son los problemas que podrían impedirle progresar adecuadamente, mejorando de manera continua su productividad. El facilitador se encargará de eliminar o escalar los obstáculos identificados que estén más allá del ámbito de acción del equipo.

## **CRONOGRAMA DE ACTIVIDADES**

**Primer parcial**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Programación de revisiones | | | |
| Actividad | 26-feb | 05-mar | 12-mar | 19-mar |
| 1. Recopilación de información |  |  |  |  |
| 1.1 introducción |  |  |  |  |
| 1.2 fundamentos |  |  |  |  |
| 1.3 antecedentes |  |  |  |  |
| 1.4 planteamiento del problema |  |  |  |  |
| 2. Planeación y estimación |  |  |  |  |
| 2.1 objetivo general |  |  |  |  |
| 2.2 objetivos específicos |  |  |  |  |
| 2.3 justificación |  |  |  |  |
| 3. Análisis de requerimientos |  |  |  |  |
| 3.1 delimitación del problema |  |  |  |  |
| 3.2 definición de requerimientos |  |  |  |  |
| 4. Análisis y especificación del sistema |  |  |  |  |
| 4.1 diagrama de casos de uso |  |  |  |  |
| 4.2 diagrama de secuencia |  |  |  |  |
| 4.4 diagrama de clases |  |  |  |  |
| 4.5 diagramas entidad-relación |  |  |  |  |
| 5. Arquitectura de la base de datos |  |  |  |  |
| 5.1 diseño de la base de datos |  |  |  |  |
| 5.2 diccionario de datos |  |  |  |  |
| 5.3 diseño del prototipo |  |  |  |  |
| 6. Diseño de interfaces |  |  |  |  |
| 6.1 diseño del prototipo |  |  |  |  |
| 7. Codificación |  |  |  |  |
| Avances de programación |  |  |  |  |

**Segundo parcial**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Revisiones | | | |
| Actividad | 16-abrl | 23-abr | 07-may | 21-may |
| 7. Codificación |  |  |  |  |
| Avances de programación |  |  |  |  |
| 8. Documentación |  |  |  |  |
| 8.1 Manual técnico |  |  |  |  |
| 8.2 Manual de Usuario |  |  |  |  |
| 8.3 Manual de instalación |  |  |  |  |
|  |  |  |  |  |
| 9. Implementación |  |  |  |  |
| 10. Pruebas de software |  |  |  |  |
| 12. Resultados obtenidos |  |  |  |  |
| 13. Conclusiones |  |  |  |  |

# **ANÁLISIS DE REQUERIMIENTOS**

## **DEFINICIÓN DE REQUERIMIENTOS**

**Requerimientos funcionales**

|  |  |
| --- | --- |
| **Id** | **Descripción** |
| 01 | La mochila guardara objetos libreta. |
| 02 | El objeto libreta este compuesto de hojas. |
| 03 | Las hojas son objetos que tienen texto e imágenes. |
| 04 | Las libretas se guardarán en archivos binarios en la computadora. |
| 05 | Se pueden importar y exportar libretas. |
| 06 | La base de datos guardara horarios, materias. |
| 07 | La aplicación permitirá hacer un horario de clases. |
| 08 | El horario preguntara de que día hasta que día dura el periodo escolar. |
| 09 | El horario preguntara materia, horas, días, profesor. |
| 10 | El horario generará la libreta respectiva. |
| 11 | La aplicación mandara notificación antes de la hora de inicio para avisar sobre una clase. |

**Requerimientos no funcionales**

|  |  |
| --- | --- |
| **Id** | **Descripción** |
| 20 | La interfaz debe ser vistosa |
| 21 | La representación debe ser en escritorio. |

## **CONTROL Y SEGUIMIENTO DE LOS REQUERIMIENTOS**

Las tablas que enseguida se presentan, reúnen todas las actualizaciones o cambios que se han realizado en la implementación de los requerimientos funcionales y no funcionales (se pueden distinguir por su id).

|  |  |  |
| --- | --- | --- |
| Id | 01 |  |
| Descripción | La mochila guardara objetos libreta. | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Fecha | Estatus | Observaciones | Bd | Interfaz |
| 04/05/2021 | En desarrollo | La aplicación ya crea archivos en la carpeta documentos de todo usuario | Materias | Creacion |
| 05/05/2021 | Terminado | La aplicación ya crea la libreta y sus subcarpetas de hojas | No | Creacion  Libreta |

|  |  |  |
| --- | --- | --- |
| Id | 02 |  |
| Descripción | El objeto libreta este compuesto de hojas. | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Fecha | Estatus | Observaciones | Bd | Interfaz |
| 05/05/2021 | En desarrollo | La libreta ya genera la estructura de las hojas | No | Libretas  Opciones |
| 06/05/2021 | Terminado | La aplicación ya detecta en diferenciar las distintas libretas | No | Libretas  Opciones |

|  |  |  |
| --- | --- | --- |
| Id | 03 |  |
| Descripción | Las hojas son objetos que tienen texto, imágenes. | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Fecha | Estatus | Observaciones | Bd | Interfaz |
| 06/05/2021 | Terminado | Las hojas guardan el título y texto | No | Libreta |
| 06/05/2021 | Terminado | Las hojas ya guardan las imágenes y las renombra conforme a la estructura de las libretas | No | Libreta |
| 25/05/2021 | Terminado | La libreta puede modificar la fuente y el tamaño de letra | No | Libreta |

|  |  |  |
| --- | --- | --- |
| Id | 04 |  |
| Descripción | Las libretas se guardarán en archivos en la computadora. | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Fecha | Estatus | Observaciones | Bd | Interfaz |
| 05/05/2021 | Terminado | Se guardan conforme a la estructura de la libreta | No | Creacion  Eliminacion  Libreta  Opciones |

|  |  |  |
| --- | --- | --- |
| Id | 05 |  |
| Descripción | Se pueden importar y exportar libretas. | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Fecha | Estatus | | Observaciones | | Bd | Interfaz |
| 20/05/2021 | Terminado | | El programa comprime y descomprime carpetas | | No | Importar  Exportar |
| Id | | 06 | |  | | |
| Descripción | | La base de datos guardara horarios, materias. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Fecha | Estatus | Observaciones | Bd | Interfaz |
| 16/04/2020 | En desarrollo | Las demás clases ya están definidas, y las tablas creadas, así como los métodos para la conexión, pero no hemos implementado a las interfaces |  |  |
| 04/05/2021 | Terminada | La bd ya se conecta con la aplicación |  |  |

|  |  |  |
| --- | --- | --- |
| Id | 07 |  |
| Descripción | La aplicación permitirá hacer un horario de clases. | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Fecha | Estatus | Observaciones | Bd | Interfaz |
| 22/04/2021 | Desarrollo | Faltan terminar detalles visuales y limitaciones para evitar errores del usuario | Horarios | Horarios |
| 04/05/2021 | Desarrollo | Los horarios ya traen los datos de la bd | Horarios | Horarios |
| 07/05/2021 | Desarrollo | Los campos están siendo modificados conforme la metodología scrum | Horarios | Horarios |
| 20/05/2021 | Terminado | Los campos ya son usables de manera completa para el usuario, aunque pueden ser mejoradas | Horarios | Horarios |

|  |  |  |
| --- | --- | --- |
| Id | 08 |  |
| Descripción | El horario preguntara de que día hasta que día dura el periodo escolar. | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Fecha | Estatus | Observaciones | Bd | Interfaz |
| 24/04/2021 | Terminado | El sistema revisa todos los días si ya termino el periodo dado | No | Opciones |

|  |  |  |
| --- | --- | --- |
| Id | 09 |  |
| Descripción | El horario preguntara materia, horas, días, profesor. | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Fecha | Estatus | Observaciones | Bd | Interfaz |
| 23/04/2021 | Terminado | Limitar la forma en que el usuario elija las horas | Horarios | Horarios |

|  |  |  |
| --- | --- | --- |
| Id | 10 |  |
| Descripción | El horario generará la libreta respectiva. | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Fecha | Estatus | Observaciones | Bd | Interfaz |
| 06/05/2021 | Terminada | Ninguna | No | Horarios |

|  |  |  |
| --- | --- | --- |
| Id | 11 |  |
| Descripción | La aplicación mandara notificación antes de la hora de inicio para avisar sobre una clase. | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Fecha | Estatus | Observaciones | Bd | Interfaz |
| 07/05/2021 | Terminada | Activa la notificación en la hora | Horarios | Opciones |

|  |  |  |
| --- | --- | --- |
| Id | 20 |  |
| Descripción | La interfaz debe ser vistosa | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Fecha | Estatus | Observaciones | Bd | Interfaz |
| 27/05/2021 | Terminada | La interfaz usa lo mínimo necesario | Horarios  Materias | Todas |

|  |  |  |
| --- | --- | --- |
| Id | 21 |  |
| Descripción | La representación debe ser en escritorio. | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Fecha | Estatus | Observaciones | Bd | Interfaz |
| 27/05/2021 | Terminado | Por el momento es la única versión que se tendrá en cuenta | Horarios  Materias | Todos |

# **ESPECIFICACIONES DEL SISTEMA**

## **DIAGRAMAS**

**Diagrama de casos de uso**

Diagrama

Descripción generada automáticamente

**Diagrama de secuencias**

**Manipulación de horario**

Diagrama

Descripción generada automáticamente

**Manipulación de notas (creación, modificación, eliminación y consulta)**

Diagrama

Descripción generada automáticamente

**Manipulación de notas (importación y exportación)**

Diagrama

Descripción generada automáticamente

**Diagrama de clases**

Diagrama

Descripción generada automáticamente

**Diagrama entidad-relación**

Diagrama

Descripción generada automáticamente

## **BASE DE DATOS**

Nuestra base de datos no es muy complicada dado que solo se guardan datos planos de horarios y materias; necesarios para realizar las operaciones contempladas para el sistema en los requerimientos antes mencionados.

**Diseño de la base de datos**

Diagrama

Descripción generada automáticamente

Horarios: tabla usada para almacenar en una bd el horario de las materias que el usuario añadió y será usado por la app para mandar las notificaciones.

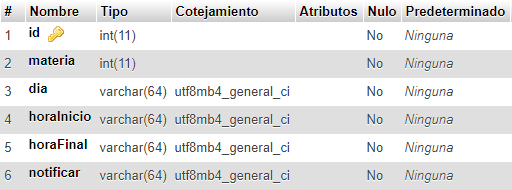
Materia: tabla usada para almacenar las materias que contienen su horario y los datos correspondientes de identificación de la materia

**Diccionario de datos**

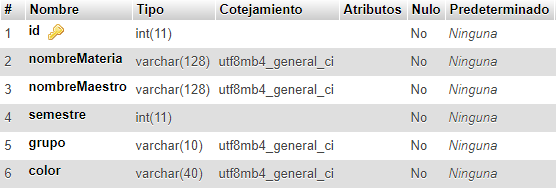
|  |  |  |  |
| --- | --- | --- | --- |
| Campo | Tamaño | Tipo | Descripción |
| Horarios | | | |
| ID | 11 | int | Identificador del horario |
| materia | 11 | Int | Identificador para relacionar la materia |
| dia | 64 | varchar | Día de la semana en la que hay una clase (lunes, martes, etc). |
| horaInicio | 64 | varchar | Hora en la que inicia la clase |
| horaFinal | 64 | varchar | Hora en la que finaliza la clase |
| notificar | 64 | varchar | Este campo puede contener TRUE o FALSE dependiendo de si se quiere notificar las clases de esta materia, o no. |
| Materia | | | |
| Id | 11 | int | Identificador de la materia |
| nombreMateria | 128 | varchar | Nombre de la materia |
| nombreMaestro | 128 | varchar | Nombre del docente que da la materia |
| semestre | 11 | Int | Semestre al que pertenece la materia |
| grupo | 10 | varchar | Grupo en el que se cursa la materia |
| color | 40 | varchar | Representación textual de un objeto Color de java. Se usa para identificar a la materia en el horario de la aplicación. |

**Diseño del prototipo**

**Horarios:**



**Materia:**



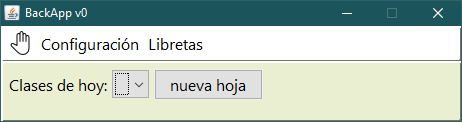
# **IMPLEMENTACIÓN Y PRUEBAS**

## **INTERFACES DE LA APLICACIÓN**

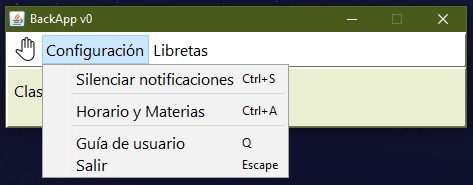
A continuación, se presentan de forma general las interfaces de la aplicación y sus características. No se entrará a fondo sobre el funcionamiento u operación de cada una ya que esos temas son cubiertos con los manuales técnicos y de usuario.

**DISEÑO DE PROTOTIPOS**

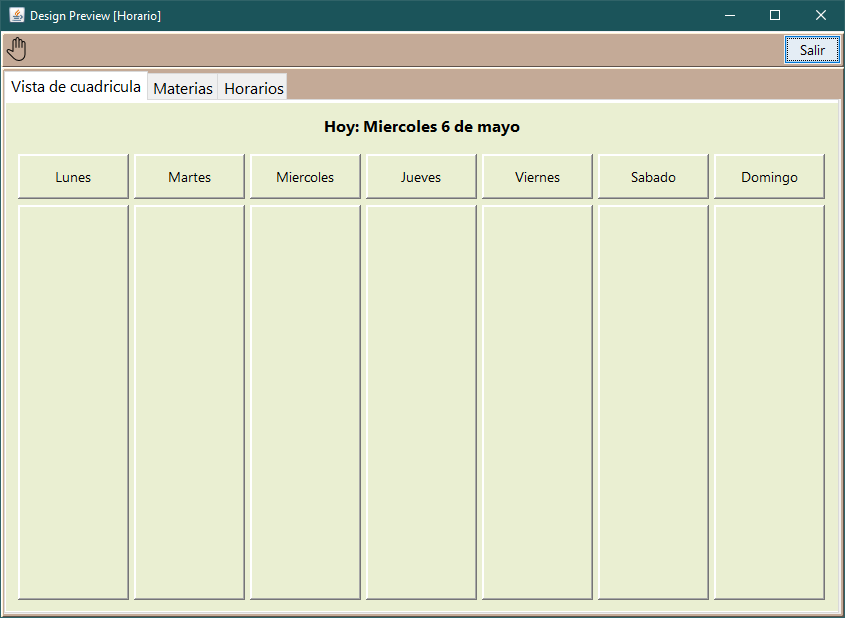
* **Inicio:**



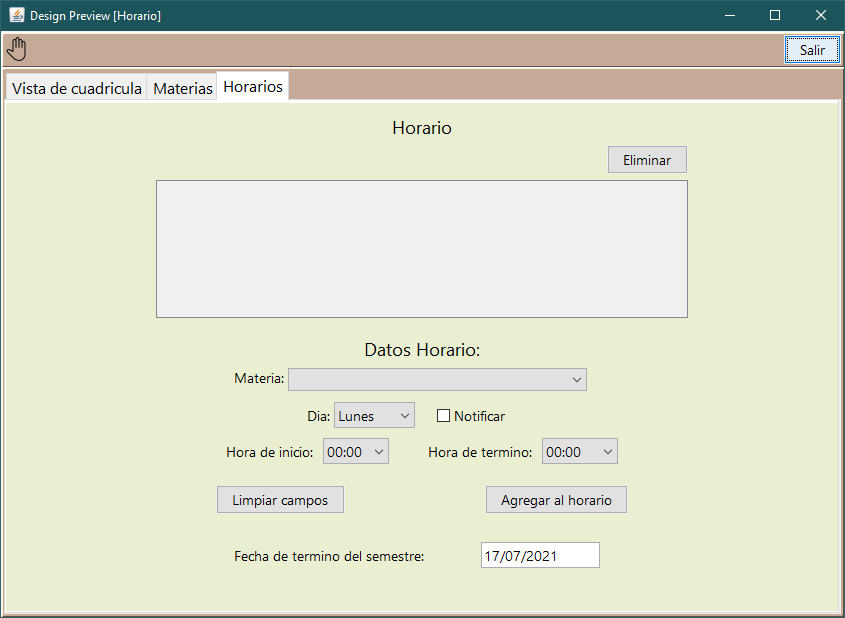
* **Configuración:**



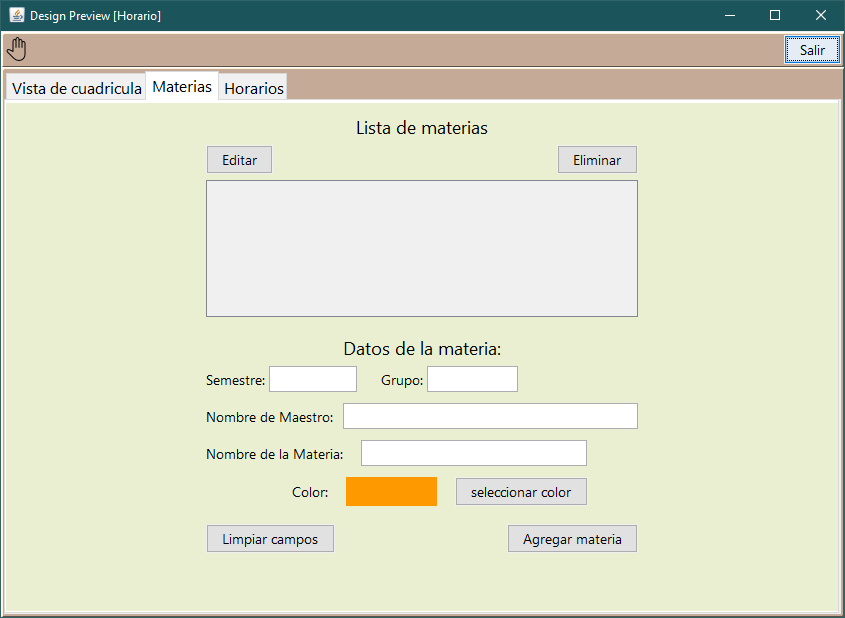
* + **Vista de cuadricula:**



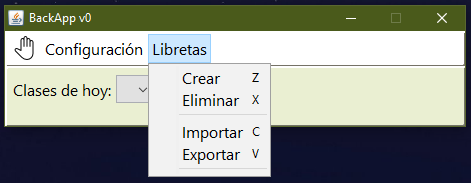
* + **Editar horario:**



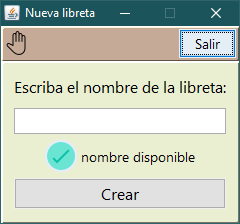
* + **Editar materias:**

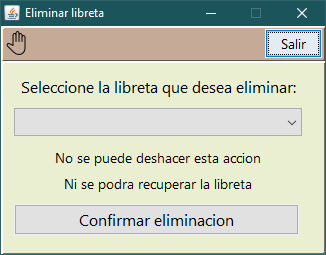


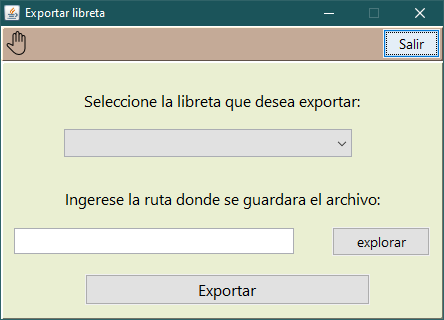
* **Libretas**

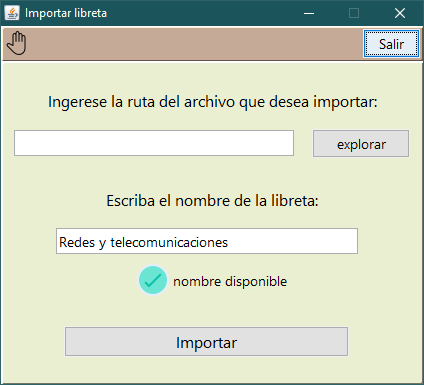


* + **Operaciones con libretas**

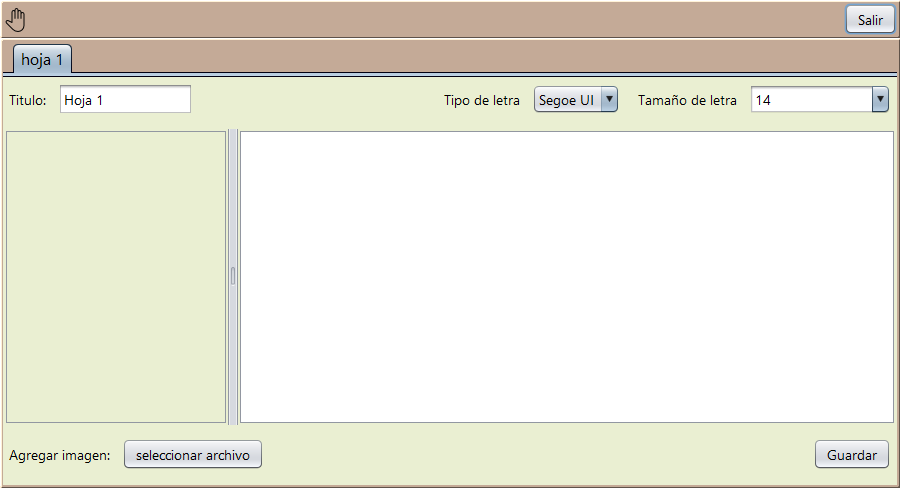








* + **Visualización de hojas en la libreta**





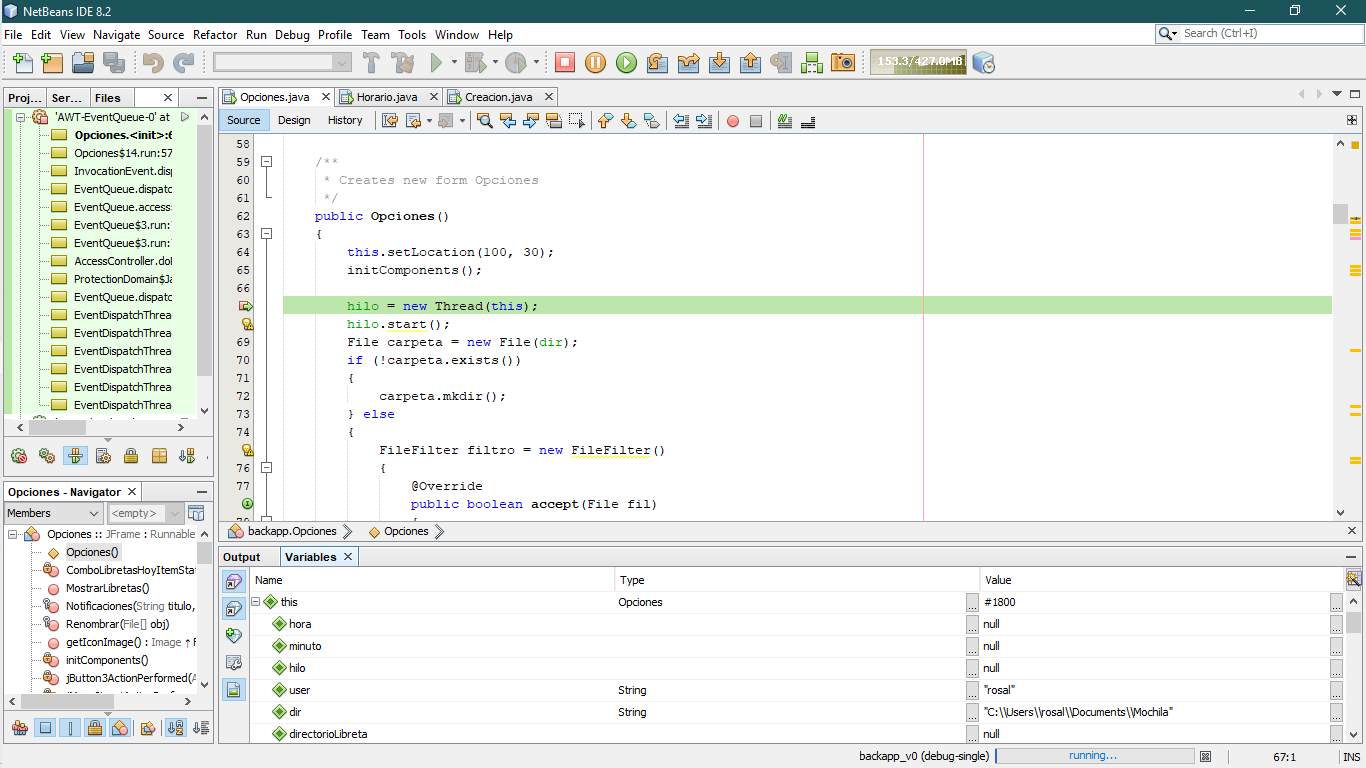
**CARACTERÍSTICAS GLOBALES**

|  |  |  |
| --- | --- | --- |
| **Categoría** | **Elemento** | **Especificaciones** |
| Tipografía | Fuente | Segoe ui |
| Títulos | Plain, tamaño 18 |
| Subtítulos | Plain, tamaño 16 |
| Contenido | Plain, tamaño 14 |
| Resaltar palabras | Bold |
| Colores | Fondo principal | Verde [234,239,210] |
| Fondo secundario | Café [196,170,151] |
| Para las materias en el horario | Personalizado (el usuario lo selecciona) |
| Texto | Negro [0,0,0] |
| Cuadros de texto | Blanco [255,255,255] |
| Botones, menús, combobox | Predeterminado |
| Ventanas | Dimensiones | Default |
| Layout | Free desing, |
| Resizable | Solo horarios, materias y libretas |
| Iconos | Paquete principal | Https://www.flaticon.es/packs/interface-87 |
| Otros iconos | Si se requiere, se usarán iconos de otro paquete (estilo flat). |
| Dimensiones | 32 px |

## **PRUEBAS**

La mayor cantidad de pruebas realizadas se llevó a cabo durante la fase de codificación del proyecto. Estas pruebas fueron de caja blanca. Se probo cada una de las funciones del programa para asegurarse de que se cumplan con éxito.

Una herramienta muy usada para hacer estas pruebas fue el debugger de la IDE NetBeans, como ejemplo se muestra la siguiente captura de debug del código.



Aprovechamos este espacio para dejar el link del repositorio en GitHub donde se puede consultar la evolución del proyecto:

<https://github.com/kinggnome115/backapp_v0>

## **RESULTADOS OBTENIDOS**

Después de 2 meses de desarrollo, la versión 1 de BackApp cuenta con las siguientes funciones:

* Gestionar horario incluye las siguientes operaciones:
  + Creación, Edición y Eliminación.
  + Consulta (forma de cuadricula y en una tabla).
  + Gestionar materias, que incluye las siguientes operaciones:
    - Creación, Edición y Eliminación.
    - Consulta (en una tabla).
* Manipular “libretas”
  + Operaciones sobre las libretas:
    - Creación (desde la ventana principal o de forma directa al registrar una nueva materia), Edición y Eliminación.
    - Consulta (desde la ventana principal).
    - Importación y Exportación.
  + Añadir hojas a las libretas, las cuales solo tienen 3 operaciones:
    - Creación y Edición. y Eliminación.

Otras pequeñas funciones añadidas, complementarias e independientes a las antes mencionadas son:

* Botón de ayuda en la ventana principal (de inicio).
* Activar y desactivar las notificaciones de la aplicación.

Con estas funciones se cubre en su totalidad lo previsto desde la etapa de toma de requerimientos; por lo que, como resultado, tenemos un software que satisface la problemática para la que fue pensado.

## **CONCLUSIONES**

Como en todo desarrollo de software, surgen problemas en cada etapa desde la toma de requerimientos, sin embargo, como ingenieros en software debemos aprender a sobre llevar estas dificultades, pero aún más importante es saber prevenirlas.

Este proyecto, aunque parezca pequeño, no fue la excepción ante las dificultades; pero su tamaño supuso una ventaja a la hora de resolverlas. Uno de los principales problemas fueron los requerimientos, que cambiaron ligeramente a lo largo del proyecto, modificando el proyecto de lo que se tenía planteado a los resultados.

Así que, como conclusión y experiencia, nos llevamos a casa el destinar más tiempo a la toma de requerimientos y diseño del software. Es indispensable, tener 100% claro cómo queremos que sea “algo” para obtener como resultado ese “algo”.

Al no tener definidos en su totalidad los requerimientos, o no ser muy específicos con algunos de ellos, damos pie a futuras interpretaciones erróneas u omisiones de elementos importantes para el funcionamiento del sistema.