



Documento de Propuesta de
Diseño de Software I, II y III

Metodología de Creación Módulo de Pantalla Compartida

Autor

Santiago Enrique Palomo Otero

Jhoan sebastian Fernandez Julio

Elian David Morelo Rivas

Duvan Mangonez Rodriguez

Aury Sandrith Angulo Palencia

Asesor

Raul Toscano Miranda

Universidad de Córdoba

Facultad de Educación y Ciencias Humanas

Licenciatura en Informática

2024



INTRODUCCIÓN

El desarrollo del módulo de pantalla compartida para la plataforma CREAVI representó una oportunidad para combinar avances tecnológicos y principios pedagógicos, con el objetivo de mejorar la colaboración y la interacción en entornos educativos. Este proyecto no solo abordó los desafíos técnicos asociados con la creación de un sistema colaborativo en tiempo real, sino que también sirvió como un espacio de aprendizaje para los desarrolladores involucrados. Utilizando la metodología ágil de Scrum, el equipo organizó el trabajo en sprints, permitiendo la entrega incremental de funcionalidades clave como la edición colaborativa, la carga de multimedia y la seguridad en la autenticación de usuarios.

Durante el proceso de desarrollo, se llevaron a cabo rigurosas pruebas de integración y unitarias, con el fin de asegurar que todos los componentes del sistema funcionaran de manera coherente y eficiente. Estas pruebas, junto con un proceso continuo de retroalimentación, fueron fundamentales para detectar y resolver problemas de compatibilidad y optimizar el rendimiento del sistema. Además, las revisiones periódicas ofrecieron valiosas oportunidades de aprendizaje, tanto a nivel técnico como colaborativo, mejorando la calidad del producto y fortaleciendo las habilidades del equipo.

Aunque el módulo ha alcanzado un estado funcional avanzado, el proceso de desarrollo continúa en curso, con el objetivo de refinar la experiencia del usuario y agregar nuevas funcionalidades que maximicen el potencial del sistema. Este enfoque iterativo y colaborativo no solo asegura un producto final más robusto y eficiente, sino que también fomenta un aprendizaje continuo y el perfeccionamiento de las prácticas de desarrollo ágil.



Metodología de Trabajo en el Desarrollo del módulo de pantalla compartido

El desarrollo del módulo de pantalla compartida no solo representó un desafío tecnológico, sino también una experiencia educativa enriquecedora. Desde el inicio, el proyecto tuvo un doble propósito: diseñar una herramienta eficaz para fomentar la colaboración y el intercambio de contenido multimedia, y proporcionar a los desarrolladores en formación una valiosa oportunidad de aprendizaje práctico. El equipo adoptó la metodología Scrum, combinando enfoques colaborativos, iteraciones ágiles y estrategias innovadoras. Esto convirtió el proyecto en un laboratorio activo de aprendizaje, donde se integraron habilidades técnicas, principios de diseño de software educativo y prácticas de desarrollo ágil para alcanzar los objetivos propuestos.

DISEÑO DE SOFTWARE (1-2)

Metodología SCRUM: Desde el comienzo de este proyecto, ya estaba hecho por el profesor **ALEXANDER TOSCANO** cual nos facilitó el proyecto y nos dio su consentimiento para que nos adaptamos ya que el proyecto donde aportaba con los objetivos, contextualización del problema, caso de usos, modelo entidad-relación, requisitos funcionales y no funcionales, diagrama relacional, modelo NSQL anterior de nosotros estaba incompleto y por motivos de tiempo para primer corte no daban cuenta para adaptarnos por eso nos tocó cambiar y para comenzar ahora en software 2 se eligió trabajar con la metodología Scrum como el marco metodológico para estructurar y organizar el desarrollo del módulo de pantalla compartida en ciclos repetitivos, conocidos como sprints. Esta metodología no solo facilitó la gestión eficiente del proyecto, sino que también potenció el aprendizaje y la colaboración dentro del equipo. Los roles estuvieron claramente definidos: el de la documentación se encargó de establecer los requisitos, priorizar tareas y garantizar que el módulo cumpliera con las expectativas funcionales y educativas. El líder actuó como facilitador, eliminando obstáculos y asegurando que el equipo mantuviera el enfoque en los objetivos de cada sprint. Por su parte, el backend, compuesto por estudiantes poco capacitados en backend, frontend y diseño de interfaces, ejecutó las tareas técnicas definidas en **github**.

El trabajo se organizó de manera iterativa, con reuniones diarias para revisar



avances, identificar problemas y ajustar los planes. Al final de cada sprint, se realizaban revisiones y retrospectivas, lo que permitió evaluar logros, superar desafíos y mejorar la estrategia para el siguiente ciclo. Este enfoque permitió combinar eficazmente el trabajo colaborativo con tareas independientes, posibilitando que cada miembro profundiza en su especialidad sin perder de vista la visión integral del proyecto. Así, el desarrollo no solo cumplió con los objetivos tecnológicos, sino que se convirtió en un laboratorio activo de aprendizaje práctico y multidisciplinario.

Trabajo En Equipo: El equipo se organizó en subgrupos especializados para abordar las áreas clave del desarrollo, como la implementación del backend, el diseño de la interfaz de usuario y las pruebas de integración. teníamos cada clase de 2 horas para adelantar el proyecto de manera compartida, contábamos con horas independientes trabajando duro 4 o 6 horas por semana arreglando y reconstruyendo lo valorado por el profesor Esta división estratégica no solo facilitó la colaboración y la resolución conjunta de problemas, sino que también potenció el aprendizaje mutuo entre los miembros. En el contexto del desarrollo del módulo de pantalla compartida, esta estructura permitió aprovechar al máximo las capacidades del equipo, aplicando la metodología Scrum para garantizar iteraciones ágiles y un flujo de trabajo eficiente. Así, cada grupo contribuyó de manera significativa a la creación de un producto innovador y funcional, mientras adquirían experiencia práctica en técnicas de desarrollo colaborativo y ágil.

Trabajo Independiente: En el desarrollo del módulo de pantalla compartida, ciertas tareas específicas, como la codificación de componentes y la elaboración de la documentación, se realizaron de manera individual. Este enfoque permitió que cada miembro del equipo se concentrará en su especialidad, aportando soluciones detalladas y de alta calidad al proyecto global. Integrando esta dinámica en el marco de la metodología Scrum, se complementa el trabajo colaborativo con contribuciones individuales, logrando un equilibrio entre la autonomía técnica y la cohesión del equipo. Esto enriqueció tanto el desarrollo del producto como el aprendizaje práctico de los participantes.

Estrategias de Enseñanza: Durante el desarrollo del módulo de pantalla compartida, se implementaron diversas estrategias pedagógicas para enriquecer el proceso de aprendizaje. Entre ellas destacó el enfoque de **Aula Invertida**, que permitió a los desarrolladores explorar teoría y tecnologías antes de abordar cada nueva etapa del proyecto. Esta metodología les proporcionó una base técnica sólida, indispensable para enfrentar los retos del diseño y la programación. Además,



se integró la práctica de Chat GPT, para resolver tareas complejas de manera eficiente. Trabajar en conjunto no solo facilitó revisiones de código inmediatas y de alta calidad, sino que también fomenta el aprendizaje colaborativo, alineándose con los valores de las metodologías ágiles como Scrum. Estas estrategias, combinadas con un entorno de desarrollo ágil y modular, promovieron una experiencia educativa y profesional profundamente enriquecedora.

CASOS DE USOS/FUNCIONALIDADES

1. **Página Principal:** Es el punto de inicio de la interacción, donde el usuario encuentra opciones como menú, título y retroceder. El diseño busca facilitar la navegación mediante iconos y acciones intuitivas.
2. **Página Crear:** Esta sección permite la creación y modificación de contenido en tiempo real. Aquí, los usuarios pueden interactuar mediante iconos para editar, transmitir, silenciar o activar funciones como el micrófono y capturar pantallas.
3. **Página Compartir:** Con esta opción, el usuario puede generar enlaces para compartir el contenido con otros usuarios. También se incluye la posibilidad de agregar manualmente a personas mediante un sistema de invitaciones, promoviendo un entorno colaborativo.
4. **Página Exportar:** Permite adjuntar contenido multimedia y documentos, como vídeos, imágenes y archivos de texto, mejorando la capacidad de transmisión de conocimiento.
5. **Página Comentar:** Esta ventana facilita la comunicación interactiva mediante la función de comentarios, permitiendo que docentes y estudiantes realicen observaciones en tiempo real.

Pruebas y Retroalimentación: Durante el desarrollo del módulo de pantalla compartida, se llevaron a cabo numerosas pruebas para garantizar su correcto funcionamiento y la calidad del sistema. Muchas de ellas siempre han sido fallidas pero tratamos de buscar soluciones con ayuda del profesor, además las pruebas de integración jugaron un papel clave al validar la interoperabilidad de los componentes, asegurando que funcionalidades esenciales como la autenticación segura y la colaboración en tiempo real operaran de manera fluida. Además, estas pruebas se complementaron con revisiones continuas de retroalimentación del



equipo, y subidas a la plataforma de **GitHub**, además de eso aprovechando el enfoque ágil de Scrum para iterar y mejorar tanto la experiencia del usuario como la eficacia técnica del sistema. Este proceso no solo fortaleció el producto final, sino que también potenció el aprendizaje práctico del equipo en entornos de desarrollo colaborativos y ágiles.

DISEÑO DE SOFTWARE 3

Metodología SCRUM: En la tercera parte el desarrollo del módulo de pantalla compartida utilizó SCRUM como marco metodológico, estructurando el trabajo en ciclos repetitivos semanales denominados sprints. Este enfoque permitió al equipo adaptarse de manera ágil a los cambios y mantener un ritmo constante de avance. Cada sprint iniciaba con una planificación detallada liderada por el de la documentación, quien define y prioriza los requisitos en el documento. Estos se dividían en tareas específicas asignadas al Equipo de Desarrollo, compuesto por especialistas en backend, frontend y diseño de componentes, con la ayuda de 2 integrantes al equipo se realizó el guion multimedial para tener base de la idea que uno va a diseñar también los nuevos integrantes los cuales se asignaron uno para documentación y otro para backend y frontend

el líder desempeñó un papel crucial como facilitador, eliminando obstáculos y asegurando que el equipo se mantuviera alineado con los objetivos del sprint. Además, fomenta una cultura de mejora continua, promoviendo la reflexión y el aprendizaje al final de cada iteración.

El ciclo de trabajo de cada sprint se apoyó en varias ceremonias clave:

- Reuniones en clase presencial: sincronizaciones realizadas a primera hora del día donde los miembros del equipo compartían avances, obstáculos y planes inmediatos, permitiendo detectar y resolver problemas de forma temprana antes de iniciar clase.
- Reuniones de revisión del sprint: Al final de cada semana, el equipo evaluaba el progreso del producto mediante demostraciones de las funcionalidades desarrolladas. Esto garantiza la validación temprana del incremento del producto y facilita recibir retroalimentación directa de los interesados.
- Retrospectivas: Espacios de reflexión donde el equipo analizaba éxitos y desafíos, identificando oportunidades para mejorar procesos y colaboración.



Estos aprendizajes se aplicaban en los siguientes sprints, lo que resultó en un incremento constante de la calidad del producto.

El trabajo se organizó combinando esfuerzos colaborativos e independientes. Subgrupos especializados abordaron tareas clave, como el desarrollo del backend, la creación de la interfaz de usuario y las pruebas de integración. Para mantener la cohesión y alineación, se realizaron reuniones semanales por Google Meet, ZOOM, Videollamadas por Whatsapp y mensajes de texto donde cada subgrupo presenta avances, desafíos y soluciones, asegurando que los componentes individuales se integraran sin problemas en el sistema final. Este enfoque repetitivo y altamente colaborativo, con énfasis en la comunicación continua y la mejora constante, no solo permitió cumplir con los requisitos técnicos y funcionales del módulo, sino también potenciar el aprendizaje del equipo, garantizando un producto de alta calidad y escalabilidad.

Estrategia de Aprendizaje: En el desarrollo del módulo de pantalla compartida para la plataforma CREAVI, se implementó una estrategia de aprendizaje integral que combinó metodologías pedagógicas y prácticas colaborativas para maximizar el aprendizaje de los desarrolladores en formación. Uno de los enfoques principales fue el uso del **Aula Invertida**, donde los integrantes del equipo estudiaron previamente conceptos teóricos y exploraron tecnologías relacionadas con cada fase del proyecto, como herramientas de desarrollo frontend, backend y estrategias de seguridad. Este enfoque permitió que los desarrolladores llegaran preparados a las sesiones prácticas, asegurando una comprensión sólida de los principios técnicos antes de enfrentarse a los desafíos del proyecto.

Además, se adoptó la utilización de chat GPT para abordar tareas críticas y complejas, como la implementación de la funcionalidad de colaboración en tiempo real y la autenticación segura. Este método promueve un aprendizaje colaborativo continuo, ya que los desarrolladores más experimentados guiaron a los menos experimentados, fomentando un intercambio activo de conocimientos y habilidades. Durante estas sesiones, cada par de desarrolladores asumió roles alternados de "programador" y "script": el programador escribía el código, mientras que el script revisaba y sugería mejoras en tiempo real. Esta dinámica no solo fortaleció la calidad del código, sino que también facilitó la identificación temprana de errores y la optimización de soluciones.

Las sesiones de codificación en pares se planifican estratégicamente para tareas que requerían alta coordinación, como el diseño de APIs para la pantalla compartida



y la integración de funcionalidades críticas de seguridad. De esta forma, se creó un entorno de aprendizaje constante, colaborativo y enriquecedor, que permitió al equipo abordar los retos del proyecto con eficacia y mejorar sus habilidades técnicas y de trabajo en equipo.

Pruebas y Retroalimentación

El desarrollo del módulo de pantalla compartida incluyó un enfoque continuo de pruebas y retroalimentación para garantizar la funcionalidad del sistema y fomentar la mejora constante. Se llevaron a cabo pruebas de integración con el objetivo de validar la interoperabilidad de los componentes desarrollados por los distintos subgrupos, abarcando funcionalidades clave como la autenticación de usuarios y la colaboración en tiempo real. Asimismo, se realizaron pruebas unitarias para verificar el desempeño individual de cada módulo, asegurando su alineación con los requisitos iniciales.

Al final de cada sprint, el equipo participó en revisiones exhaustivas donde se evaluaron tanto aspectos técnicos, como la calidad del código y el cumplimiento de los objetivos, como colaborativos, enfocándose en la comunicación y la eficiencia del trabajo en equipo. Este proceso de retroalimentación permitió detectar áreas de mejora y realizar ajustes iterativos. Aunque el sistema alcanzó un estado funcional avanzado, todavía se encuentra en proceso de refinamiento, con futuras iteraciones planeadas para incorporar nuevas funcionalidades y optimizar la experiencia de usuario. Este enfoque iterativo asegura que el producto continúe evolucionando en base a las necesidades del usuario y las oportunidades de mejora identificadas



MARCO CONCEPTUAL

1. Desarrollo Ágil de Software

El desarrollo ágil de software se basa en principios que promueven la adaptabilidad y la colaboración entre equipos, permitiendo respuestas rápidas a los cambios en los requisitos del proyecto. Según Beck et al. (2001), los métodos ágiles, como Scrum, se centran en la entrega incremental de valor al usuario mediante iteraciones frecuentes. En este proyecto, Scrum fue esencial para estructurar el trabajo en sprints, facilitando la integración continua y las mejoras iterativas del módulo de pantalla compartida, alineándose con las necesidades de colaboración en tiempo real y escalabilidad.

2. Diseño de Software Educativo

Laurillard (2012) señala que el diseño de software educativo debe integrar principios pedagógicos con tecnologías interactivas, permitiendo a los usuarios no solo consumir información, sino también interactuar activamente con ella. El módulo desarrollado implementa estas ideas mediante herramientas de dibujo, colaboración en tiempo real y un enfoque en la personalización del contenido, maximizando la interacción educativa dentro de la plataforma CREAVI.

3. Pruebas de Software

Las pruebas de software son un componente clave del desarrollo, asegurando que los sistemas cumplan con los requisitos funcionales y no funcionales definidos. Myers, Sandler y Badgett (2011) destacan la importancia de las pruebas de integración para validar la comunicación entre los diferentes componentes del sistema. En el módulo de pantalla compartida, estas pruebas confirmaron el correcto funcionamiento de características como la autenticación, la colaboración en tiempo real y la gestión segura de usuarios.

4. Seguridad en Sistemas Colaborativos

Gollmann (2011) enfatiza que la seguridad en sistemas colaborativos debe garantizar la autenticación, la integridad de los datos y la privacidad de los usuarios. En este proyecto, se implementaron medidas de seguridad como la autenticación mediante JWT y el cifrado de datos en tránsito, protegiendo la información sensible de los usuarios y asegurando un acceso seguro al módulo.

5. Educación y Tecnología Interactiva



Laurillard (2012) resalta que el uso de tecnologías interactivas, como la pantalla compartida, transforma los procesos de enseñanza y aprendizaje al promover la interacción activa y la colaboración entre estudiantes y docentes. El módulo desarrollado aplica este principio, permitiendo a los usuarios trabajar de manera conjunta en actividades educativas, como creación de contenido visual, discusión de ideas y resolución de problemas en tiempo real.

6. Pantalla Compartida en Sistemas Colaborativos

La funcionalidad de pantalla compartida es un componente esencial en sistemas colaborativos, permitiendo que múltiples usuarios visualicen y trabajen simultáneamente en un entorno digital. Según Ellis, Gibbs y Rein (1991), los sistemas colaborativos deben proporcionar mecanismos para facilitar la interacción en tiempo real entre los participantes, como la edición simultánea y el control compartido de recursos visuales. En este proyecto, la pantalla compartida no solo ofrece estas capacidades, sino que también integra herramientas para resaltar elementos y cargar multimedia, enriqueciendo la experiencia de colaboración.

7. Colaboración en Tiempo Real

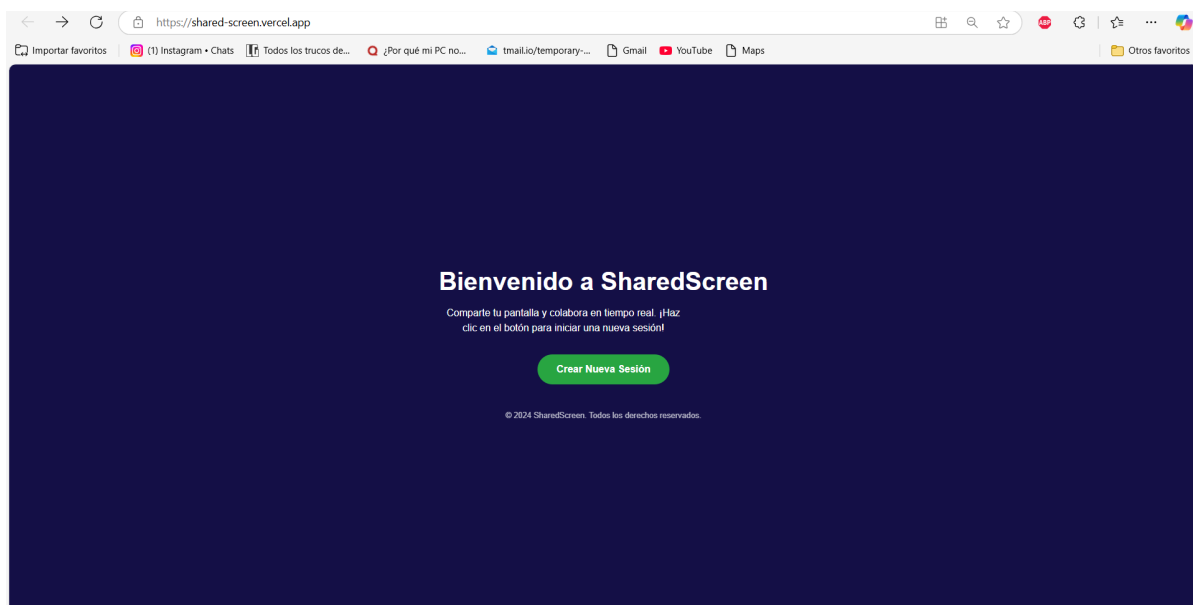
Gutwin y Greenberg (2002) señalan que la colaboración efectiva en tiempo real requiere que los usuarios tengan un "awareness" o conocimiento constante de las acciones de los demás participantes. En el módulo desarrollado, la pantalla compartida implementa actualizaciones en tiempo real, lo que permite a los usuarios visualizar cambios instantáneamente, facilitando el trabajo colaborativo y mejorando la productividad en entornos educativos.

SOFTWARE MÓDULO DE PANTALLA COMPARTIDA TERMINADO

El software ya se terminó completamente y cuenta para operar con 2 ventanas una para trabajar y cuentan con el servidores diferente además una es la pantalla principal y la otra es en segundo plano la que le da la opción de transmitir lo que se está compartiendo en la pizarra de trabajo a continuación se dirán las funcionalidades realizadas y visibles y no funcionales.

el software nos recibe con esta interfaz(no estaba escrito en el guion multimedial)

(ver imagen 1)



(ver imagen 1)

Fase 1

Funcionales

- Al momento de crear la pantalla esta despliega un menu en otra ventana donde muestra exportar pantalla, editar, crear, cargar documentos son las opciones creadas, también cuenta con parar e iniciar pantalla también habilitar camara de el usuario y para finalizar comentar lo que se transmite. lo cual se manejo con html, css, javascripts (ver imagen 2)
- En la otra pagina esta crea la pantalla en forma de pizarra de trabajo (ver imagen 3) metiéndose enseguida a la funcionalidad de editar lo cual con lleva



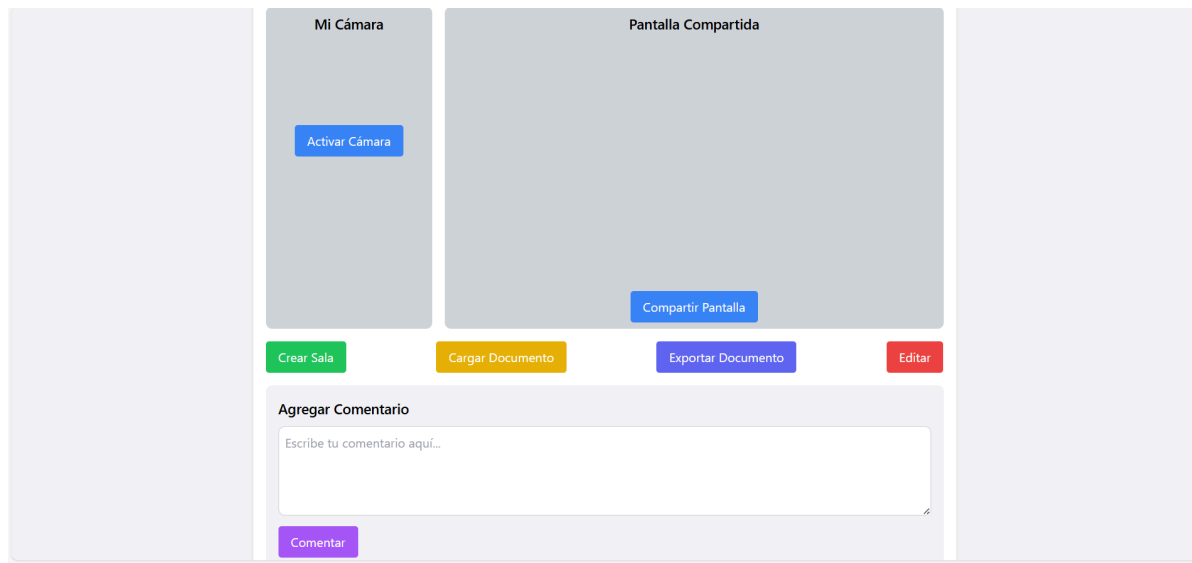
con las demás herramientas como se fuera paint, conlleva con seleccionar, formas, pincel, borrar, rellenar, dar color, lápiz y adjuntar

No Funcionales

- El menu desplegable que aparece en el guion multimedial en la parte izquierda no se realizó por razones que desconocemos cómo se crean ese comando y administrarlo consume tiempo lo cual contábamos con poquito, (Ver imagen 1, imagen 2).
- No se colocó el botón del icono ni la funcionalidad que le da el texto título (Ver imagen 1, imagen 2).
- No se colocó el icono ni la funcionalidad de retroceder (Ver imagen 1, imagen 2).
- Tuvo compliques y se vinculó la opción de comentar pantalla en la pantalla principal (ver imagen 2)
- Se complicó Y se adjuntó subir imágenes en la pantalla de inicio (ver imagen 3)
- No se terminó de dar funcionalidad al botón de Exportar documentos y cargar documentos en la página principal (segundo plano)
- El diseño de la página principal (segundo plano) como estaba en el guion no se realizó correctamente.
- No se terminó de terminar de crear el botón del menu compartir con la funcionalidad de generar el link y mandarlo
- de comentar no se terminó el diseño y no se digitaron las fechas cuando se envían.

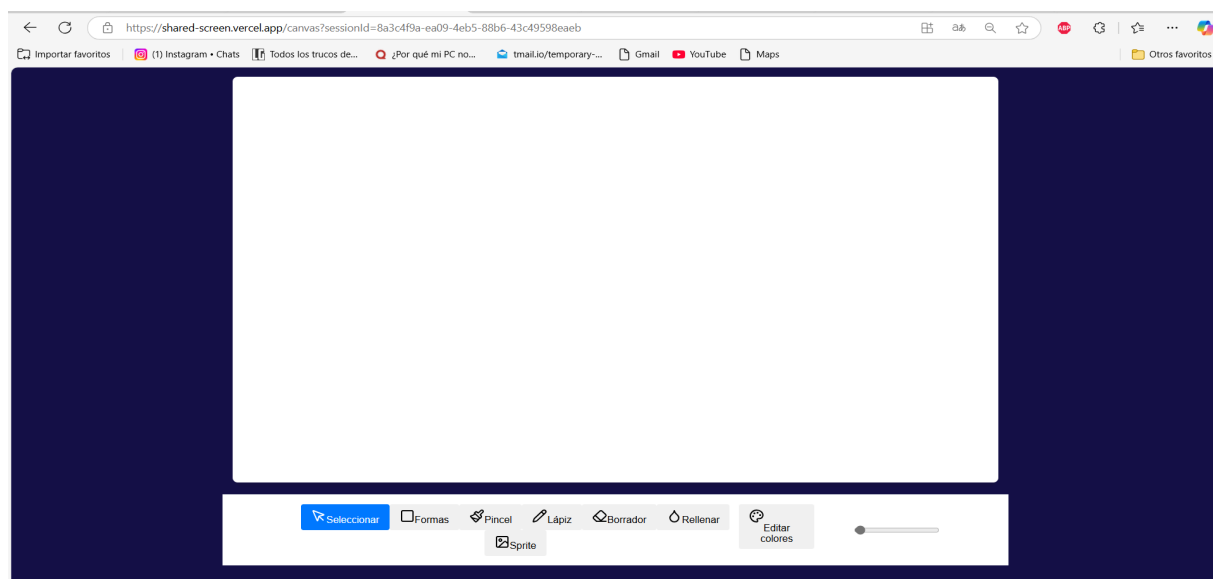


PAGINA PRINCIPAL (Segundo plano)



(ver Imagen 2)

PAGINA PRINCIPAL



(Ver imagen 3)

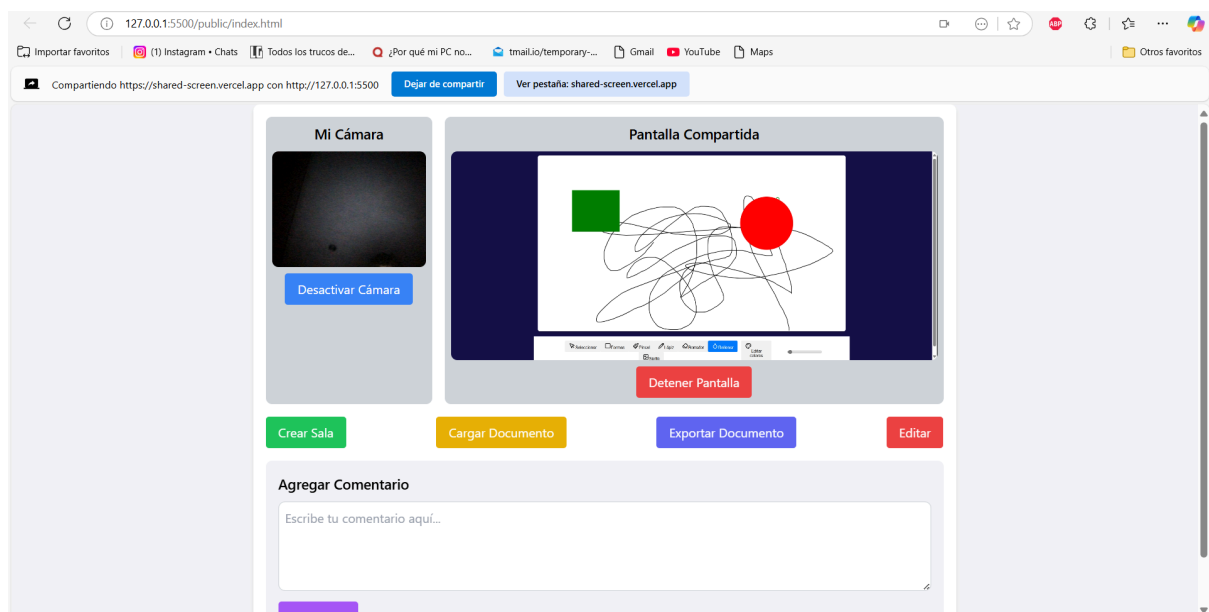
Fase 2

Funcionalidades

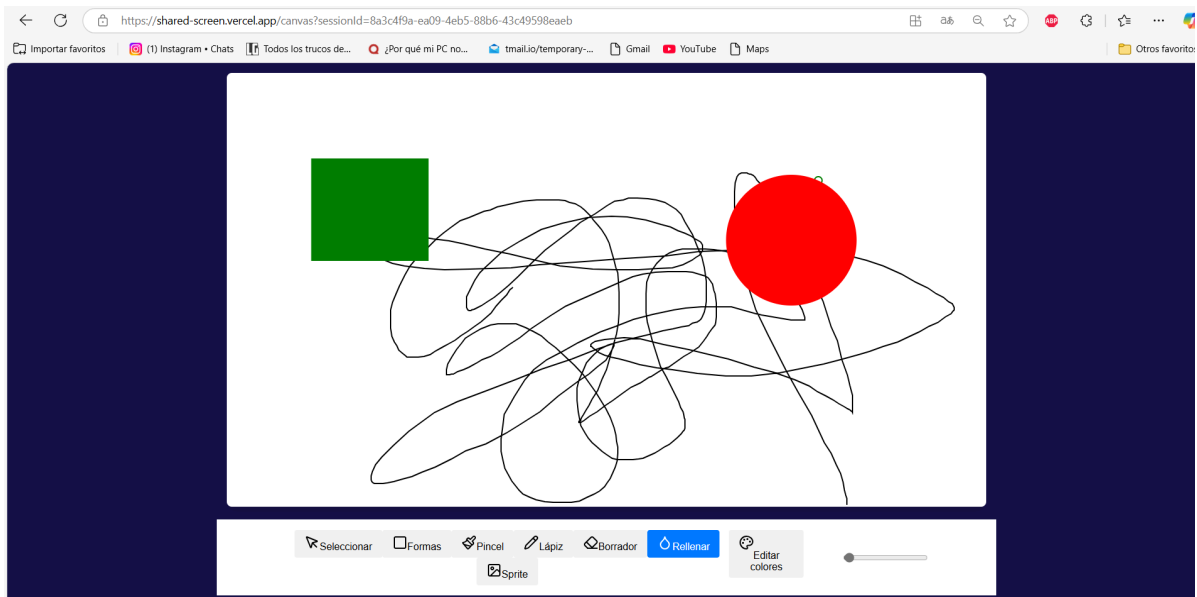
- Al abrir la primera pantalla esta da la opción si queremos encender la cámara la cual podemos observar en la imagen (ver imagen 4) igual también transmitir lo que se está haciendo en el banco de trabajo de la página principal.
- en la parte inferior se puede comentar y quedan guardados los mensajes para que los demás que están en la transmisión puedan verlo
- Todas las herramientas como las que tienen paint funcionan(relleno, seleccionar, colores, formas, lápiz, pincel, borrador) se puede adjuntar a la pantalla una imagen directamente desde el ordenador
- se puede trabajar colaborativamente con otra persona desde la misma pantalla con el mismo internet en el mismo servidor

No funcionales

- se cambió la barra lateral por la inferior
- no se codificó para el cargue del micrófono
- silenciar sonido no se implementó



(ver imagen 4)



(ver imagen 5)

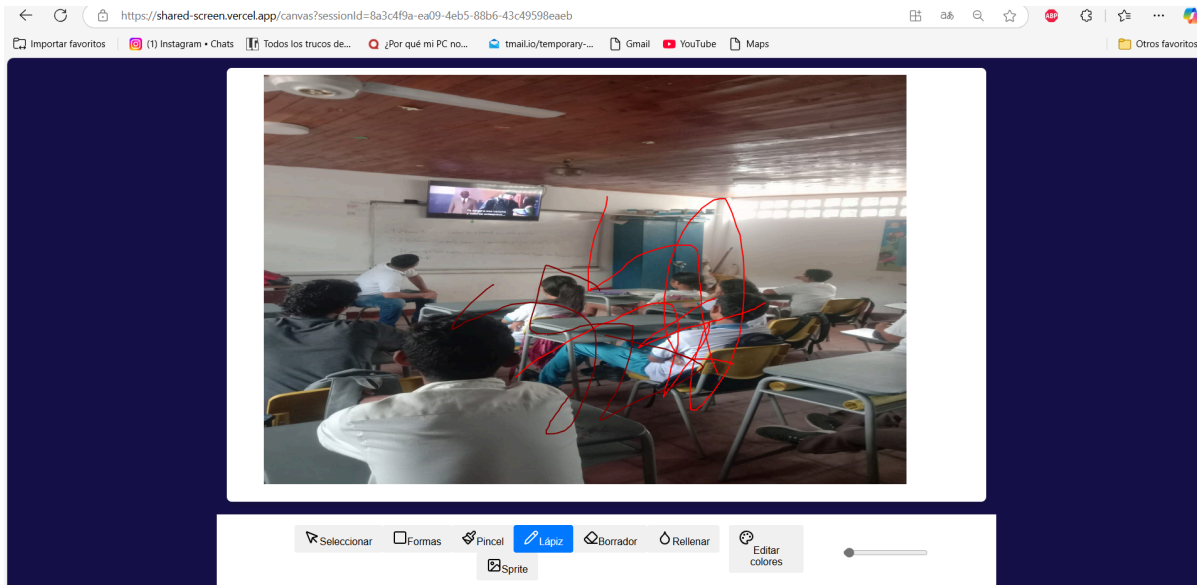
fase 3

Funcionalidades

- se puede ampliar la imagen de las fotos y reducirlas (ver imagen 6).
- se puede colorearlas o pintarlas (ver imagen 6).

No funcionales

- no se puede poner texto.
- no se puede recortar Imagenes



(ver imagen 6)

Referencias

- Beck, K., Beedle, M., van Bennekum, A., et al. (2001). *Manifesto for Agile Software Development*. Agile Alliance.
- Laurillard, D. (2012). *Teaching as a Design Science: Building Pedagogical Patterns for Learning and Technology*. Routledge.
- Myers, G. J., Sandler, C., & Badgett, T. (2011). *The Art of Software Testing*. Wiley.
- Gollmann, D. (2011). *Computer Security*. Wiley.
- Ellis, C. A., Gibbs, S. J., & Rein, G. (1991). *Groupware: Some issues and experiences*. Communications of the ACM, 34(1), 39–58.
- Gutwin, C., & Greenberg, S. (2002). *A descriptive framework of workspace awareness for real-time groupware*. Computer Supported Cooperative Work (CSCW), 11(3), 411–446.