Proyecto “Forgotten History”

Dylan Samuel Cantillo Arrieta - Director de Pruebas

Luisa Maria Escobar Linero - Directora de UI/UX

Luis Felipe Evilla Rodríguez - Director de Gerencia

Mariana Oquendo Hernández - Directora de Diseño

Juan Pablo Prada Mejía - Director de Documentación

Director Pedro Wightman

Diseño con fines académicos (no comercial) para proyecto



Departamento de Sistemas

División de Ingenierías

Universidad del Norte

Barranquilla, Colombia

09/06/2020

Resumen Ejecutivo

Forgotten History está realizado bajo un paradigma de programación modular. Es un videojuego de rol en 2d que busca visibilizar a mujeres que no han tenido el reconocimiento que merecen por su aporte a la humanidad. Para cumplir con esta labor de enseñanza, el juego cuenta con objetos o ‘logros’ que contienen una breve información acerca de la vida de una variedad de mujeres cuyo legado ha impactado la sociedad, aún desconociendo quién es en verdad la persona detrás del descubrimiento. Es una manera didáctica de aprender acerca de ellas, fomentando la investigación.

Con respecto al código fuente, los desarrolladores pudieron darse cuenta de que la ejecución de este proyecto iba a requerir de una gran exigencia de responsabilidades. Las cuales eran casi imposibles de manejar si se realizaba el código mediante la programación procedimental. Esto generó en ellos la consideración del uso de clases, objetos y métodos que ayudaran a la comunicación entre clases. Para así tener obtener un modelo más ordenado y fácil de manipular más adelante. Además de estas ventajas pensaron en la posibilidad de extender su núcleo de trabajo si el juego resultaba exitoso, para lo cual sería factible que aquellos personas que quisieran ser parte este proyecto les fuese fácil entender el código y así poder evolucionarlo o encontrarle errores producidos por los usuarios.

El juego está organizado en paquetes de clases relacionadas. Cada una de estas clases posee una serie de funcionalidades y conexiones con otras mediante métodos. Además, se da uso a conceptos fundamentales de la programación modular como encapsulamiento, para restringir la información y así asegurar el acceso, herencia para definir clases que comparten atributos y métodos, y los distintos usos de polimorfismo para hacer uso de la herencia de la mejor manera. También se hizo uso de múltiples librerías de las que Java dispone, las cuales proporcionaban herramientas necesarias para la ejecución del juego. Para poder verificar que el código funcionara se realizaron numerosas pruebas con el uso de JUnit .

Este juego generará una ola de información y surgirán una variedad de dudas en cada uno de los jugadores al darse cuenta de que la historia que siempre les han mostrado no es tan acertada como parece, descubrirán un gran bagaje de información nunca antes expuesto y los llevará a cuestionarse por qué la educación se lleva de cierta manera, ocasiona el origen o el desarrollo a la idea de que la mujer es tan capaz como el hombre y se dejará parcialmente de lado el pensamiento machista anticuado de tiempos anteriores.

Objetivos

**Objetivo General:**

Visibilizar la historia olvidada de algunas mujeres en la historia de la humanidad e intentar desvincular el pensamiento que el hombre es superior a la mujer..

**Objetivos Específicos:**

Proporcionar, a pesar de las falencias del sistema educativo actual y desigualdad de género, un medio de diversión y aprendizaje sobre la importancia de la mujer.

Aprender a programar orientado a objetos.

Aprender las nociones básicas de un videojuego en el lenguaje de programación Java.

Referentes de Soluciones Existentes

No es la primera vez que se presenta la situación en la que cada vez que se habla de historia son más los nombres de hombres que aparecen o llegan a tomar parte de la conversación. Sin embargo, hay muchas mujeres que han cumplido un papel importante en ella. Para darle solución a esta ignorancia y darle reconocimiento a estas grandes mujeres se han creado juegos tales como “Who ́s she”: un juego de mesa basado en el conocido juego “Quien soy?” pero enfocado a mujeres que han tenido importancia a lo largo de la historia. Otro ejemplo similar es “Herstóricas Pioneras”, el cual es un juego donde se puede conocer a algunas de las mujeres que se abriendo camino en muchos de los ámbitos que hasta el momento habían sido vetados para ellas por el simple hecho de ser mujeres.Y por último en esta lista de soluciones tenemos Luana Games: este gran juego fue creado por Anouk Charles y Benoit Fries en la búsqueda de modelos a seguir para su hija de 7 años, Luana. Se les ocurrió la idea de crear una baraja de cartas en las que se vieran plasmadas diferentes personalidades femeninas que han logrado grandes cosas y que no figuran en los libros de texto. Otro referente es el libro “Cuentos de buenas noches para niñas rebeldes: 100 historias de mujeres extraordinarias” escrito por Elena Favilli y Francesca Cavallo, que surge como una iniciativa para empoderar a niñas y mujeres por medio de cortas biografías y retratos de mujeres que se desempeñaron de manera excepcional en diversos campos profesionales.

Alcance

El videojuego va dirigido a personas en cualquier rango de edad que tengan nociones básicas del uso de un videojuego y puedan leer y entender español, también deben tener acceso a un computador que pueda ejecutar la versión de escritorio del juego y permita reproducir audios en formato mp3. No debe ser un equipo especial, ya que es un producto multiplataforma, puede ser usado en Windows, MacOS y Linux, además, no cuenta con gráficos altos, por lo que es compatible con diferentes resoluciones de pantalla.

El videojuego está desarrollado con el lenguaje de programación Java y no emplea ningún motor de desarrollo. El código fuente es orientado a objetos, emplea una metodología ágil y eficaz y tiene comentarios para facilitar la programación. Forgotten History es un diseño con fines académicos (no comercial) para el proyecto ‘WStem’ que se distribuye con una licencia Open Source y sin ánimo de lucro.

Metodología

**Pasos Metodológicos:**

**I) Descripción del proceso:** Lo primero que se estableció antes de empezar a realizar el proyecto fue la designación de tareas donde cada integrante tenía que responder con una parte del trabajo, luego se establecieron fechas de resultado donde se podía ver el avance en cada una de las ramas del proyecto. Después de tomadas estas decisiones se propuso usar una página web, GitHub, que utiliza el sistema controlador de versiones Git, para poder estar al día con las nuevas actualizaciones y así cada uno poder trabajar desde casa. Se implementó la programación modular como paradigma de desarrollo, hecho que facilitó la realización del algoritmo. Otro de los pasos importantes fue la investigación y aprendizaje parcial de programas los cuales realizaban tareas específicas como: creación de tiles, Diagramas de Gantt, Git, sprites, etc. Luego de lo establecido se empezó a trabajar semana a semana y se tenían una y a veces dos reuniones por semana para estar viendo que se había conseguido y en que se tenían fallas. Esto lo fuimos plasmando en un diagrama de Gantt el cual es un indicador del desarrollo del proyecto, también se recurrió a las diferentes herramientas educativas que nos brinda tanto la universidad como la internet en temas de fuente de información .

**II) Resultados:** Durante el proceso se tuvieron bastantes resultados. Se logró realizar un aprendizaje en conjunto de muchas de las temáticas aprendidas a lo largo de la estancia en la universidad como la programación procedimental en el lenguaje Java, en el cual se perfeccionó el uso de estructuras así como de librerías para realizar el proyecto. Esto se puede ver con la implementación del sonido, imágenes, tiempo de ejecución y pruebas realizadas a los métodos con Junit. El uso de github también se perfeccionó demasiado porque era muy necesario para la realización del juego ya que esta herramienta brinda un repositorio del cual todos los participantes tienen acceso y pueden hacer distintas modificaciones de un mismo proyecto en este caso forgotten History y se logró la uniformidad y poca pérdida de información . Y por último y más importante se logró realizar el juego gracias a la ganancia de conocimiento con respecto a la programación orientada a objetos (POO) En este vimos el gran alcance de este paradigma y cada semana se fue adquiriendo más conocimiento sobre las nociones de este estilo de programación como el encapsulamiento, la herencia y el polimorfismo y los diferentes métodos de las clases y/o objetos.

**III) Recursos utilizados:**

* Lenguaje de programación Java
* Git
* GitHub
* Google Meet
* Google Drive
* Google Docs
* PowerPoint
* Youtube
* Api de Java
* NetBeans (IDE)
* Excel

**Diagrama de Gantt**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Actividad |  | | | | | | | | | | | | | | | |
| Febrero | | Marzo | | | | Abril | | | | Mayo | | | | Junio | |
| Semana | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 |
| Investigación del tema a tratar (Mujeres en la ciencia) |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Aprender las nociones básicas de hacer un videojuego |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Definición del problema |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Alternativas de solución |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Evaluación y selección de la solución |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Propuesta de diseño |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Implementación de la propuesta |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Desarrollo de elementos gráficos |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Realización de pruebas |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Informe final |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Sustentación del proyecto |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Referencias y Bibliografía

Audio Library – Música para creadores de contenido . (2020a, junio 8). Calm Music | Audio Library. Recuperado de <https://www.youtube.com/playlist?list=PLzCxunOM5WFIRlYt5KJngVDXt9uIqX_1B>

Bloch, M. (s. f.). 1001 Fonts · Free Fonts Baby! Recuperado 8 de junio de 2020, de <https://www.1001fonts.com/>

COMPUTRABAJO . (2014). Computrabajo Colombia. Recuperado de <https://www.computrabajo.com.co/salarios/programadora>

COMPUTRABAJO . (2014). Computrabajo Colombia. Recuperado de <https://www.computrabajo.com.co/salarios/gerente>

COMPUTRABAJO. (2014). Computrabajo Colombia. Recuperado de <https://www.computrabajo.com.co/salarios/desarrollador-java>

CREATYPE STUDIO. (s. f.). Stadella | dafont.com. Recuperado de

<https://www.dafont.com/es/stadella.font>

[Fotografía]. (s. f.). *Jocelyn Bell Burnell*. Recuperado de

<http://www.heroinas.net/2017/01/jocelyn-bell-burnell-la-astrofisica-que.html>

[Fotografía]. (s. f.). *Katherine Johnson*. Recuperado de

<https://images.app.goo.gl/BfGL8n83tD52FEcP9>

[Fotografía]. (s. f.). *Dorothy Vaughan*. Recuperado de

<https://images.app.goo.gl/dU6jomNWcf5PeBj68>

[Fotografía]. (s. f.). *Huelga Islandia*. Recuperado de

<https://images.app.goo.gl/SfxX5TBDt5nQCKhJ9>

[Fotografía]. (s. f.). *Huelga Islandia 2*. Recuperado de

<https://images.app.goo.gl/qhRP1ntq66Zjo7v69>

[Fotografía]. (s. f.). *Ada Lovelace*. Recuperado de

<https://images.app.goo.gl/MRtcNMW6XCnvbtmHA>

[Fotografía]. (s. f.). *Mary Jackson*. Recuperado de

<https://images.app.goo.gl/UaaYmZKrpShVCr7t6>

[Fotografía]. (s. f.). *Pixel Art ADN*. Recuperado de

<https://images.app.goo.gl/zDEGHHs6joWVtnB97>

[Fotografía]. (s. f.). *Pixel Art Estrella*. Recuperado de

<https://images.app.goo.gl/6QYKX4cF7FMpF2WC9>

[Fotografía]. (s. f.). *Pixel Art Luna*. Recuperado de

<https://images.app.goo.gl/5HoP8xSsw6PSBYhcA>

[Fotografía]. (s. f.). *Pixel Art Polilla*. Recuperado de

<https://www.shutterstock.com/search/pixel+art+bugs>

Java Dev One. (2014, Abril 32). Aprender a hacer un juego en java. Recuperado de <https://www.youtube.com/channel/UChbgp4v5VDtySNOOU0JinSA>

Most Popular Art of AllG Time. (s. f.). Recuperado de <https://opengameart.org/popular>

PELS, S. (s. f.). THE BOLD FONT | dafont.com. Recuperado de

<https://www.dafont.com/es/the-bold-font.font>

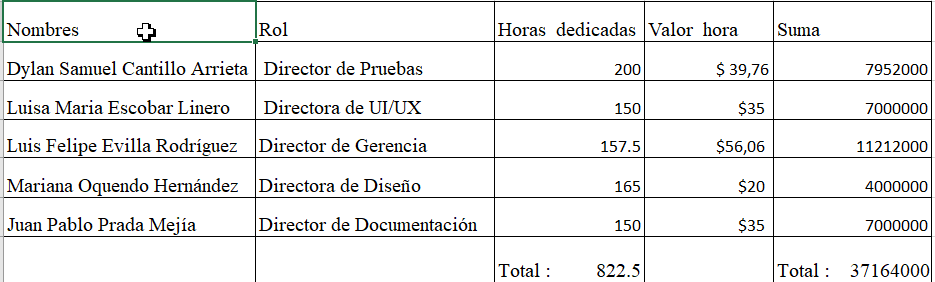
Pixelshell. (2005). Freesound - Freesound. Recuperado de <https://freesound.org/>

POWERPOINT UNIVERSITY. (2020, Febrero 8). PowerPoint tutorial No 320 How to make Gantt Chart easily in PowerPoint. Recuperado de <https://www.youtube.com/watch?v=wp0vr6OkW8Y>

W-STEM. (2020). W-STEM Project – Erasmus+. Recuperado de <https://wstemproject.eu/es/inicio/>

**Presupuesto**

**Personal:**

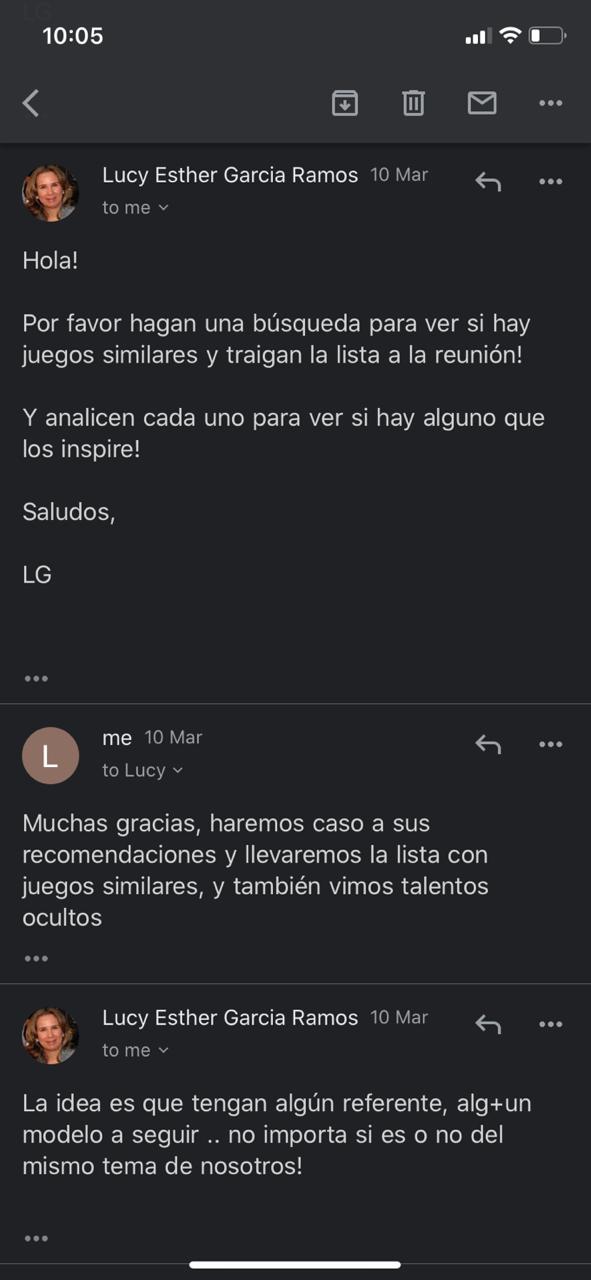
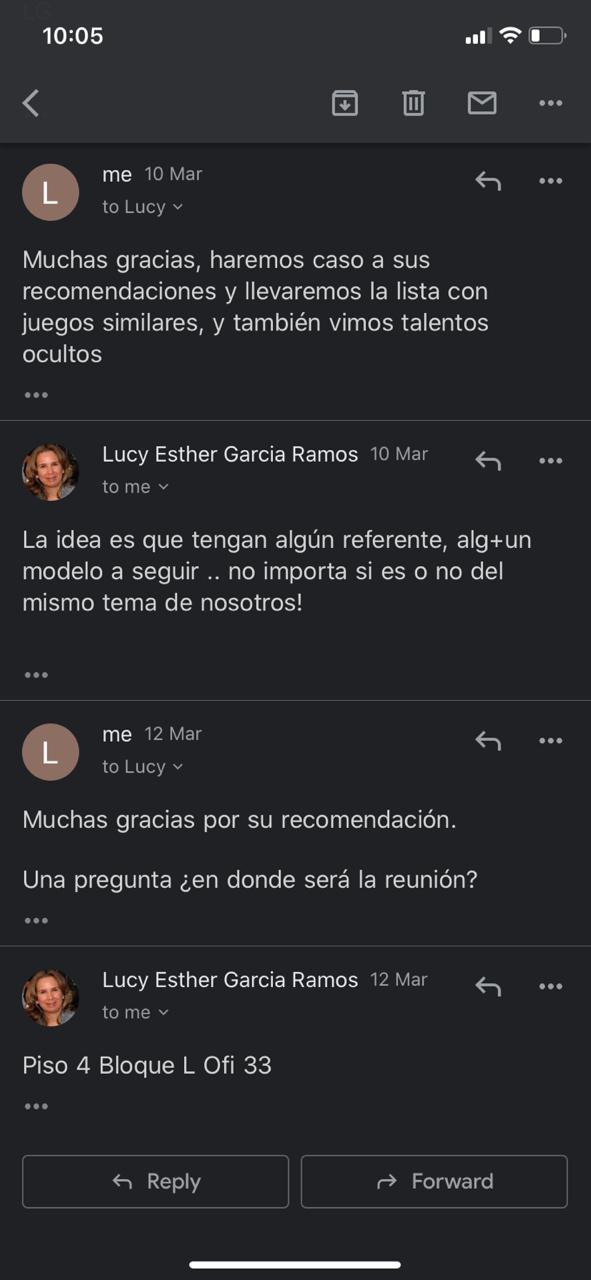
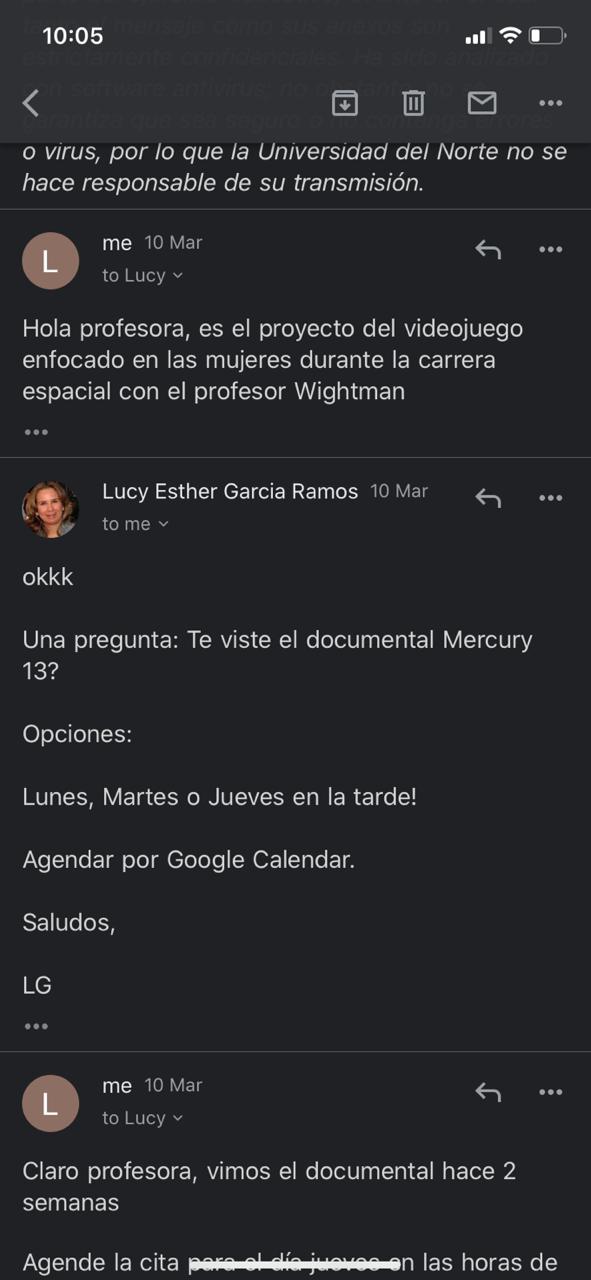
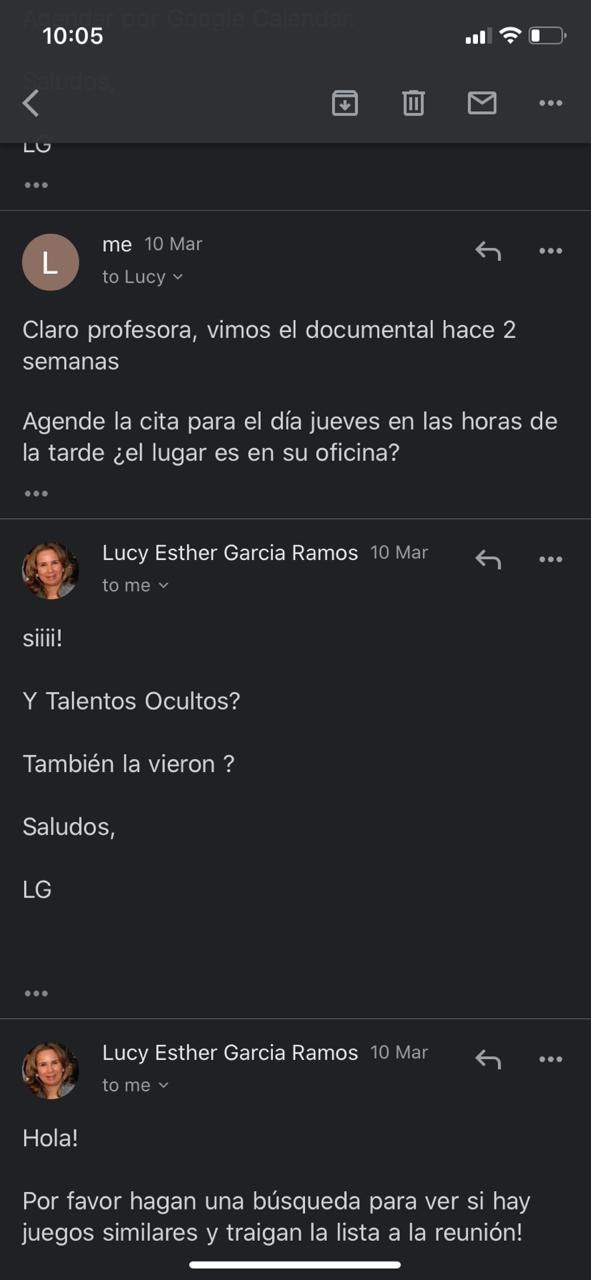
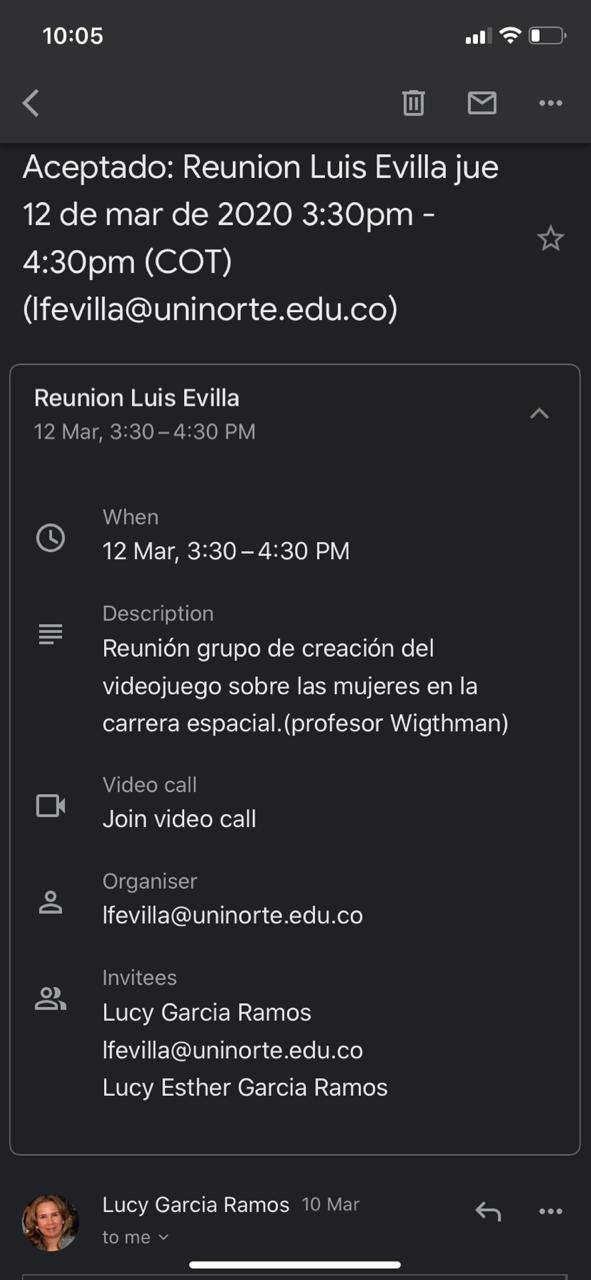


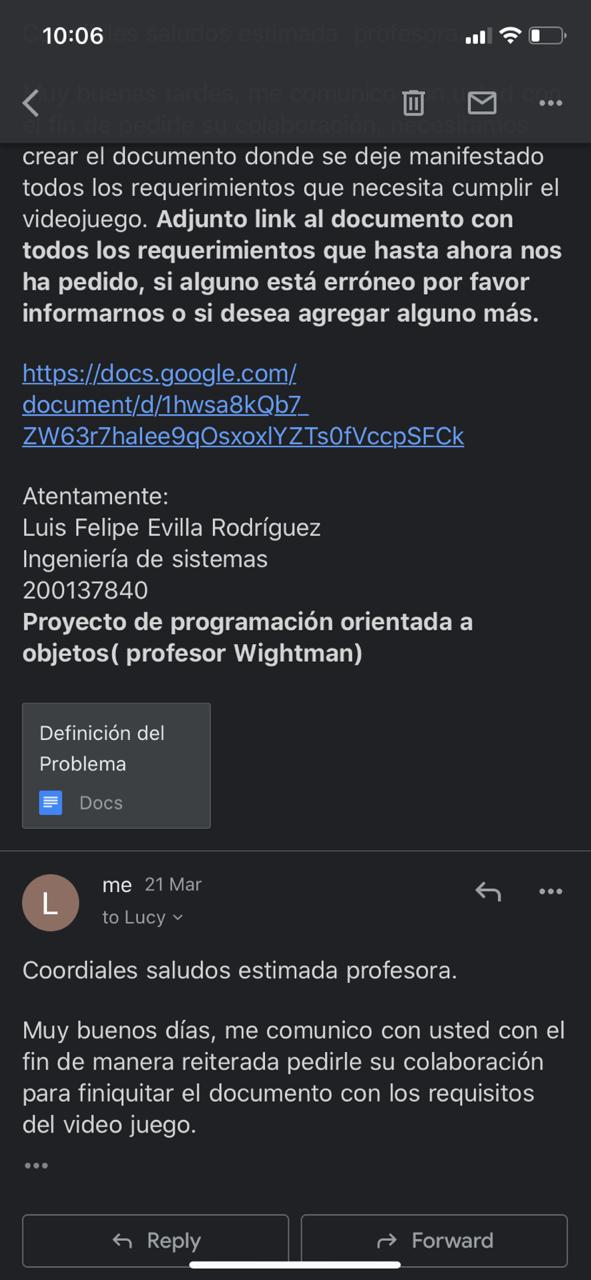
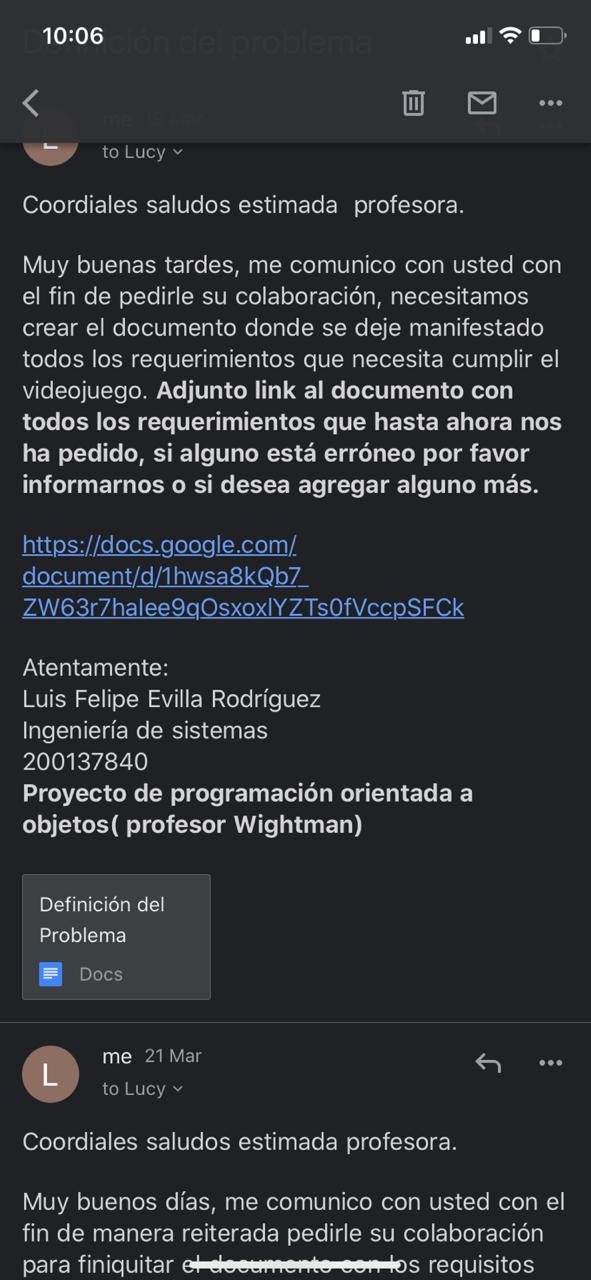
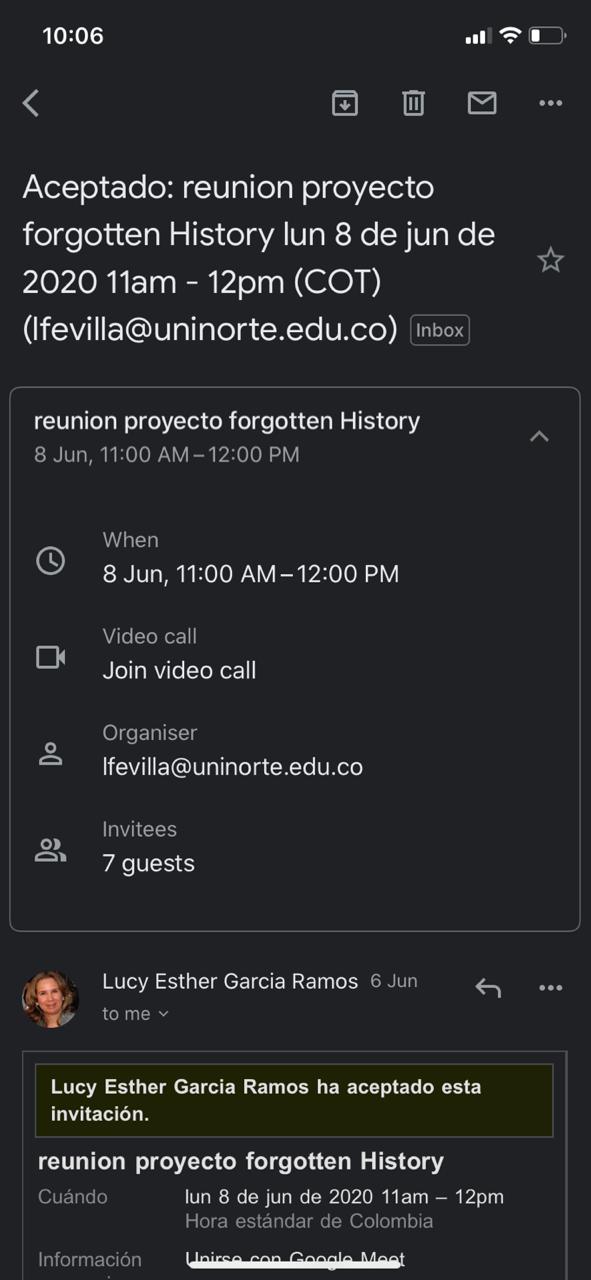
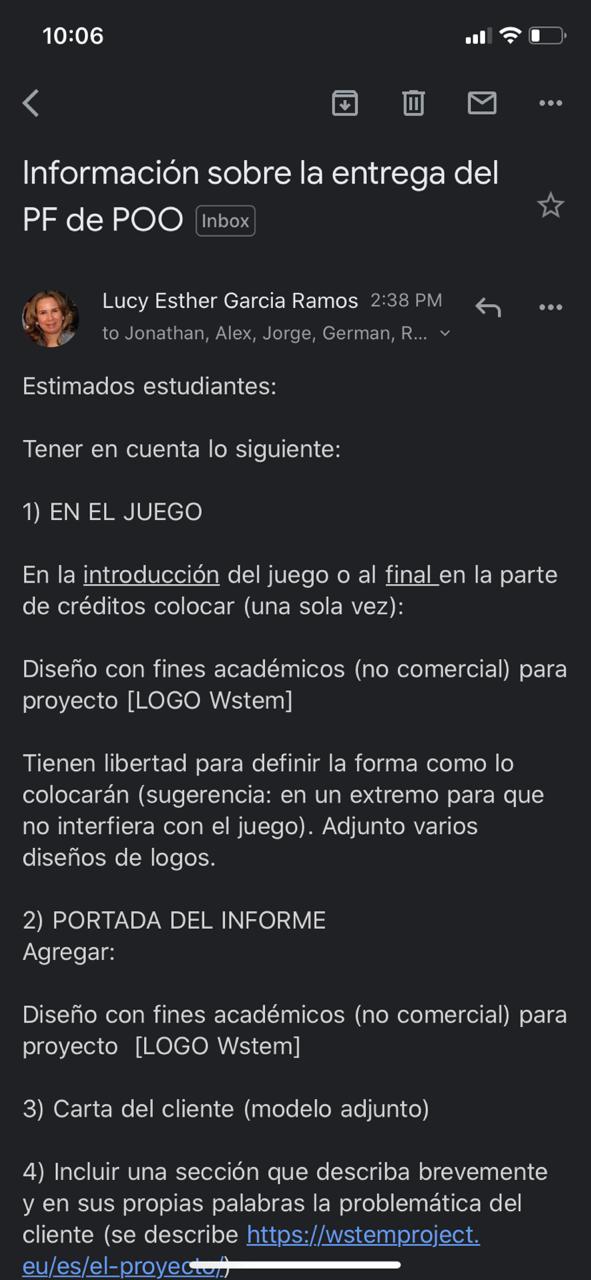
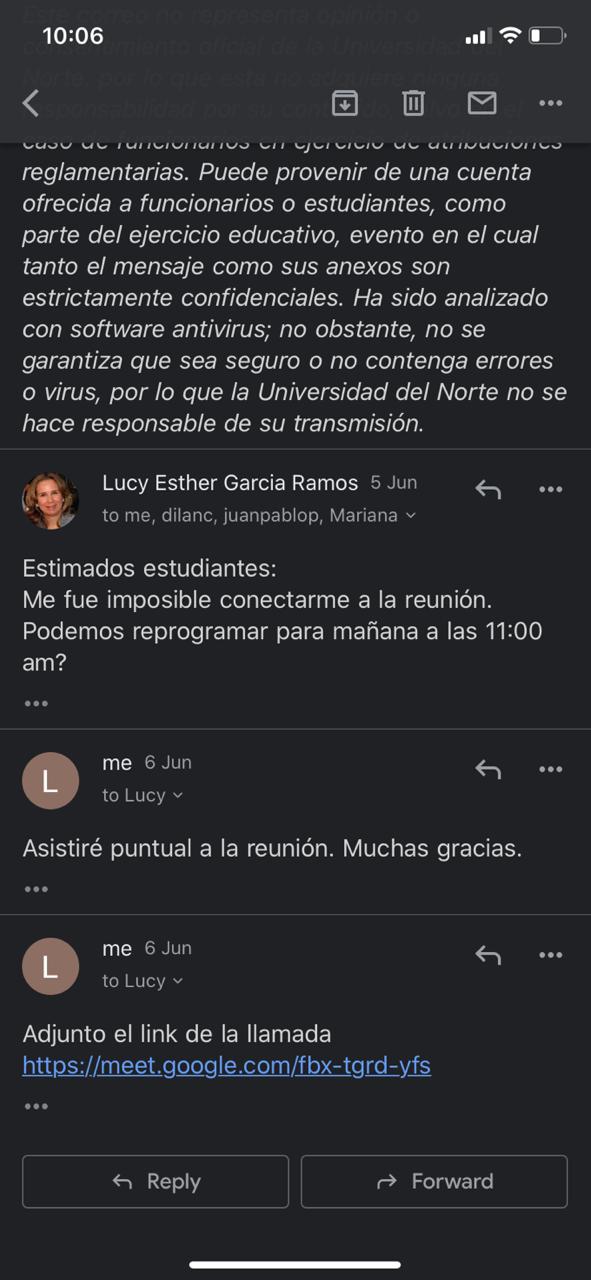
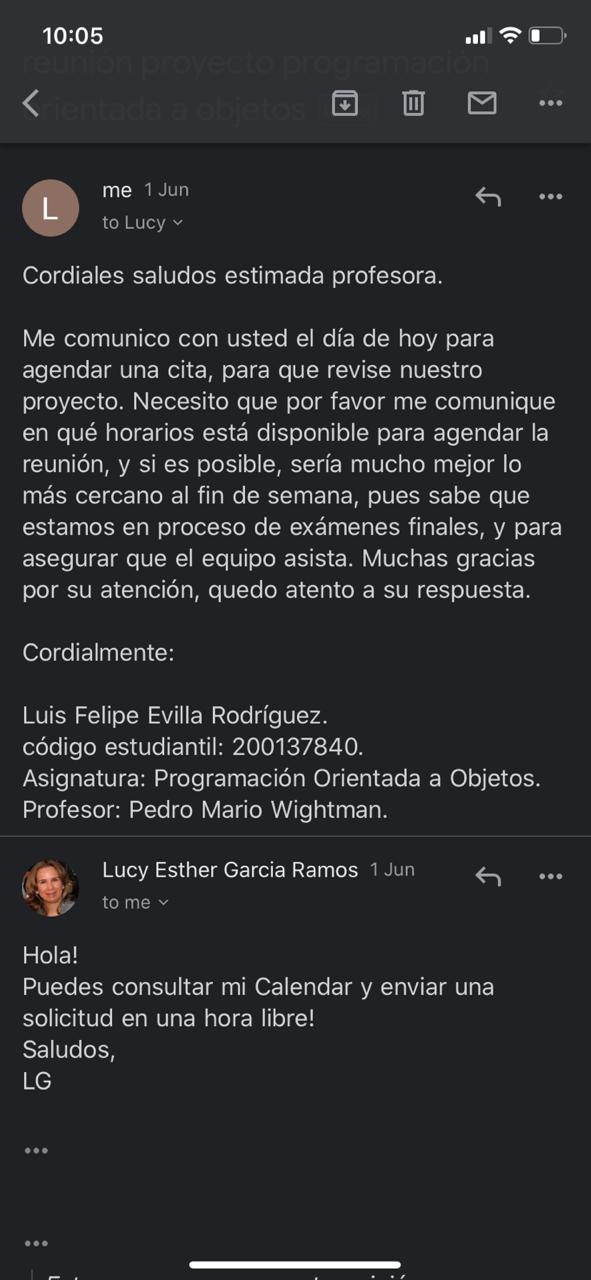
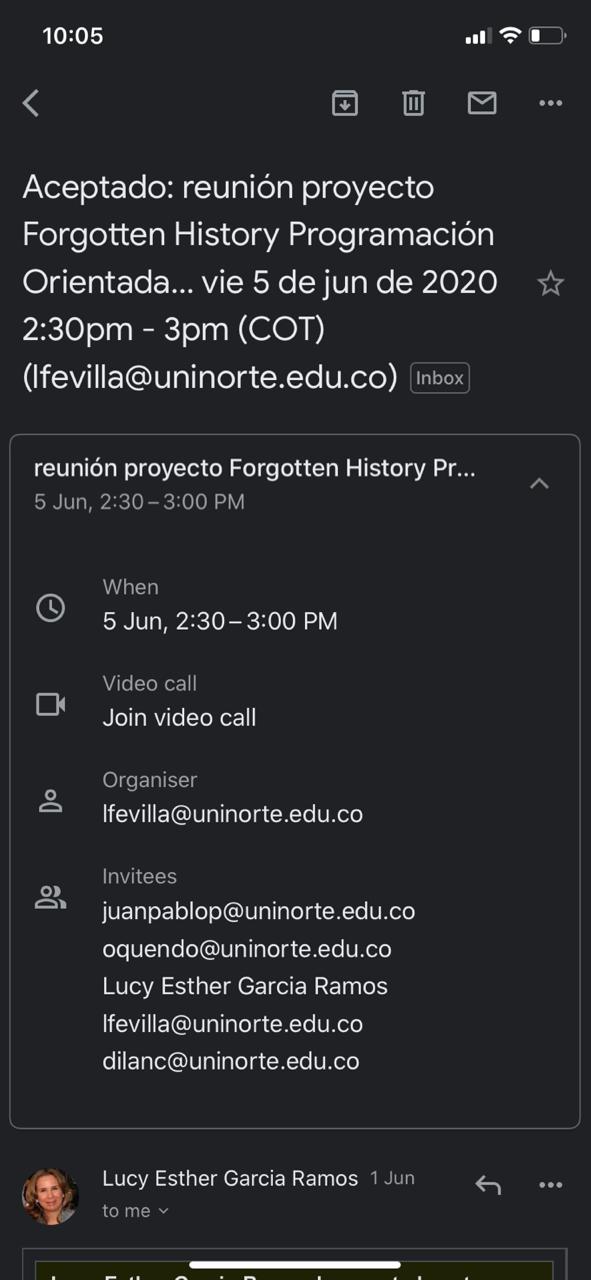
**Equipos:**



Gerencia

**Evidencia de comunicación con el cliente**

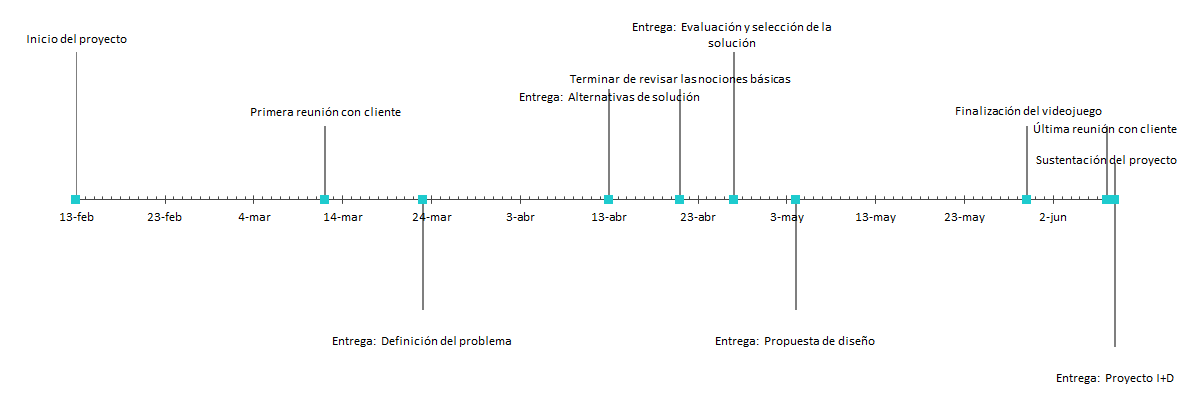




**Carta del cliente con percepción general del proyecto**

Documento ‘Carta PF POO’ en la carpeta Anexos.

**Bitácora de cumplimiento de cronograma – hitos del proyecto**



**Alternativas de solución:**

1. Un juego tipo metroidvania en el cual el usuario coleccionará cartas y las mejorará. Estas cartas contendrán mujeres históricas que tendrán distintas estadísticas, tales como vida, “poder histórico” y stamina. Estas cartas se dividirán en distintos tipos de rareza las cuales estarán basadas en su ámbito histórico, ya sea político, social, científico, etc. La idea es luchar a medida que se va pasando de mundos con el poder de las cartas. Conforme el usuario avanza se aumentaría la dificultad de los enemigos, añadiendoles vida, daño y poder de evasión (serán inmunes a ciertos poderes de algunas cartas). Como el fin de la creación del juego será mitigar la desinformación de las personas en cuanto a la participación de las mujeres en la historia, cada vez que se consigan las cartas se presentará un pequeño lore (historia) para que al tener conocimiento de este, se le relacione con el poder de la carta.
2. Otra alternativa para darle solución a esta problemática es la creación de un juego digital basado en una serie de rompecabezas los cuales tendrán como imagen final, es decir, al desarrollar el puzzle, una imagen informativa de una mujer en particular evidenciando su aporte a la humanidad y el entorno donde se encontraba en dicho momento. Esto último para entender en qué época apareció esta importante personalidad. Dicho esto, la meta del usuario deberá ser completar todos los rompecabezas generados por esta plataforma y así asegurarse de ampliar su gama de conocimientos de tipo histórico.
3. La última opción de solución es un juego de rol, donde el personaje, controlado por el jugador, debe vencer a sus enemigos sin que estos lo maten, recolectando en el camino items que le permitirán acceder a la biografía de una mujer que hizo historia y por su género no recibió el reconocimiento que merecía. Contaría con un inventario donde tendrá acceso a las biografías ya desbloqueadas.

La solución escogida fue el juego de plataforma. Ya que es la alternativa que más se adecua a los requisitos, puede ser de agrado para un mayor público y su tiempo de desarrollo es menor que los otros tipos de videojuegos. Además de que existe una amplia y extensa documentación acerca de cómo crear este tipo de videojuegos.

**Criterios para elegir la solución:**

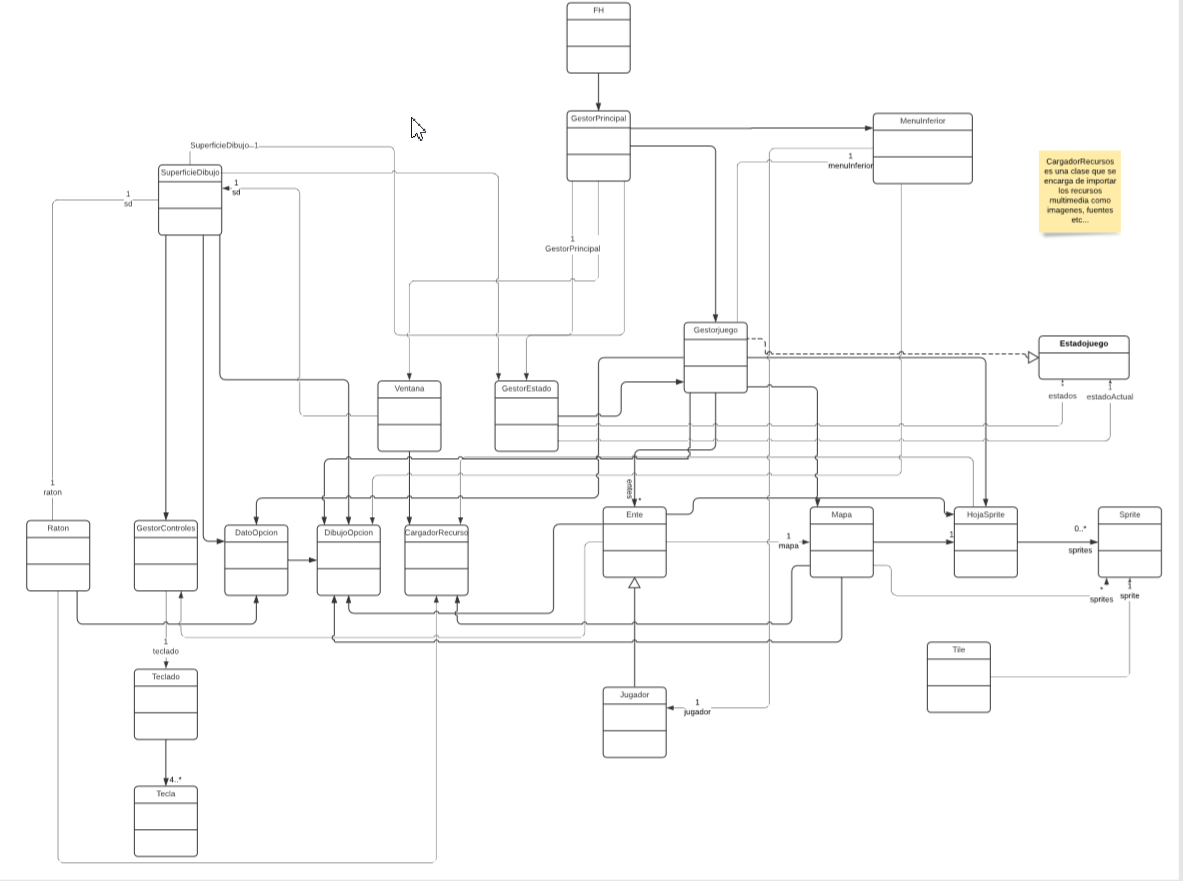
1. Facilidad y tiempo de desarrollo.
2. Disponibilidad del equipo (limitada por el distanciamiento social).
3. Requiere un menor conocimiento del desarrollo de videojuegos, y es más fácil obtener dicho conocimiento gracias a las diversas plataformas que implementan este tipo de juego utilizando el lenguaje de programación Java.
4. Es la idea más clara, concreta y con más caminos a escoger para su desarrollo.
5. La idea nos permite dar información sobre mujeres que han tenido un papel en la historia y a pesar de eso han sido condenadas al olvido.

Diseño

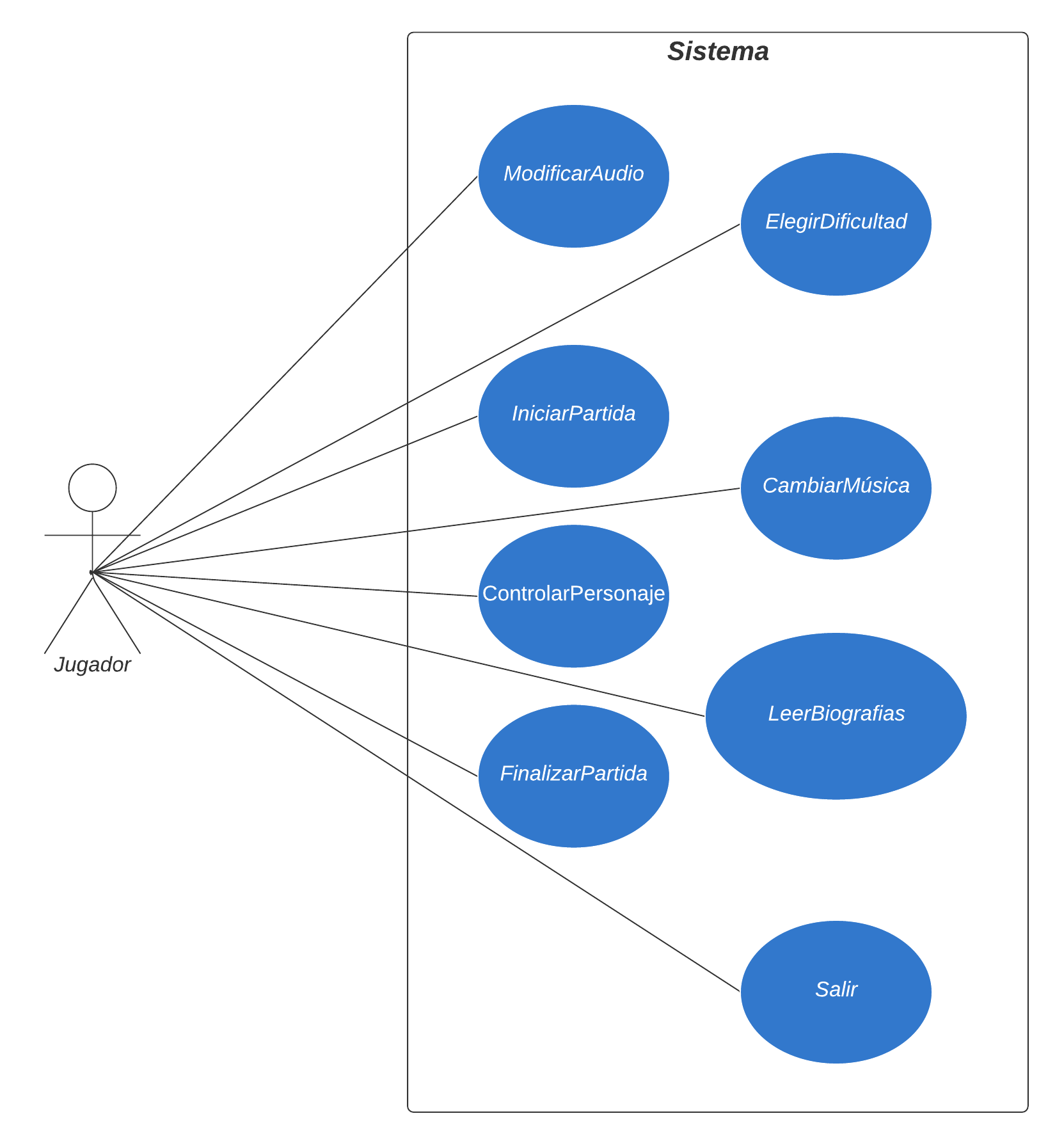
El diseño del videojuego es completamente orientado a objetos, cuenta con una gran cantidad de clases las cuales se encargan de que el juego funcione correctamente, a pesar de que todas son importantes, el videojuego cuenta con algunas clases principales o fundamentales, las cuales son:

1. **Gestores:** Como su nombre lo dice son los encargados de gestionar todo el juego, controlando la información actual y que métodos se ejecutan.
   1. **GestorPrincipal**: Contiene el método main y es el director de orquesta de toda la aplicación, al ejecutar el juego, esta clase será la encargada de invocar todos los demás gestores (sonido, controles, y estado). Posee los métodos:
      1. **iniciarJuego**(): Realiza el setup inicial del juego, instancia la superficie de dibujo (canvas), la ventana sobre la cual habitará el juego, y el gestor de estados, que es el encargado de gestionar el estado del juego(pausa, activo, menú ajustes, etc.).
      2. **iniciarBuclePrincipal**(): Es el encargado de llamar a los métodos “actualizar()” y “dibujar()” cada 1 segundo para que se alcancen las 60 actualizaciones por segundo y 60 fps.
      3. **actualizar**(): Se encarga de llamar los métodos de actualizar en tiempo real toda la información de los elementos del juego, los controles para poder actualizar el movimiento del jugador, el abrir menús, etc., y actualiza la información sobre la superficie de dibujo (en que posición se encuentran los elementos dentro de la pantalla, etc.)
   2. **GestorSonido**: Se encarga de gestionar la información sobre el sonido del juego (nivel de volumen de la música, botones, enemigos, etc.). Consta de cuatro métodos principales:
      1. **reproducir**(): Reproduce un sonido una sola vez, es utilizado para reproducir sonidos como el de los botones al presionar, o los enemigos al atacar.
      2. **reproducirBucle**(): Reproduce un sonido indefinidamente, hasta que se desactive, es utilizado para producir los soundtracks o canciones del juego.
      3. **setVolumen**(float volumen): Es el encargado de establecer el nivel de volumen del juego, ya sea de la música, los efectos, o ambos. Su principal función es comunicarse con la API de Java que reproduce los sonidos, y pasarle el valor de sonido en forma de decibeles.
      4. **aumentarVolumenJuego**(float volumen): Recibe el valor de decremento o aumento del nivel de volumen de los efectos de sonido del juego, guarda estos cambios y luego llama al método setVolumen(), para que se comunique con la API.
      5. **aumentarVolumenMusica**(): Recibe el valor de decremento o aumento del nivel de volumen de la música del juego, guarda estos cambios y luego llama al método setVolumen(), para que se comunique con la API.
   3. **GestorControles**: Se encarga de gestionar la información sobre los movimientos y la interacción del jugador con los controles del juego, ya sea para moverse (a, w, s, d), para recoger objetos (e), o abrir menús como el de pausa o inventario. El gestor actualiza la información de cuando una tecla es presionada, y luego cualquier otra clase puede solicitar esta información. Funciona a través de un arreglo de teclas, y “listenners”, los cuales actualizan el estado de las teclas a “true” cuando son presionadas o “false” cuando no, cada vez que el jugador interactúe con ellas.
   4. **GestorEstado:** Realiza los cambios de estado en el juego, para que así el jugador pueda cambiar por ejemplo de un estado de juego(actividad) a un estado de menu pausa, o cuando inicia el juego pasar del estado menú ajustes al estado de juego. Su elemento principal es una lista de estados predefinidos los cuales pueden ser cambiados, cuando el gestor principal llama al método cambiar estado de esta clase.
      1. Todos los estados del juego deben heredar de la **interface EstadoJuego**, para que así todas implementen los métodos de dibujar y actualizar.
      2. **dibujar**(): Dibuja todos los elementos del estado actual, como por ejemplo los botones de los diferentes menús, etc.
      3. **actualizar**(): Actualiza la información actual del estado, la cual depende de cada uno en especifico, ya que esta es diferente según el tipo de estado.
2. **Constantes**: Funciona como un archivo de configuración, en el cual se guardan las rutas a todos los elementos multimedia del juego, tales como sonidos, imágenes, fuentes etc… estas rutas luego pueden ser consultadas por otras clases, y así, cuando existe un recurso que solicitan muchas clases, y este se elimina o modifica, solo hace falta cambiar su ruta en el archivo de constantes y no en cada clase o elemento que lo solicite. Además también guarda información sobre constantes numéricas como el factor de escalado, el tamaño de los sprites etc… que luego también pueden ser solicitadas.
3. **CargadorRecursos**: Clase que cuenta con los métodos necesarios para cargar elementos multimedia para ser utilizados en el juego, tales como sonidos, imágenes, fuentes, textos, etc.
4. **Jugador**: Usada para controlar toda la información y acciones del avatar del usuario, como vida, resistencia etc, posee los métodos necesarios para realizar las acciones como atacar a un enemigo, eliminar enemigos cuando su barra de vida llegue a 0, mover el personaje por el mapa, comprobar las colisiones para que el personaje no pueda salirse del mapa, o atravesar objetos, etc…
5. **Mapa:** Guarda la información de los mapas del juego. La posición de cada sprite en el mapa, que objetos son sólidos, y cuáles no, los límites hasta dónde puede llegar el jugador, la decoración etc….
6. **SpriteSheet:** Con esta clase se almacena y se utiliza la información de las spriteSheets del juego, las cuales contienen la mayor parte de elementos gráficos, para posteriormente poder manipular o acceder a cada sprite por separado.
7. **Sprite:** Se encarga de almacenar y manipular la información de cada uno de los sprites, dentro de una spritesheet, para que así luego otras clases como jugador, o mapa por ejemplo, puedan solicitar sprites específicos de una spritesheet y utilizarlos.
8. **Enemigo:** Usada para controlar toda la información y acciones de los enemigos del juego, tales como sus movimientos, estado de salud, información de ataque, sprites etc…, también permite que los enemigos puedan realizar acciones como atacar al jugador.

**Diagrama de Clase**



**Diagrama de Casos de Uso**



Documentación

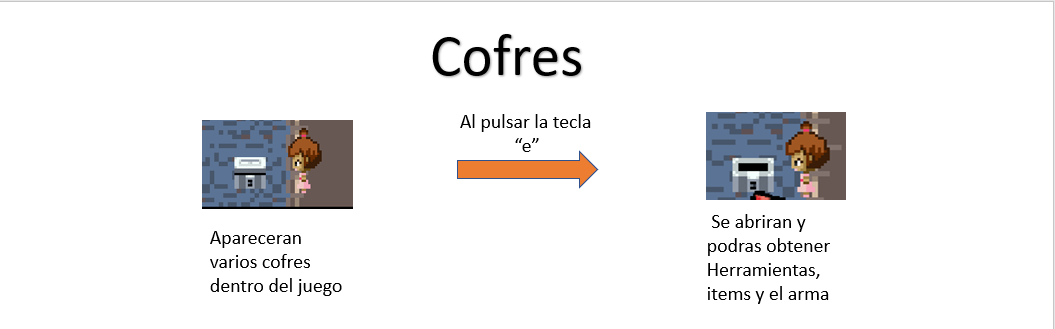
**Manual de usuario**

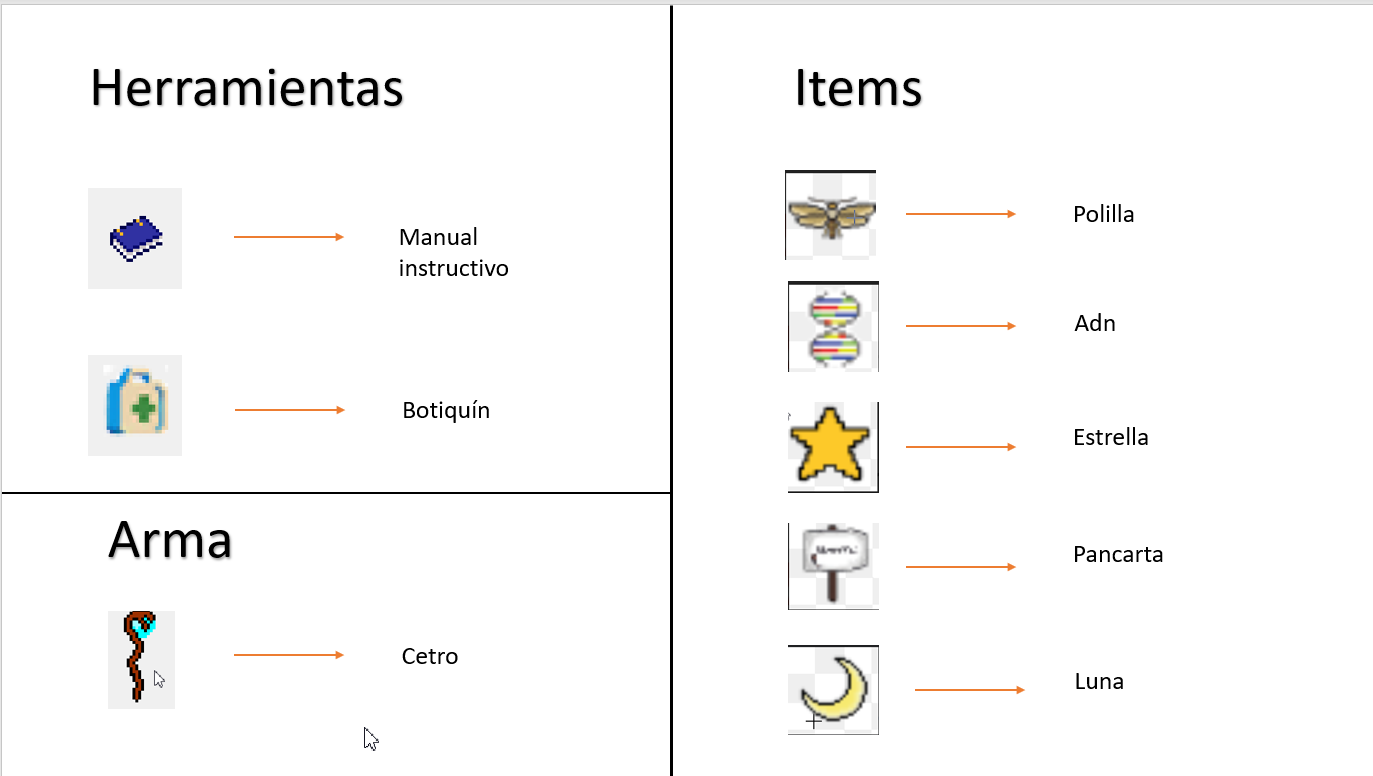


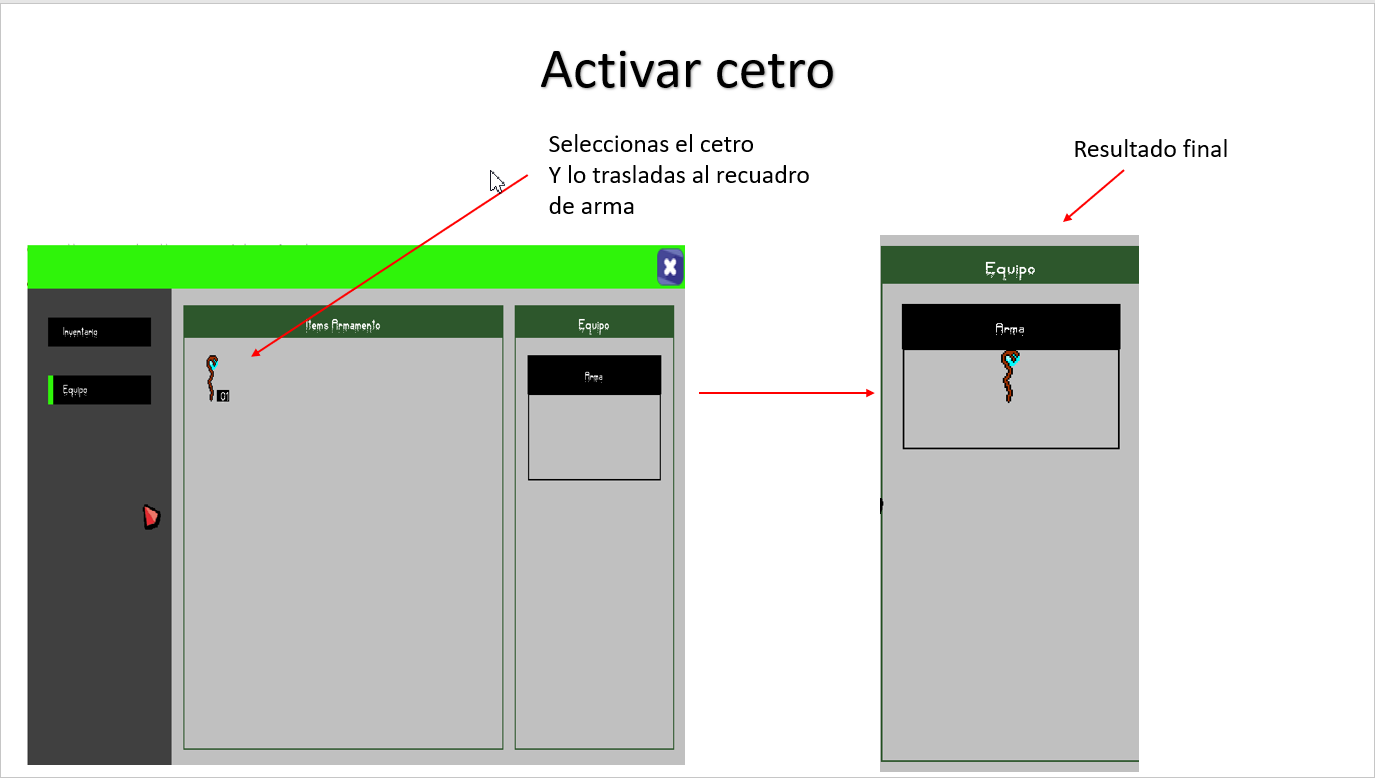


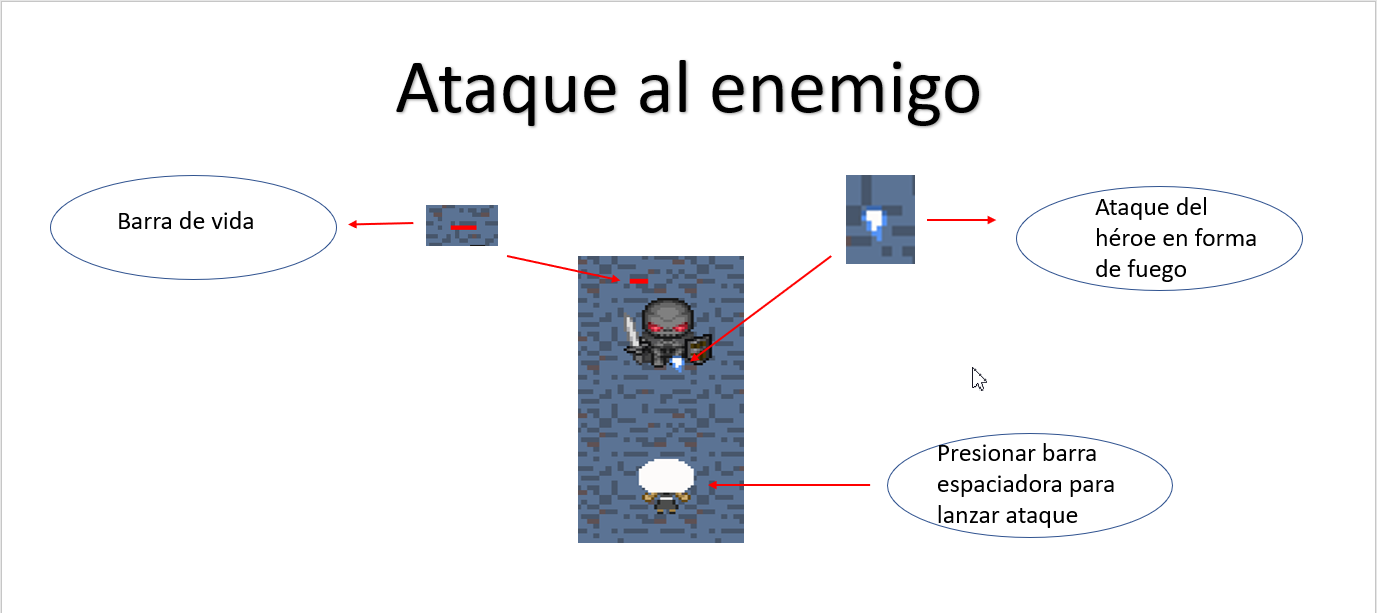
Guía de juego

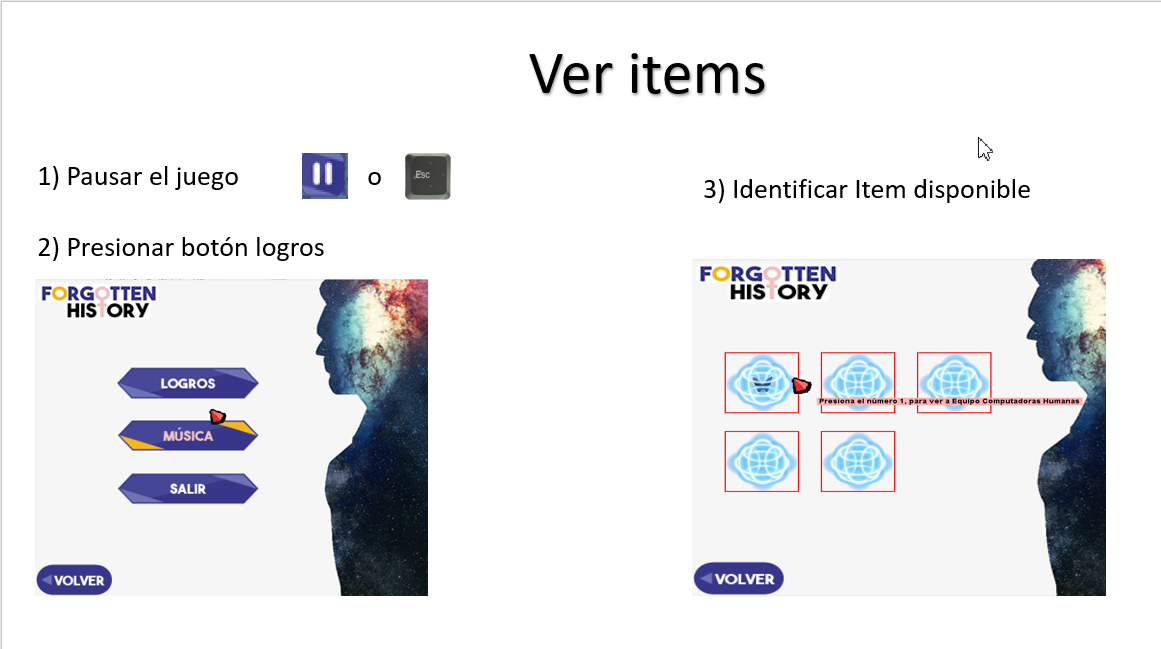


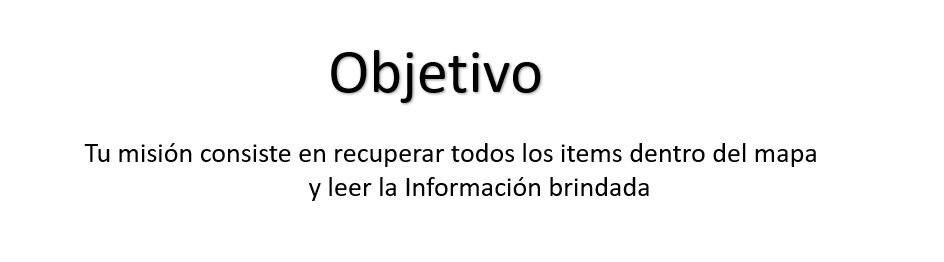












Pruebas

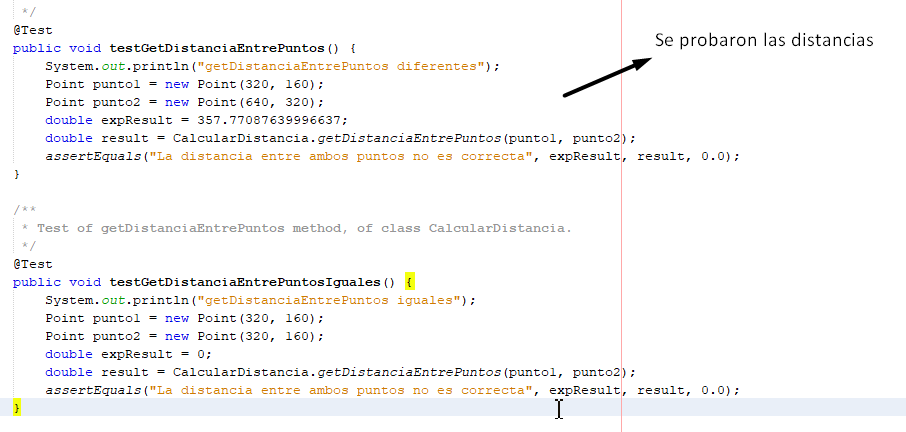
**Plan de pruebas propuesto**

Se llevó un plan de pruebas unitarias, ya que todas las clases están directamente conectadas y su principal funcionalidad es dibujar en pantalla y realizar una prueba de integración implicaría dibujar, por lo tanto la mejor forma de probar dichas clases que su principal función es dibujar, es dibujando en pantalla. Clases como “cargadorRecuersos”, “hojaSprites” son aquellas que les dan base a todas las demás, otorgándole los datos, objetos, entre otros, por lo tanto son las más óptimas para realizar pruebas, si estas funcionan, podemos asegurar que lo demás puede funcionar de manera óptima.

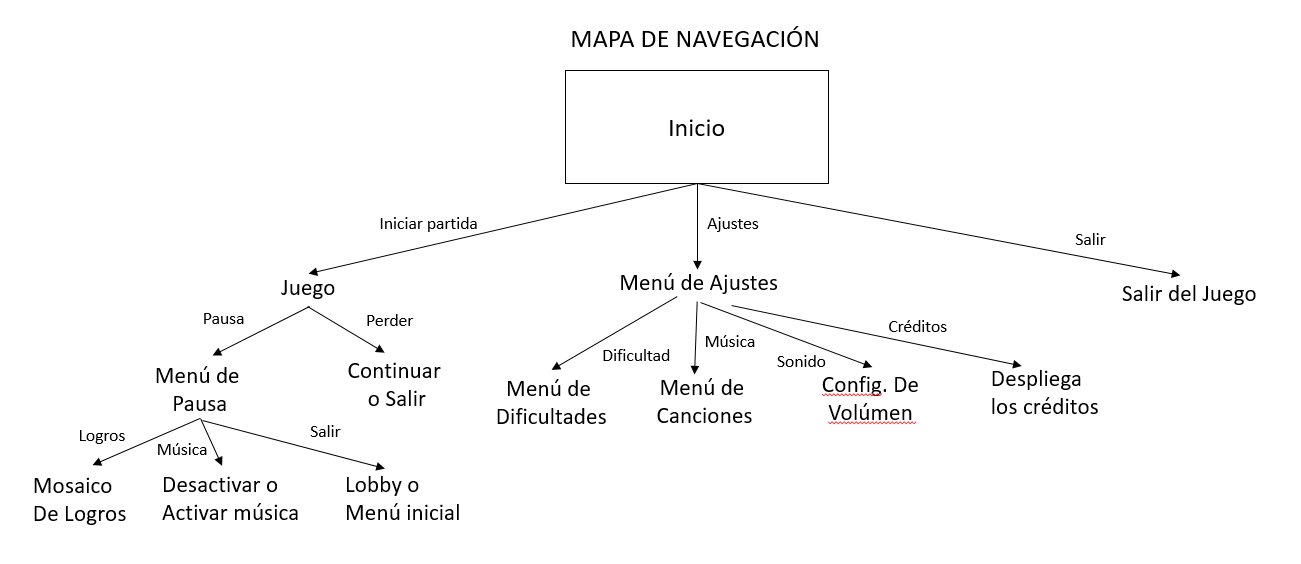
Documento ‘Plan de pruebas’ en la carpeta Anexos

**Evidencias de aplicación de plan de pruebas**





Experiencia de Usuario e Interfaz Gráfica

**Mapa de navegación de la aplicación**

**Descripción de los elementos gráficos del juego**







**Encuesta de usabilidad**

La encuesta arrojó que se realizaron el 86.11% de las tareas en el tiempo establecido y con un estado anímico estable por parte de los usuarios.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Tareas** | | | | | |
|  | Cambiar canción | Subir volumen | Cambiar dificultad | Encontrar arma | Encontrar dos cofres | Eliminar obstáculo |
| **Tiempo** | 1 minuto | | | 1 minuto | | |
| **Usuario** | **Cumplió** | | | | | |
| 1 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 2 | ✓ | ✓ | ✓ | ✓ |  | ✓ |
| 3 | ✓ | ✓ | ✓ |  | ✓ |  |
| 4 | ✓ | ✓ | ✓ | ✓ |  | ✓ |
| 5 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 6 | ✓ | ✓ | ✓ |  | ✓ | ✓ |

Conclusiones

La realización del proyecto no fue un proceso fácil, por el contrario, se presentaron una serie de complicaciones que alteraron el desarrollo. Para empezar no había seguridad de qué rol debería cumplir cada uno, además de que cada quien tenía opiniones distintas, por lo que el trabajo grupal no fue el más idílico durante las primeras semanas. Afortunadamente, con el tiempo se adquirieron mejores capacidades comunicativas que permitieron un ameno trabajo colaborativo. En lo que respecta a la realización del videojuego surgieron problemas a nivel de código: algunos de los métodos no funcionaban como deberían, o en lo absoluto, y no se lograba encontrar el porqué del error. Después de investigar se descubrió la manera de realizar pruebas a los métodos principales para encontrar el error y solucionarlo. En ocasiones alguno se abstenía de adelantar el código por la incertidumbre de si alguien más ya lo estaba haciendo, esto fue solucionado gracias a herramientas como GitHub, JUnit y JavaDoc que permitieron, junto con la mejora en la comunicación, que el resto del desarrollo se diera de manera fluida. La mayor dificultad a superar fue el repentino cambio de presencialidad a virtualidad, que limitó el tiempo posible para la implementación proyecto, no porque la entrega se haya adelantado, sino por el desafío que resultó ser acostumbrarse a la situación y aprender a manejar el tiempo de mejor manera.

El crecimiento que se dio gracias a este proyecto no fue únicamente académico sino también personal. En el proceso del desarrollo del proyecto se lograron reforzar habilidades como el aprendizaje autónomo y colaborativo y el trabajo en equipo. Además se lograron perfeccionar técnicas y estudiar programas que serán muy útiles en el resto de la carrera y la vida laboral.

Entre las posibles mejoras que podrían ser aplicadas al proyecto está la opción de configurar el idioma en inglés y hacerlo adaptable a diferentes plataformas como celulares y tablets, para así poder llegar a una población mayor y darles a conocer estas historias que suelen ser ignoradas. Permitirle al usuario crear una cuenta con su respectiva contraseña generaría una mayor conexión entre el usuario y el juego, además de dar la oportunidad de informar al jugador de próximas actualizaciones del juego. Esto también permitirá guardar la partida, lo que podría ser una gran funcionalidad, ya que las personas podrían apagar el dispositivo y luego seguir en donde lo dejaron, esto se puede hacer con el botón pausa pero esto sin apagar la consola. Incluso se podrían incluir una serie de minijuegos en forma de retos para ofrecerle al jugador una mayor versatilidad y diferentes experiencias dentro del mismo juego.