Proyecto Final: Batalla Naval.

Autores.
Laura Daniela Marín Barragan.
Laura Juliana Rodríguez Ramírez.
Miguel Andres Ordoñez Hernandez.

Universidad Nacional de Colombia, Departamento de Ingeniería.
Bogotá D.C, Colombia

Resumen - En este documento se realiza una evaluación de todos los requerimientos para la creación del proyecto: "Batalla Naval" cuyo propósito es ser un juego interactivo y funcional; se hace una revisión profunda de conceptos previos necesarios para el tratamiento del proyecto como lo son plataformas de bases de datos, trabajo en equipo, asignación de tareas y desarrollo de aplicaciones web. Así mismo, se realiza la investigación de los fundamentos para un planteamiento óptimo como el conocimiento de las reglas y modalidad del juego, de las historias de usuario y cómo plantearlas correctamente; las bibliotecas necesarias para cumplir con los requerimientos resultantes del análisis de los actores y el funcionamiento técnico de los detalles adicionales que se desean agregar.

Palabras clave - Proyecto, Programa, Python, Py Games, Historias de usuario, Requerimientos, Firestore, Firebase, Real time, Interfaz, GitHub, GitLab, Bases de datos, Juego, Batalla naval, Stackoverflow, Librerías, Jugador, Desarollo.

Abstract - In this document an evaluation of all the requirements for the creation of the project: "Battleship" whose purpose is to be an interactive and functional game is made; a deep review of previous concepts necessary for the treatment of the project such as database platforms, teamwork, task assignment and web application development is made. Likewise, research is done on the fundamentals for an optimal approach such as knowledge of the rules and modality of the game, user stories and how to raise them correctly; the necessary libraries to meet the requirements resulting from the analysis of the actors and the technical operation of the additional details that you want to add.

Keywords - Project, Program, Python, Py Games, User Stories, Requirements, Firestore, Firebase, Real time, Interface,GitHub,GitLab,Databases,Game,Battleship, Stackoverflow,Libraries,Player,Developer,Development.

I. Introducción.

A lo largo de este documento, se plantea dar respuesta a varias interrogantes importantes en lo referente a ¿qué es?, ¿en qué consisten? y demás, que serán desarrolladas a lo largo del documento, varias funcionalidades de los sistemas que se implementarán, librerías, historias de usuario y requerimientos; para inmediatamente después comenzar con el debido proceso de estructuración del primer sprint del proyecto.

II. Desarrollo.

¿Qué es batalla naval?

En el juego de batalla naval se dispone de 5 diferentes barcos. La meta del juego es hundir los barcos adversarios antes de que los propios sean hundidos. Lo primero que se debe hacer para jugar es determinar la ubicación de los barcos dentro del tablero.[1]

Para realizar un movimiento de ataque, se elige una casilla y se nombra según las coordenadas de letras de la izquierda y los números de arriba.[2]

¿Qué es firebase?

Es una plataforma en la nube para el desarrollo de aplicaciones web y móvil,tiene como función esencial hacer más sencilla la creación de aplicaciones webs y móviles, además de su desarrollo, esto en conjunto con procurar que el trabajo sea más rápido, pero que cuente con la calidad requerida, sus herramientas son variadas y de fácil uso, considerando que su agrupación simplifica las tareas de gestión en una misma plataforma. Es útil para los desarrolladores que no necesitan dedicar tanto tiempo al backend, esto en cuestiones de desarrollo y de mantenimiento.Dispone de funcionalidades como desarrollo,crecimiento y monetización,además de la analítica.[3]

¿Qué es firestore?

Es una base de datos flexible y escalable para el desarrollo en servidores, dispositivos móviles y en la Web desde Firebase y Google Cloud.Esta mantiene los datos sincronizados entre la aplicación ,el cliente y el desarrollador,todo a través de objetos de escucha en tiempo real, ofrece soporte sin conexión y una integración continua con otros productos de Firebase y Google Cloud, incluido Cloud Functions,en conjunto con lo anterior firestore está disponible en los SDKs nativos de Node.js, Java, Python, Unity, C++ y Go.[4]

Mediante el modelo de datos NoSQL,almacena los datos en documentos que contienen campos que se asignan a valores. Estos documentos se almacenan en colecciones, que son contenedores para los documentos y que se utilizan para organizar datos y compilar consultas. Los documentos contienen diversos tipos de datos como strings y números simples, hasta objetos anidados complejos. También se pueden crear subcolecciones dentro de documentos y crear estructuras de datos jerárquicas que se ajustan a medida que la base de datos crece. Otra de las ventajas es que admite cualquier estructura de datos que funcione mejor con la app.[4]

¿Qué es real time?

Es un método de procesamiento de datos que requiere para su funcionamiento un flujo constante de entrada y salida de datos para mantener información en tiempo real. Cuando recibe datos sin procesar, los procesa inmediatamente para facilitar así la toma de decisiones, ya que sucede casi al instante,este en lugar de almacenar, facilita la disponibilidad para promover la obtención de información lo antes posible, a fin de mejorar la rentabilidad, la eficacia y los resultados en las empresas y proyectos.[5]

¿Qué es una historia de usuario?

Una historia de usuario es una explicación breve, informal y en lenguaje sencillo escrita desde la perspectiva del usuario final para determinar qué desea hacer dentro de un producto de software con el fin de obtener algo que le resulte valioso. [6]

¿Cómo se representa una historia de usuario?

La historia de usuario suele seguir el patrón de rol - función - beneficio, y se representa de la siguiente manera: Cómo (tipo de usuario), quiero (acción), para que (beneficio o valor). [7]

¿Cómo se determina si la historia es o no adecuada?

Para determinar su precisión, esta debe ser: Independiente, a fin de poder modificarla libremente a medida que cambian las prioridades; negociable, y así dividirla en historias más pequeñas de ser necesario; valiosa para el cliente; estimable, pues se debe evaluar el esfuerzo que requiere completar la tarea; pequeña, garantizando que se cumpla su funcionalidad y comprobable, verificando que el software cumpla con el rendimiento acordado mediante criterios de validación. [8] [9]

¿Cómo se define si la historia está o no terminada?

- Es funcional
- Cuenta con un sistema de gestión del código o de versiones (código actualizado en git)
- Se han realizado las pruebas pertinentes (unitarias, funcionales, de rendimiento)
- Tiene una documentación completa (a través de manuales o tutoriales) [8]

¿Cómo planear una historia de usuario?

Para planear las historias de usuario como equipo, se utiliza la técnica de las tres C.

- Cards: Son las tarjetas en donde se almacena la información de cada historia.
- Conversation: Diálogo con el equipo para asegurar que no haya dudas en el proceso de desarrollo.
- Confirmation: Cada uno de los miembros llegan a un acuerdo y confirman que el contenido es claro [8]

¿Cuáles son sus beneficios?

Entendiendo cuáles son los requerimientos del usuario, se garantiza que este sea el centro de la conversación para así crear un producto acorde a sus necesidades que mejore su experiencia.

Se trabaja en el menor tiempo posible pues es una herramienta que permite priorizar el trabajo con eficacia. Además, se impulsan soluciones innovadoras, creativas y originales en equipo. [6]

Historias de usuario vs requerimientos

Las historias de usuario describen el beneficio que un usuario obtiene de una acción. Por otro lado, los requerimientos se refieren a las características que debe satisfacer el software para cumplir con la necesidad que el usuario manifiesta mediante ciertos criterios de aceptación establecidos. Es decir, una historia de usuario está conformada por requerimientos. [9]

¿Qué es un sistema de control de versiones?

Un sistema de control de versiones es una herramienta de software que rastrea el historial de cambios hechos por las personas y equipos que colaboran juntos en proyectos. A medida que los desarrolladores hacen

cambios al código, cualquier versión anterior de este puede recuperarse en cualquier momento.

Los desarrolladores pueden revisar el historial de proyectos para averiguar qué cambios se hicieron, quién los realizó, cuándo fueron hechos y por qué eran necesarios.

Los sistemas de control de versiones proporcionan a cada desarrollador una vista unificada de un proyecto, lo cual muestra el trabajo que ya está en progreso. El ver un historial de cambios transparente, quién los hizo y cómo contribuyen al desarrollo de un proyecto, ayuda a que los miembros de los equipos se coordinen mientras trabajan independientemente. [11] [12]

¿Qué es un sistema de control de versiones distribuido?

Son sistemas de control de versiones que permiten subir el código, crear ramas y fusionarlas sin necesidad de conectarse al servidor principal, pues cada desarrollador tiene una copia del proyecto y del historial del mismo en la que trabaja desde un repositorio clonado almacenado en la nube. Es necesaria una conexión constante a un repositorio central. [12]

¿Qué es Git?

Git es el sistema de control de versiones distribuido más popular. Git se utiliza habitualmente tanto para el desarrollo de proyectos de código abierto como comerciales y tiene beneficios significativos para los individuos, equipos y negocios.

Git se basa en los siguientes principios:

- Acceso al historial: Los desarrolladores pueden acceder al historial de cambios, retroceder en el tiempo y restaurar versiones anteriores si es necesario.
- Registro continuo: Git funciona cada cambio realizado en el código, creando una línea del tiempo detallada.
- Seguridad y protección; Cada línea de código está protegida en su propia línea del tiempo, asegurando que nada se pierda.
- Ramificación flexible: Permite que múltiples usuarios trabajen en diferentes aspectos del proyecto simultáneamente, usando varias ramas.
- Fusión eficiente: Git fusiona los cambios realizados en sus ramas de manera inteligente, actualizando solo lo necesario y preservando el código. “Esto quiere decir que, si por ejemplo en un menú sólo se ha pedido cambiar un enlace, en el archivo original **sólo se modificará este enlace**. Por otro lado, si se han creado archivos nuevos, se añadirán al proyecto original y de igual manera, si se han eliminado archivos, se eliminarán en el proyecto original.”
- Asincronicidad, pues los desarrolladores trabajan en todos los husos horarios mientras se mantiene la integridad del código fuente. Al utilizar las ramas, los desarrolladores pueden proponer cambios al código de producción asegurando que no haya ninguna pérdida. [13]

¿Cómo funciona Git?

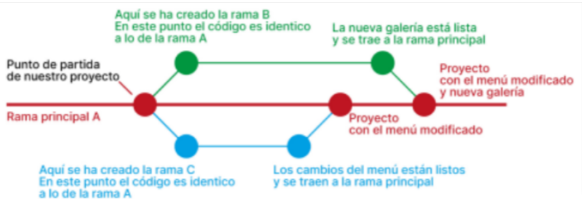

Figura 1, ¿Cómo funciona Git?. Saul Pane

¿Qué es Gitlab?

GitLab es un servicio de alojamiento de repositorios Git, con el que se pueden conectar y gestionar aplicaciones de manera colaborativa, que ofrece una alternativa a plataformas como Github. [14]

¿Para quién está diseñado Gitlab?

Desarrolladores que requieran de una plataforma que soporte el código generado, equipos de seguridad que necesiten de la protección de aplicaciones, equipos de operaciones que busquen gestionar aplicaciones, gerentes de proyectos que aseguren su rendimiento y empresas y startups que demanden eficacia y eficiencia. [15]

¿Qué se puede hacer con Gitlab?

Esta herramienta permite tener un control total de grupos y proyectos, dando cabida a gestionar el estado de la actividad realizada. De este modo, se puede acceder al total de modificaciones realizadas por los equipos implicados, junto a otra información de interés del proceso, enriqueciéndolo y dando un paso más respecto del control de versiones. [15]

¿Cuáles son sus características?

1. Gestión de Código Fuente: Permite la modificación de repositorios mediante ramificación y fusión.
2. Pipelines de CI/CD: Automatizan el proceso de prueba y despliegue, asegurando que los cambios de código se integren y desplieguen rápida y confiablemente.

3. Seguridad Integrada: Permite a los equipos realizar pruebas de seguridad de aplicaciones estáticas y dinámicas.
4. Gestión de Proyectos: Con características como seguimiento de incidencias, seguimiento de tiempo y herramientas de planificación de proyectos para mantener a los equipos organizados y en camino.
5. Monitoreo y Alertas: Para mantener un control cercano sobre el rendimiento de sus aplicaciones y configurar alertas para ser notificados de cualquier problema.
6. Colaboración y Revisión de Código: Los equipos pueden hacer revisiones de código en conjunto, dejar comentarios y discutir los cambios. Esto garantiza alta calidad de código y desarrollo colaborativo. [15]

¿Qué es stackoverflow?

Stack Overflow es un sitio web de preguntas y respuestas para programadores e ingenieros en el que los desarrolladores pueden plantear sus dudas y obtener ayudas de otros miembros de la comunidad. A medida en la que los usuarios colaboran con las preguntas de otros participantes priorizando la calidad del contenido, acumulan puntos. [16]

¿Cuáles son los beneficios de Stack Overflow?

“Cuando un desarrollador se enfrenta a un problema, puede buscar en la plataforma si alguien ya ha planteado una pregunta similar. Si encuentra una pregunta relevante, puede leer las respuestas proporcionadas y ver si alguna de ellas resuelve su problema. Si no encuentra una respuesta satisfactoria, tiene la opción de hacer su propia pregunta y esperar respuestas de la comunidad.” [16]

Además, el acceso a un vasto conocimiento, la resolución rápida de problemas, la disposición en cuanto a la colaboración de la comunidad y el desarrollo profesional al obtener nuevas perspectivas son solo algunas de las muchas ventajas que tiene el sitio. [16]

¿Qué es la librería Pygame?

Esta es una librería diseñada para el desarrollo y diseño de videojuegos en 2D, basada en SDL este es un conjunto de bibliotecas en lenguaje C para la gestión de gráficos en 2D [17]. Permite desde la manipulación y creación de ventanas, utilización de imágenes e implementación de audio y periféricos (por ejemplo, mouse y teclado)

¿Cómo utilizar?

Se puede realizar una enumeración de las grandes funcionalidades que posee esta librería, la que cabe aclarar que no viene preinstalada en Python, por tal razón previamente se debe realizar la correspondiente instalación.

- Inicializando la terminal de la consola (en Visual Studio Code)
- Procedemos a revisar si ya está instalada:
    - Escribimos pip list, presionamos enter
    - Visual nos mostrará las librerías externas que posee Python
    - En dado caso que la librería Pygames no esté en el computador, procedemos con la instalación.
    - En la terminal, escribimos pip install pygames, presionamos enter y dejamos que el proceso de instalación terminé
    - Ya se puede implementar la librería.
- Para utilizarla en el código, utilizamos la palabra reservada “Import” a continuación llamamos a la librería “pygames”. De esta manera ya puede implementarse esta librería en el desarrollo de videojuegos en 2D

¿Qué funcionalidades tiene?

- En el apartado gráfico:
    - Creación propia de formas.
    - Implementación de imágenes, siendo capaz de soportar diversos formatos
- Uso de animaciones
    - Gracias al ciclo principal permite que se presente una actualización en pantalla, lo que daría esa sensación de animación fluida.
- Uso de audio
    - Implementación de sonido, siendo capaz de soportar algunos formatos.
    - También existen funcionalidades propias que permiten el uso de efectos sonoros.
- Reconocimiento de periféricos:
    - Logra reconocer periféricos, mouse y teclado, y sus determinadas señales e ingreso de información.

Historias de usuario con sus respectivos requerimientos

Como jugador, quiero ingresar a una interfaz llamativa e intuitiva, para que estimule la rejugabilidad.	Implementar una paleta de colores y tipografía visualmente agradable.
	Ajustar el tamaño de la ventana a uno cómodo.
	Crear controles sencillos de comprender.
Como jugador, quiero que los modelos de los barcos y el área de juego sea llamativa, para que sea entretenido y atractivo.	Crear diseños llamativos, lo que corresponde a un estilo visual, para atraer jugadores.
	Implementar un atractivo, para no cansar la visión del usuario, con su respectiva cuadrícula y demás elementos funcionales
	implementar menús sencillos e intuitivos
Como jugador, quiero seleccionar el modo de juego para aprovechar el juego en diferentes situaciones.	Implementar una interfaz que muestre ambos modos de juego.
	Desarrollar un sistema funcional, en el que se implemente un modo de un jugadores.
	Desarrollar un sistema funcional, en el que se implemente un modo de dos jugadores.
Como jugador, quiero determinar los parámetros de dificultad de tiempo, para tener mejor control en lo que respecta a su gestión.	Generar una alternativa de tiempo que represente el nivel de dificultad fácil, intermedio y complejo.
	El usuario escoge el nivel de dificultad que más le llame la atención.
	Implementar un menú de selección.
	Generar una cuadrícula que represente el nivel de dificultad fácil, intermedio y complejo.
Como jugador quiero determinar los parámetros de dificultad en lo respectivo al tamaño del mapa, para que se ajuste a mis gusto y nivel de juego.	El usuario escoge el nivel de dificultad que más le llame la atención.
	Implementar un menú de selección.
Como jugador, quiero conocer las reglas bajo las que se regirá el juego, para tener claridad de qué se debe hacer.	Establecer las reglas del juego.
	Determinar la puntuación que tendrá cada tiro.
	Desarrollar en una ventana la lista de normas a tener en cuenta.
Como jugador, quiero seleccionar la posición y sentido de mis barcos, para plantear mi estrategia.	Como jugador, quiero seleccionar la posición y sentido de mis barcos, para plantear mi estrategia.
	Realizar una validación de posición, para evitar la superposición de las naves
	Interfaz intuitiva en lo respectivo a la facilidad para realizar la colocación de los barcos.
Como jugador, quiero seleccionar la coordenada donde atacaré a mi oponente para llevar un control de las decisiones que he tomado.	Ofrecer al jugador un medio para ingresar el valor de las coordenadas.
	Agregar una confirmación de ataque, para evitar movimientos innecesarios.
	Mostrar gráficamente en dónde ha caído el elemento lanzado.
	Generar un sonido e imagen que identifique si se dió o no el impacto.

Como jugador, quiero conocer si he acertado o no en alguna nave de mi oponente para determinar mi siguiente movimiento.	Si se dió, en la plantilla de mi oponente identifico el tipo de barco que impacté.
	Si no, generar un rastro que indique que allí no hay ninguna parte de la nave.
Como jugador, quiero conocer si mi oponente ha acertado o no en alguna de mis naves, para conocer cuántas me quedan disponibles.	Realizar una validación de posición, para conocer mis naves restantes.
	Interfaz que da conocimiento en tiempo real de mi situación en la partida.
	Por medio de un menú permite que el usuario active o desactive la interfaz del recuento de barcos.
Como jugador, quiero conocer la cantidad de naves vigentes de mi oponente, para saber cuántas me quedan por derribar.	Realizar una validación de posición, para conocer las naves restantes.
	Interfaz que da conocimiento en tiempo real de la situación de mi oponente en la partida.
	Por medio de un menú permitir que el usuario active o desactive la interfaz del recuento de barcos.
Como jugador, quiero llevar un registro de mis victorias y derrotas, para conocer como fue mi desempeño durante las partidas.	Implementar un contador de victorias, que muestre en tiempo real el conteo
	Implementar un contador de derrotas, que muestre en tiempo real el conteo
	En las esquinas de la interfaz se mostrarán dichos conteos

Tabla 1, Descripción Historias y Requerimientos.

III. Extras.

En primera instancia estas funcionalidades agregadas al producto final son pensadas con la idea de hacer que el juego de batalla naval escogido, sea más completo y a su vez atractivo para el usuario.

Por tal razón, en el proceso investigativo, evidenciamos que la librería más conocida en lo respectivo al diseño de programas por medio de ventanas (Tkinter), se quedaba algo corta al momento de implementar varias de las ideas que se quieren realizar, además sería más complejo el hecho de ir fusionando e importando elemento que nos permitan desarrollos interesantes, pero de manera independiente. Por ende, se desea trabajar con la librería (Pygame), la cual engloba muchas funcionalidades en un solo lugar.

Detalles agregados

- *Realizar una validación de disparo*, para evitar que el usuario dispare a una misma coordenada varias veces. Esto creando una función que posea dentro un if y un else, dentro de la función se valida el valor ingresado por el usuario y se implementa el condicional.

```
#bosquejo inicial funciones agregadas

import pygames

#-----------------------------Validacion del disparo----------------------

disparoGuardado=[] #inicializamos la lista, para guardar las coordenadas que se vayan marcando

def validacion(x,y): # x, y serian las coordenadas del disparo
    if((x,y) in disparoGuardado):
        print("El disparo en la coordenada ",x,y," ya fue realizado")
        return False #disparo incorrecto
    else:
        disparoGuardado.append((x,y)) #se guarda la coordenada en la lista, en la ultima posicion
        print("Disparo correcto",x,y,"disparo guerdado")
        return True #disparo correcto
```

Figura 2, Muestra de la función "Validación del disparo"

- *Animaciones*, gracias a las funcionalidades de la librería podemos cargar imágenes, se implementan dos imágenes, salpicadura (para cuando el disparo da al agua) y explosión (para cuando el disparo logra golpear a un barco). Esto creando una función que según la coordenada ingresada se valide si dio o no en el blanco y proceda a mostrar la imagen que según corresponda. Con la idea de hacer que la interfaz sea llamativa.

```
ejo inicial funciones agregadas

 pygames

-------------------------implementacion de imagenes----------------------

ygames.image.load('nombredelaImagen.png')
adura=pygames.image.load('nombredelaImagen.png') #Gracias a pygames.image,load() se cargan las imagenes

imacio(x,y, screen):
reen.blit(boom,(x,y)) #aqui especificamos que quereos ver la imagen en la coordenada
games.display.update() #aqui actualizamos el display para lograr ver la imagen cargada
```

Figura 3, Muestra de la función "Implementacion de imágenes".

- *Sonidos*, gracias a las funcionalidades de la librería podemos cargar sonidos, se implementa en el momento en que se realiza el disparo, si este da en el agua o si logra golpear un barco. Para darle un toque más inmersivo y a su vez divertido, también se utilizaría una función que se ejecute cada vez que el usuario realiza el disparo, luego de este sonido inicial, por medio de un if /else, se podría validar si dio en el objetivo o si fue al agua y posteriormente reproducir el sonido.

```
#bosquejo inicial funciones agregadas

import pygames

#----------------------------implementacion de Sonidos----------------------

ingresoDisparo=(0,0)
disparo=pygames.mixer.sound('NombredelArchivo.mp3')
impacto=pygames.mixer.sound('NombredelArchivo.mp3')
salpicadura1=pygames.mixer.sound('NombredelArchivo.mp3') #Gracias al pygames.mixer.sound se carga el audio

def sound(x,y):
    disparo.play() #.play se encarga de reproducir el sonido asignado a esa variable
    if(ingresoDisparo==x,y): #disparo correcto, dio al barco
        impacto.play()
    else:
        salpicadura1.play #disparo erroneo, dio al agua
```

Figura 4, Muestra de la función "Implementación de sonido".

Historia de usuario de los extras - 3 requerimientos.

Como jugador, quiero ser notificado por medios visuales, para saber cuando logró alcanzar una nave o me equivoco en el disparo.	Agregar imágenes de referencia para las dos situaciones, explosión y salpicadura
	Implementar una función que realice este trabajo, para optimizar la extensión del código
	Realizar la animación al momento de hacer el disparo en la coordenada especificada.
Como jugador, quiero ser notificado por medios sonoros, para confirmar que la orden del disparo	Cargar en el programa el respectivo sonido de disparo, que se ejecuta cada vez que confirma la coordenada a la que quiero bombardear
	Implementar una función que realice este trabajo, para optimizar código.
	Permitir al usuario activar y desactivar el sonido desde un menú
Como jugador, quiero ser notificado por medios sonoros si di en el blanco o tuve un disparo erróneo.	Cargar en el programa el respectivo sonido de disparo, que se ejecuta cada vez que confirma la coordenada a la que quiero bombardear
	Implementar una función que realice este trabajo, para optimizar código.
	Permitir al usuario activar y desactivar el sonido desde un menú
Como jugador, quiero que se muestre una animación al momento en que el disparo impactó tanto al agua como a las embarcaciones.	Implementar la animación de disparo al agua.
	Implementar la animación de impacto.
	Duración y fluidez de la animación.

Tabla 2, Descripción Historias y Requerimientos.

IV. Referencias

[1] Big Bad Wolf. *"Estrategias."*. https://brainking.com/es/GameRules?tp=46 (Accedido 30-Dic-2024)

[2] Coescrito por personal de wikiHow. *"Cómo jugar a la batalla naval."*. https://es.wikihow.com/jugar-a-la-batalla-naval (Accedido 30-Nov-2024)

[3] Digital 55. *"¿Qué es firebase,funcionalidades,ventajas y conclusiones?"*. https://digital55.com/blog/que-es-firebase-funcionalidades-ventajas-conclusiones/ (Accedido 30-Nov-2024)

[4] Google. *"Cloud Firestore"* .https://firebase.google.com/docs/firestore?hl=es-419 (Accedido 30-Nov-2024)

[5]Hewlett Packard Enterprise. *"¿Qué es el procesamiento en tiempo real?"*. https://www.hpe.com/lamerica/es/what-is/real-time-processing.html#:~:text=El%20procesamiento%20en%20tiempo%20real%20es%20un%20m%C3%A9todo%20de%20procesamiento,mantener%20informaci%C3%B3n%20en%20tiempo%20real. https://www.elternativa.com/real-time/ . (Accedido 30-Nov-2024)

[6] Team Asana. *"Historias de usuario: 3 ejemplos para generar valor para el usuario."*. https://asana.com/es/resources/user-stories . (Accedido 1-Dic-2024)

[7] Nimble Humanize Work. *"Historias de usuarios: Qué son, por qué y cómo utilizarlas."*. https://www.nimblework.com/es/agile/historias-de-usuarios/ (Accedido 1-Dic-2024)

[8] G. Romero. Platzi.*"¿Cómo crear historias de_usuario?"*. https://platzi.com/clases/1750-scrum/24285-que-nos-cuentan-las-historias-de-usuario/. (Accedido 1-Dic-2024)

[9] P. Sabadí. Srumizate. *"Historias de usuario para definir los requisitos de un proyecto."*. http://scrumizate.com/post/14/historias-de-usuario-para-definir-los-requisitos-de-un-proyecto (Accedido 1-Dic-2024)

[10] Miro. *"Cómo escribir buenas historias de usuario en el método Agile"*. https://miro.com/es/agile/que-es-historia-usuario/ (Accedido 1-Dic-2024)

[11] Documentación de Github. *"Acerca del control de versiones y de Git"*. https://docs.github.com/es/get-started/using-git/about-git (Accedido 1-Dic-2024)

[12]Unity. *"¿Qué es un sistema de control de versiones?"*. https://unity.com/es/topics/what-is-version-control. (Accedido 1-Dic-2024)

[13] S. Pane. Nuclio. *¿Qué es Git y cómo funciona?* https://nuclio.school/blog/que-es-git-y-como-funciona/ . (Accedido 1-Dic-2024)

[14] Lis Data Solutions. *¿Qué es Gitlab?* https://www.lisdatasolutions.com/es/blog/que-es-gitlab/ (Accedido 1-Dic-2024)

[15] GetGuru. "Cómo usar Gitlab: una guía completa" https://www.getguru.com/es/reference/how-to-use-gitlab-a-comprehensive-guide (Accedido 1-Dic-2024)

[16] S. Fabado. "¿Qué es Stack Overflow? https://ebootcamp.net/blog/que-es-stack-overflow/#:~:text=Stack%20Overflow%20es%20un%20sitio,la%20calidad%20de%20sus%20respuestas. (Accedido 1-Dic-2024)

[17] R. Maldonado. *"¿Qué es Pygame?"*. KeepCoding Tech School. [en linea] https://keepcoding.io/blog/que-es-pygame/ (Accedido 1-Dic-2024)

[18] J. Fincher. *"PyGame: Introducción a la programación de juegos en Python"*. Real Python. [en línea]. https://realpython.com/pygame-a-primer/#background-and-setup (Accedido 1-Dic-2024)