

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, Desarrollo.

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 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 Realtime Firebase ?

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 los beneficios de una historia de usuario?

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 la librería ?:“ Las librerías de Python son conjuntos de funciones que permiten realizar distintas funciones, ahorrando tiempo y esfuerzo al programador. Gracias a la amplia variedad de librerías existentes, Python es uno de los lenguajes más flexibles y potentes que pueden utilizarse.”[13]

¿Qué es la librería Random?:“Este módulo implementa generadores de números pseudoaleatorios para varias distribuciones”. [14]

¿Qué es la librería Time?:“Este módulo proporciona varias funciones relacionadas con el tiempo.”[15]

¿Qué es la librería Tkinter?

Tkinter es el paquete estándar de Python para el desarrollo de interfaces gráficas de usuario. Por ese motivo, se incluye instalar el propio entorno, sin que sea necesario importar nada adicional para empezar a utilizarlo. Su nombre procede de la interfaz Tk en la que está basado. En realidad, se trata de un interfaz de programación de aplicaciones creada específicamente para utilizar Tk desde Python.

Tk es un entorno de trabajo multiplataforma de software libre, que proporciona una biblioteca de controles gráficos para el desarrollo de interfaces de usuario Tcl (es un lenguaje de programación interpretado y de propósito general que fue diseñado para facilitar la creación de aplicaciones de software). Este sencillo lenguaje de propósito general es interpretado y soporta múltiples paradigmas de programación, incluyendo la orientada a objetos y la imperativa o tradicional. Por lo tanto, Tkinter es un recubrimiento de Tcl que permite usar Tk desde Python, motivo por el que el intérprete de Tcl está incrustado en el de Python. [16]

¿Qué es SQLAlchemy?:

SQLAlchemy es un ORM potente y eficaz que permite trabajar con muchas bases de datos realizando una abstracción de sus diferencias y utilizando aspectos del modelo de objetos de Python. [17]

¿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 [18]. Permite desde la manipulación y creación de ventanas, utilización de imágenes e implementación de audio y periféricos (por ejemplo, mouse y teclado)

¿Cómo utilizarla?

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 pygame, presionamos enter y dejamos que el proceso de instalación termine
 - Ya se puede implementar la librería.
- Para utilizarla en el código, utilizamos la palabra reservada “Import” a continuación llamamos a la librería “pygame”. 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

HDU	Historias de usuario
1.	Como jugador, quiero que el juego se ejecute en una ventana, para obtener una mayor flexibilidad al adaptarse a varios tipos de configuraciones.
2.	Como jugador, quiero que un área determinada de la ventana se destiné para el tablero de juego, para que se pueda tener mejor distribución de espacios y funcionalidades.
3	Como jugador, quiero determinar los parámetros de dificultad de las preguntas y la habilidad de mi oponente.
4	Como jugador, quiero conocer las reglas bajo las que se regirá el juego, para tener claridad de qué se debe hacer y cómo resolver las preguntas.
5	Como jugador, quiero seleccionar la posición y sentido de mis barcos, para plantear mi estrategia.
6	Como jugador, quiero seleccionar la posición de lanzamiento, para atacar a mi oponente.
7	Como jugador, quiero conocer si he acertado o no en alguna nave de mi oponente para determinar mi siguiente movimiento.
8	Como jugador, quiero conocer cuáles han sido mis victorias y derrotas, para saber cómo ha sido mi desempeño durante las partidas.
9	Como jugador, quiero que se muestre una animación al momento en que el disparo impacto, tanto al agua como a las embarcaciones, para proporcionar una retroalimentación visual sobre las decisiones tomadas.
10	Como jugador, quiero ser notificado por medios sonoros si dí en el blanco o tuve un disparo erróneo, para hacer que el juego sea más dinámico

Tabla 1, Historias de Usuario.

RHDU	Requerimientos
1.1	Desarrollar un programa que se ejecute en una ventana ajustable.
1.2	Delimitar el tamaño que deben tener los elementos que se muestran en la ventana.
1.3	Determinar la tipografía y la paleta de colores que deberá tener la ventana.
2.1	Delimitar una proporción de los elementos respecto al tamaño de la ventana, para que se ajusten activamente.
2.2	Configurar la posición de cada elemento, para que en su funcionamiento no se sobreponga a otras.
2.3	Determinar un máximo y un mínimo del tamaño del tablero(grilla), para conservar una visualización adecuada.
3.1	Desarrollar los niveles de dificultad, siendo estos fácil, medio y difícil ,realizando tres funciones para determinarlo.
3.2	El programa debe interpretar cuál fue la selección de dificultad, mediante una concatenación de condicionales.
3.3	Configurar tres botones mediante métodos de la librería a implementar ubicados en la ventana principal, uno para cada nivel.
4.1	Desarrollar un programa que se ejecute en una ventana ajustable y en la que aparezcan las reglas implementadas creadas mediante strings.
4.2	El programa debe interpretar si tengo claras las reglas mediante mi selección de continuar o no, esto gracias a una concatenación de condicionales.
4.3	Configurar dos botones mediante métodos de la librería a implementar ubicados en la ventana emergente, uno para cada opción, si o no.
5.1	El programa debe interpretar cuál fue la coordenada seleccionada en mi tablero , mediante una concatenación de condicionales.

5.2	<i>El programa debe interpretar cuál fue la posición seleccionada, mediante una concatenación de condicionales</i>
5.3	<i>El programa debe interpretar si esa es mi elección final, mediante una concatenación de condicionales.</i>
6.1	<i>El programa debe interpretar cuál fue la coordenada seleccionada en el tablero del oponente, mediante una concatenación de condicionales.</i>
6.2	<i>Generar funciones que aseguren que la celda no haya sido previamente seleccionada.</i>
6.3	<i>El programa debe interpretar si esa es mi elección final, mediante una concatenación de condicionales.</i>
7.1	<i>Mediante el uso de una librería, crear una animación de objeto que identifique el caso de acierto y el caso de error.</i>
7.2	<i>Verificar que la posición a la que ha disparado el jugador coincida con la posición en la que está alojada la nave.</i>
7.3	<i>Hacer un recuento de cuántas partes de la nave quedan a flote y mostrar visualmente esos datos actualizándolos en tiempo real.</i>
8.1	<i>Definir qué puntaje debe haber alcanzado el usuario para obtener una victoria o una derrota</i>
8.2	<i>Almacenar en una base de datos el resultado que obtuvo el jugador luego de cada partida.</i>
8.3	<i>Mediante el uso de una librería, crear un menú que permita al jugador visualizar el resumen de su juego.</i>
9.1	<i>Cargar ilustraciones mediante métodos funcionales de la librería que se esté implementando.</i>
9.2	<i>Configurar la aparición de una de las dos ilustraciones una determinada unidad de tiempo en la coordenada seleccionada, según corresponda.</i>

9.3	<i>Implementar una función que se encargue de trabajar todo este apartado, buscando la optimización y longitud del programa.</i>
10.1	<i>Determinar los tipos de sonidos que se desean implementar para posteriormente cargarlos al proyecto</i>
10.2	<i>Implementar una opción para desactivar las opciones de sonido, tanto de ambientación como de acciones jugables</i>
10.3	<i>Configurar el volumen predeterminado para que los sonidos sean tenues y no lleguen a molestar al usuario.</i>

Tabla 2, Requerimientos de las Historias de Usuario

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

Cuando el jugador desea atacar una coordenada, aparece una ventana con una pregunta del examen de admisión de la Universidad Nacional de Colombia, específicamente sobre las áreas de matemáticas, lógica y análisis de imagen.

- Tendrá máximo minuto y medio para responder satisfactoriamente.
- De lo contrario perderá una vida y la coordenada seleccionada será atacada en su propio tablero.

HDU	Historias de Usuario
1	<i>El jugador, quiere ver en una ventana la pregunta específica al seleccionar la coordenada, para implementar el juego como un método de preparación y estudio.</i>

2	<i>El jugador, quiere ver dos listas de soluciones al finalizar la partida, para tener una retroalimentación de sus errores y aciertos</i>
3	<i>El jugador, quiere ver una cuenta regresiva en alguna sección de la ventana para conocer el tiempo disponible para dar solución a la pregunta dada.</i>

Tabla 3, Historia de Usuario Extras

<i>RH DU</i>	<i>Requerimientos</i>
1.1	<i>Al momento de querer atacar una coordenada, el programa generará una ventana con una pregunta del examen.</i>
1.2	<i>Por medio de una función, implementar condicionales, que hagan ciertas acciones sí es correcto o incorrecto.</i>
1.3	<i>Implementar métodos para cargar las imágenes del examen que se mostrarán en la ventana.</i>
2.1	<i>Implementar listas para poder plasmar los resultados correctos e incorrectos</i>
2.2	<i>Realizar una carga preliminar de las soluciones al programa.</i>
2.3	<i>Implementar contadores de aciertos y desaciertos, para mostrar dos listados diferentes de las respuestas.</i>
3.1	<i>Implementar una librería que contribuya en todo lo relacionado a mediciones de tiempo.</i>
3.2	<i>Ubicar en la ventana un espacio donde se muestre el reloj</i>
3.3	<i>Implementar una ventana que me alerta cuando el tiempo se ha terminado.</i>

Tabla 4, Requerimientos Historias de Usuario

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.eltiempo.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-no-s-cuentan-las-historias-de-usuario/>. (Accedido 1-Dic-2024)
- [9] P. Sabadí. Scrumizate. “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] “El lenguaje Python y sus principales librerías”
<https://formadoresit.es/el-lenguaje-python-y-sus-principales-librerias/#:~:text=Las%20librer%C3%ADas%20de%20Python%20son,y%20potentes%20que%20pueden%20utilizarse.>
(Accedido 3-Dic-2024)
- [14] “Random — Generar números pseudoaleatorios”
<https://docs.python.org/es/3.10/library/random.htm>
(Accedido 3-Dic-2024)
- [15] “time — Time access and conversions”
<https://docs.python.org/es/3/library/time.html>
(Accedido 3-Dic-2024)
- [16] MINGUEZ, T. D. (1900). *Desarrollo de interfaces gráficas en python 3 con tkinter*. MARCOMBO BOIXAREU EDITORES.
- [17] M. R. J. Gutierrez, *Python 3: Los fundamentos del lenguaje*, 2da ed. 2018.
- [18] R. Maldonado. “¿Qué es Pygame?”.
KeepCoding Tech School. [en línea]
<https://keepcoding.io/blog/que-es-pygame/>
(Accedido 1-Dic-2024)
- [19] 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)