Documento srs

Juego Run Island

Andrés de Avila

Steven Lopez

Introducción

Propósito

Audiencia

Uso previsto

Alcance

Acrónimos y definiciones

Descripción general

Necesidades del usuario

Dependencias y Asunciones

Requisitos y características del sistema

Requisitos funcionales

Requisitos de interfaz externa

Características del sistema

Requisitos no funcionales

INTRODUCCION

Propósito

El propósito principal de este producto de software es desarrollar y proporcionar un juego interactivo basado en la tecnología React que tenga como objetivo fomentar el aprendizaje de JavaScript entre los estudiantes de análisis y desarrollo de software. Este juego se concibe como una herramienta educativa efectiva y atractiva que aborda los siguientes objetivos:

Aprendizaje Divertido y Motivador: El producto tiene la intención de hacer que el proceso de aprendizaje de JavaScript sea más atractivo y motivador para los estudiantes. Utilizando un enfoque lúdico, se pretende captar la atención de los estudiantes y mantener su interés en el estudio de este lenguaje de programación fundamental.

Reforzar Conceptos y Prácticas: A través de escenarios de juego interactivos, desafíos y ejercicios, el juego proporcionará oportunidades para que los estudiantes refuercen y apliquen los conceptos y las prácticas de JavaScript que han aprendido en sus cursos académicos o tutoriales previos. Se espera que esta aplicación práctica mejore su comprensión y retención de conocimientos.

Experiencia de Aprendizaje Autodirigido: El juego está diseñado para ser una herramienta de aprendizaje autodirigido, lo que permite a los estudiantes avanzar a su propio ritmo y explorar diferentes aspectos de JavaScript según sus necesidades y niveles de habilidad. Esto fomenta la autonomía y la independencia en el proceso de aprendizaje.

Seguimiento del Progreso: El sistema incluirá una funcionalidad de seguimiento del progreso que permitirá a los estudiantes monitorear su desempeño y su mejora a lo largo del tiempo. Esto proporciona retroalimentación valiosa y motivación adicional para los usuarios.

Fomentar la Colaboración: Además de los aspectos individuales del juego, se incorporarán elementos que fomenten la colaboración y el aprendizaje social. Esto podría incluir la posibilidad de compartir logros.

Facilitar la Transición a Proyectos Reales: El juego tiene como objetivo preparar a los estudiantes para aplicar sus habilidades de JavaScript en proyectos de desarrollo de software del mundo real. Al proporcionar escenarios y problemas prácticos, se busca que los estudiantes se sientan más confiados y competentes al enfrentar desafíos en entornos profesionales

Alcance del Juego:

Módulo de Aprendizaje de JavaScript:

El juego incluirá 16 módulos dedicados al aprendizaje de JavaScript.

Se proporcionarán lecciones interactivas que cubrirán conceptos básicos y avanzados de JavaScript, incluyendo introducción a JavaScript, estructuras de control, funciones, objetos y propiedades, arreglos, manipulación del DOM, peticiones fetch y Ajax, programación orientada a objetos, programación asíncrona, manejo de errores, almacenamiento en el navegador, herramientas de desarrollo, frameworks y bibliotecas, gestión de módulos, seguridad , optimización y rendimiento

Cada lección estará compuesta de 10 preguntas para poder pasar al siguiente modulo , de igual forma cada pregunta tendrá su retroalimentación al ser respondida.

Desafíos y Ejercicios Prácticos:

El juego ofrecerá una variedad de desafíos y ejercicios prácticos que requerirán que los usuarios comprendan la salidas de código JavaScript real para resolver problemas.

Los desafíos se centrarán en escenarios del mundo real que los estudiantes pueden encontrar en proyectos de desarrollo de software.

Se proporcionarán retroalimentación inmediata para evaluar la corrección y eficacia del código escrito.

Sistema de Puntuación y Recompensas:

El juego contará con un sistema de puntuación que recompensará a los usuarios por completar lecciones y desafíos con éxito. Cada módulo completado brindara 50 puntos que podrán ser cambiados para desbloquear personajes dentro del juego para motivar a los estudiantes a alcanzar sus objetivos de aprendizaje.

Seguimiento del Progreso:

El juego registrará y mostrará el progreso de los estudiantes a lo largo del tiempo, incluyendo estadísticas de desempeño y el avance en las lecciones.

Colaboración y Competencia:

Se incorporarán características sociales que permitan a los estudiantes colaborar en proyectos específicos o competir entre sí de manera amigable.

Los usuarios podrán ver los logros y el progreso de otros estudiantes y compartir sus propios logros.

Vidas

Cada jugador contara con 5 vidas que se pierden al cometer un error en la respuesta de alguna lección, al llegar las vidas a cero, deberá esperar 12 horas para que se restauren y pueda continuar jugando

Compatibilidad con Plataformas:

El juego estará disponible como una aplicación web que funcionará en una variedad de dispositivos móviles, el jugador podrá acceder y jugarlo desde el navegador de su celular y se le deberá brindar la opción de instalarlo como una app nativa en su celular mediante Progresive web app

Localización:

Se considera que el juego este en dos idiomas español e ingles para tomar posibles jugadores

Actualizaciones y Mantenimiento:

Se planificarán actualizaciones regulares para mejorar el contenido, agregar nuevas lecciones y desafíos, y abordar problemas de seguridad o errores.

Acrónimos y definiciones

HTML: HyperText Markup Language.

Definición: El lenguaje de marcado utilizado para crear documentos web. Define la estructura y el contenido de una página web.

CSS: Cascading Style Sheets.

Definición: Un lenguaje utilizado para definir la presentación y el estilo visual de documentos HTML, incluyendo diseño, colores y tipografía.

JavaScript:

Definición: Un lenguaje de programación ampliamente utilizado en el desarrollo web para agregar interactividad y funcionalidad a las páginas web.

React:

Definición: Una biblioteca de JavaScript de código abierto utilizada para construir interfaces de usuario interactivas y componentes reutilizables.

DOM: Document Object Model.

Definición: Una representación en forma de árbol de la estructura de un documento HTML o XML, que permite la manipulación dinámica de elementos en una página web.

Sistema de Puntuación:

Definición: Un sistema dentro del juego que registra y asigna puntos a los usuarios en función de su desempeño y éxito en las actividades del juego.

Logros Virtuales:

Definición: Reconocimientos o recompensas virtuales que los usuarios obtienen al alcanzar ciertos hitos o logros en el juego.

Seguimiento del Progreso:

Definición: La capacidad del juego para rastrear y mostrar el progreso individual de los usuarios, incluyendo estadísticas y logros alcanzados.

Aprendizaje Autodirigido:

Definición: El enfoque educativo que permite a los estudiantes dirigir y controlar su propio proceso de aprendizaje, eligiendo qué, cuándo y cómo aprenden.

Accesibilidad:

Definición: La capacidad del juego para ser utilizado de manera efectiva por personas con discapacidades, incluyendo la compatibilidad con lectores de pantalla y la navegación fácil de usar.

Idiomas:

Definición: La adaptación del juego para que pueda ser utilizado en diferentes idiomas facilitando su acceso a una audiencia global.

Actualizaciones y Mantenimiento:

Definición: Proceso continuo de mejora y corrección de errores del juego después de su lanzamiento inicial, que incluye la adición de nuevo contenido y características.

Necesidades de los usuarios

Contenido Educativo de Calidad:

Los usuarios necesitan acceso a contenido educativo de alta calidad que aborde conceptos de JavaScript de manera clara y efectiva.

Interactividad y Participación:

Los usuarios esperan que el juego sea interactivo y que les permita participar activamente en actividades relacionadas con JavaScript, como escribir y ejecutar código.

Desafíos Graduales:

Los estudiantes desean desafíos que se adapten a su nivel de habilidad y se vuelvan más complejos a medida que avanzan en su aprendizaje.

Retroalimentación Inmediata:

Los usuarios valoran la retroalimentación inmediata que les indique si su código es correcto o si hay errores, así como sugerencias para mejorarlo.

Seguimiento de Progreso:

Los estudiantes necesitan la capacidad de rastrear su progreso a lo largo del tiempo, incluyendo estadísticas sobre las lecciones completadas y el rendimiento en desafíos.

Flexibilidad de Horarios:

Los usuarios pueden necesitar acceder al juego en horarios flexibles, lo que significa que debe estar disponible en línea y permitir el aprendizaje autodirigido.

Motivación y Recompensas:

Los usuarios se benefician de sistemas de puntuación y recompensas que los motiven a continuar aprendiendo y alcanzar metas específicas.

Localización:

Usuarios de diferentes regiones e idiomas esperan que el juego sea localizable, lo que facilita su uso en todo el mundo.

Actualizaciones y Soporte Continuo:

Los usuarios valoran las actualizaciones regulares que agregan nuevo contenido y características, así como el soporte para resolver problemas técnicos.

Entretenimiento y Enganche:

Los usuarios desean que el juego sea entretenido y atractivo, lo que les motivará a seguir jugando y aprendiendo.

Preparación para el Mundo Real:

Los estudiantes buscan una experiencia que les prepare para aplicar sus habilidades de JavaScript en proyectos reales de desarrollo de software.

Dependencias y Asunciones

Dependencias:

React y Bibliotecas Asociadas: El juego se desarrollará utilizando React y otras bibliotecas de JavaScript que sean necesarias para su funcionamiento, como React Router para la navegación.

Navegadores Web Modernos: Se asume que el juego será compatible con los navegadores web modernos, como Google Chrome, Mozilla Firefox, Safari y Microsoft Edge, y se deben realizar pruebas en estos navegadores para garantizar la compatibilidad.

Servidor Web o Plataforma de Alojamiento: Si el juego se alojará en línea, se requerirá un servidor web o una plataforma de alojamiento para publicar y servir el contenido.

Herramientas de Desarrollo: El desarrollo y la depuración del juego pueden depender de herramientas como editores de código, sistemas de control de versiones (por ejemplo, Git) y entornos de desarrollo integrados (IDEs).

Conexión a Internet : Los usuarios necesitarán acceso a Internet para jugar en línea y acceder a recursos relacionados con el juego.

Suposiciones:

Conocimiento Básico de JavaScript: Se asume que los estudiantes que utilizan el juego tienen conocimientos básicos de JavaScript y programación, lo que permite abordar conceptos más avanzados.

Accesibilidad a Dispositivos y Tecnología: Se asume que los usuarios tienen acceso a dispositivos (computadoras, tabletas, dispositivos móviles) y tecnología necesaria para ejecutar el juego, como navegadores web actualizados.

Interacción y Participación Activa: Se supone que los usuarios están interesados en aprender JavaScript y están dispuestos a participar activamente en las actividades del juego.

Seguridad de Datos: Se asume que se tomarán medidas para garantizar la seguridad de los datos de los usuarios, especialmente si se almacenan datos personales o de progreso.

Desarrollo Continuo: Se supone que habrá ciclos de desarrollo continuo para mejorar el juego, corregir errores y agregar contenido adicional.

Cumplimiento Normativo: Se supone que el juego cumplirá con las regulaciones y leyes locales relacionadas con la privacidad de datos y la accesibilidad, según sea necesario.

Requisitos de Hardware: Se supone que los usuarios tienen hardware y recursos suficientes en sus dispositivos para ejecutar el juego de manera óptima.

Participación Voluntaria: Se asume que la participación de los estudiantes en el juego es voluntaria y no está vinculada a un currículo académico obligatorio.

Mantenimiento de Servidores : el juego se aloja en línea, se supone que habrá un plan de mantenimiento y disponibilidad del servidor.

Requisitos Funcionales

1. Registro de usuarios con nombre de usuario y contraseña: Los usuarios deben poder crear cuentas utilizando un nombre de usuario y una contraseña.
2. Inicio de sesión de usuarios registrados: Los usuarios registrados deben poder iniciar sesión en sus cuentas existentes.
3. Perfiles de usuario personalizables: Los usuarios deben tener la capacidad de personalizar sus perfiles con información como nombre, foto de perfil, etc.
4. Sistema de puntaje y recompensas: Los usuarios ganan 50 monedas al completar lecciones y desafíos, lo que les motivara a seguir aprendiendo
5. Sistema de seguimiento de progreso de aprendizaje: Se registrara y muestra el progreso de cada usuario en las lecciones de JavaScript, mediante el mapa del tesoro
6. Lecciones iniciales de JavaScript para principiantes: Lecciones introductorias diseñadas para usuarios sin experiencia previa en programación.
7. Niveles de dificultad creciente en las lecciones: Las lecciones aumentan en complejidad a medida que los usuarios avanzan y el numero de preguntas para las lecciones mas complejas aumentara entre 10 a 15 preguntas por lección
8. Ejercicios de programación en JavaScript: Los usuarios practican escribiendo y ejecutando código JavaScript. Mediante imágenes de código el jugador deberá completar la salida del código o añadir lo que falta para su correcto funcionamiento
9. Sistema de retroalimentación para respuestas incorrectas: Los usuarios reciben retroalimentación útil cuando cometen errores en los ejercicios. Una breve descripción se mostrara después de q el jugador allá cometido un error para que no se quede con la duda de por que le quedo mal la respuesta
10. Opciones para repetir lecciones anteriores: Los usuarios pueden volver a tomar lecciones previas para reforzar su comprensión.
11. Retos y preguntas de opción múltiple: Los desafíos incluyen preguntas de opción múltiple para evaluar el conocimiento. Cada pregunta contendrá 4 opciones de respuesta
12. Exámenes de JavaScript al final de cada nivel: Los usuarios toman exámenes al completar niveles para evaluar su comprensión. Cada lección contendrá un pequeño quiz para evaluar la correcta comprensión de la sección serán preguntas de los temas que se tocaron en la lección pero evaluadas con preguntas o ejercicios diferentes
13. Historia o narrativa que guía al jugador: El juego contendrá una historia que motiva a los usuarios a avanzar en el aprendizaje ya que a medida que el jugador avanza la historia se desarrolla
14. Personajes con diálogos y roles en la historia: Los personajes interactúan con los usuarios y desempeñan papeles en la historia para guiar al jugador o darle pequeñas ayudas como tips sobre un tema en especifico
15. Animaciones y gráficos atractivos: Elementos visuales y animaciones se utilizan para hacer el juego más atractivo.
16. Sistema de monedas virtuales o recursos: Los usuarios ganan monedas virtuales o recursos que pueden utilizar para comprar mejoras.
17. Tienda virtual para adquirir mejoras y personalizaciones: Los usuarios pueden gastar monedas virtuales en mejoras para sus avatares u otras personalizaciones.
18. Compartir logros en redes sociales: Los usuarios pueden compartir sus logros y progreso en las redes sociales.
19. Modo sin conexión: el juego deberá mostrar un anuncio cuando se quede sin conexión y guardar el progreso del jugador
20. Estadísticas detalladas de progreso: Se proporciona información detallada sobre el progreso de aprendizaje de cada usuario.
21. Función de búsqueda: Los usuarios pueden buscar lecciones o temas específicos.
22. Sistema de registro de errores: Los usuarios pueden informar errores o problemas técnicos para mejorar la calidad del juego. Se proporcionará un canal de comunicación como correo con los desarrolladores del juego para que los jugadores puedan reportar alguna falla
23. Opción de invitación a amigos: Invitar a amigos a unirse al juego para competir o estudiar juntos.
24. Sistema de recomendación de lecciones: el juego deberá estructurar las lecciones de las bases y fáciles a lo especifico y difícil para que el jugador tenga un progreso gradual
25. Sistema de informes de seguridad: se deberá garantizar la protección de datos de usuario y la privacidad. Como el uso de contraseñas almacenadas en el inicio de sesión

Requerimientos de sistema

1. Entorno de Producción Seguro: El juego debe ser desplegado en un entorno de producción seguro con medidas de seguridad adecuadas para proteger los datos de los usuarios y el código fuente.
2. Alta Disponibilidad: El sistema debe garantizar una alta disponibilidad para que los usuarios puedan acceder al juego en cualquier momento.
3. Gestión de Versiones: Utilizar un sistema de gestión de versiones para controlar y rastrear cambios en el código fuente.
4. Automatización de Implementación: Automatizar el proceso de implementación para facilitar actualizaciones y correcciones de errores.
5. CDN (Red de Distribución de Contenido): Utilizar una CDN para entregar recursos estáticos (imágenes, archivos CSS y JavaScript) de manera eficiente.
6. Certificado SSL: Implementar un certificado SSL para habilitar la comunicación segura mediante HTTPS.
7. Registro de Auditoría: Registrar todas las acciones realizadas en el sistema para fines de auditoría y seguridad.
8. Política de Respuesta a Incidentes: Tener una política y procedimientos establecidos para abordar incidentes de seguridad.
9. Política de Copias de Seguridad: Definir una política de copias de seguridad que incluya frecuencia, retención y ubicación de respaldos.
10. Escaneo de Vulnerabilidades: Realizar escaneos regulares de vulnerabilidades en el sistema y remediar las vulnerabilidades detectadas.
11. Política de Retirada de Recursos Obsoletos: Establecer una política para retirar recursos y datos obsoletos o innecesarios.
12. Pruebas de Continuidad: Realizar pruebas regulares de continuidad del negocio para garantizar que el juego pueda seguir funcionando en caso de interrupciones.
13. Gestión de Contenido Estático en Caché: Implementar una estrategia de caché para el contenido estático y asegurar una entrega rápida.
14. Actualizaciones Programadas: Realizar actualizaciones programadas en momentos de baja demanda para minimizar interrupciones.
15. Monitorización de Rendimiento y Uso: Supervisar continuamente el rendimiento del sistema y el uso de recursos para optimizarlo y ajustarlo según sea necesario.

Requerimientos no funcionales

1. Usabilidad:

El juego debe ser fácil de usar y de navegación intuitiva, con una interfaz sencilla sin muchos componentes minimalista y con botones claros y accesibles

1. Rendimiento:

El juego debe funcionar de manera eficiente y responder de manera rápida a las interacciones del usuario, minimizando los tiempos de carga y las demoras.

1. Seguridad:

Los datos de usuario, incluyendo información de registro y progreso, deben ser almacenados de forma segura y protegidos contra accesos no autorizados.

1. Disponibilidad:

El juego debe estar disponible para su uso en línea durante la mayor parte del tiempo, con tiempos de inactividad planificados para mantenimiento.

1. Localización:

Si se prevé una audiencia internacional, el juego podría requerir soporte para la localización en diferentes idiomas y regiones.

1. Compatibilidad con Navegadores:

El juego debe ser compatible con una variedad de navegadores web modernos, como Chrome, Firefox, Safari y Edge.

1. Mantenibilidad:

El código del juego debe ser fácil de mantener y actualizar, con buenas prácticas de programación.

1. Cumplimiento Normativo:

El juego debe cumplir con las regulaciones y leyes locales relacionadas con la privacidad de datos y la accesibilidad.

1. Tiempo de Carga:

El tiempo de carga inicial del juego debe ser razonablemente corto para evitar que los usuarios se frustren.

1. Confiabilidad:

El juego debe ser confiable y resistente a fallas, minimizando los errores y las interrupciones del servicio.

1. Reducción de Carga del Servidor:

El juego debe implementar técnicas de caché y optimización para reducir la carga del servidor y mejorar la eficiencia.

1. Respaldo y Recuperación de Datos:

Debe existir un plan de respaldo y recuperación de datos para proteger la integridad de los datos del usuario en caso de fallos del sistema.

1. Informes y Estadísticas (opcional):

Si se implementa la funcionalidad de informes y estadísticas, se deben generar y mostrar estos informes de manera eficiente.

1. Colores:

los colores del juego deben ser ambientados a la historia deben ser colores tropicales y la ambientación de la app será hacia la isla con palmeras hojas y demás elementos decorativos de los divs y botones

1. Tamaño de Descarga:

El tamaño de la descarga inicial para los usuarios debe ser lo más pequeño posible para acelerar el tiempo de carga.

1. Compatibilidad con Pantallas Retina/HD:

Debe mostrar gráficos nítidos en pantallas de alta resolución.

1. Cumplimiento de Derechos de Autor: Garantizar que todo el contenido esté libre de infracciones de derechos de autor y licencias adecuadas.
2. Eficiencia de Memoria: Utilizar recursos de memoria de manera eficiente para no agotar los recursos del dispositivo.
3. Compatibilidad con Pantallas Pequeñas: Asegurarse de que la interfaz sea legible y usable en pantallas pequeñas.
4. Optimización de Caché del Navegador: Aprovechar la caché del navegador para reducir las solicitudes al servidor.
5. Estabilidad de la Conexión en Red Móvil: Funcionar de manera confiable en conexiones de red móvil con diferentes niveles de calidad de señal
6. Compatibilidad con Dispositivos Antiguos: Asegurarse de que el juego sea compatible con dispositivos y navegadores más antiguos para ampliar la audiencia.
7. Navegación Intuitiva: Los elementos de navegación y menús deben ser fáciles de entender y utilizar, con una estructura lógica. Deberán disponerse de manera accesible a los controles móviles
8. Tiempo de Carga de Páginas: Las páginas del juego deben cargar rápidamente para minimizar la frustración del usuario y mejorar la retención.
9. Consistencia Visual: Mantener una consistencia en la apariencia de los elementos visuales, como colores, fuentes y estilos, en todo el juego.
10. Legibilidad: El texto debe ser legible en diferentes tamaños de pantalla y fuentes, y cumplir con las pautas de accesibilidad.
11. Diseño Atractivo: El juego debe tener un diseño atractivo y visualmente agradable que atraiga a los usuarios y los mantenga interesados.
12. Feedback Visual y de Estado: Proporcionar feedback visual claro para indicar acciones realizadas o errores, como mensajes de éxito, advertencias y errores.
13. Animaciones y Transiciones Suaves: Cualquier animación o transición en la interfaz debe ser fluida y no causar ralentizaciones ni parpadeos incómodos
14. Espacios en Blanco Bien Utilizados: El diseño debe incluir espacios en blanco de manera efectiva para mejorar la legibilidad y la organización.
15. Adaptación a Diferentes Resoluciones: La interfaz debe adaptarse sin problemas a diversas resoluciones de pantalla sin pérdida de funcionalidad ni contenido.
16. Retroalimentación de Sonido: Si se incluyen efectos de sonido, deben ser agradables y no intrusivos, y los usuarios deben poder desactivarlos.
17. Compatibilidad con Modo Oscuro/Claro: Permitir a los usuarios elegir entre modos oscuro y claro para la interfaz según sus preferencias.
18. Gestos Intuitivos: Si se implementan gestos, deben ser intuitivos y consistentes con las expectativas del usuario.
19. Resiliencia a Fallos de Red: Diseñar la interfaz para manejar situaciones de pérdida de conexión o latencia de red de manera elegante.
20. Cumplimiento de Estándares de Diseño: Asegurarse de que el diseño cumpla con los estándares actuales de diseño de interfaces de usuario (UI) y experiencia de usuario (UX).
21. Orientación al Usuario Final: Obtener comentarios de los usuarios reales para realizar ajustes y mejoras continuas en la interfaz
22. Diseño Minimalista: Seguir un enfoque de diseño minimalista para evitar la saturación de información y facilitar la concentración del usuario.
23. Uso Eficiente de la Pantalla: Aprovechar eficientemente el espacio en pantalla para mostrar contenido relevante y minimizar desplazamientos innecesarios.
24. Reproducción de Medios en Línea y sin Interrupciones: Garantizar una reproducción fluida de elementos multimedia, como videos o audio, sin interrupciones.
25. Interfaz sin Desorden: Mantener la interfaz libre de elementos innecesarios o distracciones que puedan confundir al usuario.
26. Transiciones de Página Suaves: Asegurarse de que las transiciones entre páginas o secciones sean suaves y visuales para evitar desorientar a los usuarios.
27. Retroalimentación Significativa: Proporcionar retroalimentación clara y significativa a los usuarios en respuesta a sus acciones, aciertos y errores. Es decir que cuando en jugador cometa un error la retroalimentación sea específica del error cometido
28. Experiencia de Aprendizaje Continua: Facilitar una experiencia de aprendizaje fluida y continua, sin interrupciones innecesarias o bloqueos de contenido.
29. Estímulo a la Exploración: Fomentar la exploración y el descubrimiento a través de recompensas, desafíos y contenido adicional oculto.
30. Experiencia Multisensorial: es relevante, aprovechar los sentidos adicionales como el sonido para brindar una experiencia audiovisual al jugador como lo hace cualquier juego
31. Evaluación Formativa: Proporcionar retroalimentación constante y evaluación formativa para ayudar a los usuarios a identificar áreas de mejora
32. Experiencia de Usuario Basada en Historia: Crear una narrativa envolvente a lo largo del juego que guíe a los usuarios a través de lecciones y desafíos.
33. Navegación Basada en Objetivos: Permitir a los usuarios establecer metas de aprendizaje y proporcionar rutas de navegación claras para alcanzar esos objetivos.
34. Asistencia de Ayuda en Línea: Proporcionar acceso a recursos de ayuda en línea, tutoriales y documentación para resolver preguntas de los usuarios. Mostrar links de la documentación donde pueden estudiar un tema en especifico si el jugador tiene alguna duda