## ****Introducción****

En la actualidad, las consejerías familiares son cruciales en la promoción del bienestar familiar y social. El Ministerio de Inclusión Económica y Social (MIES) en Ecuador ha establecido normativas técnicas destinadas a fortalecer y regular las consejerías familiares, con el objetivo de garantizar la calidad de los servicios ofrecidos a los niños.

El bienestar familiar es cada vez más importante en la sociedad, por lo que resulta beneficioso contar con herramientas eficaces para trabajar en las consejerías familiares. Este proyecto se propone crear una página web que integre diversas herramientas y recursos para abordar las cuatro áreas principales de la norma técnica del MIES: Hábitos de higiene, desarrollo de juegos de interacción y actividades lúdicas, alimentación saludable y entorno seguro.

La necesidad de esta capacitación surge de un análisis de las demandas actuales dentro del entorno educativo, donde se evidencia la importancia de proporcionar a los docentes herramientas necesarias para abordar las consejerías familiares de manera efectiva. Se observa que los cambios sociales y culturales han impactado significativamente en las dinámicas familiares, lo que exige a los docentes contar con habilidades específicas para comprender y atender las necesidades de las familias en el ámbito escolar.

El enfoque metodológico utilizado para el diagnóstico de estas necesidades se basa en un enfoque cuantitativo, que permite una recolección objetiva de datos a través de encuestas aplicadas a los docentes. Este enfoque también incluye un análisis cualitativo, que profundiza en las percepciones y experiencias de los docentes respecto a las prácticas de consejería familiar en el contexto educativo.

Los resultados obtenidos de este diagnóstico revelan aspectos claves, como el nivel de conocimientos tecnológicos de los docentes, acceso a internet, la frecuencia de uso de recursos digitales en su práctica y su familiaridad con plataformas de enseñanza en línea. Estos datos proporcionan una línea base fundamental para el desarrollo del proyecto y orientan la creación de herramientas específicas adaptadas a las necesidades identificadas.

Para cumplir con los objetivos establecidos, es fundamental integrar los principios y prácticas del desarrollo de software en la creación de la página web educativa. Este proyecto emplea tecnologías básicas pero efectivas para asegurar la funcionalidad y accesibilidad del portal:

* HTML para la Estructura: La estructura de la página web se desarrolló utilizando HTML puro, lo que garantiza una base sólida y compatible con todos los navegadores. HTML permite organizar el contenido de manera semántica, facilitando su comprensión y accesibilidad.
* CSS para Estilos y Responsividad: Se utilizó CSS para diseñar y estilizar la página web, asegurando una apariencia atractiva y profesional. Además, se implementaron técnicas de diseño responsivo para garantizar que el sitio sea accesible y fácil de usar en diversos dispositivos, desde computadoras de escritorio hasta teléfonos móviles.
* JavaScript para Dinamismo e Interactividad: JavaScript se empleó para añadir dinamismo e interactividad a la página. Esta tecnología permite la creación de funcionalidades interactivas, como formularios dinámicos, validaciones en tiempo real y la manipulación del DOM para mejorar la experiencia del usuario.
* Evaluación de Usabilidad: La usabilidad de la plataforma se evaluará mediante pruebas con usuarios, asegurando que los docentes puedan navegar y utilizar las herramientas de manera intuitiva. Se harán ajustes basados en el feedback recibido para mejorar la accesibilidad y la funcionalidad general del sitio.

La conjunción de los conocimientos en educación inicial y desarrollo de software es crucial para el éxito de este proyecto. Desde el ámbito educativo, se aporta una comprensión profunda de las necesidades de los docentes y las mejores prácticas en consejería familiar. Desde el ámbito tecnológico, se implementan soluciones eficientes y accesibles que facilitan la capacitación y el acceso a recursos educativos.

El trabajo de campo realizado en diferentes puntos clave, como la interacción con la dirección del distrito y la entrega de encuestas a las docentes, permitió una comprensión profunda de las necesidades específicas del CDI MIES. Este acercamiento directo con los actores involucrados facilitó la identificación de temas de mejora y la formulación de estrategias adaptadas a las circunstancias particulares del contexto educativo. Además, se establecieron relaciones de colaboración con organizaciones dedicadas a la capacitación docente y al desarrollo de recursos educativos, lo que enriqueció el proceso de diseño y desarrollo del Portal Educativo.

El análisis de los resultados obtenidos del diagnóstico revela la existencia de varios desafíos que deben ser abordados para mejorar la calidad de las consejerías familiares en los CDI-MIES. Entre estos desafíos se destacan la actualización del material de apoyo utilizado en las capacitaciones, la importancia de capacitar a los docentes en el uso de plataformas educativas en línea y la identificación de temas específicos que deben ser incluidos en dichas plataformas.

Además, se identificó la necesidad de desarrollar estrategias de seguimiento y evaluación para monitorear el impacto de las capacitaciones y asegurar su efectividad a largo plazo.

Con base en estos hallazgos, se establecen objetivos claros y específicos para el proyecto, que incluyen la elaboración de información precisa sobre consejería familiar, la construcción de una página web con contenido innovador y la implementación de capacitaciones interactivas para los docentes del CDI MIES. Estos objetivos se apoyan en indicadores de resultado y monitoreo que permitirán evaluar el impacto y la efectividad de las acciones llevadas a cabo. De igual forma, se proponen diversas estrategias para alcanzar los objetivos establecidos, como la capacitación continua de los docentes en metodologías de enseñanza innovadoras.

Este proyecto no solo busca mejorar las habilidades de los docentes en consejería familiar, sino que también tiene como objetivo contribuir al desarrollo sostenible a través de la integración de la tecnología en el proceso educativo. Al proporcionar el acceso a la información y recursos educativos de calidad, se busca empoderar a las familias y mejorar el bienestar general de la comunidad educativa en su conjunto.

## ****Marco Teórico****

### ****HTML5****

HTML5, la más reciente evolución del Lenguaje de Marcado de Hipertexto ha revolucionado el desarrollo web desde su lanzamiento. Esta versión se creó para superar las limitaciones de versiones anteriores y proporcionar una plataforma más robusta y eficiente para crear aplicaciones web modernas. (W3C, 2014).

HTML5 ha sido un cambio de juego en la manera en que desarrollan y diseñan aplicaciones web, permitiendo así crear experiencias más ricas e interactivas para los usuarios. Las nuevas características y APIs de HTML5 han sido esenciales para desarrollar soluciones web avanzadas.

#### ****Características****

##### ****Soporte para Aplicaciones Web Complejas****

HTML5 introduce nuevas APIs que facilitan el desarrollo de aplicaciones web complejas. Entre ellas se encuentra la API de Geolocalización, que permite a las aplicaciones web acceder a la ubicación del usuario, y la API de Almacenamiento Web, que proporciona almacenamiento persistente de datos del lado del cliente. Estas innovaciones son particularmente útiles en proyectos donde la personalización y el almacenamiento local de datos son cruciales.

##### Canvas y SVG para Gráficos Dinámicos

El elemento <canvas> y el soporte para gráficos vectoriales escalables (SVG) permiten a los desarrolladores crear gráficos dinámicos y animaciones directamente en la página web. Estas herramientas son fundamentales para aplicaciones que requieren visualizaciones de datos en tiempo real, como paneles de control y herramientas analíticas. Personalmente, he utilizado <canvas> y SVG para desarrollar dashboards interactivos que mejoran la toma de decisiones basada en datos.

##### Mejoras en la Interactividad

HTML5 mejora la interactividad en las páginas web mediante la introducción de nuevas APIs, como la API de arrastrar y soltar (Drag and Drop) y la API de Historial, que permite a las aplicaciones web manipular el historial del navegador para mejorar la navegación. Estas mejoras hacen que la experiencia del usuario sea más fluida y personalizada.

##### Streaming de Video y Audio

Con el soporte nativo para video y audio mediante las etiquetas <video> y <audio>, HTML5 ha revolucionado el streaming de medios. Plataformas de video como YouTube y servicios de música como Spotify utilizan HTML5 para proporcionar contenido multimedia de alta calidad sin necesidad de plugins adicionales. Esta capacidad ha sido esencial en la creación de aplicaciones de streaming más eficientes y accesibles.

#### ****Beneficios****

##### Aplicaciones Web Progresivas (PWA)

Las Aplicaciones Web Progresivas (PWA) son aplicaciones web que utilizan HTML5 junto con otras tecnologías modernas como Service Workers y el Manifiesto de Aplicaciones Web para ofrecer experiencias similares a las aplicaciones nativas. Las PWAs pueden funcionar offline, enviar notificaciones push y ser instaladas en la pantalla de inicio del dispositivo del usuario. Estas aplicaciones han transformado la manera en que los usuarios interactúan con la web, proporcionando una experiencia más integrada y accesible.

##### Interactividad Mejorada

La introducción de nuevas APIs, como la API de arrastrar y soltar y la API de Historial, mejora la interactividad en las páginas web. Estas herramientas permiten a los desarrolladores crear interfaces de usuario más intuitivas y dinámicas, lo que resulta en una mejor experiencia del usuario.

##### Gráficos y Animaciones Dinámicos

El uso de <canvas> y SVG permite a los desarrolladores crear gráficos y animaciones dinámicos que se integran directamente en la página web. Estas herramientas son esenciales para aplicaciones que requieren visualizaciones de datos en tiempo real, mejorando la capacidad de análisis y la toma de decisiones.

##### Mejor Soporte para Multimedia

HTML5 ha transformado el streaming de video y audio al incluir soporte nativo para estos medios mediante las etiquetas <video> y <audio>. Esto elimina la necesidad de plugins adicionales y mejora la eficiencia y accesibilidad del contenido multimedia en la web.

#### **Referencias**

World Wide Web Consortium (W3C). (2014). HTML5 Specification. Recuperado de <https://www.w3.org/TR/html5/>

### JAVASCRIPT

JavaScript, creado por Brendan Eich en 1995, ha evolucionado significativamente desde su origen como un lenguaje de scripting básico hasta convertirse en una herramienta esencial para el desarrollo web moderno. Diseñado inicialmente para agregar interactividad a las páginas web, ha experimentado numerosas actualizaciones importantes, destacándose ECMAScript 6 en 2015, que introdujo nuevas funcionalidades como clases y módulos. JavaScript se ejecuta directamente en el entorno de ejecución sin necesidad de compilación previa, lo que permite la creación y modificación dinámica de objetos en tiempo real. Su integración con HTML y CSS facilita la construcción de interfaces web interactivas (Flanagan, 2020).

#### Características

JavaScript se ejecuta sin necesidad de compilación previa y su naturaleza dinámica permite la creación y modificación de objetos en tiempo real. Es conocido por manejar eventos y su integración con HTML y CSS. Soporta varios paradigmas de programación, destacando la programación funcional y orientada a objetos, facilitada por funciones de orden superior y clausuras. En el frontend, es indispensable para crear aplicaciones web interactivas utilizando frameworks como React, Angular y Vue.js. En el backend, Node.js ha ampliado su uso, permitiendo manejar tareas del servidor y construir APIs robustas (Flanagan, 2020).

#### Beneficios

Los beneficios de JavaScript incluyen su flexibilidad, la posibilidad de desarrollo rápido y eficiente, y su capacidad para crear aplicaciones escalables y altamente interactivas. Su uso tanto en el frontend como en el backend, gracias a frameworks como React, Angular, Vue.js, y Node.js, así como en el desarrollo móvil con frameworks como React Native y Ionic, lo consolidan como una herramienta versátil y poderosa en la industria tecnológica (Flanagan, 2020).

#### Referencia

Flanagan, D. (2020). *JavaScript: The Definitive Guide: Master the World's Most-Used Programming Language* (7th ed.). O'Reilly Media.

### UI

La Interfaz de Usuario (UI) se refiere a cómo el usuario se guía mientras utiliza una aplicación de software o dispositivo de hardware. La UI es un concepto básico y esencial para la interacción con cualquier software (Lenis, 2023).

La UI hace que el usuario realice las acciones deseadas de manera natural e intuitiva. Por ejemplo, cuando una persona visita un sitio web, hay ciertas acciones específicas que se espera que tome. Un buen diseño de UI guiará a los usuarios en la navegación, llevándolos a tomar esas acciones de manera intuitiva. Si el objetivo es que el usuario contacte a un vendedor, todo en la página de colaborar para generar esa necesidad en el usuario (Rockcontent, 2019).

#### Características

El atractivo visual es fundamental en la UI: “Esto debe ser una prioridad de los equipos de desarrollo de interfaz del usuario, ya que permite que el usuario se sienta identificado y cómodo con el programa" (Lenis, 2023).

Se debe transmitir la información de manera clara y concisa al usuario, para que al momento de que interactúe con el programa no haya errores e inconsistencias (Lenis, 2023),

Mantener la coherencia y unidad en el diseño ayuda en que “los usuarios pueden crear patrones de uso de forma intuitiva, sin la necesidad de aprender procesos muy complejos que podrían desinhibir su uso de tecnología" (Lenis, 2023).

#### Beneficios

La interfaz de Usuario cumple un papel fundamental en el mundo digital, satisfaciendo la experiencia que el usuario tiene con un software o aplicación. Una interfaz atractiva puede significar la diferencia entre un buen producto, que sea exitoso, de otro que fracasa en el mercado. Así las empresas modernas reconocen que una buena interfaz es esencial para la satisfacción del usuario (Jesús, 2023).

#### Referencias

Lenis, A. (17 de 05 de 2023). Qué es la interfaz de usuario, qué tipos existen y ejemplos. Hubspot. <https://blog.hubspot.es/website/interfaz-usuario>

Rockcontent. (3 de noviembre de 2019). Diseño UI y UX: ¡descubre cuál es la diferencia entre ambos! Rockcontent. <https://rockcontent.com/es/blog/ui-ux/>

Jesús. (29 de octubre de 2023). ¿Qué es la Interfaz de Usuario? Una Guía Completa para Principiantes. Dongee. <https://www.dongee.com/tutoriales/que-es-la-interfaz-de-usuario/>

### UX

La experiencia de usuario (UX) se define en el contacto que tiene el usuario con los productos o servicio digital de una empresa. La responsabilidad de las empresas es proporcionar a los usuarios soluciones que sean agradables, intuitivas, atractivas y eficientes, logrando que la interacción sea eficiente. Comprender y aplicar las mejores prácticas de UX en una empresa puede ser una manera muy eficiente de mejorar la calidad de satisfacción de los clientes (Zendesk, 2024).

#### Características

##### Útil

Los productos que tienen el mejor diseño UX son aquellos que cumplen con un propósito especifico. Un producto que no es útil tendrá poca demanda. Sin embargo, si el producto final proporciona una solución adecuada para el usuario el objetivo, tendrá una mayor probabilidad de éxito (Salinas, 2022).

##### Usable

Aunque un producto puede cumplir su propósito sin una alta frecuencia de usabilidad, es menos posible que tenga éxito en el mercado. Pero un producto que facilita y agiliza las tareas diarias de los usuarios destacará en el mercado gracias a su usabilidad (Salinas, 2022).

##### Encontrable

La capacidad de encontrar se refiere a asegurar que los usuarios puedan localizar y acceder a un producto cuando lo necesiten. En el caso de un producto, como un sitio web, esto implica que los usuarios puedan encontrar el contenido que buscan de manera sencilla (Salinas, 2022).

##### Accesible

Mejorar la accesibilidad implica eliminar la mayor cantidad posible de obstáculos entre el producto y el cliente (Salinas. 2022).

#### Beneficios

La experiencia de usuario (UX), ayuda a entender las necesidades y expectativas que tiene el usuario, así poder tener un mayor índice de retención, con un buen diseño para analizar el comportamiento del usuario, guiándolo a realizar las acciones deseadas por la empresa. El UX optimiza la funcionalidad del producto, asegurando que tenga utilidad (Unir, 2024).

#### Referencias

Zendesk. (31 de 01 de 2024). ¿Qué es UX y para qué sirve? Obtenido de zendesk: <https://www.zendesk.com.mx/blog/ux-que-es/>

Danae, S. (28 de julio de 2022). ¿Qué es UX? Todo lo que necesitas saber. Wixblog. <https://es.wix.com/blog/que-es-ux#viewer-bo3tk>

Unir. (09 de abril de 2024). ¿Qué es UX/UI y cuáles son sus ventajas? Unir. <https://mexico.unir.net/noticias/ingenieria/que-es-ux-ui/>

### CSS

Según Santos (2023) CSS con las siglas en inglés (Cascading Style Sheets). Es un lenguaje de programación para dar estilo y presentación a una página web, es decir, lo que va a visualizar un usuario al visitar la página. CSS funciona a la par que el Lenguaje de Marcado HTML, el cual es el encargado del contenido de la página. Además, agrega que la denominación de hojas de estilo en cascada se debe a que se puede tener varias de ellas con las propiedades heredadas de otras.

#### Características

CSS tiene una sintaxis relativamente sencilla que consiste, en sus propiedades, valores y bloques de declaración (Fernández, 2022).

Con el lenguaje de diseño CSS “puedes crear reglas para decirle a tu sitio web cómo quieres mostrar la información y guardar los comandos para elementos de estilo (como fuentes, colores, tamaños, etc.) separados de los que configuran el contenido” (Santos, 2023).

Por ser un lenguaje popular para dar formato a páginas web, su uso es amplio para un gran número de dispositivos, formatos y plataformas como Edge, Safari, Chrome, etc. Por esto se facilita dar formato a las páginas según el navegador utilizado por cada usuario (Santos, 2023).

#### Beneficios

Usar CSS para el desarrollo web es importante para lograr un diseño y una presentación coherente en las páginas. Utilizando CSS, se hace posible separar el diseño del contenido, garantizando la consistencia en las páginas. Nos ayuda a tener flexibilidad y eficiencia en la estilización, y nos permite adaptar el diseño a diferentes dispositivos. CSS es una herramienta indispensable para la creación de páginas web (Ridge, 2023).

#### Referencias

Santos, D. (28 de julio de 2023). Introducción al CSS: qué es, para qué sirve y otras 10 preguntas frecuentes. HubSpot. <https://blog.hubspot.es/website/que-es-css>

Fernández. E. (10 de marzo de 2022). Propiedades de CSS: cuáles son y su importancia. TokioSchool. <https://www.tokioschool.com/noticias/propiedades-css/>

Ridge, B. (24 de agosto de 2023). La importancia de aprender CSS en el desarrollo web. MediumMultimedia. <https://www.mediummultimedia.com/web/es-importante-aprender-css/>

### ****MAQUETACIÓN****

La maquetación en el desarrollo de software es crucial para asegurar una experiencia de usuario intuitiva y agradable. Organizar los elementos visuales y estructurales de manera coherente facilita la navegación y mejora la accesibilidad del software. Además, contribuye a la eficiencia del desarrollo y mantenimiento al permitir una separación clara entre el contenido y el diseño.

La maquetación se refiere al proceso de diseñar y organizar los elementos visuales y estructurales de una aplicación o sitio web, incluyendo texto, imágenes, gráficos y videos, para lograr una presentación estética y funcional. No solo se enfoca en la apariencia visual, sino también en la usabilidad y accesibilidad del software (HubSpot, 2024; Cyberstream, 2024).

#### ****Características****

##### Flexibilidad

Permite una gran flexibilidad en el diseño y la presentación de contenido, lo que significa que los diseñadores pueden crear interfaces únicas y personalizadas para satisfacer las necesidades específicas de los usuarios.

##### Adaptabilidad

Con la proliferación de dispositivos con diferentes tamaños de pantalla, la maquetación moderna se ha vuelto altamente adaptable. Las interfaces pueden diseñarse para responder, adaptándose automáticamente al tamaño de la pantalla del dispositivo.

##### SEO (Optimización para Motores de Búsqueda)

Una maquetación bien estructurada y optimizada puede mejorar el rendimiento de un sitio web en los motores de búsqueda.

##### Facilidad de mantenimiento

Separar el contenido del diseño mediante la implementación de HTML y CSS facilita el mantenimiento a largo plazo del software.

#### ****Beneficios****

##### Mejora de la experiencia del usuario

Una buena maquetación asegura que los usuarios puedan navegar y utilizar el software de manera intuitiva y eficiente.

##### Aumento de la accesibilidad

Diseñar con accesibilidad en mente permite que el software sea utilizado por personas con diversas capacidades.

##### Optimización del rendimiento

Una maquetación eficiente puede contribuir a tiempos de carga más rápidos y un mejor rendimiento general del software.

#### **Referencias**

HubSpot. (2024). Maquetación web: qué es, ejemplos y características. Recuperado de <https://blog.hubspot.es/website/maquetacion-web>.

Cyberstream. (2024). Guía completa sobre maquetar en software: definición y pasos esenciales. Recuperado de <https://www.byronvargas.com/web/que-es-maquetar-en-software/>.

### VISUAL STUDIO CODE

Visual Studio Code (VS Code), creado por Microsoft, es un editor de código fuente que ha ganado popularidad entre los desarrolladores de software a nivel global. Gracias a su interfaz intuitiva, amplia gama de extensiones y robusto conjunto de funcionalidades, VS Code proporciona una experiencia de desarrollo eficiente y adaptable para diversos lenguajes de programación y tecnologías.

#### Características:

Aiweb Learning (2024) afirma que Visual Studio Code cuenta con una variedad de características que lo hacen único y muy apreciado por los desarrolladores. Algunas de sus principales características son:

Interfaz de Usuario Intuitiva: VS Code tiene una interfaz fácil de usar que simplifica la navegación y la escritura de código.

Soporte para Múltiples Lenguajes de Programación: VS Code admite una gran variedad de lenguajes de programación, como JavaScript, Python, Java, C#, HTML, CSS, entre otros.

Extensiones Personalizables: Con una amplia gama de extensiones disponibles en el Marketplace de Visual Studio Code, los usuarios pueden personalizar el editor para satisfacer sus necesidades y preferencias específicas.

Depuración Integrada: VS Code incluye herramientas de depuración que permiten a los desarrolladores depurar su código de manera eficiente y resolver problemas rápidamente.

Control de Versiones Integrado: Con integración nativa con sistemas de control de versiones como Git, VS Code facilita la colaboración y el trabajo en equipo en proyectos de desarrollo de software.

#### Beneficios de Utilizar Visual Studio Code:

Visual Studio Code ofrece numerosas ventajas que lo convierten en uno de los editores de código fuente más populares y utilizados hoy en día. Algunas de sus principales ventajas son:

Gratuito y de Código Abierto: Visual Studio Code es totalmente gratuito y de código abierto, lo que permite a cualquier persona descargarlo, utilizarlo y contribuir a su desarrollo.

Ligero y Rápido: A diferencia de otros editores de código más pesados, VS Code es ligero y rápido, garantizando un rendimiento óptimo incluso en equipos con recursos limitados.

Altamente Personalizable: Gracias a su amplia gama de extensiones y opciones de configuración, Visual Studio Code es altamente personalizable y adaptable a las necesidades de cada desarrollador.

Soporte para Diversas Tecnologías: VS Code admite una gran variedad de tecnologías y herramientas, haciéndolo una opción versátil para diferentes proyectos de desarrollo de software.

Gran Comunidad y Ecosistema: Visual Studio Code cuenta con una extensa comunidad de desarrolladores y una amplia variedad de extensiones y recursos disponibles en el Marketplace, lo que facilita la colaboración y el intercambio de conocimientos.

##### Visual Studio Code para la Web:

Flores (2022) afirma que Visual Studio Code para la Web es una versión que se ejecuta directamente en el navegador, eliminando la necesidad de instalación. Facilita la visualización y edición rápida de repositorios de código fuente y se puede acceder a través de [vscode.dev](https://vscode.dev/).

Esta versión web incluye muchas de las características de VS Code Desktop, como la búsqueda, autocompletado y resaltado de sintaxis. También soporta extensiones y permite abrir repositorios desde GitHub y Azure Repos, además de trabajar con código almacenado localmente. No obstante, tiene algunas limitaciones en comparación con la versión de escritorio: no cuenta con una terminal integrada, no permite compilar o depurar código, y solo algunas extensiones están disponibles para la versión web.

#### Referencias

Aiweb Learning. (2024, Mayo 15). ¿Qué es Visual Studio Code y cuáles son sus ventajas? - Ai web Learning. Ai Web Learning. <https://aiweblearning.com/que-es-visual-studio-code-y-cuales-son-sus-ventajas/>

Flores, F. (2022, Julio 22). *Qué es Visual Studio Code y qué ventajas ofrece*. OpenWebinars.net. <https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/##qu%C3%A9-es-visual-studio-code>

### EXTENSIONES DE VISUAL STUIDO CODE

"Las extensiones de Visual Studio Code (VS Code) son herramientas poderosas que pueden mejorar significativamente la experiencia de desarrollo, estas extensiones ofrecen una amplia gama de funcionalidades, desde la mejora de la legibilidad del código hasta la facilitación de la colaboración en equipo" (Visual Studio Code, n.d.).

##### Paste JSON as Code (Conversión de JSON a Código)

Paste JSON as Code convierte automáticamente estructuras JSON en código TypeScript o C#, facilitando el manejo de datos estructurados en aplicaciones.

Dicha extensión convierte JSON a tipos de datos en TypeScript o C#, genera interfaces y clases a partir de estructuras JSON e integración directa con el portapapeles de Windows.(Visual Studio Code, n.d.).

#### Beneficios:

Ahorra tiempo al generar automáticamente el código correspondiente a estructuras.

Reduce errores manuales en la creación de tipos de datos.

Facilita el manejo y validación de datos en aplicaciones.

##### Simple React Snippets (Inserción Rápida de Código para React)

Simple React Snippets proporciona fragmentos de código reutilizables para React, acelerando el desarrollo de componentes y funcionalidades comunes.

Esta extensión tiene amplia colección de snippets para React, incluyendo componentes, hooks y contextos, compatible con JavaScript y TypeScript, además de la personalización y creación de snippets personalizados. (Visual Studio Code, n.d.).

#### Beneficios:

Acelera el desarrollo de aplicaciones React mediante la reutilización de código.

Reduce errores y tiempo de escritura de código repetitivo.

Facilita la adopción de buenas prácticas en el desarrollo con React.

##### Quokka.js (Ejecución en Tiempo Real)

Es una extensión para la ejecución de código JavaScript y TypeScript en tiempo real, proporcionando resultados instantáneos directamente en el editor.

Dicha extensión permite la ejecución en tiempo real de JavaScript y TypeScript, muestra resultados de ejecución en línea, soporte para múltiples frameworks y bibliotecas y la depuración rápida con feedback instantáneo. (Visual Studio Code, n.d.).

#### Beneficios:

Mejora la eficiencia al proporcionar resultados inmediatos al escribir código.

Facilita la experimentación y aprendizaje de nuevas funcionalidades.

Aumenta la productividad al permitir una depuración rápida y efectiva.

##### TypeScript Importer (Automatización de Importaciones)

TypeScript Importer facilita la automatización de importaciones en archivos TypeScript, sugiriendo y agregando automáticamente las importaciones necesarias mientras se escribe código.

Esta extensión da sugerencias automáticas de importaciones, agrega importaciones a medida que se escriben nuevos símbolos, configuración de rutas de importación y alias, integración con proyectos TypeScript y JavaScript. (Visual Studio Code, n.d.).

#### Beneficios:

Aumenta la eficiencia al eliminar la necesidad de escribir manualmente.

Reduce errores por importaciones faltantes.

Facilita la navegación y gestión de dependencias en proyectos grandes.

#### Referencia

*Visual Studio Code. (n.d.).* [*VS Code Extensions*](https://code.visualstudio.com/docs/editor/extension-marketplace)*.* [*https://code.visualstudio.com/docs/editor/extension-marketplace*](https://code.visualstudio.com/docs/editor/extension-marketplace)

### ****REPOSITORIO DE ALMACENAMIENTO****

Un repositorio, o repo, es un tipo de almacenamiento digital centralizado que los desarrolladores utilizan para realizar y administrar cambios en el código fuente de una aplicación. Los desarrolladores tienen que almacenar y compartir carpetas, archivos de texto y otros tipos de documentos al desarrollar software. Un repositorio cuenta con características que permiten a los desarrolladores rastrear con facilidad cambios en el código, editar archivos de manera simultánea y colaborar de forma eficiente en el mismo proyecto desde cualquier ubicación. (Amazon, 2023)

Un repositorio de almacenamiento es una herramienta que sirve para guardar y gestionar el código de los proyectos que se realizan. Es un como una carpeta central en la que pueden almacenar todo el código y los archivos relacionados con su aplicación. Además, permite hacer un seguimiento de los cambios que hacen en el código, trabajar en equipo de manera coordinada y colaborar con otros desarrolladores sin importar dónde se encuentren.

#### ****Características****

##### ****Ramificación****

La ramificación es el proceso de creación de una copia del repositorio en GitHub. Por defecto, un repositorio de GitHub tiene una rama principal que contiene los códigos fuente originales y los archivos que los desarrolladores suben inicialmente. Si necesitan modificar una parte concreta del código, crean una ramificación que contenga una copia completa de los archivos de código y etiquetan la ramificación en consecuencia. (Amazon, 2023)

##### ****Confirmación****

La confirmación es una característica que permite a los desarrolladores guardar todos los cambios que han realizado en los archivos de código de la ramificación. GitHub permite a los desarrolladores, cuando envían cambios, describir las nuevas modificaciones que han introducido en el código. (Amazon, 2023)

##### ****Solicitud de extracción****

Una solicitud de extracción envía un mensaje oficial a otros colaboradores que trabajan en la ramificación principal o al propietario de esta para que revisen los cambios que se han confirmado. Los colaboradores pueden revisar, comentar o realizar cambios adicionales en la confirmación original antes de enviarla para su fusión. Cuando revisan una solicitud de extracción, los colaboradores pueden ver los cambios en el código original. (Amazon, 2023)

##### ****Fusión****

La fusión es un proceso de GitHub que aplica los cambios que se han confirmado a la ramificación principal. Esto se hace cuando los colaboradores han revisado y aprobado los cambios confirmados. A continuación, activan la solicitud de fusión en GitHub, que pone en marcha una serie de comprobaciones antes de fusionar los cambios con el código original. (Amazon, 2023)

#### ****Beneficios****

Las empresas son más rápidas y responden mejor a las exigencias en constante evolución de los consumidores cuando utilizan repositorios para el desarrollo de aplicaciones. Los desarrolladores pueden trabajar rápidamente en características nuevas sin que ello afecte a la estabilidad de la aplicación en funcionamiento. Los repositorios permiten a los desarrolladores introducir cambios y resolver problemas potenciales con rapidez. Las organizaciones también pueden coordinar las tareas de desarrollo de aplicaciones entre desarrolladores que trabajan a distancia.

#### Referencias

Amazon. (2023). *Que es Repositorio*. Recuperado el 20 de junio de 2024, de AWS Amazon: <https://aws.amazon.com/es/what-is/repo/#:~:text=Un%20repositorio%2C%20o%20repo%2C%20es,de%20documentos%20al%20desarrollar%20software>.

### SCRUM

SCRUM es un marco de trabajo ágil para la gestión de proyectos originados en la industria del software, pero se ha adoptado en varios campos por su efectividad. Desarrollado por Ken Schwaber y Jeff Sutherland en la década de 1990, SCRUM se basa en principios de iteración e incremento, promoviendo la entrega continua de productos funcionales (Schwaber & Sutherland, 2020).

#### Características

Su estructura incluye roles definidos como el Scrum Master, Product Owner y el equipo de desarrollo, que trabajan en ciclos de trabajo cortos llamados sprints. Estos sprints permiten una revisión y adaptación constantes, mejorando la capacidad de respuesta a cambios y garantizando que el producto final cumpla con las necesidades del cliente (Schwaber, 2004).

#### Beneficios

Los beneficios de SCRUM incluyen una mayor transparencia, mejor colaboración entre equipos y stakeholders, y la capacidad de entregar productos de alta calidad en plazos más cortos (Rubin, 2013).

#### Fases

El proceso SCRUM se compone de varias fases clave. Inicia con la planificación del sprint, donde se define el trabajo a realizar. Luego sigue la ejecución del sprint, durante la cual el equipo trabaja en las tareas seleccionadas. A lo largo del sprint, se llevan a cabo reuniones diarias (Daily Stand-ups) para coordinar esfuerzos y resolver problemas. Al final del sprint, se realiza una revisión (Sprint Review) para evaluar el trabajo completado y una retrospectiva (Sprint Retrospective) para identificar mejoras en el proceso (Schwaber & Sutherland, 2020).

#### Roles

En SCRUM, los roles principales incluyen el Scrum Master, quien facilita el proceso y elimina impedimentos; el Product Owner, quien gestiona el backlog del producto y prioriza las tareas para maximizar el valor del producto; y el equipo de desarrollo, que es multifuncional y auto-organizado, encargado de convertir las tareas seleccionadas en incrementos de producto funcionales durante cada sprint (Rubin, 2013; Schwaber, 2004).

#### Referencia

Rubin, K. S. (2013). *Essential Scrum: A Practical Guide to the Most Popular Agile Process*. Addison-Wesley.

Schwaber, K. (2004). *Agile Project Management with Scrum*. Microsoft Press.

Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide*. Scrum.org.

## Marco Conceptual

**API (Interfaz de Programación de Aplicaciones):** Un conjunto de definiciones y protocolos que permiten la comunicación entre diferentes aplicaciones de software. Las API facilitan la integración de diversas aplicaciones y servicios en una página web. (Sommerville, 2015)   
Referencia: Sommerville, I. (2015). *Software Engineering* (10th ed.). Pearson.

**Canvas**: Un elemento HTML5 que permite a los desarrolladores dibujar gráficos y animaciones directamente en la página web utilizando JavaScript. Es fundamental para visualizaciones de datos y aplicaciones interactivas. (Mozilla\_Developer\_Network, 2021)  
Referencia: Mozilla Developer Network. (2021). *Canvas API*. Retrieved from [**https://developer.mozilla.org/en-US/docs/Web/API/Canvas\_API**](https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API)

**Etiqueta (Tag):** Una etiqueta es un componente del lenguaje HTML que se utiliza para marcar el inicio y el final de un elemento. Las etiquetas se utilizan para estructurar y dar formato al contenido de las páginas web. (W3C, 2014)  
Referencia: W3C. (2014). *HTML5: A vocabulary and associated APIs for HTML and XHTML*. Retrieved from [**https://www.w3.org/TR/html5/**](https://www.w3.org/TR/html5/)

**Geolocalización:** Tecnología que permite determinar la ubicación geográfica de un dispositivo. En HTML5, la API de Geolocalización permite que las aplicaciones web accedan a la ubicación del usuario. (Mozilla\_Developer\_Network, 2021  
Referencia: Mozilla Developer Network. (2021). *Geolocation API*. Retrieved from [**https://developer.mozilla.org/en-US/docs/Web/API/Geolocation\_API**](https://developer.mozilla.org/en-US/docs/Web/API/Geolocation_API)

**HTML (Lenguaje de Marcado de Hipertexto):** El estándar principal para crear y estructurar páginas web. HTML utiliza etiquetas para definir elementos como párrafos, encabezados, enlaces e imágenes. (W3C, 2014)  
Referencia: W3C. (2014). *HTML5: A vocabulary and associated APIs for HTML and XHTML*. Retrieved from [**https://www.w3.org/TR/html5/**](https://www.w3.org/TR/html5/).

**Motor de Búsqueda:** Un sistema de software diseñado para buscar información en la web. Los motores de búsqueda indexan sitios web y permiten a los usuarios encontrar páginas relevantes *a través de palabras clave.* (Brin & Page, 1998)  
*Referencia:* Brin, S., & Page, L. (1998). The Anatomy of a Large-Scale Hypertextual Web Search Engine. Computer Networks and ISDN Systems.

**PWA (Aplicaciones Web Progresivas):** *Aplica*ciones web que utilizan tecnologías modernas para ofrecer experiencias similares a las aplicaciones nativas. Pueden funcionar offline, enviar notificaciones push y ser instaladas en la pantalla de inicio del dispositivo del usuario. (Google\_Developers, 2020)  
Referencia: Google Developers. (2020). *Progressive Web Apps*. Retrieved from [**https://developers.google.com/web/progressive-web-apps**](https://developers.google.com/web/progressive-web-apps)

**Plugin:** Un componente de software que añade funciones específicas a una aplicación existente. En el contexto de los navegadores web, los plugins permiten la reproducción de contenido multimedia, gráficos avanzados, y otras funcionalidades. (Su, 2010)   
Referencia: Su, J. (2010). *Understanding Browser Plug-ins*. Journal of Computing Sciences in Colleges, 25(6), 183-190.

**Servidor:** Un sistema de computación que proporciona recursos, datos, servicios o programas a otros ordenadores, conocidos como clientes, a través de una red. Los servidores web alojan sitios web y responden a las solicitudes de los navegadores.(Tanenbaum & Van\_Steen, 2016)   
Referencia: Tanenbaum, A. S., & Van Steen, M. (2016). *Distributed Systems: Principles and Paradigms* (2nd ed.). Pearson.

**SVG (Gráficos Vectoriales Escalables):** Un formato de imagen basado en XML para describir gráficos vectoriales bidimensionales. SVG permite la creación de gráficos que se pueden escalar a cualquier tamaño sin perder calidad. (Mozilla\_Developer\_Network, 2021)  
Referencia: Mozilla Developer Network. (2021). *SVG*. Retrieved from [**https://developer.mozilla.org/en-US/docs/Web/SVG**](https://developer.mozilla.org/en-US/docs/Web/SVG)

JavaScript: Un lenguaje de programación utilizado tanto en el desarrollo del lado del cliente como en el servidor. Creado por Brendan Eich en 1995, es conocido por su capacidad para manejar eventos e integrarse con HTML y CSS para crear aplicaciones web interactivas (Flanagan, 2020).

ECMAScript 6 (ES6): Una actualización importante de JavaScript lanzada en 2015 que introdujo características como clases y módulos, mejorando la capacidad del lenguaje para manejar aplicaciones complejas (Flanagan, 2020).

Lenguaje Interpretado: Un lenguaje de programación en el cual el código se ejecuta en tiempo real por un intérprete, en lugar de ser compilado previamente. Esto permite la creación y modificación de objetos en tiempo de ejecución (Flanagan, 2020).

Prototipos: Un mecanismo en JavaScript mediante el cual los objetos pueden heredar propiedades y métodos de otros objetos, permitiendo la extensión en tiempo de ejecución (Flanagan, 2020).

Programación Funcional: Un paradigma de programación en JavaScript que se facilita mediante el uso de funciones de orden superior y clausuras, permitiendo la creación de código modular y reutilizable (Flanagan, 2020).

Programación Orientada a Objetos: Un paradigma de programación en JavaScript que se logra a través de prototipos y, más recientemente, clases introducidas en ECMAScript 6, facilitando la creación de objetos y la reutilización de código (Flanagan, 2020).

Frontend: La parte de una aplicación web que interactúa directamente con el usuario, donde JavaScript se utiliza junto con HTML y CSS para crear interfaces de usuario interactivas. Frameworks populares incluyen React, Angular y Vue.js (Flanagan, 2020).

Backend: La parte de una aplicación web que se ejecuta en el servidor, donde JavaScript se utiliza con Node.js para manejar tareas del servidor, procesos asíncronos y construir APIs robustas (Flanagan, 2020).

Node.js: Un entorno de ejecución para JavaScript que permite ejecutar el código del lado del servidor, facilitando el desarrollo de aplicaciones de servidor eficientes y escalables (Flanagan, 2020).

Document Object Model (DOM): Una representación estructural de una página web en la cual JavaScript puede interactuar para manipular y actualizar la estructura y el contenido del documento (Flanagan, 2020).

CTAs: Un call to action (CTA) es una llamada a la acción mediante la que se busca atraer a usuarios potenciales hasta convertirles en clientes finales a través de una pieza gráfica o enlace con un fuerte poder de atracción. Se trata de un elemento clave para generar nuevos clientes uniendo dos piezas complejas en el proceso de captación: el tráfico entrante y las oportunidades mediante las que convertir las nuevas visitas en clientes (MOLINA, 2022). Referencia: MOLINA, D. (27 de 01 de 2022). Qué es un call to action y ejemplos. Obtenido de iebschoo: [**https://www.iebschool.com/blog/como-hacer-llamada-a-la-accion-call-to-action-e-commerce/**](https://www.iebschool.com/blog/como-hacer-llamada-a-la-accion-call-to-action-e-commerce/)

Customer Journey: El Journey o viaje, comienza con la recolección de objetivos y acciones del usuario en un boceto con una línea de tiempo. Posteriormente, al boceto, se le añaden los pensamientos y emociones de los usuarios para poder crear la narración. Como paso final, esta narración se reduce y perfila, lo que finalmente lleva a la visualización (Prado, 2020). Referencia: Prado, J. A. (22 de 03 de 2020). Qué es el Customer Journey Map y cómo hacerlo. Obtenido de uxables: [**https://www.uxables.com/investigacion-ux/que-es-el-customer-journey-map-y-como-hacerlo/**](https://www.uxables.com/investigacion-ux/que-es-el-customer-journey-map-y-como-hacerlo/)

Estilos en Cascada: “La cascada en las hojas de estilo significa que el orden de las reglas importa en CSS: cuando dos reglas tienen la misma especificidad, se aplica la que aparece en último lugar en el CSS” (MDN, s.f.). Referencia: MDN Web Docs (MDN). (s.f.). Cascada y herencia. [**https://developer.mozilla.org/es/docs/Learn/CSS/Building\_blocks/Cascade\_and\_inheritance**](https://developer.mozilla.org/es/docs/Learn/CSS/Building_blocks/Cascade_and_inheritance)

Selector: “Los selectores de CSS son básicamente unos códigos que nos permiten indicar el elemento o elementos sobre los que queremos aplicar determinados estilos” (Alvarez, 2023). Referencia: Miguel, A. (2 de mayo de 2023). Selectores de CSS. Desarrolloweb. [**https://desarrolloweb.com/articulos/selectores-css**](https://desarrolloweb.com/articulos/selectores-css)

**Propiedades**: “Las propiedades CSS son elementos que determinan cómo se visualiza y se comporta el contenido en una página web HTML” (Apinem, s.f.). Referencia: Apiem. (s.f.). Propiedades CSS: Descubre los más usados y ejemplo!!. Apiem. [**https://www.apinem.com/propiedades-css/**](https://www.apinem.com/propiedades-css/)

**Responsive**: “Diseño que se adapta a diferentes tamaños de pantalla” (HubSpot, 2023). Recuperado de <https://blog.hubspot.es/website/diseno-responsive>

**Grid**: “Sistema para alinear y distribuir elementos web” (MDN Web Docs, 2023). Recuperado de <https://developer.mozilla.org/es/docs/Web/CSS/CSS_grid_layout>

**SEO**: “Técnicas para aumentar la visibilidad online” (Semrush, 2024). Recuperado de <https://es.semrush.com/blog/seo-visibilidad/>

**Usabilidad**: “Facilidad con la que se interactúa con el software” (Blogs UOC, sin fecha). Recuperado de <https://blogs.uoc.edu/informatica/es/usabilidad-que-es-y-cuales-son-sus-principios/>

**Accesibilidad**: “Diseño inclusivo para usuarios con distintas capacidades” (Platzi, 2024). Recuperado de <https://platzi.com/blog/como-el-diseno-inclusivo-y-la-accesibilidad-estan-moldeando-el-mundo-digital/>

Depuración: La depuración es el proceso de identificar y corregir errores en el código fuente de un software. Cuando un programa no funciona como se espera, los programadores examinan el código para determinar la causa de los errores. Para ello, utilizan herramientas de depuración que les permiten ejecutar el software en un entorno controlado, revisar el código línea por línea, analizar el problema y resolverlo (aws, 2022). Referencia: aws. (2022). ¿Qué es la depuración? - Explicación de la depuración - AWS. Amazon Web Services, Inc. https://aws.amazon.com/es/what-is/debugging/#:~:text=La%20depuraci%C3%B3n%20es%20el%20proceso,por%20qu%C3%A9%20ocurren%20algunos%20errores.

Compilar: Compilar es el proceso de convertir un programa informático escrito en un lenguaje en un conjunto de instrucciones en otro formato o lenguaje. Un compilador es el programa que realiza esta tarea.

Generalmente, un compilador transforma código escrito en un lenguaje de alto nivel, como C++, Rust o Java, en código ejecutable, también conocido como código binario o código máquina. WebAssembly, por ejemplo, es un tipo de código binario ejecutable que puede ser compilado a partir de lenguajes como C++, Rust, C#, Go, Swift, entre otros, y luego ejecutado en cualquier página web desde cualquier navegador (*MDN Web Docs,* 2023). Referencia: MDN Web Docs. (2023, Noviembre 13). Compilar - Glosario de MDN Web Docs: Definiciones de términos relacionados con la Web | MDN. MDN Web Docs. https://developer.mozilla.org/es/docs/Glossary/Compile

Framework: Un framework es un esquema o estructura base que proporciona un marco de trabajo para desarrollar proyectos con objetivos específicos. Sirve como una plantilla inicial que facilita la organización y el desarrollo de software, simplificando tareas y procesos para mejorar la agilidad y productividad de los Digital Workers.

Los programadores usan ampliamente estos esquemas para acelerar el trabajo, fomentar la colaboración, reducir errores y obtener resultados de calidad.

En el ámbito online, los frameworks son aplicables a diversas áreas como la optimización del proceso de compra de clientes o la mejora de productos digitales para aumentar conversiones. La automatización juega un papel crucial en la eficiencia y efectividad de estos procesos de trabajo modernos (*‌*UNIR, 2022). Referencia: UNIR. (2022, Septiembre 22). *Framework: qué es, para qué sirve y algunos ejemplos*. UNIR FP; UNIR. [**https://unirfp.unir.net/revista/ingenieria-y-tecnologia/framework/**](https://unirfp.unir.net/revista/ingenieria-y-tecnologia/framework/)

Repositorio (Repository): Un repositorio, o repo, es un tipo de almacenamiento digital centralizado que los desarrolladores utilizan para realizar y administrar cambios en el código fuente de una aplicación. Facilita el almacenamiento y la compartición de carpetas, archivos de texto y otros documentos necesarios durante el desarrollo de software. Referencia: Microsoft Docs. (2023). Introduction to Repositories. Recuperado el 10 de julio de 2024, de [**https://docs.microsoft.com/en-us/azure/devops/repos/**](https://docs.microsoft.com/en-us/azure/devops/repos/)

Ramificación (Branching): La ramificación es el proceso de creación de una copia del repositorio en GitHub. Por defecto, un repositorio de GitHub tiene una rama principal que contiene los códigos fuente originales. Los desarrolladores pueden crear una rama para modificar una parte específica del código sin afectar la rama principal. Referencia: GitHub Docs. (2023). About branches. Recuperado el 10 de julio de 2024, de [**https://docs.github.com/en/github/collaborating-with-issues-and-pull-requests/about-branches**](https://docs.github.com/en/github/collaborating-with-issues-and-pull-requests/about-branches)

Confirmación (Commit): La confirmación es una característica que permite a los desarrolladores guardar todos los cambios que han realizado en los archivos de código de una rama. Al enviar estos cambios, los desarrolladores pueden describir las modificaciones que han introducido. Referencia: Atlassian. (2023). What is a commit? Recuperado el 10 de julio de 2024, de [**https://www.atlassian.com/git/tutorials/saving-changes**](https://www.atlassian.com/git/tutorials/saving-changes)

Solicitud de Extracción (Pull Request): Una solicitud de extracción envía un mensaje oficial a otros colaboradores que trabajan en la rama principal o al propietario de esta para que revisen los cambios confirmados. Permite a los colaboradores revisar, comentar y realizar cambios adicionales antes de fusionar los cambios en la rama principal. Referencia: GitHub Docs. (2023). About pull requests. Recuperado el 10 de julio de 2024, de [**https://docs.github.com/en/github/collaborating-with-issues-and-pull-requests/about-pull-requests**](https://docs.github.com/en/github/collaborating-with-issues-and-pull-requests/about-pull-requests)

Fusión (Merge): La fusión es el proceso mediante el cual se aplican los cambios confirmados a la rama principal. Se realiza después de que los colaboradores hayan revisado y aprobado los cambios, asegurando que el código se integre de manera segura y coherente. Referencia: Bitbucket. (2023). Merge strategies. Recuperado el 10 de julio de 2024, de [**https://support.atlassian.com/bitbucket-cloud/docs/merge-strategies/**](https://support.atlassian.com/bitbucket-cloud/docs/merge-strategies/)

SCRUM: Un marco de trabajo ágil para la gestión de proyectos, desarrollado por Ken Schwaber y Jeff Sutherland en la década de 1990, que promueve la entrega continua de productos funcionales a través de iteraciones e incrementos (Schwaber & Sutherland, 2020).

Sprint: Un ciclo de trabajo corto en SCRUM, generalmente de 1 a 4 semanas, durante el cual el equipo de desarrollo trabaja en tareas definidas para completar un incremento de producto funcional (Schwaber & Sutherland, 2020).

Scrum Master: Un rol en SCRUM responsable de facilitar el proceso, eliminar impedimentos y asegurar que el equipo siga los principios y prácticas de SCRUM (Rubin, 2013).

Product Owner: Un rol en SCRUM que gestiona el backlog del producto, prioriza las tareas y maximiza el valor del producto para el cliente (Rubin, 2013).

Equipo de Desarrollo: Un grupo multifuncional y auto-organizado en SCRUM encargado de convertir las tareas seleccionadas en incrementos de producto funcionales durante cada sprint (Schwaber, 2004). Referencia: Schwaber, K. (2004). Agile Project Management with Scrum. Microsoft Press.

Backlog del Producto: Una lista priorizada de tareas y requisitos del producto gestionada por el Product Owner, que sirve como fuente principal de trabajo para el equipo de desarrollo (Rubin, 2013). Referencia: Rubin, K. S. (2013). Essential Scrum: A Practical Guide to the Most Popular Agile Process. Addison-Wesley.

Daily Stand-up: Una reunión diaria de corta duración en SCRUM donde el equipo revisa el progreso, coordina esfuerzos y resuelve problemas (Schwaber & Sutherland, 2020). Referencia: Schwaber, K., & Sutherland, J. (2020). The Scrum Guide. Scrum.org.

Sprint Planning: La fase inicial de un sprint donde el equipo de desarrollo y el Product Owner se reúnen para definir y planificar el trabajo a realizar durante el sprint (Schwaber & Sutherland, 2020). Referencia: Schwaber, K., & Sutherland, J. (2020). The Scrum Guide. Scrum.org.

Sprint Review: Una reunión al final de un sprint donde el equipo de desarrollo presenta el trabajo completado y recibe retroalimentación del Product Owner y otros stakeholders (Schwaber & Sutherland, 2020). Referencia: Schwaber, K., & Sutherland, J. (2020). The Scrum Guide. Scrum.org.

Sprint Retrospective: Una reunión al final de un sprint donde el equipo de desarrollo reflexiona sobre el proceso, identifica mejoras y planifica acciones para mejorar en futuros sprints (Schwaber & Sutherland, 2020). Referencia: Schwaber, K., & Sutherland, J. (2020). The Scrum Guide. Scrum.org.