|  |  |  |  |
| --- | --- | --- | --- |
| TIPO DE APLICACIÓN | DEFINICIÓN | CARACTERÍSTICAS | ETAPAS |
| APLICACIÓN WEB | **Sitio web que tras ser tratado por un programador se ha podido adaptar para que los usuarios puedan acceder a través de un servidor web** utilizando Internet mediante un navegador, ya sea Chrome, Mozilla, Microsoft Edge, etc. | * Usar un único diseño, normalmente en HTML5, para todos los dispositivos. * El usuario no tiene que descargarlas. Se alojan en un servidor y se accede a ellas desde el navegador. * Son compatibles con cualquier navegador. * No aparecen en los buscadores de aplicaciones, como Google Play Store, pero sí en los buscadores de Internet. | * Fase 1 – Análisis. * Fase 2 – Planificación. * Fase 3 – Diseño. * Fase 4 – Contenidos. * Fase 5 – Desarrollo. * Fase 6 – Pruebas. * Fase 7 – Lanzamiento. |
| APLICACIÓN  MÓVIL | Tipo de aplicación diseñada para ejecutarse en un dispositivo móvil, que puede ser un teléfono inteligente o una tableta. Incluso si las aplicaciones suelen ser pequeñas unidades de software con funciones limitadas, se las arreglan para proporcionar a los usuarios servicios y experiencias de calidad. | * Las apps están destinadas sobre todo a smartphones o tablets. * Son más dinámicas que los programas. * Algunas dependen de internet, aunque no es una característica indispensable. * Ocupan menos espacio que los programas y son más rápidas de instalar * Cualquiera con ciertos conocimientos informáticos puede crear una app para android o iOS. | * Elaboración del presupuesto * La idea * Elegir al proveedor * Análisis de los requerimientos * La planificación * UX y diseño de la aplicación * Desarrollo app * Testing * Lanzamiento * Mantenimiento |
| APLICACIÓN DE ESCRITORIO | Una aplicación de escritorio es aquella que se encuentra instalado en el ordenador o sistema de almacenamiento y podemos ejecutarlo sin internet en nuestro sistema operativo, al contrario que las aplicaciones en la nube que se encuentran en otro ordenador al que accedemos a través de la red o internet a su software. | * Pueden ser más robustas * Tiempo de respuesta más rápido * Se puede hacer cualquier cosa que permita el Software * Facilita el uso de teclas en caliente * Requiere instalación en cada cliente * Generalmente se hacen para un Sistema Operativo especifico * Se requiere actualizar en cada cliente | * Conceptualización. * Definición. * Diseño. * Desarrollo. * Publicación. |
| CLIENTE - SERVIDOR | **Modelo de comunicación** que vincula a varios dispositivos informáticos a través de una **red**. El cliente, en este marco, **realiza peticiones de servicios** al servidor, que se encarga de satisfacer dichos requerimientos. | * El Cliente y el Servidor pueden actuar como una sola entidad y también pueden actuar como entidades separadas, realizando actividades o tareas independientes. * Las funciones de Cliente y Servidor pueden estar en plataformas separadas, o en la misma plataforma. * Cada plataforma puede ser escalable independientemente. Los cambios realizados en las plataformas de los Clientes o de los Servidores, ya sean por actualización o por reemplazo tecnológico, se realizan de una manera transparente para el usuario final. * La interrelación entre el hardware y el software está basada en una infraestructura poderosa, de tal forma que el acceso a los recursos de la red no muestra la complejidad de los diferentes tipos de formatos de datos y de los protocolos. * Su representación típica es un centro de trabajo (PC), en donde el usuario dispone de sus propias aplicaciones de oficina y sus propias bases de datos, sin dependencia directa del sistema central de información de la organización. | • Aceptar los requerimientos de bases de datos que hacen los clientes.  • Procesar requerimientos de bases de datos.  • Formatear datos para trasmitirlos a los clientes.  • Procesar la lógica de la aplicación y realizar validaciones a nivel de bases de datos |

|  |  |  |  |
| --- | --- | --- | --- |
| TIPO DE APLICACIÓN | CLASIFICACIÓN | METODOLOGÍA DE DESARROLLO DE SOFTWARE RECOMENDADA | EJEMPLO |
| APLICACIÓN WEB | * Sitios estáticos y dinámicos. (D. Foros, encuestas, pedidos online). * Sitios institucionales o corporativos. * Blogs. * Tiendas online o e-commerce. * Buscadores. * Comparadores. | * Análisis de requerimientos. * Arquitectura y tecnología. * Diseño de la estructura lógica y física del sitio. * Creación de contenidos. * Diseño gráfico. * Creación de las páginas estáticas. * Creación de las páginas dinámicas. * Verificación del funcionamiento del sitio | *Gmail, Google Drive, Google Docs, etc.* |
| APLICACIÓN  MÓVIL | **Aplicaciones móviles nativas:**  Las aplicaciones nativas (Native App) son aquellas en las que se lleva a cabo específicamente un desarrollo para cada una de las plataformas. En el caso de los móviles: Android, iOS…  Si realizáramos dos versiones de nuestra aplicación nativa:  **Aplicaciones móviles híbridas o multiplataforma**  Las aplicaciones híbridas (Hibrid App) o Multiplataformas (Cross Platform) son aquellas que nos permiten desarrollar para varias plataformas a la vez con tan solo realizar un desarrollo.  Si realizáramos dos versiones de nuestra aplicación híbrida:  **Aplicaciones móviles web**  Las aplicaciones web (Web App) son aquellas en las que se realiza un único desarrollo para todas las plataformas y podemos acceder desde un navegador web a dichas aplicaciones. | **Modelo Watefall**  Es uno de los modelos de metodologías para el desarrollo de aplicaciones móviles clásico. Esta aplicación sólo es adaptable cuando están totalmente cerrados los requisitos y no van a cambiar. No hay retroalimentación entre las fases en que se divide el proyecto. Por lo que cada fase se va cerrando de forma secuencial. Todo el proceso está fijado por fechas límites y presupuestos. Este modelo sólo es aconsejable para proyectos móviles muy controlados y previsibles, no existe incertidumbre por lo que se quiere hacer ni influyen los cambios en la industria. **Desarrollo rápido de aplicaciones** Esta metodología da énfasis en la obtención de un prototipo funcional de una aplicación para posteriormente ir mejorándolo. Después va incluyendo más funcionalidades y complejidad. Es recomendable el uso de patrones de diseño bien conocidos para adaptarse a los cambios de requisitos. Dentro de las metodologías para el desarrollo de aplicaciones móviles se suele usar cuando los plazos de entrega son muy cortos. Se precisa tener un entregable de forma inmediata. No se descarta utilizar otras metodologías de forma posterior, ya que este tipo de desarrollo puede ser usado para mostrar un esbozo de la aplicación a un cliente, generalmente en un par de días.  **Mobile-D**  El objetivo de esta metodología es conseguir ciclos de desarrollo muy rápidos en equipos muy pequeños. Se basa en metodologías para el desarrollo de aplicaciones móviles conocidas pero aplicadas de forma estricta como: extreme programming, Crystal Methodologies y Rational Unified Process. Tiene distintas fases: exploración, inicialización, fase de producto, fase de estabilización y la fase de pruebas. Cada una tiene un día de planificación y otro de entrega.  1.Fase **exploración:** se centra la atención en la planificación y en los conceptos básicos del proyecto. Aquí es donde se define el alcance del proyecto y su establecimiento con las funcionalidades donde se quiere llegar.  2.Fase de **iniciación:**configuramos el proyecto identificando y preparando todos los recursos necesarios como hemos comentado anteriormente en esta fase la dedicaremos un día a la planificación y el resto al trabajo y publicación.  3.Fase de **producto:**se repiten iterativamente las subfases. Se usa el desarrollo dirigido por pruebas (TDD), antes de iniciar el desarrollo de una funcionalidad debe existir una prueba que verifique su funcionamiento. En esta fase podemos decir que se lleva a acabo toda la implementación.  4.Fase de **estabilización:**se realizan las acciones de integración para enganchar los posibles módulos separados en una única aplicación.  5.Fase de **pruebas**: una vez parado totalmente el desarrollo se pasa una fase de testeo hasta llegar a una versión estable según lo establecido en las primeras fases por el cliente. Si es necesario se reparan los errores, pero no se desarrolla nada nuevo. | *Instagram* |
| APLICACIÓN DE ESCRITORIO | * Aplicaciones de gestión. * Programas de facturación. * Aplicaciones para la venta online. * Software para tratamiento de audio y video. * Software de grabación y gestión de conferencias. * Aplicaciones de gestión documental. * Control de procesos productivos. | **Waterfall (cascada):** es una metodología en la que las etapas se organizan de arriba a abajo, de ahí el nombre. Se desarrollan las diferentes funciones en etapas diferenciadas y obedeciendo un riguroso orden. Antes de cada etapa se debe revisar el producto para ver si está listo para pasar a la siguiente fase. Los requisitos y especificaciones iniciales no están predispuestos para cambiarse, por lo que no se pueden ver los resultados hasta que el proyecto ya esté bastante avanzado.  **Prototipado:** se basa en la construcción de un prototipo de software que se construye rápidamente para que los usuarios puedan probarlo y aportar feedback. Así, se puede arreglar lo que está mal e incluir otros requerimientos que puedan surgir. Es un modelo iterativo que se basa en el método de prueba y error para comprender las especificidades del producto.  **Espiral:** es una combinación de los dos modelos anteriores, que añade el concepto de análisis de riesgo. Se divide en cuatro etapas: planificación, análisis de riesgo, desarrollo de prototipo y evaluación del cliente. El nombre de esta metodología da nombre a su funcionamiento, ya que se van procesando las etapas en forma de espiral. Cuanto más cerca del centro se está, más avanzado está el proyecto.  **Incremental:** en esta metodología de desarrollo de software se va construyendo el producto final de manera progresiva. En cada etapa incremental se agrega una nueva funcionalidad, lo que permite ver resultados de una forma más rápida en comparación con el modelo en cascada. El software se puede empezar a utilizar incluso antes de que se complete totalmente y, en general, es mucho más flexible que las demás metodologías.  **Diseño rápido de aplicaciones (RAD):** esta metodología permite desarrollar software de alta calidad en un corto periodo de tiempo. Los costes son mucho más altos y el desarrollo más flexible, aunque requiere una mayor intervención de los usuarios. Por otro lado, el código puede contener más errores, y sus funciones son limitadas debido al poco tiempo del que se dispone para desarrollarlas. El objetivo es iterar el menor número posible de veces para conseguir una aplicación completa de forma rápida. | Microsoft Office, Visual Studio Code |
| CLIENTE - SERVIDOR | * **Cliente Activo, Servidor Pasivo:** El cliente realiza la practica totalidad del trabajo de procesado de la información. Ejemplo: Google Earth. * **Cliente Pasivo, Servidor Pasivo:** Tanto el cliente como el Servidor simplemente pasan información. Ejemplo: Gateways de comunicaciones VoIP. * **Cliente Pasivo, Servidor Activo:** El Servidor realiza todo el trabajo de procesado y el cliente simplemente presenta los datos. Ejemplo: Servidores de terminales. * **Cliente Activo, Servidor Activo:** Tanto el Servidor como el Cliente procesan la información. Ejemplo: Servicios de Correo Electrónico. | * Planificación y Análisis de Sistemas * Costos * DISEÑO * Desarrollo * Aplicaciones de Prueba * Entrega del sistema | Uso del servicio de correo electrónico de un ISP para enviar, recibir y almacenar correo electrónico. El cliente de correo electrónico en una PC doméstica emite una solicitud al servidor de correo electrónico del ISP para que se le envíe todo correo no leído. |