**FORMATO PARA EL DESARROLLO DE COMPONENTE FORMATIVO**

|  |  |
| --- | --- |
| PROGRAMA DE FORMACIÓN | ADSO |

|  |  |  |  |
| --- | --- | --- | --- |
| COMPETENCIA | 220501095. Diseñar la solución de *software* de acuerdo con procedimientos y requisitos técnicos. | RESULTADOS DE APRENDIZAJE | 220501095-03. Determinar las características técnicas de la interfaz gráfica del  *software* adoptando estándares. |

|  |  |
| --- | --- |
| NÚMERO DEL COMPONENTE FORMATIVO | 012 |
| NOMBRE DEL COMPONENTE FORMATIVO | Diseño y construcción de interfaces web |
| BREVE DESCRIPCIÓN | Este componente formativo ofrece herramientas para la construcción de interfaces gráficas, especialmente en entornos web. Se presentarán conceptos y una guía básica para desarrollar un proyecto desde cero utilizando una estrategia de caso de estudio. |
| PALABRAS CLAVE | Vistas, maquetación, entornos de desarrollo, lenguajes, *software,* interfaz. |

|  |  |
| --- | --- |
| ÁREA OCUPACIONAL | 2 - CIENCIAS NATURALES, APLICADAS Y RELACIONADAS |
| IDIOMA | Español |

1. **TABLA DE CONTENIDOS:**
2. Sistema de navegación en plataformas *software*
3. Sistema de interacción en plataformas de *software*
4. Estructura de la interfaz de la aplicación
5. Construcción de la estructura del *front-end*
6. Análisis de requerimientos
7. Diseño gráfico: herramientas de prototipado
8. Principios básicos de usabilidad
9. Componentes de *software*
   1. Seguridad
   2. *API Firebase*
   3. Herramientas
10. **INTRODUCCIÓN**

Este componente formativo abordará los conceptos y técnicas que respaldan el desarrollo de la interfaz gráfica de un producto de *software.* Se iniciará con una introducción teórica sobre el sistema de navegación e interacción de las aplicaciones. A partir de esta base, se revisará el proceso de construcción de la interfaz gráfica mediante un caso práctico, comenzando con un boceto y luego avanzando a la maquetación con herramientas de prototipado rápido y tecnologías *front-end.*

|  |  |
| --- | --- |
| Ilustración del concepto de interfaz | Además, se destacará el uso del *framework CSS Materialize*, el cual proporciona una interfaz dinámica, usable y de fácil aprendizaje. Se brindarán herramientas y recursos para facilitar el desarrollo de futuros proyectos de *software.* |



Abordar un objetivo sin una planificación adecuada puede resultar en un análisis menos efectivo, generando así una discrepancia entre el producto entregado y las expectativas del cliente o los usuarios finales. Esto es particularmente crítico en el desarrollo de aplicaciones móviles, donde la interacción con el usuario ofrece desafíos únicos. Una planificación inadecuada puede resultar en frustración para el usuario o el cliente, y en el mejor de los casos, en un exceso de esfuerzo o incumplimiento de los plazos establecidos.

Establecer los requisitos del sistema de manera precisa es esencial para diseñar adecuadamente la interacción con el usuario, optimizando así el tiempo y maximizando la inversión. Por ello, se recomienda realizar dos actividades fundamentales antes de empezar a codificar:

1. **DESARROLLO DE CONTENIDOS:**

1. **Sistema de navegación en plataformas *software***

Un sistema de navegación permite que los usuarios se desplacen libremente por las distintas secciones de una plataforma. No se trata simplemente de un aspecto menor de la navegación.

Considérese dos situaciones hipotéticas:

|  |  |
| --- | --- |
|  | Un aficionado a la lectura entra en una biblioteca buscando una edición antigua del Quijote. |
| Diseño de pirámides y viajes mundiales | Un joven empresario visita una web de viajes en busca de un vuelo de ida y vuelta a Perú. |

Estos escenarios presentan cómo diferentes personas acceden a la información según sus necesidades específicas. Por lo tanto, antes de diseñar cualquier sistema de navegación, ya sea web o móvil, es crucial entender cómo buscan información los usuarios.

A continuación, se explorará cómo se realiza la búsqueda de información y la navegación en sitios web y dispositivos móviles:

|  |
| --- |
| Slide  CF012\_1\_Sistema de navegación en plataformas *software* |

Es necesario implementar al menos una técnica de prototipado rápido para facilitar la navegación entre las vistas del *software* y mejorar la interacción. Existe una correlación evidente entre la navegación a través de las diversas vistas de un sitio o aplicación y las acciones que debe realizar el usuario, conocido comúnmente como sistemas de interacción.

1. **Sistema de interacción en plataformas de *software***

Este sistema se define como un conjunto de componentes que responden a las necesidades de los usuarios al navegar por una plataforma de *software.* En otras palabras, comprende el diseño de cada acción que un usuario puede realizar en un sitio web o aplicación, tales como hacer clic, *scrolling* o utilizar gestos (como ampliar o disminuir el tamaño). Cada acción desencadena una respuesta específica en la aplicación desarrollada, a lo que se le denomina sistema de interacción.



Adicionalmente, en las páginas de interacción, se permite a los usuarios escribir, dibujar, editar, borrar y ejecutar todo tipo de acciones. Estas pueden ser muy simples, como la página de inicio de Google, o muy complejas, integrando múltiples elementos de interacción.

En la actualidad, muchos de los sitios más visitados presentan características adaptadas a los cambios tecnológicos recientes en la movilidad de los dispositivos. Según el artículo de la Digital Guide de IONOS (2021):

|  |  |
| --- | --- |
| Página de inicio vista en la pantalla de la computadora | Las páginas web deben ser hoy en día más eficientes e intuitivas para poder subsistir en un mercado *online* extremadamente competitivo, ya sea desde un PC de escritorio, una *tablet* o un teléfono móvil. El diseño recargado de la década de los años 2000 ha pasado a la historia: ahora la prioridad es concentrarse en lo esencial. |

**Tendencias actuales en diseño web**

Esta tendencia se ha consolidado en la realidad actual, donde se observa que la gran mayoría de las consultas y actividades se realizan a través de Internet.

En este contexto, es crucial que las páginas web sean fáciles de usar, ya que la usabilidad y accesibilidad se convierten en elementos decisivos dentro del marco del diseño responsivo.

|  |  |
| --- | --- |
| concepto de medios, Internet, negocios y tecnología: cierre de la página web del blog en la pantalla de la computadora portátil, tableta, computadora portátil y anteojos en la mesa de madera | El diseño web moderno permite que una página o sitio web **"responda o reaccione"** adecuadamente al dispositivo utilizado por la persona, asegurando una presentación correcta del contenido. Es importante entender que no se trata solo de adaptar el contenido, sino de modificar la interfaz en general para mejorar la usabilidad y, por ende, la experiencia del usuario. |

Con esta perspectiva, y abordando de manera sucinta estos conceptos, se propone como caso de estudio la estructura y maquetación de la interfaz de un servicio de publicación de posts. Para ello, el primer recurso que se utilizará será una herramienta de prototipado rápido llamada Balsamiq Mockups, la cual se detallará en el siguiente apartado.

1. **Estructura de la interfaz de la aplicación**

**Caso de estudio sobre el servicio de publicación de *posts***

|  |  |
| --- | --- |
| Plantillas de marca | Cuando se inicia un nuevo proyecto de *software,* es habitual realizar *sketches,* bocetos o dibujos rápidos para obtener una visión preliminar de cómo será la interfaz gráfica del sistema. Este proceso es muy beneficioso para los equipos de desarrolladores y diseñadores, ya que les permite plasmar una visión concreta de lo que será el proyecto. Todo comienza con un boceto en papel. Durante esta etapa, se hacen cambios y correcciones, se descartan ideas y se generan otras mejores. Estos *sketches* se transforman gradualmente en el prototipo final del proyecto, finalmente evolucionando en *Mockups*. A este proceso se le conoce como *Wireframing,* que consiste en crear la estructura de una interfaz gráfica. |

**Instalación de Balsamiq**

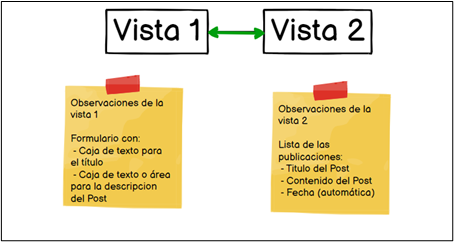
Una de las herramientas más efectivas para crear prototipos, bocetos o *wireframes* es Balsamiq *Mockups.* Para comenzar a utilizar Balsamiq, se optará por el modo de servicio en la nube (versión de prueba), disponible en: <https://balsamiq.cloud/>, la instalación es muy sencilla: basta con registrarse y hacer clic en "nuevo proyecto", como se presenta a continuación:

|  |
| --- |
| Slides  CF012\_3\_ Estructura de la interfaz de la aplicación |

**Diseño de la estructura de la interfaz**

Esta herramienta permite generar de manera estructural las vistas de la aplicación. Es decir, se describe en palabras cuántas vistas son necesarias y cuántos objetos de interacción se requieren. En decir, proporciona una propuesta de la estructura de la interfaz de la aplicación. Esta actividad se suele realizar tradicionalmente con lápiz y papel. Sin embargo, con Balsamiq, esta tarea se puede llevar a cabo sin dificultad, como se presenta a continuación:

**Figura 1.** Diseño de la estructura de la interfaz

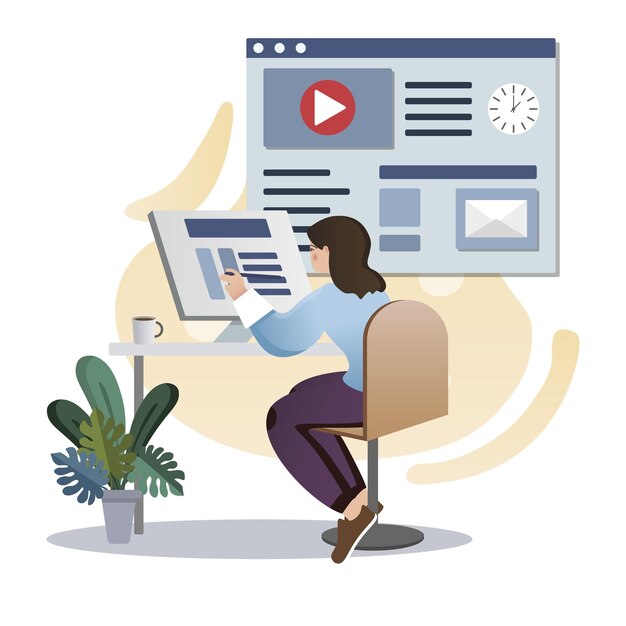


**Creación de las vistas del sistema**

Para este componente formativo, el caso de estudio es un servicio básico de publicación de *posts,* que consta de dos vistas en la aplicación. Se ha diseñado la aplicación web de forma que sea *responsive*, tomando como plataforma un dispositivo móvil (tableta). La barra de herramientas de Balsamiq proporciona todo lo necesario para esta primera tarea. En la sección de material de apoyo, se adjunta un video que destaca las ventajas de esta herramienta. A continuación, se presentan las vistas de la aplicación:

|  |  |
| --- | --- |
| Figura 2. Creación de las vistas del sistema | Para este caso, se crean dos vistas. La primera es un formulario que consta de tres (3) objetos de interacción. El primero es una caja de texto donde se introduce el título, el segundo es un área de texto para describir el contenido del post, y por último, un botón que envía estos datos a una segunda vista. Desde Balsamiq, se puede generar esta interacción para facilitar la navegación entre vistas, revisando las opciones del objeto *'button'*.  Por otro lado, la segunda vista muestra de manera ordenada la lista de *posts* publicados por el usuario, con el formato detallado en la ilustración adjunta. Es importante resaltar que la fecha debe ser tomada automáticamente por el sistema al momento de la publicación. |

1. **Construcción de la estructura del *front-end***



Para el desarrollo de la estructura del *front-end*, que en el ámbito de este componente formativo se limita exclusivamente a las tecnologías de HTML5 y CSS3, se utilizará el *framework* de maquetación llamado Materialize. Este *framework* CSS permite crear sitios y aplicaciones web siguiendo los principios de *Material Design.* La versión estándar del *framework* incluye CSS y JavaScript, con los estilos CSS listos para usar.

Además, Materialize incorpora buenas prácticas para el desarrollo del *front-end,* lo que contribuye a que la aplicación sea más eficiente y rápida en su despliegue. Por otra parte, es un *framework* *responsive,* lo que significa que se adapta a cualquier tamaño de dispositivo.

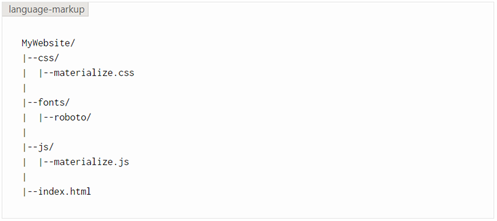
A continuación, se detallan los pasos para la instalación y manejo del *framework* (Materialize, 2021):

1. **Instalación de Materialize:**

Existen varias formas de instalar el framework. En este caso, se utilizará la descarga de los archivos, disponible en el enlace proporcionado en la sección de material de apoyo con el nombre **"Descarga de archivos Materialize".**

Una vez descargada la carpeta, se encontrará una estructura de directorios muy similar a la guía que presenta Materialize en su página web.

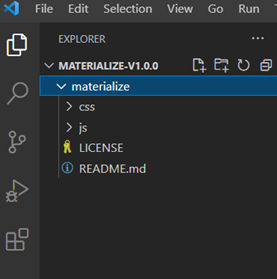
Figura 3. Estructura de directorios de Materialize



1. **Proyecto de Materialize en Visual Studio Code**

Ahora, lo siguiente que debe hacerse es incorporar o referenciar esa carpeta desde un entorno de desarrollo integrado. Para este ejemplo, se utilizará Visual Studio Code, como se presenta a continuación:

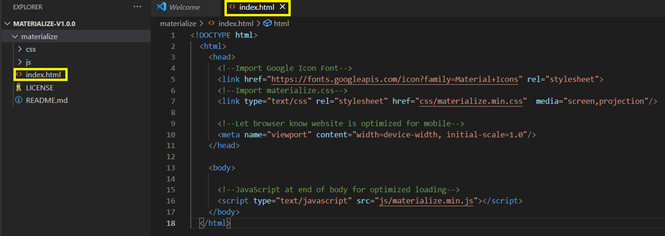
Figura 4. Proyecto de Materialize en Visual Studio Code



1. **Configuración del proyecto**

Siguiendo la guía proporcionada por el *framework*, que puede consultarse en el enlace indicado en la sección de material de apoyo con el nombre "Guía rápida de Materialize", se crea un archivo llamado index.html en la raíz de la estructura de directorios. Este archivo contendrá la siguiente información, incluyendo las referencias a los archivos de estilos y JavaScript que proporcionan el funcionamiento y la apariencia a la interfaz desarrollada:

**Figura 5.** Archivo index.html de la guía de Materialize



1. **Desarrollo del proyecto (primera vista)**

Hasta el momento, tanto el *framework* como el entorno para crear las vistas del caso de estudio están preparados en este componente formativo. Por lo tanto, es necesario referirse nuevamente a la primera vista diseñada para revisar los componentes necesarios para el desarrollo.

En la primera vista, se requiere lo siguiente:

* Título.
* Formulario que consta de dos cajas de texto y un botón.

|  |
| --- |
| * <div class="section no-pad-bot" id="index-banner"> * <div class="container"> * <br><br> * <h1 class="header center orange-text">Publica tu Post</h1> * <div class="row center"> * <h5 class="header col s12 light">Un ejemplo de maquetación utilizando el framework de Materialize</h5> * </div> * <br><br> * </div> * </div> |

Para esta maquetación, es necesario dimensionar el *grid* o rejilla, lo cual ayuda a mantener el contenido de manera organizada y estructurada. Para lograrlo, se utiliza la clase *"container"*, que centra el contenido de la página y ocupa aproximadamente el 70% del ancho de la ventana, lo que ayuda a centrar y contener el contenido de la página.

Además, en el código se muestra el uso de colores. Por ejemplo, el título de la página utiliza el atributo *"orange-text",* que proporciona el color naranja al texto "Publica tu Post", como se presenta a continuación:

Figura 6. Publica tu post



Ahora continúa el desarrollo del formulario, el cual consta de dos (2) *inputs*: uno destinado al título del *post* y otro al área de texto, tal como se presenta a continuación:

|  |
| --- |
| <div class="row">  <div class="container">  <form class="col s12">  <div class="row">  <div class="input-field col offset-s4 s4">  <input placeholder="Título del Post" id="tituloPost" type="text" class="validate">  <label for="tituloPost">Título del Post</label>  </div>  </div>  <div class="row">  <div class="input-field col offset-s4 s4">  <textarea placeholder="Título del Post" id="textarea1" class="materialize-textarea"></textarea>  <label for="textarea1">Contenido del Post</label>  </div>  </div>  <button class="btn waves-effect waves-light col offset-s4 s4" type="submit" name="action">Enviar  <i class="material-icons right">send</i>  </button>  </form>  </div>  </div> |

**E) Resultado de la primera vista**

Como se puede observar en el código, se sigue utilizando la estructura tradicional de un formulario en HTML. Lo que varía son los atributos de las clases que se utilizan para centrar los *inputs* o mejorar la apariencia de cada objeto de interacción. Por ejemplo, el atributo "col offset-s4 s4" se emplea en cada una de las etiquetas, lo que permite que cada componente del formulario se encuentre centrado.

Figura 7. Resultado de la primera vista



1. **Desarrollo de la segunda vista**

Para esta parte se utiliza el concepto de listas en HTML, como se presenta en el siguiente código de HTML:

|  |
| --- |
| <div class="container">  <ul class="collection">  <li class="collection-item avatar">  <i class="material-icons circle">face</i>  <span class="title">Título del Post</span>  <p>Este es un contenido para el primer Post de prueba <br>  <i>Fecha de publicación: 1 junio de 2021</i>  </p>  <a href="#" class="secondary-content"><i class="material-icons">grade</i></a>  </li>  </ul>  </div> |
|  |

1. **Resultado de la segunda vista**

En el código HTML anterior, se muestra la estructura estándar de una lista desordenada, con los únicos cambios relacionados con el uso de los atributos propios del *framework.* Por lo tanto, el resultado esperado se presenta en la siguiente figura:

Figura 8. Resultado de la segunda vista



1. **Análisis de requerimientos**

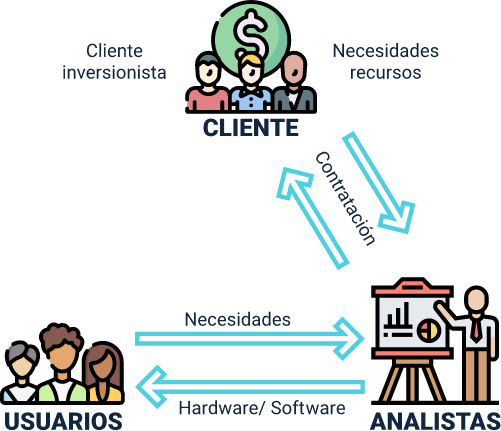
El desarrollo de aplicaciones móviles va más allá de la mera codificación o programación; también implica analizar y comprender las necesidades del cliente, crear modelos que faciliten el proceso de desarrollo, planificar el proyecto y otras actividades relacionadas.



La aplicación de técnicas para elicitar requisitos permite identificar las fuentes de estos con los interesados en el sistema (*stakeholders*). Sin embargo, estas actividades pueden llevar a que el cliente o los usuarios excedan lo que realmente necesitan o que no consideren las limitaciones a nivel de *hardware, software*, presupuesto, tiempo o requisitos. Por lo tanto, es fundamental conocer con detalle lo que se pretende alcanzar y las necesidades reales a cubrir. Así, se deben detallar y modificar los requisitos básicos elicitados, lo cual se logra mediante el consenso de las partes involucradas.

Se deben responder las siguientes preguntas: ¿Qué es lo prioritario?, ¿qué es esencial? y ¿para cuándo se requiere? Esto se especifica de alguna manera y luego se revisa o válida para garantizar que la comprensión del problema por parte del analista coincida con la de los demás participantes.

**Figura 9.** Esquema de acuerdos



Cuando los sistemas no son tan complejos en su enlace, como suele ser el caso para aplicaciones móviles, se recomienda el empleo de historias de usuario. Sin embargo, se sugiere revisar algunos aspectos que no son fáciles de identificar en un proceso de licitación de requisitos y que se mencionan a continuación, ya que están enmarcados en el contexto de aplicaciones móviles.

|  |
| --- |
| Slide  CF012\_5\_Análisis de requerimientos (1) |

Cuando un requisito o un conjunto de ellos forma parte de todo un ecosistema tecnológico en el que intervienen distintos tipos de aplicaciones (web, de escritorio, móviles, interacciones con otros sistemas de proveedores, clientes o socios), como ocurre, por ejemplo, con los sistemas bancarios que disponen de aplicaciones móviles, aplicaciones web, terminales de escritorio y portales para asociados, las historias de usuario no siempre ofrecen una visión completa de la complejidad del problema.

Estos sistemas generalmente cuentan con lo que se denomina **"Reglas de Negocio"** o un "**Conjunto de Reglas de Negocio"** que describen las políticas, operaciones, normativas, definiciones y restricciones que rigen en una organización. Dichas reglas son fundamentales para los sistemas de información o aplicaciones, ya que consideran los factores críticos para alcanzar los objetivos misionales del cliente. En algunos casos, incluso se requiere de un glosario de términos específicos del negocio para poder especificar dichas reglas de forma adecuada.

Las reglas de negocio pueden llegar a ser tan complejas como las operaciones del cliente. En situaciones de esta índole, se sugiere el uso de artefactos como los casos de uso para definir con detalle la especificación del sistema. Para guiar el proceso, se proponen algunos artefactos a elaborar:

|  |
| --- |
| Tarjetas  CF012\_5\_Análisis de requerimientos (2) |

Como se puede identificar, los requerimientos de una aplicación móvil suelen estar cubiertos por técnicas como la elaboración de historias de usuario. No obstante, cuando la aplicación móvil forma parte de un ecosistema tecnológico más complejo, el proceso requiere de algún artefacto que facilite la identificación y validación de los requisitos.

1. **Diseño gráfico: herramientas de prototipado**

Un diseñador *front-end* necesita contar con herramientas esenciales para realizar su labor, y el primer paso suele ser el desarrollo de una propuesta de interfaz gráfica o un prototipo de interfaz de usuario (UI) que permita al cliente visualizar cómo se verá el sistema una vez finalizado. Hoy en día, no basta con diseñar únicamente la versión para ordenadores de escritorio o el prototipo para móviles. Es necesario crear tres versiones distintas para escritorio, *tablet* y móvil, asegurando que el producto digital mantenga o adapte ciertas propiedades visuales de acuerdo al dispositivo en el que se utiliza. A continuación, se presentan algunas herramientas recomendadas para dicho propósito.

**Adobe XD**



Adobe XD es la primera herramienta creada por la compañía Adobe, que fue específicamente diseñada para prototipar la experiencia de usuario (UX) e interfaz de usuario (UI). Esta herramienta ha sido el resultado de la empresa Adobe al trabajar con diseñadores de UX, usan sus herramientas de diseño gráfico como Ilustrator, Photoshop y otros, de tal manera que un solo producto agrupa características que tiene como objetivo hacer que los diseñadores de interfaces sean más productivos.

|  |
| --- |
| SLIDES  CF012\_6\_Diseño gráfico: herramientas de prototipado (1) |

**Canva**



Canva es una plataforma de diseño gráfico simplificada, establecida en 2012. Ofrece una interfaz intuitiva de arrastrar y soltar, y brinda acceso a una amplia biblioteca con más de 60 millones de fotografías y 5 millones de vectores, gráficos y fuentes. Para utilizar este servicio, es necesario registrarse en el sitio web. Al iniciar sesión, el usuario puede comenzar a crear diseños, como presentaciones para dispositivos móviles, utilizando el botón de crear diseño.

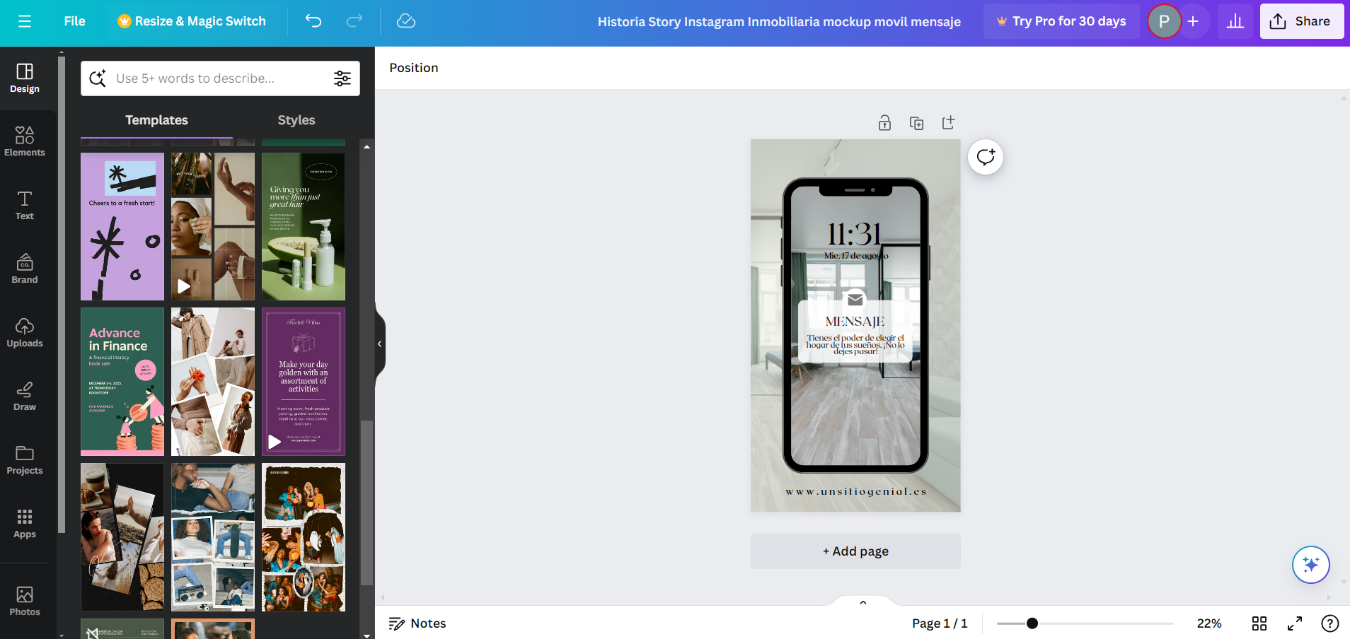
**Figura 10.** Crear presentación móvil CANVA

A screenshot of a computer

Description automatically generated

Canva, al igual que Adobe XD, permite diseñar interfaces para dispositivos móviles y ofrece un proceso similar. La plataforma sugiere comenzar con plantillas prediseñadas, lo que facilita el inicio de un proyecto.

**Figura 11.** Uso de plantilla



Finalmente, cabe destacar que estas no son las únicas herramientas disponibles en el mercado. Se han mencionado las anteriores debido a su popularidad, aceptación en el mercado, y la amplia disponibilidad de documentación y videotutoriales en diversas plataformas en línea.

**Conceptos de Material Design**

Material Design es un estilo de diseño desarrollado por Google, anunciado el 25 de junio de 2014 durante la conferencia Google I/O. Este enfoque de diseño se ha implementado en el sistema operativo Android a partir de la versión *Lollipop,* así como en otros servicios gratuitos de Google, incluyendo Docs, Calendar y Drive, entre otros.

|  |  |
| --- | --- |
| Material design logo - Social media & Logos Icons | **Material Design**  Google ha dispuesto al público una guía sobre el Material Design, para que los diseñadores se animen a implementarlo en aplicaciones móviles y sitios web.  <https://m3.material.io/> |

Los objetivos de esta propuesta son:

a) Emplear un lenguaje visual que combine los principios de un buen diseño con las opciones disponibles en las nuevas tecnologías.

b) Ser un sistema que facilite una experiencia uniforme en diversas plataformas y dispositivos, como pantallas táctiles, control de voz, entre otros.

Google junto con los objetivos de Material Design, presenta también sus tres principios fundamentales:

|  |
| --- |
| Acordeón  CF012\_6\_Diseño gráfico: herramientas de prototipado (2) |

En conclusión, Material Design es una propuesta de diseño novedosa y que se está incorporando en las nuevas versiones de Android. Los principios del Material Design van guiados a mejorar la experiencia de usuario teniendo en cuenta las últimas tecnologías.

1. **Principios básicos de usabilidad**

La usabilidad es un término que evalúa la calidad de la experiencia de usuario al interactuar con un producto o sistema. Se mide a través del análisis de la relación entre las herramientas diseñadas y los usuarios que las emplean, con el objetivo de determinar la eficiencia y efectividad en el uso de los elementos ofrecidos en las interfaces y en el cumplimiento de las tareas que se realizan mediante ellas. Por tanto, la usabilidad puede definirse como la disciplina que estudia el diseño de aplicaciones móviles o sitios web para facilitar una interacción fácil, fluida e intuitiva por parte de los usuarios.

Existen varios aspectos relacionados con el uso de aplicaciones y cómo las personas interactúan con ellas. Estos estudios han derivado en un conjunto de criterios para evaluar dichos aspectos, funcionando como un mecanismo de validación. Esta validación se realiza considerando diversos factores, entre los que se incluyen los siguientes:

|  |
| --- |
| Pestañas  CF012\_7\_Principios básicos de usabilidad (1) |

Los anteriores aspectos aplican en principio a cualquier tipo de aplicación, web o móvil, sin embargo, el compendio de buenas prácticas para aplicaciones móviles se destaca por aspectos como los relacionados a continuación:

|  |
| --- |
| Pasos  CF012\_7\_Principios básicos de usabilidad (2) |

1. **Componentes de *software***

Los sistemas de información se componen de diversos elementos que, operando conjuntamente bajo premisas definidas, logran el objetivo de mantener en operación dichos sistemas. A continuación, se detallan algunos de estos elementos esenciales para garantizar la continuidad del funcionamiento del sistema.

**Licenciamiento**

Una licencia de *software* es un contrato que establece los derechos legales del uso autorizado de un sistema de información, programa o aplicación. En otras palabras, es un acuerdo mediante el cual el fabricante o proveedor del *software* concede a una persona u organización el permiso para usar un producto específico.

|  |  |
| --- | --- |
| Concepto de certificación en línea | Una aceptación inequívoca de este contrato se realiza cuando se descarga e instala el *software* a través del medio oficial de distribución del producto. Es decir, aunque no se firme un documento de manera explícita, las acciones descritas en el contrato (generalmente: descarga, instalación, configuración) que implican la aceptación de las condiciones, son consideradas legalmente como equivalentes a la firma de un contrato de licencia. |

**Tipos de licencia de *software***

En Colombia, la normativa que regula las licencias de *software* está contenida en la Ley 603 de 2000, conocida como la ley de licencias de *software* o Ley para el cumplimiento de las licencias de *software.* Las licencias se clasifican según el uso final y las libertades que el autor concede sobre su producto al usuario final. Por lo tanto, es crucial conocer los tipos de licenciamiento*:*

|  |
| --- |
| Infografía interactiva  CF012\_8\_Componentes de software (1) |

El incumplimiento del uso de *software* no licenciado puede acarrear sanciones administrativas de hasta 200 salarios mínimos legales mensuales vigentes (200 SMMLV) para todos los administradores de la empresa, según lo estipulado por la Superintendencia de Sociedades.

* 1. **Seguridad**

La seguridad de las aplicaciones se refiere a las medidas implementadas a nivel de aplicación, incluida la codificación, cuyo objetivo es prevenir el acceso no autorizado, así como el robo o secuestro de datos o códigos dentro de la aplicación. Esto abarca las consideraciones de seguridad que se deben tener en cuenta durante el desarrollo y diseño de aplicaciones, además de los sistemas y técnicas empleados para proteger las aplicaciones una vez distribuidas.

|  |  |
| --- | --- |
| Concepto de ilustración de seguridad cibernética con personajes. | Para asegurar las aplicaciones, es fundamental incluir medidas en el *hardware, software* y procedimientos que minimicen las vulnerabilidades de seguridad. La estrategia de publicación de servicios de información requiere el uso de un conjunto de protocolos o procedimientos, y para garantizar la seguridad, es esencial realizar pruebas pertinentes. |

La seguridad de las aplicaciones implica desarrollar, incorporar y probar características de seguridad destinadas a proteger las aplicaciones de amenazas, como la modificación y el acceso no autorizados. Es crucial asegurar esto, ya que las aplicaciones modernas suelen estar disponibles a través de múltiples redes y conectadas a la nube, lo que incrementa las vulnerabilidades a las que están expuestas.

|  |
| --- |
| Pestañas verticales  CF012\_8.1\_Seguridad (1) |

En conclusión, los tipos básicos de seguridad de la aplicación son la autenticación, el cifrado y el registro de incidencias. Sin embargo, según el contexto del funcionamiento de la aplicación, se deben considerar otros aspectos relevantes.

|  |
| --- |
| Pestañas  CF012\_8.1\_Seguridad (2) |

* 1. ***API Firebase***

Las organizaciones están adoptando modelos de arquitectura de *software* basados en microservicios. Un microservicio es un servicio pequeño destinado a una función específica, y para realizar procesos más complejos, se coordinan múltiples microservicios. Esto permite que el mismo servicio, aunque publicado en internet, no sea para usuarios finales sino para su uso por aplicaciones. De esta manera, una aplicación móvil puede utilizar ese servicio de la misma forma que una aplicación web o un *chatbot.*

|  |  |
| --- | --- |
| Programa de red de aplicación API de concepto de ilustración isométrica plana | **Definición de *API*:** imaginemos que una empresa necesita que sus clientes registren solicitudes a través de una página web, una aplicación móvil y un *chatbot*. La principal ventaja de los microservicios es que, si se desea cambiar una regla de negocio, como aplicar ciertos descuentos, no es necesario modificar los tres sistemas, sino solo los microservicios involucrados. Estos se publican mediante una API (Interfaz de Programación de Aplicaciones), que es un conjunto de subrutinas, funciones y procedimientos que se ofrecen para ser utilizados por otro *software,* actuando como una capa de abstracción que oculta los detalles de codificación. |
| Ilustración de vector de código abierto Concepto de lenguaje de programación Interfaz de plataforma de protocolo de desarrollador con información de código Script de software digital | ***Firebase*:** es una plataforma en la nube de Google para el desarrollo de aplicaciones web y móviles, creada en 2011 y adquirida por Google en 2014. Su principal función es facilitar la creación de aplicaciones web y móviles, proporcionando herramientas que agilizan el desarrollo sin sacrificar calidad ni seguridad. Para utilizar los servicios de *Firebase,* es necesario consumir la API proporcionada por Google. |



La herramienta principal de *Firebase* son las bases de datos en tiempo real, alojadas en la nube y de tipo NoSQL en formato JSON. Este servicio permite alojar y gestionar datos de la aplicación en tiempo real, manteniéndolos actualizados incluso si el usuario no realiza ninguna acción. *Firebase* también envía automáticamente eventos que activan funcionalidades de las aplicaciones cuando los datos cambian, lo que facilita el desarrollo de aplicaciones con funcionalidades en tiempo real.

En cuanto a seguridad, *Firebase* ofrece un sistema de autenticación que permite tanto el inicio de sesión mediante correo electrónico y contraseña, como el acceso utilizando perfiles de otros sistemas externos, generalmente redes sociales.

|  |  |
| --- | --- |
| Cursor with solid fill | **Creación de una base de datos en *Firebase***  El proceso básico para crear una base de datos en *Firebase,* se presenta en el siguiente Anexo. |

Uno de los principales inconvenientes de *Firebase* podría ser la necesidad de realizar pagos, ya que no todas sus funcionalidades son gratuitas y la versión sin costo limita el número de usuarios. Sin embargo, para proyectos en sus primeras etapas, la versión gratuita de *Firebase* suele ser más que suficiente.

**8.3 Herramientas**

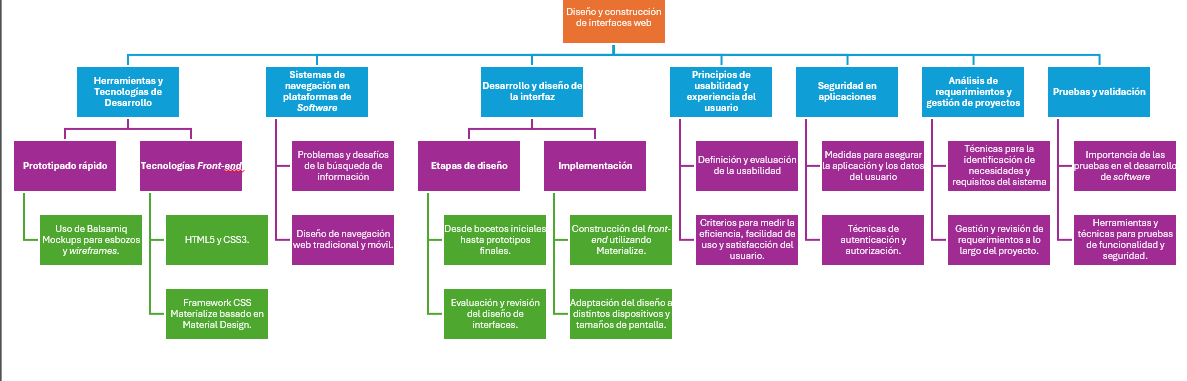
Durante el desarrollo de proyectos que emplean microservicios, las pruebas de *API* son esenciales para garantizar la ejecución, funcionalidad, confiabilidad y seguridad de la aplicación, tanto en la fase de desarrollo como en la de consumo de la *API*. A continuación, se presentan varias herramientas de libre distribución que facilitan el proceso de prueba y diseño de *APIS* publicadas a través de los diferentes protocolos web.

|  |
| --- |
| Tarjetas  CF012\_8.3 Herramientas |

Se puede concluir que existe una amplia variedad de herramientas disponibles para todas las fases de desarrollo e implementación de aplicaciones web y móviles. Estas herramientas ofrecen diversos tipos de licenciamiento, y muchas de ellas son de libre distribución, lo que permite iniciar un proyecto sin incurrir en costos iniciales significativos.

1. **SÍNTESIS**

A continuación, se presenta una síntesis de la temática estudiada en el componente formativo.



1. **ACTIVIDADES DIDÁCTICAS (Se debe incorporar mínimo 1, máximo 2)**

|  |  |
| --- | --- |
| DESCRIPCIÓN DE ACTIVIDAD DIDÁCTICA | |
| Nombre de la Actividad | Desarrollo de interfaces gráficas en *software* |
| Objetivo de la actividad | Identificar los elementos esenciales en el proceso de desarrollo de interfaces gráficas para aplicaciones de *software.* |
| Tipo de actividad sugerida | Relacionar conceptos |
| Archivo de la actividad  (Anexo donde se describe la actividad propuesta) | *CF012\_Actividad didáctica* |

1. **MATERIAL COMPLEMENTARIO:**

|  |  |  |  |
| --- | --- | --- | --- |
| Tema | Referencia APA del Material | Tipo de material  (Video, capítulo de libro, artículo, otro) | Enlace del Recurso o  Archivo del documento o material |
| Sistema de interacción en plataformas de *software* | UXTips. (2018). ¿Qué es Arquitectura de la Información? [Vídeo]. YouTube. | Video | <https://www.youtube.com/watch?v=twVvK1eYmig> |
| Estructura de la interfaz de la aplicación | yReyALD. (2020). BALSAMIQ MOCKUPS / Instalación y maquetación de una aplicación móvil - FACIL (música GOT de fondo) [Vídeo]. YouTube. | Video | <https://www.youtube.com/watch?v=idX3aFx67Vc> |
| Estructura de la interfaz de la aplicación | Bluuweb! (2018). Materialize CSS Material Design Framework [Curso tutorial Español] [Vídeo]. YouTube. | Video | <https://www.youtube.com/watch?v=YJKWkUi2r5g> |
| Construcción de la estructura del *front-end* | Materialize.Zip (1.0.0). (2021). [Framework CSS]. | Archivo | <https://github.com/Dogfalo/materialize/releases/download/1.0.0/materialize-v1.0.0.zip> |
| Construcción de la estructura del *front-end* | Materialize. (2021). Getting Started - Materialize. | Guía de instalación | <https://materializecss.com/getting-started.html> |
| Diseño gráfico: herramientas de prototipado | FalconMasters. (2018). *Curso de Adobe XD | 1. Introducción y que Aprenderemos en el Curso* [Video] Youtube. | Video | <https://www.youtube.com/watch?v=5HfqCSm9k3Y&feature=youtu.be> |
| Diseño gráfico: herramientas de prototipado | EDteam. (2017). *Minicurso de Balsamiq Mockups (1) - Los wireframes y la interfaz*. [Video] Youtube. | Video | <https://www.youtube.com/watch?v=4WqfxF1QjTM&feature=youtu.be> |
| Diseño gráfico: herramientas de prototipado | Aprende, G. (2020). *COMO USAR CANVA: Canva tutorial - Aprende a diseñar FÁCIL y de manera PROFESIONAL*. [Video] Youtube. | Video | <https://www.youtube.com/watch?v=kdE6ZmT_5MI> |
| Diseño gráfico: herramientas de prototipado | Conde, J. (2016). *01.- Material Design con Android Studio. ¿Qué es Material Design?* [Video] Youtube. | Video | <https://www.youtube.com/watch?v=wt0Jzc9UHNw> |
| *API Firebase* | Novato, P. (2019). *Firebase 01.- Que es Firebase*. [Video] Youtube. | Video | <https://www.youtube.com/watch?v=0Bhrwc8VfGw> |

1. **GLOSARIO:**

|  |  |
| --- | --- |
| TÉRMINO | SIGNIFICADO |
| Accesibilidad: | la accesibilidad web hace referencia a un diseño web que permite que las personas con cierta limitación puedan percibir, entender, navegar e interactuar con la Web, aportando a su vez contenidos. La accesibilidad web además beneficia a otras personas, incluyendo a adultos mayores que han visto mermadas sus habilidades a consecuencia de la edad. |
| Material Design: | se puede entender como un lineamiento o principio de diseño enfocado en la visualización del sistema operativo Android. Este concepto también abarca para el diseño en la web y en cualquier plataforma. Fue desarrollado por Google. |
| Usabilidad: | dentro del contexto de la informática y tecnologías, es un concepto utilizado para mencionar la facilidad con que las personas pueden utilizar una herramienta *software* particular o cualquier otro objeto tecnológico fabricado por humanos con el fin de alcanzar un objetivo concreto o satisfacer una necesidad. |
| *Wereable:* | dispositivo conectado que se puede llevar puesto y que se conecta al teléfono móvil. |
| *Wireframing:* | es un esquema de página o plano de pantalla, como guía visual que representa el esqueleto o estructura visual de una interfaz. |

1. **REFERENCIAS BIBLIOGRÁFICAS:**

Aula formativa. (2019, julio 29). ¿Qué es el Material Design?. [Web log post]. blog.aulaformativa.com. <https://blog.aulaformativa.com/que-es-el-material-design>

Calvo, L. (2015). Sistemas de navegación. Sistemas de Navegación, Madrid, España.

Curso Wordpress. (2014). Mapa del Sitio, Navegación y Contenidos. Curso WordPress Gratis. <https://cursowpress.wordpress.com/mapa-del-sitio-y-estrategia-de-navegacion/>

Ferraris, J. C. (2018). Patrones básicos de Navegación en Apps Móviles. Medium. <https://medium.com/@juancaferraris/patrones-b%C3%A1sicos-de-navegaci%C3%B3n-en-apps-m%C3%B3viles-5b0b160ed1bb>

IONOS. (2021). Tendencias de diseño web. IONOS Digital Guide. <https://www.ionos.es/digitalguide/paginas-web/diseno-web/tendencias-de-diseno-web/>

Material.io. (2021). Material Design. <https://material.io/design/color/the-color-system.html#color-usage-and-palettes>

Materialize. (2021). Getting Started - Materialize. <https://materializecss.com/getting-started.html>

1. **CONTROL DEL DOCUMENTO**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Nombre | Cargo | Dependencia | Fecha |
| Autor (es) | Santiago Muñoz de la Rosa | Experto temático | Regional Cauca - Centro de Teleinformática y Producción Industrial | Noviembre 2022 |
| Peter Emerson Pinchao | Experto Temático | Regional Popayán - Centro de Teleinformática y Producción Industrial | Noviembre 2022 |
| Paola Alexandra Moya | Evaluadora instruccional | Regional Antioquia - Centro de Servicios de Salud | Abril 2024 |
|  | Olga Constanza Bermúdez Jaimes | Responsable Línea de Producción Antioquia | Regional Antioquia - Centro de Servicios de Salud | Abril 2024 |

1. **CONTROL DE CAMBIOS**

**(Diligenciar únicamente si realiza ajustes a la Unidad Temática)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Nombre | Cargo | Dependencia | Fecha | Razón del Cambio |
| Autor (es) |  |  |  |  |  |