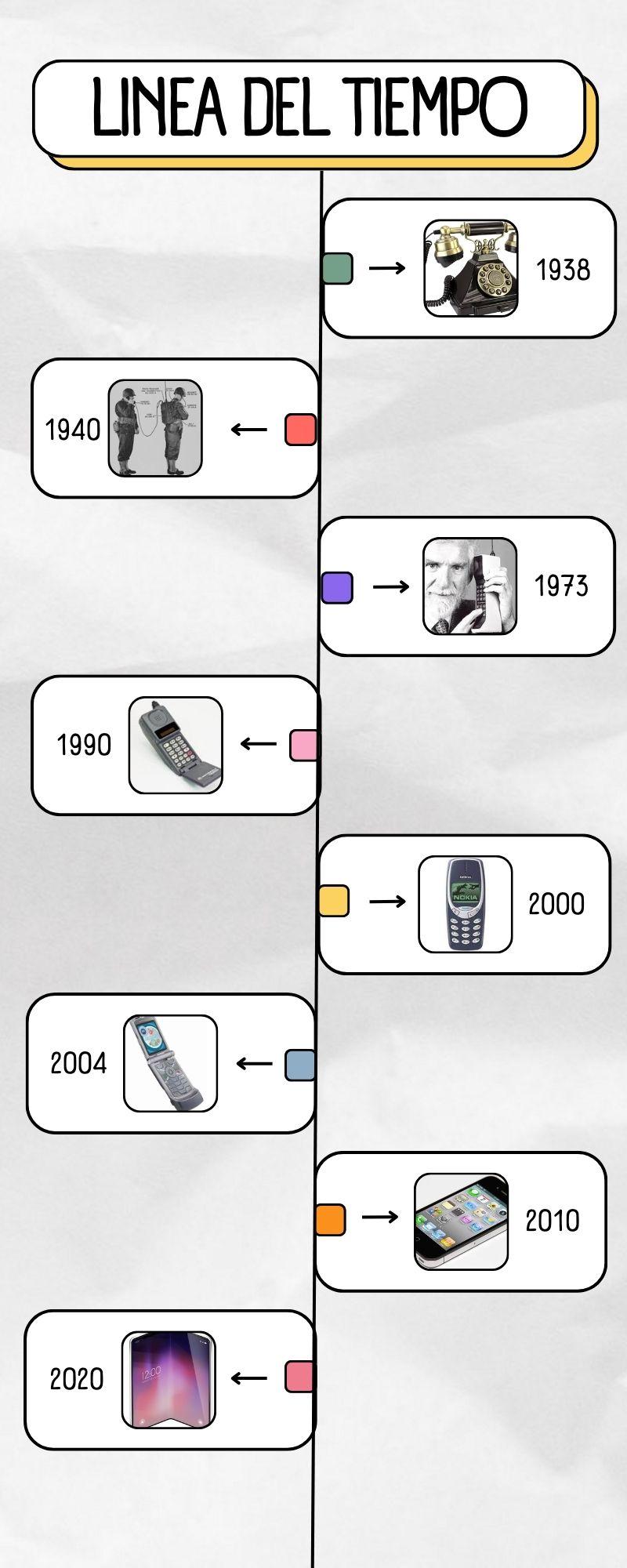
**TALLER**

1. De acuerdo a la historia de los dispositivos móviles elaborar una linea de tiempo



1. Cuadro comparativo entre android y ios



1. Ventajas y desventajas de una aplicación nativa

**Ventajas**

* Alto rendimiento
* Acceso a funciones del dispositivo
* Experiencia de usuario mejorada
* Integración con el ecosistema
* Rendimiento sin conexión
* Seguridad mejorada

**Desventajas**

* Desarrollo y mantenimiento más costosos
* Requiere habilidades de desarrollo específicas
* Actualizaciones y parches más complicados
* Menos alcance potencial
* Aprobación de la tienda de aplicaciones
* Problemas de compatibilidad de versiones

1. Qué es un aplicación Nativa, Híbrida, Multiplataforma

**Nativa:** Es aquella que ha sido desarrollada en el lenguaje de programación específico de un sistema operativo. Es decir, si hablamos de Android, la app nativa habrá sido desarrollada con el lenguaje de programación Kotlin o Java.

**Híbrida:** Tienen el mismo rendimiento y la misma experiencia de usuario que las aplicaciones nativas, pero con un costo inferior. Los desarrolladores pueden crearlas con tecnologías y lenguajes de desarrollo de aplicaciones de uso común, como JavaScript, CSS y HTML5.

**Multiplataforma:** Es la práctica de desarrollar una aplicación para que pueda ser distribuida en más de una plataforma: dispositivos móviles o tabletas, o un ordenador.

1. ¿Estás de acuerdo o no, en que el diseño debe estar presente en todo el ciclo de desarrollo de software? Justifica tu respuesta.

* Sí, estoy de acuerdo en que el diseño debe estar presente en todo el ciclo de desarrollo de software. El diseño desempeña un papel crucial en la creación de software de alta calidad y en la satisfacción de las necesidades de los usuarios porque el diseño ayuda a definir claramente los objetivos y requisitos del software antes de comenzar la implementación.

1. Explica los 5 pasos del proceso del pensamiento de diseño. Luego aplícalo a un ejemplo.

* Empatiza
* Define
* Idea
* Prototipar
* Testear
* Ejemplo: Una empresa tiene como finalidad de cada proyecto satisfacer la necesidad de los usuarios entonces en **Empatizar** aprende la necesidad que tiene el usuario, en **Define** se filtra información y se identifican las soluciones para así darle un resultado óptimo al usuario, la **Idea** es generar y comenzar con la parte creativa para el desarrollo a la solución, el **Prototipar** es crear el tipo de solución que se le va a dar al usuario y **Testear** es hacerle pruebas y corregir posibles errores que tenga el prototipo para darle el resultado final al usuario y así este se vaya satisfecho.

1. ¿Qué es Design Thinking?

* Es un método de trabajo que enfrenta y soluciona los retos y problemas que surgen en las empresas a base de creatividad, multidisciplinariedad y trabajo en equipo.

1. Explica y aplica con un ejemplo las etapas del Design Thinking

* El design thinking es un procedimiento de trabajo en el que las soluciones de problemas y el desarrollo de nuevos productos o servicios se realizan de manera grupal.
* Por ejemplo: Las ventas de una compañía de alimentos han disminuido. Por eso, los integrantes del equipo de design thinking analizan qué tipo de comida es el más comprado y llegan a la conclusión de que los clientes prefieren los productos elaborados con ingredientes orgánicos. Al mes siguiente, la compañía desarrolla una línea saludable de alimentos con ingredientes orgánicos y las ventas mejoran.

1. ¿Qué es ser diseñador inclusivo? ¿Por qué es importante? ¿Estás de acuerdo en que debemos ser diseñadores inclusivos? Justifica tu respuesta.

* nace con el objetivo de ofrecer la misma calidad de producto digital a una amplia diversidad de personas, independientemente de su género, edad o capacidades.
* Ser un diseñador inclusivo de software es importante por varias razones fundamentales:

1. Accesibilidad para todos: El diseño inclusivo se centra en garantizar que el software sea accesible para todas las personas, independientemente de sus capacidades, discapacidades o limitaciones. Esto permite que un amplio espectro de usuarios pueda beneficiarse del software, lo que es esencial para cumplir con principios éticos de igualdad y equidad.

2. Cumplimiento legal y regulatorio: En muchos lugares, existen leyes y regulaciones que requieren que el software sea accesible para personas con discapacidades.

* Si, porque ser un diseñador inclusivo es esencial para crear software que sea verdaderamente accesible y beneficioso para una amplia variedad de usuarios. Promover la inclusión es una parte importante de la responsabilidad ética y profesional en el diseño de software.

1. De acuerdo a las herramientas que podemos utilizar para ser diseñadores inclusivos, Escoge 2 herramientas y realiza una investigación de cómo funciona y cómo se puede implementar en una aplicación web y una móvil.

* Sketch: Es una herramienta de diseño UX/UI muy conocida y querida. Es fácil trabajar con ella y te permite realizar cambios universales así como cambios en el tamaño o alinear, superponer y cambiar estilos de texto. La implementación de cambios como estos ahorra mucho tiempo a los diseñadores y garantiza la coherencia en todo el producto. También hay muchos complementos fáciles de integrar disponibles para Sketch que amplían lo que puede hacer la herramienta. Además, tiene una opción incorporada para importar, editar e inspeccionar su propio código.
* Implementa en un aplicación web
* Exportar elementos de la interfaz de usuario: Puedes diseñar tus elementos de interfaz de usuario, como botones, iconos y fondos, en Sketch y luego exportarlos en formatos de imagen (por ejemplo, PNG, SVG) o archivos de código (como CSS o SVG) que puedas incorporar directamente en tu aplicación web.
* Crear prototipos interactivos: Utiliza Sketch para diseñar las pantallas de tu aplicación web y luego crea prototipos interactivos. Puedes usar complementos de Sketch como Craft by InVision
* Implementa en una aplicación móvil
* Diseña las pantallas y elementos de la interfaz de usuario: Utiliza Sketch para crear los diseños de las pantallas de tu aplicación móvil, como la pantalla de inicio, la pantalla de inicio de sesión, las páginas de contenido, los formularios, los botones, los menús y otros elementos de la interfaz de usuario.
* Crea prototipos interactivos: Puedes utilizar Sketch para crear prototipo
* Exporta recursos y activos: Exporta los elementos de diseño
* Colabora con el equipo de desarrollo: Comparte tus diseños y prototipos con el equipo de desarrollo que trabajará en la implementación de la aplicación móvil.
* Considere la adaptabilidad: Ten en cuenta que los diseños deben ser adaptables a diferentes tamaños de pantalla y orientaciones.
* Figma: Es una popular herramienta de diseño basada en la nube. Es similar en diseño y funcionalidad a Sketch, pero es altamente colaborativa y es una de las mejores herramientas de creación de prototipos para la colaboración en equipo. Todos, desde diseñadores UX hasta desarrolladores, pueden acceder y trabajar en un proyecto en Figma. Con la integración de Slack y la opción de compartir proyectos, páginas, marcos y archivos con compañeros de trabajo seleccionados, Figma es una poderosa herramienta UX basada en equipos. Una característica particularmente divertida y fácil de usar es la opción de hacer clic en el avatar de cualquier persona para ver su vista.
* Implementa en una aplicación web
* Diseño de la interfaz de usuario: Utiliza Figma para diseñar la interfaz de usuario de tu aplicación web. Esto incluye crear diseños de páginas, elementos gráficos, botones, iconos, formularios y cualquier otro componente visual que formará parte de tu aplicación.
* Creación de prototipos: Figma te permite crear prototipos interactivos de tu aplicación web. Puedes vincular diferentes pantallas y elementos para simular la navegación y las interacciones del usuario.
* Implementa en una aplicación móvil
* Creación de prototipos: Figma te permite crear prototipos interactivos de tu aplicación móvil. PAG
* Colaboración en equipo: Figma es una herramienta de diseño colaborativo que permite a varios miembros del equipo trabajar en el mismo proyecto en tiempo real. Invita a diseñadores, desarrolladores y otros interesados ​​a colaborar en el diseño y proporcionar retroalimentación.
* Exportación de activos: Desde Figma, puedes exportar elementos gráficos y recursos de diseño en diferentes formatos, como imágenes (PNG, SVG), archivos de diseño (Figma) o incluso generar código CSS para los estilos de tu aplicación móvil.
* Integración con herramientas de desarrollo: Figma ofrece integraciones con varias herramientas de desarrollo que facilitan la transferencia de activos y diseños desde Figma a las herramientas de desarrollo que estés utilizando para crear la aplicación móvil.
* Pruebas de usabilidad: Utiliza prototipos en Figma para realizar pruebas de usabilidad con usuarios reales y recopilar comentarios para mejorar la interfaz de usuario de tu aplicación móvil.
* Mantenimiento y actualización: A medida que tu aplicación móvil evolucione, puedes usar Figma para mantener actualizados los diseños y asegurarte de que los cambios se reflejan en el diseño original.

1. Explica la importancia de la empatía con el usuario en el desarrollo del software.

* La empatía con el usuario desempeña un papel fundamental en el desarrollo de software por varias razones importantes
* Mejora la usabilidad: La empatía permite a los diseñadores y desarrolladores comprender las necesidades, preferencias y desafíos de los usuarios. Esto resulta en interfaces de usuario más intuitivas y funcionales, lo que mejora la usabilidad del software.
* Satisfacción del cliente: Cuando los desarrolladores demuestran empatía hacia los usuarios, estos se sienten valorados y escuchados. La satisfacción del cliente aumenta, lo que puede llevar a la retención de clientes y positivas.
* Reducción de errores y retrabajos: Comprender las necesidades del usuario desde el principio ayuda a evitar errores.

1. Qué es y cómo se implementa un mapa de empatía.

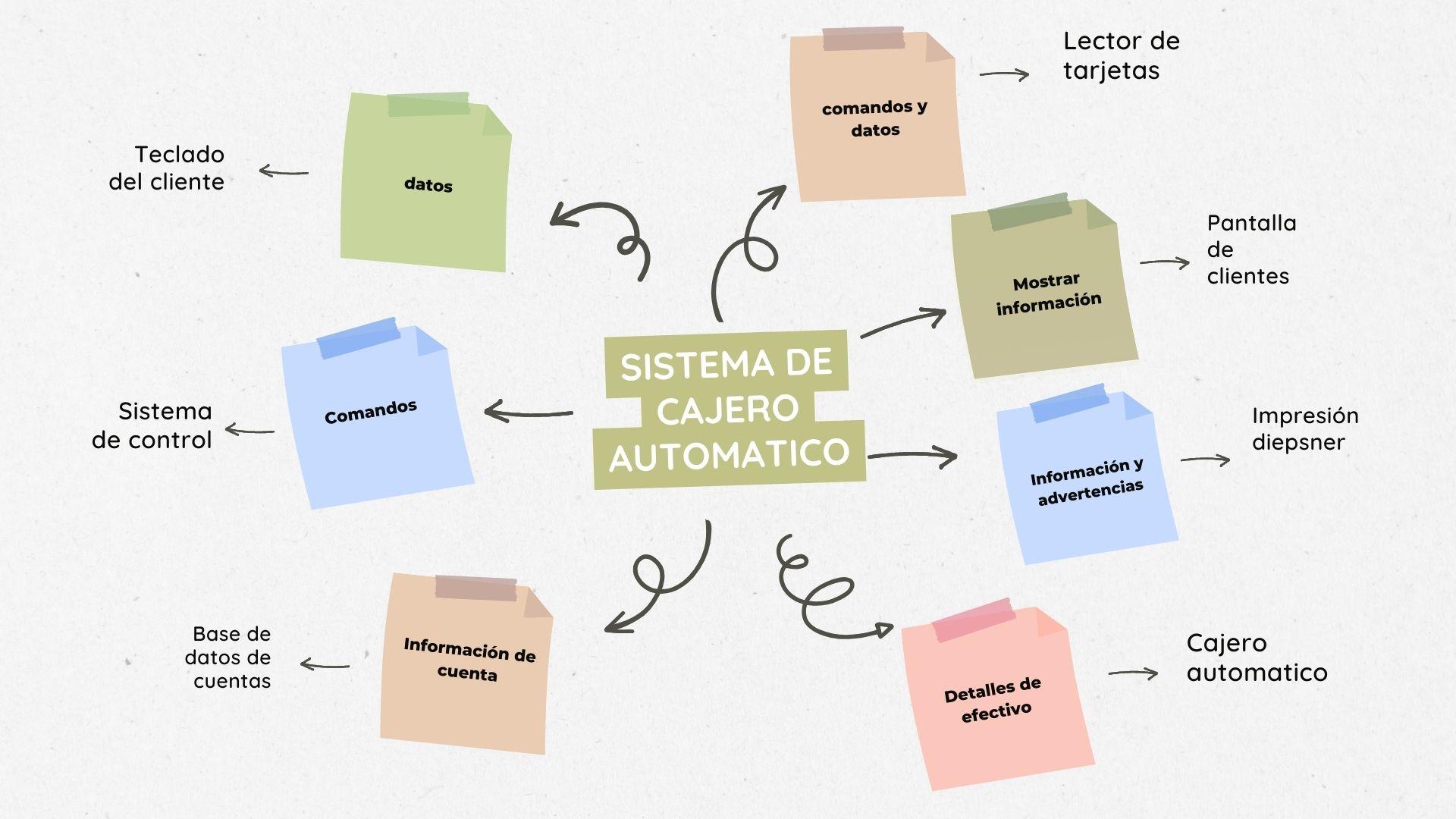
* Un mapa de empatía es una herramienta visual que se utiliza para comprender mejor a un grupo específico de usuarios o clientes. Ayuda a los equipos de diseño y desarrollo a profundizar en las necesidades, deseos, pensamientos, sentimientos y comportamientos de ese grupo de usuarios. El objetivo principal de un mapa de empatía es crear empatía con los usuarios y desarrollar una comprensión más profunda de su experiencia y perspectiva. Un mapa de empatía típico se divide en secciones que representan diferentes aspectos de la experiencia del usuario.
* Definir a los usuarios objetivo: Comienza por identificar a quiénes están dirigidos tus esfuerzos. Define un grupo específico de usuarios o clientes a los que deseas comprender. Esto puede incluir detalles demográficos, como edad, género, ubicación, pero también aspectos más relacionados con el comportamiento y las necesidades específicas.
* Llenar las secciones del mapa de empatía: Pensamientos y Sentimientos: En esta sección, describe los pensamientos y sentimientos que los usuarios experimentan en relación con el problema o el contexto en el que están utilizando.

1. ¿Qué es un diagrama de contexto? ¿Y qué es el context canvas?

* Un diagrama de contexto es el nivel más alto en un Diagrama de Flujo de Datos. Es una herramienta popular para los analistas de negocios, los que la utilizan para entender los detalles y los límites del sistema que se va a diseñar en un proyecto. Señala el flujo de información entre el sistema y los componentes externos.
* Context canvas es una DOMString que contiene el identificador del contexto que define el contexto de dibujo asociado a el lienzo. Los posibles valores son: "2d ", dando lugar a la creación de un objeto CanvasRenderingContext2D que representa un contexto de renderizado de dos dimensiones.

1. Elabora un cuadro comparativo entre el diagrama de contexto y el context canvas
2. Elabora un ejemplo de diagrama de contexto y context canvas

* Se usa un diagrama de contexto para mostrar el software del Sistema de Cajero automático y el hardware que interactúa con él. Las flechas indican las direcciones y los tipos de datos que fluyen entre el software y cada uno de los componentes de hardware.



1. Con base a la explicación. Elabora un ejemplo de personas y usuario.